Smarter. Greener. Together.

DVP-PLC Application Manual (Programming)

DVP-PLC Application Manual
(Programming)

Smarter. Greener. Together.

DVP-PLC Application Manual: Programming

Table of Contents

Chapter 9 Application Instructions API 150-199

Chapter 10 Application Instructions API 202-328

Chapter 11 Appendix

The models that every series includes are as follows.

Series	Model name
DVP-ES DVP-EC	DVP14ES00R2, DVP14ES00T2, DVP14ES01R2, DVP14ES01T2, DVP24ES00R, DVP24ES00R2, DVP24ES00T2, DVP24ES01R2, DVP24ES01T2, DVP24ES11R2, DVP30ES00R2, DVP30ES00T2, DVP32ES00R, DVP32ES00R2, DVP32ES00T2, DVP32ES01R2, DVP32ES01T2, DVP40ES00R2, DVP40ES00T2, DVP60ES00R2, DVP60ES00T2 DVP10EC00R3, DVP10EC00T3, DVP14EC00R3, DVP14EC00T3, DVP16EC00R3, DVP16EC00T3, DVP20EC00R3, DVP20EC00T3, DVP24EC00R3, DVP24EC00T3, DVP30EC00R3, DVP30EC00T3, DVP32EC00R3, DVP32EC00T3, DVP40EC00R3, DVP40EC00T3, DVP60EC00R3, DVP60EC00T3 The EC series stated in this manual is V8.60 or later versions. It is an updated series with a programming capacity of 8K. When operating on ISPSoft, select EC3-8K for this series. Here also uses the model name EC3-8K to distinguish it from the previous versions.
DVP-EX	DVP20EX00R2, DVP20EX00T2, DVP20EX11R2
DVP-SS	DVP14SS11R2, DVP14SS11T2 (Year 2013 discontinued products; deleted the relevant information from this manual)
DVP-SA	DVP12SA11R, DVP12SA11T (Year 2013 discontinued products; deleted the relevant information from this manual)
DVP-SX	DVP10SX11R, DVP10SX11T
DVP-SC	DVP12SC11T (Year 2013 discontinued products; deleted the relevant information from this manual)
DVP-EH2	DVP16EH00R2, DVP16EH00T2, DVP20EH00R2, DVP20EH00T2, DVP32EH00M2, DVP32EH00R2, DVP32EH00T2, DVP40EH00R2, DVP40EH00T2, DVP48EH00R2, DVP48EH00T2, DVP60EH00T2, DVP64EH00R2, DVP64EH00T2, DVP80EH00R2, DVP80EH00T2, DVP32EH00R2-L, DVP32EH00T2-L (Year 2015 discontinued products; deleted the relevant information from this manual)
DVP-SV	DVP28SV11R, DVP28SV11T (Year 2015 discontinued products; deleted the relevant information from this manual)
DVP-EH3	DVP16EH00R3, DVP16EH00T3, DVP20EH00R3, DVP20EH00T3, DVP32EH00M3, DVP32EH00R3, DVP32EH00T3, DVP40EH00R3, DVP40EH00T3, DVP48EH00R3, DVP48EH00T3, DVP60EH00T3, DVP64EH00R3, DVP64EH00T3, DVP80EH00R3, DVP80EH00T3, DVP32EH00R3-L, DVP32EH00T3-L
DVP-SV2	DVP28SV11R2, DVP28SV11T2, DVP24SV11T2

Note: This manual does not include descriptions for the discontinued products.

Foreword: Background and Functions of PLC

PLC (Programmable Logic Controller) is an electronic device, previously called “sequence controller”. In 1978, NEMA (National Electrical Manufacture Association) in the United States officially named it as “programmable logic controller”. PLC reads the status of the external input devices, e.g. keypad, sensor, switch and pulses, and execute by the microprocessor logic, sequential, timing, counting and arithmetic operations according the status of the input signals as well as the pre-written program stored in the PLC. The generated output signals are sent to output devices as the switch of a relay, electromagnetic valve, motor drive, control of a machine or operation of a procedure for the purpose of machine automation or processing procedure. The peripheral devices (e.g. personal computer/handheld programming panel) can easily edit or modify the program and monitor the device and conduct on-site program maintenance and adjustment. The widely used language in designing a PLC program is the ladder diagram.
With the development of the electronic technology and wider applications of PLC in the industry, for example in position control and the network function of PLC, the input/output signals of PLC include DI (digital input), AI (analog input), PI (pulse input), NI (numeric input), DO (digital output), AO (analog output), and PO (pulse output). Therefore, PLC will still stand important in the industrial automation field in the future.

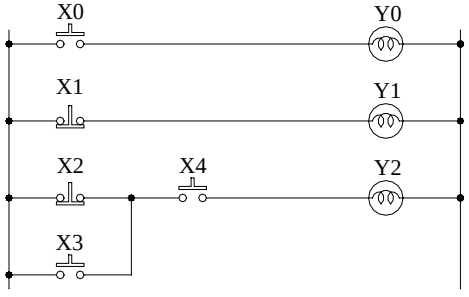
1.1 The Working Principles of Ladder Diagram

The ladder diagram was a diagram language for automation developed in the WWII period, which is the oldest and most widely adopted language in automation. In the initial stage, there were only A (normally open) contact, B (normally closed) contact, output coil, timer and counter…the sort of basic devices on the ladder diagram (see the power panel that is still used today). After the invention of programmable logic controllers (PLC), the devices displayable on the ladder diagram are added with differential contact, latched coil and the application commands which were not in a traditional power panel, for example the addition, subtraction, multiplication and division operations.
The working principles of the traditional ladder diagram and PLC ladder diagram are basically the same. The only difference is that the symbols on the traditional ladder diagram are more similar to its original form, and PLC ladder diagram adopts the symbols that are easy to recognize and shown on computer or data sheets. In terms of the logic of the ladder diagram, there are combination logic and sequential logic.

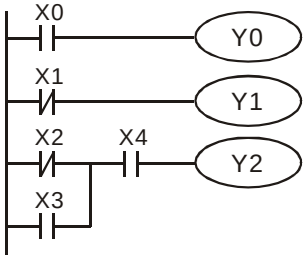
1. Combination Logic

   Examples of traditional ladder diagram and PLC ladder diagram for combination logic:

Traditional Ladder Diagram



PLC Ladder Diagram



Row 1: Using a normally open (NO) switch X0 (“A” switch or “A" contact). When X0 is not pressed, the contact will be open loop (Off), so Y0 will be Off. When X0 is pressed, the contact will be On, so Y0 will be On.

Row 2: Using a normally closed (NC) switch X1 (“B” switch or “B” contact). When X1 is not pressed, the contact will be On, so Y1 will be On. When X1 is pressed, the contact will be open loop (Off), so Y1 will be Off.

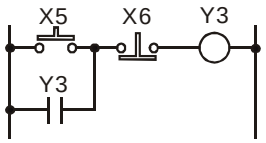
Row 3: The combination logic of more than one input devices. Output Y2 will be On when X2 is not pressed or X3 and X4 are pressed.

2. Sequential Logic

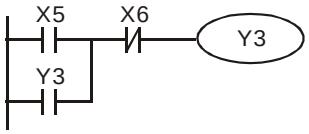
Sequential logic is a circuit with "draw back” structure, i.e. the output result of the circuit will be drawn back as an input criterion. Therefore, under the same input criteria, different previous status or action sequence will follow by different output results.

Examples of traditional ladder diagram and PLC ladder diagram for sequential logic:

Traditional Ladder Diagram



PLC Ladder Diagram



When the circuit is first connected to the power, though X6 is On, X5 is Off, so Y3 will be Off. After X5 is pressed, Y3 will be On. Once Y3 is On, even X5 is released (Off), Y3 can still keep its action because of the draw back (i.e. the self-retained circuit). The actions are illustrated in the table below.

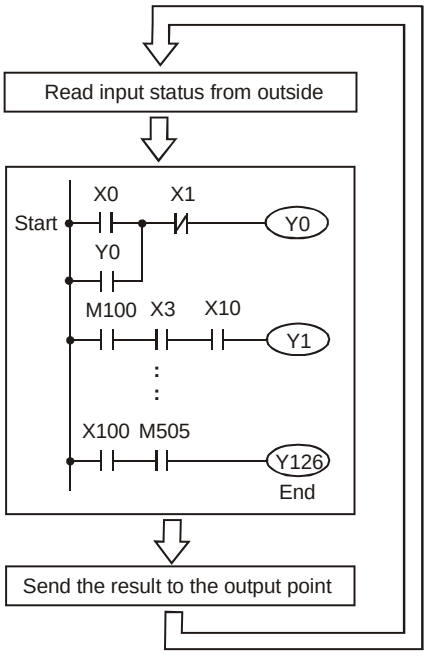
Device status / Action sequence	X5	X6	Y3
1	No action	No action	Off
2	Action	No action	On
3	No action	No action	On
4	No action	Action	Off
5	No action	No action	Off

From the table above, we can see that in different sequence, the same input status can result in different output results. For example, switch X5 and X6 of action sequence 1 and 3 do not act, but Y3 is Off in sequence 1 and On in sequence 3. Y3 output status will then be drawn back as input (the so-called “draw back”), making the circuit being able to perform sequential control, which is the main feature of the ladder diagram circuit. Here we only explain contact A, contact B and the output coil. Other devices are applicable to the same method. See Chapter 3 “Basic instructions” for more details.

1.2 Differences Between Traditional Ladder Diagram and PLC Ladder Diagram

Though the principles of traditional ladder diagram and PLC ladder diagram are the same, in fact, PLC adopts microcomputer to simulate the motions of the traditional ladder diagram, i.e. scan-check status of all the input devices and output coil and calculate to generate the same output results as those from the traditional ladder diagram based on the logics of the ladder diagram. Due to that there is only one microcomputer, we can only check the program of the ladder diagram one by one and calculate the output results according to the program and the I/O status before the cyclic process of sending the results to the output interface → re-reading of the input status → calculation → output. The time spent in the cyclic process is called the “scan time” and the time can be longer with the expansion of the program. The scan time can cause delay from the input detection to output response of the PLC. The longer the delay, the bigger the error is to the control. The control may even be out of control. In this case, you have to choose a PLC with faster scan speed. Therefore, the scan speed is an important specification requirement in a PLC. Owing to the advancement in microcomputer and ASIC (IC for special purpose), there has been great improvement in the scan speed of PLC nowadays. See the figure below for the scan of the PLC ladder diagram program.

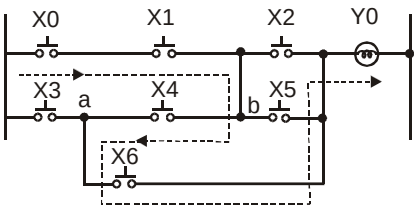
The output result is calculated based on the ladder diagram. (The result has not yet sent to the external output point, but the internal device will perform an immediate output.)



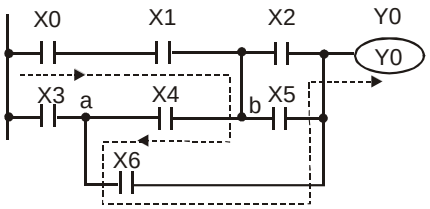
Executing in cycles

Besides the difference in the scan time, PLC ladder and traditional ladder diagram also differ in “reverse current”. For example, in the traditional ladder diagram illustrated below, when X0, X1, X4 and X6 are On and others are Off, Y0 output on the circuit will be On as the dotted line goes. However, the PLC ladder diagram program is scanned from up to down and left to right. Under the same input circumstances, the PLC ladder diagram editing tool WPLSoft will be able to detect the errors occurring in the ladder diagram.

Reverse current of traditional ladder diagram



Reverse current of PLC ladder diagram



Error detected in the third row

1.3 How to Edit Ladder Diagram

Ladder diagram is a diagram language frequently applied in automation. The ladder diagram is composed of the symbols of electric control circuit. The completion of the ladder diagram by the ladder diagram editor is the completion of the PLC program design. The control flow illustrated by diagram makes the flow more straightforward and acceptable for the technicians of who are familiar with the electric control circuit. Many basic symbols and actions in the ladder diagram come from the frequently-seen electromechanical devices, e.g. buttons, switches, relay, timer and counter, etc. in the traditional power panel for automation control.

Internal devices in the PLC: The types and quantity of the devices in the PLC vary in different brand names. Though the internal devices in the PLC adopt the names, e.g. transistor, coil, contact and so on, in the traditional electric control circuit, these physical devices do not actually exist inside the PLC. There are only the corresponding basic units (1 bit) inside the memory of the PLC. When the bit is “1”, the coil will be On, and when the bit is “0”, the coil will be Off. The normally open contact (NO or contact A) directly reads the value of the corresponding bit. The normally close contact (NC or contact B) reads the opposite state of the value of the corresponding bit. Many relays will occupy many bits. 8 bits equal a “byte”. 2 bytes construct a “word” and 2 words combined is “double word”. Byte, word or double words are used when many relays are processed (e.g. addition/subtraction, displacement) at the same time. The other two devices, timer and counter, in the PLC have coil, timer value and counter value and they have to process some values in byte, word or double word.

All kinds of internal devices in the value storage area in the PLC occupy their fixed amount of storage units. When you use these devices, you are actually read the contents stored in the form of bit, byte or word.

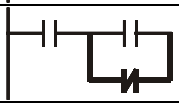
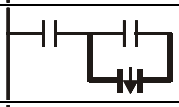
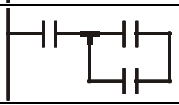
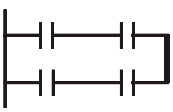
Introductions on the basic internal devices in the PLC (See Ch 2. Functions of Devices in DVP-PLC for more details.)

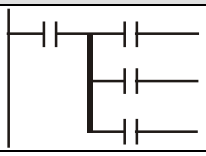
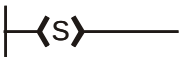
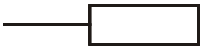
Device	Functions
Input relay	The input relay is an internal memory (storage) unit in the PLC corresponding to an external input point and is used for connecting to the external input switches and receiving external input signals. The input relay will be driven by the external input signals which make it “0” or “1". Program designing cannot modify the status of the relay, i.e. it cannot re-write the basic unit of a relay, nor can it force On/Off of the relay by HPP/WPLSoft. SA/SX/SC/EH2/SV/EH3/SV2 series MPU can simulate input relay X and force On/Off of the relay. But the status of the external input points will be updated and disabled, i.e. the external input signals will not be read into their corresponding memories inside PLC, but only the input points on the MPU. The input points on the extension modules will still operate normally. There are no limitations on the times of using contact A and contact B of the input relay. The input relays without corresponding input signals can only be left unused and cannot be used for other purposes. ☞ Device indication: X0, X1, …X7, X10, X11, … are indicated as X and numbered in octal form. The numbers of input points are marked on MPU and extension modules.
Output relay	The output relay is an internal memory (storage) unit in the PLC corresponding to an external output point and is used for connecting to the external load. The output relay will be driven by the contact of an input relay, contacts of other internal devices and the contacts on itself. A normally open contact of the output relay is connected to the external load. Same as the input

Device	Functions
	contacts, there are no limitations on the times of using other contacts of the output relay. The output relay without corresponding output signals can only be left unused and can be used as input relay if necessary. ☞ Device indication: Y0, Y1, …Y7, Y10, Y11, …are indicated as Y and numbered in octal form. The No. of output points are marked on MPU and extension modules.
Internal relay	The internal relay does not have connection with the external. It is an auxiliary relay inside the PLC with the functions same as those of the auxiliary (middle) relay in the electric control circuit. Every internal relay corresponds to a basic internal storage unit and can be driven by the contacts of the input relay, contacts of the output relay and the contacts of other internal devices. There are no limitations on the times of using the contacts of the internal relay and there will be no output from the internal relay, but from the output point. ☞ Device indication: M0, M1, …, M4095 are indicated as M and numbered in decimal form.
Step	DVP series PLC offers a step-type control program input method. STL instruction controls the transfer of step S, which makes it easy for the writing of the control program. If you do not use any step program in the control program, step S can be used as an internal relay M as well as an alarm point. ☞ Device indication: S0, S1, …S1023 are indicated as S and numbered in decimal form.
Timer	The timer is used for timing and has coil, contact and register in it. When the coil is On and the estimated time is reached, its contact will be enabled (contact A closed, contact B open). Every timer has its fixed timing period (unit: 1ms/10ms/100ms). Once the coil is Off, the contact iwlwl be disabled (contact A open, contact B closed) and the present value on the timer will become “0”. ☞ Device indication: T0, T1, …, T255 are indicated as T and numbered in decimal form. Different No. refers to different timing period.
Counter	The counter is used for counting. Before using the counter, you have to give the counter a set value (i.e. the number of pulses for counting). There are coil, contact and registers in the counter. When the coil goes from Off to On, the counter will regard it as an input of 1 pulse and the present value on the counter will plus “1”. We offer 16-bit and 32-bit high-speed counters for our users. ☞ Device indication: C0, C1, …, C255 are indicated as C and numbered in decimal form.
Data register	Data processing and value operations always occur when the PLC conducts all kinds of sequential control, timing and counting. The data register is used for storing the values or all kinds of parameters. Every register is able to store a word (16-bit binary value). Double words will occupy 2 adjacent data registers. ☞ Device indication: D0, D1, …, D11999 are indicated as D and numbered in decimal form.
File register	The file register is used for storing the data or all kinds of parameters when the data registers required for processing the data and value operations are insufficient. Every file register is able to store a 16-bit word. Double words will occupy 2 adjacent file registers. In SA/SX/SC series MPU, there are 1,600 file registers. In EH2/SV/EH3/SV2 series MPU, there are 10,000 file registers. There is not an actual device No. for a file register. The reading and writing of file registers should be executed by instructions API 148 MEMR, API 149 MEMW, or through the

Device	Functions
	peripheral device HPP02 and WPLSoft. ☞ Device indication: K0 ~ K9,999, numbered in decimal form.
Index register	E and F index registers are 16-bit data registers as other data registers. They can be read and written and can be used in word devices, bit devices or as a constant for index indication. ☞ Device indication: E0 ~ E7, F0 ~ F7 are indicated as E and F and numbered in decimal form.

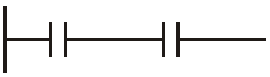
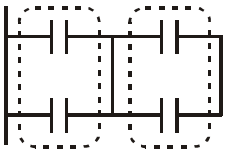
The structure of a ladder diagram:

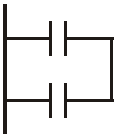
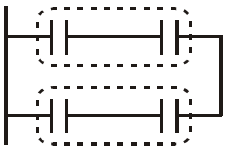
Structure	Explanation	Instruction	Devices Used
	Normally open, contact A	LD	X, Y, M, S, T, C
	Normally closed, contact B	LDI	X, Y, M, S, T, C
	Normally open in series connection	AND	X, Y, M, S, T, C
	Normally closed in series connection	ANI	X, Y, M, S, T, C
	Normally open in parallel connection	OR	X, Y, M, S, T, C
	Normally closed in parallel connection	ORI	X, Y, M, S, T, C
	Rising-edge trigger switch	LDP	X, Y, M, S, T, C
	Falling-edge trigger switch	LDF	X, Y, M, S, T, C
	Rising-edge trigger in series connection	ANDP	X, Y, M, S, T, C
	Falling-edge trigger in series connection	ANDF	X, Y, M, S, T, C
	Rising-edge trigger in parallel connection	ORP	X, Y, M, S, T, C
	Falling-edge trigger in parallel connection	ORF	X, Y, M, S, T, C
	Block in series connection	ANB	-
	Block in parallel connection	ORB	-

Structure	Explanation	Instruction	Devices Used
	Multiple output	MPS MRD MPP	-
	Coil driven output instruction	OUT	Y, M, S
	Step ladder	STL	S
	Basic instruction Application instruction	Application instructions	See Ch.3 for basic instructions (RST/SET and CNT/TMR) and Ch.5 ~ 10 for application instructions
	Inverse logic	INV	-

Block:

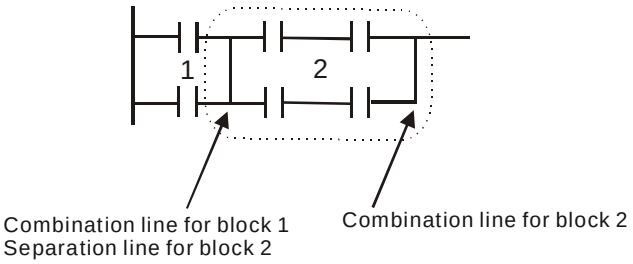
A block is a series or parallel operation composed of more than 2 devices. There are series block and parallel block.

Series block  

Parallel block  

Separation line and combination line:

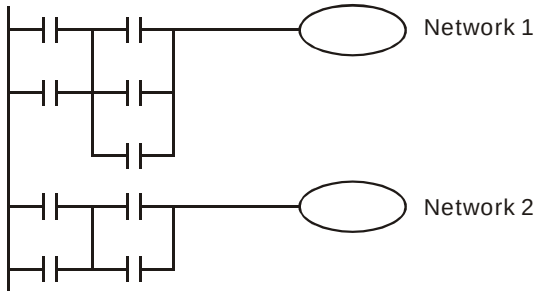
The vertical line is used for separating the devices. For the devices on the left, the vertical line is a combination line, indicating that there are at least 2 rows of circuits on the left connected with the vertical line. For the devices on the right, the vertical line is a separation line, indicating that there are at least 2 rows of circuits interconnected on the right side of the vertical line).



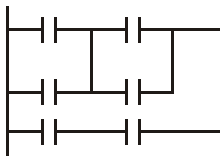
Network:

A complete block network is composed of devices and all kinds of blocks. The blocks or devices connectable by a vertical line or continuous line belong to the same network.

An independent network

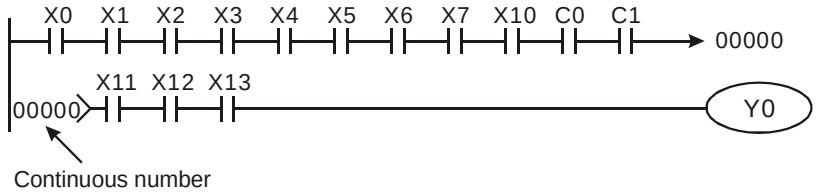


An incomplete network

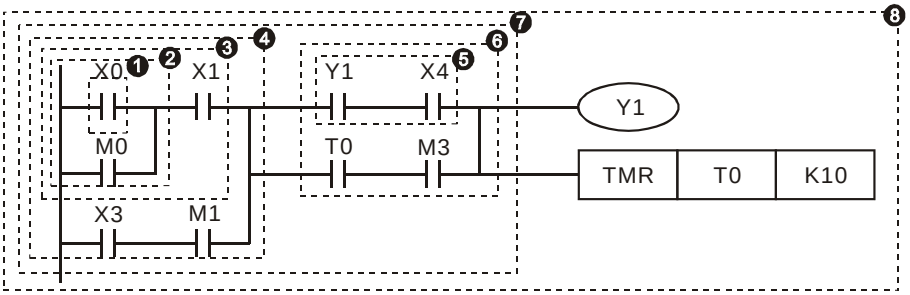


1.4 How to Edit a PLC Ladder Diagram

The editing of the program should start from the left power line and ends at the right power line, a row after another. The drawing of the right power line will be omitted if edited from WPLSoft. A row can have maximum 11 contacts on it. If 11 is not enough, you can continuously connect more devices and the continuous number will be generated automatically. The same input points can be used repeatedly. See the figure below:



The operation of the ladder diagram program is scanning from top left to bottom right. The coil and the operation frame of the application instruction belong to the output side in the program and are placed in the right if the ladder diagram. Take the figure below for example, we will step by step explain the process of a ladder diagram. The numbers in the black circles indicate the order.

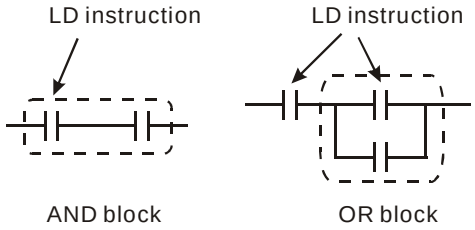


The order of the instructions:

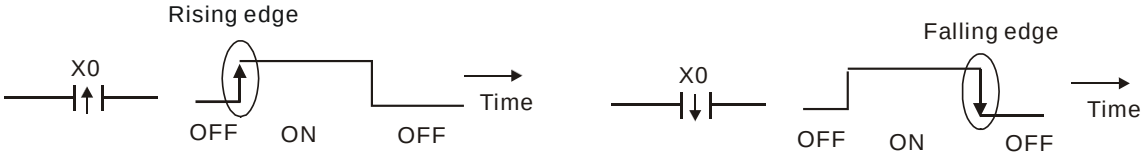
1	LD	X0	
2	OR	M0	
3	AND	X1	
4	LD	X3	
	AND	M1	
	ORB		
5	LD	Y1	
	AND	X4	
6	LD	T0	
	AND	M3	
	ORB		
7	ANB		
8	OUT	Y1	
	TMR	T0	K10

Explanations on the basic structures in the ladder diagram:

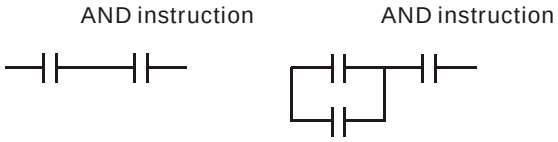
1. LD (LDI) instruction: Given in the start of a block.



The structure of LDP and LDF instructions are the same as that of LD instruction, and the two only differ in their actions. LDP and LDF instructions only act at the rising edge or falling edge when the contact is On, as shown in the figure below.

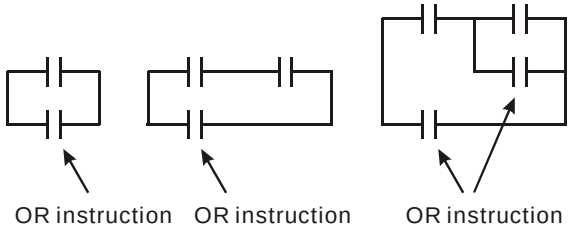


2. AND (ANI) instruction: A single device connects to another single device or a block in series



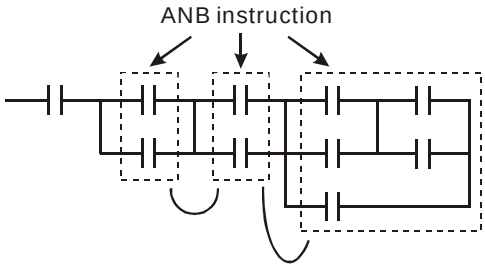
The structure of ANDP and ANDF instructions are the same. ANDP and ANDF instructions only act at the rising edge or falling edge.

3. OR (ORI) instruction: A single device connects to another single device or a block

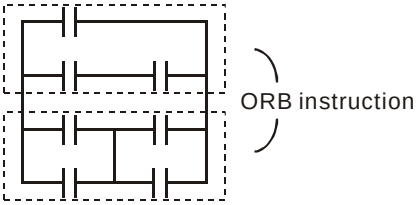


The structure of ORP and ORF instructions are the same. ORP and ORF instructions only act at the rising edge or falling edge.

4. ANB instruction: A block connects to a device or another block in series



5. ORB instruction: A block connects to a device or another block in parallel



If the ANB and ORB operations are with several blocks, the operation should be performed from up to down or left to right, combining into a block or network.

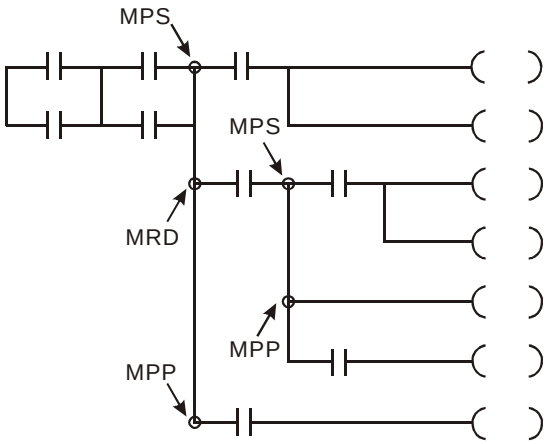
6. MPS, MRD, MPP instructions: Bifurcation point of multiple outputs, for generating many and diverse outputs.

MPS instruction is the start of the bifurcation point. The bifurcation point is the intersection of the horizontal line and vertical line. We will have to determine whether to give a contact memory instruction by the contact status of the same vertical line. Basically, every contact can be given a memory instruction, but considering the convenience of operating the PLC and the limitation on its capacity, some parts in the ladder diagram will be omitted during the conversion. We can determine the type of contact memory instruction by the structure of the ladder diagram. MPS is recognized as "┬" and the instruction can be given continuously for 8 times.

MRD instruction is used for reading the memory of the bifurcation point. Due to that the same vertical line is of the same logic status, in order to continue analyzing other ladder diagrams, we have to read the status of the original contact again. MRD is recognized as "├".

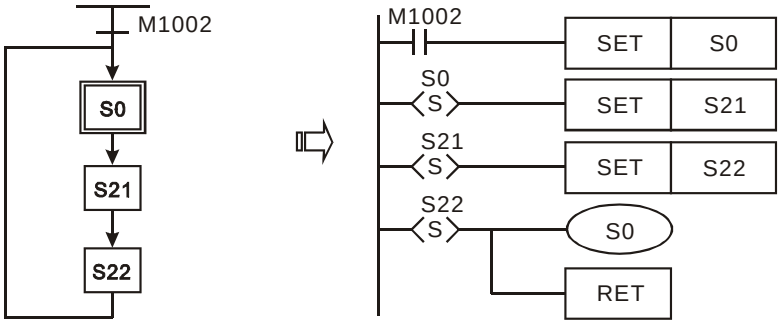
MPP instruction is used for reading the start status of the top bifurcation point and popping it out from the stack. Since MPP is the last item on the vertical line, the vertical line ends at this point.

MPP is recognized as "└". Using the method given above for the analysis cannot be wrong. However, sometimes the compiling program will ignore the same output status, as shown in the figure.



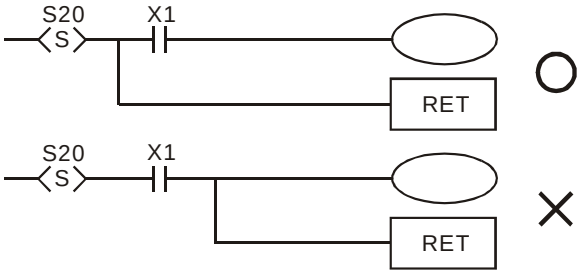
7. STL instruction: Used for designing the syntax of the sequential function chart (SFC).

   STL instruction allows the program designer a clearer and readable picture of the sequence of the program as when they draw a sequence chart. From the figure below, we can see clearly the sequence to be planned. When the step S moves to the next step, the original S will be “Off". Such a sequence can then be converted into a PLC ladder diagram and called “step ladder diagram”.



8. RET instruction: Placed after the completed step ladder diagram.

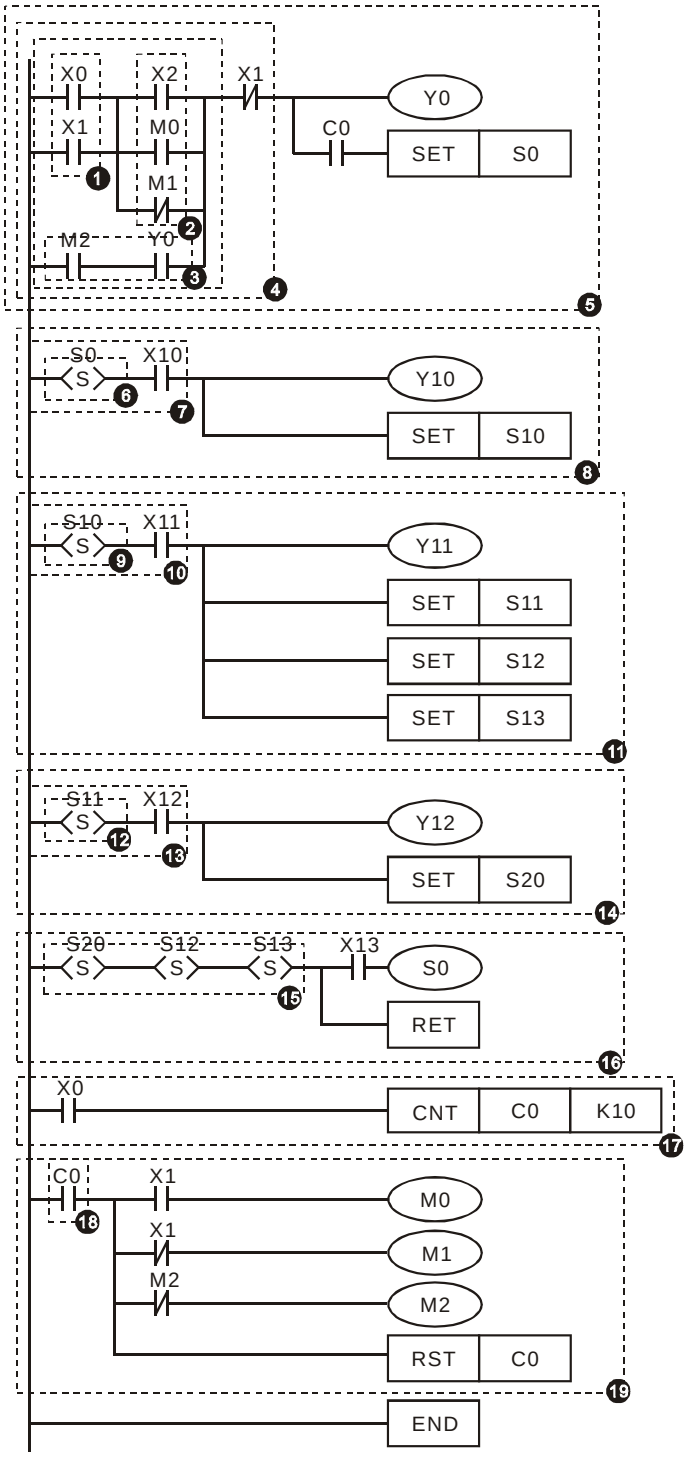
   RET also has be placed after STL instruction. See the example below.



   See step ladder instructions [STL], [RET] in Ch. 4 for the structure of the ladder diagram.

1.5 The Conversion of PLC Command and Each Diagram Structure

Ladder Diagram



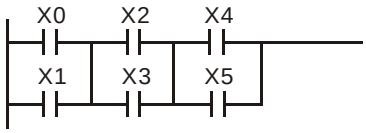

```
LD    X0      ❶ OR block
OR    X1
LD    X2      ❷ OR block
OR    M0
ORI   M1
ANB           Series connection blcok
LD    M2      ❸ AND block
AND   Y0
ORB           Parallel connection block
ANI   X1      ❹ ANI
OUT   Y0      ❺ The output will continue following the status of ❹
AND   C0      Multiple outputs
SET   S0
STL   S0      Step ladder Start ❻
LD    X10     Status S0 and X10 operation ❼ ❽
OUT   Y10     Status working item and step point transfer
SET   S10
STL   S10     ❾ Withdraw S10 status
LD    X11     ❿ Withdraw X11 status ⓫
OUT   Y11
SET   S11     Status working item and step point transfer
SET   S12
SET   S13
STL   S11     ⓬ Withdraw S11 status
LD    X12     ⓭ Withdraw X12 status ⓮
OUT   Y12     Status working item and step point transfer
SET   S20
STL   S20
STL   S12     ⓯ Bifurcation convergence
STL   S13
LD    X13     Status working item and step point transfer
OUT   S0      ⓰ End of step ladder
RET           Return
LD    X0      ⓱
CNT   C0 K10
LD    C0      ⓲ Read C0
MPS
AND   X1      ⓳
OUT   M0
MRD
ANI   X1      Multiple outputs
OUT   M1
MPP
ANI   M2
OUT   M2
RST   C0
END           End of program
```

■ **Fuzzy Syntax**

The correct ladder diagram analysis and combination should be conducted from up to down and left to right. However, without adopting this principle, some instructions can make the same ladder diagram.

Example Program 1

See the ladder diagram below. There are 2 ways to indicate the ladder by instruction programs with the same result.

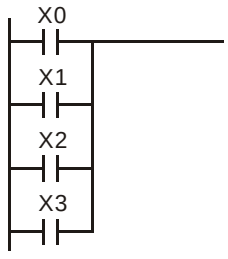


Ideal way		Less ideal way	
LD	X0	LD	X0
OR	X1	OR	X1
LD	X2	LD	X2
OR	X3	OR	X3
ANB		LD	X4
LD	X4	OR	X5
OR	X5	ANB	
ANB		ANB	

The two instruction programs will be converted into the same ladder diagram. The difference between the ideal one and less ideal one is the operation done by the MPU. For the ideal way, the combination is done block by block whereas the less idea way combines all the blocks combine with one another in the last step. Though the length of the program codes of the two ways are equal, the combination done in the last step (by ANB instruction, but ANB cannot be used continuously for more than 8 times) will have to store up the previous calculation results in advance. In our case, there are only two blocks combined and the MPU allows such kind of combination. However, once the number of blocks exceeds the range that the MPU allows, problems will occur. Therefore, the best way is to execute the block combination instruction after a block is made, which will also make the logic sequence planned by the programmer more in order.

Example Program 2

See the ladder diagram below. There are 2 ways to indicate the ladder by instruction programs with the same result.

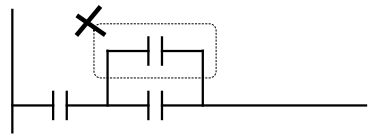
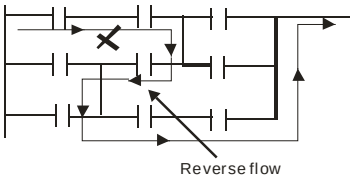
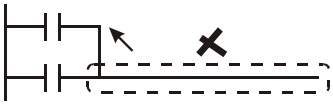
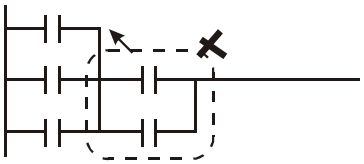
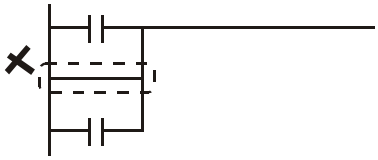
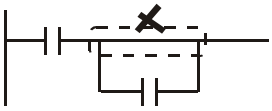
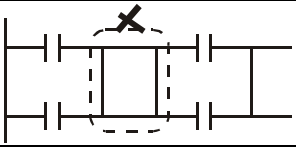
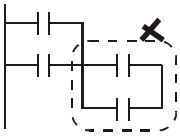
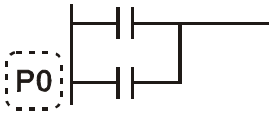
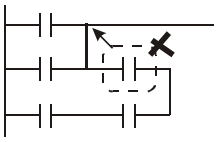


Ideal way		Less ideal way	
LD	X0	LD	X0
OR	X1	LD	X1
OR	X2	LD	X2
OR	X3	LD	X3
		ORB	
		ORB	
		ORB	

In this example, the program codes and the operation memory in the MPU increase in the less ideal way. Therefore, it is better that you edit the program following the defined sequence.

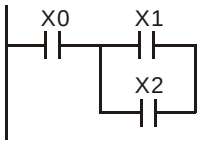
■ **Incorrect Ladder Diagram**

PLC processes the diagram program from up to down and left to right. Though we can use all kinds of ladder symbols to combine into various ladder diagrams, when we draw a ladder diagram, we will have to start the diagram from the left power line and end it at the right power line (In WPLSoft ladder diagram editing area, the right power line is omitted), from left to right horizontally, one row after another from up to down. See bellows for the frequently seen incorrect diagrams:

	OR operation upward is not allowed.
	“Reverse flow” exists in the signal circuit from the beginning of input to output.
	The up-right corner should output first.
	Combining or editing should be done from the up-left to the bottom-right. The dotted-lined area should be moved up.
	Parallel operation with empty device is not allowed.
	Empty device cannot do operations with other devices.
	No device in the middle block.
	Devices and blocks in series should be horizontally aligned.
	Label P0 should be in the first row of a complete network.
	Blocks connected in series should be aligned with the upmost horizontal line.

1.6 Simplified Ladder Diagram

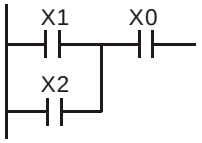
- When a series block is connected to a parallel block in series, place the block in the front to omit ANB instruction.



⇩

Ladder diagram complied into instruction

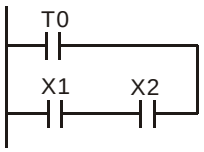
LD	X0
LD	X1
OR	X2
ANB	



Ladder diagram complied into instruction

LD	X1
OR	X2
AND	X0

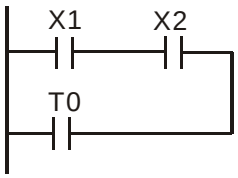
- When a single device is connected to a block in parallel, place the block on top to omit ORB instruction.



⇩

Ladder diagram complied into instruction

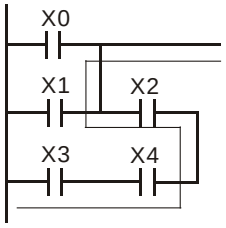
LD	T0
LD	X1
AND	X2
ORB	



Ladder diagram complied into instruction

LD	X1
AND	X2
OR	T0

- In diagram (a), the block on top is shorter than the block in the bottom, we can switch the position of the two blocks to achieve the same logic. Due to that diagram (a) is illegal, there is a “reverse flow” in it.

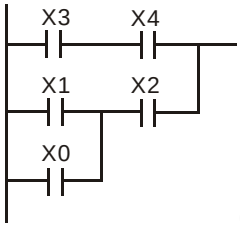


(a)

⇩

Ladder diagram complied into instruction

LD	X0
OR	X1
AND	X2
LD	X3
AND	X4
ORB	

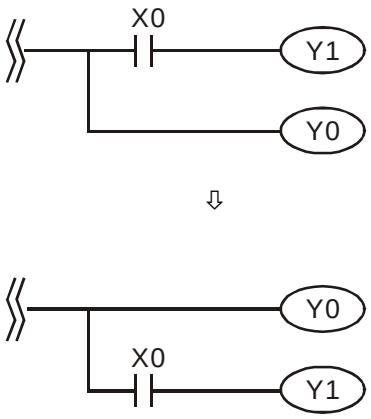


(b)

Ladder diagram complied into instruction

LD	X3
AND	X4
LD	X1
OR	X0
AND	X2
ORB	

- MPS and MPP instruction can be omitted when the multiple outputs in the same horizontal line do not need to operate with other input devices.



Ladder diagram complied into instruction

MPS	
AND	X0
OUT	Y1
MPP	
OUT	Y0

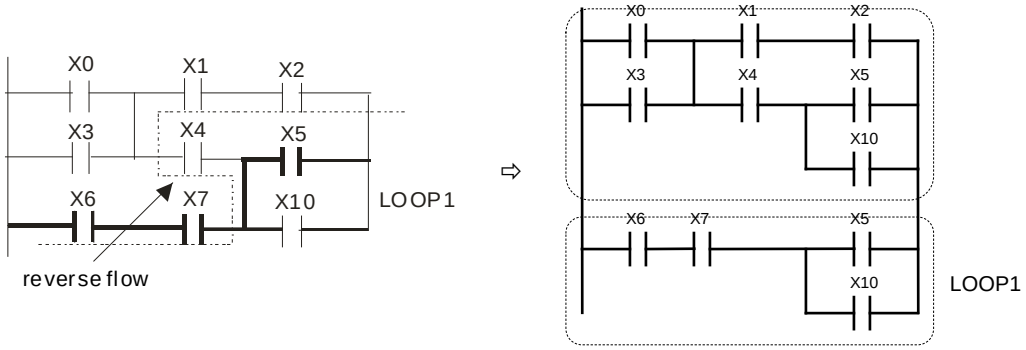
Ladder diagram complied into instruction

OUT	Y0
AND	X0
OUT	Y1

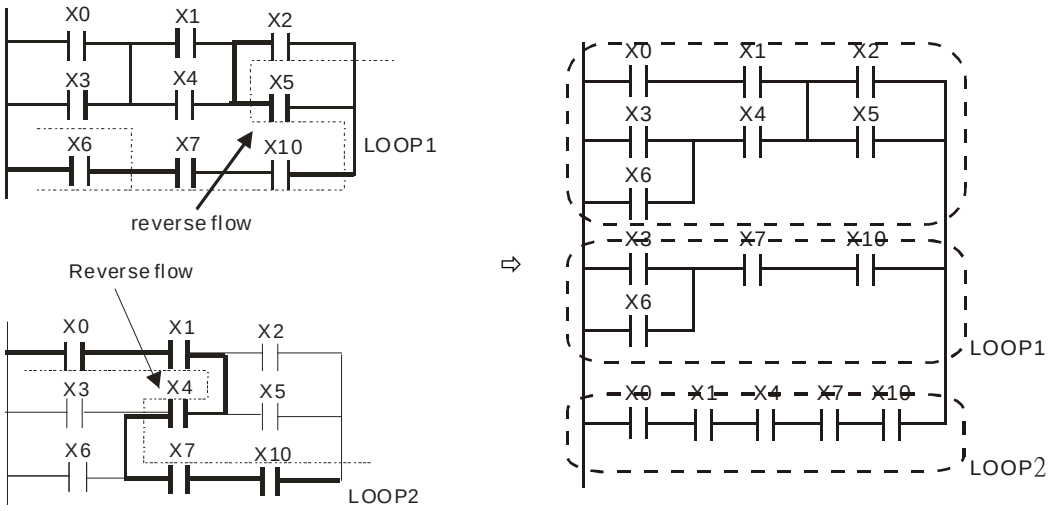
- Correct the circuit of reverse flow

In the following two examples, the diagram in the left hand side is the ladder diagram we desire. However, the illegal "reverse flow" in it is incorrect according to our definition on the ladder diagram. We modify the diagram into the diagram in the right hand side.

Example 1



Example 2



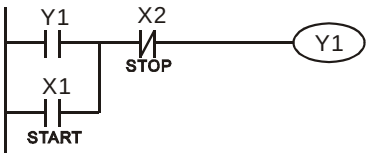
1.7 Basic Program Designing Examples

■ **Start, Stop and Latched**

In some application occasions, we need to use the transient close/open buttons for the start and stop of equipment. To maintain its continuous action, you have to design latched circuits.

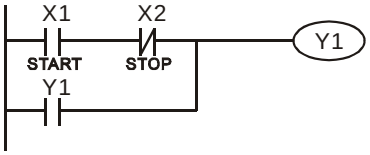
Example 1: Stop first latched circuit

When the normally open contact X1 = On and the normally closed contact X2 = Off, Y1 will be On. If you make X2 = On at this time, Y1 will be Off. It is the reason why this is called “stop first”.



Example 2: Start first latched circuit

When the normally open contact X1 = On and the normally closed contact X2 = Off, Y1 will be On and latched. If you make X2 = On at this time, Y1 will continue to be On because of the latched contact. It is the reason why this is called “start first”.

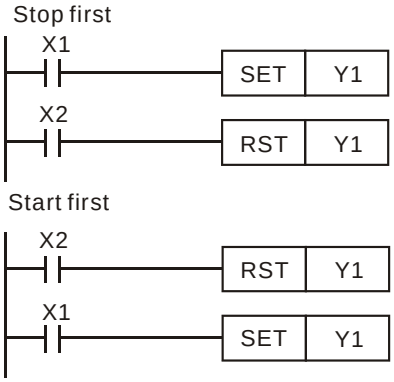


Example 3: Latched circuit for SET and RST instructions

See the diagram in the right hand side for the latched circuit consist of RST and SET instructions.

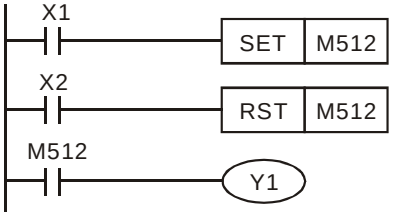
In the stop first diagram, RST is placed after SET. PLC executes the program from up to down, so the On/Off of Y1 will be determined upon its status in the end of the program. Therefore, when X1 and X2 are enabled at the same time, Y1 will be Off. It is the reason why this is called “stop first”.

In the start first diagram, SET is placed after RST. When X1 and X2 are enabled at the same time, Y1 will be On. It is the reason why this is called “start first”.



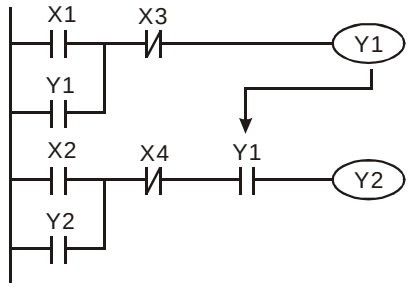
Example 4: Power shutdown latched

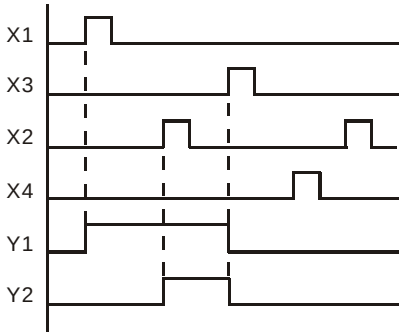
The auxiliary relay M512 is latched (see instruction sheets for DVP series PLC MPU). The circuit can not only be latched when the power is on, but also keep the continuity of the original control when the power is shut down and switched on again.



■ **Frequently Used Control Circuit**

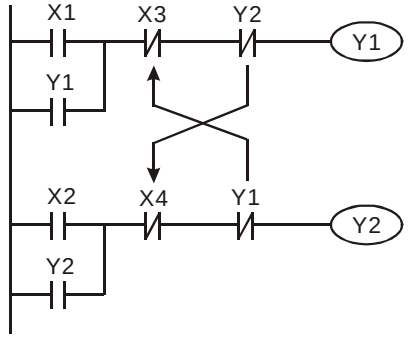
Example 5: Conditional control

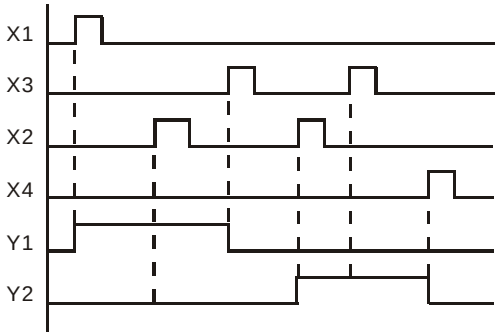




X1 and X3 enables and disables Y1; X2 and X4 enables and disables Y2, and all are latched. Due to that the normally open contact of Y1 is connected to the circuit of Y2 in series, Y1 becomes an AND condition for Y2. Therefore, only when Y1 is enabled can Y2 be enabled.

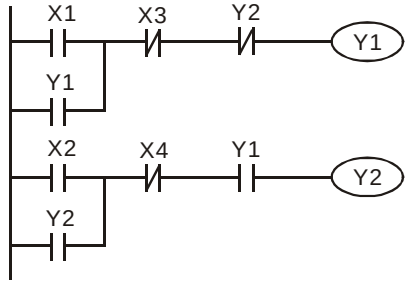
Example 6: Interlock control





Which of the X1 and X2 is first enabled decides either the corresponding output Y1 or Y2 will be enabled first. Either Y1 or Y2 will be enabled at a time, i.e. Y1 and Y2 will not be enabled at the same time (the interlock). Even X1 and X2 are enabled at the same time, Y1 and Y2 will not be enabled at the same time due to that the ladder diagram program is scanned from up to down. In this ladder diagram, Y1 will be enabled first.

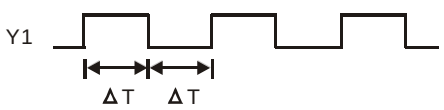
Example 7: Sequential control



If we serially connect the normally closed contact of Y2 in example 5 to the circuit of Y1 as an AND condition for Y1 (as the diagram in the left hand side), the circuit can not only make Y1 as the condition for Y2, but also allow the stop of Y1 after Y2 is enabled. Therefore, we can make Y1 and Y2 execute exactly the sequential control.

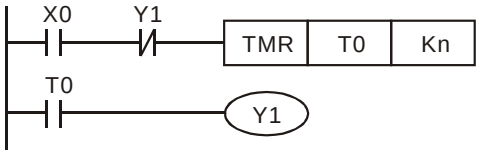
Example 8: Oscillating circuit

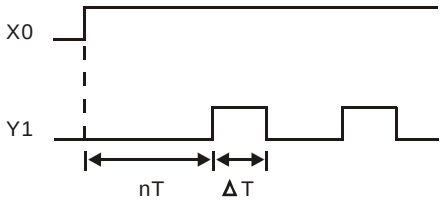
An oscillating circuit with cycle ΔT+ΔT



The ladder diagram above is a very simple one. When the program starts to scan the normally closed contact Y1, Y1 will be closed because coil Y1 is Off. When the program then scan to coil Y1 and make it On, the output will be 1. When the program scans to the normally closed contact Y1 again in the next scan cycle, because coil Y1 is On, Y1 will be open and make coil Y1 Off and output 0. The repeated scans will result in coil Y1 outputs oscillating pulses by the cycle ΔT(On)+ΔT(Off).

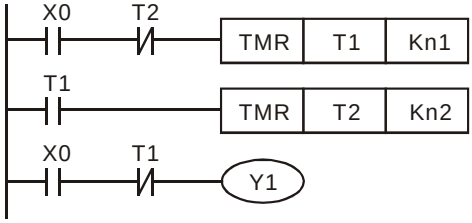
An oscillating circuit with cycle nT+ΔT

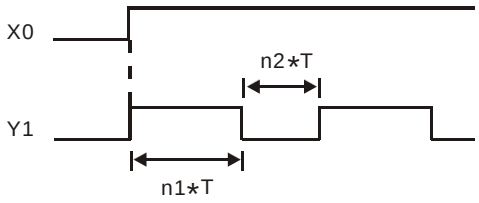




The ladder diagram program controls the On time of coil Y1 by timer T0 and disable timer T0 in the next scan cycle, resulting in the oscillating pulses in the output of Y1. n refers to the decimal set value in the timer and T is the cycle of the clock.

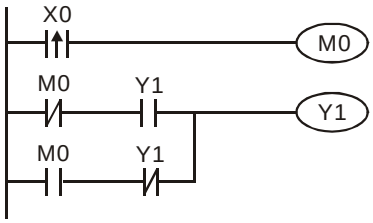
Example 9: Flashing circuit

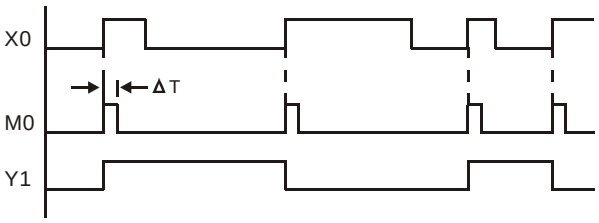




The ladder diagram is an oscillating circuit which makes the indicator flash or enables the buzzer alarms. It uses two timers to control the On/Off time of coil Y1. n1 and n2 refer to the set values in T1 and T2 and T is the cycle of the clock.

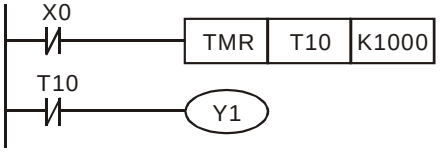
Example 10: Trigger circuit

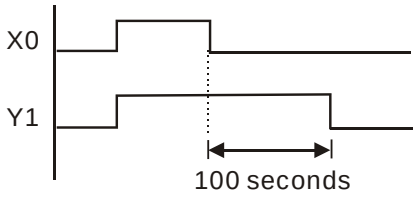




The rising-edge differential instruction of X0 makes coil M0 generate a single pulse of ΔT (one scan cycle). Coil Y1 will be On during this scan period. In the next scan period, coil M0 will be Off and the normally closed contact M0 and Y1 will all be closed, making coil Y1 continue to be On until another rising-edge arrives in input X0, making coil M0 On for another scan period and Y1 Off. Such kind of circuit relies on an input to make two actions execute interchangeably. Also from the timing diagram on the last page, we can see that input X0 are square pulse signals of the cycle T and coil Y1 output are square pulse signals of the cycle 2T.

Example 11: Delay circuit



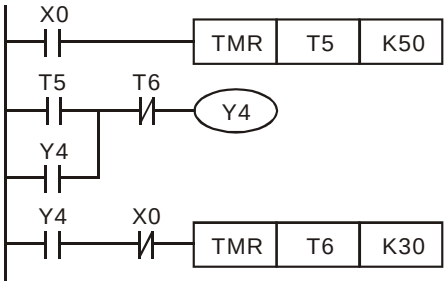


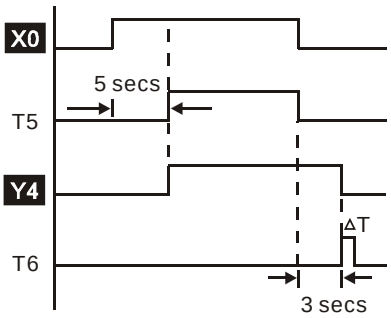
Time base: T = 0.1 sec

When input X0 is On, due to that its corresponding normally closed contact is Off, time T10 will be Off and the output coil Y1 will be On. T10 will be On and start to count until input X0 is Off. Output coil Y1 will be delayed for 100 seconds (K1,000 × 0.1 sec = 100 secs) and be Off. See the timing diagram above.

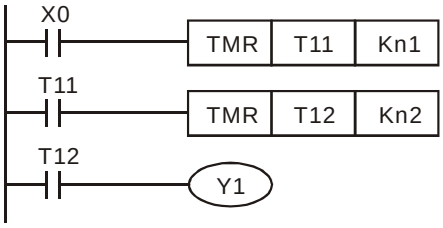
Example 12: Output delay circuit

The output delay circuit is the circuit composed of two timers. When input X0 is On and Off, output Y4 will be delayed.





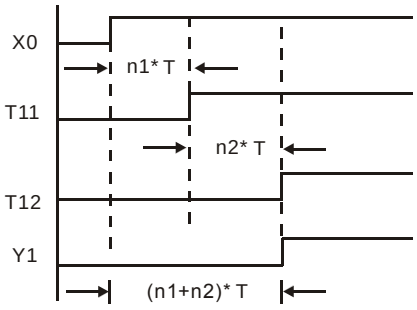
Example13: Timing extension circuit



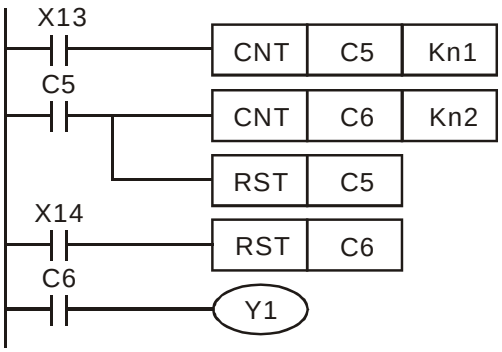
Timer = T11, T12

Clock cycle: T

The total delay time from input X0 is closed to output Y1 is On = (n1+n2)* T. T refers to the clock cycle.

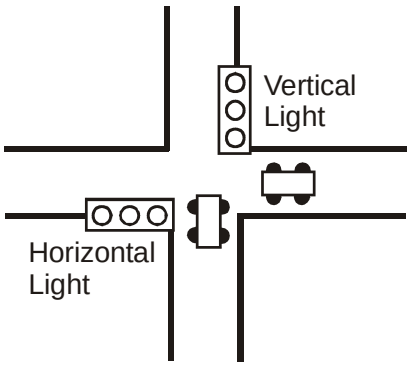


Example 14: How to enlarge the counting range



The counting range of a 16-bit counter is 0 ~ 32,767. As the circuit in the left hand side, using two counters can increase the counting range to n1*n2. When the counting of counter C5 reaches n1, C6 will start to count for one time and reset for counting the pulses from X13. When the counting of counter C6 reaches n2, the pulses from input X13 will be n1*n2.

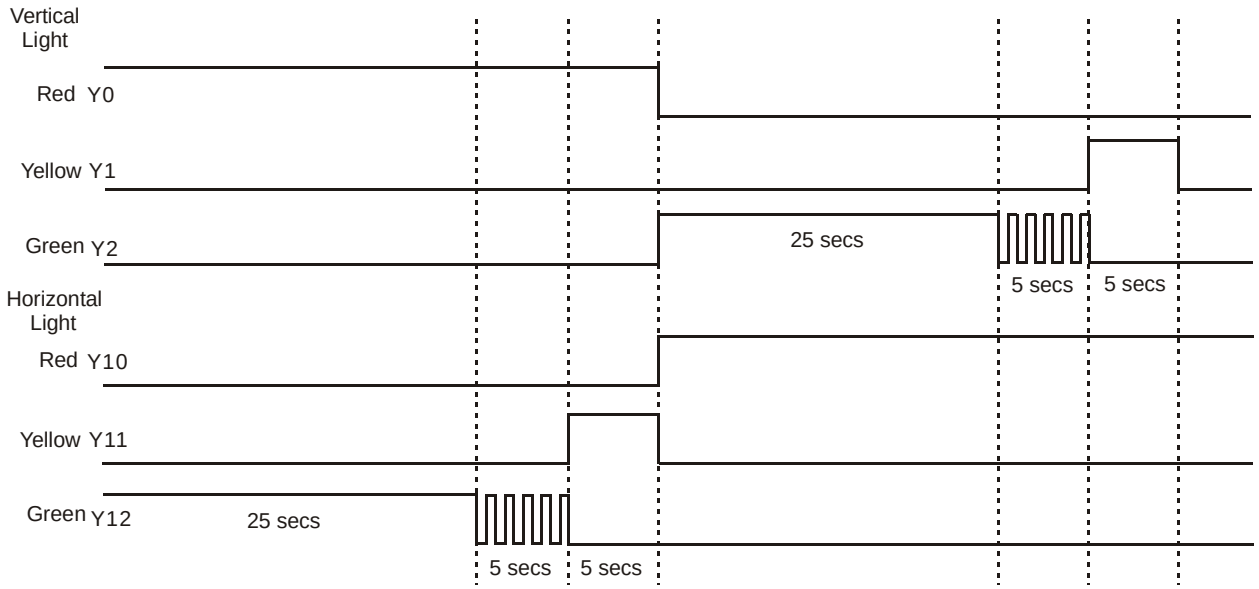
Example 15: Traffic light control (by using step ladder instruction)



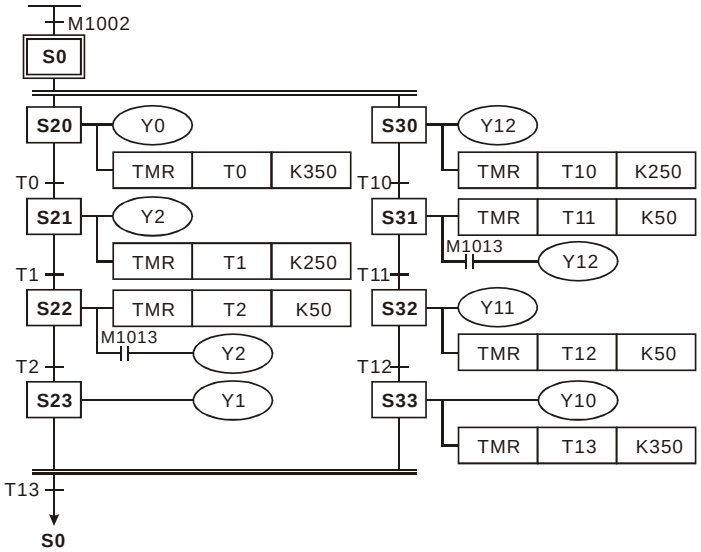
Traffic light control

	Red light	Yellow light	Green light	Green light flashes
Vertical light	Y0	Y1	Y2	Y2
Horizontal light	Y10	Y11	Y12	Y12
On time	35 secs	5 secs	25 secs	5 secs

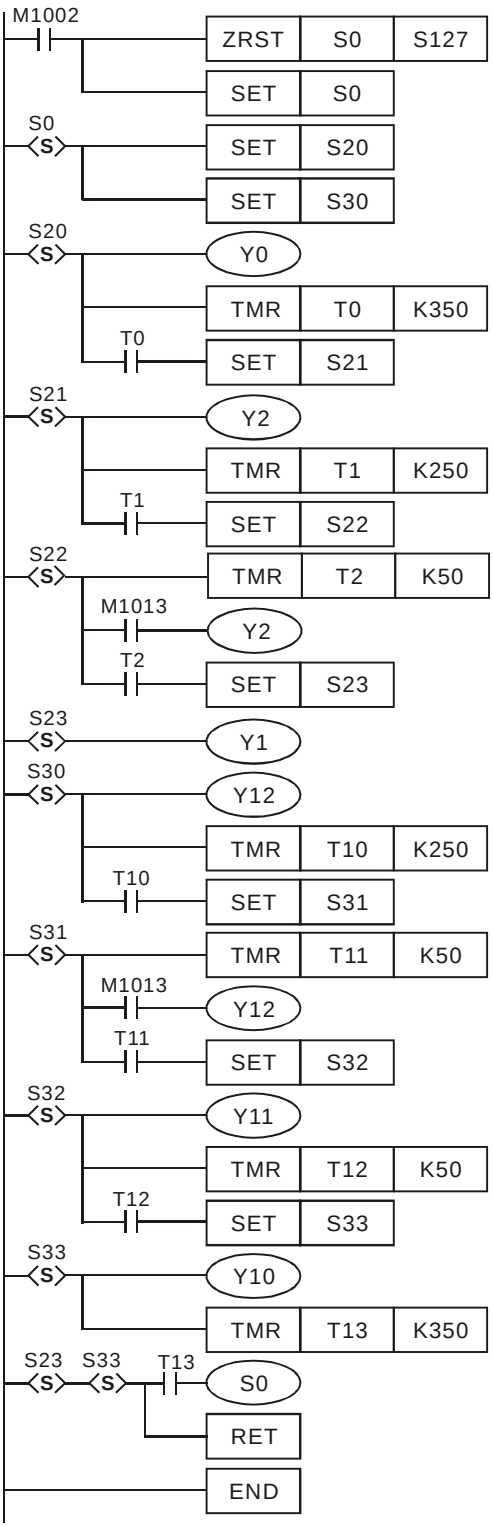
Timing Diagram:



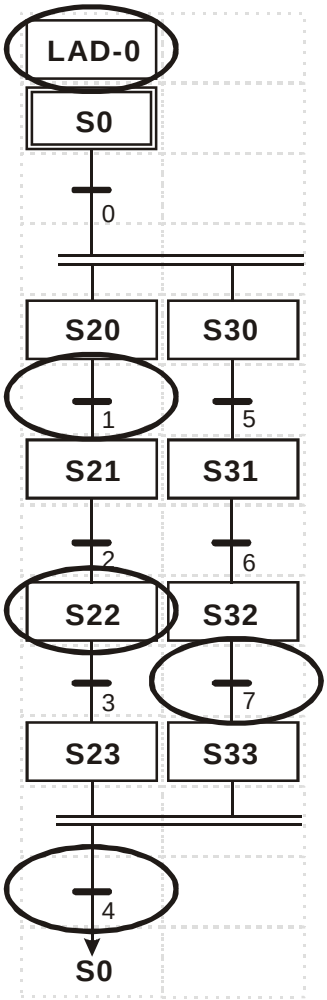
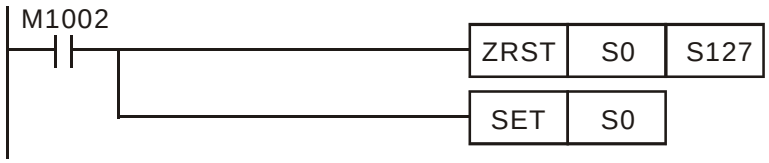
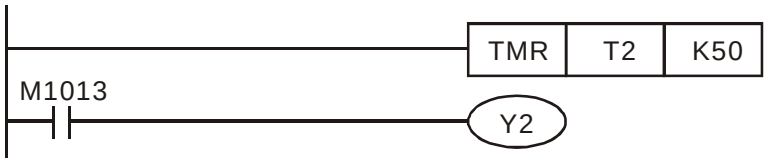
SFC Figure:



Ladder Diagram:


M1002
ZRST S0 S127
SET S0
S0
SET S20
SET S30
S20
Y0
TMR T0 K350
T0
SET S21
S21
Y2
TMR T1 K250
T1
SET S22
S22
TMR T2 K50
M1013
Y2
T2
SET S23
S23
Y1
S30
Y12
TMR T10 K250
T10
SET S31
S31
TMR T11 K50
M1013
Y12
T11
SET S32
S32
Y11
TMR T12 K50
T12
SET S33
S33
Y10
TMR T13 K350
S23 S33 T13
S0
RET
END

■ Drawing by SFC Editor (WPLSoft)

Drawn by SFC	Internal Ladder Diagram
	■ LAD-0  ■ Transferring Condition 1  ■ S22  ■ Transferring Condition 4  ■ Transferring Condition 7 

2.1 All Devices in DVP-PLC

ES/EX/EC Series CPU: (EC3-8K is not included in this table; refer to the next table for devices for EC3-8K.)

Type	Device	Item			Range		Function
Relay (bit)	X	External input relay			X0 ~ X177, 128 points, octal	Total 256 points	Corresponds to external input points
	Y	External output relay			Y0 ~ Y177, 128 points, octal		Corresponds to external output points
	M	Auxiliary relay	General purpose		M0 ~ M511, M768 ~ M999, 744 points	Total 1,280 points	The contact can be On/Off in the program.
			Latched*		M512 ~ M767, 256 points		
			Special purpose		M1000 ~ M1279, 280 points (some are latched)		
	T	Timer	100ms timer		T0 ~ T63, 64 points	Total 128 points	Timer indicated by TMR instruction. If timing reaches its target, the T contact of the same No. will be On.
			10ms timer (M1028 = On)		T64 ~ T126, 63 points (M1028 = Off: 100ms)		
			1ms timer		T127, 1 points		
	C	Counter	16-bit counting up (general purpose)		C0 ~ C111, 112 points	Total 128 points	Counter indicated by CNT (DCNT) instruction. If counting reaches its target, the C contact of the same number will be On.
			16-bit counting up (latched*)		C112 ~ C127, 16 points		
			32-bit counting up/down high-speed counter (latched*)	1-phase 1 input	C235 ~ C238, C241, C242, C244, 7 points	Total 13 points	
				1-phase 2 inputs	C246, C247, C249, 3 points		
				2-phase 2 inputs	C251, C252, C254, 3 points		
	S	Step	Initial step (latched*)		S0 ~ S9, 10 points	Total 128 points	Used for SFC.
			Zero return (latched*)		S10 ~ S19, 10 points (used with IST instruction)		
			Latched*		S20 ~ S127, 108 points		
Register (word data)	T	Present value of timer			T0 ~ T127, 128 points		When the timing reaches the target, the contact of the timer will be On.
	C	Present value of counter			C0 ~ C127, 16-bit counter, 128 points C235 ~ C254, 32-bit counter, 13 points		When the counting reaches the target, the contact of the counter will be On.
	D	Data register	General purpose		D0 ~ D407, 408 points	Total 600 points	Memory area for data storage; E, F can be used for index indication.
			Latched*		D408 ~ D599, 192 points		
			Special purpose		D1000 ~ D1311, 312 points	Total 312 points	
			Index indication		E, F, 2 points		
Pointer	N	For master control nested loop			N0 ~ N7, 8 points		Control point for main control loop
	P	For CJ, CALL instructions			P0 ~ P63, 64 points		Position index for CJ and CALL
	I	Interruption	External interruption		I001, I101, I201, I301, 4 points		Position index for interruption subroutine.
			Timed interruption		I6□□, 1 point (□□=10 ~ 99, time base = 1ms) (for V5.7 and versions above)		
			Communication interruption		I150, 1 point		
Constant	K	Decimal form			K-32,768 ~ K32,767 (16-bit operation) K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)		
	H	Hexadecimal form			H0000 ~ HFFFF (16-bit operation) H00000000 ~ HFFFFFFFF (32-bit operation)		

* The latched area is fixed and cannot be changed.

EC3-8K Series CPU (FW V8.60 or later)

Type	Device	Item		Range		Function
Relay (bit)	X	External input relay		X0 ~ X177, 128 points, octal	Total 256 points	Corresponds to external input points
	Y	External output relay		Y0 ~ Y177, 128 points, octal		Corresponds to external output points
	M	Auxiliary Relay	General purpose	M0 ~ M511, 512 points (*1) M768~M999, 232 points (*1) M2000~M2047, 48 points (*1)	Total 4,096 points	The contact can be On/Off in the program.
			Latched*	M512~M767, 256 points (*2) M2048~M4095, 2,048 points (*2)		
			Special purpose	M1000 ~ M1999, 1,000 points (some are latched)		
	T	Timer	100ms	T0~T63, 64 points T64~T126, 63 points (M1028=Off, 100ms) T128~T183, 56 points (*1) T184~T199, 16 points for subroutine (*1) T250~T255, 6 accumulative points (*1)	Total 256 points	Timer indicated by TMR instruction. If timing reaches its target, the T contact of the same No. will be On.
			10ms	T64~T126, 63 points (M1028=On, 10ms) T200~T239, 40 points(*1) (M1038=Off) T240~T245, 6 accumulative points (*2)		
			1ms	T200~T245, 46 points (*2) (M1038=On) T246~T249, 4 accumulative points (*2) T127, 1 point		
	C	Counter	16-bit counting up	C200~C234, 35 points (*1) C235~C254, 13 points (*2)	Total 248 points	Counter indicated by CNT (DCNT) instruction. If counting reaches its target, the C contact of the same No. will be On.
			32-bit counting up/down	C200 ~ C215, 16 points (*1) C216 ~ C234, 19 points (*3)		
	S	Step point	Initial step	S0 ~ S9, 10 points (*2)	Total 1,024 points	Used for SFC.
			Zero return	S10 ~ S19, 10 points (used with IST instruction) (*2)		
			Latched*	S20 ~ S127, 108 points (*2) S912~1023, 112 points (*2)		
			General purpose	S128 ~ S911, 784 points (*1)		
Register (word data)	T	Present value of timer		T0 ~ T255, 256 points		When the timing reaches the target, the contact of the timer will be On.
	C	Present value of counter		C0 ~ C199, 16-bit counter, 200 points C200 ~ C254, 32-bit counter, 48 points		When the counting reaches the target, the contact of the counter will be On.
	D	Data register	General purpose	D0~D407, 408 points (*1) D600~D999, 400 points (*1) D3920~D4999, 1080 points (*1)	Total 5,000 points	Memory area for data storage; E, F can be used for index indication.
			Latched*	D408~D599, 192 points (*2) D2000~D3919, 1920 points (*2)		
			Special purpose	D1000 ~ D1999, 1,000 points (some are latched)		
			Index indication	E0 ~ E7, F0 ~ F7, 16 points (*1)		

Type	Device	Item		Range	Function
Pointer	N	For Master control loop		N0 ~ N7, 8 points	Control point for main control loop
	P	For CJ, CALL instructions		P0 ~ P255, 256 points	Position index for CJ and CALL
	I	Interruption	External interruption	I001, I101, I201, I301, I401, I501, I601, I701, total 8 points	Position index for interruption subroutine.
			Timed interruption	I6□□, I7□□, 2 points (□□ = 2 ~ 99, time base = 1ms)	
			Interruption inserted	I8□□, 1 points (□□ = 10 ~ 99, time base = 0.1ms)	
			Communication interruption	I150, 1 point	
Constant	K	Decimal form		K-32,768 ~ K32,767 (16-bit operation) K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal form		H0000 ~ HFFFF (16-bit operation) H00000000 ~ HFFFFFFFF (32-bit operation)	

*1. Non-latched area cannot be modified.

*2. The fixed latched area cannot be modified

SX Series CPU

Type	Device	Item		Range		Function
Relay (bit)	X	External input relay		X0 ~ X177, 128 points, octal	Total 256 points	Corresponds to external input points
	Y	External output relay		Y0 ~ Y177, 128 points, octal		Corresponds to external output points
	M	Auxiliary Relay	General purpose	M0 ~ M511, 512 points (*1)	Total 4,096 points	The contact can be On/Off in the program.
			Latched*	M512~M999, 488 points (*3) M2000~M4095, 2,096 points (*3)		
			Special purpose	M1000 ~ M1999, 1,000 points (some are latched)		
	T	Timer	100ms	T0~T199, 200 points (*1) T192~T199, 8 points for subroutine T250~T255, 6 accumulative points (*4)	Total 256 points	Timer indicated by TMR instruction. If timing reaches its target, the T contact of the same No. will be On.
			10ms	T200~T239, 40 points (*1) T240~T245, 6 accumulative points (*4)		
			1ms	T246~T249, 4 accumulative points (*4)		
	C	Counter	16-bit counting up	C0~C95, 96 points (*1) C96~C199, 104 points (*3)	Total 235 points	Counter indicated by CNT (DCNT) instruction. If counting reaches its target, the C contact of the same No. will be On.
			32-bit counting up/down	C200 ~ C215, 16 points (*1) C216 ~ C234, 19 points (*3)		
			32-bit high-speed	C235~C244, 1 phase, 1 input, 9 points (*3) C246~C249, 1 phase, 2 inputs, 3 points (*3) C251~C254, 2 phases, 2 inputs, 4 points (*3)	Total 16 points	
	S	Step point	Initial step	S0 ~ S9, 10 points (*1)	Total 1,024 points	Used for SFC.
			Zero return	S10 ~ S19, 10 points (used with IST instruction) (*1)		
			General purpose	S20~S511, 492 points (*1)		
			Latched*	S512~S895, 384 points (*3)		
			Alarm	S896~S1023, 128 points (*3)		
Register (word data)	T	Present value of timer		T0 ~ T255, 256 points		When the timing reaches the target, the contact of the timer will be On.
	C	Present value of counter		C0 ~ C199, 16-bit counter, 200 points C200 ~ C254, 32-bit counter, 50 points (SC series: 53 points)		When the counting reaches the target, the contact of the counter will be On.
	D	Data register	General purpose	D0~D199, 200 points (*1)	Total 5,000 points (for SX Series FW V3.0 or later: 10,000 points)	Memory area for data storage; E, F can be used for index indication.
			Latched*	D200~D999, 800 points (*3) D2000~D4999, 3,000 points (*3)		
			Special purpose	D1000~D1999, 1,000 points		
			Index indication	D5000~D9999, 5,000 points (*1) (only availbe for SX Series FW V3.0 or later)		
	N/A	File register		K0 ~ K1,599 (1,600 points) (*4)		Expanded register for data storage.

Type	Device	Item		Range	Function
Pointer	N	For Master control loop		N0 ~ N7, 8 points	Control point for main control loop
	P	For CJ, CALL instructions		P0 ~ P255, 256 points	Position index for CJ and CALL
	I	Interruption	External interruption	I001, I101, I201, I301, I401, I501, total 6 points	Position index for interruption subroutine.
			Timed interruption	I6□□, I7□□, 2 points (□□ = 1 ~ 99, time base = 1ms)	
			Interruption inserted when high-speed counter reaches target	I010, I020, I030, I040, I050, I060, total 6 points	
			Communication interruption	I150, 1 point	
Constant	K	Decimal form		K-32,768 ~ K32,767 (16-bit operation) K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal form		H0000 ~ HFFFF (16-bit operation) H00000000 ~ HFFFFFFFF (32-bit operation)	

*1. Non-latched area cannot be modified.

*2. The preset non-latched area can be modified into latched area by setting up parameters.

*3. The preset latched area can be modified into non-latched area by setting up parameters.

*4. The fixed latched area cannot be modified

Latched settings for all devices in SX Series CPU:

M (Auxiliary relay)	General purpose		Latched		Special auxiliary relay		Latched			
	M0 ~ M511		M512 ~ M999		M1000 ~ M1999		M2000 ~ M4095			
	It is fixed to be non-latched		Default: latched Start: D1200 (K512) End: D1201 (K999)		Some are latched and cannot be modified		Default: latched Start: D1202 (K2,000) End: D1203 (K4,095)			
T (Timer)	100 ms		10 ms		10 ms	1 ms	100 ms			
	T0 ~ T199		T200 ~ T239		T240 ~ T245	T246 ~ T249	T250 ~ T255			
	It is fixed to be non-latched		It is fixed to be non-latched		Accumulative type It is fixed to be latched					
C (Counter)	16-bit counting up			32-bit counting up/down			32-bit high-speed counting up/down			
	C0 ~ C95	C96 ~ C199		C200 ~ C215	C216 ~ C234		C235 ~ C255			
	It is fixed to be non-latched	Default: latched Start: D1208 (K96) End: D1209 (K199)		It is fixed to be non-latched	Default: latched Start: D1210 (K216) End: D1211 (K234)		Default: latched Start: D1212 (K235) End: D1213 (K255)			
S (Step relay)	Initial	Zero return	General purpose		Latched		Alarm step			
	S0 ~ S9	S10 ~ S19	S20 ~ S511		S512 ~ S895		S896 ~ S1023			
	It is fixed to be non-latched				Default: latched Start: D1214 (K512) End: D1215 (K895)		It is fixed to be latched			
D (Register)	General purpose		Latched		Special register		Latched			
	D0 ~ D199 D5000~D9999 (Only available for SX Series FW V3.0 and later)		D200 ~ D999		D1000 ~ D1999		D2000 ~ D4999			
	It is fixed to be non-latched		Default: latched Start: D1216 (K200) End: D1217 (K999)		Some are latched and cannot be modified.		Default: latched Start: D1218 (K2,000) End: D1219 (K4,999)			
File Register	K0 ~ K1599									
	It is fixed to be latched.									

EH2/SV Series CPU:

Type	Device	Item		Range		Function
Relay (bit)	X	External input relay		X0 ~ X377, 256 points, octal	Total 512 points	Corresponds to external input points
	Y	External output relay		Y0 ~ Y377, 256 points, octal		Corresponds to external output points
	M	Auxiliary relay	General purpose	M0 ~ M499, 500 points (*2)	Total 4,096 points	The contact can be On/Off in the program.
			Latched	M500 ~ M999, 500 points (*3) M2000 ~ M4095, 2,096 points (*3)		
			Special purpose	M1000 ~ M1999, 1,000 points (some are latched)		
	T	Timer	100ms	T0 ~ T199, 200 points (*2) T192 ~ T199 is for subroutine T250~T255, 6 accumulative points (*4)	Total 256 points	Timer indicated by TMR instruction. If timing reaches its target, the T contact of the same No. will be On.
			10ms	T200 ~ T239, 40 points (*2) T240 ~ T245, 6 accumulative points (*4)		
			1ms	T246 ~ T249, 4 accumulative points (*4)		
	C	Counter	16-bit counting up	C0 ~ C99, 100 points (*2) C100 ~ C199, 100 points (*3)	Total 253 points	Counter indicated by CNT (DCNT) instruction. If counting reaches its target, the C contact of the same No. will be On.
			32-bit counting up/down	C200 ~ C219, 20 points (*2) C220 ~ C234, 15 points (*3)		
			32-bit high-speed counter	C235 ~ C244, 1-phase 1 input, 10 points (*3) C246 ~ C249, 1-phase 2 inputs, 4 points(*3) C251 ~ C254, 2-phases 2 inputs, 4 points (*3)		
	S	Step	Initial step point	S0 ~ S9, 10 points (*2)	Total 1,024 points	Used for SFC.
			Zero return	S10 ~ S19, 10 points (used with IST instruction) (*2)		
			General purpose	S20 ~ S499, 480 points (*2)		
			Latched	S500 ~ S899, 400 points (*3)		
			Alarm	S900 ~ S1023, 124 points (*3)		
Register (word data)	T	Present value of timer		T0 ~ T255, 256 points		When the timing reaches the target, the contact of the timer will be On.
	C	Present value of counter		C0 ~ C199, 16-bit counter, 200 points C200 ~ C254, 32-bit counter, 53 points		When the counting reaches the target, the contact of the counter will be On.
	D	Data register	General purpose	D0 ~ D199, 200 points, (*2)	Total 10,000 points	Memory area for data storage; E, F can be used for index indication.
			Latched	D200 ~ D999, 800 points (*3) D2000 ~ D9999, 8,000 points (*3)		
			Special purpose	D1000 ~ D1999, 1,000 points		
			Index indication	E0 ~ E7, F0 ~ F7, 16 points (*1)		
	N/A	File register		K0 ~ K9,999 (10,000 points) (*4)		Expanded register for data storage.
Pointer	N	For master control loop		N0 ~ N7, 8 points		Control point for main control loop
	P	For CJ, CALL instructions		P0~P255, 256 points		Position index for CJ and CALL
	I	Interruption	External interruption (*5)	I00□(X0), I10□(X1), I20□(X2), I30□(X3), I40□(X4), I50□(X5), 6 points (□ = 1, rising-edge trigger ┘‾, □ = 0, falling-edge trigger ‾└)		Position index for interruption subroutine.
			Timed interruption	I6□□, I7□□, 2 points(□□ = 01~99ms) time base = 1ms I8□□, 1 point (□□ = 05~99, time base = 0.1ms)		
			Interruption inserted when high-speed counter reaches target	I010, I020, I030, I040, I050, I060, 6 points		
			Pulse interruption	I110, I120, I130, I140, 4 points		
			Communication interruption	I150, I160, I170, 3 points		
			Frequency measurement card interruption	I180, 1 point		

Type	Device	Item	Range	Function
Constant	K	Decimal form	K-32,768 ~ K32,767 (16-bit operation) K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal form	H0000 ~ HFFFF (16-bit operation) H00000000 ~ HFFFFFFFF (32-bit operation)	

*1. Non-latched area cannot be modified.

*2. The preset non-latched area can be modified into latched area by setting up parameters.

*3. The preset latched area can be modified into non-latched area by setting up parameters.

*4. The fixed latched area cannot be modified.

*5. The speed at which an external interrupt subroutine is executed depends on the size of the external interrupt subroutine. It is suggested that external interrupt subroutines not be used with high-speed counters.

Latched settings for all devices in EH2/SV Series CPU:

M (Auxiliary relay)	General purpose		Latched		Special auxiliary relay	Latched
	M0 ~ M499		M500 ~ M999		M1000 ~ M1999	M2000 ~ M4095
	Start: D1200 (K500) End: D1201 (K999)				Some are latched and cannot be modified.	Start: D1202 (K2,000) End: D1203 (K4,095)
T (Timer)	100 ms		10 ms		10 ms	1 ms / 100 ms
	T0 ~ T199		T200 ~ T239		T240 ~ T245	T246 ~ T249 / T250 ~ T255
	Default: non-latched		Default: non-latched		Accumulative type It is fixed to be latched.	
	Start: D1204 (K-1)*1 End: D1205 (K-1)*1		Start: D1206 (K-1)*1 End: D1207 (K-1)*1			
C (Counter)	16-bit counting up		32-bit counting up/down		32-bit high-speed counting up/down	
	C0 ~ C99	C100 ~ C199	C200 ~ C219	C220 ~ C234	C235 ~ C245	C246 ~ C255
	Default: non-latched	Default: latched	Default: non-latched	Default: latched	Default: latched	
	Start: D1208 (K100) End: D1209 (K199)		Start: D1210 (K220) End: D1211 (K234)		Start: D1212 (K235) End: D1213 (K255)	
S (Step relay)	Initial	Zero return	General purpose	Latched	Step alarm	
	S0 ~ S9	S10 ~ S19	S20 ~ S499	S500 ~ S899	S900 ~ S1023	
	Non-latched (default)			Latched (default)	It is fixed to be latched.	
	Start: D1214 (K500) End: D1215 (K899)					
D (Register)	General purpose		Latched		Special register	Latched
	D0 ~ D199		D200 ~ D999		D1000 ~ D1999	D2000 ~ D9999
	Default: non-latched		Default: latched		Some is latched and cannot be modified.	Default: latched
	Start: D1216 (K200) End: D1217 (K999)					Start: D1218 (K2,000) End: D1219 (K9,999)
File register	K0 ~ K9,999					
	It is fixed to be latched.					

*1: K-1 refers to the default setting is non-latched.

EH3/SV2 Series CPU:

Type	Device	Item		Range		Function
Relay (bit)	X	External input relay		X0 ~ X377, 256 points, octal	Total 512 points	Corresponds to external input points
	Y	External output relay		Y0 ~ Y377, 256 points, octal		Corresponds to external output points
	M	Auxiliary relay	General purpose	M0 ~ M499, 500 points (*2)	Total 4,096 points	The contact can be On/Off in the program.
			Latched	M500 ~ M999, 500 points (*3) M2000 ~ M4095, 2,096 points (*3)		
			Special purpose	M1000 ~ M1999, 1,000 points (some are latched)		
	T	Timer	100ms	T0 ~ T199, 200 points (*2) T192 ~ T199 is for subroutine T250~T255, 6 accumulative points (*4)	Total 256 points	Timer indicated by TMR instruction. If timing reaches its target, the T contact of the same No. will be On.
			10ms	T200 ~ T239, 40 points (*2) T240 ~ T245, 6 accumulative points (*4)		
			1ms	T246 ~ T249, 4 accumulative points (*4)		
	C	Counter	16-bit counting up	C0 ~ C99, 100 points (*2) C100 ~ C199, 100 points (*3)	Total 253 points	Counter indicated by CNT (DCNT) instruction. If counting reaches its target, the C contact of the same No. will be On.
			32-bit counting up/down	C200 ~ C219, 20 points (*2) C220 ~ C234, 15 points (*3)		
			32-bit high-speed counter	C235 ~ C244, 1-phase 1 input, 10 points (*3) C246 ~ C249, 1-phase 2 inputs, 4 points(*3) C251 ~ C254, 2-phases 2 inputs, 4 points (*3)		
	S	Step	Initial step point	S0 ~ S9, 10 points (*2)	Total 1,024 points	Used for SFC.
			Zero return	S10 ~ S19, 10 points (used with IST instruction) (*2)		
			General purpose	S20 ~ S499, 480 points (*2)		
			Latched	S500 ~ S899, 400 points (*3)		
			Alarm	S900 ~ S1023, 124 points (*3)		
Register (word data)	T	Present value of timer		T0 ~ T255, 256 points		When the timing reaches the target, the contact of the timer will be On.
	C	Present value of counter		C0 ~ C199, 16-bit counter, 200 points C200 ~ C254, 32-bit counter, 53 points		When the counting reaches the target, the contact of the counter will be On.
	D	Data register	General purpose	D0 ~ D199, 200 points, (*2)	Total 12,000 points	Memory area for data storage; E, F can be used for index indication.
			Latched	D200 ~ D999, 800 points (*3) D2000 ~ D9799, 7,800 points (*3) D10000 ~D11999, 2,000 points (*3)		
			Special purpose	D1000 ~ D1999, 1,000 points		
			Right-side special module	D9900~D9999, 100 points (*3) (*6)		
			Left-side special modules	D9800~D9899, 100 points (*3) (*7)		
			Index indication	E0 ~ E7, F0 ~ F7, 16 points (*1)		
	N/A	File register		K0 ~ K9,999 (10,000 points) (*4) (*9)		Expanded register for data storage.

Type	Device	Item		Range	Function
Pointer	N	For master control loop		N0 ~ N7, 8 points	Control point for main control loop
	P	For CJ, CALL instructions		P0~P255, 256 points	Position index for CJ and CALL
	I	Interruption	External interruption (*5)	I00□(X0), I10□(X1), I20□(X2), I30□(X3), I40□(X4), I50□(X5), I60□(X6), I70□(X7), I90□(X10), I91□(X11), I92□(X12), I93□(X13), I94□(X14), I95□(X15), I96□(X16), I97□(X17), 16 points (□=1, rising-edge trigger ┘‾, □=0, falling-edge trigger ‾└)	Position index for interruption subroutine.
			Timed interruption	I6□□, I7□□, 2 points (□□=02~99ms) time base=1ms) I8□□, 1 point (□□=05~99，time base=0.1ms)	
			Interruption inserted when high-speed counter reaches target	I010, I020, I030, I040, I050, I060, 6 points	
			Pulse interruption	I110, I120, I130, I140, 4 points	
			Communication interruption (*8)	I150, I151, I153, I160, I161, I163, I170, 7 points	
Constant	K	Decimal form		K-32,768 ~ K32,767 (16-bit operation) K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Hexadecimal form		H0000 ~ HFFFF (16-bit operation) H00000000 ~ HFFFFFFFF (32-bit operation)	

*1. Non-latched area cannot be modified.

*2. The preset non-latched area can be modified into latched area by setting up parameters.

*3. The preset latched area can be modified into non-latched area by setting up parameters.

*4. The fixed latched area cannot be modified

*5. The speed at which an external interrupt subroutine is executed depends on the size of the external interrupt subroutine. It is suggested that external interrupt subroutines not be used with high-speed counters.

*6. If a PLC is connected to right-side special modules, and M1183 is reset to OFF, the data registers will be available. Every right-side special module connected to a PLC occupies 10 data registers.

*7. If a PLC is connected to left-side special modules, and M1182 is reset to OFF, the data registers will be available. Every left-side special module connected to a PLC occupies 10 data registers.

*8. Please refer to section 2.9 for more information.

*9. The following series EH3 FW V1.40, SV2 FW V1.20 or later support 50000 file registers.

Latched settings for all devices in EH3/SV2 Series CPU:

M (Auxiliary relay)	General purpose	Latched	Special auxiliary relay	Latched		
	M0 ~ M499	M500 ~ M999	M1000 ~ M1999	M2000 ~ M4095		
	Start: D1200 (K500) End: D1201 (K999)		Some are latched and cannot be modified.	Start: D1202 (K2,000) End: D1203 (K4,095)		
T (Timer)	100 ms	10 ms	10 ms	1 ms	100 ms	
	T0 ~ T199	T200 ~ T239	T240 ~ T245	T246 ~ T249	T250 ~ T255	
	Default: non-latched	Default: non-latched	Accumulative type It is fixed to be latched.			
	Start: D1204 (K-1)*1 End: D1205 (K-1)*1	Start: D1206 (K-1)*1 End: D1207 (K-1)*1				
C (Counter)	16-bit counting up		32-bit counting up/down		32-bit high-speed counting up/down	
	C0 ~ C99	C100 ~ C199	C200 ~ C219	C220 ~ C234	C235 ~ C245	C246 ~ C255
	Default: non-latched	Default: latched	Default: non-latched	Default: latched	Default: latched	
	Start: D1208 (K100) End: D1209 (K199)		Start: D1210 (K220) End: D1211 (K234)		Start: D1212 (K235) End: D1213 (K255)	
S (Step relay)	Initial	Zero return	General purpose	Latched	Step alarm	
	S0 ~ S9	S10 ~ S19	S20 ~ S499	S500 ~ S899	S900 ~ S1023	
	Non-latched (default)			Latched (default)	It is fixed to be latched.	
	Start: D1214 (K500) End: D1215 (K899)					
D (Register)	General purpose	Latched	Special register	Latched		
	D0 ~ D199	D200 ~ D999	D1000 ~ D1999	D2000 ~ D12000		
	Default: non-latched	Default: latched	Some is latched and cannot be modified.	Default: latched		
	Start: D1216 (K200) End: D1217 (K999)			Start: D1218 (K2,000) End: D1219 (K9,999)		
File register	K0 ~ K9,999					
	It is fixed to be latched.					

*1: K-1 refers to the default setting is non-latched.

■ **Power On/Off or the CPU switches between RUN/STOP:**

Memory of ES/EX/EC3-8K Series FW V5.5 and later versions

Memory type	Power Off→On	STOP→RUN	RUN→STOP	Clear all non-latched areas (M1031)	Clear all latched areas (M1032)	Default setting
Non-latched	Clear	Clear when M1033 = Off		Clear	Unchanged	0
		Remain unchanged when M1033 = On				
Latched	Unchanged			Unchanged	Clear	Unchanged
Special M, Special D, index register	Initial	Unchanged		Unchanged		Initial setting

Memory of SX/EH3/SV2 Series CPU:

Memory type	Power Off→On	STOP→RUN	RUN→STOP	Clear all non-latched area (M1031)	Clear all latched area (M1032)	Default setting
Non-latched	Clear	Unchanged	Clear when M1033 = Off	Clear	Unchanged	0
			Remain unchanged when M1033 = On			
Latched	Unchanged			Unchanged	Clear	0
Special M, Special D, index register	Initial	Unchanged		Unchanged		Initial setting
File Register	Unchanged					0

2.2 Values, Constants [K] / [H]

Constant	K	Decimal form	K-32,768 ~ K32,767 (16-bit operation) K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)
	H	Hexadecimal form	H0 ~ HFFFF (16-bit operation) H0 ~ HFFFFFFFF (32-bit operation)

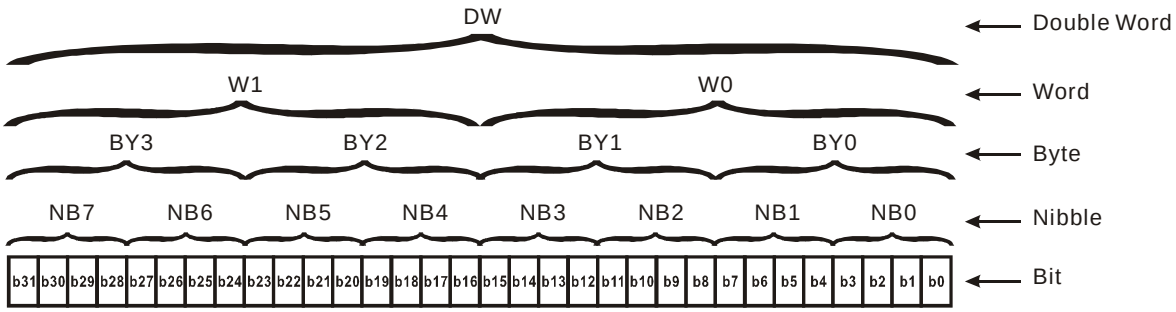
For different control purposes, there are five types of values inside DVP-PLC for executing the operations. See the explanations bellows for the functions and works of every type of value.

1. Binary value (BIN)

   All the operations and storage of values in PLC are conducted in BIN. Belows are the terms for BIN values.

Bit:	The basic unit for a BIN value, either 1 or 0.
Nibble:	Composed of 4 continuous bits (e.g. b3 ~ b0). Presented as the decimal value 0 ~ 9 of a digit or 0 ~ F in hex.
Byte:	Composed of 2 continuous nibble (i.e. 8 bits, b7 ~ b0). Presented as 00 ~ FF in hex.
Word:	Composed of 2 continuous bytes (i.e. 16 bits, b15 ~ b0). Presented as 4-digit 0000 ~ FFFF in hex.
Double word:	Composed of 2 continuous words (i.e. 32 bits, b31 ~ b0). Presented as 8 digit 00000000 ~ FFFFFFFF.

   Bit, nibble, byte, word, and double word in a binary system:



2. Octal value (OCT)

   The No. of external input and output terminals in DVP-PLC is numbered in octal system.

   For example:

   External input: X0 ~ X7, X10 ~ X17…(device No.)

   External output: Y0 ~ Y7, Y10 ~ Y17…(device No.)

3. Decimal value (DEC)

   Occasions of using decimal values in DVP-PLC:

   - Set value in timer T and counter C, e.g. TMR C0 K50 (constant K)
   - No. of device S, M, T, C, D, E, F, P, I, e.g. M10, T30. (device No.)
   - Operands in application instructions, e.g. MOV K123 D0 (constant K)

4. Binary code decimal (BCD)

   A decimal datum is presented by a nibble or 4 bits. Therefore, a continuous 16 bits can be presented as a 4-digit decimal value. BCD is mainly used on reading the input value from the DIP switch or the data output to a 7-section display.

5. Hexadecimal value (HEX)

Occasion of using hexadecimal values:

- Operands in application instructions, e.g. MOV H1A2B D0 (constant H)

Constant K:

"K" is normally placed before a decimal value in the PLC. For example, K100 refers to a decimal value, 100.

Exception: K and bit devices X, Y, M and S can combine into data in bit, byte, word or double word, e.g. K2Y10, K4M100. Here K1 refers to a 4-bit data and K2 ~ K4 refer to 8-bit, 12-bit and 16-bit data.

Constant H:

"H" is normally placed before a hexadecimal value in the PLC. For example, H100 refers to a hexadecimal value, 100.

Reference table:

Binary (BIN)		Octal (OCT)	Decimal (DEC)	Binary Code Decimal (BCD)		Hexadecimal (HEX)
For PLC internal operation		No. of device X, Y	Constant K, No. of device M, S, T, C, D, E, F, P, I	For DIP switch and 7-section display		Constant H
0 0 0 0	0 0 0 0	0	0	0 0 0 0	0 0 0 0	0
0 0 0 0	0 0 0 1	1	1	0 0 0 0	0 0 0 1	1
0 0 0 0	0 0 1 0	2	2	0 0 0 0	0 0 1 0	2
0 0 0 0	0 0 1 1	3	3	0 0 0 0	0 0 1 1	3
0 0 0 0	0 1 0 0	4	4	0 0 0 0	0 1 0 0	4
0 0 0 0	0 1 0 1	5	5	0 0 0 0	0 1 0 1	5
0 0 0 0	0 1 1 0	6	6	0 0 0 0	0 1 1 0	6
0 0 0 0	0 1 1 1	7	7	0 0 0 0	0 1 1 1	7
0 0 0 0	1 0 0 0	10	8	0 0 0 0	1 0 0 0	8
0 0 0 0	1 0 0 1	11	9	0 0 0 0	1 0 0 1	9
0 0 0 0	1 0 1 0	12	10	0 0 0 1	0 0 0 0	A
0 0 0 0	1 0 1 1	13	11	0 0 0 1	0 0 0 1	B
0 0 0 0	1 1 0 0	14	12	0 0 0 1	0 0 1 0	C
0 0 0 0	1 1 0 1	15	13	0 0 0 1	0 0 1 1	D
0 0 0 0	1 1 1 0	16	14	0 0 0 1	0 1 0 0	E
0 0 0 0	1 1 1 1	17	15	0 0 0 1	0 1 0 1	F
0 0 0 1	0 0 0 0	20	16	0 0 0 1	0 1 1 0	10
0 0 0 1	0 0 0 1	21	17	0 0 0 1	0 1 1 1	11
:		:	:	:		:
0 1 1 0	0 0 1 1	143	99	1 0 0 1	1 0 0 1	63

2.3 Numbering and Functions of External Input/Output Contacts [X] / [Y]

No. of input/output contacts (in octal):

The No. of input and output contacts on the PLC CPU starts from X0 and Y0. The range of the No. varies upon the number of points on the CPU. For I/O extension units, the No. of input and output contacts is calculated according to its connection sequence with the CPU.

■ **ES/EX Series CPU:**

DVP Series	14ES	20EX	24ES	32ES	40ES	60ES	I/O Extension Unit
Input X	X0 ~ X7 (8 points)	X0 ~ X7 (8 points)	X0 ~ X17 (16 points)	X0 ~ X17 (16 points)	X0 ~ X27 (24 points)	X0 ~ X43 (36 points)	X20/30/50 ~ X177 (Note)
Output Y	Y0 ~ Y5 (6 points)	Y0 ~ Y5 (6 points)	Y0 ~ Y7 (8 points)	Y0 ~ Y17 (16 points)	Y0 ~ Y17 (16 points)	Y0 ~ Y27 (24 points)	Y20/30 ~ Y177 (Note)

Note: The input points on I/O extension units start from X20 and output points from Y20, except input points on DVP-40ES start from X30 and output from Y20; input points on DVP-60ES start from X50 and output from Y30. The No. of input/output points on the I/O extension units increases by 8's multiple. If the number of points is less than 8, it will be counted as 8.

■ **EC/EC3-8K Series CPU:**

DVP Series	10EC3	14EC3	16EC3	20EC3	24EC3	30EC3	32EC3	40EC3	48EC3	60EC3	I/O Extension Unit
Input X	X0~X5 (6 points)	X0~X7 (8 points)	X0~X7 (8 points)	X0~X13 (12 points)	X0~X13 (12 points)	X0~X21 (18 points)	X0~X17 (16 points)	X0~X27 (24 points)	X0~X33 (28 points)	X0~X43 (36 points)	None
Output Y	Y0~Y3 (4 points)	Y0~Y5 (6 points)	Y0~Y7 (8 points)	Y0~Y7 (8 points)	Y0~Y13 (12 points)	Y0~Y13 (12 points)	Y0~Y17 (16 points)	Y0~Y17 (16 points)	Y0~Y23 (20 points)	Y0~Y27 (24 points)	None

■ **SX Series CPU:**

DVP Series	10SX (Note1)	I/O Extension Unit (Note2)
Input X	X0 ~ X3 (4 points)	X20 ~ X177
Output Y	Y0 ~ Y1 (2 points)	Y20 ~ Y177

Note 1: Besides 4DI and 2DO, SX Series CPU has also 2AI (12-bit) and 2AO (12-bit).

Note 2: SX Series CPU shares the extension units with Silm type Series CPU. The input points on I/O extension units start from X20 and output points start from Y20. The calculation on the number of I/O points is the same as that in Silm type Series CPU.

■ **SV Series CPU:**

DVP Series	28SV2 (Note1)	28SV2 (Note2)	I/O Extension Unit (Note3)
Input X	X0 ~ X17 (16 points)	X0 ~ X11 (10 points)	X20 ~ X377
Output Y	Y0 ~ Y13 (12 points)	Y0 ~ Y13 (12 points)	Y20 ~ Y377

Note 1: The output type is transistor, among which CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4) and CH3 (Y6) are high-speed transistor output (200kHz); others are normal transistor output (10kHz).

Note 2: The output type is transistor, among which CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4) and CH3 (Y6) are high-speed transistor output (200kHz); others are normal transistor output (10kHz). And 2AI (12-bit) are analog input points.

Note 3: The input points on I/O extension units start from X20 and output points start from Y20. The calculation on the No. of I/O points is the same as that in SS series.

■ **EH3 Series CPU:**

DVP Series	16EH3	20EH3	32EH3 (Note3)	40EH3	48EH3	64EH3	80EH3	I/O Extension Unit (Note4)
Input X (Note1)	X0 ~ X7 (8 points)	X0 ~ X13 (12 points)	X0 ~ X17 (16 points)	X0 ~ X27 (24 points)	X0 ~ X27 (24 points)	X0 ~ X37 (32 points)	X0 ~ X47 (40 points)	X※~X377
Output Y (Note2)	Y0 ~ Y7 (8 points)	Y0 ~ Y7 (8 points)	Y0 ~ Y17 (16 points)	Y0 ~ Y17 (16 points)	Y0 ~ Y27 (24 points)	Y0 ~ Y37 (32 points)	Y0 ~ Y47 (40 points)	Y※~Y377

Note 1: High-speed input points (X0~X17) on the CPU: The 200KHz input points on 16EH3 are Ch0(X0/X1) and Ch1(X4/X5); the 200KHz input points on 20EH3 are Ch0(X0/X1) and Ch1(X4/X5); the 20KHz input point on 20EH3 is Ch2(X10/X11); the 200KHz input points on other CPUs are Ch0(X0/X1), Ch1(X4/X5), Ch2(X10/X11) and Ch3(X14/X15). X2~X17 which are not listed above are 10KHz input points.

Note 2: High-speed output points (Y0~Y17) on the CPU: The 200KHz output points on 16EH3 and 20EH3 are Ch0 (Y0) and Ch1 (Y2); the 200KHz output points on 32EH3, 40EH3, 48EH3, 64EH3, and 80EH3 are Ch0 (Y0/Y1), Ch1 (Y2/Y3), Ch2 (Y4), and CH3 (Y6). Other output points which are not listed are 10KHz output points.

Note 2: The high-speed input points Ch0 (X0/X1) and Ch1 (X4/X5) on DVP32EH00M3 are 200KHz differential input points; Ch2 (X10/X11) and Ch3 (X14/X15) are 200KHz open collector input points; other input points are 10KHz open collector input points. The high-speed output points Ch0 (Y0/Y1) and Ch1 (Y2/Y3) are 200KHz differential output points; other output points are 10KHz open collector output points.

Note 4: The I/O points on I/O extension units follow the I/O points on CPUs. The input points on DVP-16EH3 and DVP-20EH3 start from X20 and output points from Y20. The I/O points on I/O extension units are numbered in sequence. The maximal input number is X377, and the maximal output number is Y377.

The placement of input points and output points on the EH DIO module:

When connecting to a less than 32-points CPU, the first input point of the connected I/O module starts from X20 and counting up accordingly for the rest input points. As for the first output point of the connected I/O module, it starts from Y20 and counting up accordingly for the rest output points. When connecting to a more than 32-points CPU, the numbering for the first input point of the connected I/O module starts right after the last input point of the CPU and the same rule applies to the output point numbering.

See the example below for reference. Connected to a more than 32-points CPU, the input/output point number starts right after the last CPU input/output point. The last digit of the input/output ranges from 0 to 7. Thus when the last digit of the CPU input point number ends at 7, the input point number of the connected I/O starts at 0, right after the CPU input point number. If the last digit of the input point number ends at 3, the last digit of the next I/O module input point number starts at 4 as the example of 08HP and 16HP shown below.

MPU EXT1 EXT2 EXT3 EXT4

PLC	Series	Input points	Output points	Input point number	Output point number
CPU	64EH3	32	32	X0~X37	Y0~Y37
EXT1	32HP	16	16	X40~X57	Y40~Y57
EXT2	48HP	24	24	X60~X107	Y60~Y107
EXT3	08HP	4	4	X110~X113	Y110~Y113
EXT4	16HP	8	8	X114~X123-	Y114~Y123

■ **Input relay X0 ~ X377**

The numbering of input relays (or input terminals) is in octal form. EH Series CPU can have up to 256 points and the range is: X0 ~ X7, X10 ~ X17, …, X370 ~ X377.

■ **Output relay Y0 ~ Y377**

The numbering of output relays (or output terminals) is also in octal form. EH2 Series CPU can have up to 256 points and the range is: Y0 ~ Y7, Y10 ~ Y17, …, Y370 ~ Y377.

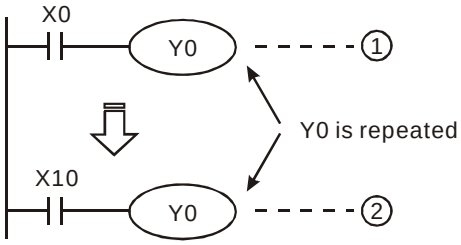
■ **Functions of input contact X**

The input contact X is connected to the input device and reads the input signals into the PLC. There is no limitation on the times of using contact A or B of input contact X in the program. On/Off of the input contact X only changes with On/Off of the input device. You cannot use the peripheral devices (HPP or WPLSoft) to force On/Off of the input contact X.

The special relay M1304 in ES/EC3-8K/EX/SX/EH3/SV2 Series CPU allows WPLSoft to set up On/Off of the CPU input contact X, but the PLC will not be able to receive external input signals at this time.

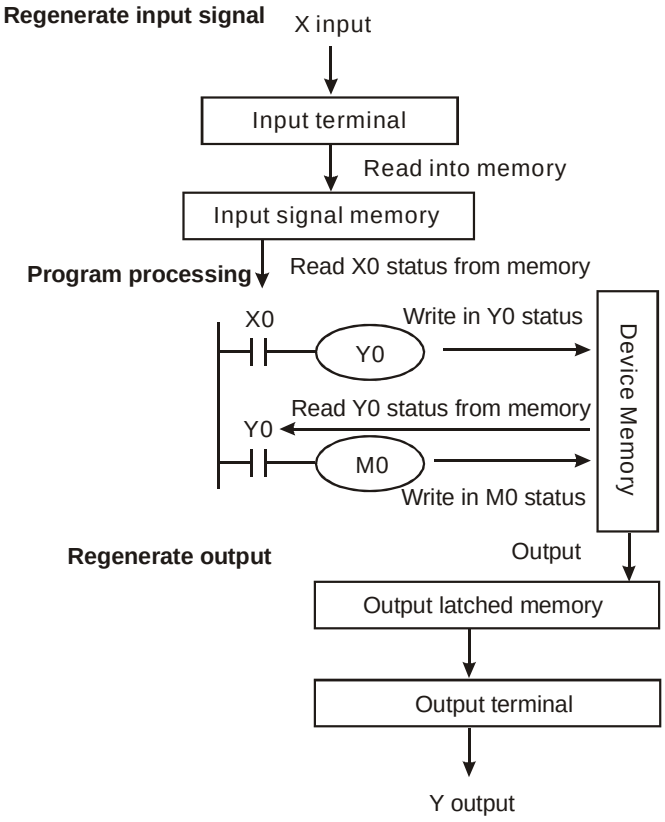
■ **Functions of output contact Y**

Output contact Y sends out On/Off signals to drive the load connected to output contact Y. There are two types of output contacts, relay and transistor. There is no limitation on the times of using contact A or B of output contact Y in the program, but the No. of output coil Y can only be used once in the program; otherwise according to the scan principle of the PLC program, the output status will be determined by the circuit of the last output Y in the program.



The output of Y0 will be determined by circuit ②, i.e. On/Off of X10 will determine the output status of Y0.

The Handling Process of PLC Program (Batch I/O)



■ **Regenerate input signal**

1. Before the execution of the program, PLC reads the On/Off status of the external input signals into the input signal memory at a time.
2. The On/Off status of the input signal during the execution of the program will not change the signal status in the input signal memory. The new On/Off status will be read in the next scan.
3. There will be approximately a 10ms delay from the On→Off or Off→On changes to the status being recognized by the contact in the program. The delay time may be affected by the scan time in the program.

■ **Program processing**

After the PLC reads the On/Off status of every input signal in the input signal memory, it will start to execute every instruction in the program in order starting from address 0. The execution result (On/Off of every output coil) will be stored in order into the device memory.

■ **Regenerate output**

1. When the program executes to END instruction, it will send the On/Off status of Y in the device memory to the output latched memory. The output latched memory is the coil of the output relay.
2. There will be a 10ms delay from On→Off or Off→On of the relay coil to the On/Off status of the contact.
3. There will be a 10 ~ 20us delay from On→Off or Off→On of the transistor module to the On/Off status of the contact.

2.4 Numbering and Functions of Auxiliary Relays [M]

No. of auxiliary relays (in decimal)

■ **ES/EX/EC Series CPU:**

Auxiliary relay M	General purpose	M0 ~ M511, M768 ~ M999, 744 points. Fixed to be non-latched.	Total 1,280 points
	Latched	M512 ~ M767, 256 points. Fixed to be latched.	
	Special purpose	M1000 ~ M1279, 280 points. Some are latched.	

■ **EC3-8K Series CPU (FM V8.60 or later):**

Auxiliary relay M	General purpose	M0 ~ M511, M768 ~ M999, 744 points. Fixed to be non-latched. M2000 ~ M2047, 48 points. Fixed to be non-latched.	Total 4,096 points
	Latched	M512 ~ M767, 256 points. Fixed to be latched. M2048 ~ M4095, 2048 points. Fixed to be latched.	
	Special purpose	M1000 ~ M1279, 280 points. Some are latched.	

■ **SX Series CPU:**

Auxiliary relay M	General purpose	M0 ~ M511, 512 points. Fixed to be non-latched.	Total 4,096 points
	Latched	M512 ~ M999, M2000 ~ M4095, 2,584 points. Can be modified to be non-latched by setting up parameters.	
	Special purpose	M1000 ~ M1999, 1000 points. Some are latched.	

■ **EH3/SV2 Series CPU:**

Auxiliary relay M	General purpose	M0 ~ M499, 500 points. Can be modified to be latched by setting up parameters.	Total 4,096 points
	Latched	M500 ~ M999, M2000 ~ M4095, 2,596 points. Can be modified to be non-latched by setting up parameters.	
	Special purpose	M1000 ~ M1999, 1,000 points. Some are latched.	

Functions of auxiliary relays:

Both auxiliary relay M and output relay Y have output coils and contact A, B, and there is no limitation on the times of using the contact. You can use auxiliary relay M to assemble a control loop, but it cannot directly drive the external load. There are three types of auxiliary relays:

1. **General purpose auxiliary relay:** If the relay encounters power cut during the operation of the PLC, its status will be reset to Off and stay Off when the power is on again.
2. **Latched auxiliary relay:** If the relay encounters power cut during the operation of the PLC, its status will be retained and stay at the status before the power cut when the power is on again.
3. **Special purpose auxiliary relay:** Every relay of this kind has its specific function. Do not use undefined special purpose auxiliary relay. See 2.10 for special purpose auxiliary relay of all Series CPU and 2.11 for its functions.

2.5 Numbering and Functions of Step Relays [S]

No. of step relays (in decimal)

■ **ES/EX/EC Series CPU:**

Step relay S	Initial latched	S0 ~ S9, 10 points. Fixed to be latched.	Total 128 points
	Zero return latched	S10 ~ S19, 10 points, used with IST instruction. Fixed to be latched.	
	Latched	S20 ~ S127, 108 points. Fixed to be latched.	

■ **EC3-8K Series CPU:**

Step relay S	Initial	S0 ~ S9, 10 points. Fixed to be non-latched.	Total 1,024 points
	Zero return	S10 ~ S19, 10 points, used with IST instruction. Fixed to be non-latched.	
Step relay S	General purpose	S20 ~ S127, 108 points. Fixed to be non-latched.	
	Latched	S912 ~ S1023, 112 points. Can be modified to be non-latched by setting up parameters.	
	Alarm	S128 ~ S911, 784 points. Fixed to be latched.	

■ **SX Series CPU:**

Step relay S	Initial	S0 ~ S9, 10 points. Fixed to be non-latched.	Total 1,024 points
	Zero return	S10 ~ S19, 10 points, used with IST instruction. Fixed to be non-latched.	
Step relay S	General purpose	S20 ~ S511, 492 points. Fixed to be non-latched.	Total 1,024 points
	Latched	S512 ~ S895, 384 points. Can be modified to be non-latched by setting up parameters.	
	Alarm	S896 ~ S1023, 128 points. Fixed to be latched.	

■ **EH3/SV2 Series CPU:**

Step relay S	Initial	S0 ~ S9, 10 points. Can be modified to be latched by setting up parameters.	Total 1,024 points
	Zero return	S10 ~ S19, 10 points, used with IST instruction. Can be modified to be latched by setting up parameters.	
	General purpose	S20 ~ S499, 480 points. Can be modified to be latched by setting up parameters.	
	Latched	S500 ~ S899, 400 points. Can be modified to be non-latched by setting up parameters.	
	Alarm	S900 ~ S1023, 124 points. Can be modified to be latched by setting up parameters.	

Functions of step relays:

The step relay S can easily set up the procedure in the industrial automation, which is the most basic device in the sequential function chart (SFC) and has to be used with STL, RET instructions.

The device No. of S is S0 ~ S1023 (total 1,024 points) and both step relay S and output relay Y have output coils and contact A, B, and there is no limitation on the times of using the contact. S cannot directly drive the external load. When the step relay is not used in SFC, it can be used as a normal auxiliary relay. There are four types of step relays:

1. **Initial step relay:** S0 ~ S9, total 10 points, used for initial steps.
2. **Zero return step relay:** S10 ~ S19, total 10 points. S10 ~ S19 are planned for zero return when used with API 60 IST instruction in the program. If they are not used with IST, they will become normal step relays.
3. **General purpose step relay:** S128 ~ S911, total 784 points (for EC3-8K Series CPU); S20 ~ S511, total 492 points (for SX Series CPU); S20 ~ S499, total 480 points (for EH3/SV2 Series CPU). Used for general purposes in SFC and their status will all be cleared when the operation of the PLC encounters power cut.
4. **Latched step relay:** S20 ~ S127, total 108 points (for ES/EX/EC Series CPU); S912 ~ S1023, total 112 points (for EC3-8K Series CPU); S512 ~ S895, total 384 points (for SX Series CPU); S500 ~ S899, total 400 points (for EH3/SV2 Series CPU). Used for latched function in SFC and their status will all be retained when the operation of the PLC encounters power cut. They will remain at the status before the power cut when the PLC is powered again.
5. **Alarm step relay:** S896 ~ S1023, total 128 points (for SX Series CPU); S900 ~ S1023, total 124 points (for EH3/SV2 Series CPU). Used with alarm driving instruction API 46 ANS as an alarm contact for recording the alarm messages or eliminating external malfunctions.

2.6 Numbering and Functions of Timers [T]

No. of timers (in decimal)

■ **ES/EX/SS Series CPU:**

Timer T	100ms general purpose	T0 ~ T63, 64 points	Total 128 points
	10ms general purpose	T64 ~ T126, 63 points (M1028 = On:10ms; M1028 = Off:100ms)	
	1ms general purpose	T127, 1 point	

■ **EC3-8K Series CPU:**

Timer T	100ms general purpose	T0 ~ T63, 64 points T64 ~ T126, 63 points (M1028 = Off:100ms) T128~T183, 56 points T184~T199, 16 points for subroutine T250~T255, 6 accumulative points	Total 256 points
	10ms general purpose	T64 ~ T126, 63 points (M1028 = On:10ms) T200~T239, 40 points (M1038=Off: 10ms) T240~T245, 6 accumulative points	
	1ms general purpose	T127, 1 point T246~T249, 4 accumulative points	

■ **SX Series CPU:**

Time T	100ms general purpose	T0 ~ T199, 200 points. T192 ~ T199 are the timers for subroutine. Fixed to be non-latched	Total 256 points
	100ms accumulative	T250 ~ T255, 6 points. Fixed to be latched.	
	10ms general purpose	T200 ~ T239, 40 points. Fixed to be non-latched	
	10ms accumulative	T240 ~ T245, 6 points. Fixed to be latched.	
	1ms accumulative	T246 ~ T249, 4 points. Fixed to be latched.	

■ **EH3/SV2 Series CPU:**

Timer T	100ms general purpose	T0 ~ T199, 200 points. Can be latched by setting up parameters. T192 ~ T199 are the timers for subroutine.	Total 256 points
	100ms accumulative	T250 ~ T255, 6 points. Fixed to be latched.	
	10ms general purpose	T200 ~ T239, 40 points. Can be latched by setting up parameters.	
	10ms accumulative	T240 ~ T245, 6 points. Fixed to be latched.	
	1ms accumulative	T246 ~ T249, 4 points. Fixed to be latched.	

Functions of timers:

The units of the timer are 1ms, 10ms and 100ms and the counting method is counting up. When the present value in the timer equals the set value, the output coil will be On. The set value should be a K value in decimal and the data register D can also be a set value.

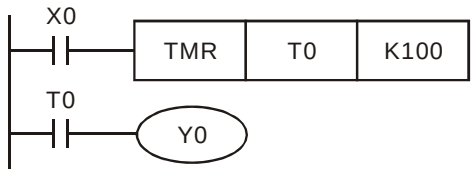
The actual set time in the timer = timing unit × set value

There are three types of timers:

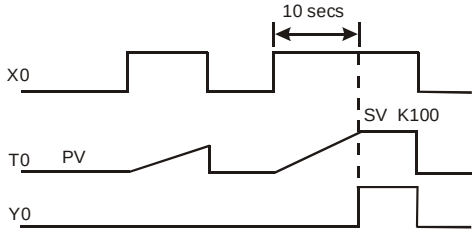
1. **General purpose timer:**

   For ES/EC3-8K/SX Series CPU: The timer executes once when the program reaches END instruction. When TMR instruction is executed, the output coil will be On when the timing reaches its target.

   For EH3/SV2 Series CPU: The timer executes once when the program reaches TMR instruction. When TMR instruction is executed, the output coil will be On when the timing reaches its target.



- When X0 = On, The PV in timer T0 will count up by 100ms. When the PV = SV K100, the output coil T0 will be On.

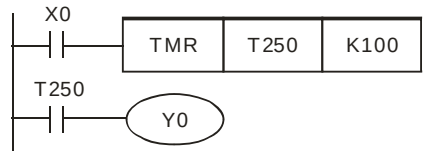


- When X0 = Off or the power is off, the PV in timer T0 will be cleared as 0, and the output coil T0 will be Off.

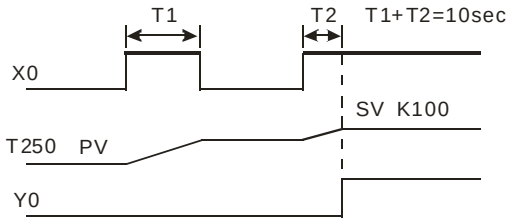
2. **Accumulative type timer:**

   For ES/EC3-8K/SX Series CPU: The timer executes once when the program reaches END instruction. When TMR instruction is executed, the output coil will be On when the timing reaches its target.

   For EH3/SV2 Series CPU: The timer executes once when the program reaches TMR instruction. When TMR instruction is executed, the output coil will be On when the timing reaches its target.



- When X0 = On, The PV in timer T250 will count up by 100ms. When the PV = SV K100, the output coil T0 will be On.
- When X0 = Off or the power is off, timer T250 will temporarily stop the timing and the PV remain unchanged. When X0 is On again, the timing will resume and the PV will count up and when the PV = SV K100, the output coil T0 will be On.



3. **Subroutine timer:**

   Timer T192 ~ T199 are used in subroutines or interruption subroutines for EH3/SV2 Series; T184 ~T199 are for EC3-8K..

   For EC3-8K Series CPU: The timer executes once when the program reaches END instruction. When END instruction is executed, the output coil will be On when the timing reaches its target.

   For EH3/SV2 Series CPU: The timer executes once when the program reaches TMR or END instruction. When TMR or END instruction is executed, the output coil will be On when the PV equals SV.

   If the general purpose timer is used in a subroutine or interruption subroutine but the subroutine is not being executed, the timer will not be able to time correctly.

   How to designate SV: The actual set time in the timer = timing unit × set value

   a) Designating constant K: SV is a constant K

   b) Indirectly designating D: SV is data register D

2.7 Numbering and Functions of Counters [C]

No. of counters (in decimal)

■ **ES/EX/EC Series CPU:**

Counter C	16-bit counting up, for general purpose	C0 ~ C111, 112 points. Fixed to be non-latched.	Total 141 points
	16-bit counting up, for latched	C112 ~ C127, 16 points. Fixed to be latched.	
32-bit counting up/down high-speed counter C	1-phase 1 input	C235 ~ C238, C241, C242, C244, 7 points. Fixed to be latched.	
	1-phase 2 inputs	C246, C247, C249, 3 points. Fixed to be latched.	
	2-phase 2 inputs	C251, C252, C254, 3 points. Fixed to be latched.	

■ **EC3-8K Series CPU:**

Counter C	16-bit counting up, for general purpose	C0 ~ C111, 112 points. Fixed to be non-latched. C128~C199, 72 points. Fixed to be non-latched.	Total 248 points
	16-bit counting up, for latched	C112~C127, 16 points. Fixed to be latched.	
32-bit counting up/down high-speed counter C	32-bit counting up/down, for general purpose	C200~C234, 35 points. Fixed to be non-latched.	
	1-phase 1 input	C235~C238, C241, C242, C244, 7 points. Fixed to be latched.	
	1-phase 2 inputs	C246, C247, C249, 3 points. Fixed to be latched.	
	2-phase 2 inputs	C251, C252, C254, 3 points. Fixed to be latched.	

■ **SX Series CPU:**

Counter C	16-bit counting up, for general purpose	C0 ~ C95, 96 points. Fixed to be non-latched.		Total 235 points
	16-bit counting up, for latched	C96 ~ C199, 104 points. Can be modified to be non-latched by setting up parameters.		
	32-bit counting up/down, for general purpose	C200 ~ C215, 16 points. Fixed to be non-latched.		
	32-bit counting up/down, for latched	C216 ~ C234, 19 points. Can be modified to be non-latched by setting up parameters.		
(SA/SX) 32-bit counting up/down high-speed counter C	1-phase 1 input, for latched	C235 ~ C242, C244, 9 points	Can be modified to be non-latched by setting up parameters.	Total 16 points
	1-phase 2 inputs, for latched	C246, C247, C249, 3 points		
	2-phase 2 inputs, for latched	C251 ~ C254, 4 points		
(SC) 32-bit counting up/down high-speed counter C	1-phase 1 input, for latched	C235 ~ C245, 11 points		Total 19 points
	1-phase 2 inputs, for latched	C246 ~ C250, 4 points		
	2-phase 2 inputs, for latched	C251 ~ C255, 4 points		

■ **EH3/SV2 Series CPU:**

Counter C	16-bit counting up, for general purpose	C0 ~ C99, 100 points. Can be modified to be latched by setting up parameters.		Total 253 points
	16-bit counting up, for latched	C100 ~ C199, 100 points. Can be modified to be non-latched area by setting up parameters.		
	32-bit counting up/down, for general purpose	C200 ~ C219, 20 points. Can be modified to be latched by setting up parameters.		
	32-bit counting up/down, for latched	C220 ~ C234, 15 points. Can be modified to be non-latched by setting up parameters.		
32-bit counting up/down high-speed counter C	Software 1-phase 1 input	C235 ~ C240, 6 points	Can be modified to be non-latched by setting up parameters.	
	Hardware 1-phase 1 input	C241 ~ C244, 4 points		
	Hardware 1-phase 2 inputs	C246 ~ C249, 4 points		
	Hardware 2-phase 2 inputs	C251 ~ C254, 4 points		

■ **Features of counter:**

	16 bits counters	32 bits counters	
Type	General purpose	General purpose	High speed
Counting direction	Counting up	Counting up, counting down	
Set value	0 ~ 32,767	-2,147,483,648 ~ +2,147,483,647	
SV designation	Constant K or data register D	Constant K or data register D (designating 2 values)	
Present value	Counting will stop when the SV is reached.	Counter will continue when the SV is reached.	
Output contact	On and being retained when the counting reaches SV.	On and keeps being On when counting up reaches SV. Reset to Off when counting down reaches SV.	
Reset	PV will be return to 0 when RST instruction is executed and the contact will be reset to Off.		
Contact action	Acts when the scanning is completed.	Acts when the scanning is completed.	Acts immediately when the counting reaches its target, has nothing to do with the scan period.

Functions of counters:

When the pulse input signals of the counter go from Off to On and the present value in the counter equals the set value, the output coil will be On. The set value should be a K value in decimal and the data register D can also be a set value.

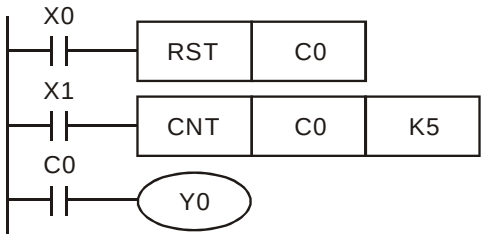
16-bit counters C0 ~ C199:

1. The setup range of 16-bit counter: K0 ~ K32,767. K0 is the same as K1. The output contact will be On immediately when the first counting starts.
2. PV in the general purpose counter will be cleared when the power of the PLC is switched off. If the counter is a latched type, the counter will retain the PV and contact status before the power is off and resume the counting after the power is on again.
3. If you use MOV instruction, WPLSoft or HPP to send a value bigger than the SV to the present value register of C0, next time when X1 goes from Off to On, the contact of counter C0 will be On and its PV will equal SV.
4. The SV in the counter can be constant K (set up directly) or the values in register D (set up indirectly, excluding special data registers D1000~ D1999).

5. If you set up a constant K as the SV, it should be a positive value. Data register D as SV can be positive or negative. When the PV reaches up to 32,767, the next PV will turn to -32,768.

Example:

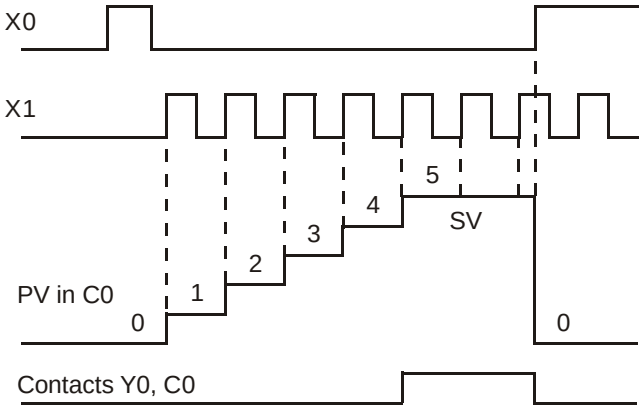
```
LD      X0
RST     C0
LD      X1
CNT     C0   K5
LD      C0

OUT     Y0
```



a) When X0 = On, RST instruction will be executed, PV in C0 will be “0” and the output contact will be reset to Off.

b) When X1 goes from Off to On, the PV in the counter will count up (plus 1).

c) When the counting of C0 reaches SV K5, the contact of C0 will be On and PV of C0 = SV = K5. The X1 trigger signal comes afterwards will not be accepted by C0 and the PV of C0 will stay at K5.

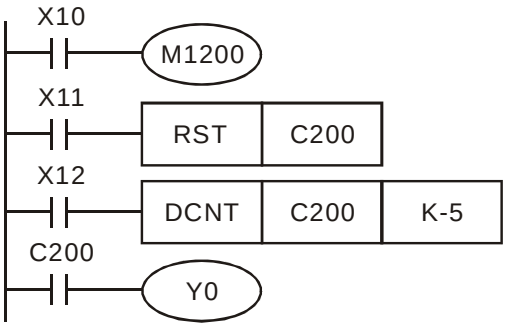


32-bit general purpose addition/subtraction counters C200 ~ C234:

1. The setup range of 32-bit counter: K-2,147,483,648 ~ K2,147,483,647 (not available for ES/EX/SS Series CPU).
2. Addition or subtraction of the counters is designated by On/Off status of special auxiliary relays M1200 ~ M1234. For example, when M1200 = Off, C200 will be an addition counter; when M1200 = On, C200 will be a subtraction counter.
3. The SV can be constant K or data register D (excluding special data registers D1000 ~ D1999). Data register D as SV can be a positive or negative value and an SV will occupy two consecutive data registers.
4. PV in the general purpose counter will be cleared when the power of the PLC is switched off. If the counter is a latched type, the counter will retain the PV and contact status before the power is off and resume the counting after the power is on again.
5. When the PV reaches up to 2,147,483,647, the next PV will turn to -2,147,483,648. When the PV reaches down to -2,147,483,648, the next PV will turn to 2,147,483,647.

Example:

```
LD      X10
OUT     M1200
LD      X11
RST     C200
LD      X12
CNT     C200   K-5
LD      C200
OUT     Y0
```



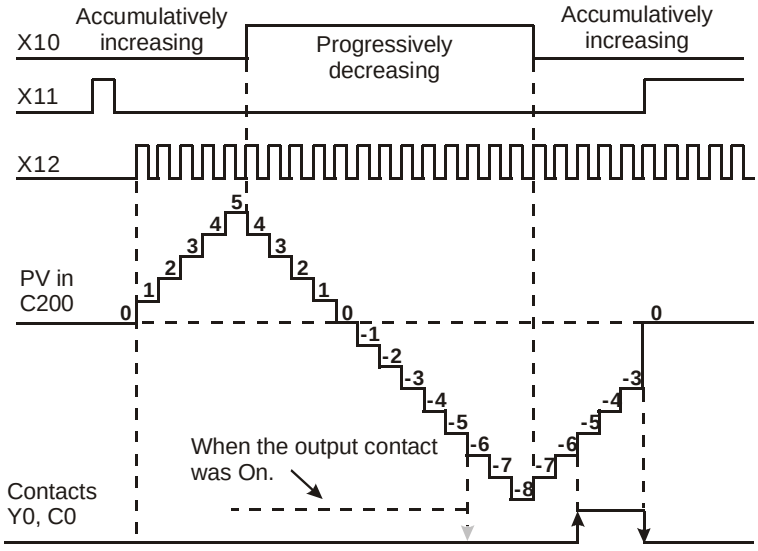
a) X10 drives M1200 to determine whether C200 is an addition or subtraction counter.

b) When X11 goes from Off to On, RST instruction will be executed and the PV in C200 will be cleared to “0” and the contact will be Off.

c) When X12 goes from Off to On, the PV in the counter will count up (plus 1) or count down (minus 1).

d) When the PV in C200 changes from K-6 to K-5, the contact of C200 will go from Off to On. When the PV in C200 changes from K-5 to K-6, the contact of C200 will go from On to Off.

e) If you use MOV instruction, WPLSoft or HPP to send a value bigger than the SV to the present value register of C0, next time when X1 goes from Off to On, the contact of counter C0 will be On and its PV will equal SV.



32-bit high-speed addition/subtraction counters C235 ~ C255:

1. The setup range of 32-bit counter: K-2,147,483,648 ~ K2,147,483,647
2. Addition or subtraction of C235 ~ C244 is designated by On/Off status of special auxiliary relays M1235 ~ M1244. For example, when M1235 = Off, C235 will be an addition counter; when M1235 = On, C235 will be a subtraction counter.
3. Addition or subtraction of C246 ~ C255 is designated by On/Off status of special auxiliary relays M1246 ~ M1255. For example, when M1246 = Off, C246 will be an addition counter; when M1246 = On, C246 will be a subtraction counter.
4. The SV can be constant K or data register D (excluding special data registers D1000 ~ D1999). Data register D as SV can be a positive or negative value and an SV will occupy two consecutive data registers.
5. If using DMOV instruction, WPLSoft or HPP to send a value which is large than the setting to any high-speed counter, next time when the input point X of the counter goes from Off to On, this contact will remain unchanged and it will perform addition and subtraction with the present value.
6. When the PV reaches up to 2,147,483,647, the next PV will turn to -2,147,483,648. When the PV reaches down to -2,147,483,648, the next PV will turn to 2,147,483,647.

■ **High-speed counters for ES/EX/EC3-8K Series CPU, total bandwidth: 20kHz**

Type / Input	1-phase input							1-phase 2 inputs			2-phase 2 inputs		
	C235	C236	C237	C238	C241	C242	C244	C246	C247	C249	C251	C252	C254
X0	U/D				U/D		U/D	U	U	U	A	A	A
X1		U/D			R		R	D	D	D	B	B	B
X2			U/D			U/D			R	R		R	R
X3				U/D		R	S			S			S

U: Progressively increasing input A: A phase input S: Input started

D: Progressively decreasing input B: B phase input R: Input cleared

1. Input points X0 and X1 can be planned as counters of higher speed with 1-phase 1 input reaching 20kHz. But the two counting frequencies added together have to be smaller or equal 20kHz. If the input is a 2-phase 2 input signal, the counting frequency will be approximately 4kHz. The 1-phase input of high-speed counters X2 and X3 and reach 10kHz.
2. The use of DHSCS instruction together with DHSCR instruction in ES Series CPU cannot exceed 4 times.

■ **High-speed counters for SX Series CPU, total bandwidth: 40kHz**

Type / Input	1-phase input									1-phase 2 inputs			2-phase 2 inputs			
	C235	C236	C237	C238	C239	C240	C241	C242	C244	C246	C247	C249	C251	C252	C253	C254
X0	U/D						U/D		U/D	U	U	U	A	A	B	A
X1		U/D					R		R	D	D	D	B	B	A	B
X2			U/D					U/D			R	R		R		R
X3				U/D				R	S			S				S
X4					U/D											
X5						U/D										

U: Progressively increasing input A: A phase input S: Input started

D: Progressively decreasing input B: B phase input R: Input cleared

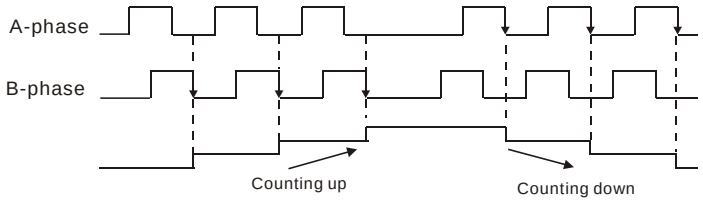
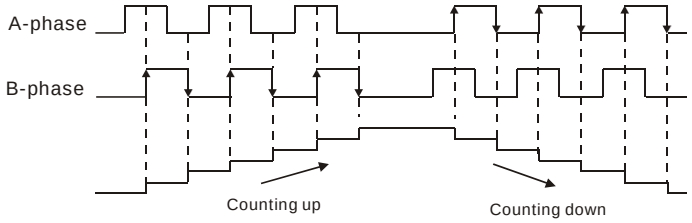
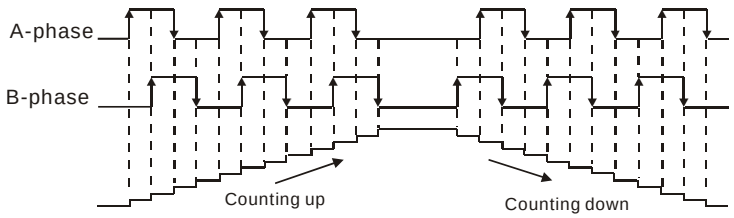
1. The frequency of input points X0 and X1 of 1-phase input can reach up to 20kHz, X2 ~ X5 can reach 10kHz. The frequency of C251, C252 and C254 of 2-phase input (X0, X1) can reach up to 4kHz. The maximum frequency of C253 is 4kHz (only supports 4 times frequency counting).
2. Input point X5 has two functions:
   a) When M1260 = Off, C240 will be normal U/D high-speed counter.
   b) When M1260 = On and DCNT instruction enables C240, X5 will be the shared reset signal for C235 ~ C239. Counter C240 will continue to receive the input signals from X5.

Counting modes:

a) The 2-phase 2 inputs counting mode of the high speed counters in ES/EX/EC/EC3-8K (V5.5 and versions above) and SX Series CPU is set by special D1022 with normal frequency, double frequency and 4 times frequency modes. The contents in D1022 will be loaded in the first scan when PLC is switched from STOP to RUN.

Device No.	Function
D1022	Setting up the multiplied frequency of the counter
D1022 = K1	Normal frequency mode selected
D1022 = K2 or 0	Double frequency mode selected (default)
D1022 = K4	4 times frequency mode selected

b) Multiplied frequency mode (↑↓ indicates the occurrence of counting)

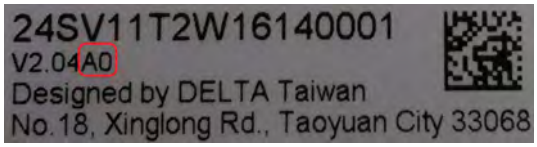
Counting mode		Counting wave pattern
2-phase 2 inputs	1 (Normal frequency)	
	2 (Double frequency)	
	4 (4 times frequency)	

EH3/SV2 Series CPU supports high speed counters. C235~C240 are program-interruption 1-phase high speed counter, and can be used with a counting frequency of up to 10kHz.

C241 ~ C254 are hardware high speed counter (HHSC). See the table below.

	HHSC1	HHSC2	HHSC3	HHSC4
Counter number	C241, C246, C251	C242, C247, C252	C243, C248, C253	C244, C249, C254
16,20 EH3 SV2 hardware version: A0, A1	200 kHz	200 kHz	20 kHz	20 kHz
32,40,48,64,80 EH3 SV2 hardware version: A2	200 kHz	200 kHz	200 kHz	200 kHz

You can find the hardware version on the side of the CPU. See the example photo below.



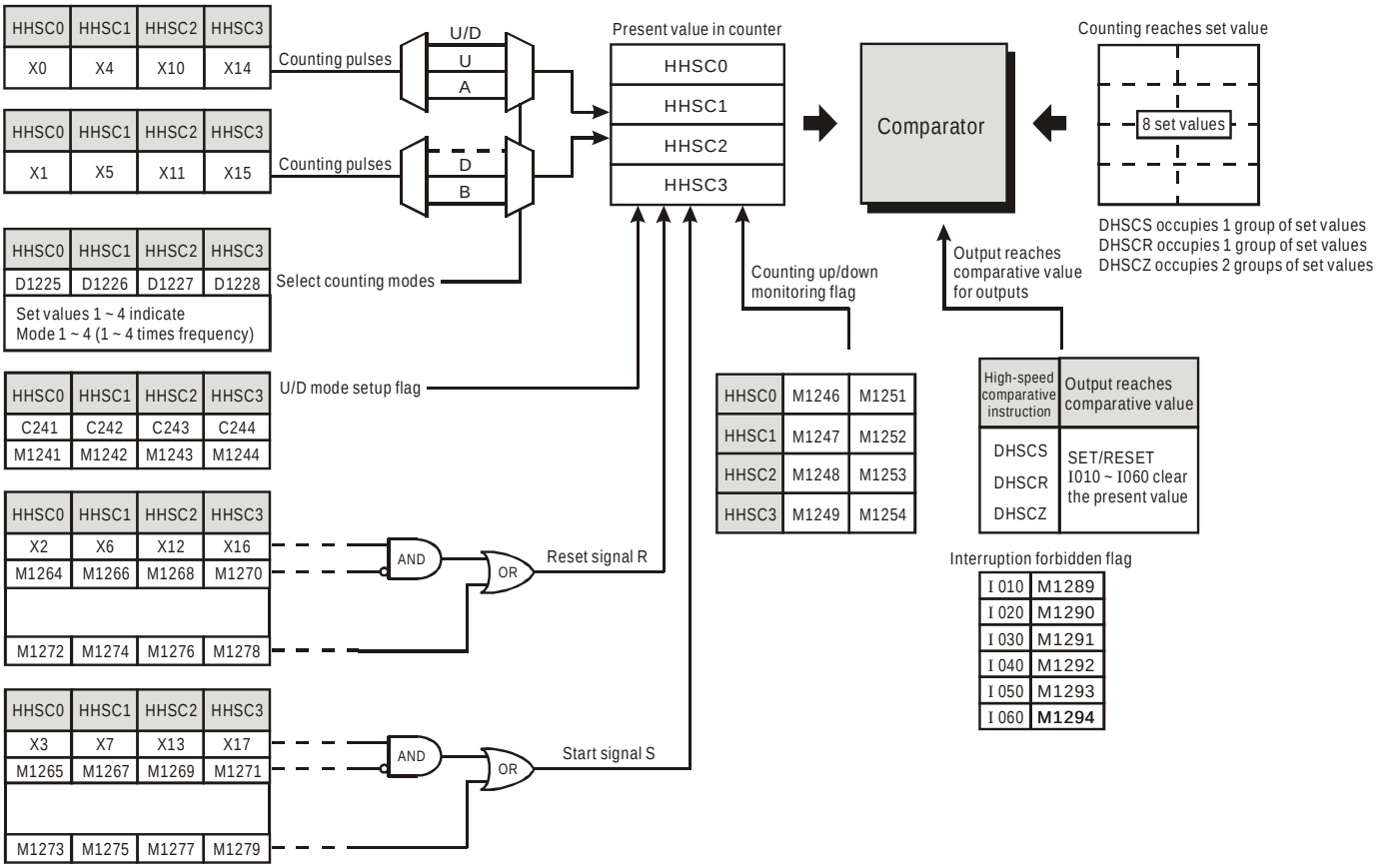
1. Every HHSC can only be designated to one counter by DCNT instruction.
2. Each HHSC contains three kinds of counter mode.
   a) 1-phase 1 input refers to “pulse/direction” mode.
   b) 1-phase 2 inputs refers to “clockwise/counterclockwise (CW/CCW)” mode.
   c) 2-phase 2 inputs refers to “A-B phase” mode.

Counter type	Program-interruption high speed counter						Hardware high speed counter											
Input \ Type	1-phase 1 input						1-phase 1 input				1-phase 2 inputs				2-phase 2 inputs			
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C246	C247	C248	C249	C251	C252	C253	C254
X0	U/D						U/D				U				A			
X1		U/D									D				B			
X2			U/D				R				R				R			
X3				U/D			S				S				S			
X4					U/D			U/D				U				A		
X5						U/D						D				B		
X6								R				R				R		
X7								S				S				S		
X10									U/D				U				A	
X11													D				B	
X12									R				R				R	
X13									S				S				S	
X14										U/D				U				A
X15														D				B
X16										R				R				R
X17										S				S				S

U: Progressively increasing input A: A phase input S: Input started
B: Progressively decreasing input B: B phase input R: Input cleared

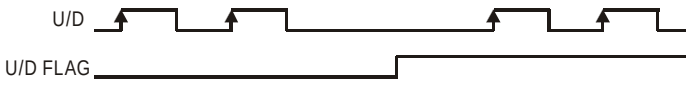
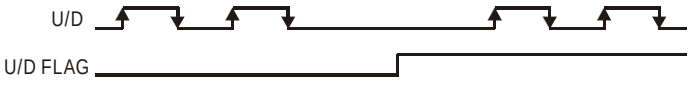
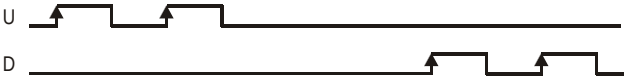
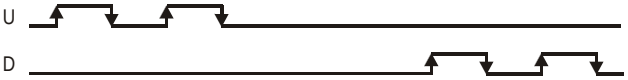
3. System structure of the hardware high speed counters:
   a) HHSC0 ~ 3 have reset signals and start signals from external inputs. Settings in M1272, M1274, M1276 and M1278 are reset signals of HHSC0, HHSC1, HHSC2 and HHSC3. Settings in M1273, M1275, M1277 and M1279 are start signals of HHSC0, HHSC1, HHSC2 and HHSC3.
   b) If the external control signal inputs of R and S are not in use, you can set M1264/M1266/M1268/M1270 and M1265/M1267/M1269/M1271 as True and disable the input signals. The corresponding external inputs can be used again as general input points (see the figure below).

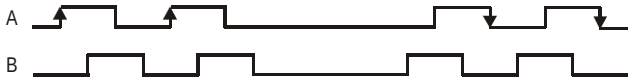
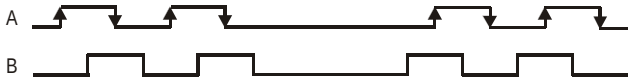
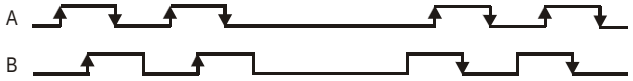
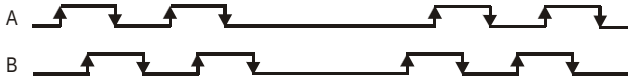
c) When special M is used as a high speed counter, the inputs controlled by START and RESET will be affected by the scan time.



4. Counting modes:

The counting modes of the hardware high-speed counters in EH3/SV2 Series CPU can be set in D1225 ~ D1228.

Counting modes		Wave pattern	
Type	Set value in special D	Counting up(+1)	Counting down(-1)
1-phase 1 input	1 (Normal frequency)		
	2 (Double frequency)		
1-phase 2 inputs	1 (Normal frequency)		
	2 (Double frequency)		

Counting modes		Wave pattern
Type	Set value in special D	Counting up(+1) Counting down(-1)
2-phase 2 inputs	1 (Normal frequency)	
	2 (Double frequency)	
	3 (Triple frequency)	
	4 (4 times frequency)	

Special registers for relevant flags and settings of high speed counters:

Flag	Function
M1150	DHSZ instruction in multiple set values comparison mode
M1151	The execution of DHSZ multiple set values comparison mode is completed.
M1152	Set DHSZ instruction as frequency control mode
M1153	DHSZ frequency control mode has been executed.
M1235 ~ M1244	Designating the counting direction of high speed counters C235 ~ C245 When M12□□ = Off, C2□□ will perform a counting up. When M12□□ = On, C2□□ will perform a counting down.
M1245~ M1255	Designating the counting direction of high speed counters C246 ~ C255 When M12□□ = Off, C2□□ will perform a counting up. When M12□□ = On, C2□□ will perform a counting down.
M1160	X5 as the reset input signal of all high speed counters
M1261	High-speed comparison flag for DHSCR instruction
M1264	Disable the external control signal input point of HHSC0 reset signal point (R)
M1265	Disable the external control signal input point of HHSC0 start signal point (S)
M1266	Disable the external control signal input point of HHSC1 reset signal point (R)
M1267	Disable the external control signal input point of HHSC1 start signal point (S)
M1268	Disable the external control signal input point of HHSC2 reset signal point (R)
M1269	Disable the external control signal input point of HHSC2 start signal point (S)
M1270	Disable the external control signal input point of HHSC3 reset signal point (R)
M1271	Disable the external control signal input point of HHSC3 start signal point (S)

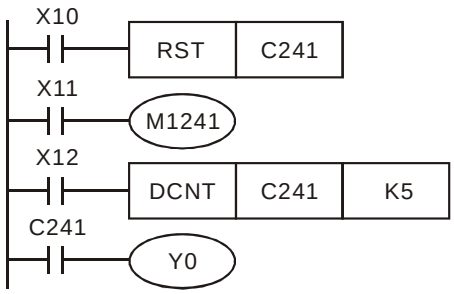
Flag	Function
M1272	Internal control signal input point of HHSC0 reset signal point (R)
M1273	Internal control signal input point of HHSC0 start signal point (S)
M1274	Internal control signal input point of HHSC1 reset signal point (R)
M1275	Internal control signal input point of HHSC1 start signal point (S)
M1276	Internal control signal input point of HHSC2 reset signal point (R)
M1277	Internal control signal input point of HHSC2 start signal point (S)
M1278	Internal control signal input point of HHSC3 reset signal point (R)
M1279	Internal control signal input point of HHSC3 start signal point (S)
M1289	High speed counter I010 interruption forbidden
M1290	High speed counter I020 interruption forbidden
M1291	High speed counter I030 interruption forbidden
M1292	High speed counter I040 interruption forbidden
M1293	High speed counter I050 interruption forbidden
M1294	High speed counter I060 interruption forbidden
M1312	C235 Start input point control (not supported by EH3/SV2)
M1313	C236 Start input point control (not supported by EH3/SV2)
M1314	C237 Start input point control (not supported by EH3/SV2)
M1315	C238 Start input point control (not supported by EH3/SV2)
M1316	C239 Start input point control (not supported by EH3/SV2)
M1317	C240 Start input point control (not supported by EH3/SV2)
M1320	C235 Reset input point control (not supported by EH3/SV2)
M1321	C236 Reset input point control (not supported by EH3/SV2)
M1322	C237 Reset input point control (not supported by EH3/SV2)
M1323	C238 Reset input point control (not supported by EH3/SV2)
M1324	C239 Reset input point control (not supported by EH3/SV2)
M1325	C240 Reset input point control (not supported by EH3/SV2)
M1328	Enable Start/Reset of C235 (not supported by EH3/SV2)
M1329	Enable Start/Reset of C236 (not supported by EH3/SV2)
M1330	Enable Start/Reset of C237 (not supported by EH3/SV2)
M1331	Enable Start/Reset of C238 (not supported by EH3/SV2)
M1332	Enable Start/Reset of C239 (not supported by EH3/SV2)
M1333	Enable Start/Reset of C240 (not supported by EH3/SV2)
D1022	Multiplied frequency of A-B phase counters for ES/EX/SS and SX Series CPU
D1150	Table counting register for DHSZ multiple set values comparison mode
D1151	Register for DHSZ instruction frequency control mode (counting by table)
D1152 (low word) D1153 (high word)	In frequency control mode, DHSZ reads the upper and lower limits in the table counting register D1153 and D1152.

Flag	Function
D1166	Switching between rising/falling edge counting modes of X10 (for SC_V1.4 Series CPU only)
D1167	Switching between rising/falling edge counting modes of X11 (for SC_V1.4 Series CPU only)
D1225	The counting mode of the 1st group counters (C241, C246, C251)
D1226	The counting mode of the 2nd group counters (C242, C247, C252)
D1227	The counting mode of the 3rd group counters (C243, C248, C253)
D1228	The counting mode of the 4th group counters (C244, C249, C254)
D1225 ~ D1228	Counting modes of HHSC0 ~ HHSC3 in EH2/SV/EH3/SV2 Series CPU (default = 2) 1: Normal frequency counting mode 2: Double frequency counting mode 3: Triple frequency counting mode 4: 4 times frequency counting mode

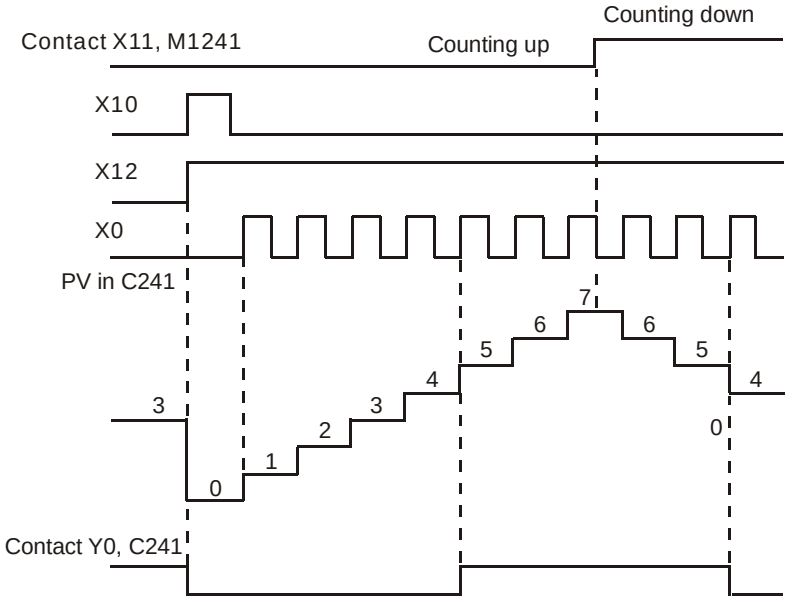
1-phase 1 input high-speed counter

Example:

```
LD      X10
RST     C241
LD      X11
OUT     M1241
LD      X12
DCNT    C241   K5
LD      C241
OUT     Y0
```



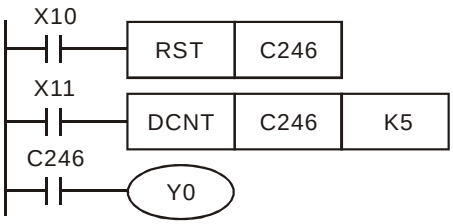
1. X11 drives M1241 to determine whether C241 is an addition or subtraction counter.
2. When X10 is On, RST instruction will be executed and the PV in C241 will be cleared to “0” and the contact will be Off.
3. In C241, when X12 is On and C241 receives the signals from X0, the PV in the counter will count up (plus 1) or count down (minus 1).
4. When the counting of C241 reaches SV K5, the contact of C241 will be On. If there are still input signals from X0, the counting will continue.
5. C241 in ES/EX/EC/EC3-8K and SX Series CPU has external input signals to reset X1.
6. C241 in EH3/SV2 Series CPU has external input signals to reset X2 and start X3.
7. The external input contact of reset signal of C241 (HHSC0) in EH3/SV2 Series CPU is disabled by M1264. The external input contact of start signal is disabled by M1265.
8. The internal input contact of reset signal of C241 (HHSC0) in EH3/SV2 Series CPU is disabled by M1272. The internal input contact of start signal is disabled by M1273.
9. The counting modes (normal frequency or double frequency) of C246 (HHSC0) in EH3/SV2 Series CPU can be set up by D1225. The default setting is double frequency mode.



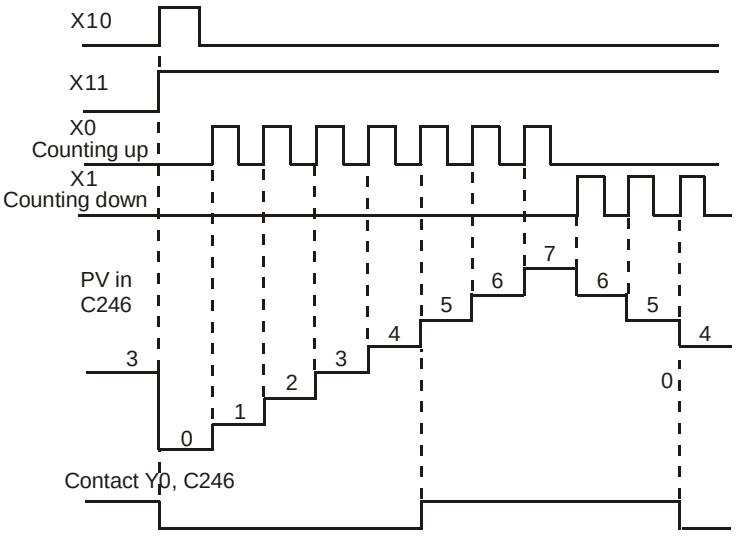
1-phase 2 inputs high-speed counter

Example:

LD	X10
RST	C246
LD	X11
DCNT	C246 K5
LD	C246
OUT	Y0



1. When X10 is On, RST instruction will be executed. The PV in C246 will be cleared to “0” and the output contact will be reset to be Off.
2. In C246, when X11 is On and C246 receives the signals from X0, the PV in the counter will count up (plus 1) or count down (minus 1).
3. When the counting of C246 reaches SV K5, the contact of C246 will be On. If there are still input signals from X0, the counting will continue.
4. C246 in EH3/SV2 Series CPU has external input signals to reset X2 and start X3.

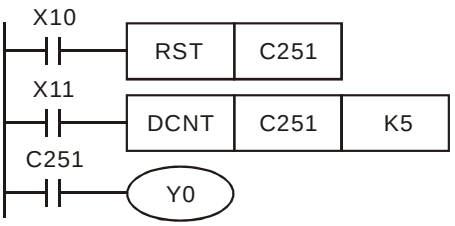


5. The counting modes (normal frequency or double frequency) of C246 (HHSC0) in EH3/SV2 Series CPU can be set up by D1225. The default setting is double frequency mode.
6. The external input contact of reset signal of C246 (HHSC0) in EH3/SV2 Series CPU is disabled by M1264. The external input contact of start signal is disabled by M1265.
7. The internal input contact of reset signal of C246 (HHSC0) in EH3/SV2 Series CPU is disabled by M1272. The internal input contact of start signal is disabled by M1273.

2-phase AB input high-speed counter

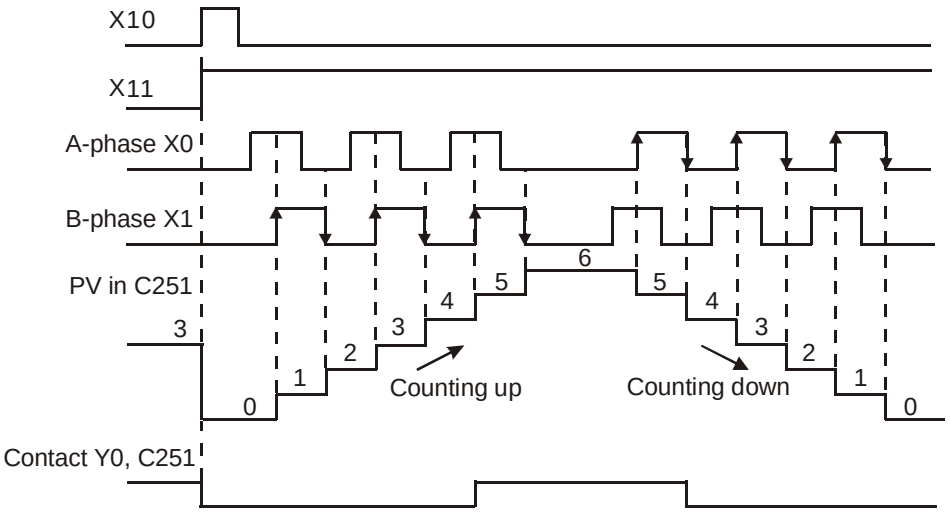
Example:

```
LD      X10
RST     C251
LD      X11
DCNT    C251 K5
LD      C251
OUT     Y0
```

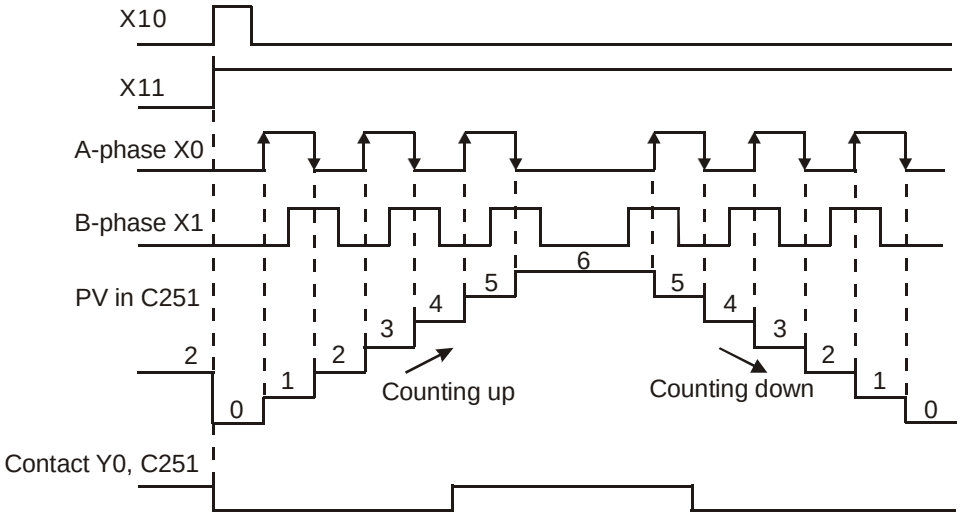


1. When X10 is On, RST instruction will be executed. The PV in C251 will be cleared to “0” and the output contact will be reset to be Off.
2. In C251, when X11 is On and C251 receives the A-phase signals from X0 and B-phase signals from X1, the PV in the counter will count up (plus 1) or count down (minus 1). You can select different counting modes if you use EH3/SV2 Series CPU.
3. When the counting of C251 reaches SV K5, the contact of C251 will be On. If there are still input signals coming in, the counting will continue.
4. The counting modes (normal frequency, double frequency or 4 times frequency) of C251 (HHSC0) in ES/EC3-8K Series CPU can be set up by D1022. The default setting is double frequency mode.
5. C251 in EH3/SV2 Series CPU has external input signals to reset X2 and start X3.
6. The counting modes (normal frequency, double frequency, triple frequency or 4 times frequency) of C251 (HHSC0) in EH3/SV2 Series CPU can be set up by D1225. The default setting is double frequency mode.
7. The external input contact of reset signal of C246 (HHSC0) in EH3/SV2 Series CPU is disabled by M1264. The external input contact of start signal is disabled by M1265.
8. The internal input contact of reset signal of C246 (HHSC0) in EH3/SV2 Series CPU is disabled by M1272. The internal input contact of start signal is disabled by M1273.

ES/EX/EC3-8K and SX Series CPU (double frequency)



EH3/SV2 Series CPU (double frequency)



2.8 Numbering and Functions of Registers [D], [E], [F]

2.8.1 Data register [D]

A data register is for storing a 16-bit datum of values between -32,768 to +32,767. The highest bit is “+” or “-" sign. Two 16-bit registers can be combined into a 32-bit register (D + 1; D of smaller No. is for lower 16 bits). The highest b it is “+” or “-” sign and it can store a 32-bit datum of values between -2,147,483,648 to +2,147,483,647.

■ **ES/EX/EC Series CPU:**

Data register D	General purpose	D0 ~ D407, 408 points	Total 912 points
	Latched	D408 ~ D599, 192 points. Fixed to be latched.	
	Special purpose	D1000 ~ D1311, 312 points. Some are latched.	
	Index register E, F	E, F, 2 points	

■ **EC3-8K Series CPU:**

Data register D	General purpose	D0 ~ D407, 408 points D600 ~ D999, 400 points D3920 ~ D4999, 1080 points	Total 5000 points
	Latched	D408 ~ D599, 192 points. Fixed to be latched. D2000 ~ D3919, 1920 points. Fixed to be latched.	
	Special purpose	D1000 ~ D1999, 1000 points. Some are latched.	
	Index register E, F	E0 ~ E7, F0 ~ F7, 16 points	

■ **SX Series CPU:**

Data register D	General purpose	D0 ~ D199, 200 points. Fixed to be non-latched.	Total 5,000 points (SX v.3.0 and above: 10,000 points)
	Latched	D200 ~ D999, D2000 ~ D4999, 3,800 points. Can be modified to be non-latched by setting up parameter.	
	Special purpose	D1000 ~ D1999, 1000 points. Some are latched.	
	General purpose	D5000~D9999, 5000 points (Only supported by SX v.3.0 and above) Fixed to be non-latched.	
	Index register E, F	E0 ~ E3, F0 ~ F3, 8 points	
File register		K0 ~ K1,599, CPU 1,600 points. Fixed to be latched.	1,600 points

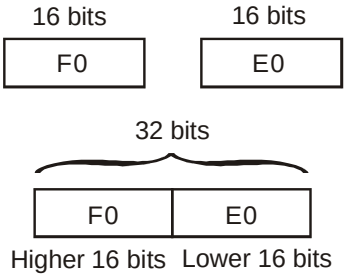
■ **EH3/SV2 Series CPU:**

Data register D	General purpose	D0 ~ D199, 200 points. Can be modified to be latched by setting up parameters.	Total 10,000 points
	Latched	D200 ~ D999, 800 points. D200 ~ D9799, 7800 points D10000 ~ D11999, 2,000 points. Can be modified to be non-latched by setting up parameters.	
	Special purpose	D1000 ~ D1999, 1,000 pints. Some are latched.	
	For right-side modues	D9900~D9999, 100 points	
	For left-side modules	D9800~D9899, 100 points	
	Index register E, F	E0 ~ E7, F0 ~ F7, 16 points.	
File register		K0 ~ K9,999, CPU 10,000 points. Fixed to be latched.	10,000 points

There are five types of registers:

1. **General purpose register:** When PLC goes from RUN to STOP or the power of the PLC is switched off, the data in the register will be cleared to “0”. When M1033 = On and PLC goes from RUN to STOP, the data will not be cleared, but will still be cleared to “0” when the power is off.
2. **Latched register:** When the power of PLC is switched off, the data in the register will not be cleared but will retain at the value before the power is off. You can use RST or ZRST instruction to clear the data in the latched register.
3. **Special purpose register:** Every register of this kind has its special definition and purpose, mainly for storing the system status, error messages and monitored status. See 2.10 and 2.11 for more details.
4. **Index register E, F:** The index register is a 16-bit register. There are 2 points of E, F in ES/EX/EC Series CPU; 8 points (E0 ~ E3, F0 ~ F3) in SX Series CPU; 16 points (E0 ~ E7, F0 ~ F7) in EC3-8K/EH3/SV2 Series CPU. If the index register is to be used as a 32-bit register, please designate E. When E is already designated in a 32-bit instruction, using also F will not be allowed.
5. **File register:** There are 1,600 file registers (K0 ~ K1,599) in SX Series CPU and 10,000 file registers (K0 ~ K9,999) in EH3/SV2 Series CPU. The file register does not have an exact device No.; therefore the read/write function of file registers has to be executed by instruction API 148 MEMR, API 149 MEMW or through WPLSoft.

2.8.2 Index Register [E], [F]

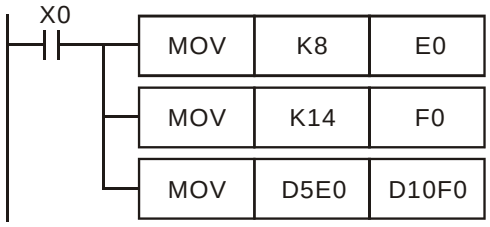


Index registers E, F are 16-bit data registers and can be written and read.

If you need to use a 32-bit register, you have to designate E. In this case, F will be covered by E and cannot be used anymore; otherwise, the content in E (32-bit) will be incorrect. We suggest you use DMOVP K0 E instruction, the content in E (including F) will be cleared to “0” when the power of PLC is switched on.

The combination of E, F when you use a 32-bit index register:

(F0, E0), (F1, E1), (F2, E2), …(F7, E7)



When X0 = On, E0 = 8, F0 = 14, D5E0 = D (5 + 8) = D13, D10F0 = D (10 + 14) = D24. At this moment, the content in D13 will be moved to D24.

The index register is the same as normal operands, can be used for moving or comparison on word devices (KnX, KnY, KnM, KnS, T, C, D) and bit devices (X, Y, M, S). ES/EC/EC3-8K Series CPU does not support constant (K, H) index register, but EH2/SV/EH3/SV2 Series CPU supports constant (K, H) index register.

ES/EX/EC Series CPU has 2 points of index registers E0, F0

SX Series CPU has 8 points of index registers E0 ~ E3, F0 ~ F3

EC3-8K/EH3/SV2 Series CPU has 16 points of index registers E0 ~ E7, F0 ~ F7

- Some instructions do not support index registers. For how to use index register E, F to modify the operands, see Chapter 5.3 for more details.

- When you use the instruction mode in WPLSoft to generate constant (K, H) index register function, please use symbol "@". For example, "MOV K10@E0 D0F0"
- When you use index register E, F to modify the operands, the modification range CANNOT exceed the range of special purpose registers D1000 ~ D1999 and special auxiliary registers M1000 ~ M1999 in case errors may occur.

2.8.3 Functions and Features of File Registers

When the power of PLC is switched on, SX and EH3/SV2 Series CPU will check the following devices:

1. M1101 (whether the file register is enabled)
2. D1101 (No. of file registers in SX Series CPU: K0 ~ K1,599; No. of file registers in EH3/SV2 Series CPU: K0 ~ K9,999)
3. D1102 (Number of file registers to be read in SX Series CPU: K0 ~ K1,600; number of file registers to be read in EH3/SV2 Series CPU: K0 ~ K8,000)
4. D1103 (devices for storing the data read from file registers; the No. of designated data register D starts from K2,000 ~ K9,999; determining whether to automatically send the content in the file register to the designated data register.)

Note:

1. When D1101 of SX Series CPU is bigger than 1,600, D1101 of EH3/SV2 Series CPU is bigger than 8,000 and D1103 is smaller than 2,000 or bigger than 9,999, the data read from file registers will not be sent to data register D.
2. When the program starts to send the data read from the file register to data register D and the address of the file register or the data register D exceed their ranges, PLC will stop the reading.
3. There are 1,600 file registers in SX Series CPU and 10,000 in EH3/SV2 Series CPU. The file register does not have an exact device No.; therefore the read/write function of file registers has to be executed by instruction API 148 MEMR, API 149 MEMW or through peripheral devices HPP and WPLSoft.
4. If you tend to read a file register with an address that is not within the range, the read value will be "0".

2.9 Pointer [N], Pointer [P], Interruption Pointer [I]

■ **ES/EX/EC Series CPU:**

Pointer	N	For master control loop		N0 ~ N7, 8 points	Control point of master control loop
	P	For CJ, CALL instructions		P0 ~ P63, 64 points	Position pointer of CJ, CALL
	I	Interruption	External interruption	I001, I101, I201, I301, 4 points	Position pointer of interruption subroutine
			Timed interruption	I6□□, 1 point (□□=10 ~ 99, time base = 1ms) (for V5.7)	
			Communication interrupt	I150, 1 point	

■ **EC3-8K Series CPU:**

Pointer	N	For master control loop		N0 ~ N7, 8 points	Control point of master control loop
	P	For CJ, CALL instructions		P0 ~ P255, 256 points	Position pointer of CJ, CALL
	I	Interruption	External interruption	I001, I101, I201, I301, I401, I501, I1601, I701, 8 points	Position pointer of interruption subroutine
			Timed interruption	I6□□, I7□□, 2 points (□□=2~99, time base=1ms); I8□□, 1 point (□□=10~99, time base = 0.1ms)	
			Communication interrupt	I150, 1 point	

■ **SX Series CPU:**

Pointer	N	Master control loop		N0 ~ N7, 8 points	Control point of master control loop
	P	For CJ, CALL instructions		P0 ~ P255, 256 points	Position pointer of CJ, CALL
	I	Interruption	External interruption	I001, I101, I201, I301, I401, I501, 6 points	Position pointer of interruption subroutine
			Timer interruption	I6□□, I7□□, 2 points (□□ = 1 ~ 99, time base = 1ms)	
			High-speed counter interruption	I010, I020, I030, I040, I050, I060, 6 points	
			Communication interruption	I150, 1 point	

Note: Among the 6 pairs of interruption No. (I001, I010), (I101, I020), (I201, I030), (I301, I040), (I401, I050), (I501, I060), only 1 No. in the pair is allowed to be used in the program. If you use both No. in the pair and write them into the program, there may be syntax errors occurring.

■ **EH3/SV2 Series CPU:**

Pointer	N	Master control loop		N0 ~ N7, 8 points	Control point of master control loop
	P	For CJ, CALL instructions		P0 ~ P255, 256 points	Position pointer of CJ, CALL
	I	Interruption	External interruption	I00□(X0), I10□(X1), I20□(X2), I30□(X3), I40□(X4), I50□(X5), I60□(X6), I70□(X7), I90□(X10), I91□(X11), I92□(X12), I93□(X13), I94□(X14), I95□(X15), I96□(X16), I97□(X17), 16 points (□=1, rising-edge trigger ↑, □=0, falling-edge trigger ↓)	Position pointer of interruption subroutine
			Timed interruption	I6□□, I7□□, 2 points (□□ = 02 ~ 99, time base = 1ms) I8□□, 1 point (□□ = 05 ~ 99, time base = 0.1ms)	
			High-speed counter interruption	I010, I020, I030, I040, I050, I060, 6 points	
			Pulse interruption	I110, I120, I130, I140, 4 points	
			Communication interruption	I150, I160, I170, 3 points	

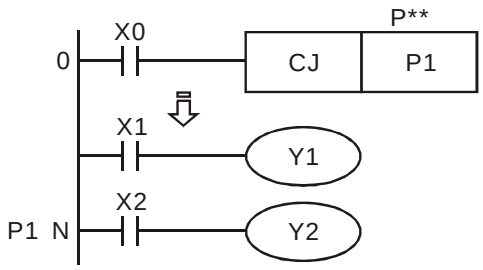
Note 1: Input point X as a high-speed counter cannot be used as an external interruption signal. For example, if C251 occupies X0,

X1, X2 and X3, the external input interruption No. I00□(X0), I10□(X1), I20□(X2), and I30□(X3) cannot be used.
Note 2: If an interrupt subroutine is executed, the next interrupt subroutine will not be executed until the execution of the interrupt is complete.
Note 3: The time it takes for an interrupt subroutine in a PLC to be executed affects the efficiency of the PLC. It is suggested that the size of an interrupt subroutine not be large.

Pointer N: Used with MC and MCR instructions. MC is the master control start instruction. When MC instruction is executer, the instructions between MC and MCR will still be executed normally. See Chapter 3 explanations on MC and MCR instructions for more details.

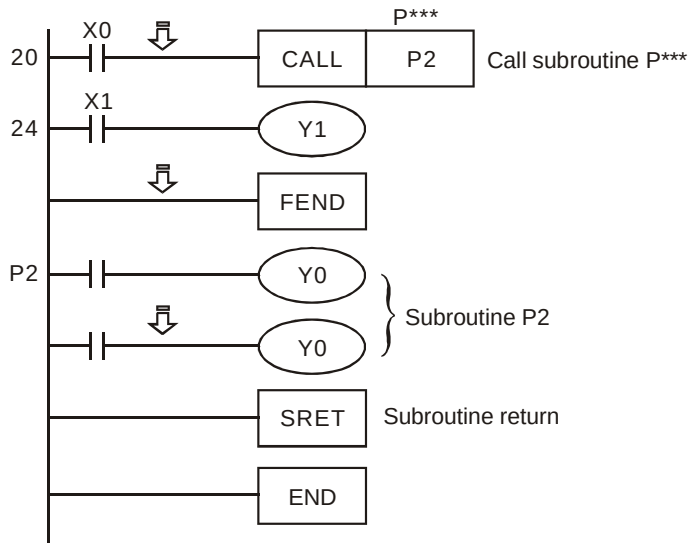
Pointer P: Used with API 00 CJ, API 01 CALL and API 02 SRET. See Chapter 6 explanations on CJ, CALL and SRET instructions for more details.

CJ Conditional Jump:



- When X0 = On, the program will jump from address 0 to N (designated label P1) and keep on the execution. The addresses in the middle will be ignored.
- When X0 = Off, the program will execute from address 0 and keep on executing. At this time, CJ instruction will not be executed.

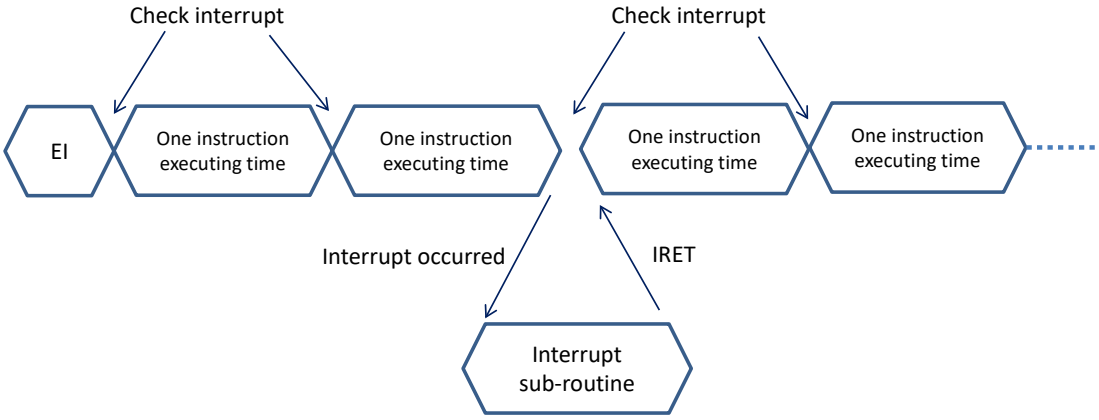
CALL Call Subroutine, SRET Subroutine Return:



- When X0 = On, CALL instruction will be executed and the program will jump to P2 and executed the designated subroutine. When SRET instruction is executed, the program will return to address 24 and keep on the execution.

Interruption Pointer I: Used with API 04 EI, API 05 DI, API 03 IRET. See Chapter 6 for more details. There are 6 types of interruption pointer. To insert an interruption, you have to combine the action with EI (enable interruption), DI (disable interruption), IRET (interruption return) instructions.

When you use EI instruction to activate an interruption, PLC checks if there is any interrupt occurs automatically after each instruction is executed. If the interrupt is inserted in the middle of the instrucitons in the program, PLC stops executing the current instruction in the main program and starts executing the interrupt. If the interrupt is not inserted in the middle of the instructions, PLC executes the instructions in the program by order. Thus, the longest time for the interrupt to occur is one instruction executing time.



1. External interruption: Due to the special hardware design inside the CPU, the input signals coming in at input terminals X0 ~ X5 (EH3/SV2: X0~X17) when rising-edge or falling-edge triggers will not be affected by the scan cycle. The currently executed program will be interrupted immediately and the execution will jump to the designated interruption subroutine pointer I00□(X0), I10□(X1), I20□(X2), I30□(X3), I40□(X4), I50□(X5). Till the execution reaches IRET instruction, the program will return to the original position and keep on its execution.
   In SX Series CPU, X0 (pulse input point) works with X4 (external interruption point), corresponding to C235, C251 and C253 work with I401, which will be able to interrupt and intercept the present value in the high-speed counter. D1181 is the device to store the 32-bit value. X1 (pulse input point) works with X5 (external interruption point), corresponding to C236 works with I501, which will be able to interrupt and intercept the present value in the high-speed counter. D1198 and D1199 are the devices to store the 32-bit value.
2. Timed interruption: PLC automatically interrupts the currently executed program every a fixed period of time and jumps to the execution of a designated interruption subroutine.
3. Interruption when the counting reaches the target: The high-speed counter comparison instruction API 53 DHSCS can designates that when the comparison reaches the target, the currently executed program will be interrupted and jump to the designated interruption subroutine executing the interruption pointers I010, I020, I030, I040, I050 and I060.
4. Pulse interruption: The pulse output instruction API 57 PLSY can be set up that the interruption signal is sent out synchronously when the first pulse is sent out by enabling flags M1342 and M1343. The corresponding interruptions are I130 and I140. You can also set up that the interruption signal is sent out after the last pulse is sent out by enabling flags M1340 and M1341. The corresponding interruptions are I110 and I120.
5. Communication interruption:
   I150: After COM2 receives a specific character by means of the communication instruction RS, I150 will be enabled. The specific character is set in the low byte in D1168. If a PLC is connected to a communication device,

and the length of the data that the PLC receives is not the same, this function can be used.

I160: After COM2 receives a certain number of data by means of the communication instruction RS, I160 will be enabled. The number of data can be set in the low byte in D1169. If D1169 = 0, I160 will not be triggered.

I170: After the slave station COM2 finishes receiving data, I170 will be enabled. Normally when the communication terminal of the PLC is in Slave mode, PLC will not immediately process the communication data entered but process it after the END is executed. Therefore, when the scan time is very long and you need the communication data to be processed immediately, you can use interruption I170 for this matter.

I151, I161, I153, and I163 are only applicable to EH3/EH3-L/SV2 version 2.00 and above.

I151: After COM1 receives a specific character by means of the communication instruction RS, I151 will be enabled. The specific character is set in the low byte in D1397. If a PLC is connected to a communication device, and the length of the data that the PLC receives is not the same, this function can be used. If D1397 = 0, I151 will not be triggered.

I161: After COM1 receives a certain number of data by means of the communication instruction RS, I161 will be enabled. The number of data can be set in the low byte in D1398. If D1398 = 0, I161 will not be triggered.

I153: After COM3 receives a specific character by means of the communication instruction RS, I153 will be enabled. The specific character is set in the low byte in D1242. If a PLC is connected to a communication device, and the length of the data that the PLC receives is not the same, this function can be used. If D1242 = 0, I153 will not be triggered.

I163: After COM3 receives a certain number of data by means of the communication instruction RS, I163 will be enabled. The number of data can be set in the low byte in D1243. If D1243 = 0, I163 will not be triggered.

In the program in a EH3/SV2 series PLC, three communication interrupts at most can be enabled. Please see the table below for more information. (SV2 series PLCs do not support COM3.)

Communication interrupt number	1	2	3
COM1 communication interrupt	I161	I151	
COM2 communication interrupt	I150	I160	I170
COM3 communication interrupt	I163		I153

Example: If the COM1 communication interrupt I161 has been selected, the communication interrupts I150 and I163 can not be used. Although there is no such warning during the writing of a program, a warning message will appear after the program is downloaded to a PLC.

2.10 Special Auxiliary Relays and Special Data Registers

The types and functions of special auxiliary relays (special M) and special data registers (special D) are listed in the table below. Please be noted that some devices of the same No. may bear different meanings in different Series CPUs. Special M and special D marked with "*" will be further illustrated in the 2.11. Columns marked with "R" refers to "read only", "R/W" refers to "read and write", "-" refers to the status remains unchanged and "#" refers to the system will set it up according to the status of the PLC.

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1000*	Monitoring normally open contact (A)	○	○	○	○	Off	On	Off	R	NO	Off
M1001*	Monitoring normally closed contact (B)	○	○	○	○	On	Off	On	R	NO	On
M1002*	Enable single positive pulse at the moment when RUN is activated (Normally OFF)	○	○	○	○	Off	On	Off	R	NO	Off
M1003*	Enable single negative pulse at the moment when RUN is activated (Normally ON)	○	○	○	○	On	Off	On	R	NO	On
M1004*	On when syntax errors occur	○	○	○	○	Off	Off	-	R	NO	Off
M1005*	Password of data backup memory card and password of CPU do not match	×	×	×	○	Off	-	-	R	NO	Off
M1006*	Data backup memory card has not been initialized	×	×	×	○	Off	-	-	R	NO	Off
M1007*	Data do not exist in the program area of data backup memory card	×	×	×	○	Off	-	-	R	NO	Off
M1008*	Scanning watchdog timer (WDT) On	○	○	○	○	Off	Off	-	R	NO	Off
M1009	Insufficient 24V DC supply, LV signal has been occurred.	○	○	○	○	Off	-	-	R	NO	Off
M1010	ES/EC/EC3-8K: PLSY Y0 mode selection, continuous output when On EH3/SV2: Pulse output when reaching END instruction	○	○	○	○	Off	-	-	R/W	NO	Off
M1011*	10ms clock pulse, 5ms On/5ms Off	○	○	○	○	Off	-	-	R	NO	Off
M1012*	100ms clock pulse, 50ms On / 50ms Off	○	○	○	○	Off	-	-	R	NO	Off
M1013*	1s clock pulse, 0.5s On / 0.5s Off	○	○	○	○	Off	-	-	R	NO	Off
M1014*	1min clock pulse, 30s On / 30s Off	○	○	○	○	Off	-	-	R	NO	Off
M1015*	Enabling high-speed counter	×	○	×	○	Off	-	-	R/W	NO	Off
M1016*	Displaying real time clock in A.D.	×	○	×	○	Off	-	-	R/W	NO	Off
M1017*	±30 seconds correction on real time clock	×	○	×	○	Off	-	-	R/W	NO	Off
M1018	Flag for radian/degree, On: for degree	×	○	○	○	Off	-	-	R/W	NO	Off
M1019	The error LED keeps blinking or not, when the supplied 24V power is from NOT stable to stable. ON: The error LED stops blinking once the power is stable again. OFF: The error LED keeps blinking even when the power is stable again.	×	×	×	V2.06	Off	-	-	R/W	NO	Off
M1020	Zero flag	○	○	○	○	Off	-	-	R	NO	Off
M1021	Borrow flag	○	○	○	○	Off	-	-	R	NO	Off
M1022	Carry flag	○	○	○	○	Off	-	-	R	NO	Off
M1023	PLSY Y1 mode selection, continuous output when On	○	○	○	×	Off	-	-	R/W	NO	Off
M1024	Requesting COM1 monitoring	○	○	○	○	Off	-	-	R	NO	Off
M1025*	There is incorrect request for communication.	○	○	○	○	Off	-	-	R	NO	Off
M1026	Enabling RAMP module	×	○	○	○	Off	-	-	R/W	NO	Off
M1027	Number of PR outputs	×	○	×	○	Off	-	-	R/W	NO	Off
M1028	10ms time switching flag. Off: time base of T64 ~ T126 = 100ms	○	×	×	×	Off	-	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	On: time base of T64 ~ T126 = 10ms										
M1029*	ES/EC/EC3-8K: Y0 pulse output of PLSY, PLSR instructions is completed, or other relevant instructions complete their executions. EH3/SV2: the 1st group pulse output CH0 (Y0, Y1) is completed, or other relevant instructions complete their executions.	○	○	○	○	Off	-	-	R	NO	Off
M1030*	ES/EC/EC3-8K: Y1 pulse output of PLSY, PLSR instructions is completed, or other relevant instructions complete their executions. EH3/SV2: the 2nd group pulse output CH1 (Y2, Y3) is completed, or other relevant instructions complete their executions.	○	○	○	○	Off	-	-	R	NO	Off
M1031*	Clear all non-latched areas	○	○	○	○	Off	-	-	R/W	NO	Off
M1032*	Clear all latched areas	○	○	○	○	Off	-	-	R/W	NO	Off
M1033*	Memory latched when STOP	○	○	○	○	Off	-	-	R/W	NO	Off
M1034*	Disabling all Y outputs	○	○	○	○	Off	-	-	R/W	NO	Off
M1035*	Enabling input point X as the RUN/STOP switch, corresponding to D1035 (SX designates X3 only; EC3-8K designates X7 only)	×	○	○	○	-	-	-	R/W	YES	Off
M1036*	EH3/SV2: the 3rd group pulse output CH2 (Y4, Y5) is completed. (Not available in EH)	×	×	×	○	Off	-	-	R	NO	Off
M1037	EH3/SV2: the 4th group pulse output CH3 (Y6, Y7) is completed. (Not available in EH)	×	×	×	○	Off	-	-	R	NO	Off
M1038	Off: The time base of T0~T99 is 100ms. On: The time base of T0~T99 is 1ms.	×	×	×	○	Off	-	-	R/W	NO	Off
M1039*	Fixing time scan mode	○	○	○	○	Off	-	-	R/W	NO	Off
M1040	Disabling step	○	○	○	○	Off	-	-	R/W	NO	Off
M1041	Starting step	○	○	○	○	Off	-	Off	R/W	NO	Off
M1042	Enabling pulses	○	○	○	○	Off	-	-	R/W	NO	Off
M1043	Zero return completed	○	○	○	○	Off	-	Off	R/W	NO	Off
M1044	Zero point condition	○	○	○	○	Off	-	Off	R/W	NO	Off
M1045	Disabling all output reset	○	○	○	○	Off	-	-	R/W	NO	Off
M1046	Setting STL status as On	○	○	○	○	Off	-	-	R	NO	Off
M1047	Enabling STL monitoring	×	○	×	○	Off	-	-	R/W	NO	Off
M1048	Alarm status	×	○	×	○	Off	-	-	R	NO	Off
M1049	Setting up alarm monitoring	×	○	×	○	Off	-	-	R/W	NO	Off
M1050	Inhibiting I001	○	○	○	×	Off	-	-	R/W	NO	Off
M1051	Inhibiting I101	○	○	○	×	Off	-	-	R/W	NO	Off
M1052	Inhibiting I201	○	○	○	×	Off	-	-	R/W	NO	Off
M1053	Inhibiting I301	○	○	○	×	Off	-	-	R/W	NO	Off
M1054	Inhibiting I401	×	○	○	×	Off	-	-	R/W	NO	Off
M1055	Inhibiting I501	×	○	○	×	Off	-	-	R/W	NO	Off
M1056	Inhibiting I6□□	○	○	○	×	Off	-	-	R/W	NO	Off
	Enabling X1 interrupt to get the counting value of C241	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1057	Inhibiting I7□□	×	○	○	×	Off	-	-	R/W	NO	Off
	Enabling X2 interrupt to get the counting value of C241	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1058	COM3 monitoring request	×	×	×	○	Off	-	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1059	Inhibiting I010 ~ I060	×	○	×	×	Off	-	-	R/W	NO	Off
	Enabling X3 interrupt to get the counting value of C241	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1060	System error message 1: The peripheral circuit of the CPU breaks down.	○	○	○	×	Off	-	-	R	NO	Off
M1061	System error message 2: The CPU flag register breaks down.	×	○	○	×	Off	-	-	R	NO	Off
	System error message 2: An error occurs when the data in the latched area is read.	○	×	×	×	Off	-	-	R	NO	Off
M1062	System error message 3: The CPU BIOS ROM breaks down.	○	○	○	×	Off	-	-	R	NO	Off
M1063	System error message 4: The RAM in the CPU breaks down.	○	○	○	×	Off	-	-	R	NO	Off
M1064	Incorrect use of operands	○	○	○	○	Off	Off	-	R	NO	Off
M1065	Syntax error	○	○	○	○	Off	Off	-	R	NO	Off
M1066	Loop error	○	○	○	○	Off	Off	-	R	NO	Off
M1067*	Calculation error	○	○	○	○	Off	Off	-	R	NO	Off
M1068*	Calculation error locked (D1068)	○	○	○	○	Off	-	-	R	NO	Off
M1070	ES/EC3-8K/SX: Y1 time base switching for PWM instruction (On: 100us; Off: 1ms) EH3/SV2: Y0 time base switching for PWM instruction (On: 100us; Off: 1ms), when On, D1371 will decide the time base)	○	○	○	○	Off	-	-	R/W	NO	Off
M1071	EC3-8K: Y3 time base switching for PWM instruction (On: 100us; Off: 1ms) EH3/SV2: Y2 time base switching for PWM instruction, Off: 1ms, when On, D1372 will decide the time base	×	×	○	○	Off	-	-	R/W	NO	Off
M1072	Executing PLC RUN instruction	○	○	○	○	Off	On	Off	R/W	NO	Off
M1074	SRAM access error	×	×	○	○	Off	-	-	R	NO	Off
M1075	Error occurring when writing FLASH card or Flash ROM	×	×	○	○	Off	-	-	R	NO	Off
M1076*	Real time clock malfunction	×	○	×	○	Off	-	-	R	NO	Off
M1077	Battery in low voltage, malfunction or no battery	×	○	×	○	Off	-	-	R	NO	Off
M1078	Immediately stopping Y0 pulse output for PLSY instruction	○	○	○	×	Off	-	-	R/W	NO	Off
M1079	Immediately stopping Y1 pulse output for PLSY instruction	○	○	○	×	Off	-	-	R/W	NO	Off
M1080	Requesting COM2 monitoring	○	○	○	○	Off	-	-	R	NO	Off
M1081	Changing direction for FLT instruction	×	○	○	○	Off	-	-	R/W	NO	Off
M1082	Real time clock has been changed	×	○	×	○	Off	-	-	R	NO	Off
M1083	Allowing interruption subroutine in FROM/TO instructions (Not available in SX V3.0 and above)	×	○	×	○	Off	-	-	R/W	NO	Off
M1084*	Detecting bandwidth (only available in ES/EX/EC V6.4, SX V1.6 and EC3-8K)	○	○	○	×	Off	Off	Off	R/W	NO	Off
M1085	Selecting DVP-PCC01 duplicating function	○	○	○	○	Off	-	-	R/W	NO	Off
M1086	Setting up the switch for enabling password function of DVP-PCC01	○	○	○	○	Off	-	-	R/W	NO	Off
M1087*	Enabling LV signal	×	×	×	○	Off	-	-	R/W	NO	Off
M1088	Matrix comparison. Comparing between equivalent values (M1088 = 1) or different values (M1088 = 0).	×	○	×	○	Off	Off	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1089	Matrix search end flag. When the comparison reaches the last bit, M1089 = 1.	×	○	×	○	Off	Off	-	R	NO	Off
M1090	Matrix search start flag. Comparing from bit 0 (M1090 = 1).	×	○	×	○	Off	Off	-	R	NO	Off
M1091	Matrix bit search flag. When the comparison is completed, the comparison will stop immediately (M1091=1).	×	○	×	○	Off	Off	-	R	NO	Off
M1092	Matrix pointer error flag. When the pointer Pr exceeds its range, M1092 = 1.	×	○	×	○	Off	Off	-	R	NO	Off
M1093	Matrix pointer increasing flag. Adding 1 to the current value of the Pr.	×	○	×	○	Off	Off	-	R/W	NO	Off
M1094	Matrix pointer clear flag. Clearing the current value of the Pr to 0.	×	○	×	○	Off	Off	-	R/W	NO	Off
M1095	Matrix rotation/displacement/output carry flag	×	○	×	○	Off	Off	-	R	NO	Off
M1096	Matrix displacement/input complement flag	×	○	×	○	Off	Off	-	R/W	NO	Off
M1097	Matrix rotation/displacement direction flag	×	○	×	○	Off	Off	-	R/W	NO	Off
M1098	Matrix counting the number of bits which are “1” or “0”	×	○	×	○	Off	Off	-	R/W	NO	Off
M1099	On when the matrix counting result is “0”	×	○	×	○	Off	Off	-	R/W	NO	Off
M1100	SPD instruction sampling once	×	×	×	○	Off	-	-	R/W	NO	Off
M1101*	Whether to enable file registers	×	○	×	○	-	-	-	R/W	Yes	Off
M1102*	EC3-8K: output completion flag of Y2 output or CH1 (Y2/Y3)	×	×	○	×	Off	Off	-	R/W	NO	Off
M1103*	EC3-8K: output completion flag of Y3 output	×	×	○	×	Off	Off	-	R/W	NO	Off
M1104	EC3-8K: output immediately stop flag of Y2 output or CH1 (Y2/Y3)	×	×	○	×	Off	Off	-	R/W	NO	Off
M1105	EC3-8K: output immediately stop flag of Y3 output	×	×	○	×	Off	Off	-	R/W	NO	Off
M1108	EC3-8K: Y0 output or CH1 (Y2/Y3) output is decelerating to stop	×	×	○	×	Off	Off	-	R/W	NO	Off
M1109	EC3-8K: Y1 output is decelerating to stop	×	×	○	×	Off	Off	-	R/W	NO	Off
M1110	EC3-8K: Y2 output or CH1 (Y2/Y3) output is decelerating to stop	×	×	○	×	Off	Off	-	R/W	NO	Off
M1111	EC3-8K: Y3 output is decelerating to stop	×	×	○	×	Off	Off	-	R/W	NO	Off
M1112	EC3-8K: Y0 time base switching for PWM instruction	×	×	○	×	Off	-	-	R/W	NO	Off
M1113	EC3-8K: Y2 time base switching for PWM instruction	×	×	○	×	Off	-	-	R/W	NO	Off
M1119*	Using the instruction DDRVI/DDRVA to enable two target frequencies.	×	×	×	○	Off	-	-	R/W	NO	Off
M1120*	Retaining the communication setting of COM2 (RS-485), modifying D1120 will be invalid when M1120 is set.	○	○	○	○	Off	Off	-	R/W	NO	Off
M1121	Waiting for the sending of COM2 (RS-485) communication data	○	○	○	○	Off	On	-	R	NO	Off
M1122	COM2 (RS-485) sending request	○	○	○	○	Off	Off	-	R/W	NO	Off
M1123	Receiving through COM2 (RS-485) is completed	○	○	○	○	Off	Off	-	R/W	NO	Off
M1124	Waiting for receiving through COM2 (RS-485)	○	○	○	○	Off	Off	-	R	NO	Off
M1125	COM2 (RS-485) communication reset	○	○	○	○	Off	Off	-	R/W	NO	Off
M1126	Selecting COM2 (RS-485) STX/ETX user defined or system defined	○	○	○	○	Off	Off	-	R/W	NO	Off
M1127	Sending/receiving data of COM2 (RS-485) communication instruction is completed (RS	○	○	○	○	Off	Off	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	instruction not included)										
M1128	Sending COM2 (RS-485)/receiving COM2 (RS-485) indication	○	○	○	○	Off	Off	-	R	NO	Off
M1129	COM2 (RS-485) receiving time-out	○	○	○	○	Off	Off	-	R/W	NO	Off
M1130	Selecting COM2 (RS-485) STX/ETX user defined or system defined	○	○	○	○	Off	Off	-	R/W	NO	Off
M1131	On during COM2 (RS-485) MODRD/RDST/MODRW data are converted to hex data	○	○	○	○	Off	Off	-	R	NO	Off
M1132	On when there are no communication related instructions in the program	○	○	○	○	Off	-	-	R	NO	On
M1136*	Retaining the communication setting of COM3	×	×	×	○	Off	-	-	R/W	NO	Off
M1137	DNET mapping data are retained in STOP status.	×	×	×	○	Off	-	-	R/W	NO	Off
M1138*	Retaining the communication setting of COM1 (RS-232), modifying D1036 will be invalid when M1138 is set.	○	○	○	○	Off	-	-	R/W	NO	Off
M1139*	Selecting ASCII or RTU mode of COM1 (RS-232) when in Slave mode Off: ASCII; On: RTU	○	○	○	○	Off	-	-	R/W	NO	Off
M1140	MODRD/MODWR/MODRW data receiving error	○	○	○	○	Off	Off	-	R	NO	Off
M1141	MODRD/MODWR/MODRW parameter error	○	○	○	○	Off	Off	-	R	NO	Off
M1142	Data receiving of VFD-A commands error	○	○	○	○	Off	Off	-	R	NO	Off
M1143*	Selecting ASCII or RTU mode of COM2 (RS-485) when in Slave mode Off: ASCII; On: RTU Selecting ASCII or RTU mode of COM2 (RS-485) when in Master mode (used together with MODRD/ MODWR/MODRW instructions) Off: ASCII; On: RTU	○	○	○	○	Off	-	-	R/W	NO	Off
M1145*	ON: Read the MAC Address of the left-side module (should work with D1400~D1403)	×	×	×	V2.2	Off	Off	Off	R/W	NO	Off
M1148*	The delay unit for the instruction Delay is 5 us.	×	×	×	V1.62	Off	Off	Off	R/W	NO	Off
M1150	DHSZ instruction in multiple set values comparison mode	×	×	×	○	Off	-	-	R/W	NO	Off
M1151	The execution of DHSZ multiple set values comparison mode is completed.	×	×	×	○	Off	-	-	R	NO	Off
M1152	Setting up DHSZ instruction as frequency control mode	×	×	×	○	Off	-	-	R/W	NO	Off
M1153	DHSZ frequency control mode has been executed.	×	×	×	○	Off	-	-	R	NO	Off
M1154*	PWD bandwidth detection duty-off/duty-on	×	×	×	○	Off	Off	-	R/W	NO	Off
M1155	The instruction DCIMA or DCIMR enables the automatic acceleration/deceleration.	×	×	×	○	Off	-	-	R/W	NO	Off
M1156*	Enabling X0 interruption, decelerating immediately and stopping CH0 high-speed output (When M1156 is enabled and M1538 = On, clear M1156 to send the remaining output pulses.)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1157*	Enabling X1 interruption, decelerating immediately and stopping CH1 high-speed output	×	×	×	○	Off	Off	-	R/W	NO	Off
M1158*	Enabling X2 interruption, decelerating immediately and stopping CH2 high-speed output	×	×	×	○	Off	Off	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1159*	Enabling X3 interruption, decelerating immediately and stopping CH3 high-speed output	×	×	×	○	Off	Off	-	R/W	NO	Off
M1160	SX: X4, X5 bandwidth detection flag (Not available in SX V3.0 and above)	×	○	×	×	Off	Off	Off	R/W	NO	Off
M1161	8-bit mode On: in 8-bit mode	○	○	○	○	Off	-	-	R/W	NO	Off
M1162	Switching between decimal integer and binary floating point for SCLP instruction On: binary floating point; Off: decimal integer	○	○	○	○	Off	-	-	R/W	NO	Off
M1163*	Read/write memory card according to value in D1063 (automatically Off once the execution is completed)	×	×	×	○	Off	-	-	R/W	NO	Off
M1164*	Read/write internal Flash ROM according to value in D1064 (automatically Off once the execution is completed)	×	×	×	○	Off	-	-	R/W	NO	Off
M1167	16-bit mode for HKY input	×	○	×	○	Off	-	-	R/W	NO	Off
M1168	Designating work mode of SMOV	×	○	○	○	Off	-	-	R/W	NO	Off
M1169	Selecting PWD modes	×	×	×	○	Off	-	-	R/W	NO	Off
M1170*	Enabling single step execution	×	×	×	○	Off	-	-	R/W	NO	Off
M1171*	Single step execution	×	×	×	○	Off	-	-	R/W	NO	Off
M1172*	Switch for 2-phase pulse output On: switch on (Not available in SX V3.0 and above)	×	○	×	×	Off	Off	Off	R/W	NO	Off
M1173*	On: continuous output (Not available in SX V3.0 and above)	×	○	×	×	Off	-	-	R/W	NO	Off
M1174*	The number of output pulses reaches the target (Not available in SX V3.0 and above)	×	○	×	×	Off	Off	-	R/W	NO	Off
M1175*	Losing PLC parameter data (not available in EH2)	×	×	×	○	-	-	-	R	YES	Off
M1176*	Losing the data in PLC program (not available in EH2)	×	×	×	○	-	-	-	R	YES	Off
M1177	The instruction DABSR is used with a servo drive.	×	×	×	○	Off	-	-	R/W	NO	Off
M1178*	Enabling VR0 rotary switch	×	○	○	○	Off	-	-	R/W	NO	Off
M1179*	Enabling VR1 rotary switch	×	○	○	○	Off	-	-	R/W	NO	Off
M1181	Enabling X2 interruption (I201) followed by immediately clearing X0 high-speed counting input value. PS1: Only supports SX_V1.8 and versions above. PS2: After the high-speed counting value is obtained, the high-speed counting present value will be cleared immediately.	×	○	×	×	Off	Off	-	R/W	NO	Off
M1182*	Enabling X3 interruption (I301) followed by immediately clearing X1 high-speed counting input value. PS1: Only supports SX_V1.8 and versions above. PS2: After the high-speed counting value is obtained, the high-speed counting present value will be cleared immediately.	×	○	×	×	Off	Off	-	R/W	NO	Off
	The default value of M1182 is Off. When M1182 is On, the auto-mapping function is disabled.	×	×	×	○	On	-	-	R/W	NO	On

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	The analog-to-digital values/digital-to-analog values correspond to D9800~. If the first left-side module connected is a communication module, the analog-to-digital values / digital-to-analog values correspond to D9810~. For example, if the modules connected from left to right are 04DA-SL and EN01-SL and EH3-L / SV2, and M1182 is Off, D9810~D9813 will be assigned to CH1~CH4 in 04DA-SL.										
M1183*	On: The auto-mapping function of the special module is disabled. PS1: Mapping onto D9900~ PS2: The right side module should support this function.	×	×	×	○	#	-	-	R/W	NO	#
M1184*	Enabling modem function (not available in SV2)	×	×	×	○	Off	-	-	R/W	NO	Off
M1185*	Enabling initialization of modem (not available in SV2)	×	×	×	○	Off	-	-	R/W	NO	Off
M1186*	Initialization of modem fails (not available in SV2)	×	×	×	○	Off	-	-	R/W	NO	Off
M1187*	Initialization of modem is completed (not available in SV2)	×	×	×	○	Off	-	-	R/W	NO	Off
M1188*	Displaying whether modem is connecting currently (not available in SV)	×	×	×	○	Off	-	-	R/W	NO	Off
M1189	Read/write of Memory card/Flash ROM completed flag (Automatically reset to Off every time when enabled)	×	×	×	○	Off	-	-	R/W	NO	Off
M1190	Enabling PLSY for Y0 high-speed output of 0.01 ~ 100Hz	×	×	×	○	Off	Off	-	R/W	NO	Off
M1191	Enabling PLSY for Y2 high-speed output of 0.01 ~ 100Hz	×	×	×	○	Off	Off	-	R/W	NO	Off
M1192	Enabling PLSY for Y4 high-speed output of 0.01 ~ 100Hz	×	×	×	○	Off	Off	-	R/W	NO	Off
M1193	Enabling PLSY for Y6 high-speed output of 0.01 ~ 100Hz	×	×	×	○	Off	Off	-	R/W	NO	Off
M1194	I40X, I50X interruptions is able to immediately update the present pulse output value at CH0.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1195	I40X, I50X interruptions is able to immediately update the present pulse output value at CH1.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1196	Setting up the content type in the display (for SX) On: hex; Off: decimal	×	○	×	×	Off	-	-	R/W	NO	Off
M1197	Setting up the display of the 100ths digit (for SX)	×	○	×	×	Off	-	-	R/W	NO	Off
M1198	Setting up the display of the 10ths digit (for SX)	×	○	×	×	Off	-	-	R/W	NO	Off
M1200	Counting mode of C200 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1201	Counting mode of C201 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1202	Counting mode of C202 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1203	Counting mode of C203 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1204	Counting mode of C204 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1205	Counting mode of C205 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1206	Counting mode of C206 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1207	Counting mode of C207 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1208	Counting mode of C208 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1209	Counting mode of C209 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1210	Counting mode of C210 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1211	Counting mode of C211 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1212	Counting mode of C212 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1213	Counting mode of C213 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1214	Counting mode of C214 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1215	Counting mode of C215 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1216	Counting mode of C216 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1217	Counting mode of C217 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1218	Counting mode of C218 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1219	Counting mode of C219 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1220	Counting mode of C220 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1221	Counting mode of C221 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1222	Counting mode of C222 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1223	Counting mode of C223 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1224	Counting mode of C224 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1225	Counting mode of C225 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1226	Counting mode of C226 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1227	Counting mode of C227 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1228	Counting mode of C228 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1229	Counting mode of C229 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1230	Counting mode of C230 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1231	Counting mode of C231 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1232	Counting mode of C232 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1233	Counting mode of C233 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1234	Counting mode of C234 (On: counting down)	×	○	○	○	Off	-	-	R/W	NO	Off
M1235	Counting mode of C235 (On: counting down)	○	○	○	○	Off	-	-	R/W	NO	Off
M1236	Counting mode of C236 (On: counting down)	○	○	○	○	Off	-	-	R/W	NO	Off
M1237	Counting mode of C237 (On: counting down)	○	○	○	○	Off	-	-	R/W	NO	Off
M1238	Counting mode of C238 (On: counting down)	○	○	○	○	Off	-	-	R/W	NO	Off
M1239	Counting mode of C239 (On: counting down)	×	○	×	○	Off	-	-	R/W	NO	Off
M1240	Counting mode of C240 (On: counting down)	×	○	×	○	Off	-	-	R/W	NO	Off
M1241	Counting mode of C241 (On: counting down)	○	○	○	○	Off	-	-	R/W	NO	Off
M1242	Counting mode of C242 (On: counting down)	○	○	○	○	Off	-	-	R/W	NO	Off
M1243	Counting mode of C243 (On: counting down) (Not available in SX V3.0 and above)	×	○	×	○	Off	-	-	R/W	NO	Off
M1244	Counting mode of C244 (On: counting down)	○	○	○	○	Off	-	-	R/W	NO	Off
M1245	Counting mode of C245 (On: counting down) (Not available in SX V3.0 and above)	×	○	×	×	Off	-	-	R/W	NO	Off
M1246	C246 counter monitoring (On: counting down)	○	○	○	○	Off	-	-	R	NO	Off
M1247	C247 counter monitoring (On: counting down)	○	○	○	○	Off	-	-	R	NO	Off
M1248	C248 counter monitoring (On: counting down)	×	×	×	○	Off	-	-	R	NO	Off
M1249	C249 counter monitoring (On: counting down)	○	○	○	○	Off	-	-	R	NO	Off
M1250	C250 counter monitoring (On: counting down)	×	○	×	×	Off	-	-	R	NO	Off
M1251	C251 counter monitoring (On: counting down)	○	○	○	○	Off	-	-	R	NO	Off
M1252	C252 counter monitoring (On: counting down)	○	○	○	○	Off	-	-	R	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1253	C253 counter monitoring (On: counting down)	×	×	×	○	Off	-	-	R	NO	Off
M1254	C254 counter monitoring (On: counting down)	○	○	○	○	Off	-	-	R	NO	Off
M1255	C255 counter monitoring (On: counting down)	×	○	×	×	Off	-	-	R	NO	Off
M1257	The acceleration/deceleration slope of the high-speed pulse output is an S curve.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1258	Y0 pulse output signal reversing for PWM instruction	×	×	×	○	Off	-	-	R/W	NO	Off
M1259	Y2 pulse output signal reversing for PWM instruction	×	×	×	○	Off	-	-	R/W	NO	Off
M1260	X5 as the reset input signal for all high-speed counters	×	○	×	×	Off	-	-	R/W	NO	Off
M1261	High-speed comparator comparison flag for DHSCR instruction	×	×	×	○	Off	-	-	R/W	NO	Off
M1262	Enabling the instruction DPTPO to output the circulatory pulse output.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1264	Enabling reset function of HHSC0	×	×	×	○	Off	-	-	R/W	NO	Off
M1265	Enabling start function of HHSC0	×	×	×	○	Off	-	-	R/W	NO	Off
M1266	Enabling reset function of HHSC1	×	×	×	○	Off	-	-	R/W	NO	Off
M1267	Enabling start function of HHSC1	×	×	×	○	Off	-	-	R/W	NO	Off
M1268	Enabling reset function of HHSC2	×	×	×	○	Off	-	-	R/W	NO	Off
M1269	Enabling start function of HHSC2	×	×	×	○	Off	-	-	R/W	NO	Off
M1270	Enabling reset function of HHSC3	×	×	×	○	Off	-	-	R/W	NO	Off
M1271	Enabling start function of HHSC3	×	×	×	○	Off	-	-	R/W	NO	Off
M1272	Reset control of HHSC0	×	×	×	○	Off	-	-	R/W	NO	Off
M1273	Start control of HHSC0	×	×	×	○	Off	-	-	R/W	NO	Off
M1274	Reset control of HHSC1	×	×	×	○	Off	-	-	R/W	NO	Off
M1275	Start control of HHSC1	×	×	×	○	Off	-	-	R/W	NO	Off
M1276	Reset control of HHSC2	×	×	×	○	Off	-	-	R/W	NO	Off
M1277	Start control of HHSC2	×	×	×	○	Off	-	-	R/W	NO	Off
M1278	Reset control of HHSC3	×	×	×	○	Off	-	-	R/W	NO	Off
M1279	Start control of HHSC3	×	×	×	○	Off	-	-	R/W	NO	Off
M1280	EH3/SV2: Inhibiting I00□ EC3-8K: I001 forced to go toward the opposite directon when rising/falling edge triggered	×	×	○	○	Off	-	-	R/W	NO	Off
M1281	Inhibiting I10□	×	×	×	○	Off	-	-	R/W	NO	Off
M1282	Inhibiting I20□	×	×	×	○	Off	-	-	R/W	NO	Off
M1283	Inhibiting I30□	×	×	×	○	Off	-	-	R/W	NO	Off
M1284	EH3/SV2: Inhibiting I40□ EC3-8K: I401 forced to go toward the opposite directon when rising/falling edge triggered	×	×	×	○	Off	-	-	R/W	NO	Off
M1285	Inhibiting I50□	×	×	×	○	Off	-	-	R/W	NO	Off
M1286	EH3/SV2: Inhibiting I6□□ EC3-8K: I601 forced to go toward the opposite directon when rising/falling edge triggered	×	×	○	○	Off	-	-	R/W	NO	Off
M1287	Inhibiting I7□□	×	×	○	○	Off	-	-	R/W	NO	Off
M1288	Inhibiting I8□□	×	×	×	○	Off	-	-	R/W	NO	Off
M1289	Inhibiting I010	×	×	×	○	Off	-	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3-8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1290	Inhibiting I020	×	×	×	○	Off	-	-	R/W	NO	Off
M1291	Inhibiting I030	×	×	×	○	Off	-	-	R/W	NO	Off
M1292	Inhibiting I040	×	×	×	○	Off	-	-	R/W	NO	Off
M1293	Inhibiting I050	×	×	×	○	Off	-	-	R/W	NO	Off
M1294	Inhibiting I060	×	×	×	○	Off	-	-	R/W	NO	Off
M1295	Inhibiting I110	×	×	×	○	Off	-	-	R/W	NO	Off
M1296	Inhibiting I120	×	×	×	○	Off	-	-	R/W	NO	Off
M1297	Inhibiting I130	×	×	×	○	Off	-	-	R/W	NO	Off
M1298	Inhibiting I140	×	×	×	○	Off	-	-	R/W	NO	Off
M1299	Inhibiting I150 (Not available in SX series PLCs)	×	○	○	○	Off	-	-	R/W	NO	Off
M1300	Inhibiting I160	×	×	×	○	Off	-	-	R/W	NO	Off
M1301	Inhibiting I170	×	×	×	○	Off	-	-	R/W	NO	Off
M1302	Inhibiting I180	×	×	×	○	Off	-	-	R/W	NO	Off
M1303	High/low bits exchange for XCH instruction	×	○	○	○	Off	-	-	R/W	NO	Off
M1304*	Enabling set On/Off of CPU input point X	○	○	○	○	Off	-	-	R/W	NO	Off
M1305	Reverse output direction of the 1st group pulse CH0 (Y0, Y1) for PLSV, DPLSV, DRVI, DDRVI, DRVA, DDRVA instructions	×	×	○	○	Off	-	-	R	NO	Off
M1306	Reverse output direction of the 2nd group pulse CH1 (Y2, Y3) for PLSV, DPLSV, DRVI, DDRVI, DRVA, DDRVA instructions	×	×	○	○	Off	-	-	R	NO	Off
M1308	Off->On: The 1st pulse group CH0 (Y0, Y1) high-speed output immediately stops. On->Off: Completing remaining number of output pulses	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1309	Off->On: The 1st pulse group CH1 (Y2, Y3) high-speed output immediately stops. On->Off: Completing remaining number of output pulses	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1310*	Disabling Y10 pulse output (for SC V1.4 and above) (Not available in SX V3.0 and above)	×	○	×	×	Off	Off	-	R/W	NO	Off
	Off->On: The 1st pulse group CH2 (Y4, Y5) high-speed output immediately stops. On->Off: Completing remaining numbe of output pulses	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1311*	Disabling Y11 pulse output (for SC V1.4 and above) (Not available in SX V3.0 and above)	×	○	×	×	Off	Off	-	R/W	NO	Off
	Off->On: The 1st pulse group CH3 (Y6, Y7) high-speed output immediately stops. On->Off: Completing remaining number of output pulses	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1312	Sending request of COM1 (RS-232) communication instruction (only available in the instructions MODRW and RS)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1313	Waiting to receive the data of COM1 (RS-232) communication instruction is completed (only available in the instructions MODRW and RS)	×	×	×	○	Off	Off	-	R	NO	Off
M1314	Receiving the data of COM1 (RS-232) communication instruction is completed (only	×	×	×	○	Off	Off	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	available in the instructions MODRW and RS)										
M1315	An error occurs when receiving the data of COM1 (RS-232) communication instruction (only available in the instructions MODRW and RS)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1316	Sending request of COM3 (RS-485) communication instruction (only available in the instructions MODRW and RS)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1317	Waiting to receive the data of COM3 (RS-485) communication instruction is completed (only available in the instructions MODRW and RS)	×	×	×	○	Off	Off	-	R	NO	Off
M1318	Receiving data of COM3 (RS-485) communication instruction is completed (only available in the instructions MODRW and RS)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1319	An error occurs when receiving the data of COM3 (RS-485) communication instruction (only available in the instructions MODRW and RS)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1320	EH3: Slave mode: COM3 (RS-485) is in the ASCII/RTU mode. (Off: ASCII mode; On: RTU mode) Master mode: COM3 (RS-485) is in the ASCII/RTU mode. (Off: Off: ASCII mode; On: RTU mode) M1320 is used with the instruction MODRW/FWD.	×	×	×	○	Off	-	-	R/W	NO	Off
M1326	24SV2: output completion flag of pulse output CH4 (Y10/Y11)	×	×	×	24SV2	Off	-	-	R/W	NO	Off
M1327	24SV2: output completion flag of pulse output CH5 (Y12/Y13)	×	×	×	24SV2	Off	-	-	R/W	NO	Off
M1334*	EH3/SV2: stopping the 1st group pulse output CH0 (Y0, Y1)	×	×	×	○	Off	-	-	R/W	NO	Off
M1335*	EH2/SV/EH3/SV2: stopping the 2nd group pulse output CH1 (Y2, Y3)	×	×	×	○	Off	-	-	R/W	NO	Off
M1336	Sending out the 1st group pulse output CH0 (Y0, Y1)	×	×	×	○	Off	Off	Off	R	NO	Off
M1337	Sending out the 2nd group pulse output CH1 (Y2, Y3)	×	×	×	○	Off	Off	Off	R	NO	Off
M1338	Enabling offset pulses of the 1st group pulse output CH0 (Y0, Y1)	×	×	×	○	Off	-	-	R/W	NO	Off
M1339	Enabling offset pulses of the 2nd group pulse output CH1 (Y2, Y3)	×	×	×	○	Off	-	-	R/W	NO	Off
M1340	Generating interruption I110 after the 1st group pulse output CH0 (Y0, Y1) is sent out	×	×	×	○	Off	-	-	R/W	NO	Off
M1341	Generating interruption I120 after the 2nd group pulse output CH1 (Y2, Y3) is sent out	×	×	×	○	Off	-	-	R/W	NO	Off
M1342	Generating interruption I130 when the 1st group pulse output CH0 (Y0, Y1) is sent out	×	×	×	○	Off	-	-	R/W	NO	Off
M1343	Generating interruption I140 when the 2nd group pulse output CH1 (Y2, Y3) is sent out	×	×	×	○	Off	-	-	R/W	NO	Off
M1344	Enabling the offset of the 1st group pulse output CH0 (Y0, Y1)	×	×	×	○	Off	-	-	R/W	NO	Off
M1345	Enabling the offset of the 2nd group pulse output CH1 (Y2, Y3)	×	×	×	○	Off	-	-	R/W	NO	Off
M1346	Enabling ZRN CLEAR output signal	×	×	×	○	Off	-	-	R/W	NO	Off
M1347	EH3/SV2: Automatic zero return after the 1st group pulse output CH0 (Y0, Y1) is completed.	×	○	○	○	Off	-	-	R/W	NO	Off

Special M	Function		ES EX EC	SX	EC3-8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	SX/EC3-8K: Automatic zero return after Y0 high-speed pulse output is completed.											
M1348	EH3/SV2: Automatic zero return after the 2nd group pulse output CH1 (Y2, Y3) is completed. SX/EC3-8K: Automatic zero return after Y1 high-speed pulse output is completed.		×	○	○	○	Off	-	-	R/W	NO	Off
M1350*	Enabling PLC LINK		×	○	×	○	Off	-	Off	R/W	NO	Off
M1351*	Enabling auto mode on PLC LINK		×	○	×	○	Off	-	-	R/W	NO	Off
M1352*	Enabling manual mode on PLC LINK		×	○	×	○	Off	-	-	R/W	NO	Off
M1353*	Enable 32 slave unit linkage and up to 100 data length of data exchange on PLC LINK	EH3 V1.2/SV2 V1.0	×	×	×	○	-	-	-	R/W	YES	Off
		Others					Off	-	-	R/W	NO	Off
M1354*	Enable simultaneous data read/write in a polling of PLC LINK	EH3 V1.2/SV2 V1.0	×	×	×	○	-	-	-	R/W	YES	Off
		Others					Off	-	-	R/W	NO	Off
M1355	When M1355 = Off, enable PLC LINK for slave connection detection. M1360 ~ M1375 (M1440 ~ M1455) will be the slave connection detection. Then this falg can only be read. When M1355 = On, M1360 ~ M1375 (M1440 ~ M1455) will be the flag designating connection, not for slave connection detection. Then this flag can be read and written.		×	○	×	○	Off	-	-	R/W	YES	Off
M1356	When the PLC link is enabled and M1356 is ON, the values in D1900~D1931 are taken as the station address. The default station address in D1399 is not used.		×	×	×	○	-	-	-	R/W	YES	Off
M1360*	Slave ID#1 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1361*	Slave ID#2 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1362*	Slave ID#3 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1363*	Slave ID#4 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1364*	Slave ID#5 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1365*	Slave ID#6 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1366*	Slave ID#7 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1367*	Slave ID#8 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1368*	Slave ID#9 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1369*	Slave ID#10 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1370*	Slave ID#11 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1371*	Slave ID#12 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1372*	Slave ID#13 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1373*	Slave ID#14 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1374*	Slave ID#15 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1375*	Slave ID#16 status on PLC LINK network		×	○	×	○	Off	-	-	R/W	YES	Off
M1376*	Indicating Slave ID#1 data exchange status on PLC LINK		×	○	×	○	Off	-	-	R	NO	Off
M1377*	Indicating Slave ID#2 data exchange status on PLC LINK		×	○	×	○	Off	-	-	R	NO	Off
M1378*	Indicating Slave ID#3 data exchange status on		×	○	×	○	Off	-	-	R	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	PLC LINK										
M1379*	Indicating Slave ID#4 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1380*	Indicating Slave ID#5 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1381*	Indicating Slave ID#6 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1382*	Indicating Slave ID#7 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1383*	Indicating Slave ID#8 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1384*	Indicating Slave ID#9 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1385*	Indicating Slave ID#10 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1386*	Indicating Slave ID#11 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1387*	Indicating Slave ID#12 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1388*	Indicating Slave ID#13 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1389*	Indicating Slave ID#14 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1390*	Indicating Slave ID#15 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1391*	Indicating Slave ID#16 data exchange status on PLC LINK	×	○	×	○	Off	-	-	R	NO	Off
M1392*	Slave ID#1 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1393*	Slave ID#2 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1394*	Slave ID#3 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1395*	Slave ID#4 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1396*	Slave ID#5 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1397*	Slave ID#6 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1398*	Slave ID#7 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1399*	Slave ID#8 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1400*	Slave ID#9 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1401*	Slave ID#10 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1402*	Slave ID#11 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1403*	Slave ID#12 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1404*	Slave ID#13 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1405*	Slave ID#14 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1406*	Slave ID#15 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1407*	Slave ID#16 linking error	×	○	×	○	Off	-	-	R	NO	Off
M1408*	Indicating reading from Salve ID#1 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1409*	Indicating reading from Salve ID#2 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1410*	Indicating reading from Salve ID#3 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1411*	Indicating reading from Salve ID#4 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1412*	Indicating reading from Salve ID#5 is completed	×	○	×	○	Off	-	-	R	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1413*	Indicating reading from Salve ID#6 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1414*	Indicating reading from Salve ID#7 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1415*	Indicating reading from Salve ID#8 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1416*	Indicating reading from Salve ID#9 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1417*	Indicating reading from Salve ID#10 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1418*	Indicating reading from Salve ID#11 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1419*	Indicating reading from Salve ID#12 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1420*	Indicating reading from Salve ID#13 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1421*	Indicating reading from Salve ID#14 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1422*	Indicating reading from Salve ID#15 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1423*	Indicating reading from Salve ID#16 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1424*	Indicating writing to Salve ID#1 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1425*	Indicating writing to Salve ID#2 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1426*	Indicating writing to Salve ID#3 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1427*	Indicating writing to Salve ID#4 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1428*	Indicating writing to Salve ID#5 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1429*	Indicating writing to Salve ID#6 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1430*	Indicating writing to Salve ID#7 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1431*	Indicating writing to Salve ID#8 is completed	×	○	×	○	Off	-	-	R	NO	Off
M1432*	Indicating writing to Salve ID#9 is completed	×	○	○	○	Off	-	-	R	NO	Off
M1433*	Indicating writing to Salve ID#10 is completed	×	○	○	○	Off	-	-	R	NO	Off
M1434*	Indicating writing to Salve ID#11 is completed	×	○	○	○	Off	-	-	R	NO	Off
M1435*	Indicating writing to Salve ID#12 is completed	×	○	○	○	Off	-	-	R	NO	Off
M1436*	Indicating writing to Salve ID#13 is completed	×	○	○	○	Off	-	-	R	NO	Off
M1437*	Indicating writing to Salve ID#14 is completed	×	○	○	○	Off	-	-	R	NO	Off
M1438*	Indicating writing to Salve ID#15 is completed	×	○	○	○	Off	-	-	R	NO	Off
M1439*	Indicating writing to Salve ID#16 is completed	×	○	○	○	Off	-	-	R	NO	Off
M1440*	Slave ID#17 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1441*	Slave ID#18 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1442*	Slave ID#19 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1443*	Slave ID#20 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1444*	Slave ID#21 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1445*	Slave ID#22 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1446*	Slave ID#23 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1447*	Slave ID#24 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1448*	Slave ID#25 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1449*	Slave ID#26 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1450*	Slave ID#27 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off

Special M	Function	ES EX EC	SX	EC3-8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1451*	Slave ID#28 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1452	Slave ID#29 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1453*	Slave ID#30 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1454*	Slave ID#31 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1455*	Slave ID#32 status on PLC LINK network	×	×	×	○	-	-	-	R/W	YES	Off
M1456*	Indicating Slave ID#17 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1457*	Indicating Slave ID#18 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1458*	Indicating Slave ID#19 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1459*	Indicating Slave ID#20 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1460*	Indicating Slave ID#21 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1461*	Indicating Slave ID#22 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1462*	Indicating Slave ID#23 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1463*	Indicating Slave ID#24 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1464*	Indicating Slave ID#25 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1465*	Indicating Slave ID#26 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1466*	Indicating Slave ID#27 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1467*	Indicating Slave ID#28 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1468*	Indicating Slave ID#29 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1469*	Indicating Slave ID#30 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1470*	Indicating Slave ID#31 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1471*	Indicating Slave ID#32 data exchange status on PLC LINK	×	×	×	○	Off	-	-	R	NO	Off
M1472*	Slave ID#17 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1473*	Slave ID#18 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1474*	Slave ID#19 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1475*	Slave ID#20 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1476*	Slave ID#21 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1477*	Slave ID#22 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1478*	Slave ID#23 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1479*	Slave ID#24 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1480*	Slave ID#25 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1481*	Slave ID#26 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1482*	Slave ID#27 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1483*	Slave ID#28 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1484*	Slave ID#29 linking error	×	×	×	○	Off	-	-	R	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1485*	Slave ID#30 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1486*	Slave ID#31 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1487*	Slave ID#32 linking error	×	×	×	○	Off	-	-	R	NO	Off
M1488*	Indicating reading from Salve ID#17 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1489*	Indicating reading from Salve ID#18 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1490*	Indicating reading from Salve ID#19 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1491*	Indicating reading from Salve ID#20 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1492*	Indicating reading from Salve ID#21 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1493*	Indicating reading from Salve ID#22 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1494*	Indicating reading from Salve ID#23 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1495*	Indicating reading from Salve ID#24 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1496*	Indicating reading from Salve ID#25 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1497*	Indicating reading from Salve ID#26 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1498*	Indicating reading from Salve ID#27 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1499*	Indicating reading from Salve ID#28 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1500*	Indicating reading from Salve ID#29 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1501*	Indicating reading from Salve ID#30 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1502*	Indicating reading from Salve ID#31 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1503*	Indicating reading from Salve ID#32 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1504*	Indicating writing to Salve ID#17 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1505*	Indicating writing to Salve ID#18 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1506*	Indicating writing to Salve ID#19 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1507*	Indicating writing to Salve ID#20 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1508*	Indicating writing to Salve ID#21 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1509*	Indicating writing to Salve ID#22 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1510*	Indicating writing to Salve ID#23 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1511*	Indicating writing to Salve ID#24 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1512*	Indicating writing to Salve ID#25 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1513*	Indicating writing to Salve ID#26 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1514*	Indicating writing to Salve ID#27 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1515*	Indicating writing to Salve ID#28 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1516*	Indicating writing to Salve ID#29 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1517*	Indicating writing to Salve ID#30 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1518*	Indicating writing to Salve ID#31 is completed	×	×	×	○	Off	-	-	R	NO	Off

Special M	Function	ES EX EC	SX	EC3-8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1519*	Indicating writing to Salve ID#32 is completed	×	×	×	○	Off	-	-	R	NO	Off
M1520	Stopping the 3rd group pulse output CH2 (Y4, Y5)	×	×	×	○	Off	-	-	R/W	NO	Off
M1521	Stopping the 4th group pulse output CH3 (Y6, Y7)	×	×	×	○	Off	-	-	R/W	NO	Off
M1522	Sending the indication flag of the 3rd group pulse output CH2 (Y4, Y5)	×	×	×	○	Off	-	Off	R	NO	Off
M1523	Sending the indication flag of the 4th group pulse output CH3 (Y6, Y7)	×	×	×	○	Off	-	Off	R	NO	Off
M1524	EH3/SV2: Automatic zero return after the 3rd group pulse output CH2 (Y4, Y5) is completed. EC3-8K: Automatic zero return after Y2 or CH1 (Y2, Y3) high-speed pulse output is completed.	×	×	○	○	Off	-	-	R/W	NO	Off
M1525	EH3/SV2: Automatic zero return after the 4th group pulse output CH3 (Y6, Y7) is completed. EC3-8K: Automatic zero return after Y3 high-speed pulse output is completed.	×	×	○	○	Off	-	-	R/W	NO	Off
M1526	Reversing Y4 pulse output signal for PWM instruction	×	×	×	○	Off	-	-	R/W	NO	Off
M1527	Reversing Y6 pulse output signal for PWM instruction	×	×	×	○	Off	-	-	R/W	NO	Off
M1528*	Enabling the instruction DICF to execute the constant speed output section	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1529*	Enabling the instruction DICF to execute the final output section	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1530	Switching time base unit of Y4 output for PWM instruction On: 100us; Off: 1ms (Not available in EH)	×	×	×	○	Off	-	-	R/W	NO	Off
M1531	Switching time base unit of Y6 output for PWM instruction On: 100us; Off: 1ms (Not available in EH)	×	×	×	○	Off	-	-	R/W	NO	Off
M1532	Reverse operation of the 3rd group pulse CH2 (Y4, Y5) for PLSV/DPLSV/DRVI/DDRVI/DRVA /DDRVA instruction	×	×	×	○	Off	-	-	R/W	NO	Off
M1533	Reverse operation of the 4th group pulse CH3 (Y6, Y7) for PLSV/DPLSV/DRVI/DDRVI/DRVA /DDRVA instruction	×	×	×	○	Off	-	-	R/W	NO	Off
M1534	EH3/SV2: CH0 being able to designate deceleration time. Has to be used with D1348. EC3-8K: CH0 (Y0/Y1) being able to designate deceleration time.	×	×	○	○	Off	-	-	R/W	NO	Off
M1535	EH3/SV2: CH1 being able to designate deceleration time. Has to be used with D1349. EC3-8K: CH1 (Y2/Y3) being able to designate deceleration time.	×	×	○	○	Off	-	-	R/W	NO	Off
M1536	CH2 being able to designate deceleration time. Has to be used with D1350.	×	×	×	○	Off	-	-	R/W	NO	Off
M1537	CH3 being able to designate deceleration time. Has to be used with D1351.	×	×	×	○	Off	-	-	R/W	NO	Off
M1538*	EH3/SV2: Displaying CH0 high-speed output paused flag EC3-8K: Displaying Y0 high-speed output paused flag	×	×	○	○	Off	Off	-	R/W	NO	Off
M1539*	EH3/SV2: Displaying CH1 high-speed output paused flag EC3-8K: Displaying Y1 high-speed output paused flag	×	×	○	○	Off	Off	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3-8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
M1540*	EH3/SV2: Displaying CH2 high-speed output paused flag EC3-8K: Displaying Y2 high-speed output paused flag	×	×	○	○	Off	Off	-	R/W	NO	Off
M1541*	EH3/SV2: Displaying CH3 high-speed output paused flag EC3-8K: Displaying Y3 high-speed output paused flag	×	×	○	○	Off	Off	-	R/W	NO	Off
M1542	CH0 executes the function that the constant speed output section reaches the target frequency.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1543	CH0 executed the function that the constant speed output section reaches the target number.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1544	CH1 executes the function that the constant speed output section reaches the target frequency.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1545	CH1 executed the function that the constant speed output section reaches the target number.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1546	CH2 executes the function that the constant speed output section reaches the target frequency.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1547	CH2 executed the function that the constant speed output section reaches the target number.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1548	CH3 executes the function that the constant speed output section reaches the target frequency.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1549	CH3 executed the function that the constant speed output section reaches the target number.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1550	Used with the instruction DCIF to clear the high-speed output counting number	×	×	×	○	Off	Off	-	R/W	NO	Off
M1560	Inhibiting I900 and I901	×	×	×	○	Off	Off	-	R/W	NO	Off
M1561	Inhibiting I910 and I911	×	×	×	○	Off	Off	-	R/W	NO	Off
M1562	Inhibiting I920 and I921	×	×	×	○	Off	Off	-	R/W	NO	Off
M1563	Inhibiting I930 and I931	×	×	×	○	Off	Off	-	R/W	NO	Off
M1564	Inhibiting I940 and I941	×	×	×	○	Off	Off	-	R/W	NO	Off
M1565	Inhibiting I950 and I951	×	×	×	○	Off	Off	-	R/W	NO	Off
M1566	Inhibiting I960 and I961	×	×	×	○	Off	Off	-	R/W	NO	Off
M1567	Inhibiting I970 and I971	×	×	×	○	Off	Off	-	R/W	NO	Off
M1568	Set up a flag for a specific deceleration time for CH4 (should work with D1196)	×	×	×	24SV2	Off	-	-	R/W	NO	Off
M1569	Set up a flag for a specific deceleration time for CH5 (should work with D1196)	×	×	×	24SV2	Off	-	-	R/W	NO	Off
M1570	Enabling the negative limit function of the high-speed output CH0	×	×	×	○	Off	Off	-	R/W	NO	Off
M1571	Enabling the negative limit function of the high-speed output CH1	×	×	×	○	Off	Off	-	R/W	NO	Off
M1572	Enabling the negative limit function of the high-speed output CH2	×	×	×	○	Off	Off	-	R/W	NO	Off
M1573	Enabling the negative limit function of the high-speed output CH3	×	×	×	○	Off	Off	-	R/W	NO	Off
M1574	The DOG of CH0 in the instruction ZRN is positive stop function.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1575	The DOG of CH1 in the instruction ZRN is	×	×	×	○	Off	Off	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	positive stop function.										
M1576	The DOG of CH2 in the instruction ZRN is positive stop function.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1577	The DOG of CH3 in the instruction ZRN is positive stop function.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1578	Off: Number of times the instruction ZRN search for the Z phase On: The output designates the displacement. The flag is used with D1312.	×	×	×	○	Off	Off	-	R/W	NO	Off
M1580	The absolute position of Delta ASDA-A2 servo is read successfully by means of the instruction DABSR.	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1581	The absolute position of Delta ASDA-A2 servo is not read successfully by means of the instruction DABSR.	×	×	×	○	Off	Off	Off	R/W	NO	Off
M1584	If the left limit switch of CH0 is enabled by means of the instruction ZRN, it can be triggered either by a rising-edge signal or by a falling-edge signal. (OFF: Rising-edge signal; ON: Falling-edge signal)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1585	If the left limit switch of CH1 is enabled by means of the instruction ZRN, it can be triggered either by a rising-edge signal or by a falling-edge signal. (OFF: Rising-edge signal; ON: Falling-edge signal)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1586	If the left limit switch of CH2 is enabled by means of the instruction ZRN, it can be triggered either by a rising-edge signal or by a falling-edge signal. (OFF: Rising-edge signal; ON: Falling-edge signal)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1587	If the left limit switch of CH3 is enabled by means of the instruction ZRN, it can be triggered either by a rising-edge signal or by a falling-edge signal. (OFF: Rising-edge signal; ON: Falling-edge signal)	×	×	×	○	Off	Off	-	R/W	NO	Off
M1588*	Conversion value in analog input AD0 for 24SV2 PLC exceeding the maximum value, 4047 (voltage) / 2023 (current)	×	×	×	24SV2	Off	Off	-	R	NO	Off
M1589*	Conversion value in analog input AD1 for 24SV2 PLC exceeding the maximum value, 4047 (voltage) / 2023 (current)	×	×	×	24SV2	Off	Off	-	R	NO	Off
M1590	The speed at which data is exchanged by means of Ethernet increases. (ON: Enabled; OFF: Disabled)	×	×	×	V1.62	Off	Off	-	R/W	NO	Off
M1592	Reverse high-speed output direction of CH0 (Y1) (ON: output in the positive direction; OFF: output in the positivie direction direcition after the reserved direction flag is ON)	╳	╳	╳	V1.88	Off	Off	-	R/W	NO	Off
M1593	Reverse high-speed output direction of CH1 (Y3)	╳	╳	╳	V1.88	Off	Off	-	R/W	NO	Off
M1594	Reverse high-speed output direction of CH2 (Y5)	╳	╳	╳	V1.88	Off	Off	-	R/W	NO	Off
M1595	Reverse high-speed output direction of CH3 (Y7)	╳	╳	╳	V1.88	Off	Off	-	R/W	NO	Off
M1596	Reverse high-speed output direction of CH4 (Y11) (for 24SV2 PLC)	╳	╳	╳	V1.88	Off	Off	-	R/W	NO	Off
M1597	Reverse high-speed output direction of CH5	╳	╳	╳	V1.88	Off	Off	-	R/W	NO	Off

Special M	Function	ES EX EC	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	(Y13) (for 24SV2 PLC)										
M1604*	Fixed slope ramp-up/down for high-speed output CH0 (Y0/Y1)	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1605*	Fixed slope ramp-up/down for high-speed output CH1 (Y2/Y3)	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1606*	Fixed slope ramp-up/down for high-speed output CH2 (Y4/Y5)	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1607*	Fixed slope ramp-up/down for high-speed output CH3 (Y6/Y7)	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1608	Enable high-speed output for CH1(Y2/Y3) and to catch speed with the input group (X0/X1)	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1609	Enable MPG mode (refer to CSFO instruction for more details)	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1610*	Activate the masked area when the mark aligment function is enabled during high-speed output of CH0(Y0/Y1).	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1611*	Activate the masked area when the mark aligment function is enabled during high-speed output of CH1(Y2/Y3).	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1612*	Activate the masked area when the mark aligment function is enabled during high-speed output of CH2(Y4/Y5).	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1613*	Activate the masked area when the mark aligment function is enabled during high-speed output of CH3(Y6/Y7).	×	×	×	V1.88	Off	Off	-	R/W	NO	Off
M1614	Executing output from Ch4(Y10/Y11) (for 24SV2 PLC)	×	×	×	24SV2	Off	Off	-	R/W	NO	Off
M1615	Executing output from Ch5(Y12/Y13) (for 24SV2 PLC)	×	×	×	24SV2	Off	Off	-	R/W	NO	Off
M1620	Selecting protocols of V2.0A (ON) or V2.0B (OFF) for CANRS instruction	×	×	×	V2.06	Off	Off	-	R/W	NO	Off
M1621	Selecting Master/Slave mode for CANRS instruction	×	×	×	V2.06	Off	Off	-	R/W	NO	Off
M1622	Selecting Master/Slave mode for CANRS instruction	×	×	×	V2.06	Off	Off	-	R/W	NO	Off
M1623	CANRS instruciotn communication error	×	×	×	V2.06	Off	Off	-	R/W	NO	Off
M1630*	Activate the filtering function for a built-in single input point (X0~X17; one point at a time)	×	×	×	V2.06	Off	Off	-	R/W	NO	Off
M1631*	Change the filtering time for a built-in single input point (X0~X17; one point at a time)	×	×	×	V2.06	Off	Off	-	R/W	NO	Off
M1640	Refresh the value in special device for the current output CH0 (Y0/Y1) (REF instruction)	×	×	×	V2.06	Off	Off	-	R/W	NO	Off
M1641	Refresh the value in special device for the current output CH1 (Y2/Y3) (REF instruction)	×	×	×	V2.06	Off	Off	-	R/W	NO	Off
M1642	Refresh the value in special device for the current output CH2 (Y4/Y5) (REF instruction)	×	×	×	V2.06	Off	Off	-	R/W	NO	Off
M1643	Refresh the value in special device for the current output CH3 (Y6/Y7) (REF instruction)	×	×	×	V2.06	Off	Off	-	R/W	NO	Off

Special D	Function		ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1000*	Scanning watchdog timer (WDT) (Unit: ms)		○	○	○	○	200	-	-	R/W	NO	200
D1001	Displaying the program version of DVP-PLC (initial factory setting)	EH2/EH3	○	○	○	○	#	#	#	R	YES	#
		Others					-	-	-			
D1002*	Program capacity Sum of program memory # -> EC3-8K: 7920, ES/EC: 3792, EH3/SV2: 30000		○	○	○	○	#	-	-	R	NO	#
D1003	Sum of program memory # -> EC3-8K: -7920, ES/EC: -3792, EH3/SV2: -30000		○	○	○	○	-	-	-	R	YES	#
D1004*	Syntax check error code		○	○	○	○	0	0	-	R	NO	0
D1007	Number of times the low voltage of the battery is recorded		×	×	×	○	-	-	-	R	Yes	0
D1008*	When WDT is On, show the step number		○	○	○	○	0	-	-	R	NO	0
D1009	ES/EC/SX: recording number of occurrences of LV signals EH2/SV/EH3/SV2: register for SRAM lost data error code		○	○	○	○	-	-	-	R	YES	0
D1010*	Current scan time (Unit: 0.1ms)		○	○	○	○	0	-	-	R	NO	0
D1011*	Minimum scan time (Unit: 0.1ms)		○	○	○	○	0	-	-	R	NO	0
D1012*	Maximum scan time (Unit: 0.1ms)		○	○	○	○	0	-	-	R	NO	0
D1015*	0~32,767 (unit: 0.1ms) accumulative high-speed timer		×	○	×	○	0	-	-	R/W	NO	0
D1016	After a PLC is powered, it will delay detecting the extension modules connected to it for a certain amount of time. (Time unit: 100ms) The setting range is K20 ~ K50.		×	×	×	V1.62	-	-	-	R/W	YES	K25
D1018*	πPI (low byte)		○	○	○	○	H'0FDB	H'0FDB	H'0FDB	R/W	NO	H'0FDB
D1019*	πPI (high byte)		○	○	○	○	H'4049	H'4049	H'4049	R/W	NO	H'4049
D1020*	X0 ~ X7 input filter (Unit: ms); modulation range: 2~20ms		○	○	○	○	10	-	-	R/W	NO	10
D1021*	X10 ~ X17 input filter (Unit: ms)		○	○	○	○	10	-	-	R/W	NO	10
D1022	Multiplied frequency of A-B phase counters		○	○	×	×	0	-	-	R/W	NO	0
	The first starting frequency and the later ending frequency of CH4		×	×	×	24SV2	-	-	-	R/W	YES	200
D1023*	Register for detected pulse width, Unit: 0.1ms (Available in ES/EX/EC_V6.4, SX_V1.6, and EC3-8K)		○	○	○	×	0	-	-	R/W	NO	0
	The first starting frequency and the later ending frequency of CH4		×	×	×	24SV2	-	-	-	R/W	YES	200
D1025*	Code for communication request error		○	○	○	○	0	-	-	R	NO	0
D1026*	When M1156 is On, the (32-bit) pulse number for masking Y0 is set. If the value is less than or equal to 0, the function will not be enabled. (Default value: 0)	Low word	×	○	×	○	0	0	--	R/W	NO	0
D1027*		High word	×	○	×	○	0	0	--	R/W	NO	0
D1028	Index register E0		○	○	○	○	0	-	-	R/W	NO	0
D1029	Index register F0		○	○	○	○	0	-	-	R/W	NO	0
D1030*	Number of Y0 output pulses (low word) EC3-8K: latched device		○	○	○	×	0	-	-	R	NO	0
D1031*	Number of Y0 output pulses (high word) EC3-8K: latched device		○	○	○	×	0	-	-	R	NO	0
D1032	Number of Y1 output pulses (low word)		○	○	○	×	0	-	-	R/W	NO	0

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1033	Number of Y0 output pulses (high word)	○	○	○	×	0	-	-	R/W	NO	0
D1034	Work mode of frequency measurement card	×	×	×	○	-	-	-	R	YES	1
D1035*	No. of input point X as RUN/STOP	×	×	×	○	-	-	-	R/W	YES	0
D1036*	COM1 communication protocol	○	○	○	○	H'86	-	-	R/W	NO	H'86
D1037	Repetition time of HKY key	×	×	×	○	-	-	-	R/W	NO	0
D1038*	Delay time of data response when PLC CPU as slave in RS-485 communication, range: 0 ~ 10,000 (unit: 0.1ms) SA: delay time for sending the next communication data in PLC LINK (unit for SA/SX/EC3-8K: 1 scan cycle; EH2/SV/EH3/SV2: 0.1ms)	○	○	○	○	-	-	-	R/W	YES	0
D1039*	Fixed scan time (ms)	○	○	○	○	0	-	-	R/W	NO	0
D1040	On status of step No. 1	×	○	○	○	0	-	-	R	NO	0
D1041	On status of step No. 2	×	○	○	○	0	-	-	R	NO	0
D1042	On status of step No. 3	×	○	○	○	0	-	-	R	NO	0
D1043	On status of step No. 4	×	○	○	○	0	-	-	R	NO	0
D1044	On status of step No. 5	×	○	○	○	0	-	-	R	NO	0
D1045	On status of step No. 6	×	○	○	○	0	-	-	R	NO	0
D1046	On status of step No. 7	×	○	○	○	0	-	-	R	NO	0
D1047	On status of step No. 8	×	○	○	○	0	-	-	R	NO	0
D1048	Edit the proportional output for CSFO instruction	×	×	×	V1.88	0	-	-	R/W	NO	0
D1049	No. of alarm On	×	○	×	○	0	-	-	R	NO	0
D1050 ↓ D1055	MODRD is used to read data. The PLC system automatically converts the characters in D1070 ~ D1085 to hexadecimal values in the ASCII mode, or combine the low eight bits in D1070 ~ D1085 into eight 16-bit values in the RTU mode.	○	○	○	○	0	-	-	R	NO	0
D1056*	Present value at analog input channel CH0 in SX/EX or at CH0 on AD card in EH3/SV2	○	○	×	○	0	-	-	R	NO	0
D1057*	Present value at analog input channel CH1 in SX/EX or at CH1 on AD card in EH3/SV2	○	○	×	○	0	-	-	R	NO	0
D1058*	Present value at analog input channel CH2 in EX	○	×	×	×	0	-	-	R	NO	0
	Enabling X1 interrupt to get the counting value of C241 (M1056 is On)-Low word	×	×	×	○	0	0	-	R	NO	0
D1059*	Present value at analog input channel CH3 in EX	○	×	×	×	0	-	-	R	NO	0
	Enabling X1 interrupt to get the counting value of C241 (M1056 is On)-High word	×	×	×	○	0	0	-	R	NO	0
D1061	System error message: number of errors recorded in latched area	○	×	×	×	-	-	-	R	YES	0
D1062	Average times of AD0, AD1 in SX (2 ~ 10 times)	×	○	×	×	2	-	-	R/W	NO	2
D1063*	PLC reads/writes all programs (and password) and all latched data in the memory card. PLC reads all programs (and password) in the memory card: H55AA PLC writes all programs (and password) in the memory card: HAA55 PLC reads all latched data in the memory card:	×	×	×	○	0	-	-	R/W	NO	0

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	H55A9 PLC writes all latched data in the memory card: HA955										
D1064*	PLC reads/writes all programs (and password) and recipe in the internal FLASH ROM. PLC reads FLASH: H55AA; PLC writes FLASH: HAA55 H55A9/ H99AB/ HA955/ HAB55/ H8888 are added in EH3/SV2.	×	×	×	○	0	-	-	R/W	NO	0
D1067*	Error code for operational error	○	○	○	○	0	0	-	R	NO	0
D1068*	Locking the address of operational error	○	○	○	○	0	-	-	R	NO	0
D1070 ↓ D1085	Process of data for Modbus communication instruction. When the RS-485 communication instruction built-in the PLC sent out is received, the response messages will be stored in D1070 ~ D1085. You can view the response messages by checking these registers.	○	○	○	○	0	-	-	R	NO	0
D1086	High word of the set password in DVP-PCC01 (displayed in hex corresponded by its ASCII characters)	○	○	○	○	0	-	-	R/W	NO	0
D1087	Low word of the set password in DVP-PCC01 (displayed in hex corresponded by its ASCII characters)	○	○	○	○	0	-	-	R/W	NO	0
D1089 ↓ D1099	Process of data for Modbus communication instruction. When the RS-485 communication instruction built-in the PLC is executed, the words of the instruction will be stored in D1089 ~ D1099. You can check whether the instruction is correct by the contents in these registers.	○	○	○	○	0	-	-	R	NO	0
D1100	Corresponding status after LV signal is enabled	×	×	○	○	0	-	-	R/W	NO	0
D1101*	Start address of file registers	×	○	○	○	-	-	-	R/W	YES	0
D1102*	Number of data copied in file register	×	○	○	○	-	-	-	R/W	YES	1,600
D1103*	Start No. of file register D for storing data (has to be bigger than 2,000)	×	○	○	○	-	-	-	R/W	YES	2,000
D1109*	COM3 communication protocol setting (for EH3/SV2)	×	×	×	○	H86	-	-	R/W	NO	H86
D1110*	Average value at analog input channel CH0 in SX/EX or at CH0 on AD card in EH3/SV2	○	○	×	○	0	-	-	R	NO	0
D1111*	Average value at analog input channel CH1 in SX/EX or at CH1 on AD card in EH3/SV2	○	○	×	○	0	-	-	R	NO	0
D1112*	Average value at analog input channel CH2 in EX	○	×	×	×	0	-	-	R	NO	0
	The low word of the frequency on which CH3 (Y6/Y7) outputs pulses (EH3/SV2 V1.86 and above)	×	×	×	V1.86	0	-	-	R/W	NO	0
D1113*	Average value at analog input channel CH3 in EX	○	×	×	×	0	-	-	R	NO	0
	The high word of the frequency on which CH3 (Y6/Y7) outputs pulses (EH3/SV2 V1.86 and above)	×	×	×	V1.86	0	-	-	R/W	NO	0
D1115*	Setting for analog input work mode for 24SV2 PLC (default: Off)	×	×	×	24SV2	0	-	-	R/W	NO	-1
D1116*	CH0 of analog output in SX/EX CH0 of DA card in EH3/SV2	○	○	×	○	0	0	0	R/W	NO	0
D1117*	CH1 of analog output in SX/EX	○	○	×	○	0	0	0	R/W	NO	0

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	CH1 of DA card in EH3/SV2										
D1118*	Sampling time of analog/digital conversion in SX/EX/EH3/SV2 (ms) PS1: Only when the AD/DA card is in EH3/SV2 is the function supported.	○	○	×	○	5	-	-	R/W	NO	5
D1120*	COM2 (RS-485) communication protocol	○	○	○	○	H'86	-	-	R/W	NO	H'86
D1121	PLC communication address (latched)	○	○	○	○	-	-	-	R/W	YES	1
D1122	Remaining number of words of sent data	○	○	○	○	0	0	-	R	NO	0
D1123	Remaining number of words of received data	○	○	○	○	0	0	-	R	NO	0
D1124	Definition of start word (STX)	○	○	○	○	H'3A	-	-	R/W	NO	H'3A
D1125	Definition of the first end word	○	○	○	○	H'0D	-	-	R/W	NO	H'0D
D1126	Definition of the second end word	○	○	○	○	H'0A	-	-	R/W	NO	H'0A
D1127	Interruption request for receiving specific word in RS instruction (I150)	○	×	○	×	0	-	-	R/W	NO	0
	Number of pulses in the acceleration area of the positioning instruction (Low word)	×	×	×	○	0	-	-	R	NO	0
D1128	Number of pulses in the acceleration area of the positioning instruction (High word)	×	×	×	○	0	-	-	R	NO	0
D1129	Abnormal communication time-out (time: ms)	○	○	○	○	0	-	-	R/W	NO	0
D1130	Error code returning from Modbus	○	○	○	○	0	-	-	R	NO	0
D1131*	Low 16 bytes of high-speed counter value extracted by interruption I501 (Not available in SX series PLCs)	×	○	×	×	0	-	-	R	NO	0
	Output/input ratio of CH0 close-loop control (for EH3/SV2)	×	×	○	○	100	-	-	R/W	NO	100
D1132*	High 16 bytes of high-speed counter value extracted by interruption I501 (Not available in SX series PLCs)	×	○	×	×	0	-	-	R	NO	0
	Output/input ratio of CH1 close-loop control (for in EH3/SV2)	×	×	×	○	100	-	-	R/W	NO	100
D1133*	Number of pulses in the deceleration area of the positioning instruction (Low word)	×	×	×	○	0	-	-	R	NO	0
D1134*	Number of pulses in the deceleration area of the positioning instruction (High word)	×	×	×	○	0	-	-	R	NO	0
D1135*	Pulse number for masking Y2 (Low word) When M1158 = ON and the pulse number for masking Y2 is not 0, enabling instruction DDRVI/DPLSR for masking interrupt.	×	×	×	○	0	-	-	R/W	NO	0
D1136*	Pulse number for masking Y2 (High word) When M1158 = ON and the pulse number for masking Y2 is not 0, enabling instruction DDRVI/DPLSR and masking interrupt	×	×	×	○	0	-	-	R/W	NO	0
D1137*	Address where incorrect use of operand occurs	○	○	○	○	0	0	-	R	NO	0
D1140*	Number of right-side special extension modules (max. 8)	○	○	○	○	0	-	-	R	NO	0
D1142*	Number of points X in digital extension unit	○	○	○	○	0	-	-	R	NO	0
D1143*	Number of points Y in digital extension unit	○	○	○	○	0	-	-	R	NO	0
D1145*	Number of left-side special extension modules (max. 8) (only available in EH3-L, SV2)	×	×	×	×	0	-	-	R	NO	0
D1147	Type of memory card b0 = 0: no card existing (H0000) b0 = 1: with memory card b8 = 0: memory card Off (HFFFF) b8 = 1: memory card On (H0101)	×	×	×	○	0	-	-	R	NO	0
	Setting for the acceleration / deceleration time	×	×	×	24SV2	-	-	-	R/W	YES	100

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	for CH4 pulse output										
D1149	Type of function extension card 0: No card 1: RS-232 card (DVP-F232) 2: RS-422 card (DVP-F422) 8: Analog input card (DVP-F2AD) 9: Analog output card (DVP-F2DA) 11: Ethernet communication function extension card (DVP-FEN01) 12: CANopen communication function extension card (DVP-FCOPM) 13: RS-485 card (DVP-F485)	×	×	×	○	0	-	-	R	NO	0
	Setting for the acceleration / deceleration time for CH5 pulse output	×	×	×	24SV2	-	-	-	R/W	YES	100
D1150	Table count register in multi-group setting comparison mode of DHSZ command	×	×	×	○	0	0	0	R	NO	0
D1151	Table counting register for DHSZ multiple set values comparison mode	×	×	×	○	0	0	0	R	NO	0
D1152	High word of changed D value for DHSZ instruction	×	×	×	○	0	0	0	R	NO	0
D1153	Low word of changed D value for DHSZ instruction	×	×	×	○	0	0	0	R	NO	0
D1154*	Suggested deceleration time interval (10 ~ 32,767ms) for adjustable acceleration/deceleration pulse output Y0 (Not available in SX V3.0 and above)	×	○	×	×	200	-	-	R/W	NO	200
	Pulse number for masking Y4 (Low word)	×	×	×	○	0	0	-	R/W	NO	0
D1155*	Suggested deceleration frequency (-1 ~ -32,700Hz) for adjustable acceleration/deceleration pulse output Y0 (Not available in SX V3.0 and above)	×	○	×	×	-1,000	-	-	R/W	NO	-1,000
	Pulse number for masking Y4 (High word)	×	×	×	○	0	0	-	R/W	NO	0
D1156 ↓ D1159	Designated special D for RTMU, RTMD instructions (K0~K3)	×	×	×	○	0	-	-	R/W	NO	0
D1160	Designated special D for RTMU, RTMD instructions (K4)	×	×	×	○	0	-	-	R/W	NO	0
	Low word of the present output pulse frequency of CH0 (Y0/Y1)	×	×	×	V1.62	0	-	-	R/W	NO	0
D1161	Designated special D for RTMU, RTMD instructions (K5)	×	×	×	○	0	-	-	R/W	NO	0
	High word of the present output pulse frequency of CH0 (Y0/Y1)	×	×	×	V1.62	0	-	-	R/W	NO	0
D1162	Designated special D for RTMU, RTMD instructions (K6)	×	×	×	○	0	-	-	R/W	NO	0
	Low word of the present output pulse frequency of CH1 (Y2/Y3)	×	×	×	V1.62	0	-	-	R/W	NO	0
D1163	Designated special D for RTMU, RTMD instructions (K7)	×	×	×	○	0	-	-	R/W	NO	0
	High word of the present output pulse frequency of CH1 (Y2/Y3)	×	×	×	V1.62	0	-	-	R/W	NO	0
D1164	Designated special D for RTMU, RTMD instructions (K8)	×	×	×	○	0	-	-	R/W	NO	0
	Low word of the present output pulse frequency of CH2 (Y4/Y5)	×	×	×	V1.62	0	-	-	R/W	NO	0

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1165	Designated special D for RTMU, RTMD instructions (K9)	×	×	×	○	0	-	-	R/W	NO	0
	High word of the present output pulse frequency of CH2 (Y4/Y5)	×	×	×	V1.62	0	-	-	R/W	NO	0
D1166*	Low word of the present output pulse frequency of CH3 (Y6/Y7) (EH3/SV2 V1.62 ~ V1.84)	×	×	×	V1.62	0	-	-	R/W	NO	0
D1167*	High word of the present output pulse frequency of CH3 (Y6/Y7) (EH3/SV2 V1.62 ~ V1.84)	×	×	×	V1.62	0	-	-	R/W	NO	0
D1168	Interruption request for receiving specific word in RS instruction (I150)	×	○	○	○	0	-	-	R/W	NO	0
D1169	Interruption request for receiving specific word in RS instruction (I160)	×	×	×	○	0	-	-	R/W	NO	0
D1170*	PC value when executing single step (Single stSC)	×	×	×	○	0	0	0	R	NO	0
D1172*	Frequency of 2-phase pulse output (12Hz ~ 20kHz) (Not available in SX V3.0 and above)	×	○	×	×	0	-	-	R/W	NO	0
	EH3: enable the software filtering function for X20~X37 input points	×	×	×	○	0	-	-	R/W	NO	0
D1173*	Modes of 2-phase pulse output (K1 and K2) (Not available in SX V3.0 and above)	×	○	×	×	0	-	-	R/W	NO	0
	EH3: enable the software filtering function for X20~X37 input points	×	×	×	○	2	-	-	R/W	NO	2
D1174*	Low 16 bits of target numbers of 2-phase output pulses (Not available in SX V3.0 and above)	×	○	×	×	0	-	-	R/W	NO	0
	EH3: enable the software filtering function for X40~X57 input points	×	×	×	○	0	-	-	R/W	NO	2
D1175*	High 16 bits of target numbers of 2-phase output pulses (Not available in SX V3.0 and above)	×	○	×	×	0	-	-	R/W	NO	0
	EH3: enable the software filtering function for X20~X37 input points When CANRS instruction is in broadcast mode, the sum of th accumulated communication packet received	×	×	×	○ V2.06	2	-	-	R/W	NO	2
D1176	Low 16 bits of current numbers of 2-phase output pulses (Not available in SX V3.0 and above)	×	○	×	×	0	-	-	R/W	NO	0
D1177	High 16 bits of current numbers of 2-phase output pulses (Not available in SX V3.0 and above)	×	○	×	×	0	-	-	R/W	NO	0
	CANRS instruction communication timeout (ms)	×	×	×	V2.06	200	0	-	R/W	NO	0
D1178*	VR0 value	×	○	○	○	0	-	-	R	NO	0
D1179*	VR1 value	×	○	○	○	0	-	-	R	NO	0
D1180*	When X2 interruption (I201) occurs, immediately extracting the low 16 bytes from X0 high-speed counting value. (Only supports V1.8 and above versions.)	×	○	×	×	0	0	-	R/W	NO	0
	Enabling X2 to get the counting value of the high-speed counter C241 (M1057 is On)(Low word)	×	×	×	○	0	0	-	R	NO	0
D1181*	When X2 interruption (I201) occurs, immediately extracting the high 16 bytes from X0 high-speed counting value.	×	○	×	×	0	0	-	R/W	NO	0

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	Enabling X2 to get the counting value of the high-speed counter C241 (M1057 is On)(High word)	×	×	×	○	0	0	-	R	NO	0
D1182	Index register E1	×	○	○	○	0	-	-	R/W	NO	0
D1183	Index register F1	×	○	○	○	0	-	-	R/W	NO	0
D1184	Index register E2	×	○	○	○	0	-	-	R/W	NO	0
D1185	Index register F2	×	○	○	○	0	-	-	R/W	NO	0
D1186	Index register E3	×	○	○	○	0	-	-	R/W	NO	0
D1187	Index register F3	×	○	○	○	0	-	-	R/W	NO	0
D1188	Index register E4	×	×	○	○	0	-	-	R/W	NO	0
D1189	Index register F4	×	×	○	○	0	-	-	R/W	NO	0
D1190	Index register E5	×	×	○	○	0	-	-	R/W	NO	0
D1191	Index register F5	×	×	○	○	0	-	-	R/W	NO	0
D1192	Index register E6	×	×	○	○	0	-	-	R/W	NO	0
D1193	Index register F6	×	×	○	○	0	-	-	R/W	NO	0
D1194	Index register E7	×	×	○	○	0	-	-	R/W	NO	0
D1195	Index register F7	×	×	○	○	0	-	-	R/W	NO	0
D1196	Content in the display (available in SX)	×	○	×	×	0	-	-	R/W	NO	0
	When M1568 is ON, you can set the deceleration time for CH4 pulse output.	×	×	×	24SV2	0	-	-	R/W	YES	100
D1197	Refreshing the display (unit: 100ms) (for SX)	×	○	×	×	5	-	-	R/W	NO	5
	When M1569 is ON, you can set the deceleration time for CH5 pulse output.	×	×	×	24SV2	0	-	-	R/W	YES	100
D1198*	SX: When X3 interruption (I301) occurs, immediately extracting the low 16 byte from X1 high-speed counting value. (Only supports V1.8 and above versions.) EH3/SV2: Enabling X3 to get the counting value of the high-speed counter C241 (M1058 is On) (Low word)	×	○	×	○	0	0	-	R	NO	0
D1199*	SX: When X3 interruption (I301) occurs, immediately extracting the high 16 byte from X1 high-speed counting value. (Only supports V1.8 and above versions.) EH3/SV2: Enabling X3 to get the counting value of the high-speed counter C241 (M1058 is On) (High word)	×	○	×	○	0	0	-	R	NO	0
D1200*	Start latched address for auxiliary relays M0 ~ M999 # -> EH3/SV2: 500; SX: 512	×	○	×	○	-	-	-	R/W	YES	#
D1201*	End latched address for auxiliary relays M0 ~ M999	×	○	×	○	-	-	-	R/W	YES	999
D1202*	Start latched address for auxiliary relays M2000 ~ M4095	×	○	×	○	-	-	-	R/W	YES	2,000
D1203*	End latched address for auxiliary relays M2000 ~ M4095	×	○	×	○	-	-	-	R/W	YES	4,095
D1204*	Start latched address for 100ms timers T0 ~ T199	×	×	×	○	-	-	-	R/W	YES	H'FFFF
D1205*	End latched address for 100ms timers T0 ~ T199	×	×	×	○	-	-	-	R/W	YES	H'FFFF

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1206*	Start latched address for 10ms timers T200 ~ T2	×	×	×	○	-	-	-	R/W	YES	H'FFFF
D1207*	End latched address for 10ms timers T200 ~ T239	×	×	×	○	-	-	-	R/W	YES	H'FFFF
D1208*	Start latched address for16-bit counters C0 ~ C199 # -> EH3/SV2: 100; SX: 96	×	○	×	○	-	-	-	R/W	YES	#
D1209*	End latched address for 16-bit counters C0 ~ C199	×	○	×	○	-	-	-	R/W	YES	199
D1210*	Start latched address for 32-bit counters C200 ~ C234 # -> EH3/SV2: 220; SX: 216	×	○	×	○	-	-	-	R/W	YES	#
D1211*	End latched address for 32-bit counters C200 ~ C234	×	○	×	○	-	-	-	R/W	YES	234
D1212*	Start latched address for 32-bit high-speed counters C235 ~ C255	×	○	×	○	-	-	-	R/W	YES	235
D1213*	End latched address for 32-bit high-speed counters C235 ~ C255	×	○	×	○	-	-	-	R/W	YES	255
D1214*	Start latched address for steps S0 ~ S899 # -> EH3/SV2: 500; SX: 512	×	○	×	○	-	-	-	R/W	YES	#
D1215*	End latched address for steps S0 ~ S899 # -> EH3/SV2: 899; SX: 895	×	○	×	○	-	-	-	R/W	YES	#
D1216*	Start latched address for registers D0 ~ D999	×	○	×	○	-	-	-	R/W	YES	200
D1217*	End latched address for registers D0 ~ D999	×	○	×	○	-	-	-	R/W	YES	999
D1218*	Start latched address for registers D2000 ~ D9999	×	○	×	○	-	-	-	R/W	YES	2,000
D1219*	End latched address for registers D2000 ~ D9999(# -> EH3/SV2: 11999; EH2: 9999; SX: 4999)	×	○	×	○	-	-	-	R/W	YES	#
D1220	Phase of the 1st group pulse output CH0 (Y0, Y1)	×	SX V3.0	×	○	0	-	-	R/W	NO	0
D1221	Phase of the 2nd group pulse output CH1 (Y2, Y3)	×	×	×	○	0	-	-	R/W	NO	0
D1222	Time difference between direction signal and pulse output for the 1st group pulse CH0 (Y0, Y1) in DRVI, DDRVI, DRVA, DDRVA, PLSV, DPLSV	×	×	×	○	0	-	-	R/W	NO	0
D1223	Time difference between direction signal and pulse output for the 2nd group pulse CH1 (Y2, Y3) in DRVI, DDRVI, DRVA, DDRVA, PLSV, DPLSV	×	×	×	○	0	-	-	R/W	NO	0
D1225	Counting mode of the counter HHSC0	×	×	×	○	2	-	-	R/W	NO	2
D1226	Counting mode of the counter HHSC1	×	×	×	○	2	-	-	R/W	NO	2
D1227	Counting mode of the counter HHSC2	×	×	×	○	2	-	-	R/W	NO	2
D1228	Counting mode of the counter HHSC3	×	×	×	○	2	-	-	R/W	NO	2
D1229	Phase of the 3rd group pulse output CH2 (Y4, Y5) (available in EH3/SV2)	×	×	×	○	0	-	-	R/W	NO	0
D1230	Phase of the 4th group pulse output CH3 (Y6, Y7) (available in EH3/SV2)	×	×	×	○	0	-	-	R/W	NO	0
D1232*	Designating number of output pulses for CH0 deceleration and stop (low 16 bits)	×	×	×	○	0	-	-	R/W	NO	0
D1233*	Designating number of output pulses for CH0 deceleration and stop (high 16 bits)	×	×	×	○	0	-	-	R/W	NO	0

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1234*	Designating number of output pulses for CH1 deceleration and stop (low 16 bits)	×	×	×	○	0	-	-	R/W	NO	0
D1235*	Designating number of output pulses for CH1 deceleration and stop (high 16 bits)	×	×	×	○	0	-	-	R/W	NO	0
D1236*	Designating number of output pulses for CH2 deceleration and stop (low 16 bits)	×	×	×	○	0	-	-	R/W	NO	0
D1237*	Designating number of output pulses for CH2 deceleration and stop (high 16 bits)	×	×	×	○	0	-	-	R/W	NO	0
D1238*	Designating number of output pulses for CH3 deceleration and stop (low 16 bits)	×	×	×	○	0	-	-	R/W	NO	0
D1239*	Designating number of output pulses for CH3 deceleration and stop (high 16 bits)	×	×	×	○	0	-	-	R/W	NO	0
D1240	The low 16 bits of the end frequency of CH0 (available when the acceleration and the deceleration are separate)	×	×	×	○	0	0	-	R/W	NO	0
D1241	The high 16 bits of the end frequency of CH0 (available when the acceleration and the deceleration are separate)	×	×	×	○	0	0	-	R/W	NO	0
D1244	Number of idle speed output from CH0 in the instruction DCLLM	×	×	×	○	0	0	-	R/W	NO	0
D1245	Number of idle speed output from CH1 in the instruction DCLLM	×	×	×	○	0	0	-	R/W	NO	0
D1246	Number of idle speed output from CH2 in the instruction DCLLM	×	×	×	○	0	0	-	R/W	NO	0
D1247	Number of idle speed output from CH3 in the instruction DCLLM	×	×	×	○	0	0	-	R/W	NO	0
D1249	Communication timeout of COM1 instruction (unit: 1ms; the maximum value is 50ms; the value less than 50ms is count as 50ms.) (Only the instruction MODRW and RS are supported.) RS: 0 indicates that the timeout is not set.	×	×	×	○	0	-	-	R/W	NO	0
D1250	Communication error in COM1 instruction (Only the instruction MODRW and RS are supported.)	×	×	×	○	0	-	-	R/W	NO	0
D1252	Communication timeout of COM3 instruction (unit: 1ms; the maximum value is 50ms; the value less than 50ms is count as 50ms.) (Only the instruction MODRW and RS are supported.) RS: 0 indicates that the timeout is not set.	×	×	×	○	0	-	-	R/W	NO	0
D1253	Communication error in COM3 instruction (Only the instruction MODRW and RS are supported.)	×	×	×	○	0	-	-	R/W	NO	0
D1255	COM3 station address	×	×	×	○	-	-	-	R/W	YES	0
D1256 ↓ D1295	When the RS-485 communication instruction MODRW built-in the PLC is executed, the words of sent out by the instruction will be stored in D1256 ~ D1259. You can check whether the instruction is correct by the contents in these registers.	○	○	○	○	0	-	-	R	NO	0
D1296 ↓ D1311	The RS-485 communication instruction MODRW built in the PLC automatically converts the ASCII data received in the designated register into hex and store the hex data into D1296 ~ D1311.	○	○	○	○	0	-	-	R	NO	0
D1312	Number of times the instruction ZRN searches for Z phase and the number of displacement	×	×	×	○	0	0	-	R/W	NO	0
D1313*	Second in RTC: 00 ~ 59 #: read RTC and write	×	○	×	○	#	-	-	R/W	NO	0

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1314*	Minute in RTC: 00 ~ 59 #: read RTC and write	×	○	×	○	#	-	-	R/W	NO	0
D1315*	Hour in RTC: 00 ~ 23 #: read RTC and write	×	○	×	○	#	-	-	R/W	NO	0
D1316*	Day in RTC: 01 ~ 31 #: read RTC and write	×	○	×	○	#	-	-	R/W	NO	1
D1317*	Month in RTC: 01 ~ 12 #: read RTC and write	×	○	×	○	#	-	-	R/W	NO	1
D1318*	Week in RTC: 1 ~ 7 #: read RTC and write	×	○	×	○	#	-	-	R/W	NO	6
D1319*	Year in RTC: 00 ~ 99 (A.D.) #: read RTC and write	×	○	×	○	#	-	-	R/W	NO	0
D1320*	ID of the 1st right-side extension module	×	×	×	○	0	-	-	R	NO	0
D1321*	ID of the 2nd right-side extension module	×	×	×	○	0	-	-	R	NO	0
D1322*	ID of the 3rd right-side extension module	×	×	×	○	0	-	-	R	NO	0
D1323*	ID of the 4th right-side extension module	×	×	×	○	0	-	-	R	NO	0
D1324*	ID of the 5th right-side extension module	×	×	×	○	0	-	-	R	NO	0
D1325*	ID of the 6th right-side extension module	×	×	×	○	0	-	-	R	NO	0
D1326*	ID of the 7th right-side extension module	×	×	×	○	0	-	-	R	NO	0
D1327*	ID of the 8th right-side extension module	×	×	×	○	0	-	-	R	NO	0
D1328	Low word of offset pulse the 1st group pulses CH0 (Y0, Y1)	×	×	×	○	0	-	-	R/W	NO	0
D1329	High word of offset pulse the 1st group pulses CH0 (Y0, Y1)	×	×	×	○	0	-	-	R/W	NO	0
D1330	Low word of offset pulse the 2nd group pulses CH1 (Y2, Y3)	×	×	×	○	0	-	-	R/W	NO	0
D1331	High word of offset pulse the 2nd group pulses CH1 (Y2, Y3)	×	×	×	○	0	-	-	R/W	NO	0
D1332	Low word of the remaining number of pulses of the 1st group pulses CH0 (Y0, Y1)	×	×	×	○	0	-	-	R	NO	0
D1333	High word of the remaining number of pulses of the 1st group pulses CH0 (Y0, Y1)	×	×	×	○	0	-	-	R	NO	0
D1334	Low word of the remaining number of pulses of the 2nd group pulses CH1 (Y2, Y3)	×	×	×	○	0	-	-	R	NO	0
D1335	High word of the remaining number of pulses of the 2nd group pulses CH1 (Y2, Y3)	×	×	×	○	0	-	-	R	NO	0
D1336	EH3/SV2: Output low word of the present value of the 1st group pulses CH0 (Y0, Y1) EC3-8K: Output low word of the present value of Y2 and CH1 (Y2, Y3)	×	×	○	○	-	-	-	R	YES	0
D1337	EH3/SV2: Output high word of the present value of the 1st group pulses CH0 (Y0, Y1) EC3-8K: Output high word of the present value of Y2 and CH1 (Y2, Y3)	×	×	○	○	-	-	-	R	YES	0
D1338	EH3/SV2: Output low word of the present value of the 2nd group pulses CH1 (Y2, Y3) EC3-8K: Output low word of the present value of Y3 NOT latched	×	×	○	○	-	-	-	R	YES	0
D1339	EH3/SV2: Output high word of the present value of the 2nd group pulses CH1 (Y2, Y3) EC3-8K: Output high word of the present value of Y3 NOT latched	×	×	○	○	-	-	-	R	YES	0

Special D	Function		ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1340	Start/end frequency of the 1st group pulse output CH0 (Y0, Y1)		×	×	×	○	-	-	-	R/W	YES	200
D1341	Low word of max. output frequency	Fixed as 200kHz	×	○	×	○	-	-	-	R	YES	H'04D0
D1342	High word of max. output frequency		×	○	×	○	-	-	-	R	YES	3
D1343*	Acceleration/deceleration time for the 1st group pulse output CH0 (Y0, Y1)		×	×	×	○	-	-	-	R/W	YES	100
D1344	Low word of the number of compensation pulses of the 1st group pulses CH0 (Y0, Y1)		×	×	×	○	-	-	-	R/W	YES	0
D1345	High word of the number of compensation pulses of the 1st group pulses CH0 (Y0, Y1)		×	×	×	○	-	-	-	R/W	YES	0
D1346	Low word of the number of compensation pulses of the 2nd group pulses CH1 (Y2, Y3)		×	×	×	○	-	-	-	R/W	YES	0
D1347	High word of the number of compensation pulses of the 2nd group pulses CH1 (Y2, Y3)		×	×	○	○	-	-	-	R/W	YES	0
D1348	CH0 pulse output. When M1534 = On, it refers to the deceleration time		×	×	×	○	-	-	-	R/W	YES	100
D1349	CH1 pulse output. When M1535 = On, it refers to the deceleration time		×	×	×	○	-	-	-	R/W	YES	100
D1350	CH2 pulse output. When M1536 = On, it refers to the deceleration time		×	×	×	○	-	-	-	R/W	YES	100
D1351	CH3 pulse output. When M1537 = On, it refers to the deceleration time		×	×	×	○	-	-	-	R/W	YES	100
D1352	Start/end frequency of the 2nd group pulse output CH1 (Y2, Y3)		×	×	×	○	-	-	-	R/W	YES	200
D1353*	Acceleration/deceleration time of the 2nd group pulse output CH1 (Y2, Y3)		×	×	×	○	-	-	-	R/W	YES	100
D1354	Scan cycle for the PLC link (unit: 1ms) PS1: The maximum value is K32000 PS2: K0: The PLC link stops or the first detection is complete.		×	×	×	○	-	-	-	R/W	YES	100
D1355*	Starting reference for Master to read from Salve ID#1		×	○	×	○	-	-	-	R/W	YES	H'1064
D1356*	Starting reference for Master to read from Salve ID#2		×	○	×	○	-	-	-	R/W	YES	H'1064
D1357*	Starting reference for Master to read from Salve ID#3		×	○	×	○	-	-	-	R/W	YES	H'1064
D1358*	Starting reference for Master to read from Salve ID#4		×	○	×	○	-	-	-	R/W	YES	H'1064
D1359*	Starting reference for Master to read from Salve ID#5		×	○	×	○	-	-	-	R/W	YES	H'1064
D1360*	Starting reference for Master to read from Salve ID#6		×	○	×	○	-	-	-	R/W	YES	H'1064
D1361*	Starting reference for Master to read from Salve ID#7		×	○	×	○	-	-	-	R/W	YES	H'1064
D1362*	Starting reference for Master to read from Salve ID#8		×	○	×	○	-	-	-	R/W	YES	H'1064
D1363*	Starting reference for Master to read from Salve ID#9		×	○	×	○	-	-	-	R/W	YES	H'1064
D1364*	Starting reference for Master to read from Salve ID#10		×	○	×	○	-	-	-	R/W	YES	H'1064
D1365*	Starting reference for Master to read from Salve ID#11		×	○	×	○	-	-	-	R/W	YES	H'1064
D1366*	Starting reference for Master to read from Salve ID#12		×	○	×	○	-	-	-	R/W	YES	H'1064

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1367*	Starting reference for Master to read from Salve ID#13	×	○	×	○	-	-	-	R/W	YES	H'1064
D1368*	Starting reference for Master to read from Salve ID#14	×	○	×	○	-	-	-	R/W	YES	H'1064
D1369*	Starting reference for Master to read from Salve ID#15	×	○	×	○	-	-	-	R/W	YES	H'1064
D1370*	Starting reference for Master to read from Salve ID#16	×	○	×	○	-	-	-	R/W	YES	H'1064
D1371	Time unit of PWM Y0 pulse output when M1070=On	×	×	×	○	1	-	-	R/W	NO	1
D1372	Time unit of PWM Y2 pulse output when M1071=On	×	×	×	○	1	-	-	R/W	NO	1
D1373	Time unit of PWM Y4 pulse output when M1530=On	×	×	×	○	1	-	-	R/W	NO	1
D1374	Time unit of PWM Y6 pulse output when M1531=On	×	×	×	○	1	-	-	R/W	NO	1
D1375	Low word of the present value of the 3rd group pulses CH2 (Y4, Y5)	×	×	×	○	-	-	-	R/W	YES	0
D1376	High word of the present value of the 3rd group pulses CH2 (Y4, Y5)	×	×	×	○	-	-	-	R/W	YES	0
D1377	Low word of the present value of the 4th group pulses CH3 (Y6, Y7)	×	×	×	○	-	-	-	R/W	YES	0
D1378	High word of the present value of the 4th group pulses CH3 (Y6, Y7)	×	×	×	○	-	-	-	R/W	YES	0
D1379	Start frequency of the 1st section and end frequency of the last section for the 3rd group pulse output CH2 (Y4, Y5)	×	×	×	○	-	-	-	R/W	YES	200
D1380	Start frequency of the 1st section and end frequency of the last section for the 4th group pulse output CH3 (Y6, Y7)	×	×	×	○	-	-	-	R/W	YES	200
D1381	Acceleration/deceleration time for the 3rd pulse output CH2 (Y4, Y5)	×	×	×	○	-	-	-	R/W	YES	100
D1382	Acceleration/deceleration time for the 4th pulse output CH3 (Y6, Y7)	×	×	×	○	-	-	-	R/W	YES	100
D1383*	Setting up the time difference between direction signal and pulse output point for the 1st set of pulses CH2 (Y4, Y5) for DRVI, DDRVI, DRVA, DDRVA, PLSV, DPLSV instructions.	×	×	×	○	0	-	-	R/W	NO	0
D1384*	Setting up the time difference between direction signal and pulse output point for the 1st set of pulses CH3 (Y6, Y7) for DRVI, DDRVI, DRVA, DDRVA, PLSV, DPLSV instructions.	×	×	×	○	0	-	-	R/W	NO	0
D1386* ↓ D1393*	ID of the left-side extension module (available in SV2 and EH3-L)	×	×	×	○	0	-	-	R	NO	0
D1399*	Starting Salve ID designated by PLC LINK	×	○	×	○	-	-	-	R/W	YES	1
D1400*	Read the MAC Address of the left-side module (the numbering for the 1st left-side module is K100, the 8th one is K107) should work with M1145	×	×	×	V2.2	-	-	-	R/W	NO	0
D1401* ↓ D1403*	Storine the MAC Address by order	×	×	×	V2.2	-	-	-	R	NO	0
D1410* D1411*	Fixed slope ramp-up/down for high-speed output CH0 (Y0/Y1)	×	×	×	V1.88	200k	-	-	R/W	NO	200k
D1412*	Fixed slope ramp-up/down for high-speed	×	×	×	V1.88	200k	-	-	R/W	NO	200k

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1413*	output CH1 (Y2/Y3)										
D1415*	Starting reference for Master to write in Salve ID#1	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1416*	Starting reference for Master to write in Salve ID#2	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1417*	Starting reference for Master to write in Salve ID#3	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1418*	Starting reference for Master to write in Salve ID#4	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1419*	Starting reference for Master to write in Salve ID#5	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1420*	Starting reference for Master to write in Salve ID#6	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1421*	Starting reference for Master to write in Salve ID#7	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1422*	Starting reference for Master to write in Salve ID#8	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1423*	Starting reference for Master to write in Salve ID#9	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1424*	Starting reference for Master to write in Salve ID#10	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1425*	Starting reference for Master to write in Salve ID#11	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1426*	Starting reference for Master to write in Salve ID#12	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1427*	Starting reference for Master to write in Salve ID#13	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1428*	Starting reference for Master to write in Salve ID#14	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1429*	Starting reference for Master to write in Salve ID#15	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1430*	Starting reference for Master to write in Salve ID#16	×	○	×	○	-	-	-	R/W	YES	H'10C8
D1431*	Times of PLC LINK polling cycle	×	○	×	○	0	-	-	R/W	NO	0
D1432*	Current times of PLC LINK polling cycle	×	○	×	○	0	-	-	R/W	NO	0
D1433*	Number of salve units linked to PLC LINK	×	○	×	○	0	-	-	R/W	NO	0
D1434*	Data length to be read on Salve ID#1	×	○	×	○	-	-	-	R/W	YES	16
D1435*	Data length to be read on Salve ID#2	×	○	×	○	-	-	-	R/W	YES	16
D1436*	Data length to be read on Salve ID#3	×	○	×	○	-	-	-	R/W	YES	16
D1437*	Data length to be read on Salve ID#4	×	○	×	○	-	-	-	R/W	YES	16
D1438*	Data length to be read on Salve ID#5	×	○	×	○	-	-	-	R/W	YES	16
D1439*	Data length to be read on Salve ID#6	×	○	×	○	-	-	-	R/W	YES	16
D1440*	Data length to be read on Salve ID#7	×	○	×	○	-	-	-	R/W	YES	16
D1441*	Data length to be read on Salve ID#8	×	○	×	○	-	-	-	R/W	YES	16
D1442*	Data length to be read on Salve ID#9	×	○	×	○	-	-	-	R/W	YES	16
D1443*	Data length to be read on Salve ID#10	×	○	×	○	-	-	-	R/W	YES	16
D1444*	Data length to be read on Salve ID#11	×	○	×	○	-	-	-	R/W	YES	16
D1445*	Data length to be read on Salve ID#12	×	○	×	○	-	-	-	R/W	YES	16
D1446*	Data length to be read on Salve ID#13	×	○	×	○	-	-	-	R/W	YES	16
D1447*	Data length to be read on Salve ID#14	×	○	×	○	-	-	-	R/W	YES	16

Special D	Function	ES EX SS	SX	EC3-8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1448*	Data length to be read on Salve ID#15	×	○	×	○	-	-	-	R/W	YES	16
D1449*	Data length to be read on Salve ID#16	×	○	×	○	-	-	-	R/W	YES	16
D1450*	Data length to be written on Slave ID#1	×	○	×	○	-	-	-	R/W	YES	16
D1451*	Data length to be written on Slave ID#2	×	○	×	○	-	-	-	R/W	YES	16
D1452*	Data length to be written on Slave ID#3	×	○	×	○	-	-	-	R/W	YES	16
D1453*	Data length to be written on Slave ID#4	×	○	×	○	-	-	-	R/W	YES	16
D1454*	Data length to be written on Slave ID#5	×	○	×	○	-	-	-	R/W	YES	16
D1455*	Data length to be written on Slave ID#6	×	○	×	○	-	-	-	R/W	YES	16
D1456*	Data length to be written on Slave ID#7	×	○	×	○	-	-	-	R/W	YES	16
D1457*	Data length to be written on Slave ID#8	×	○	×	○	-	-	-	R/W	YES	16
D1458*	Data length to be written on Slave ID#9	×	○	×	○	-	-	-	R/W	YES	16
D1459*	Data length to be written on Slave ID#10	×	○	×	○	-	-	-	R/W	YES	16
D1460*	Data length to be written on Slave ID#11	×	○	×	○	-	-	-	R/W	YES	16
D1461*	Data length to be written on Slave ID#12	×	○	×	○	-	-	-	R/W	YES	16
D1462*	Data length to be written on Slave ID#13	×	○	×	○	-	-	-	R/W	YES	16
D1463*	Data length to be written on Slave ID#14	×	○	×	○	-	-	-	R/W	YES	16
D1464*	Data length to be written on Slave ID#15	×	○	×	○	-	-	-	R/W	YES	16
D1465*	Data length to be written on Slave ID#16	×	○	×	○	-	-	-	R/W	YES	16
D1466	Number of pulses required per revolution of motor at CH0 (low word)	×	×	×	○	-	-	-	R	YES	2,000
D1467	Number of pulses required per revolution of motor at CH0 (high word)	×	×	×	○	-	-	-	R	YES	0
D1468	Number of pulses required per revolution of motor at CH1 (low word)	×	×	×	○	-	-	-	R	YES	2,000
D1469	Number of pulses required per revolution of motor at CH1 (high word)	×	×	×	○	-	-	-	R	YES	0
D1470	Distance created for 1 revolution of motor at CH0 (low word)	×	×	×	○	-	-	-	R	YES	1,000
D1471	Distance created for 1 revolution of motor at CH0 (high word)	×	×	×	○	-	-	-	R	YES	0
D1472	Distance created for 1 revolution of motor at CH1 (low word)	×	×	×	○	-	-	-	R	YES	1,000
D1473	Distance created for 1 revolution of motor at CH1 (high word)	×	×	×	○	-	-	-	R	YES	0
D1474	Machine unit of CH0 movement (low word)	×	×	×	○	-	-	-	R	YES	0
D1475	Machine unit of CH0 movement (high word)	×	×	×	○	-	-	-	R	YES	0
D1476	Machine unit of CH1 movement (low word)	×	×	×	○	-	-	-	R	YES	0
D1477	Machine unit of CH1 movement (high word)	×	×	×	○	-	-	-	R	YES	0
D1478	Output/input ratio of CH2 close-loop control	×	×	×	○	100	-	-	R/W	NO	100
D1479	Output/input ratio of CH3 close-loop control	×	×	×	○	100	-	-	R/W	NO	100
D1480* ↓ D1495*	Data which is read from slave ID#1 in the PLC LINK at the time when M1353 is OFF Initial data register where the data read from slave ID#1~ID#16 in the PLC LINK is stored at the time when M1353 is ON (For EH3/SV2) (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1496* ↓ D1511*	Data which is written into slave ID#1 in the PLC LINK at the time when M1353 is OFF Initial data register where the data written into slave ID#1~ID#16 in the PLC LINK is stored at the time when M1353 is ON (For EH3/SV2) (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1512* ↓ D1527*	Data which is read from slave ID#2 in the PLC LINK at the time when M1353 is OFF Initial communication address where the data read from slave ID#17~ID#32 in the PLC LINK is stored at the time when M1353 is ON (For EH3/SV2) (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1528* ↓ D1543*	Data which is written into slave ID#2 in the PLC LINK at the time when M1353 is OFF Initial communication address where the data written into slave ID#17~ID#32 in the PLC LINK is stored at the time when M1353 is ON (For EH3/SV2) (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1544* ↓ D1559*	Data which is read from slave ID#3 in the PLC LINK at the time when M1353 is OFF Number of data read from slave ID#17~ID#32 in the PLC LINK at the time when M1353 is ON (For EH3/SV2) (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1560* ↓ D1575*	Data which is written into slave ID#3 in the PLC LINK at the time when M1353 is OFF Number of data written into slave ID#17~ID#32 in the PLC LINK at the time when M1353 is ON (For EH3/SV2) (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1576* ↓ D1591*	Data which is read from slave ID#4 in the PLC LINK at the time when M1353 is OFF Initial data register where the data read from slave ID#17~ID#32 in the PLC LINK is stored at the time when M1353 is ON (For EH3/SV2) (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1592* ↓ D1607*	Data which is written into slave ID#4 in the PLC LINK at the time when M1353 is OFF Initial data register where the data written into slave ID#17~ID#32 in the PLC LINK is stored at the time when M1353 is ON (For EH3/SV2) (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1608* ↓ D1623*	Data which is read from slave ID#5 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1624* ↓ D1639*	Data which is written into slave ID#5 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1640* ↓ D1655*	Data which is read from slave ID#6 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
D1656* ↓ D1671*	Data which is written into slave ID#6 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1672* ↓ D1687*	Data which is read from slave ID#7 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1688* ↓ D1703*	Data which is written into slave ID#7 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1704* ↓ D1719*	Data which is read from slave ID#8 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1720* ↓ D1735*	Data which is written into slave ID#8 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1736* ↓ D1751*	Data which is read from slave ID#9 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1752* ↓ D1767*	Data which is written into slave ID#9 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1768* ↓ D1783*	Data which is read from slave ID#10 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1784* ↓ D1799*	Data which is written into slave ID#10 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1800* ↓ D1815*	Data which is read from slave ID#11 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1816* ↓ D1831*	Data which is written into slave ID#11 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1832* ↓ D1847*	Data which is read from slave ID#12 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1848* ↓ D1863*	Data which is written into slave ID#12 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1864* ↓ D1879*	Data which is read from slave ID#13 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1880* ↓ D1895*	Data which is written into slave ID#13 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1896* ↓ D1911*	Data which is read from slave ID#14 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1900 ↓ D1915	When M1356 is ON, the values in these registers are defined as the station address (ID1~ID16). The default station address in D1399 is not used. Only when M1356 is ON is the latched function available.	×	×	×	○	1~16	-	-	R/W	NO	1~16
D1916 ↓ D1931	When M1356 is ON, the values in these registers are defined as the station address (ID17~ID32). The default station address in	×	×	×	○	17~32	-	-	R/W	NO	17~32

Special D	Function	ES EX SS	SX	EC3 -8K	EH3 SV2	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latched	Default
	D1399 is not used. Only when M1356 is ON is the latched function available.										
D1912* ↓ D1927*	Data which is written into slave ID#14 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1928* ↓ D1943*	Data which is read from slave ID#15 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1944* ↓ D1959*	Data which is written into slave ID#15 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1960* ↓ D1975*	Data which is read from slave ID#16 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R	YES	0
D1976* ↓ D1991*	Data which is written into slave ID#16 in the PLC LINK (They are non-latched data registers in SX V3.0 and above.)	×	○	×	○	-	-	-	R/W	YES	0
D1970* ↓ D1985*	Setthe filtering time for a built-in single input point (X0~X17; one point at a time); unit: μs; setting range 0~20,000.	×	×	×	V2.06	0	-	-	R/W	NO	0
D1988*	Fixed slope ramp-up/down for high-speed output CH2 (Y4/Y5)	×	×	×	V1.88	200k	-	-	R/W	NO	200k
D1989*											
D1990*	Fixed slope ramp-up/down for high-speed output CH3 (Y6/Y7)	×	×	×	V1.88	200k	-	-	R/W	NO	200k
D1991*											
D1992	Code of the ninth right side extension module	×	×	×	EH3	0	-	-	R	NO	0
D1993	Code of the tenth right side extension module	×	×	×	EH3	0	-	-	R	NO	0
D1994	Code of the eleventh right side extension module	×	×	×	EH3	0	-	-	R	NO	0
	DVP-PCC01 records the number of times the PLC code can be entered.	×	○	○	○	0	-	-	R/W	NO	0
D1995	Code of the twelfth right side extension module	×	×	×	EH3	0	-	-	R	NO	0
	DVPPCC01 records the length of PLC ID.	×	○	○	○	0	-	-	R/W	NO	0
D1996	The 1st word of PLC ID in DVPPCC01 (indicated by the hex value corresponding to ASCII word)	×	○	○	○	0	-	-	R/W	NO	0
D1997	The 2nd word of PLC ID in DVPPCC01 (indicated by the hex value corresponding to ASCII word)	×	○	○	○	0	-	-	R/W	NO	0
D1998	The 3rd word of PLC ID in DVPPCC01 (indicated by the hex value corresponding to ASCII word)	×	○	○	○	0	-	-	R/W	NO	0
D1999	The 4th word of PLC ID in DVPPCC01 (indicated by the hex value corresponding to ASCII word)	×	○	○	○	0	-	-	R/W	NO	0
D9800* ↓ D9879*	They are for left-side special modules which are connected to an EH3/SV2 Series CPU for data exchange.	×	×	×	○	-	-	-	R/W	YES	0
D9900* ↓ D9979*	They are for right-side special modules which are connected to an EH3/SV2 Series CPU for data exchange.	×	×	×	○	-	-	-	R/W	YES	0

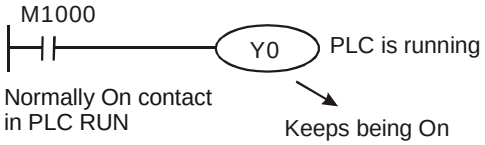
2.11 Functions of Special Auxiliary Relays and Special Registers

Function Group PLC Operation Flag

Number M1000 ~ M1003

M1000:

M1000 (A contact) is constantly “On” during operation and detection. When PLC is in RUN status, M1000 remains “On”.



M1001:

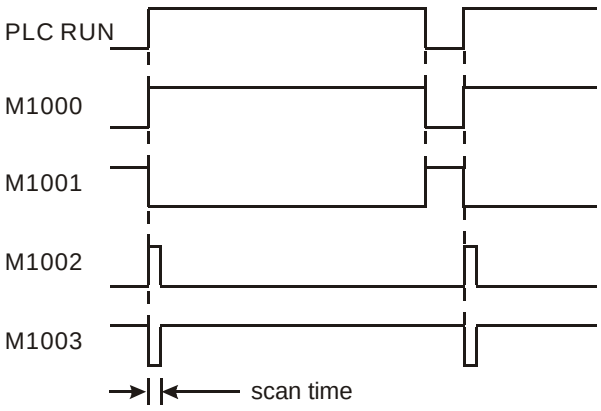
M1001 (B contact) is constantly “On” during operation and detection. When PLC is in RUN status, M1001 remains “On”.

M1002:

M1002 is to enable single positive pulse at the moment when RUN is activated (Normally OFF). The pulse width = 1 scan time. Use this contact for all kinds of initial settings.

M1003:

M1003 is to enable single negative pulse at the moment when RUN is activated (Normally ON).

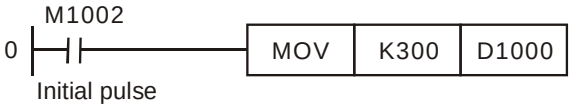


Function Group Monitor Timer

Number D1000

Contents:

1. Monitor timer is used for monitoring PLC scan time. When the scan time exceeds the set time in the monitor timer, the red ERROR LED indicator remains beaconing and all outputs will be “Off”.
2. The initial set value of the time in the monitor timer is 200ms. If the program is long or the operation is too complicated, MOV instruction can be used for changing the set value. See the example below for SV = 300ms.



3. The maximum set value in the monitor timer is 32,767ms. Please be noted that if the SV is too big, the timing of detecting operational errors will be delayed. Therefore, it is suggested that you remain the scan time of shorter than 200ms.
4. Complicated instruction operations or too many extension modules being connected to the CPU will result in the scan time being too long. Check D1010 ~ D1012 to see if the scan time exceeds the SV in D1000. In this case, besides modifying the SV in D1000, you can also add WDT instruction (API 07) into the PLC program. When the CPU execution progresses to WDT instruction, the internal monitor timer will be cleared as “0” and the scan time will not exceed the set value in the monitor timer.

Function Group Program Capacity

Number D1002

Contents:

The program capacity differs in different series of CPUs.

1. ES/EX/EC Series CPU: 3,792 steps
2. EC3-8K Series CPU: 7,920 steps
3. SX Series CPU: 7920 steps (There are 15872 steps in SX V3.0 or above.)
4. EH3/SV2 Series CPU: 30000 steps

Function Group Syntax Check

Number M1004, D1004, D1137

Contents:

1. When errors occur in syntax check, ERROR LED indicator will flash and special relay M1004 = On.
2. Timings for PLC syntax check:
   a) When the power goes from “Off” to “On”.
   b) When the program is written into PLC by WPLSoft, ISPSoft or HPP.
   c) When using WPLSoft/ISPSfot for programming in on-line mode for SX , EH3 and SV2 Series CPU.
3. The syntax check may start due to illegal use of instruction operands (devices) or incorrect program syntax loop. The error can be detected by the error code in D1004 and error table. The address where the error exists will be stored in D1137. (The address value in D1137 will be invalid if the error is a general loop error.)
4. See Chapter 2.13 for error codes for syntax check.

Function Group Data Backup Memory

Number M1005~M1007

Contents:

When the data backup memory card is installed in EH3 CPU, CPU will operate according to the On/Off of switch on the card. If the switch is “On”, the following comparisons will be conducted and the card will be copied to CPU. If the switch is “Off”, CPU will not perform any action.

1. M1005 = On: An error occurs in the comparison between the ciphers of CPU and the data backup memory card and CPU does not perform any action.
2. M1006 = On: The data backup memory card has not been initialized.
3. M1007 = On: Data in the program area of the data backup memory card do not exist, it means data doesn’t exist in the program area of data backup memory card.

Function Group Scan Time-out Timer

Number M1008, D1008

Contents:

1. M1008 = On: Scan time-out occurs during the execution of the program, and PLC ERROR LED indicator remains beaconing.
2. Users can use WPLSoft or ISPSoft to monitor the content (STEP address when WDT timer is “On”).

Function Group Checking Lost PLC SRAM Data

Number D1009, M1175, M1176

Contents:

1. bit0 ~ bit7 record the types of data lost. bit = 1 refers to losing data; bit = 0 refers to correct data.
2. What are lost

bit8 ~ 15	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reserved	PLC program	D register	T register	C register	File register	M relay	S step	password

3. After the PLC is powered, the data in SRAM will be verified. If the SRAM data are lost, the PLC will record the error in D1009 and set on M1175 or M1176 according to the content of the data.

Function Group Scan Time Monitor

Number D1010 ~ D1012

Contents:

The present value, minimum value and maximum value of scan time are stored in D1010 ~ D1012.

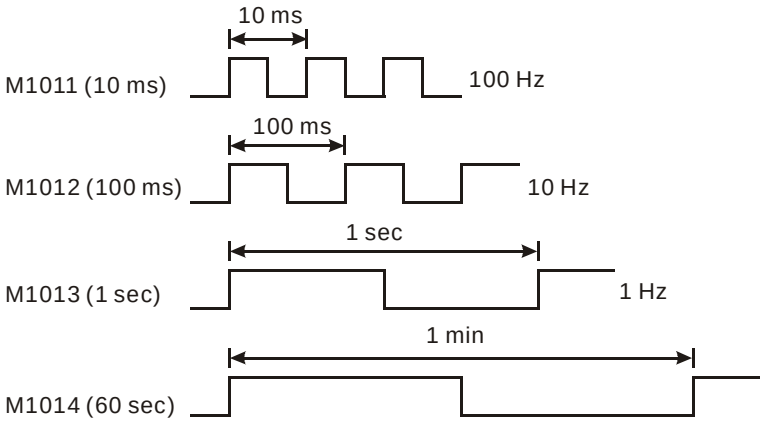
1. D1010: Present scan time value
2. D1011: Minimum scan time value
3. D1012: Maximum scan time value

Function Group Internal Clock Pulse

Number M1011 ~ M1014

Contents:

1. All PLC CPUs provide four different clock pulses. When PLC is powered, the four clock pulses will start automatically.



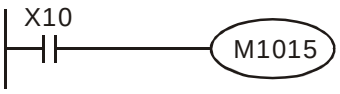
2. The clock pulses also start when PLC is in STOP status. The activation timing of clock pulses and that of RUN will not happen synchronously.

Function Group High-Speed Timer

Number M1015, D1015

Contents:

1. The steps for using special M and special D directly:
   a) High-speed counter is valid only when PLC is in RUN status for EH3/SV2, but is valid when PLC is in RUN or STOP stauts for SX.
   b) M1015 = On: High-speed counter D1015 is enabled only whtn PLC scans to END instruction. (Min. timing unit of D1015: 100us)
   c) Timing range of D1015: 0~32,767. When the timing reaches 32,767, the next timing restarts from 0.
   d) M1015 = Off: D1015 stops timing immediately.
2. EH3/SV2 Series CPU offers high-speed timer instruction HST. See API 196 HST for more details.
3. Example:
   a) When X10 = On, M1015 will be On. The high-speed timer will start to time and record the present value in D1015.
   b) When X10 = Off, M1015 will be Off. The high-speed timer will be disabled.



Function Group Real Time Clock

Number M1016, M1017, M1076, D1313 ~ D1319

Contents:

1. Special M and special D relevant to RTC

No.	Name	Function
M1016	Year (in A.D.) in RTC	Off: display the last 2 digits of year in A.D. On: display the last 2 digits of year in A.D. plus 2,000
M1017	±30 seconds correction	From “Off” to “On”, the correction is enabled. 0 ~ 29 second: minute intact; second reset to 0 30~ 59 second: minute + 1; second reset to 0
M1076	RTC malfunction	Set value exceeds the range; dead battery
M1082	Flag change on RTC	On: Modification on RTC
D1313	Second	0 ~ 59
D1314	Minute	0 ~ 59
D1315	Hour	0 ~ 23
D1316	Day	1 ~ 31
D1317	Month	1 ~ 12
D1318	Week	1 ~ 7
D1319	Year	0 ~ 99 (last 2 digits of Year in A.D.)

2. If the set value in RTC is incorrect, the time will be recovered as “Saturday, 00:00 Jan. 1, 2000” when PLC is powered and restarted.
3. D1313 ~ D1319 will immediately update the RTC only when in TRD instruction or WPLSoft monitoring mode.
4. How to make corrections on RTC:
   a) Use TWR instruction fir SX/EH3/SV2 Series CPU. See API 167 TWR for more details.
   b) Use WPLSoft or ISPSoft for setup.

Function Group π(PI)

Number D1018, D1019

Contents:

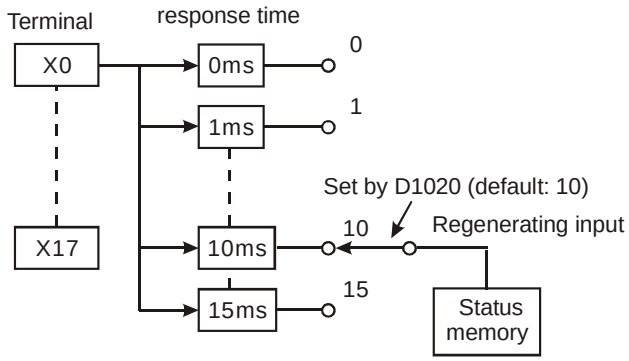
1. D1018 and D1019 are combined as 32-bit data register for storing the floating point value of π(PI)
2. Floating point value = H 40490FDB

Function Group Adjustment on Input Terminal Response Time

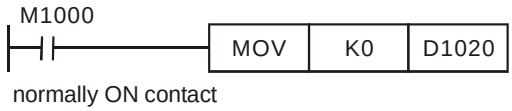
Number D1020, D1021

Contents:

1. D1020 can be used for setting up the response time of receiving pulses at X0 ~X7 for EC/ES/EX/EC3-8K/SX Series CPU. (Setup range: 0 ~ 20; Unit: ms)
2. D1021 can be used for setting up the response time of receiving pulses at X10 ~X17 for ES/EC/EC3-8K Series CPU. (Setup range: 0 ~ 20; Unit: ms)
3. D1020 can be used for setting up the response time of receiving pulses at X0 ~X7 for EH3/SV2 Series CPU. (Setup range: 0 ~ 60; Unit: ms)
4. D1021 can be used for setting up the response time of receiving pulses at X10 ~X17 for EH3/SV2 Series CPU. (Setup range: 0 ~ 60; Unit: ms)
5. When the power of PLC goes from “Off” to “On”, the content of D1020 and D1021 turn to 10 automatically.



6. If the following programs are executed during the program, the response time of X0 ~ X7 will be set to 0ms. The fastest response time of input terminals is 50μs due to that all terminals are connected with RC filter loop.



7. There is no need to make adjustment on response time when using high-speed counters and interruptions during the program.
8. Using API 51 REFF instruction has the same effect as modifying D1020 and D1021.

Function Group Software filtering function for the input points of the EH3 Series

Number D1172, D1173, D1174, D1175

Contents:

New software filtering funciton for input points X20~57, available for EH3 Series PLC with FW version 2.04 or later.

1. D1172 is the switch for X20~X37 to enable or disable the filtering funciton. Filtering time is set by D1173, ranging from 2~20, unit: 10 ms, default: 2. Refer to the following table for more information.
2. D1174 is the switch for X40~X57 to enable or disable the filtering funciton. Filtering time is set by D1175, ranging from 2~20, unit: 10 ms, default: 2. Refer to the following table for more information.

Example: Enable the filtering function for the input points X24, X25 and X30; set the filtering time to 30 ms. You need to set D1172 to H0130 and D1173 to K3.

Special Device	D1172			
Bit Number	b15~b12	b11~b8	b7~b4	b3~b0
Function	Bit ON / Bit OFF			
Input point	X37~X34	X33~X30	X27~X24	X23~X20
Value (in hex. format)	0	1	3	0

Function Group Hardware filtering function

Number M1630, M1631, D1970~D1985

Contents:

1. New hardware filtering funciton for EH3 and SV2 Series PLC with FW version 2.06 or later.

Model	Firmware version
All EH3 Series	V2.06
28SV2 Series	
24SV2 Series	

2. Use M1630 to enable or disable the filtering funciton for X0~X17.
   Use M1631 to set the filtering time for X0~X17. After you set the filtering time, PLC clears the flags automatically.

3. Use D1970~1977 to set the pulse-receiving response time for X0~X7. Use D1978~1985 to set the pulse-receiving response time for X10~X17. Setting range: 0~20,000; unit: μs.
   After updating D1970~1985, you need to use M1631 to update the filtering time to PLC.

Function Group Execution Completed Flag

Number M1029, M1030, M1036, M1037, M1102, M1103

Contents:

Using execution completed flag:

1. API 52 MTR, API 71 HKY, API 72 DSW, API 74 SEGL, API 77 PR: M1029 = On whenever the instruction completes one scan period.
2. API 57 PLSY, API 59 PLSR:
   a) M1029 will be “On” after Y0 pulse output of SX/ES/EX/EC3-8K is completed. M1030 will be “On” after Y1 pulse output is compeleted. When PLSY and PLSR instruction is “Off”, M1029 and M1030 turn “Off”. You have to reset M1029 and M1030 after the action is completed.
   b) M1029 will be “On” after Y0 and Y1 pulse output of EH3/SV2 is completed. M1030 will be “On” after Y2 and Y3 pulse output is compeleted. M1036 will be “On” after Y4 and Y5 pulse output of EH2/SV is completed. M1037 will be “On” after Y6 and Y7 pulse output is completed.When PLSY and PLSR instruction is “Off”, M1029, M1030, M1036 and M1037 turn “Off”. When the instruction is re-executed for the next time, M1029, M1030, M1036 and M1037 will turn “Off” and “On” again when the execution is completed.
3. API 63 INCD: M1029 will be “On” for a scan period when the assigned group numbers of data are compared.
4. API 67 RAMP, API 69 SORT:
   a) When the execution of the instruction is completed, M1029= On. You have to reset M1029.
   b) M1029 turns “Off” when the instruction is “Off”.
5. API 155 DABSR, API 156 ZRN, API 158 DRVI, API 159 DRVA for EH3/SV2 Series CPU:
   a) M1029 = On when the 1st output group Y0 and Y1 of EH3/SV2 is completed. M1030 = On when the 2nd output group Y2 and Y3 is completed.
   b) When the instruction is re-executed for the next time, M1029 or M1030 will turn “Off” and “On” again when the execution is completed.

Function Group Communication Error Code

Number M1025, D1025

Contents:

When PC or HMI is connected to the PLC and the PLC receives illegal communication request during the transmission of data, M1025 will be On and the error code will be written in D1025. See the error codes below.

01: Illegal instruction code

02: Illegal device address

03: Requested data exceed the range

07: Checksum error

Function Group Clear Instruction

Number M1031, M1032

Contents:

M1031 (clearing non-latched area) , M1032 (clearing latched area)

Device No.	Cleared Device
M1031	■ Contact status of Y, general-purpose M and general-purpose S ■ General-purpose contact and timing coil of T ■ General-purpose contact, counting coil reset coil of C ■ General-purpose present value register of D
M1031	■ General-purpose present value register of T ■ General-purpose present value register of C
M1032	■ Contact status of M and S for latched ■ Contact and timing coil of accumulative timer T ■ Contac and timing coil of high-speed counter C for latched ■ Present value register of D for latched ■ Present value register of accumulative timer T ■ Present value register of high-speed counter C for latched

Function Group Output Latched During STOP

Number M1033

Contents:

When M1033 = On and PLC goes from “RUN” to “STOP”, the On/Off status of output will be retained.

Assume the output contact load of the PLC is a heater, when PLC switches from RUN to STOP, the status of the heater will be retained. After the PLC program is modified, the PLC will RUN again.

Function Group All Output Y Inhibited

Number M1034

Contents:

When M1034 = On, all Y outputs will turn “Off”.

Function Group RUN/STOP Switch

Number M1035, D1035

Contents:

1. When M1035 = On, EH3/SV2 Series CPU will determine the content (K0 ~ K15) in D1035 to enable input points X0 ~ X17 as the RUN/STOP switch.
2. When M1035 = On, SX/EC3-8K Series CPU will enable the input point X3 (in SX) and X7 (in EC3-8K) as the RUN/STOP switch.

Function Group X0 Detecting Pulse Width

Number M1084, D1023

Contents:

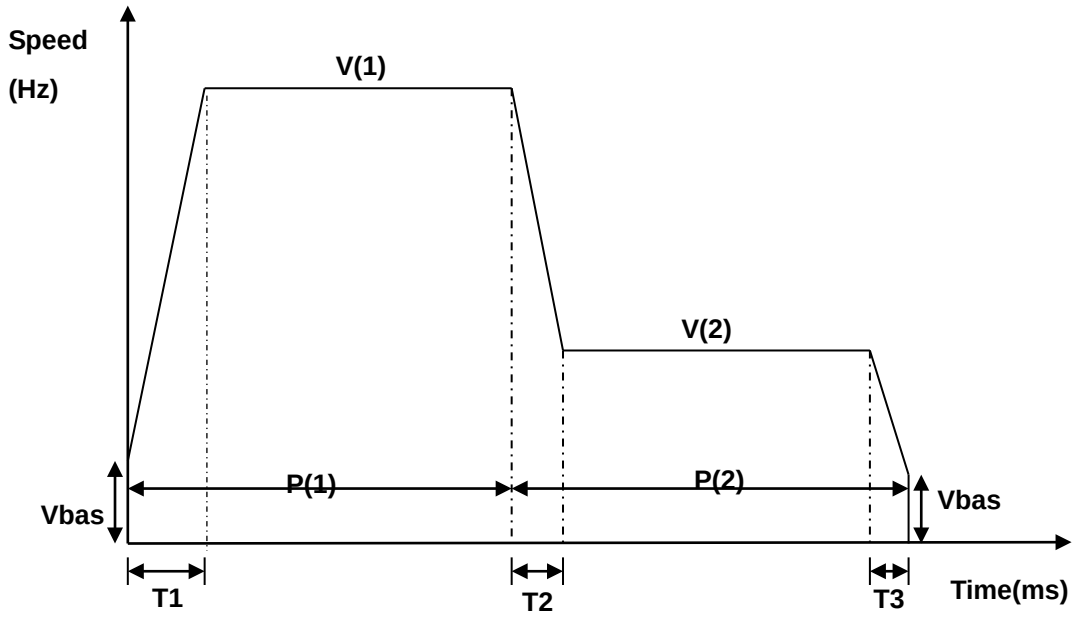
When M1084 = On, X0 of ES/EX/EC (V6.4), SX (V1.6), and EC3-8K can detect pulse width. Whenever X0 turns from "On" to "Off", the value is updated once and stored in D1023 (unit: 0.1ms). The minimum detectable width is 0.1ms and maximum 1,000ms.

Function Group Two speeds

Number M1119

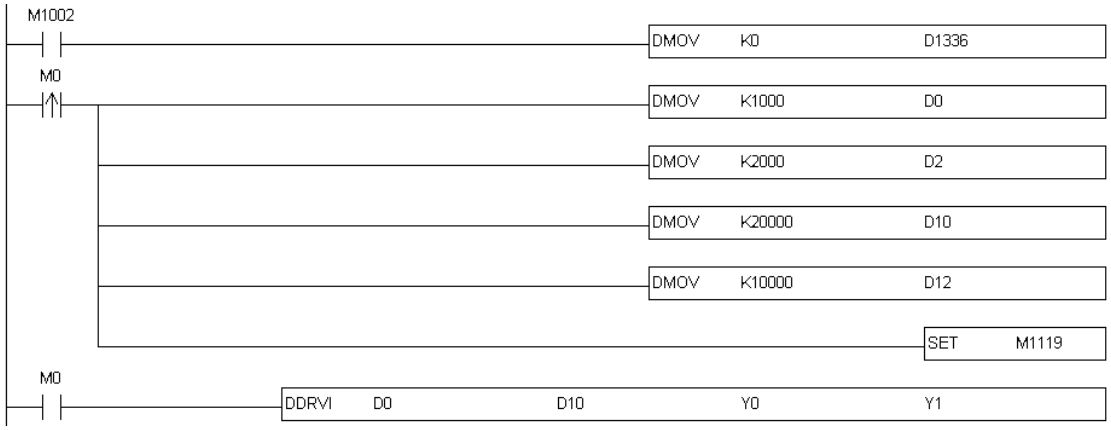
Contents:

1. Supports EH3/SV2 (V1.2) and versions above.
2. Before the instruction is enabled, M1119 has to be set to On. After the instruction is enabled, M1119 is set to Off automatically.
3. $\mathbf{S}_1$ and $\mathbf{S}_1$+1 in DDRVI/DDRVA designates the position of the first speed and the position of the second speed respectively, $\mathbf{S}_2$ and $\mathbf{S}_2$+1 designates the fist speed and the second speed respectively.
4. The second speed must be less than the first speed. Otherwise, the first speed is taken.



Vbase	T1	T2+T3	P(1)	V(1)	P(2)	V(2)
Initial frequency	Acceleration time	Deceleration time	Position of the first speed	First speed	Position of the second speed	Second speed

Example:

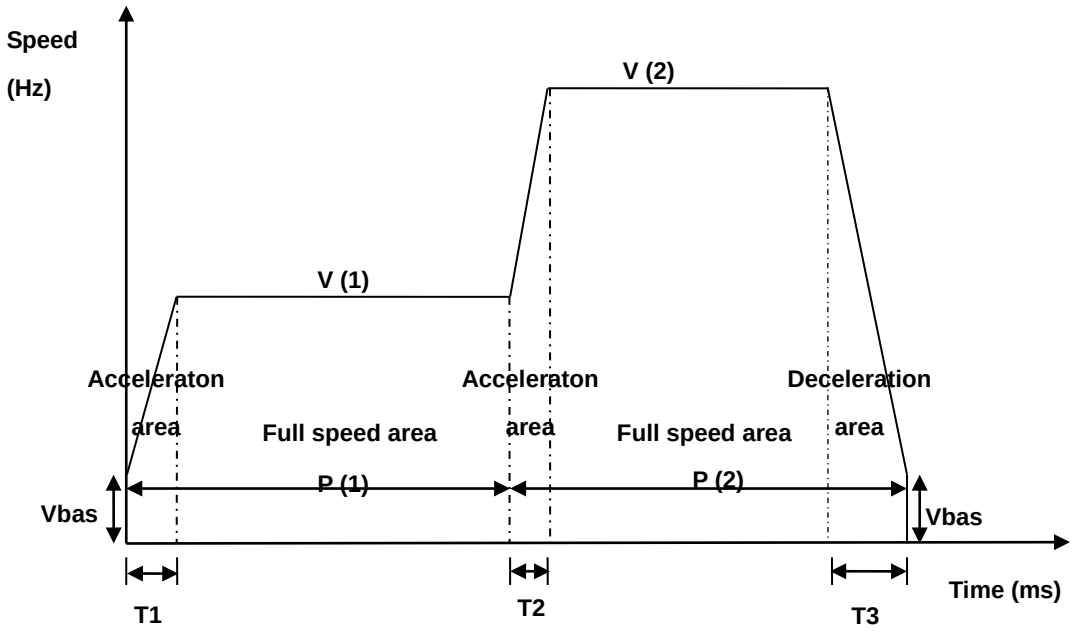


Explanation:

1. Set P(1) to 1000 pulse, P(2) to 2000 pulse, V(1) to 20kHz, and V(2) to 10kHz.
2. Set M1119 to On.
3. Execute DDRVI/DDRVA.

∗New functions only available for EH3/SV2 V1.88 and later versions:

1. The second speed can be higher than the first speed. See the description below.



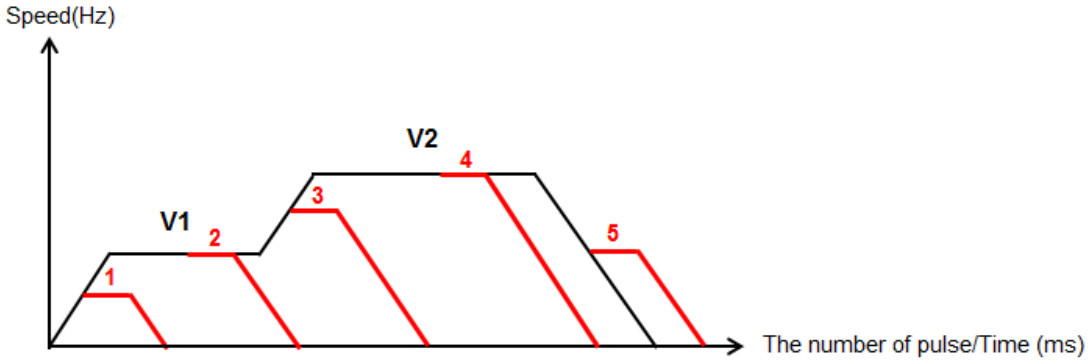
Vbase	T1+T2	T3	P(1)	V(1)	P(2)	V(2)
Initial frequency	Acceleration time	Deceleration time	Position of the first speed	First speed	Position of the second speed	Second speed

2. Marking alignment on two speeds.

Refer to M1156 to M1159 for more information on marking alignment.

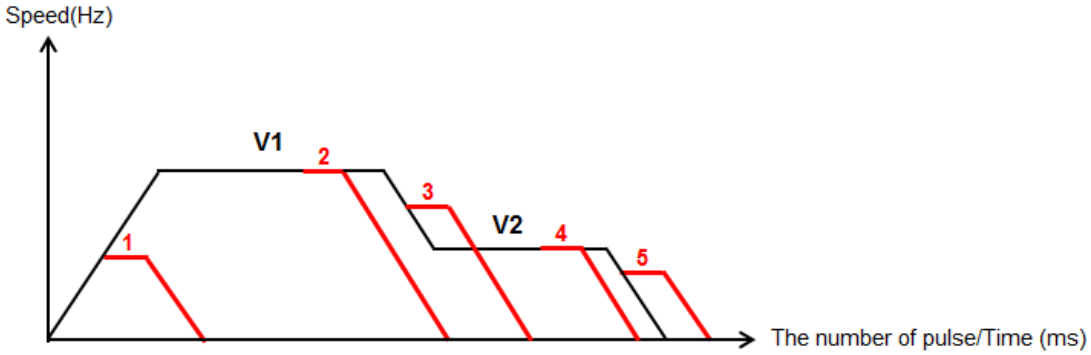
When the second speed is higher than the first speed, the marking aligment goes as below.

Acelceration area: As the number of pulse in deceleration goes to decelerate and then to a stop, see red line 1 and 3 from the image below.
Full speed area: As the number of pulse in deceleration goes to decelerate and then to a stop, see red line 2 and 4 from the image below.
Deelceration area: As the number of pulse in deceleration goes to decelerate and then to a stop, see red line 5 from the image below.



When the first speed is higher than the second speed, the marking aligment goes as below.

Acelceration area: As the number of pulse in deceleration goes to decelerate and then to a stop, see red line 1 from the image below.
Full speed area: As the number of pulse in deceleration goes to decelerate and then to a stop, see red line 2 and 4 from the image below.
Deelceration area: As the number of pulse in deceleration goes to decelerate and then to a stop, see red line 3 and 5from the image below.



Function Group Communication Port Function

Number M1120, M1136, M1138, M1139, M1143, D1036, D1109, D1120

Contents:

1. Supports ES/EX/EC (V6.0), EC3-8K, SX (V1.2), EH3/SV2 (V1.0) and higher versions.
2. COM ports (COM1: RS-232; COM2: RS-485) in SX Series CPU and COM ports (COM1: RS-232; COM2: RS-232/RS-485/RS-422) in EH3/SV2 Series CPU support Modbus ASCII/RTU communication format with speed of up to 115,200bps. COM1 and COM2 can be used at the same time. The communication port (COM3: DVP-F232, -F485, -F422) in EH3 Series CPU supports ASCII/RTU communication format with speed of up to 115,200bps.

COM1:	For slave stations only. Supports ASCII/RTU communication format, adjustable baud rate with speed of up to 115,200bps, and modification on data length (data bits, parity bits, stop bits). EH3/SV2 Series CPUs can be as a masters or slaves, and supports ASCII/RTU communication format, adjustable baud rate with speed of up to 115,200bps, and modification on data length (data bits, parity bits, stop bits).
COM2:	For master or slave stations. Supports ASCII/RTU communication format, adjustable baud rate with speed of up to 115,200bps, and modification on data length (data bits, parity bits, stop bits).
COM3 (DVP-F232, -F485, -F422 cards)	EH3 Series CPUs can be as a masters or slaves, and supports ASCII/RTU communication format, adjustable baud rate with speed of up to 115,200bps, and modification on data length (data bits, parity bits, stop bits).

■ Communication Format Settings:

COM1:
1. Communication format is set in D1036. b8 ~ b15 do not support the communication protocol of COM1 (RS-232) Slave.
2. The communication format in EH3/SV2 Series CPU is set in D1036. b8 ~ b15 do not support the communication protocol of COM1 (RS-232) Slave/Master.
3. Communication setting in M1138 remains.
4. M1139 is set in ASCII/RTU mode

COM2:
1. Communication format is set in D1120. Communication protocol of COM2 (RS-485) Master or Slave.
2. Communication setting in M1120 remains
3. M1143 is set in ASCII/RTU mode

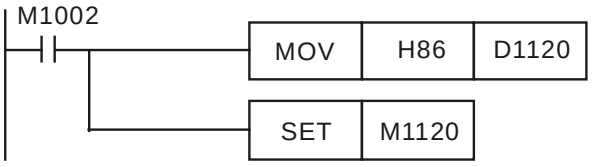
COM3:
1. The communication format in EH3-SV2 is set in D1109. b0 ~ b3 and b8 ~ b15 do not support the communication protocol of COM3 (DVP-F232, -F485, -F422 cards) Slave or Master.
2. Communication setting in M1136 remains

Communication protocols and how to set:

	Content	0	1
b0	Data length	b0 = 0:7	b0 = 1:8
b1 b2	parity bit	b2, b1 = 00 : None b2, b1 = 01 : Odd b2, b1 = 11 : Even	
b3	stop bits	b3 = 0:1 bit	b3 = 1:2 bit
b7 ~ b4	b7 ~ b4 = 0001 (H1) : 110 bps b7 ~ b4 = 0010 (H2) : 150 bps b7 ~ b4 = 0011 (H3) : 300 bps b7 ~ b4 = 0100 (H4) : 600 bps b7 ~ b4 = 0101 (H5) : 1,200 bps b7 ~ b4 = 0110 (H6) : 2,400 bps b7 ~ b4 = 0111 (H7) : 4,800 bps b7 ~ b4 = 1000 (H8) : 9,600 bps b7 ~ b4 = 1001 (H9) : 19,200 bps b7 ~ b4 = 1010 (HA) : 38,400 bps b7 ~ b4 = 1011 (HB) : 57,600 bps b7 ~ b4 = 1100 (HC) : 115,200 bps		
b8	Select start bit	b8 = 0:None	b8 = 1:D1124
b9	Select the 1st end bit	b9 = 0:None	b9 = 1:D1125
b10	Select the 2nd end bit	b10 = 0:None	b10 = 1:D1126
b15 ~ b11	Not defined		

Example 1: Modifying communication format of COM2

1. Add the program code below on top of the program to modify the communication format of COM2. When PLC switches from STOP to TUN, the program will detect whether M1120 is On in the first scan time. If M1120 is On, the program will modify the relevant settings of COM2 according to the value set in D1120.
2. Modify the communication format of COM2 into ASCII mode, 57,600bps, 7 data bits, even parity, 1 stop bit (57,600, 7, E, 1)

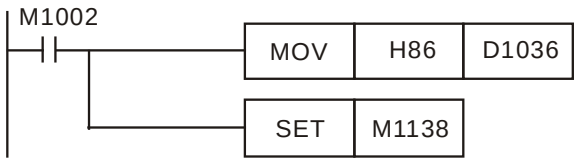


Notes:

1. If COM2 is to be used as a Slave terminal, make sure there is no communication instruction existing in the program.
2. After the communication format is modified, the format will stay intact when PLC switches from RUN to STOP.
3. If you shut down the power of the PLC and repower it again, the modified communication format will return to default setting.

Example 2: Modifying the communication format of COM1

1. Add the program code below on top of the program to modify the communication format of COM1. When PLC switches from STOP to TUN, the program will detect whether M1138 is On in the first scan time. If M1138 is On, the program will modify the relevant settings of COM1 according to the value set in D1036.
2. Modify the communication format of COM1 into ASCII mode, 115,200bps, 7 data bits, even parity, 1 stop bit (115,200, 7, E, 1)

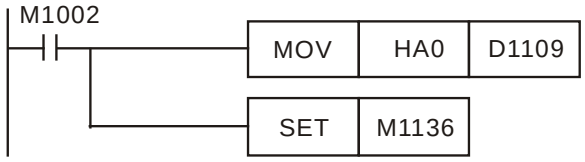


Notes:

1. After the communication format is modified, the format will stay intact when PLC switches from RUN to STOP.
2. If you shut down the power of the PLC and repower it again, the modified communication format will return to default setting.

Example 3: Modifying the communication format of COM3

1. The communication format of COM3 is fixed as 7 data bits, even parity, 1 stop bit. Add the program code below on top of the program to modify the baud rate of COM3 into 38,400bps. When PLC switches from STOP to TUN, the program will detect whether M1136 is On in the first scan time. If M1136 is On, the program will modify the relevant settings of COM3 according to the value set in D1109.
2. Modify the baud rate of COM3 into 38,400bps



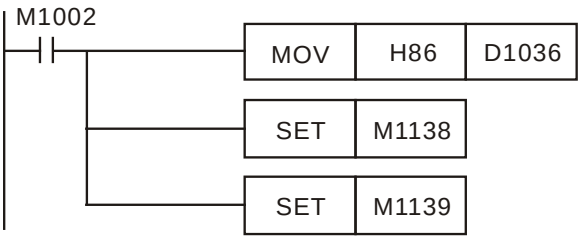
Notes:

1. After the communication format is modified, the format will stay intact when PLC switches from RUN to STOP.
2. If you shut down the power of the PLC and repower it again, the modified communication format will return to default setting.

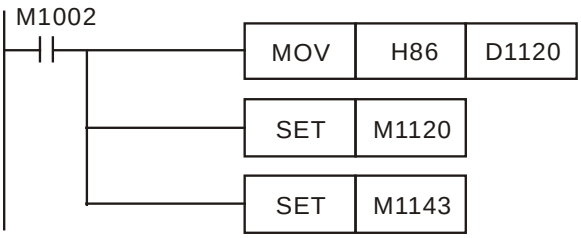
Example 4: Setting up RTU mode of COM1 and COM2

1. COM1 and COM2 support ASCII/RTU mode. COM1 is set by M1139 and COM2 is set by M1143. When the flags are On, they are in RTU mode; when the flags are Off, they are in ASCII mode.
2. How to set up RTU mode

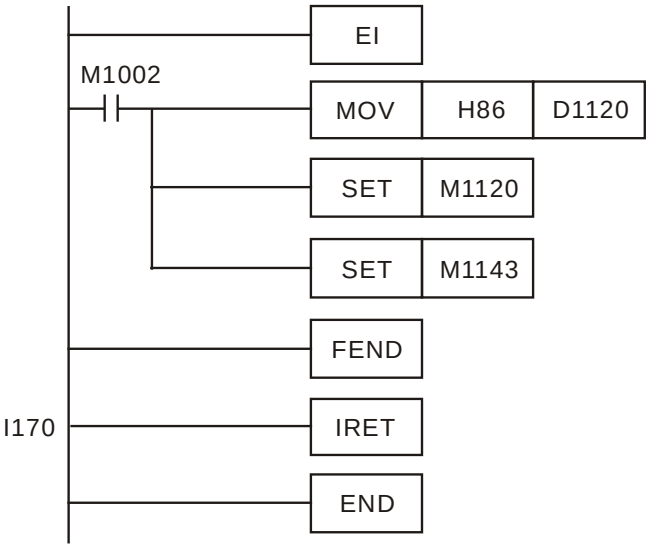
   COM1: (9,600, 8, E, 1, RTU)



COM2: (9,600, 8, E, 1, RTU)



3. EH3/SV2 Series CPU supports the generation of interruption I170 when the data receiving is completed in Slave mode.
4. Normally when the communication terminal of the PLC is in Slave mode, PLC will not immediately process the communication data entered but process it after the END is executed. Therefore, when the scan time is very long and you need the communication data to be processed immediately, you can use interruption I170 for this matter.
5. Example of interruption I170 (after the data receiving is completed in Slave mode)

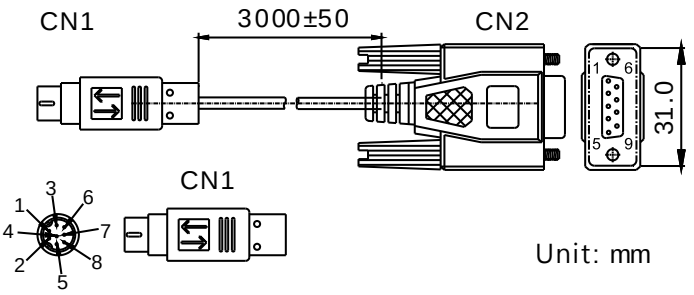


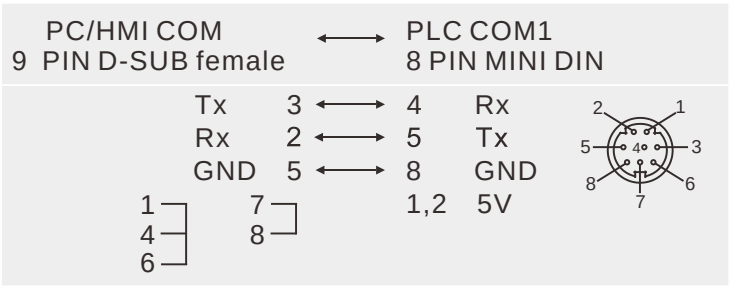
With I170 in the program, when COM2 is in Slave mode and there are communication data coming in, PLC will process the data and respond immediately.

Notes:

1. DO NOT update program on-line when using I170.
2. The scan time of PLC will be slightly longer.

Definitions of the pins in COM1: (It is suggested that the Delta communication cable DVPACAB2A30.)





Note: The 5V power from COM1 of the PLC CPU is only for DVPPCC01 and TP04G. Do NOT use it on other devices.

Function Group Communication Response Delay

Number D1038

Contents:

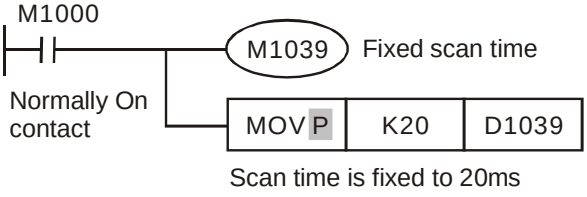
1. When PLC is used as slave station, in RS-485 communication interface, users can set up communication response delay time ranging from 0 to 10,000 (0 ~ 1 second). If the time is without the range, D1038 = O (time unit: 0.1ms). The set value of time must be less than that in D1000(scan time-out timer WDT).
2. In PLC LINK, you can set up delayed transmission of the next communication data (unit: 1 scan period for SX/EC3-8K; 0.1ms for EH3/SV2).

Function Group Fixed Scan Time

Number M1039, D1039

Contents:

1. When M1039 = On, the scan time of program is determined by the content in D1039. When the execution of the program is completed, the next scan will take place when the fixed scan time is reached. If the content in D1039 is less than the actual scan time of the program, the scan time will follow the actual scan time of the program.



2. Instructions related to scan time, RAMP (API 67), HKY (API 71), SEGL (API 74), ARWS (API 75) and PR (API 77) should be used together with “fixed scan time” or “constant interruption”.
3. Particularly for HKY instruction, when the 16-digit button input is operated by 4x4 matrix, the scan time has to be fixed to longer than 20ms.
4. The scan time in D1010 ~ D1012 also includes fixed scan time.

Function Group Analog Function

Number D1056 ~ D1059, D1062, D1110 ~ D1113, D1115 ~ D1118, D1588 ~ D1589

Contents:

1. Resolution of analog input channel: 10 bits for EX, corresponding to 0 ~ ±10 V (-512 ~ +511) or 0 ~ ±20 mA (-512 ~ +511); 12 bits for SX, corresponding to 0 ~ ±10 V (-2,000 ~ +2,000) or 0 ~ ± 20 mA (-1,000 ~ +1,000).
2. Resolution of analog output channel: 8 bits for EX, corresponding to 0 ~ 10 V (0 ~ 255) or 0 ~ 20 mA (0 ~ 255); 12 bits for SX, corresponding to 0 ~ ±10 V (-2,000 ~ +2,000) or 0 ~ ±20 mA (-2,000 ~ +2,000).
3. Sampling time of analog/digital conversion. Default setting = 5; unit: ms. If D1118 ≤ 5, it will be regarded as 5ms.
4. Resolution of EH3/SV2 analog input AD card (DVP-F2AD): 12 bits 0 ~ 10 V (0 ~ +4,000) or 11 bits 0 ~ 20 mA (0~+2,000)
5. Resolution of EH3/SV2 analog input DA card (DVP-F2DA): 12 bits 0 ~ 10 V (0 ~ +4,000) or 0 ~ 20 mA (0 ~ +4,000)
6. Resolution of 24SV2 analog input: 12 bits, current 11 bit, 0 ~ 10 V (0 ~ +4,000) or 0 ~ 20 mA (0 ~ +2,000).

Device No.	Function
D1056	Present value of EX/SX analog input channel CH0 and EH3/SV2 AD card channel CH0
D1057	Present value of EX/SX analog input channel CH1 and EH3/SV2 AD card channel CH1
D1058	Present value of EX analog input channel CH 2
D1059	Present value of EX analog input channel CH 3
D1062	Average times (2 ~ 10) of SX AD0 and AD1
	Average times (1 ~ 20) of 24SV2 AD0 and AD1
D1110	Average value of EX/SX analog input channel CH0, EH3/SV2 AD card channel CH0 and 24SV2 analog input channel AD0 (the average value is the current value of K1)
D1111	Average value of EX/SX analog input channel CH1, EH3/SV2 AD card channel CH1 and 24SV2 analog input channel AD1 (the average value is the current value of K1)
D1112	Average value of EX analog input channel CH2
D1113	Average value of EX analog input channel CH3
D1115	24SV2 analog input work mode settings, default: HFFFF bit 0~3 is AD0 work mode bit 4~7 is AD1 work mode analog input work modes: 0x0 is 0V ~ +10V voltage mode 0x1 is 0 mA ~ +20 mA current mode 0xF is OFF
D1116	EX/SX analog output channel CH0 and EH3/SV2 DA card channel CH0
D1117	EX/SX analog output channel CH1 and EH3/SV2 DA card channel CH1
D1118	Sampling time (ms) of SX/EX/EH3/SV2/24SV2 analog/digital conversion
M1588	Conversion value in analog input AD0 for 24SV2 PLC exceeding the maximum value, 4047 (voltage) / 2023 (current)
M1589	Conversion value in analog input AD1 for 24SV2 PLC exceeding the maximum value, 4047 (voltage) / 2023 (current)

Function Group Reading/Writing the data from/into the memory card

Number M1163, D1063

Contents:

1. The function of reading/writing data from/into the memory card in a PLC can be used only when the PLC stops.
2. The reading/writing of the data between the EH3 Series CPU and the memory card:

M1163 state	D1063 code	Function
On	0x55AB	The data in D2000~D11999 and file registers 0~4999 are read from the memory card, and copied to the main operation area in the CPU.
	0x55AA	The program code is read from the memory card, and copied to the main operation area in the CPU.
	0x55A9	The data in D2000~D11999 and file registers 0~4999 are read and copied to the main operation area in the CPU.
	0xAB55	The data in D2000~D11999 and file registers 0~4999 in the main operation area in the CPU are copied to the memory card.
	0xAA55	The program in the CPU is copied to the memory card.
	0xA955	The data in D2000~D11999 in the main operation area in the CPU is copied to the memory card.
	0x8888	Clearing the data in the memory card

Note:

(1) After the reading/writing of the data is complete, M1163 is automatically set to Off. After the reading/writing of the data is complete, the following flags are On.

M1189 → The data which is read/written is correct.

M1075 → An error occurs when the data is written into the memory card. Please check if the memory card is inserted or damaged.

M1005 → The PLC ID on the CPU or the main password is different from the memory card.

M1006 → No data or program is in the memory card.

(2) Enter the function code in D1063 first, and then set M1163 to On. Otherwise, M1163 is automatically reset to Off.

(3) If an error occurs during the reading/writing of the data, the special M will be set. No error LED indicator will flash or no situation in which the CPU can not run occurs. Therefore, if customers need an alarm, please make the warning message on the device or superior machine according to the flags above.

(4) Owing to the fact that the storage material of the memory card is Flash ROM, it takes some execution time to write the data into the memory card.

(5) When the program is copied, the CPU automatically copies the password function (including the main password, the limit on the number of errors, the subroutine password, and the PLC ID).

Function Group Reading/Writing the data from/into the backup area

Number M1164, D1064

Contents:

1. The function of reading/writing data from/into the backup area in a PLC can be used only when the PLC stops.
2. The reading/writing of the data between the EH3/SV2 Series CPU and the backup area:

M1164 state	D1064 code	Function
On	0x55AA	The program and the data in D2000~D9999 are read from the backup area, and copied to the main operation area.
	0x55A9	The program is read from the backup area, and copied to the main operation area in the CPU.
	0x55AB	The data is read from D2000~D11999 in the backup area, and copied to the main operation area in the CPU.
	0xAA55	The program and the data in D2000~D9999 in the main operation area are copied to the backup area.
	0xA955	The program in the main operation area is copied to the backup area.
On	0xAB55	The data in D2000~D11999 in the operation area is copied to the backup area.
	0x8888	The data in the backup area is cleared.

Note:

(1) After the reading/writing of the data is complete, M1164 is automatically set to Off. After the reading/writing of the data is complete, the following flags are On.
M1189 → The data which is read/written is correct.
M1075 → An error occurs when the data is written into the backup area. Before replacing the PLC, please read the program and the data form the main operation area first.
M1006 → No data or program is in the memory card.

(2) Enter the function code in D1064 first, and then set M1164 to On. Otherwise, M1164 is automatically reset to Off.

(3) If an error occurs during the reading/writing of the data, the special M will be set. No error LED indicator will flash or no situation in which the CPU can not run occurs. Therefore, if customers need an alarm, please make the warning message on the device or superior machine according to the flags above.

(4) Owing to the fact that the storage material of the backup area is Flash ROM, it takes some execution time to write the data into the memory card.

(5) When the program is copied, the CPU automatically copies the password function (including the main password, the limit on the number of errors, the subroutine password, and the PLC ID).

Function Group Operational Error Flag

Number M1067 ~ M1068, D1067 ~ D1068

Contents:

1. Operational error flag:

Device	Description	Latched	STOP → RUN	RUN → STOP
M1067	Operational error flag	None	Cleared	Latched
M1068	Operational error locked flag	None	Latched	Latched
D1067	Operational error code	None	Cleared	Latched
D1068	STEP value when operational error occurs	None	Latched	Latched

2. Error code explanation:

D1067 error code	Cause
H’ 0E18	BCD conversion error
H’ 0E19	Divisor is 0
H’ 0E1A	Use of device exceeds the range (including E, F index register modification)
H’ 0E1B	Square root value is negative
H’ 0E1C	FROM/TO instruction communication error

Function Group Low Voltage

Number M1087, D1100

Contents:

1. When PLC detects LV (Low Voltage) signal, it will check if M1087 is “On” or not. If M1087 is “On”, the content in D1100 will be stored in Y0 ~ Y17.
2. bit0 (LSB) of D1100 corresponds to Y0, bit1 corresponds to Y1, bit8 corresponds to Y10 and so on.

Function Group Power detector

Number M1019

Contents:

EH3/SV2 (V2.06 or later) supports this function.
The error LED keeps blinking or not, when the supplied 24V power is from NOT stable (power is lower than 17.8 V) to stable (for more than 2 seconds).
ON: The error LED stops blinking once the power is stable again.
OFF: The error LED keeps blinking even when the power is stable again.

Function Group File Register

Number M1101, D1101 ~ D1103

Contents:

1. When the power of PLC turns from “Off” to “On”, PLC determines whether to automatically send the content in the file register to the assigned data register by checking M1101, D1101 ~ D1103 (for SX/EH3/SV2).

M1101:	Whether to automatically downland data from file register
D1101:	Start No. of file register K0 ~ K1,599 (for SX) Start No. of file register K0 ~ K9,999 (for EH3/SV2)
D1102:	Number of data read from file register K1 ~ K1,600 (for SX) Number of data read from file register K1 ~ K8,000 (for EH3/SV2)
D1103:	Location for storing data read from file register Start No. of assigned data register D K2,000 ~ K4,999 (for SX) Start No. of assigned data register D K2,000 ~ K9,999 (for EH3/SV2)

2. See API 148 MEMR and API 149 MEMW for more details.

Function Group Detecting Extension

Number D1140, D1142, D1143, D1145

Contents:

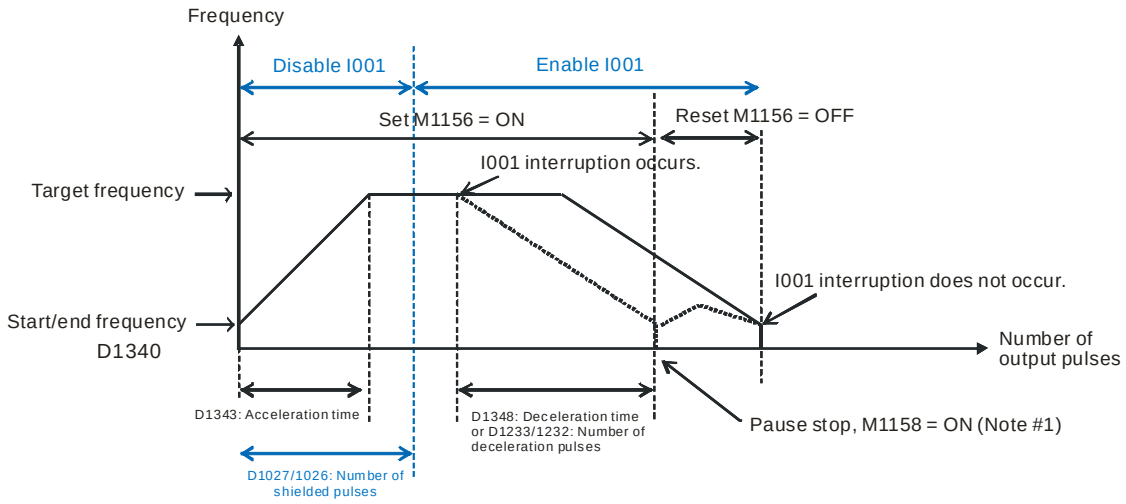
1. D1140: Number of special right-side extension modules (AD, DA, XA, PT, TC…); Max. 8
2. D1142: Number of X input points on digital extension device
3. D1143: Number of Y output points on digital extension device
4. D1145: Number of left-side special extension modules (max. 8) (only available in EH3-L, SV2)

Function Group Pulse Output Pause, Mask, Mark

Number M1156 ~ M1159, M1538 ~ M1541, D1026, D1027

Contents:

1. Actions of interruption type pulse output pause function (with deceleration):



Note: Actual line (▬) -> Action when I001 interruption does not occur.
Dotted line (▪▪▪▪) -> Action when I001 interruption occurs in unmasked area.

Note #1: After M1538 = ON and the user reset M1156 = OFF, PLC will complete the remaining number of output pulses automatically.

- Applicable instructions: DRVI/DDRVI/PLSR/DPLSR
- Usage restriction: Has to work with external interruptions, special M and special D.
- Other explanations:
  a) When this function is enabled, PLC will start to decelerate according to the set deceleration time. Even if the user does not set up the number of deceleration pulses (i.e. special D = 0), or the set number of pulses is less than the number planned for the deceleration time, PLC will still decelerate within the deceleration time. On the contrary, if the number of deceleration pulses is more than the planned number for the deceleration time, PLC will decelerate according to the number set in the special D.
  b) Range for deceleration time: 10 ~ 10,000ms
  c) There is mask interruption in CH0 high-speed output. When D1027/D1026 (32-bit) ≠ 0, the mask function will be enabled, i.e. X0 external interruption will not be activated when the number of output pulses is within the mask area.
- High-speed output CH0 ~ CH3 v.s. pause function of external input points X0 ~ X3:

Parameter / Channel	Interruption paused flag	External input point	Deceleration time Special D	Deceleration pulses Special D	Mask interruption function	Pause status flag
CH0 (Y0,Y1)	M1156	X0	D1348	D1232~D1233	D1026, D1027	M1538
CH1 (Y2,Y3)	M1157	X1	D1349	D1234~D1235	No	M1539
CH2 (Y4,Y5)	M1158	X2	D1350	D1236~D1237	No	M1540
CH3 (Y6,Y7)	M1159	X3	D1351	D1238~D1239	No	M1541

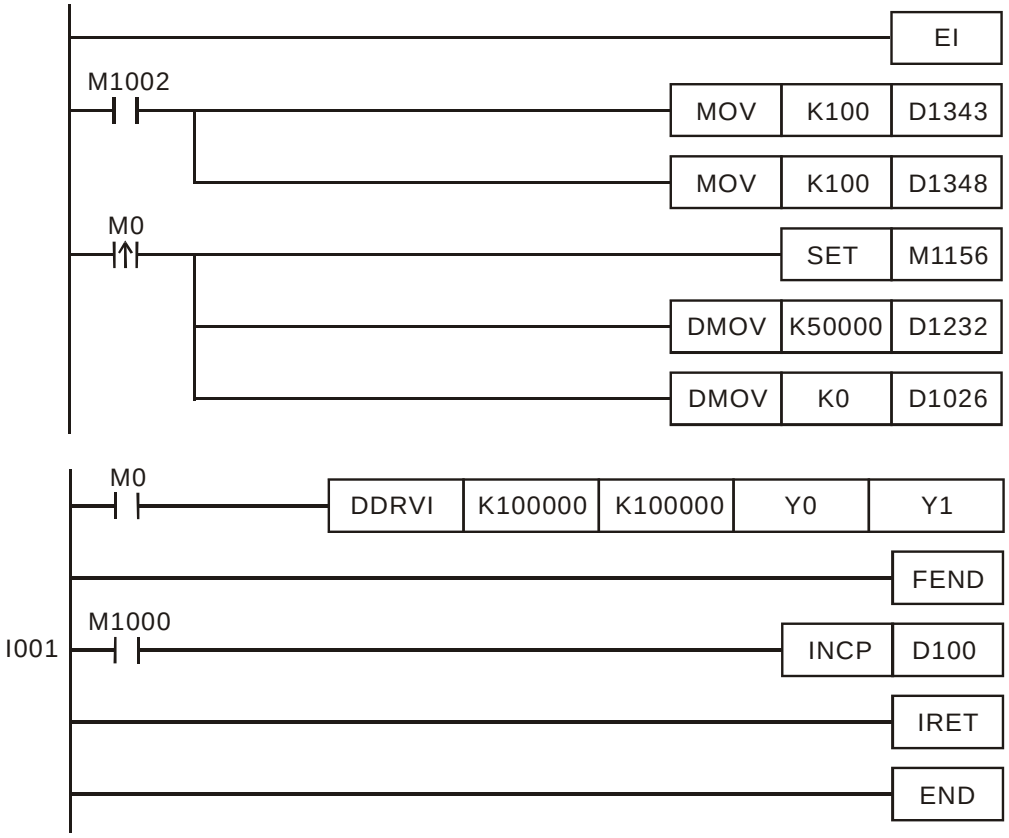
EH3/SV2

Parameter / Channel	Interruption paused flag	External input point	Deceleration time Special D	Deceleration pulses Special D	Mask interruption function	Pause status flag
CH0 (Y0,Y1)	M1156	X0	D1348	D1232~D1233	D1026, D1027	M1538
CH1 (Y2,Y3)	M1157	X1	D1349	D1234~D1235	D1135~ D1136	M1539
CH2 (Y4,Y5)	M1158	X2	D1350	D1236~D1237	D1154~ D1155	M1540
CH3 (Y6,Y7)	M1159	X3	D1351	D1238~D1239	No	M1541

- Application examples:
  - When M0 turns from Off to On, Y0 will start to output pulses and wait for the external input interruption X0 to take place. When interruption signals occur in the acceleration section or the highest speed section during the output, Y0 will immediately decelerate and stop the output after 100ms, and M1538 = On.
  - When M1538 = On, the user can reset (RST) M1156, and PLC will start to output the remaining pulses. When all the target pulses are completed, M1029 will be On.
  - If the external interruption occurs in the planned deceleration area, the output will not decelerate or set M1538 to On.

a) Example 1: Immediately decelerate and pause within deceleration time

- Application: When external interruptions occur, the high-speed output has to achieve deceleration and pause within the designated deceleration time. It is generally applied in the searching mark function in single-axis motion control.
- The program:

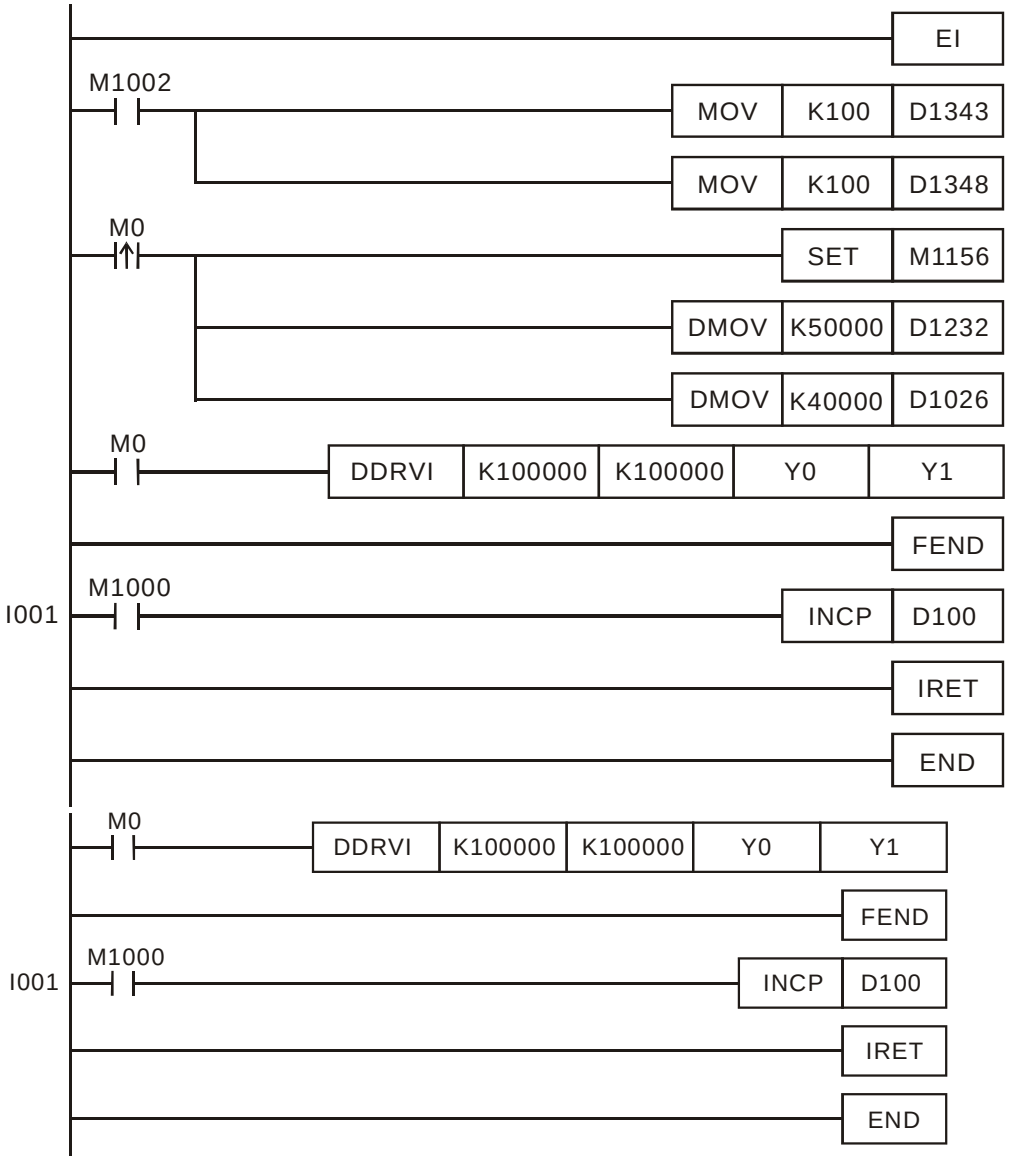


Program explanation:

- When M0 turns from Off to On, Y0 will start to output pulses and wait for the external input interruption X0 to take place. When interruption signals occur in the acceleration section or the highest speed section during the output, Y0 will immediately decelerate and stop the output after 100ms, and M1538 = On.
- When M1538 = On, the user can reset (RST) M1156, and PLC will start to output the remaining pulses. When all the target pulses are completed, M1029 will be On.
- If the external interruption occurs in the planned deceleration area, the output will not decelerate or set M1538 to On.

b) Example 2: Immediately decelerate and pause within the number of deceleration pulses

- Application: When external interruptions occur, the high-speed output has to achieve deceleration and pause within the designated number of deceleration pulses. It is generally applied in the searching mark function in single-axis motion control.
- The program:



Program explanation:

- When M0 turns from Off to On, Y0 will start to output pulses. After the external input interruption X0 occurs during the output, Y0 will immediately decelerate and output 50,000 pulses before it stops and set M1538 (pause status flag) to On.
- When M1538 = On, the user can reset (RST) M1156, and PLC will start to output the remaining pulses. When all the target pulses are completed, M1029 will be On.
- If the external interruption occurs in the planned deceleration area, the output will not decelerate or set M1538 to On.

2. Added the functions of activating mask flags (M1610 ~ M1613) and marking alignment. (available for EH3/SV2 V1.88 or later versions)

   - Special D:

     If the value in D1026/D1027, D1135/D1136, D1154/D1155 is equal to or less than 0, it means the masking function in the former section is disabled. On the other hand, if the value is larger than 0, the masking function in the former section is enabled.

     If the value is equal to or less than 0 in D1166 in the interruption area, it means the masking function in the later section is disabled. On the other hand, if the value is larger than 0, the masking function in the later section is enabled. Special D can be used for the masking function in the former section for outputting via CH0 ~ CH2 and in the later section for outputting via CH0 ~ CH3.

   - Special M:

     When any of the M1610 ~ M1613 is ON, the masking function is activated on its corresponding output point and no external marking or interrupt is allowed.

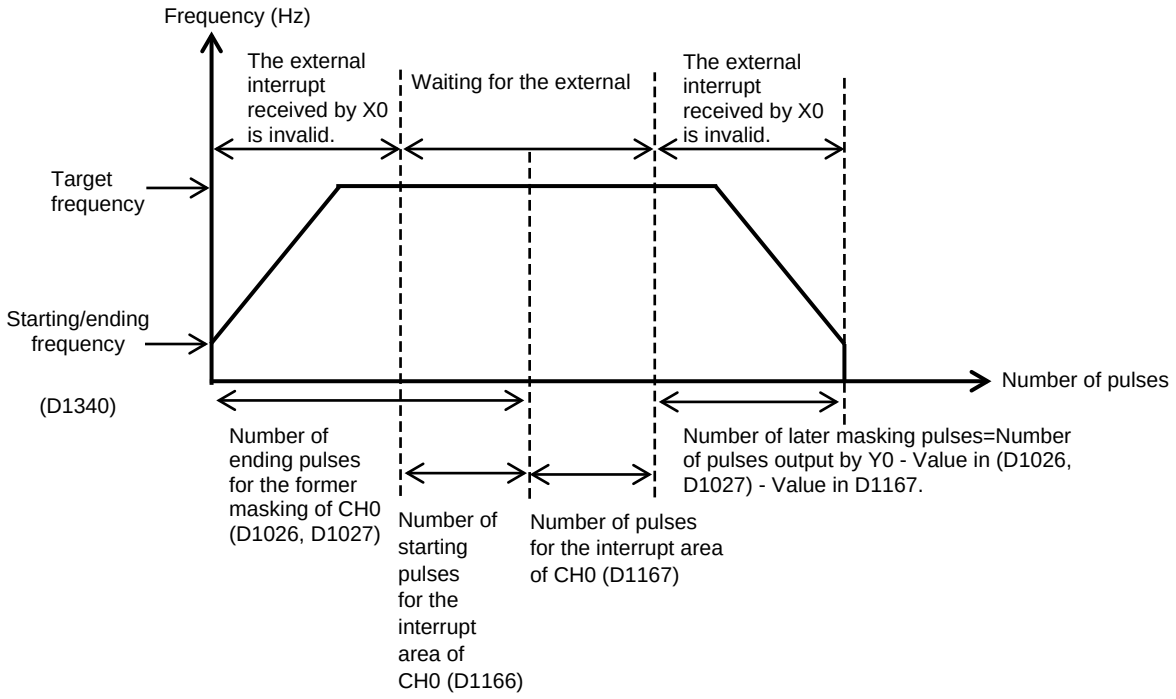
     When any of the M1610 ~ M1613 is OFF, the masking function is deactivated on its corresponding output point. The external marking or interrupt is allowed and to mask or not is determined by its corresponding special D. Speial M can be used for masking function for outputting via CH0~CH3.

   - 24SV2 doesnot support D1026/D1027 and D1135/D1136.

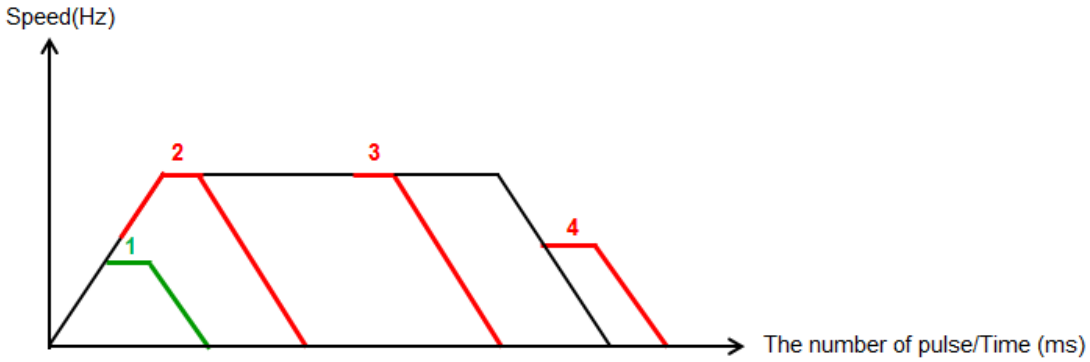
Output number	Marking and decelerating flag	Enable mask funciton	External input points	Acelceration time	Deceleration time	Start/end frequency	Deceleration pulses	Masked former section	Interrupt area
CH0 (Y0/Y1)	M1156	M1610	X0	D1343	D1348	D1340	D1232/ D1233	D1026/ D1027	D1166/ D1167
CH1 (Y2/Y3)	M1157	M1611	X1	D1353	D1349	D1352	D1234/ D1235	D1135/ D1136	NA
CH2 (Y4/Y5)	M1158	M1612	X2	D1381	D1350	D1379	D1236/ D1237	D1154/ D1155	NA
CH3 (Y6/Y7)	M1159	M1613	X3	D1382	D1351	D1380	D1238/ D1239	NA	NA

Masked actions for CH0 in the former and later sections.

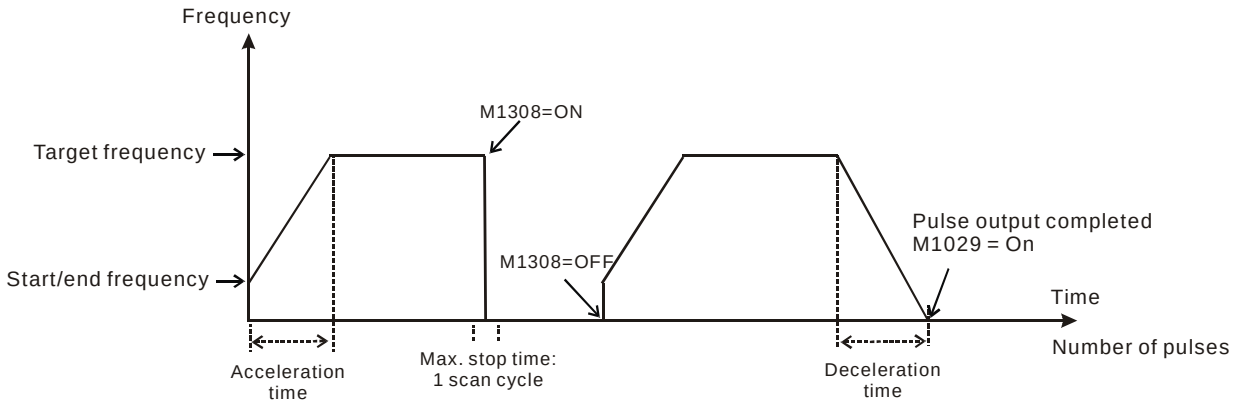
- The number of ending pulses for the front masking of CH0 includes the number of starting pulses for the interrupt area of CH0. The value in D1167 indicates the number of pulses for the interrupt area of CH0. The area in which an interrupt is allowed to occur is in the range of (D1026, D1027)-D1166 to (D1026, D1027)+D1167.
- Example: The value in (D1026, D1027) is 250000, the value in D1166 is 30000, and the value in D1167 is 30000. The area in which an interrupts is allowed to occur is in the range of 220001 to 279999. If the number of pulses output by Y0 is less than or equal to 220000, or greater than or equal to 280000, no interrupt will be valid.
- The number of later masking pulses is equal to the number of pulses output by Y0 minus the value in (D1026, D1027) minus the value in D1167.



The marking function can be used in acceleration, full speed and deceleration setions. See the image below for reference.



3. Actions of program type pulse output pause function (with no deceleration):



- Applicable model/firmware version: EH3/SV2 (V1.4 or later)
- Applicable instructions: DRVI/DDRVI/DRVA/DDRVA/PLSR/DPLSR

- During the pulse output, force On M1308 will stop the output, and force Off M1308 will start the output of remaining pulses.
- The max. stop time inaccuracy in this pause function is 1 scan cycle.
- High-speed output CH0 ~ CH3 v.s. pause function of pause flags:

Flag / Channel	Pause flag
CH0	M1308
CH1	M1309
CH2	M1310
CH3	M1311

4. Special M and special D registers for SV V1.4 and later versions when conducting deceleration to pausing output (for the ongoing high-speed pulse output encountering interruption signals), mask and mark.

Device No.	Function	
M1308	Off -> On: 1st group of CH0 (Y0, Y1) high-speed pulse output immediately pauses. On -> Off: Complete the remaining number of output pulses	
M1309	Off -> On: 2nd group of CH1 (Y2, Y3) high-speed pulse output immediately pauses. On -> Off: Complete the remaining number of output pulses	
M1310	Off -> On: 3rd group of CH2 (Y4, Y5) high-speed pulse output immediately pauses. On -> Off: Complete the remaining number of output pulses	
M1311	Off -> On: 4th group of CH3 (Y6, Y7) high-speed pulse output immediately pauses. On -> Off: Complete the remaining number of output pulses	
M1156	Enable X0 interruption to trigger immediate decelerating and pausing CH0 high-speed output. (When M1156 is enabled and M1538 = On, simply clear M1156 to finish sending out the remaining output pulses.)	
M1157	Enable X1 interruption to trigger immediate decelerating and pausing CH1 high-speed output	
M1158	Enable X2 interruption to trigger immediate decelerating and pausing CH2 high-speed output	
M1159	Enable X3 interruption to trigger immediate decelerating and pausing CH3 high-speed output	
M1538	CH0 pause status	
M1539	CH1 pause status	
M1540	CH2 pause status	
M1541	CH3 pause status	
D1026	When ≠ 0, enabling DRVI and PLSR instructions to work with X0 (mask) interruption.	Low word
D1027		High word
D1232	Number of CH0 output pulses after mark	Low word
D1233		High word
D1234	Number of CH1 output pulses after mark	Low word
D1235		High word
D1236	Number of CH2 output pulses after mark	Low word
D1237		High word
D1238	Number of CH3 output pulses after mark	Low word
D1239		High word

Function Group Fixed Slop Funciton

Number M1604 ~ M1607, D1410 ~ D1413, D1988 ~ D1991

Contents:

- Applicable model/firmware version: EH3/SV2 (V1.88 or later)
- Applicable instructions: DRVI/DDRVI/PLSR/DPLSR

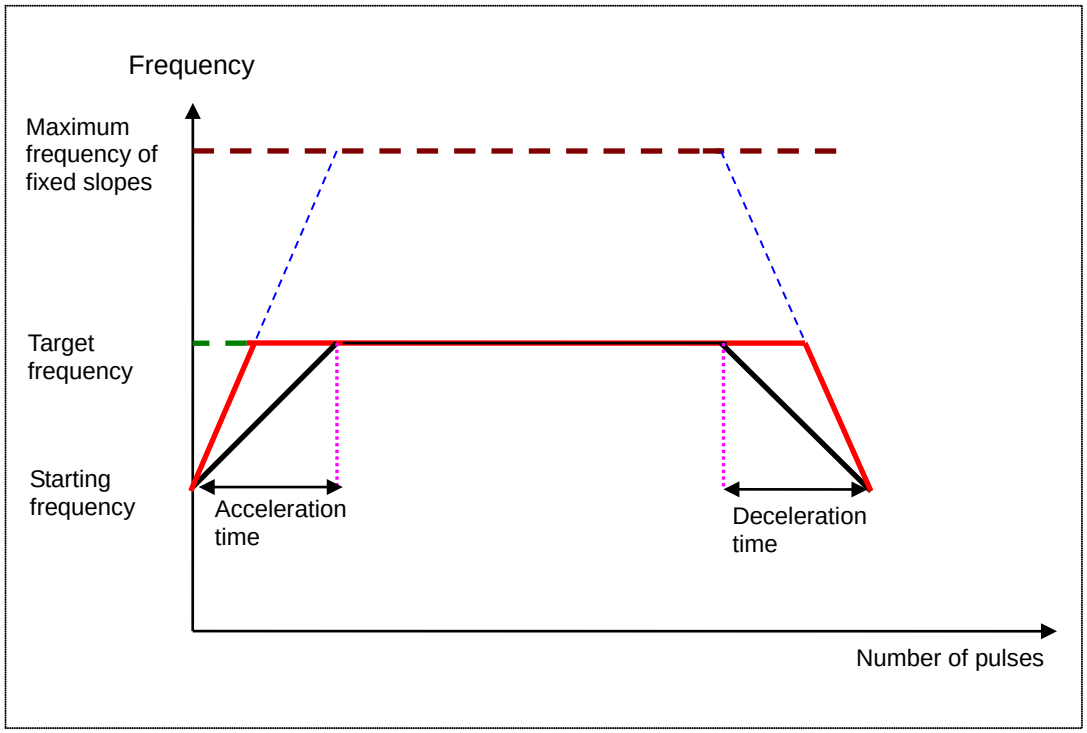
The channels CH0~CH3 support the fixed slope function. M1604~M1607 are used to enable fixed acceleration/deceleration slopes for CH0~CH3. D1410~D1413 and D1988~D1991 are used to set the maximum frequencies of the fixed acceleration/deceleration slopes for CH0~CH3.

Outputs	Fixed slope flag special M device	Maximum frequencies special D device
CH0 (Y0/Y1)	M1604	D1410/D1411
CH1 (Y2/Y3)	M1605	D1412/D1413
CH2 (Y4/Y5)	M1606	D1988/D1989
CH3 (Y6/Y7)	M1607	D1990/D1991

A general acceleration/deceleration slope is determined by a starting frequency, a closing frequency, a target frequency, acceleration time, and deceleration time. Please see the back line below for more information about acceleration and deceleration. If the starting frequency and the acceleration/deceleration time are fixed, the black slopes will vary with the target frequency.

A fixed acceleration/deceleration slope is determined by a starting frequency, a closing frequency, a maximum frequency, acceleration time, and deceleration time. Please see the red line below for more information about acceleration and deceleration. If the target frequency is changed, the acceleration slope and the deceleration slope will not be changed.

The function can be used to drive step motors in that it can prevent the adjustment of a target frequency from causing a step motor to stall.



Function Group Single Step Execution

Number M1170, M1171, D1170

Contents:

1. Special D and special M for single step execution for EH3/SV2:

Device No.	Function
M1170	Start flag
M1171	Action flag
D1170	STEP No. of the currently executed instruction

2. The function:
   a) Execution timing: The flag is valid only when PLC is in RUN status.
   b) Action steps:
      i) When M1170 is enabled, PLC enters the single step execution mode. PLC stays at a specific instruction, stores the location of STEP in D1170 and executes the instruction once.
      ii) When M1171 is forced “On”, PLC executes the next instruction and stops. At the same time, PLC auto-force “O ff” M1171 and stops at the next instruction. D1170 stores the present STEP value.
      iii) When Y output is in single step execution mode, Y outputs immediately without having to wait until END instruction is being executed.
3. Note:
   a) Instruction that will be affected by scan time will be executed incorrectly due to the single step execution. For example, when HKY instruction is executed, it takes 8 scan times to obtain a valid input value from a key. Therefore, the single step execution will result in incorrect actions.
   b) High-speed pulse input/output and high-speed counter comparison instructions are executed by hardware; therefore, they will not be affected by the single step execution.

Function Group VR Volume

Number M1178 ~ M1179, D1178 ~ D1179

Contents:

1. Special D and special M for built-in 2-point VR volume for EH3/SV2 and EC/EC3-8K:

Device No.	Function
M1178	Enable VR0 volume
M1179	Enable VR1 volume
D1178	VR0 value
D1179	VR1 value

2. The function:

   This function should be used when PLC is in RUN status. When M1178 = On, VR0 value will be converted into a value of 0 ~ 255 and stored in D1178. When M1179 = On, VR1 value will be converted into a value of 0 ~ 255 and stored in D1179.

3. See API 85 VRRD for more details.

Function Group Interruption Instruction for Reading the Number of Pulses

Number M1181 ~ M1182, D1180 ~ D1181, D1198 ~ D1199

Contents:

1. SX can use external interruption to store the present value in the middle-high-speed counter into D1180 ~ D1181 and D1198 ~ D1199, and use M1181 ~ M1182 to clear the present value in the high-speed counter.
2. The function:

a) X0 (pulse input point) has to work with X2 (external interruption point), and C235/C251/C253 (high-speed counter) has to work with I201 (interruption No.). D1180 and D1181 are the registers to store the 32-bit values. If M1181 is enabled before the interrupt is triggered, the value in C235/C251/C253 is moved to D1180 and D1181 when the interrupt is triggered, and the value in C235/C251/C253 will be cleared.
Condition: When the program enables I201 (X2 is the external interruption input), and C235, C251, and C253 are used, the function is enabled.

b) X1 (pulse input point) has to work with X3 (external interruption point), and C236 has to work with I301. D1198 and D1199 are the registers to store the 32-bit values. If M1182 is enabled before the interrupt is triggered, the value in C236 is moved to D1198 and D1199 when the interrupt is triggered, and the value in C236 will be cleared.
Condition: When the program enables I301 (X3 is the external interruption input), and C236 is used, the function is enabled.

Function Group Auto-mapping Function

Number M1182 ~ M1183, D9800 ~ D9879, D9900~D9979

1. The default value of M1182 is ON.
When M1182 is OFF, the auto-mapping function is enabled. The analog-to-digital values/digital-to-analog values correspond to D9800~D9879. If the first left-side module connected to EH3-L/SV2 is a communication module, the analog-to-digital values/digital-to-analog values correspond to D9810~. For example, if the modules connected to SV2 from left to right are 04DA-SL, EN01-SL, and 04AD-SL, and M1182 is OFF, D9820~D9823 will be assigned to CH1~CH4 in the third left-side module 04DA-SL.

04DA-SL	**EN01-SL**	**04AD-SL**	**SV2**
↓	↓	↓	

Third left-side module	Second left-side module	First left-side module	
D9820	X	D9800	CH1 AIO conversion value
D9821	X	D9801	CH2 AIO conversion value
D9822	X	D9802	CH3 AIO conversion value
D9823	X	D9803	CH4 AIO conversion value

Note: The default value of M1182 in SV2 version 1.0 is OFF. If users want to disable the auto-mapping function, they have to set M1182 to ON.

2. The default value of M1183 is ON.

When M1183 is OFF, the auto-mapping function is enabled. The analog-to-digital values/digital-to-analog values correspond to D9900~D9979. For example, if the modules connected to SV2 from left to right are 04DA-S, 04DA-S, and 06XA-S, and M1182 is OFF, D9900~D9903 will be assigned to CH1~CH4 in the first right-side module 04DA-S, D9910~D9913 will be assigned to CH1~CH4 in the second right-side module 04DA-S, and D9920~D9925 will be assigned to CH1~CH6 in the third module 06XA-S.

SV2	04DA-S	04AD-S	06XA-S
	↓	↓	↓

	First module	Second module	Third module
CH1 AIO conversion value	D9900	D9910	D9920
CH2 AIO conversion value	D9901	D9911	D9921
CH3 AIO conversion value	D9902	D9912	D9922
CH4 AIO conversion value	D9903	D9913	D9923
CH5 AIO conversion value	X	X	D9924
CH6 AIO conversion value	X	X	D9925

Function Group MODEM Connection Function

Number M1184 ~ M1188

Contents:

1. The system connection

2. Special M for MODEM connection for EH3/SV2:

Device No.	Function	Note
M1184	Enable MODEM	On: The following actions are valid
M1185	Initialize MODEM	Off: Initialization is completed
M1186	Fail to initialize MODEM	Off: M1185 = On
M1187	MODEM initialization is completed	Off: M1185 = On
M1188	Shows if MODEM is connected	On: Connecting

Note: The special Ms are both applicable when PLC is in RUN or STOP status.

3. How to connect (Please follow the steps below):

a) Set “On” M1184 (Enable PLC MODEM connection).

b) Set “On” M1185 (Enable initialization of MODEM from PLC).

c) Check if the initialization of MODEM is successful from M1186, M1187.

d) Wait for the connection.

4. Note:
   a) When PLC is to be connected with MODEM, a RS-232 extension card is required. If there is no RS-232 extension card, all special M above will be invalid.
   b) After enabling MODEM (M1184 = On), PLC has to initialize MODEM first (M1185 = On). If PLC fails to initialize MODEM, the auto-answering function of the MODEM will not be enabled.
   c) After MODEM is initialized, it will enter auto-answering mode automatically.
   d) If the remote PC is disconnected, MODEM will enter stand-by mode automatically and if the user turns off MODEM now, MODEM will have to be initialized again when it is turned on again.
   e) The connection speed is set by PLC as 9,600bps fixed and modification on the speed is not allowed. MODEM has to be able to support the speed of 9,600bps and versions above.
   f) The initialization format from PLC to MODEM are ATZ and ATS0 = 1.
   g) If PLC fails to initialize MODEM, use the super terminal in PC to initialize it by the format ATZ and ATS0 = 1.

Function Group Latched Area

Number D1200 ~ D1219

Contents:

1. The latched area for EH3/SV2 and SX is from the start address No. to the end address No.
2. See the tables in Chapter 2.1 for more details.

Function Group Set On/Off of Input Point X on CPU

Number M1304

Contents:

1. For ES/EX/EC/EC3-8K, when M1304 = On, the X input points (X0 ~ X17) on CPU can be set On/Off by peripheral devices, e.g. WPLSoft or DVP-HPP. However, the LED indicators will not respond to the setup.
2. For SX, when M1304 = On, peripheral devices, e.g. WPLSoft, can set On/Off of X0 ~ X17 on the CPU, but the LED indicators will not respond to it.
3. For EH3/SV2, when M1304 = On, peripheral devices, e.g. WPLSoft or DVP-HPP, can set On/Off of X input points on the CPU, but the LED indicators will not respond to it.

Function Group Right-Side Special Extension Module ID

Number D1320 ~ D1327

Contents:

1. The ID of right-side special extension module, if any, connected to EH3/SV2 are stored in D1320 ~ D1327 in sequence.
2. Special extension module ID for H:

Module Name	Module ID (hex)	Module Name	Module ID (hex)
DVP04AD-H	H'0400	DVP01PU-H	H'0110
DVP04DA-H	H'0401	DVP01HC-H	H'0120
DVP04PT-H	H'0402	DVP02HC-H	H'0220
DVP04TC-H	H'0403	DVP01DT-H	H'0130
DVP06XA-H	H'0604	DVP02DT-H	H'0230

3. Special extension module ID for H2 (EH3 can be connected to the special extension module of H2):

Module Name	Module ID (hex)	Module Name	Module ID (hex)
DVP04AD-H2	H'6400	DVP01HC-H2	H'6120
DVP04DA-H2	H'6401	DVP02HC-H2	H'6220
DVP04PT-H2	H'6402	DVPDT02-H2	H'0230
DVP04TC-H2	H'6403	DVPCP02-H2	H'0240
DVP06XA-H2	H'6604	DVPPF02-H2	H'0250
DVP01PU-H2	H'6110		

4. Special extension module ID for H3:

Module Name	Module ID (hex)
DVP04AD-H3	H'6407
DVP06XA-H3	H'6608
DVP04DA-H3	H'6409

Function Group Left-Side High-Speed Special Extension Module ID

Number D1386 ~ D1393

Contents:

1. The ID of left-side special extension module, if any, connected to SV2/EH2-L are stored in D1386 ~ D1393 in sequence.
2. Left-side special extension module ID for SV2/EH2-L:

Module Name	Module ID (hex)	Module Name	Module ID (hex)
DVP04AD-SL	H'4400	DVP01HC-SL	H'4120
DVP04DA-SL	H'4401	DVP02HC-SL	H'4220
DVP04PT-SL	H'4402	DVPDNET-SL	H'4131
DVP04TC-SL	H'4403	DVPEN01-SL	H'4050
DVP06XA-SL	H'6404	DVPMDM-SL	H'4040
DVP01PU-SL	H'4110	DVPCOPM-SL	H'4133

Function Group PLC LINK

Number M1350 ~ M1354, M1360 ~ M1519, D1399, D1355 ~ D1370, D1415 ~ D1465, D1480 ~ D1991

Contents:

1. Special D and special M for ID1 ~ ID8 of the 16 stations in PLC LINK (M1353 = Off) for SX/EH3/SV2:

MASTER PLC															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353 = Off: Disable 32 stations in the Link and the function of reading/writing more than 16 data (RST M1353); the No. of special D for storing the 16 read/written data.															
D1480 | D1495	D1496 | D1511	D1512 | D1527	D1528 | D1543	D1544 | D1559	D1560 | D1575	D1576 | D1591	D1592 | D1607	D1608 | D1623	D1624 | D1639	D1640 | D1655	D1656 | D1671	D1672 | D1687	D1688 | D1703	D1704 | D1719	D1720 | D1735
Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
Start Communication Address															
D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
LINK in SLAVE PLC?															
M1360		M1361		M1362		M1363		M1364		M1365		M1366		M1367	
Action flag for SLAVE PLC from MASTER PLC															
M1376		M1377		M1378		M1379		M1380		M1381		M1382		M1383	
"Read/write error" flag															
M1392		M1393		M1394		M1395		M1396		M1397		M1398		M1399	
"Reading completed" flag (turns "Off" whenever read/write a station is completed)															
M1408		M1409		M1410		M1411		M1412		M1413		M1414		M1415	
"Writing completed" flag (turns "Off" whenever read/write a station is completed)															
M1424		M1425		M1426		M1427		M1428		M1429		M1430		M1431	

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

- Default start communication address D1355 ~ D1362 to be read = H1064 (D100)
- Default start communication address D1415 ~ D1422 to be written = H10C8 (D200)

2. Special D and special M for ID9 ~ ID16 of the 16 stations in PLC LINK (M1353 = Off) for SX/EH3/SV2:

MASTER PLC															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353 = Off: Disable 32 stations in the Link and the function of reading/writing more than 16 data (RST M1353); the No. of special D for storing the 16 read/written data.															
D1736 | D1751	D1752 | D1767	D1768 | D1783	D1784 | D1799	D1800 | D1815	D1816 | D1831	D1832 | D1847	D1848 | D1863	D1864 | D1879	D1880 | D1895	D1896 | D1911	D1912 | D1927	D1928 | D1943	D1944 | D1959	D1960 | D1975	D1976 | D1991
Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
Start Communication Address															
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430
LINK in SLAVE PLC?															
M1368		M1369		M1370		M1371		M1372		M1373		M1374		M1375	
Action flag for SLAVE PLC from MASTER PLC															
M1384		M1385		M1386		M1387		M1388		M1389		M1390		M1391	
“Read/write error” flag															
M1400		M1401		M1402		M1403		M1404		M1405		M1406		M1407	
“Reading completed” flag (turns “Off” whenever read/write a station is completed)															
M1416		M1417		M1418		M1419		M1420		M1421		M1422		M1423	
“Writing completed” flag (turns “Off” whenever read/write a station is completed)															
M1432		M1433		M1434		M1435		M1436		M1437		M1438		M1439	

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

- Default start communication address D1363 ~ D1370 to be read = H1064 (D100)
- Default start communication address D1423 ~ D1430 to be written = H10C8 (D200)

3. Special D and special M for ID1 ~ ID8 of the 32 stations in PLC LINK (M1353 = On) for EH3/SV2:

MASTER PLC															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353 = On: Enable 32 stations in the Link and the function of reading/writing more than 16 data (SET M1353); the No. of D registers for storing the read/written data.															
D1480	D1496	D1481	D1497	D1482	D1498	D1483	D1499	D1484	D1500	D1485	D1501	D1486	D1502	D1487	D1503
If M1356 is ON, users can set the station numbers of slave ID1~ID8 in D1900~D1907. The master station sends commands according to the station numbers set.															
D1900		D1901		D1902		D1903		D1904		D1905		D1906		D1907	
Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
Start Communication Address															
D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
LINK in SLAVE PLC?															
M1360		M1361		M1362		M1363		M1364		M1365		M1366		M1367	
Action flag for SLAVE PLC from MASTER PLC															
M1376		M1377		M1378		M1379		M1380		M1381		M1382		M1383	
“Read/write error” flag															
M1392		M1393		M1394		M1395		M1396		M1397		M1398		M1399	
“Reading completed” flag (turns “Off” whenever read/write a station is completed)															
M1408		M1409		M1410		M1411		M1412		M1413		M1414		M1415	
“Writing completed” flag (turns “Off” whenever read/write a station is completed)															
M1424		M1425		M1426		M1427		M1428		M1429		M1430		M1431	

SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

- Default start communication address D1355 ~ D1362 to be read = H1064 (D100)
- Default start communication address D1415 ~ D1422 to be written = H10C8 (D200)

4. Special D and special M for ID9 ~ ID16 of the 32 stations in PLC LINK (M1353 = On) for EH3/SV2:

MASTER PLC															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read Out	Write in	Read out	Write in	Read out	Write in
M1353 = On: Enable 32 stations in the Link and the function of reading/writing more than 16 data (SET M1353); the No. of D registers for storing the read/written data.															
D1488	D1504	D1489	D1505	D1490	D1506	D1491	D1507	D1492	D1508	D1493	D1509	D1494	D1510	D1495	D1511
If M1356 is ON, users can set the station numbers of slave ID9~ID16 in D1908~D1915. The master station sends commands according to the station numbers set.															
D1908		D1909		D1910		D1911		D1912		D1913		D1914		D1915	
Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430

LINK in SLAVE PLC?							
M1368	M1369	M1370	M1371	M1372	M1373	M1374	M1375
Action flag for SLAVE PLC from MASTER PLC							
M1384	M1385	M1386	M1387	M1388	M1389	M1390	M1391
“Read/write error” flag							
M1400	M1401	M1402	M1403	M1404	M1405	M1406	M1407
“Reading completed” flag (turns “Off” whenever read/write a station is completed)							
M1416	M1417	M1418	M1419	M1420	M1421	M1422	M1423
“Writing completed” flag (turns “Off” whenever read/write a station is completed)							
M1432	M1433	M1434	M1435	M1436	M1437	M1438	M1439

SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

- Default start communication address D1363 ~ D1370 to be read = H1064 (D100)
- Default start communication address D1423 ~ D1430 to be written = H10C8 (D200)

5. Special D and special M for ID17 ~ ID24 of the 32 stations in PLC LINK (M1353 = On) for EH3/SV2:

MASTER PLC															
SLAVE ID 17		SLAVE ID 18		SLAVE ID 19		SLAVE ID 20		SLAVE ID 21		SLAVE ID 22		SLAVE ID 23		SLAVE ID 24	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353 = On: Enable 32 stations in the Link and the function of reading/writing more than 16 data (SET M1353); the No. of D registers for storing the read/written data.															
D1576	D1592	D1577	D1593	D1578	D1594	D1579	D1595	D1580	D1596	D1581	D1597	D1582	D1598	D1583	D1599
If M1356 is ON, users can set the station numbers of slave ID17~ID24 in D1916~D1923. The master station sends commands according to the station numbers set.															
D1916		D1917		D1918		D1919		D1920		D1921		D1922		D1923	
Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data
D1544	D1560	D1545	D1561	D1546	D1562	D1547	D1563	D1548	D1564	D1549	D1565	D1550	D1566	D1551	D1567
start Communication Address															
D1512	D1528	D1513	D1529	D1514	D1530	D1515	D1531	D1516	D1532	D1517	D1533	D1518	D1534	D1519	D1535

LINK in SLAVE PLC?							
M1440	M1441	M1442	M1443	M1444	M1445	M1446	M1447
Action flag for SLAVE PLC from MASTER PLC							
M1456	M1457	M1458	M1459	M1460	M1461	M1462	M1463
“Read/write error” flag							
M1472	M1473	M1474	M1475	M1476	M1477	M1478	M1479
“Reading completed” flag (turns “Off” whenever read/write a station is completed)							
M1488	M1489	M1490	M1491	M1492	M1493	M1494	M1495
“Writing completed” flag (turns “Off” whenever read/write a station is completed)							
M1504	M1505	M1506	M1507	M1508	M1509	M1510	M1511

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

SLAVE ID 17		SLAVE ID 18		SLAVE ID 29		SLAVE ID 20		SLAVE ID 21		SLAVE ID 22		SLAVE ID 23		SLAVE ID 24	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

- Default start communication address D1512 ~ D1519 to be read = H1064 (D100)
- Default start communication address D1528 ~ D1535 to be written = H10C8 (D200)

6. Special D and special M for ID25 ~ ID32 of the 32 stations in PLC LINK (M1353 = On) for EH3/SV2:

MASTER PLC															
SLAVE ID 25		SLAVE ID 26		SLAVE ID 27		SLAVE ID 28		SLAVE ID 29		SLAVE ID 30		SLAVE ID 31		SLAVE ID 32	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
M1353 = On: Enable 32 stations in the Link and the function of reading/writing more than 16 data (SET M1353); the No. of D registers for storing the read/written data.															
D1584	D1600	D1585	D1601	D1586	D1602	D1587	D1603	D1588	D1604	D1589	D1605	D1590	D1606	D1591	D1607
If M1356 is ON, users can set the station numbers of slave ID25~ID32 in D1924~D1931. The master station sends commands according to the station numbers set.															
D1924		D1925		D1926		D1927		D1928		D1929		D1930		D1931	
Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data	Number of data
D1552	D1568	D1553	D1569	D1554	D1570	D1555	D1571	D1556	D1572	D1557	D1573	D1558	D1574	D1559	D1575
Start Communication Address															
D1520	D1536	D1521	D1537	D1522	D1538	D1523	D1539	D1524	D1540	D1525	D1541	D1526	D1542	D1527	D1543
LINK in SLAVE PLC?															
M1448		M1449		M1450		M1451		M1452		M1453		M1454		M1455	
Action flag for SLAVE PLC from MASTER PLC															
M1464		M1465		M1466		M1467		M1468		M1469		M1470		M1471	
“Read/write” error flag															
M1480		M1481		M1482		M1483		M1484		M1485		M1486		M1487	
“Reading completed” flag (turns “Off” whenever read/write a station is completed)															
M1496		M1497		M1498		M1499		M1500		M1501		M1502		M1503	
“Writing completed” flag (turns “Off” whenever read/write a station is completed)															
M1512		M1513		M1514		M1515		M1516		M1517		M1518		M1519	

SLAVE ID 25		SLAVE ID 26		SLAVE ID 27		SLAVE ID 28		SLAVE ID 29		SLAVE ID 30		SLAVE ID 31		SLAVE ID 32	
Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in	Read out	Write in
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

- Default start communication address D1520 ~ D1527 to be read = H1064 (D100)
- Default start communication address D1536 ~ D1543 to be written = H10C8 (D200)

7. Note:
   a) PLC LINK is based on Modbus communication protocol.
   b) EH3/SV2 supports 32 stations in the LINK and reading/writing of more than 16 data (SET1353) (M1353 = On). SX supports 16 devices in the LINK and reading/writing of 16 data.
   c) EH3/SV2: When a MASTER PLC and a Slave PLC is connected, they are able to read/write maximum 100 WORD data (M1353 = On). SX/EC3-8K: Does not support M1353. When a Master PLC and a Slave PLC is connected, they are able to read/write maximum 16 WORD data.
   d) When the Master PLC is connected through COM2 (RS-485), baud rates and communication formats of all Slave PLCs must be the same (set in D1120). When SX/EH3/SV2 serves as Master, it supports ASCII and RTU format.
   e) When the Slave PLC is connected through COM2 (RS-232/RS-485/RS-422), baud rates and communication formats of all connected Slave PLCs must be the same as those in the Master PLC (set in D1120). When SX/EH3/SV2 serves as Slave, it supports ASCII and RTU format.
   f) When the Slave PLC is connected through COM1 (RS-232), baud rates and communication formats of all connected Slave PLCs must be the same as those in the Master PLC (set in D1036). When SX/EH3/SV2 serves as Slave, it supports ASCII and RTU format.
   g) When the Slave PLC is connected through COM3 (RS-232/RS-485), baud rates and communication formats of all connected Slave PLCs must be the same as those in the Master PLC (set in D1109). When SX/EH3/SV serves as Slave, it only supports ASCII format (Max. baud rate = 38,400bps).
   h) The start station No. (K1 ~ K214) of Slave ID1 is assigned by D1399 of Master PLC. Station No. of every Slave and Master PLC can not be the same (set in D1121).
   i) For one-to-one LINK: Connected through RS-232, RS-485, RS-422. PLC COM1, COM2, COM3 support many communication formats. For one-to-many LINK: Connected through RS-485. PLC COM1, COM2, COM3 support many communication formats.
8. How to operate PLC LINK:
   a) Set up the baud rates and communication formats of Master PLC and all connected Slave PLCs and make them the same. COM1_RS-232: D1136; COM2_RS-232/RS-485/RS-422: D1120; COM3_RS-232/RS-485: D1109.
   b) Set up the station No. of Master PLC (in D1121) and assign the start station No. of Slave PLC from D1399 of Master PLC. Next, set up the station No. of Slave PLC. Station No. of Master and Slave cannot be the same.
   c) Set up the number of connected Slave stations and the number of data to be read in/written to Slave stations. For EH3/SV2 (M1353 = On): Enable the function of the 32 conncected Slaves and reading/writing of more than 16 data (Max. 100 data). Next, set up the No. of D registers for storing the read data (D1480 ~ D1495, D1576 ~ D1591) and written data (D1496 ~ D1511, D1592 ~ D1607) (See the explanations above on special D). SX only supports reading/writing of 16 data.
   d) Set up the length of data to be read from/written into the Slave. (If the user does not set up set them up, PLC will follow the initial setting or the setting set in the previous operation.) (See the explanations above on special D.)

e) Set up the start communication address of the Slave to be read/written. (See the explanations above on special D). The default start communication address of Slave to be read: H1064 = D100. The default start communication address of Slave to be written: H10C8 = D200.

f) Operation procedure:

i) Enable the function of more than 32 stations connected to PLC LINK and reading/writing of 16 data (Max. 100 data) (M1353).

ii) Enable reading/writing of PLC LINK in the same polling (M1354).

iii) Set up PLC LINK as auto mode (M1351), or manual mode (M1352) and the times of polling (D1431).

iv) When M1355 = On, M1360 ~ M1375 (M1440 ~ M1455) will be the flags for the PLC designated to be connected to. When M1355 = Off, there will be detection on the slaves connected, and M1360 ~ M1375 (M1440 ~ M1455) will become the flags for the existence of connected PLC.

v) Enable PLC LINK (M1350).

9. How does Master PLC work:

a) To detect station No. of Slave: Set up the LINK to be automatic mode (M1351 = On) or manual mode (M1352 = On) (Note: M1351 and M1352 cannot be “On” at the same time) When M1350 = On, Master PLC starts to detect the total number of Slave stations connected to the LINK and records the number in D1433. The detection time may differ upon the number of Slave stations and the setting of communication timeout in D1129.

b) When M1360 ~ M1375, M1440 ~ M1455 = On, Slave ID1 ~ ID32 exist.

c) If the detected number Slave stations is 0, M1350 will be “Off” and the LINK will be stopped.

d) M1353 and M1354 should be set before PLC LINK is enabled. When PLC LINK is executed, the two special M will not afftec the action of PLC LINK.

e) When M1355 = On, M1360 ~ M1375 (M1440 ~ M1455) will be the flags for the PLC designated to be connected to. Therefore, no matter how many PLCs are actually connected, the PLC LINK function will continue to send read/write instruction in cycle according to the designated flags.

f) When 1353 = On, D1480 ~ D1607 (read/write buffer of SLAVE ID1 ~ 4) will be used for storing the No. of registers D for Slave ID1 ~ ID32, The No. of register D shall not be “> 9,900” or “< 0” ; otherwise, PLC will auto-set it as 9,900. The data length shall not be “ > 100” or “<0”; otherwise, PLC will auto-set it as 100.

g) When M1354 = On, set up Modbus Function H17 (synchronous read/write) for PLC LINK communication. If the number of written data is set to be “0”, the communication of PLC LINK will be automatically converted into Modbus Function H03 (read multiple words). Similarly, if the number of read data is set to be “0”, the communication of PLC LINK will be converted into Modbus Function H06 (write 1 word) or Modbus Function H10 (write multiple words).

h) If the number of read/written data > 16, the time for timeout (D1129) shall be > 500ms in case communication timeout may occur.

i) After M1350 = On, Master PLC detects the ID of Slave E PLC only right after the LINK is enabled and will not detect the ID again afterward.

j) After the detection on Slave PLCs is completed, Master PLC starts to read and write data from/into every Slave PLC. Please be noted that, Master PLC only reads/writes data from/into the detected Slave PLC.

Master PLC will not read/write from/into the new Slave PLC to the LINK, unless it re-detects the ID of Slave PLCs.

k) Master PLC conducts reading before writing. The ranges of Slave PLCs to be read/written will follow the setting.

l) Master PLC will move to the reading/writing of the next Slave PLC after finishing reading/writing the current Slave PLC.

10. Auto mode and maual mode:

a) Auto mode (M1351 = On): Master PLC automatically reads/writes from/to Slave PLC and stops when M1350/M1351 = Off to terminate PLC LINK.

b) Manual mode (M1352 = On): When M1352 = On, you also have to set up the times of polling in D1431. One time of polling refers to the completion of reading and writing all Slaves. After the PLC LINK is enabled, D1432 will start to count how many LINKs have been done. When D1431 = D1432, PLC LINK stops and M1352 is reset. To re-enable PLC LINK in the manual mode, simply turn “On” M1352 and D1431 will starts to count the times of LINK again.

c) Note:

i) Automatic mode (M1351) and manual mode (M1352) cannot be “On” at the same time; otherwise PLC LINK will stop and M1350 will be reset.

ii) For EH3/SV2: M1350 has to be reset before switching between automatic mode and manual mode. For SX: No such restriction.

iii) When M1355 = On, M1360 ~ M1375 (M1440 ~ M1455) will be the flags for the PLC designated to be connected to. Therefore, no matter how many PLCs are actually connected, the PLC LINK function will continue to send read/write instruction in cycle according to the designated flags.

iv) The communication timeout is adjustable (D1129, range: 200 ≤ D1129 ≤ 3,000). If D1129 falls out the range, PLC will determine the time by 200 or 3,000. The timeout setting of PLC LINK is only valid if it is set before the LINK is enabled. If the number of read/written data >16, the communication timeout shall be > 500ms in case a communication timeout may occur.

v) PLC LINK is only workable when the baud rate is > 1,200bps. If the baud rate is < 9,600, the communication timeout setting shall be > 1 second.

vi) The communication is unworkable when the number of read/written data = 0.

vii) PLC LINK does not support the reading/writing from/to 32-bit counters (C200 ~ C255).

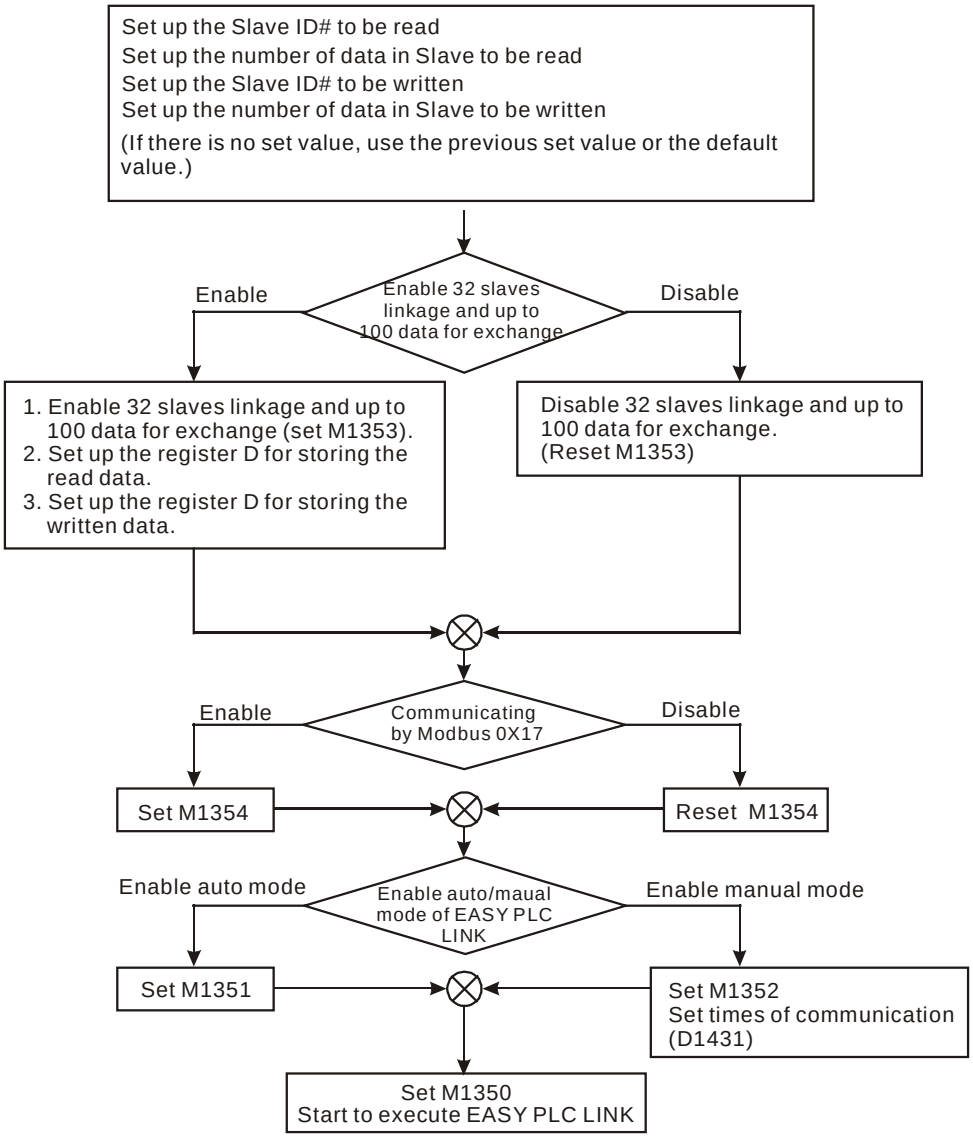
viii) The maximum set value for D1399 is 230. If the set value is bigger than 230, PLC will automatically correct it as 230. The minimum set value for D1300 is 1. If the set value is smaller than 1, PLC will automatically correct it as 1.

ix) Setting up of D1399 has to be done before PLC LINK is enabled. After PLC LINK is enabled, setting up D1399 will not result in any changes.

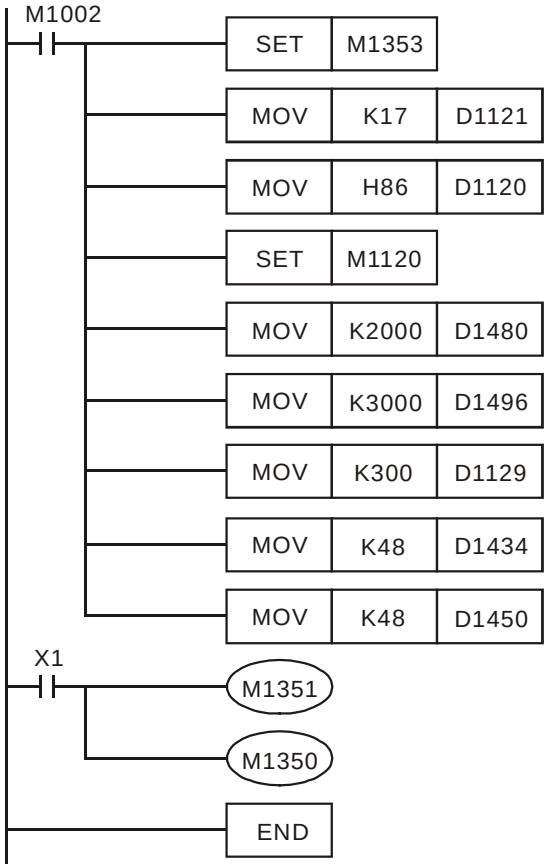
x) Advantages (when using a multi-layer network): Assume you are using a network with 3 layers and the first and second layer and the second and third layer are using PLC LINK for communication, the IDs in the second and third layer will definitely overlap due to the old version of PLC LINK detects only Slave ID#1 ~ 16. When the IDs of Slave and Master overlap, PLC LINK will ignore the PLC of overlapping ID, resulting in the

situation that the third layer can have only 15 PLCs. Therefore, D1399 allows more PLCs connected in a multi-layer network.

11. Operation Procedure of PLC LINK

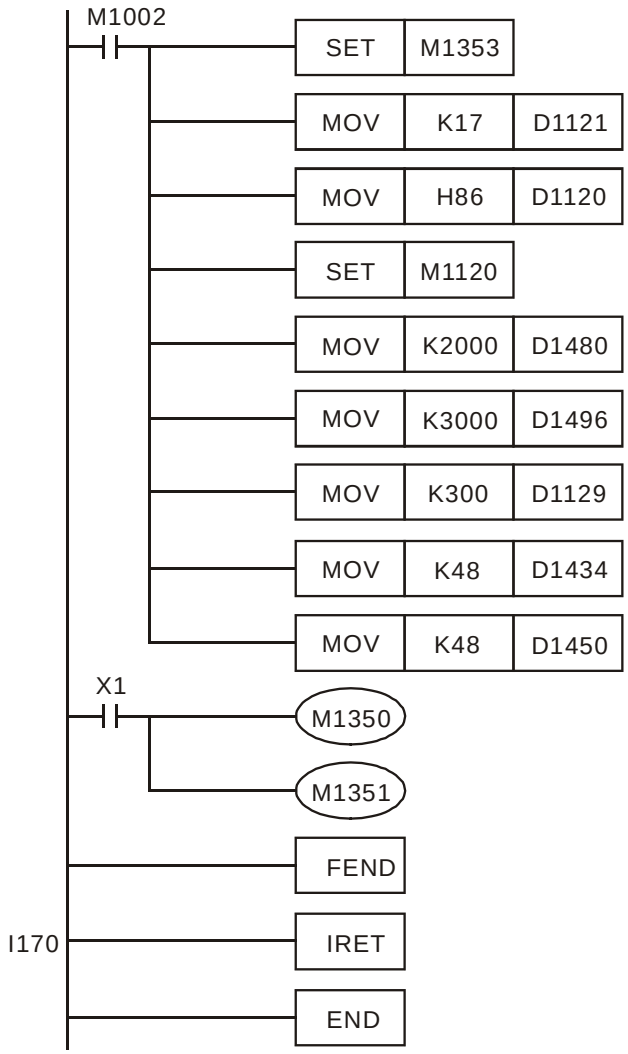


a) Example 1: Enabling 32 slave units linkage and up to 100 data for exchange in PLC LINK by M1353



- M1353 has to be set On before PLC LINK is enabled. When PLC LINK is executed, On/Off of M1353 will not affect the execution.
- Registers designated in D1480 ~ D1495, D1576~D1591 (starting register for data read from Slave ID#1 ~ 16) and D1496 ~ D1511, D1592 ~ D1607 (starting register for data written into Slave ID#1 ~ 16) can only be register D, and every special D will correspond to one Slave ID#, e.g. D1480 corresponds to ID1, D1481 to ID2, and so on.
- D1480 ~ D1495, D1576 ~ D1591, D1496 ~ D1511 and D1592 ~ D1607 have to be set before PLC LINK is enabled. In the execution of PLC LINK, you can modify the contents in these special Ds, but the modified results will take effect in the next PLC LINK polling.
- If the ID# designated by D1480 ~ D1495, D1576 ~ D1591, D1496 ~ D1511 and D1592 ~ D1607 is smaller than 0 or bigger than 9,900, PLC will automatically correct the ID# into 9,900.
- If M1353 is not enabled during the execution of PLC LINK, the range for D1434 ~ D1449, D1544 ~ D1559 (number of data read from Slave ID#) and D1450 ~ D1465, D1560 ~ D1575 (numbere of data written into Slave ID#) will be 0 ~ 16. If the setting in the special D exceeds the range, PLC will correct it to 16. When M1353 is enabled, the range will be 0 ~ 100. If the setting in the special D exceeds the range, PLC will correct it to 100.
- You can modify the settings in D1434 ~ D1449, D1544 ~ D1559, D1450 ~ D1465 and D1560 ~ D1575 during the execution of PLC LINK, but the modified results will take effect in the next PLC LINK polling.

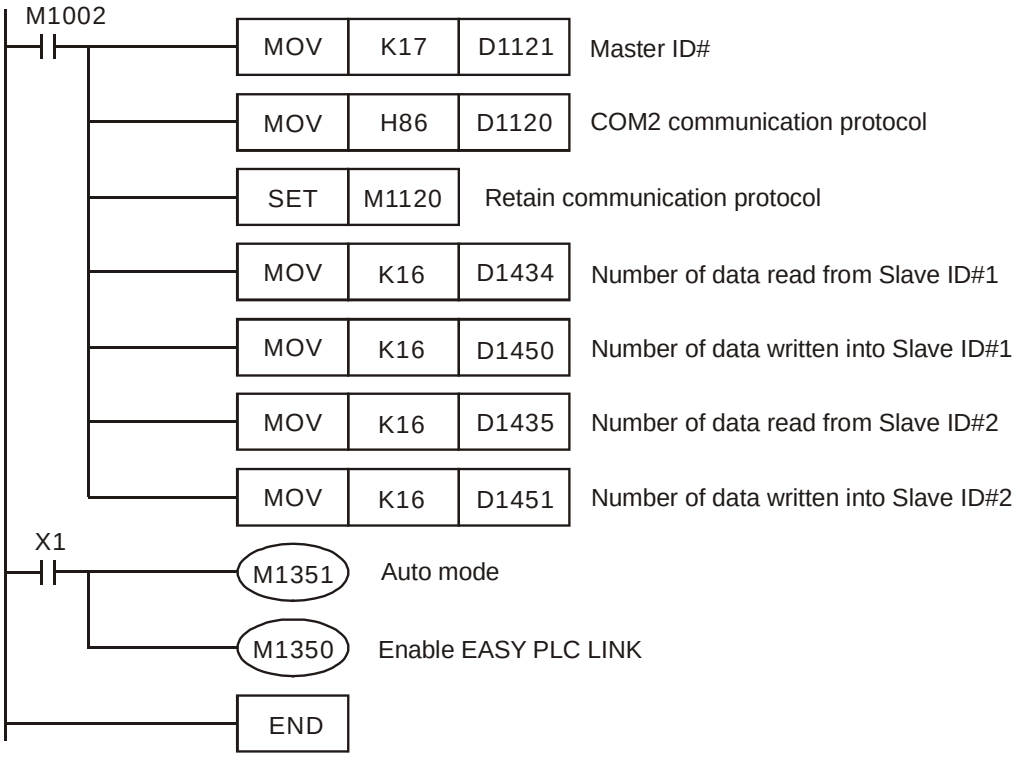
b) Example 2: PLC LINK with M1353 and I170



- With I170, the processing of data in PLC LINK will not be done at END, but at the enabling of I170 immediately after the receiving of data is completed
- If the reaction speed of the RS-485 IC direction control signal pin of the Slave is slow, it is suggested that you do not enable I170.
- You can set up D1399 (starting Slave ID designated by PLC LINK), and the ID# of the next 15 Slaves have to be in sequence. For example, when D1399 is set as K20, the Master PLC will detect Slave ID# 20 ~ 35.

c) Example 3: Connection of 1 Master and 2 Slaves by RS-485 and exchange of 16 data between Master and Slaves through PLC LINK (M1353 = Off, linkage of 16 stations, 16 data read/write mode)

- Write the ladder diagram program into Master PLC (ID#17)



- When X1 = On, the data exchange between Master and the two Slaves will be automatically done in PLC LINK, i.e. the data in D100 ~ D115 in the two Slaves will be read into D1480 ~ D1495 and D1512 ~ D1527 of the Master, and the data in D1496 ~ D1511 and D1528 ~ D1543 will be written into D200 ~ D215 of the two Slaves.

Master PLC *1		Slave PLC*2
D1480 ~ D1495	Read ←	D100 ~ D115 of Slave ID#1
D1496 ~ D1511	Write →	D200 ~ D215 of Slave ID#1
D1512 ~ D1527	Read ←	D100 ~ D115 of Slave ID#2
D1528 ~ D1543	Write →	D200 ~ D215 of Slave ID#2

- Assume the data in D for data exchange between Master and Slave before PLC LINK is enabled (M1350 = Off) are as the follow:

Master PLC	Preset value	Slave PLC	Preset value
D1480 ~ D1495	K0	D100 ~ D115 of Slave ID#1	K5,000
D1496 ~ D1511	K1,000	D200 ~ D215 of Slave ID#1	K0
D1512 ~ D1527	K0	D100 ~ D115 of Slave ID#2	K6,000
D1528 ~ D1543	K2,000	D200 ~ D215 of Slave ID#2	K0

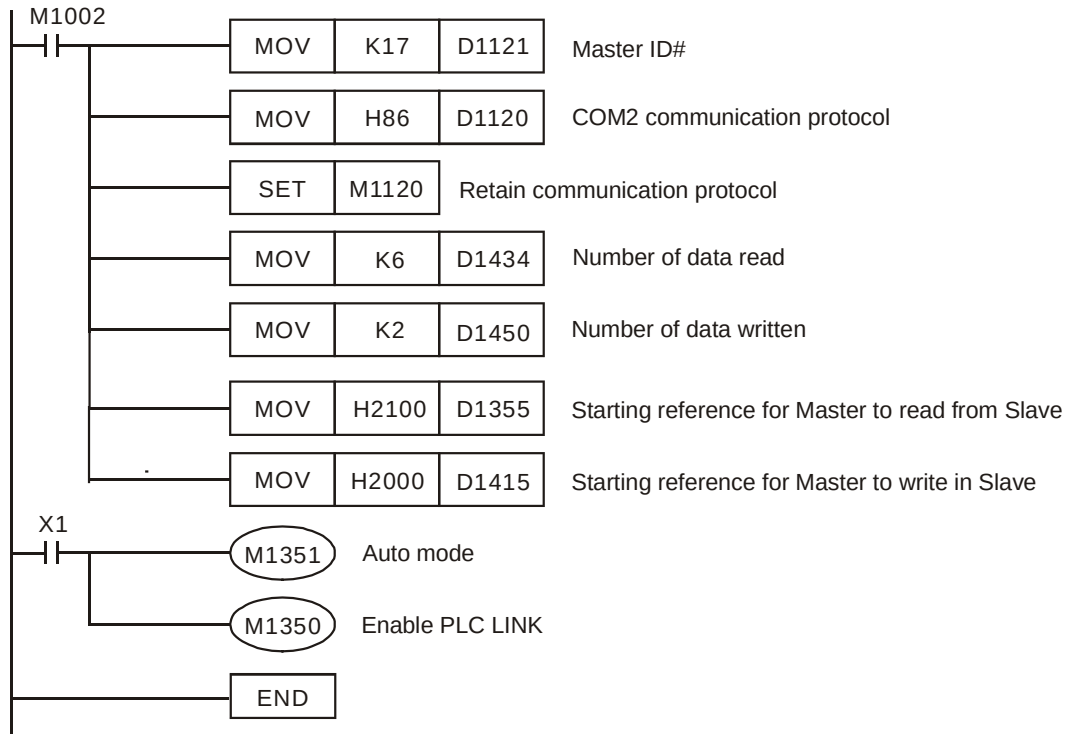
After PLC LINK is enabled (M1350 = On), the data in D for data exchange will become:

Master PLC	Preset value	Slave PLC	Preset value
D1480 ~ D1495	K5,000	D100 ~ D115 of Slave ID#1	K5,000
D1496 ~ D1511	K1,000	D200 ~ D215 of Slave ID#1	K1,000
D1512 ~ D1527	K6,000	D100 ~ D115 of Slave ID#2	K6,000
D1528 ~ D1543	K2,000	D200 ~ D215 of Slave ID#2	K2,000

- The Master PLC has to be SX/EH3/SV2 Series CPU, and the Slave PLC can be any CPU of DVP series.
- There can be maximum 16 Slave PLCs in PLC LINK. See the special Ds in the Master PLC corresponding to D100 ~ D115 and D200 ~ D215 in every Slave PLC in the tables of special M and special D.

d) Example 4: Connection between Delta PLC and Delta VFD-M AC motor drive through PLC LINK for STOP, forward/reverser revolution and writing/reading of frequency.

- Write the ladder diagram program into Master PLC (ID#17)



- D1480 ~ D1485 correspond to parameters H2100 ~ H2105 in VFD-M. When X1 = On, PLC LINK will be enabled, and the data in H2100 ~ H2105 will be displayed in D1480 ~ D1485.
- D1496 ~ D1497 correspond to parameters H2000 ~ H2001 in VFD-M. When X1 = On, PLC LINK will be enabled, and the data in H2000 ~ H2001 will be displayed in D1496 ~ D1497.
- Modify D1496 to give command to VFD, e.g. D1496 = H12: enabling forward revolution of VFD-M; D1496 = H11: enabling reverse revolution of VFD.
- Modify D1497 to change the frequency of VFD, e.g. D1497 = K5,000: changing the frequency to 50kHz.

- The Master PLC has to be SX/EH3/SV2 Series CPU, and the Slave AC motor drive can be any VFD series models except VFD-A.
- The Slave can also be Delta temperature controller DTA, DTB, Delta servo ASDA and so on which are compatible to Modbus protocol. Maximum 16 devices are connectable to the LINK.
- See the tables of special M and special D for the starting ID of Slave to be read/written and the number of data to be read/written.

Function Group Enabling the instruction DICF to execute the constant speed/final output section

Number M1528~M1529

Contents:

1. M1528→ On: Enabling the instruction DICF to execute the constant speed output section
   M1529→ On: Enabling the instruction DICF to execute the final output section
   If users want to use this special output function, they have to use M1528/M1529 with DVSPO/DICF. Please do not enable the two flags simultaneously.
2. The instruction format is **DVSPO S_1 S_2 S_3 D**. The instruction is a 32-bit instruction. **S_1** is the target frequency of output, **S_2** is the target number of pulses, **S_3** indicates the gap time and the gap frequency, and **D** is the pulse output device. (If **S_2** is K0, there will be no limit on the number of output pulses, and pulses will be output until the final output section is set.
3. The instruction format is **DICF S_1 S_2 D**. The instruction is a 32-bit instruction. **S_1** is the target frequency to be changed, **S_2** indicates the gap time and the gap frequency, and **D** is the pulse output device.
4. If users do not need to set the target number of pulses when DVSPO enables output, they can set **S_2** to K0, and then set the target frequency of the output, the gap time, and the gap frequency. If users want to end the output, they can use DICF and M1529 to execute the final output section. After the execution of the final output section is complete, DVSPO sill set the completion flag. (Please refer to example 1.)
5. When DICF is used with M1529 to execute the final output section, **S_1** is the target frequency of output (**S_1** can not be modified by an E device or an F device), and the 32-bit value indicated by **S_1**+2 is the number of pulses which need to be output in the final section. For example, if **S_1** is D100, the 32-bit value in (D101, D100) is the target frequency of output, and the 32-bit value in (D103, D102) is the number of pulses which need to be output in the final section. After M1529 successfully enables the final output section, M1529 will be reset to Off automatically.
6. In the final output section, DICF carries out acceleration/deceleration operations according to the acceleration/deceleration time specified by the axes (the parameter indicated by S2 is not used), and the current output speed increases/decreases until the number of pulses required is reached. DVSPO sets the completion flag. (Please refer to the examples in 1.1~1.2 in example 1 for more information.)
7. If the number of pulses in the final section is not sufficient for the acceleration/deceleration output set by users, DICF will automatically change the acceleration/deceleration operation to the default acceleration/deceleration operation so that the target number of pulses in the final output section can be reached.
8. When DICF is used with M1528 to execute the constant speed output section, **S_1** is the target frequency of output (**S_1** can not be modified by an E device or an F device), and the 32-bit value indicated by **S_1**+2 is the number of pulses which need to be output in the constant speed output section. For example, if **S_1** is D100, the

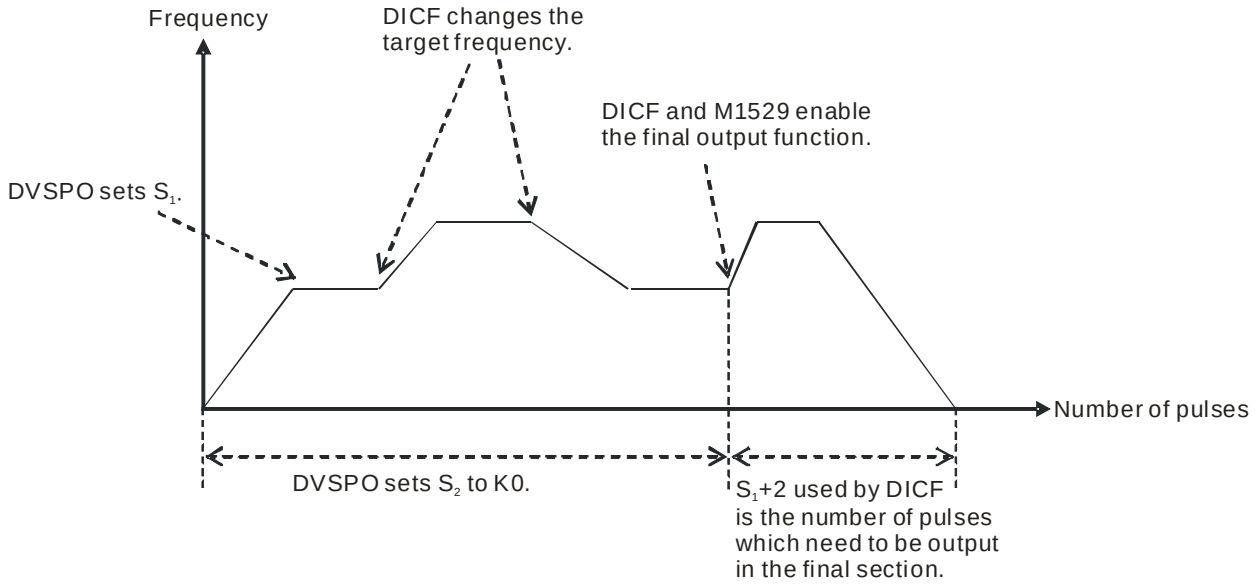
32-bit value in (D101, D100) is the target frequency of output, and the 32-bit value in (D103, D102) is the number of pulses which need to be output in the constant speed section. After M1528 successfully enables the constant speed output section, M1528 will be reset to Off automatically. (Please refer to example 2 for more information.)

9. After DICF enables the execution the final output section (M1529)/constant speed output section (M1528), the variable speed function of DVSPO/DICF will be disabled. Not until the execution is complete can the variable speed function of DVSPO/DICF be used.

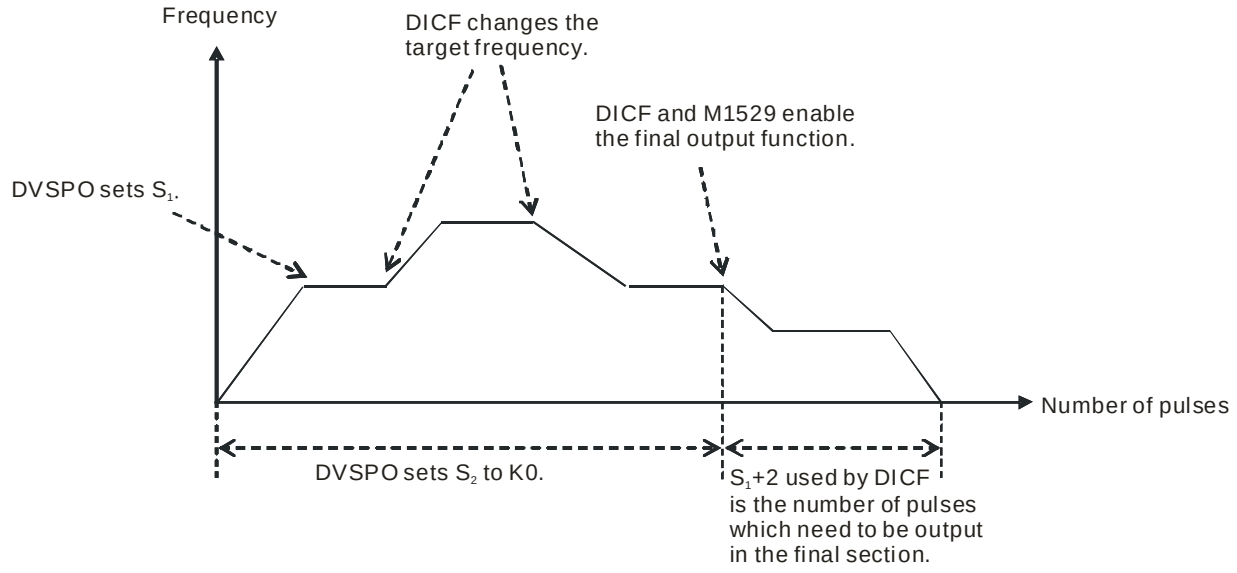
Example:

- The timing diagram for the final output section is shown below.

  1. The target frequency in the final section is higher than the current output frequency.



  2. The target frequency in the final section is lower than the current output frequency



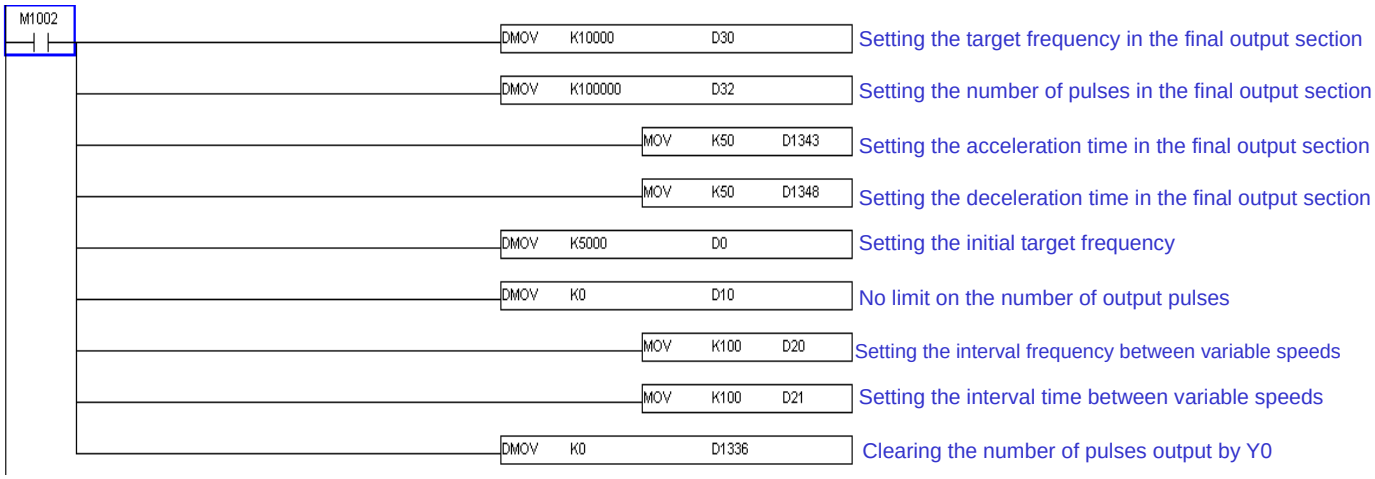
Note 1: Although there will be no limit on the number of output pulses if **S2** is K0, the number of pulses in the final section will be the target number of pulses in the final section after the final output section is enabled, and

the completion flag will be set after the target number of pulses is reached.

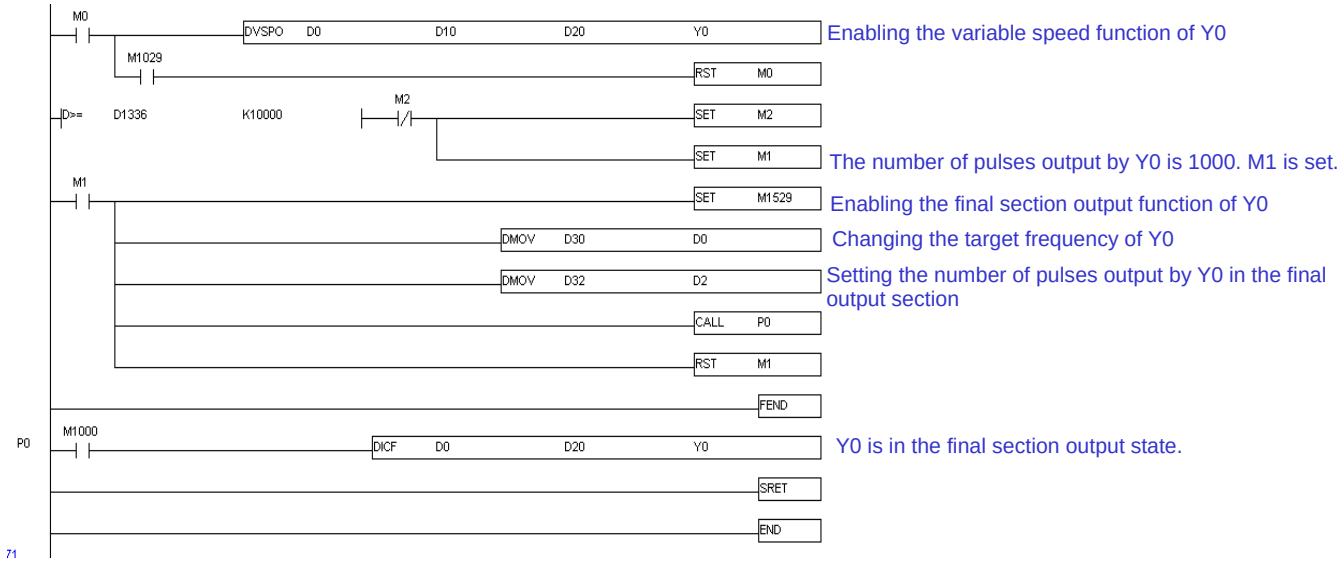
Note 2: If DICF in an interrupt is used to change the target frequency, please note that the reaction time of the interrupt will affect the execution of the final output section.

3. Sample program

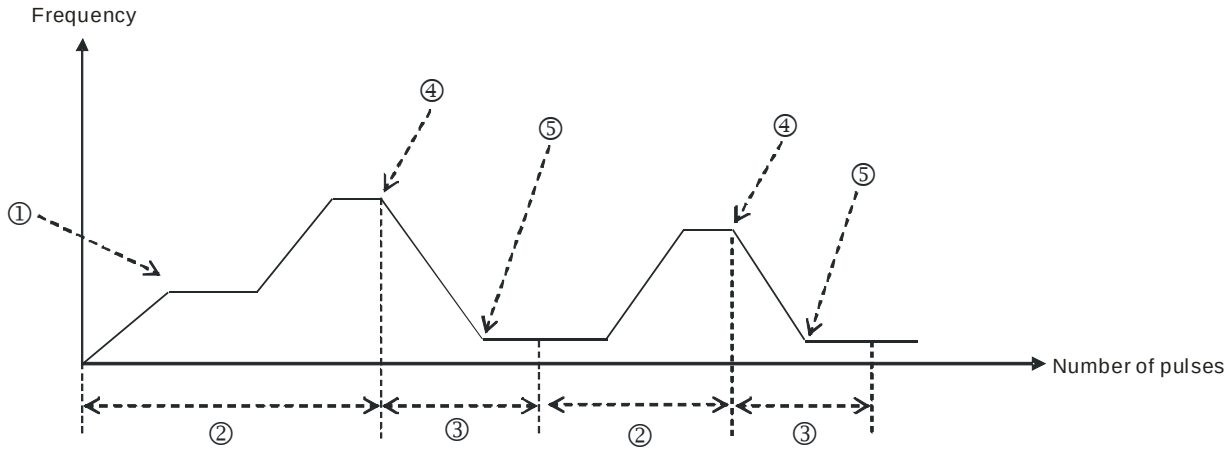
◆ Setting initial values



◆ Actual output control program



- The timing diagram for the constant speed output section is shown below.



① → DVSPO sets the target frequency (**S_1**)

② → DVSPO sets the number of pulses (**S_2**) to K0 (no limit on the number of output pulses).

③ → DICF sets the number of pulses which need to be output in the constant speed output section (**S_1**+2).

④ → DICF and M1528 are used to reach the target frequency and the target number of pulses in the constant speed output section.

⑤ → The acceleration/deceleration is complete, and M1542 is set to On.

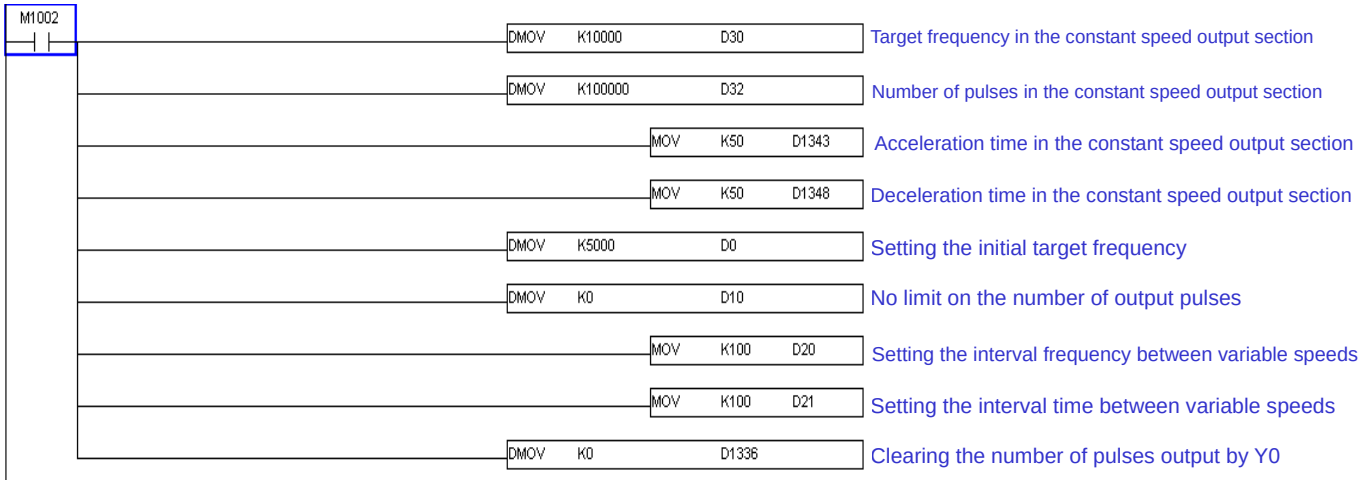
⑥ → The target number of pulses in the constant speed output section is reached, M1543 is set to On, and M1542 is reset to Off. (The output used is Y0.)

Note: Whenever the constant speed output section is executed, the instruction resets the reaching flag and the completion flag.1. The flags corresponding to the axes used to execute the constant speed output section are shown below.

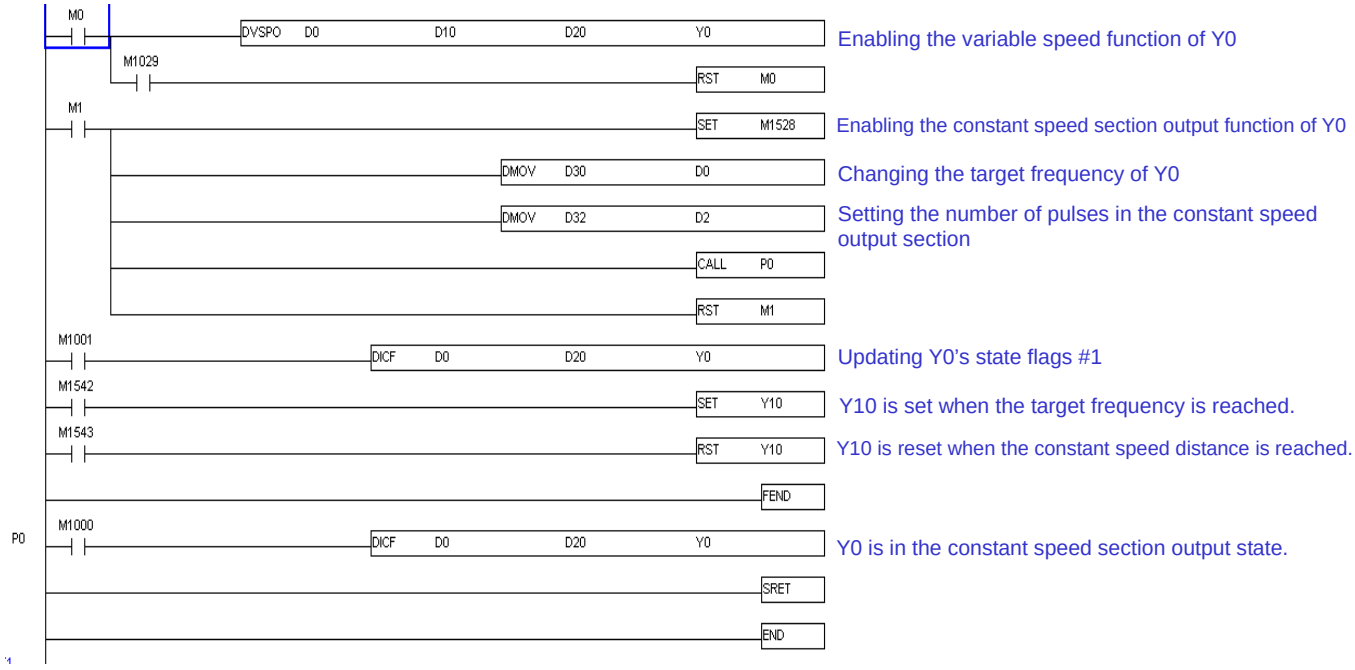
Output number	Reaching the target frequency in the constant speed output section	Finishing the constant speed output
Y0	M1542	M1543
Y2	M1544	M1545
Y4	M1546	M1547
Y6	M1548	M1549

2. Sample program:

◆ Setting initial values



◆ Actual output control program



Note 1: If DICF is not enabled, and the constant speed output section is executed, the instruction will constantly check whether the target frequency is reached and whether the output state is complete, and will set the corresponding flags. If the scan time is long, users can insert many lines of DICF instructions, or execute DICF in a time interrupt (do not need to enable the instruction) to update the output state constantly.

Function Group PLC Operation Flag

Number M1145 (should work with D1400 ~ D1403)

Contents:

Read the MAC Address of the left-side module (the numbering for the 1st left-side module is K100, the 8th one is K107. See the example below.

1. If you need the MAC address of the 2nd left-side module, you can set the value to K101 in D1400.
2. Set M1145 to ON and PLC stores the MAC address of the 2nd left-side module in D1401~1403.
3. If the MAC address of the 2nd left-side module is 11:22:33:44:55:66, D1401 D1401 = 0x1122, D1402 = 0x3344, D1403 = 0x5566.
4. Available for EH3/SV2 (V2.2 or later versions).

2.12 Communication Addresses of Devices in DVP Series PLC

Device	Range		Type	DVP Com. Address (hex)	Modbus Com. Address (dec)
S	000 ~ 255		bit	0000 ~ 00FF	000001 ~ 000256
S	246 ~ 511		bit	0100 ~ 01FF	000247 ~ 000512
S	512 ~ 767		bit	0200 ~ 02FF	000513 ~ 000768
S	768 ~ 1,023		bit	0300 ~ 03FF	000769 ~ 001024
X	000 ~ 377 (Octal)		bit	0400 ~ 04FF	101025 ~ 101280
Y	000 ~ 377 (Octal)		bit	0500 ~ 05FF	001281 ~ 001536
T	000 ~ 255		bit	0600 ~ 06FF	001537 ~ 001792
			word	0600 ~ 06FF	401537 ~ 401792
M	000 ~ 255		bit	0800 ~ 08FF	002049 ~ 002304
M	256 ~ 511		bit	0900 ~ 09FF	002305 ~ 002560
M	512 ~ 767		bit	0A00 ~ 0AFF	002561 ~ 002816
M	768 ~ 1,023		bit	0B00 ~ 0BFF	002817 ~ 003072
M	1,024 ~ 1,279		bit	0C00 ~ 0CFF	003073 ~ 003328
M	1,280 ~ 1,535		bit	0D00 ~ 0DFF	003329 ~ 003584
M	1,536 ~ 1,791		bit	B000 ~ B0FF	045057 ~ 045312
M	1,792 ~ 2,047		bit	B100 ~ B1FF	045313 ~ 045568
M	2,048 ~ 2,303		bit	B200 ~ B2FF	045569 ~ 045824
M	2,304 ~ 2,559		bit	B300 ~ B3FF	045825 ~ 046080
M	2,560 ~ 2,815		bit	B400 ~ B4FF	046081 ~ 046336
M	2,816 ~ 3,071		bit	B500 ~ B5FF	046337 ~ 046592
M	3,072 ~ 3,327		bit	B600 ~ B6FF	046593 ~ 046848
M	3,328 ~ 3,583		bit	B700 ~ B7FF	046849 ~ 047104
M	3,584 ~ 3,839		bit	B800 ~ B8FF	047105 ~ 047360
M	3,840 ~ 4,095		bit	B900 ~ B9FF	047361 ~ 047616
C	0 ~ 199	16-bit	bit	0E00 ~ 0EC7	003585 ~ 003784
			word	0E00 ~ 0EC7	403585 ~ 403784
	200 ~ 255	32-bit	bit	0EC8 ~ 0EFF	003785 ~ 003840
			word	0700 ~ 076F	403785 ~ 403840
D	000 ~ 256		word	1000 ~ 10FF	404097~404352
D	256 ~ 511		word	1100 ~ 11FF	404353 ~ 404608
D	512 ~ 767		word	1200 ~ 12FF	404609 ~ 404864
D	768 ~ 1,023		word	1300 ~ 13FF	404865 ~ 405120
D	1,024 ~ 1,279		word	1400 ~ 14FF	405121 ~ 405376
D	1,280 ~ 1,535		word	1500 ~ 15FF	405377 ~ 405632
D	1,536 ~ 1,791		word	1600 ~ 16FF	405633 ~ 405888
D	1,792 ~ 2,047		word	1700 ~ 17FF	405889 ~ 406144
D	2,048 ~ 2,303		word	1800 ~ 18FF	406145 ~ 406400

Device	Range	Type	DVP Com. Address (hex)	Modbus Com. Address (dec)
D	2,304 ~ 2,559	word	1900 ~ 19FF	406401 ~ 406656
D	2,560 ~ 2 815	word	1A00 ~ 1AFF	406657 ~ 406912
D	2,816 ~ 3,071	word	1B00 ~ 1BFF	406913 ~ 407168
D	3,072 ~ 3,327	word	1C00 ~ 1CFF	407169 ~ 407424
D	3,328 ~ 3,583	word	1D00 ~ 1DFF	407425 ~ 407680
D	3,584 ~ 3,839	word	1E00 ~ 1EFF	407681 ~ 407936
D	3,840 ~ 4,095	word	1F00 ~ 1FFF	407937 ~ 408192
D	4,096 ~ 4,351	word	9000 ~ 90FF	436865 ~ 437120
D	4,352 ~ 4,607	word	9100 ~ 91FF	437121 ~ 437376
D	4 608 ~ 4863	word	9200 ~ 92FF	437377 ~ 437632
D	4,864 ~ 5,119	word	9300 ~ 93FF	437633 ~ 437888
D	5,120 ~ 5,375	word	9400 ~ 94FF	437889 ~ 438144
D	5,376 ~ 5,631	word	9500 ~ 95FF	438145 ~ 438400
D	5,632 ~ 5,887	word	9600 ~ 96FF	438401 ~ 438656
D	5,888 ~ 6,143	word	9700 ~ 97FF	438657 ~ 438912
D	6,144 ~ 6,399	word	9800 ~ 98FF	438913 ~ 439168
D	6,400 ~ 6,655	word	9900 ~ 99FF	439169 ~ 439424
D	6,656 ~ 6,911	word	9A00 ~ 9AFF	439425 ~ 439680
D	6,912 ~ 7,167	word	9B00 ~ 9BFF	439681 ~ 439936
D	7,168 ~ 7,423	word	9C00 ~ 9CFF	439937 ~ 440192
D	7,424 ~ 7,679	word	9D00 ~ 9DFF	440193 ~ 440448
D	7,680 ~ 7,935	word	9E00 ~ 9EFF	440449 ~ 440704
D	7,936 ~ 8,191	word	9F00 ~ 9FFF	440705 ~ 440960
D	8,192 ~ 8,447	word	A000 ~ A0FF	440961 ~ 441216
D	8,448 ~ 8,703	word	A100 ~ A1FF	441217 ~ 441472
D	8,704 ~ 8,959	word	A200 ~ A2FF	441473 ~ 441728
D	8,960 ~ 9,215	word	A300 ~ A3FF	441729 ~ 441984
D	9,216 ~ 9,471	word	A400 ~ A4FF	441985 ~ 442240
D	9,472 ~ 9,727	word	A500 ~ A5FF	442241 ~ 442496
D	9,728 ~ 9,983	word	A600 ~ A6FF	442497 ~ 442752
D	9984~10239	word	A700~A7FF	442753~443008
D	10240~10495	word	A800~A8FF	443009~443246
D	10496~10751	word	A900~A9FF	443247~443502
D	10752~11007	word	AA00~AAFF	443503~443758
D	11008~11263	word	AB00~ABFF	443759~444014
D	11264~11519	word	AC00~ACFF	444015~444270
D	11520~11775	word	AD00~ADFF	444271~444526
D	11776~11999	word	AE00~AEDF	444527~444750

2.13 LED Indicators and Error Codes

Error: Power indicator blinking - communication not working

Situation: Once the PLC is supplied with power and the power indicator of the PLC CPU is blinking, that means the power supply is not sufficient or the electrical load that PLC CPU provides is not sufficient.

Solution: Make sure the power supply is correctly wired and no short-circuit or wrong wiring is presented.

Error: Error LED indicator and RUN indicator are blinking at the same time- communication not working

Situation: Once the PLC is supplied with power and both the power indicator and ERROR LED indicator of the PLC CPU are blinking at the same time; that means the last firmware upgrade had failed, if you have tried to upgrade the firmware. If you did not upgrade the firmware and the indicators are blinking for 3 to 5 times and then they stop for 2 seconds and then blink again. That means the PLC CPU cannot self-diagnose itself or its connected device.

Solution: Contact the local distributor for servicing information.

Error: Error indicator ON - communication okay

Situation: When the PLC runs and the Error LED indicator of the PLC CPU is blinking and M1008 is ON, that means the program scan cycle exceeded the value set in special register D1000 (default 200 ms).

Solution: You can use D1008 to see in which step that the error occured, and then check the steps before and after the errorous step to see if there is any problem in the procedure. If the PLC program is too large and you cannot see why the error occurs, it is suggested to delete some programs and leave some questionable ones to examine.

Error: Error indicator blinking every 0.2 seconds - communication not working

Situation: Once the PLC is supplied with power, the Error LED indicator of the PLC CPU is blinking rapidly. That means the 24V power supply is not sufficient. After the 24V power is supplied stably for more than 2 seconds, the RUN indicator will be ON.

Solution: If the rapid blinking persists, check if the power supply is normal.

Error: Error indicator blinking every 0.5 seconds - communication okay

Situation: After you download the PLC program to the PLC CPU or when the PLC is supplied with power, the Error LED indicator starts to blinking every 0.5 seconds. That means the use of operands (devices) or syntax in the program is illegal and the special register M1004 is On.

Solution: Check the error code (hex) in the special register D1004 to see the cause. See the error codes listed below for reference. The address where the error occured is stored in the data register D1137. But if the error is a general loop one, the address stored in D1137 is invalid.

Error code	Cause of error
0001	Use of device S exceeds the range
0002	Using P* repeatedly or use of P* exceeds the range
0003	Use of KnSm exceeds the range
0102	Using I* repeatedly or use of I* exceeds the range
0202	Use of MC N* exceeds the range
0302	Use of MCR N* exceeds the range
0401	Use of device X exceeds the range
0403	Use of KnXm exceeds the range
0501	Use of device Y exceeds the range
0503	Use of KnYm exceeds the range
0601	Use of device T exceeds the range
0604	Use of register T exceeds the range
0801	Use of device M exceeds the range
0803	Use of KnMm exceeds the range
0B01	Incorrect use of KH
0D01	Improper use of operands of DECO instruction
0D02	ES/EX/SS/EH3/SV2: improper use of operands of ENCO instruction SX/EC3-8K: illegal use of the first operand of ANS instruction
0D03	Improper use of operands of DHSCS instruction
0D04	Improper use of operands of DHSCR instruction
0D05	Improper use of operands of pulse output instruction
0D06	Improper use of operands of PWM instruction
0D07	Improper use of operands of FROM/TO instruction
0D08	Improper use of operands of PID instruction
0D09	Improper use of operands of SPD instruction
0D0A	Incorrect operands in DHSZ instruction
0D0B	Improper use of operands in IST instruction
0E01	Use of device C exceeds the range
0E04	Use of register C exceeds the range
0E05	Improper use of operand CXXX of DCNT instruction
0E18	BCD conversion error
0E19	Division error (divisor = 0)

Error code	Cause of error
0F0A	Times of using TTMR, STMR instruction exceed the range
0F0B	Times of using SORT instruction exceed the range
0F0C	Times of using TKY instruction exceed the range
0F0D	Times of using HKY instruction exceed the range
1000	Improper use of operands of ZRST instruction
10EF	Incorrect use of E, F, or the modification exceeds the range
2000	Times of using TTMR, PR, HOUR instructions exceed the range. Improper use of operands of MRT, ARWS instructions
C400	Illegal instruction
C401	General loop error
C402	Continuously using LD/LDI instructions for more than 9 times
C403	Continuously using MPS for more than 9 times
C404	More than 6 steps in FOR – NEXT
C405	Using STL/RET between FOR – NEXT Using SRET/IRET between FOR – NEXT Using MC/MCR between FOR – NEXT Using END/FEND between FOR – NEXT
C407	Continuously using STL for more than 9 times
C408	Using MC/MCR in STL, using I/P in STL
C409	Using STL/RET in subroutine Using STL/RET in interruption subroutine
C40A	Using MC/MCR in subroutine Using MC/MCR in interruption subroutine
C40B	MC/MCR does not start from N0, or is not continuous
C40C	Corresponding N of MC and MCR are different
C40D	Improper use of I/P
C40E	IRET does not appear after the last FEND. SRET does not appear after the last FEND.
C40F	PLC program and data in parameters have not been initialized.
C41B	Invalid RUN/STOP instruction to extension module
C41C	Points of extension module exceed the range
C41D	Number of extension modules exceeds the range
C41E	Incorrect hardware setting for extension module
C41F	Failing to write data into the memory (EH2/SV) Right-side module detection error (EH3)

Error code	Cause of error
0E1A	Use of device exceeds the range (including E, F index register modification)
0E1B	The index of the radical is a negative value
0E1C	Communication error of FROM/TO instruction
0F04	Use of register D exceeds the range
0F05	Improper use of operand DXXX of DCNT instruction
0F06	Improper use of operands of SFTR instruction
0F07	Improper use of operands of SFTL instruction
0F08	Improper use of operands of REF instruction
0F09	Improper use of operands of WSFR, WSFL instructions

Error code	Cause of error
C420	Read/write function card error
C421	Read/write memory card error
C440	Hardware error in high-speed counter
C441	Hardware error in high-speed comparator
C442	Hardware error in MCU pulse output
C443	No response from extension unit
C450	The AD/DA function in the CPU breaks down.
C4EE	No END instruction in the program
C4FF	Invalid instruction (No such instruction exists.)

BAT. LOW indicator ON - communication okay

For SX, EH Series PLC CPU:

Situation: No battery or the power of the built-in battery is low.

Solution: It is suggested to turn the power off and then change battery or you can also change battey in 3 minutes without turning the power off so that for EH series PLC CPU, the retentive data in the latched area willnot be lost and for SX and EH series PLC CPU, the RTC function to keep track of the
current time can stay normal.

For SV Series PLC CPU:

Situation: The power of the built-in battery is low and is now being charged.

Solution: It is suggested to charge it for at least 6 hours to ensure the built-in battery is sufficiently charged. If the BAT. LOW indicator is from ON to blinking, the built-in battery can no longer be charged and you need to back up the retentive data in the latched area and contact the local distributor for a battey change. If this happens, it is highly possible that the retentive data in the latched area is lost.

Finding out the module which is disconnected or damaged in an EH system:

Situation: A DVP-EH2/EH3 Series CPU is powered, the ERROR LED indicator blinks, and the error code in D1004 is H'C41E.

Reading the value in D1104: The value in D1104 indicates the GPIO which is damaged. It represents a GPIO number. (GPIOs are numbered from 0.)

The number of GPIOs in an CPU or an extension module is described below.

CPU	40 / EH3	48 / EH3	64 / EH3	80 / EH3
Number of GPIOs	1	1	1	2

DIO module	08HP/HM/HN	16HP/HM	32HP/HN/HM	48HP
Number of GPIOs	1	1	1	2

AIO module	04AD / 04DA	06XA	04PT / 04TC	08TC
Number of GPIO	1	1	1	1

Other modules	01PU	01HC	PF02/CP02/DT02
Number GPIOs	1	1	1

Example 1: The system created is 32EH3 + 16HP + 04AD + 32HM.

If the error code in D1004 is H'C41E, and the value in D1104 is K1, the GPIO numbers in the system will be as shown in the table below.

System	32EH3	16HP	04AD	32HM
GPIO number	No GPIO	0	1	2

The GPIO which is damaged is probably the GPIO in 04AD. If the error code still exists after 04AD is replaced, there may be something wrong with the communication interface in 16HP.

Example 2: The system created is 40EH3 + 48HP + 04AD + 04PT.

If the error code in D1004 is H'C41E, and the value in D1104 is K2, the GPIO numbers in the system will be as shown in the table below.

System	40EH3	48HP	04AD	04PT
GPIO number	0	1, 2	3	4

The GPIO which is damaged is probably a GPIO in 48HP.

Example 3: There is only one 64EH3.

If the error code in D1004 is H'C41E, and the value in D1104 is K0, the GPIO number in 64EH3 will be as shown in the table below.

System	64EH3
GPIO number	0

The GPIO on the driver board in 64EH3 is damaged. The driver board needs to be replaced.

3.1 Basic Instructions and Step Ladder Instructions

ES includes ES/EX/EC/EC3-8K (FW V8.60 or later); SX (FW V3.00); EH3 includes EH3/SV2.

For EH3 series CPU, the execution speed in the brackets ( ) refers to the execution speed of designated operand M1536 ~ M4095.

🕮 Basic Instructions

Instruction Code	Function	Operands	Execution speed (μs)				STEP
			ES	EC3-8K	SX	EH3	
LD	Loading in A contact	X, Y, M, S, T, C	3.8	1.24	1.24	0.24(0.56)	1~3
LDI	Loading in B contact	X, Y, M, S, T, C	3.88	1.3	1.3	0.24(0.56)	1~3
AND	Series connection- A contact	X, Y, M, S, T, C	2.32	1.12	1.12	0.24(0.56)	1~3
ANI	Series connection- B contact	X, Y, M, S, T, C	2.4	1.27	1.27	0.24(0.56)	1~3
OR	Parallel connection- A contact	X, Y, M, S, T, C	2.32	1.25	1.25	0.24(0.56)	1~3
ORI	Parallel connection- B contact	X, Y, M, S, T, C	2.4	1.25	1.25	0.24(0.56)	1~3
ANB	Series connection- loop blocks	N/A	1.76	1.34	1.34	0.24	1~3
ORB	Parallel connection- loop blocks	N/A	1.76	1.63	1.63	0.24	1~3
MPS	Store the current result of the internal PLC operations	N/A	1.68	1.43	1.43	0.24	1~3
MRD	Reads the current result of the internal PLC operations	N/A	1.6	1.19	1.19	0.24	1
MPP	Pops (recalls and removes) the currently stored result	N/A	1.6	1.19	1.19	0.24	1

🕮 Output instructions

Instruction Code	Function	Operands	Execution speed (μs)				STEP
			ES	EC3-8K	SX	EH3	
OUT	Output coil	Y, M, S	5.04	1.24	1.24	0.24(0.56)	1~3
SET	Latched（On）	Y, M, S	3.8	1.27	1.27	0.24(0.56)	1~3
RST	Clear the contacts or the registers	Y, M, S, T, C, D, E, F	7.8	2.24	2.24	0.24(0.56)	3

🕮 Timers, Counters

API	Instruction Code	Function	Operands	Execution speed (μs)				STEP
				ES	EC3-8K	SX	EH3	
96	TMR	16-bit timer	T-K or T-D	10.6	2.4	2.4	9.6	4
97	CNT	16-bit counter	C-K or C-D (16 bits)	9.7	2.24	2.24	12.8	4
97	DCNT	32-bit counter	C-K or C-D (32 bits)	10.3	2.32	2.32	14.3	6

🕮 Main control instructions

Instruction Code	Function	Operands	Execution speed (μs)				STEP
			ES	EC3-8K	SX	EH3	
MC	Master control start	N0 ~ N7	5.6	1.4	1.4	5.6	3
MCR	Master control reset	N0 ~ N7	5.7	1.5	1.5	5.7	3

🕮 Instructions for detecting the contacts of rising-/falling-edge

API	Instruction Code	Function	Operands	Execution speed (μs)				STEP
				ES	EC3-8K	SX	EH3	
90	LDP	Rising-edge detection operation	X, Y, M, S, T, C	5.1	1.99	1.99	0.56(0.88)	3
91	LDF	Falling-edge detection operation	X, Y, M, S, T, C	5.1	1.96	1.96	0.56(0.88)	3
92	ANDP	Rising-edge series connection	X, Y, M, S, T, C	4.9	1.9	1.9	0.56(0.88)	3
93	ANDF	Falling-edge series connection	X, Y, M, S, T, C	4.9	2.16	2.16	0.56(0.88)	3
94	ORP	Rising-edge parallel connection	X, Y, M, S, T, C	4.9	2.24	2.24	0.56(0.88)	3
95	ORF	Falling-edge parallel connection	X, Y, M, S, T, C	4.9	2.27	2.27	0.56(0.88)	3

🕮 Rising-/falling-edge output instructions

API	Instruction Code	Function	Operands	Execution speed (μs)				STEP
				ES	EC3-8K	SX	EH3	
89	PLS	Rising-edge output	Y, M	7.8	2.51	2.51	9.92	3
99	PLF	Falling-edge output	Y, M	7.8	2.51	2.51	10.16	3

🕮 End instruction

Instruction Code	Function	Operands	Execution speed (μs)				STEP
			ES	EC3-8K	SX	EH3	
END	Program ends	N/A	5	1.24	1.24	0.24	1

🕮 Other instructions

API	Instruction Code	Function	Operands	Execution speed (μs)				STEP
				ES	EC3-8K	SX	EH3	
	NOP	No operation	N/A	0.88	1.18	1.18	0.16	1
98	INV	Inverting operation	N/A	1.6	1.32	1.32	0.24	1
	P	Pointer	P0 ~ P255	0.88	1.32	1.32	—	1
	I	Interruption program marker	I□□□	0.88	1.32	1.32	—	1

🕮 Step ladder instructions

Instruction Code	Function	Operands	Execution speed (μs)				STEP
			ES	EC3-8K	SX	EH3	
STL	Step transition ladder start instruction	S	11.6	2.2	2.2	0.56	1
RET	Step transition ladder return instruction	N/A	7.04	1.24	1.24	0.24	1

Note 1: ES includes ES/EX/SS; SA includes SA/SX/SC; EH includes EH/EH2/SV.

Note 2: For EH series MPU, the execution speed in the brackets ( ) refers to the execution speed of designated operand M1536 ~ M4095.

3.2 Explanations on Basic Instructions

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
LD	Loading in A contact	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The LD instruction is used on the A contact that has its start from the left BUS or the A contact that is the start of a contact circuit. The functions are to save the present contents and store the acquired contact status into the accumulative register.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	**X0**	Loading in contact A of X0
AND	X1	Connecting to contact A of X1 in series
OUT	Y1	Driving Y1 coil

Note: DVP-EH3 series PLCs whose version is 1.40 and DVP-SV2 series PLCs whose version is 1.20 support the operands X, Y, M, and S. These operands can be qualified by E or F, e.g. LD X0E1.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
LDI	Loading in B contact	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The LDI instruction is used on the B contact that has its start from the left BUS or the B contact that is the start of a contact circuit. The functions are to save the present contents and store the acquired contact status into the accumulative register.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LDI	**X0**	Loading in contact B of X0
AND	X1	Connecting to contact A of X1 in series
OUT	Y1	Driving Y1 coil

Note: DVP-EH3 series PLCs whose version is 1.40 and DVP-SV2 series PLCs whose version is 1.20 support the operands X, Y, M, and S. These operands can be qualified by E or F, e.g. LDI X0E1.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
AND	Series connection- A contact	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The AND instruction is used in the series connection of A contact. The functions are to read out the status of present d series connection contacts and perform the “AND” operation with the logical operation result obtained. The final result will be store in the accumulative register.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LDI	X1	Loading in contact B of X1
AND	**X0**	Connecting to contact A of X0 in series
OUT	Y1	Driving Y1 coil

Note: DVP-EH3 series PLCs whose version is 1.40 and DVP-SV2 series PLCs whose version is 1.20 support the operands X, Y, M, and S. These operands can be qualified by E or F, e.g. AND X0E1.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
ANI	Series connection- B contact	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The ANI instruction is used in the series connection of B contact. The functions are to read out the status of present designated series connection contacts and perform the “AND” operation with the logical operation result obtained. The final result will be store in the accumulative register.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	X1	Loading in contact A of X1
ANI	**X0**	Connecting to contact B of X0 in series
OUT	Y1	Driving Y1 coil

Note: DVP-EH3 series PLCs whose version is 1.40 and DVP-SV2 series PLCs whose version is 1.20 support the operands X, Y, M, and S. These operands can be qualified by E or F, e.g. ANI X0E1.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
OR	Parallel connection- A contact	1	✓	✓	✓	✓

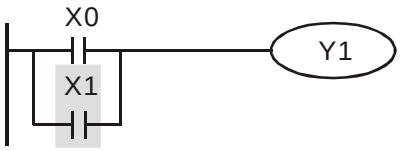
Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The OR instruction is used in the parallel connection of A contact. The functions are to read out the status of present designated parallel connection contacts and perform the “OR” operation with the logical operation result obtained. The final result will be store in the accumulative register.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in contact A of X0
OR	**X1**	Connecting to contact A of X1 in parallel
OUT	Y1	Driving Y1 coil

Note: DVP-EH3 series PLCs whose version is 1.40 and DVP-SV2 series PLCs whose version is 1.20 support the operands X, Y, M, and S. These operands can be qualified by E or F, e.g. OR X1E1.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
ORI	Parallel connection- B contact	1	✓	✓	✓	✓

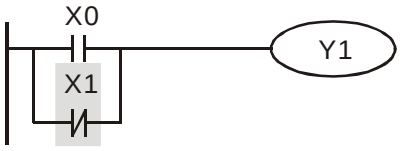
Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The OR instruction is used in the parallel connection of B contact. The functions are to read out the status of present designated parallel connection contacts and perform the “OR” operation with the logical operation result obtained. The final result will be store in the accumulative register.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in contact A of X0
ORI	**X1**	Connecting to contact B of X1 in parallel
OUT	Y1	Driving Y1 coil

Note: DVP-EH3 series PLCs whose version is 1.40 and DVP-SV2 series PLCs whose version is 1.20 support the operands X, Y, M, and S. These operands can be qualified by E or F, e.g. ORI X1E1.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
ANB	Series connection- loop blocks	1	✓	✓	✓	✓

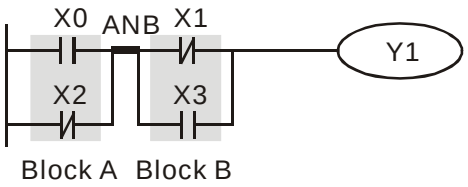
Operand	N/A

Explanations:

To perform the “AND” operation of the preserved logic results and content in the accumulative register.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in contact A of X0
ORI	X2	Connecting to contact B of X2 in parallel
LDI	X1	Loading in contact B of X1
OR	X3	Connecting to contact A of X3 in parallel
ANB		Connecting circuit block in series
OUT	Y1	Driving Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
ORB	Parallel connection- loop blocks	1	✓	✓	✓	✓

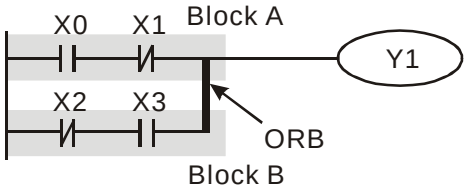
Operand	N/A

Explanations:

To perform the “OR” operation of the preserved logic results and content in the accumulative register.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in contact A of X0
ANI	X1	Connecting to contact B of X1 in series
LDI	X2	Loading in contact B of X2
AND	X3	Connecting to contact A of X3 in series
ORB		Connecting circuit block in parallel
OUT	Y1	Driving Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
MPS	Store the current result of the internal PLC operations	1	✓	✓	✓	✓

Operand	N/A

Explanations:

To save the content in the accumulative register into the operational result (the pointer of operational result will plus 1).

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
MRD	Reads the current result of the internal PLC operations	1	✓	✓	✓	✓

Operand	N/A

Explanations:

To read the operational result and store it into the accumulative register (the pointer of operational result stays intact).

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
MPP	Pops (recalls and removes) the currently stored result	1	✓	✓	✓	✓

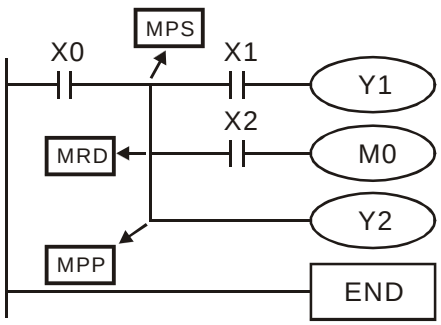
Operand	N/A

Explanations:

To retrieve the previous preserved logical operation result and store it into the accumulative register (the pointer of operational result will minus 1).

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in contact A of X0
MPS		Saving into stack
AND	X1	Connecting to contact A of X1 in series
OUT	Y1	Driving Y1 coil
MRD		Reading from stack
AND	X2	Connecting to contact A of X2 in series
OUT	M0	Driving M0 coil
MPP		Reading from stack and pop pointer
OUT	Y2	Driving Y2 coil
END		Program ends

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
OUT	Output coil	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	-	✓	✓	✓	-	-	-

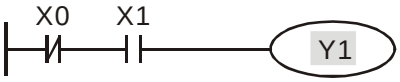
Explanations:

1. To output the logical operation result before OUT instruction into a designated device.
2. Actions of coil contact:

Operational result	OUT instruction		
	Coil	Contact	
		A contact (normally open)	B contact (normally closed)
FALSE	Off	Off	On
TRUE	On	On	Off

Program Example:

Ladder diagram:



Instruction code:		Operation:
LDI	X0	Loading in contact B of X0
AND	X1	Connecting to contact A of X1 in series
OUT	**Y1**	Driving Y1 coil

Note: DVP-EH3 series PLCs whose version is 1.40 and DVP-SV2 series PLCs whose version is 1.20 support the operands Y, M, and S. These operands can be qualified by E or F, e.g. OUT Y1E2.

Mnemonic	Function	Program Steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
SET	Latched (On)	1	✓	✓	✓	✓

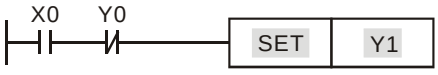
Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	-	✓	✓	✓	-	-	-

Explanations:

When the SET instruction is driven, its designated device will be “On” and keep being On both when SET instruction is still being driven or not driven. Use RST instruction to set “Off” the device.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in contact A of X0
ANI	Y0	Connecting to contact B of Y0 in series
SET	**Y1**	Y1 latched (On)

Note: DVP-EH3 series PLCs whose version is 1.40 and DVP-SV2 series PLCs whose version is 1.20 support the operands Y, M, and S. These operands can be qualified by E or F, e.g. SET Y1E2.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
RST	Clear the contacts or the registers	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999	E0 ~ E7 F0 ~ F7
	-	✓	✓	✓	✓	✓	✓	✓

Explanations:

1. When the RST instruction is driven, the actions of the designated devices are:

Device	Status
Y, M, S,	Coil and contact will be set to “Off”
T, C	Present values of the timer or counter will be set to “0”, and the coil and contact will be set to “Off”
D, E, F	The content will be set to “0”.

2. If RST instruction is not being executed, the status of the designated device will stay intact.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	X0	Loading in contact A of X0
RST	**Y5**	Resetting contact Y5

Note: DVP-EH3 series PLCs whose version is 1.40 and DVP-SV2 series PLCs whose version is 1.20 support the operands Y, M, and S. These operands can be qualified by E or F, e.g. RST Y5E2.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
TMR	16-bit timer	4	✓	✓	✓	✓

Operand		
	T-K	T0 ~ T255, K0 ~ K32,767
	T-D	T0 ~ T255, D0 ~ D9999

Explanations:

When TMR instruction is executed, the designated coil of the timer will be On and the timer will start to time. When the set value in the timer is reached (present ≥ set value), the contact will be:

NO (Normally Open) contact	Open collector
NC (Normally Closed) contact	Close collector

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	X0	Loading in contact A of X0 T5 timer
TMR	**T5 K1000**	Set value in timer T5 as K1,000

Remarks:

See the specification of each model for the range of operand T.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
ATMR	16-bit contact type timer counter	5	—	—	—	✓

Operand		
	T-K	T0 ~ T255,K0 ~ K32,767
	T-D	T0 ~ T255, D0 ~ D11999

Explanations:

1. The instruction ATMR corresponds to the combination of AND and TMR. If the contact preceding ATMR is ON, the timer specified will begin to count. When the count value is greater than or equal to the setting value, the AND contact is ON. If the contact preceding ATMR is not ON, ATMR will automatically clear the count value.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	X0	Loading in contact A of X0
ATMR	**T5 K100**	The setting value of T5 is K100.
OUT	**Y0**	When the count value is greater than or equal to the setting value, Y0 is ON.

Remarks:

1. Please refer to specifications for the model used for more information about the timers which can be used.
2. EH3 series PLCs whose version is 1.40 and SV2 series PLCs whose version is 1.20 support ATMR.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
CNT	16-bit counter	1	✓	✓	✓	✓

Operand		
	C-K	C0 ~ C199, K0~K32,767
	C-D	C0 ~ C199, D0 ~ D9999

Explanations:

3. When the CNT instruction goes from Off to On, the designated counter coil will be driven, and the present value in the counter will plus 1. When the counting reaches the set value (present value = set value), the contact will be:

NO (Normally Open) contact	Open collector
NC (Normally Closed) contact	Close collector

4. If there are other counting pulse inputs after the counting reaches its target, the contact and present value will stay intact. Use RST instruction to restart or reset the counting.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	X0	Loading in contact A of X0
CNT	**C20 K100**	Set value in counter C20 as K100

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
DCNT	32-bit counter	1	✓	✓	✓	✓

Operand	C-K	C200 ~ C255, K-2,147,483,648 ~ K2,147,483,647
	C-D	C200 ~ C255, D0 ~ D9999

Explanations:

1. DCNT is the instruction for enabling the 32-bit high-speed counters C200 ~ C255.
2. For general purpose addition/subtraction counters C200 ~ C234, when DCNT goes from Off to On, the present value in the counter will pulse 1 (counting up) or minus 1 (counting down) according to the modes set in special M1200 ~ M1235.
3. For high-speed addition/subtraction counters C235 ~ C255, when the high-speed counting pulse input goes from Off to On, the counting will start its execution. For the input terminals (X0 ~ X17) and counting methods (counting up/down) of the high-speed counter, see Chapter 2.7 Numbering and Function of Counter [C] for more details.
4. When DCNT is Off, the counting will stop, but the existing present value in the counter will not be cleared. To clear the present value and the contact, you have to use the instruction RST C2XX. Use externally designated input points to clear the present values and contacts of high-speed addition/subtraction counters C235 ~ C255.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	M0	Loading in contact A of M0
DCNT	**C254 K1000**	Set value of counter C254 as K1,000

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
MC / MCR	Master control Start/Reset	1	✓	✓	✓	✓

Operand	N0 ~ N7

Explanations:

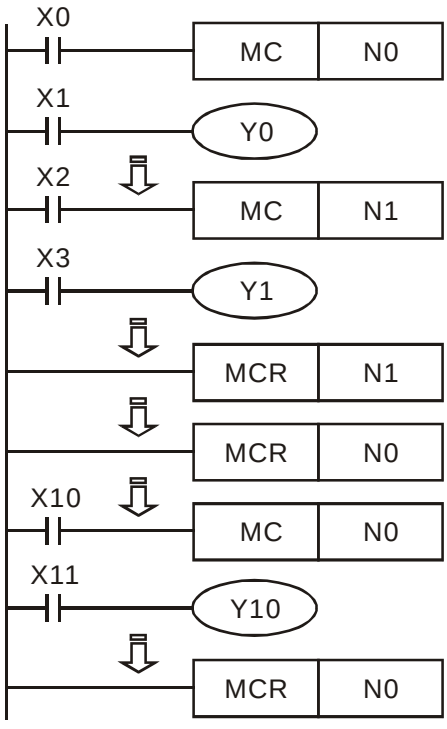
1. MC is the main-control start instruction. When MC instruction is executed, the execution of instructions between MC and MCR will not be interrupted. When MC instruction is Off, the actions of the instructions between MC and MCR are:

Instruction type	Explanation
General purpose timer	Present value = 0 Coil is Off, No action for the contact
Accumulative timer	Coil is Off, present value and contact stay intact
Subroutine timer	Present value = 0 Coil is Off, No action for the contact
Counter	Coil is Off, present value and contact stay intact
Coils driven by OUT instruction	All Off
Devices driven by SET and RST instructions	Stay intact
Application instructions	All disabled. The FOR-NEXT nested loop will still execute back and forth for N times. Instructions between FOR-NEXT will act as the instructions between MC and MCR.

2. MCR is the main-control end instruction that is placed in the end of the main-control program. There should not be any contact instructions prior to MCR instruction.
3. MC-MCR main-control program instructions support the nested program structure (max. 8 layers) and please use the instruction in the order N0 ~ N7.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in A contact of X0
MC	**N0**	Enabling N0 common series connection contact
LD	X1	Loading in A contact of X1
OUT	Y0	Driving Y0 coil
:		
LD	X2	Loading in A contact of X2
MC	**N1**	Enabling N1 common series connection contact
LD	X3	Loading in A contact of X3
OUT	Y1	Driving Y1 coil
:		
MCR	**N1**	Disabling N1 common series connection contact
:		
MCR	**N0**	Disabling N0 common series connection contact
:		
LD	X10	Loading in A contact of X10
MC	**N0**	Enabling N0 common series connection contact
LD	X11	Loading in A contact of X11
OUT	Y10	Driving Y10 coil
:		
MCR	**N0**	Disabling N0 common series connection contact

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
LDP	Rising-edge detection operation	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The method of using LDP is the same as using LD, but the actions of the two instructions differ. LDP saves the current content and store the detected status of rising-edge to the accumulative register.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LDP	**X0**	Starting X0 rising-edge detection
AND	X1	Series connecting A contact of X1
OUT	Y1	Driving Y1 coil

Remarks:

1. See the specification of each model for the range of operands.
2. If the status of a designated rising-edge is On before the PLC is powered, the contact of the rising-edge will be TRUE after PLC is powered.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
LDF	Falling-edge detection operation	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The method of using LDF is the same as using LD, but the actions of the two instructions differ. LDF saves the current content and store the detected status of falling-edge to the accumulative register.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LDF	**X0**	Starting X0 falling-edge detection
AND	X1	Series connecting A contact of X1
OUT	Y1	Driving Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
ANDP	Rising-edge series connection	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

ANDP instruction is used in the series connection of the contacts' rising-edge detection.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	X0	Loading in A contact of X0
ANDP	**X1**	X1 rising-edge detection in series connection
OUT	Y1	Driving Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
ANDF	Falling-edge series connection	1	✓	✓	✓	✓

Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

ANDF instruction is used in the series connection of the contacts' falling-edge detection.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	X0	Loading in A contact of X0
ANDF	**X1**	X1 falling-edge detection in series connection
OUT	Y1	Drive Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
ORP	Rising-edge parallel connection	1	✓	✓	✓	✓

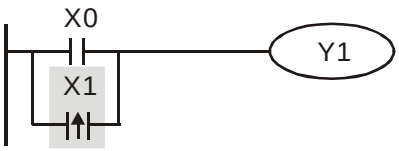
Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The ORP instructions are used in the parallel connection of the contact's rising-edge detection.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in A contact of X0
ORP	**X1**	X1 rising-edge detection in parallel connection
OUT	Y1	Driving Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
ORF	Falling-edge parallel connection	1	✓	✓	✓	✓

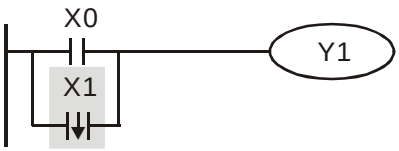
Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

The ORP instructions are used in the parallel connection of the contact's falling-edge detection.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in A contact of X0
ORF	**X1**	X1 falling-edge detection in parallel connection
OUT	Y1	Driving Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
PLS	Rising-edge output	1	✓	✓	✓	✓

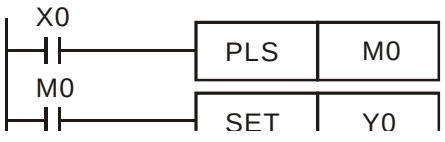
Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	-	✓	✓	-	-	-	-

Explanations:

When X0 goes from Off to On (rising-edge trigger), PLS instruction will be executed and **S** will send out pulses for once of 1 scan time.

Program Example:

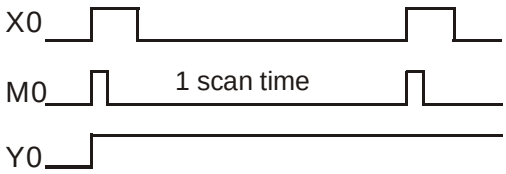
Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in A contact of X0
PLS	**M0**	M0 rising-edge output
LD	M0	Loading in contact A of M0

Timing Diagram:

SET	Y0	Y0 latched (On)



Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
PLF	Falling-edge output	1	✓	✓	✓	✓

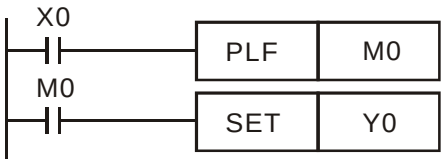
Operand	X0 ~ X377	Y0 ~ Y377	M0 ~ M4095	S0 ~ S1023	T0 ~ T255	C0 ~ C255	D0 ~ D9999
	-	✓	✓	-	-	-	-

Explanations:

When X0 goes from On to Off (falling-edge trigger), PLF instruction will be executed and **S** will send out pulses for once of 1 scan time.

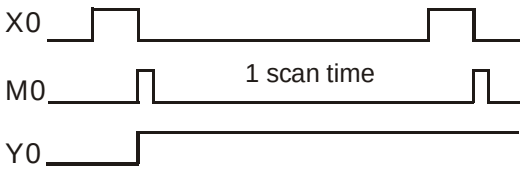
Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in A contact of X0
PLF	**M0**	M0 falling-edge output
LD	M0	Loading in contact A of M0
SET	Y0	Y0 latched (On)

Timing Diagram:



Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
END	Program End	1	✓	✓	✓	✓

Operand	N/A

Explanations:

END instruction has to be placed in the end of a ladder diagram or instruction program. PLC will start to scan from address 0 to END instruction and return to address 0 to restart the scan.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
NOP	No operation	1	✓	✓	✓	✓

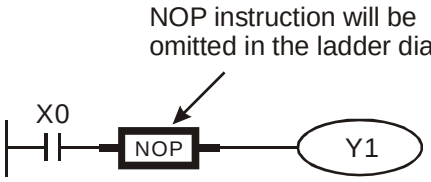
Operand	N/A

Explanations:

NOP instruction does not conduct any operations in the program; therefore, after the execution of NOP, the existing logical operation result will be kept. If you want to delete a certain instruction without altering the length of the program, you can use NOP instruction.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in B contact of X0
NOP		No operation
OUT	Y1	Driving Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
INV	Inverting Operation	1	✓	✓	✓	✓

Operand	N/A

Explanations:

The logical operation result before INV instruction will be inverted and stored in the accumulative register.

Program Example:

Ladder diagram:

Instruction code:		Operation:
LD	X0	Loading in A contact of X0
INV		Inverting the operation result
OUT	Y1	Driving Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
P	Pointer	1	✓	✓	✓	✓

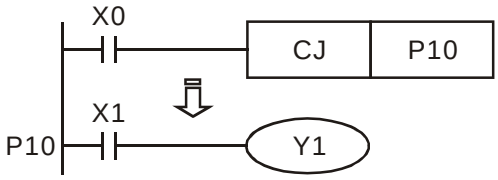
Operand	P0 ~ P255

Explanations:

Pointer P is used in API 00 CJ and API 01 CALL instructions. The use of P does not need to start from No. 0, and the No. of P cannot be repeated; otherwise, unexpected errors may occur.

Program Example:

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in A contact of X0
CJ	P10	From instruction CJ to P10
:		
P10		Pointer P10
LD	X1	Loading in A contact of X1
OUT	Y1	Driving Y1 coil

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3
I	Interruption program marker (I)	1	✓	✓	✓	✓

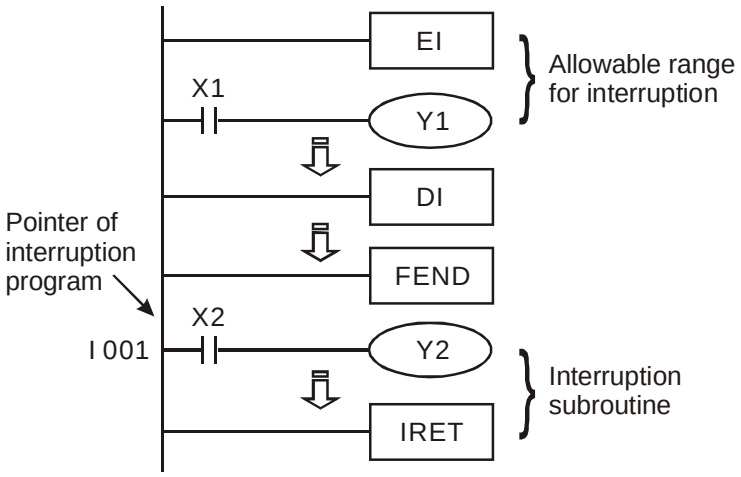
Operand	I00□, I10□, I20□, I30□, I40□, I50□, I6□□, I7□□, I8□□ I010, I020, I030, I040, I050, I060, I110, I120, I130, I140, I150, I160, I170, I180

Explanations:

A interruption program has to start with a interruption pointer (I□□□) and ends with API 03 IRET. I instruction has to be used with API 03 IRET, API 04 EI, and API 05 DI. See Chapter 2.9 for pointers of all DVP series PLCs.

Program Example:

Ladder diagram:



Instruction code:		Operation:
EI		Enabling interruption
LD	X1	Loading A contact of X1
OUT	Y1	Driving Y1 coil
:		
DI		Disabling interruption
:		
FEND		Main program ends
I001		Interruption pointer
LD	X2	Loading in A contact of X2
OUT	Y2	Driving Y2 coil
:		
IRET		Interruption return

4.1 Step Ladder Instructions [STL], [RET]

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3/SV2
STL	Step Transition Ladder Start	1	✓	✓	✓	✓

Operand	S0 ~ S1023

Explanations:

STL Sn constructs a step. When STL instruction appears in the program, the program will enter a step ladder diagram status controlled by steps. The initial status has to start from S0 ~ S9. RET instruction indicates the end of a step ladder diagram starting from S0 ~ S9 and the bus returns to a normal ladder diagram instruction. SFC uses the step ladder diagram composed of STL/RET to complete the action of a circuit. The No. of S cannot be repeated.

Mnemonic	Function	Program steps	Controllers			
			ES/EX/EC	EC3-8K	SX	EH3/SV2
RET	Step Transition Ladder Return	1	✓	✓	✓	✓

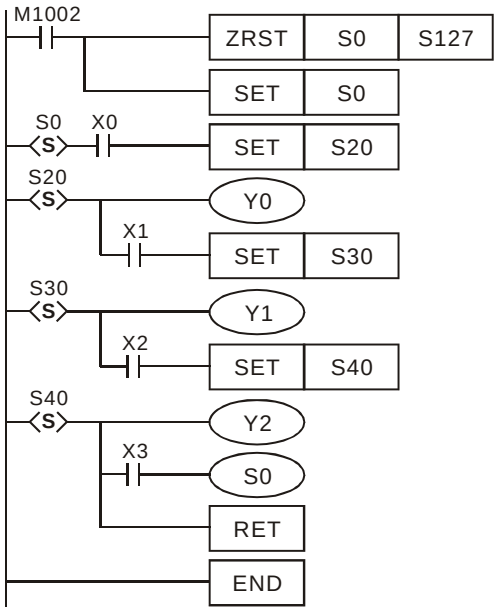
Operand	N/A

Explanations:

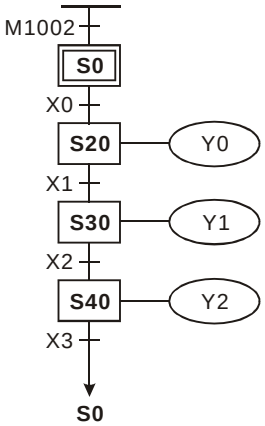
RET indicates the end of a step. There has to be a RET instruction in the end of a series of steps. One PLC program can be written in maximum 10 steps (S0 ~ S9) and every step should end with a RET.

Program Example:

Ladder diagram:



SFC:



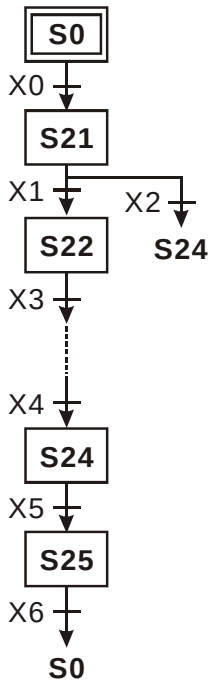
4.2 Sequential Function Chart (SFC)

In automation, we always need electric control to work with mechanical control for an automation control. The sequential control can be divided into several orderly steps (or stages). Each step has its actions that should be completed and the transition from one step to another normally requires some criteria. The action of the last step finishes when all criteria is true and the beginning of the next step will clear the actions of the last step. This is the concept of designing a sequential function chart (SFC).

Features:

1. No sequential design is required for constant step actions, and PLC will automatically execute the interlocking and dual outputs among all status. An easy sequential design is the only thing required to make the machine work normally.
2. The actions in SFC are easy to understand, adjust for a trial operation, detect the errors and maintain.
3. SFC is a type of diagram editing. The structure of a SFC looks like a flow chart. Every No. of the step relay S inside the PLC represents a step, equal to every processing procedure in a flow chart. When the current procedure is completed, the program will move to the next step according to the set transition criteria. Therefore, you can repeat the cycle and obtain the result you desire.
4. See the SFC chart in the right hand side: The initial step S0 transfers to a general purpose step S21 by making the status transition condition X0 condition true. S21 transfer to S22 or jumps to S24 by making X1 or X2 true. In step S25, X6 will be true and the chart will return to S0 to complete a cycle. The cycle and be repeated to reach a cyclic control.
5. Next are some basic icons for drawing SFC in WPLSoft SFC editor.

SFC:



LAD	Ladder diagram mode. The icon indicates that the internal editing program is a general ladder diagram, not a step ladder program.
	Initial step in SFC. Applicable for S0 ~ S9.
	General step. Applicable for S10 ~ S1023.
	Step jumps. Used for a step to jump to another non-adjacent step. (Jumping up/down to non-adjacent steps in the same sequence, returning to initial step, or jumping among different sequences.)
	Transition condition between steps.
	Alternative divergence. The same step transfers to a corresponding step by different transition condition.

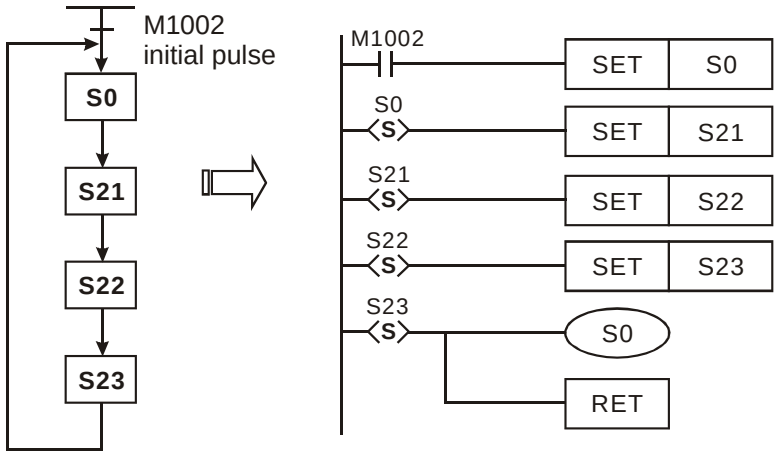
	Alternative convergence. More than 2 steps transfer to the same step by transition condition.
	Simultaneous divergence. The same step transfers to more than 2 steps by the same transition condition.
	Simultaneous convergence. More than 2 steps transfer to the same step by a single transition condition.

4.3 How does a Step Ladder Instruction Work?

STL instruction is used for designing the syntax of a sequential function chart (SFC), making the program designing similar to drawing a flow chart and allowing a more explicit and readable program. From the figure in the left hand side below, we can see very clearly the sequence to be designed, and we can convert the sequence into the step ladder diagram in the right hand side.

RET instruction has to be written at the end of every step sequence, representing the end of a sequence. There can be more than one step sequence in a program. Therefore, we have to write in RET at the end of every step sequence. There is no limitation on the times of using RET which is used together with S0 ~ S9.

If there is no RET instruction at the end of a step sequence, errors will be detected by WPL editor.



1. Actions of Step Ladder:

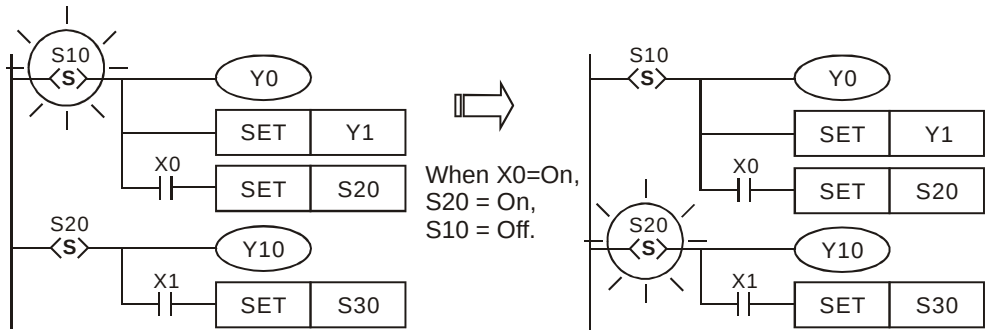
A step ladder is composed of many steps and every step controls an action in the sequence. The step ladder has to:

a) Drive the output coil

b) Designate the transition condition

c) Designate which step will take over the control from the current step

Example:

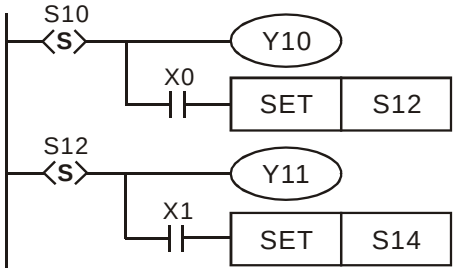


Explanation:

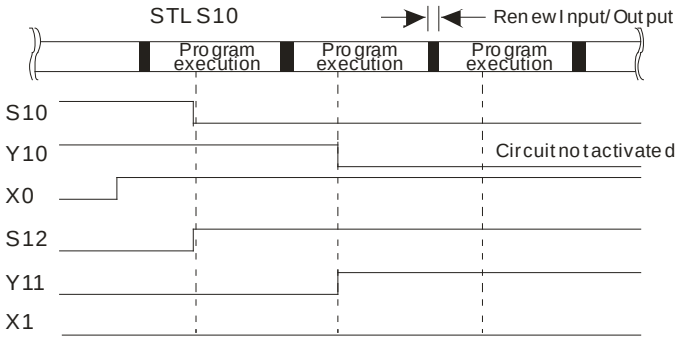
When S10 = On, Y0 and Y1 will be On. When X0 = On, S20 will be On and Y10 will be On. When S10 = Off, Y0 will be Off and Y1 will be On.

2. Timing Diagram of Step Ladder:

When the status contact Sn = On, the circuit will be activated. When Sn = Off, the circuit will be inactivated. The actions will delay for 1 scan time.



Executing the timing diagram below. After the status of S10 and S12 are transferred (taking place simultaneously), and after a delay of 1 scan time, Y10 will be Off and Y11 will be On. There will not be overlapping outputs.



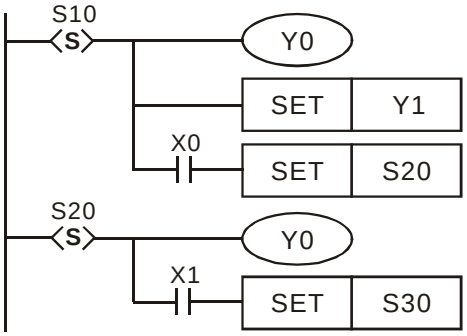
3. Repeated Use of Output Coil:

a) You can use output coils of the same No. in different steps.

b) See the diagram in the right. There can be the same output device (Y0) among different statuses. Y0 will be On when S10 or S20 is On. Such as right diagram, there is the same output device Y0 in the different state. No matter S10 or S20 is On, Y0 will be On.

c) Y0 will be Off when S10 is transferring to S20. After S20 is On, Y0 will output again. Therefore in this case, Y0 will be On when S10 or S20 is On.

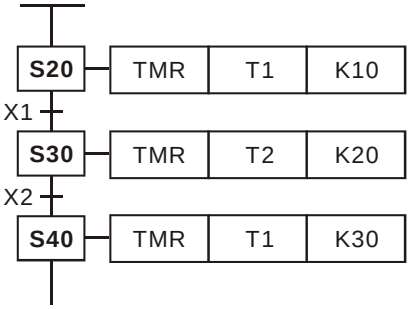
d) Normally in a ladder diagram, avoid repeated use of an output coil. The No. of output coil used by a step should also avoid being used when the step ladder diagram returns to a general ladder diagram.



4. Repeated Use of Timer:

The timers in EH2/SV series MPU are the same as general output points and can be repeatedly used in different steps. This is one of the features of the step ladder diagram. However, in a general ladder diagram, it is better not be repeatedly use the output coil. Also avoid using the No. of the output coil used by a step after the step ladder diagram returns to a general ladder diagram.

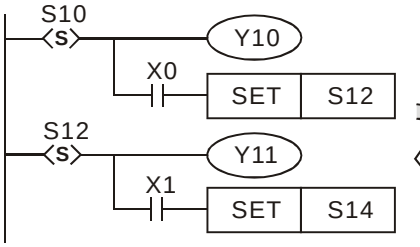
Note: See the figure in the right. The timers in ES/EX/SS/SA/SX/SC series MPU can be used repeatedly in non-adjacent steps.



5. Transfer of Step:

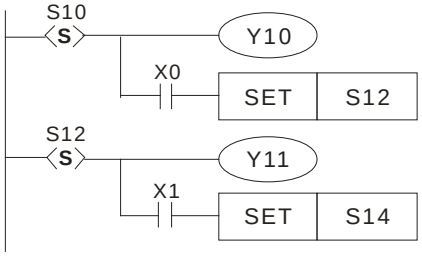
SET Sn and OUT Sn instructions are used to enable (or transfer to) another step. When the control power is shifted to another step, the status of the previous step S and the action of the output point will be cleared. Due to that there can be many step control sequences (i.e. the step ladder diagram starting with S0 ~ S9) co-existing in the program. The transfer of a step can take place in the same step sequence, or be transferred to different step sequence. Therefore, there are some slight differences regarding how to use SET Sn and OUT Sn. See the explanations below.

SET Sn Used for driving the next step in the same sequence. After the transition, all output from the previous status will be cleared.



When SET S12 instruction is executed, S10 will transfer to S12, and S10 and all its outputs (Y10) will be cleared.

If M1014 is used, and it is On, the transfer of the steps will be prohibited, and the states of the steps remain unchanged.



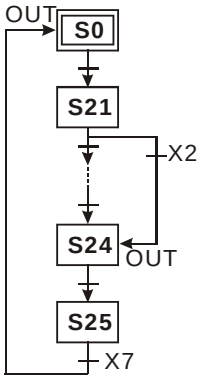
If M1040 is On, SET S12 instruction will not be executed, the state of S10 unchanged, and Y10 will be On.

OUT Sn Used for returning to the initial step in the same step sequence. Also for jumping up/down to non-adjacent steps in the same sequence, or separating steps in different sequences. After the transition, all output from the previous status will be cleared.

① Returning to the initial step in the same sequence.

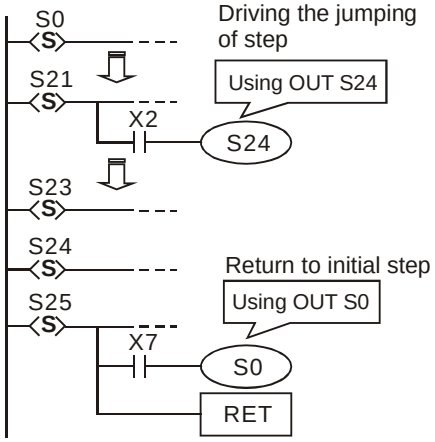
② Jumping up/down to non-adjacent steps in the same sequence.

SFC:



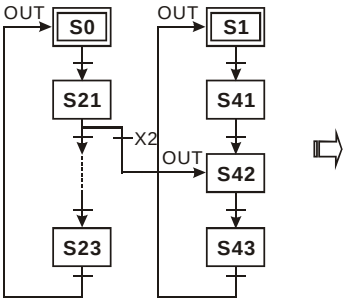
S25 returns to the initial step S0 by using OUT.

Ladder diagram:



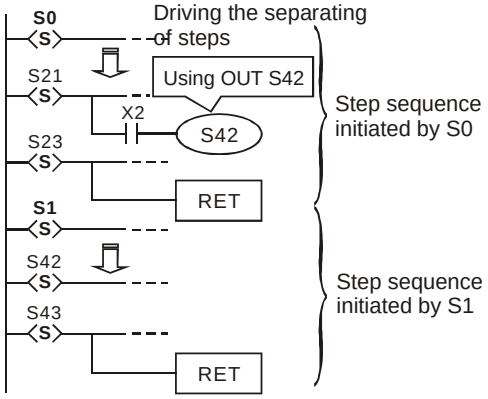
③ Separating steps in different sequences.

SFC:



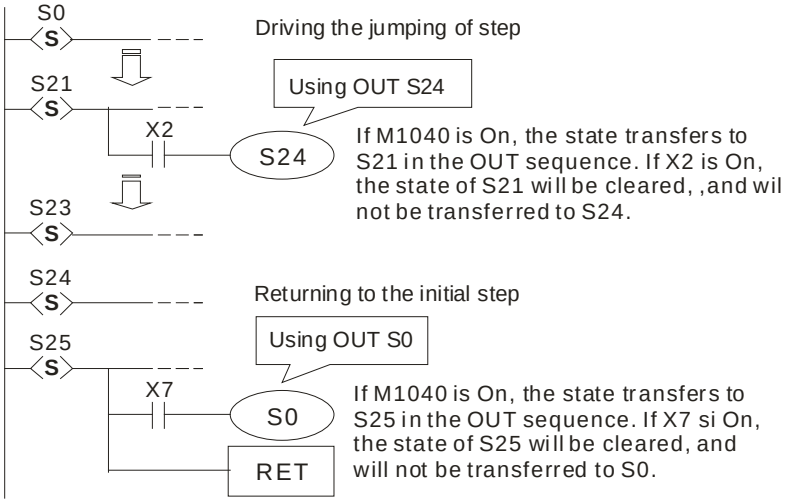
Two different step sequence: S0 and S1
S23 returns to initial step S0 by using OUT.
S43 returns to initial step S1 by using OUT.

Ladder diagram:



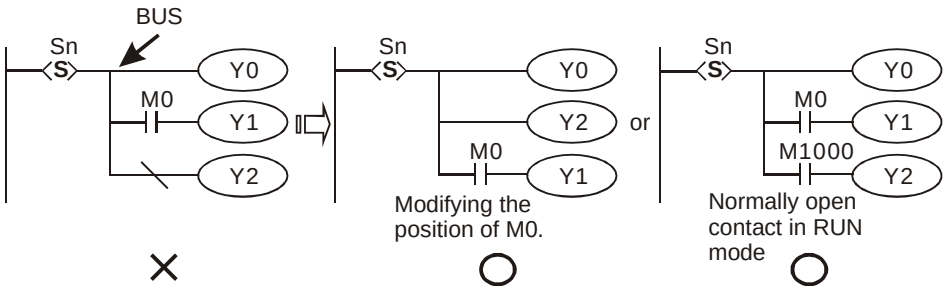
If M1014 is used, and M1040 is On, the steps in the same sequence will be cleared to Off.

Ladder diagram:



6. Cautions for Driving Output Point:

See the figure below. After the step point and once LD or LDI instructions are written into the second line, the bus will not be able to connect directly to the output coil, and errors will occur in the compilation of the ladder diagram. You have to correct the diagram into the diagram in the right hand side for a correct compilation.



7. Restrictions on Using Some Instructions:

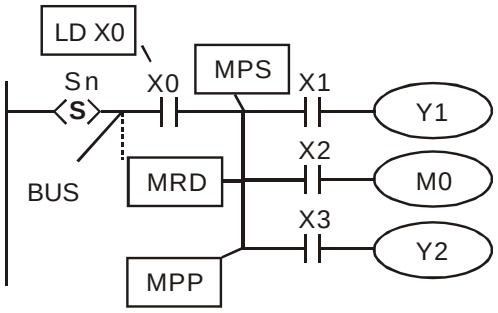
The program of every step is the same as a general ladder diagram, in which you can use all kinds of series/parallel circuits or instructions. However, there are restrictions on some of the instructions.

Basic instructions applicable in a step

Step \ Instruction		LD/LDI/LDP/LDF AND/ANI/ANDP/ANDF OR/ORI/ORP/ORF INV/OUT/SET/RST	ANB/ORB MPS/MRD/MPP	MC/MCR
Initial step/general step		Yes	Yes	No
Divergence/ convergence step	General output	Yes	Yes	No
	Step transfer	Yes	No	No

- DO NOT use MC/MCR instruction in the step.
- DO NOT use STL instruction in a general subroutine or interruption subroutine.
- You can still use CJ instruction in STL instruction, but this will make the actions more complicated. We do not recommend you do so.
- The position of MPS/MRD/MPP instruction:

Ladder diagram:



Instruction code:

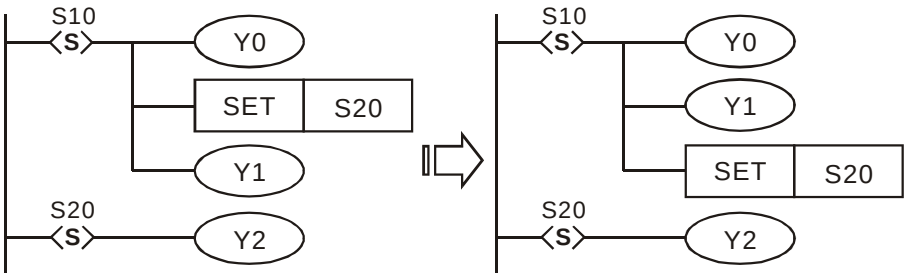
STL	Sn
LD	X0
MPS	
AND	X1
OUT	Y1
MRD	
AND	X2
OUT	M0
MPP	
AND	X3
OUT	Y2

Explanation:

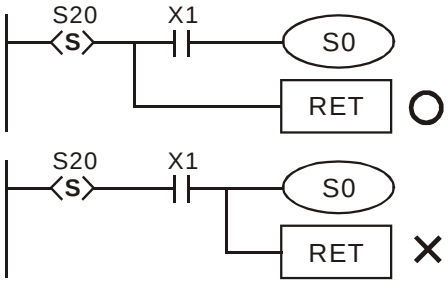
MPS/MRD/MPP instruction cannot be used directly on the new bus. You have to execute LD or LDI instruction first before applying MPS/MRD/MPP.

8. Other Points to Note:

The instruction used for transferring the step (SET S□ or OUT S□) can only be executed after all the relevant outputs and actions in the current status are completed. See the figure below. The executed results by the PLC are the same, but if there are many conditions or actions in S10, it is recommended that you modify the diagram in the left hand side into the diagram in the right hand side. SET S20 is only executed after all relevant outputs and actions are completed, which is a more explicit sequence.



Make sure to add RET instruction after STL at the end of the step ladder diagram.



4.4 Things to Note for Designing a Step Ladder Program

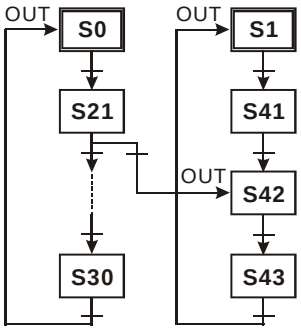
1. The first step in the SFC is called the “initial step", S0 ~ S9. Use the initial step as the start of a sequence and end a complete sequence with RET instruction.
2. If STL instruction is not in use, step S can be a general-purpose auxiliary relay.
3. When STL instruction is in use, the No. of step S cannot be repeated.
4. Types of sequences:

Single sequence: There is only one sequence without alternative divergence, alternative convergence, simultaneous divergence and simultaneous convergence in a program.

Complicated single sequence: There is only one sequence with alternative divergence, alternative convergence, simultaneous divergence and simultaneous convergence in a program.

Multiple sequences: There are more than one sequence in a program, maximum 10 sequences, S0 ~ S9.

5. Separation of sequence: Multiple sequences are allowed to be written into the step ladder diagram.
   a) See the diagram in the right hand side. There are two sequences S0 and S1. The program writes in S0 ~ S30 first and S1 ~ S43 next.
   b) You can designate a step in the sequence to jump to any step in another sequence.
   c) When the condition below S21 is true, the sequence will jump to step S42 in sequence S1, which is called “separating the step”.



6. Restrictions on diverging sequence: See 4.5 for example
   a) You can use maximum 8 diverged steps in a divergence sequence.
   b) You can use maximum 16 loops in multiple divergence sequences or in simultaneous sequences combined into one sequence.
   c) You can designate a step in the sequence to jump to any step in another sequence.
7. Reset of the step and the inhibiting output:
   a) Use ZRST instruction to reset a step to be Off.
   b) Make M1034 = On to inhibit output Y.
8. Latched step:

   The On/Off status of the latched step will be memorized when the power of the PLC is switched off. When the PLC is re-powered, the status before the power-off will be recovered and the execution will resume. Please be aware of the area for the latched steps.

9. Special auxiliary relays and special registers: See 4.6refer to chapter 4.6 IST instruction for more details.

Device No.	Function
M1040	Disabling step. Disabling all the shifting of steps when On.
M1041	Starting step. Flag for IST instruction.
M1042	Enabling pulses. Flag for IST instruction.
M1043	Zero return completed. Flag for IST instruction.

Device No.	Function
M1044	Zero point condition. Flag for IST instruction.
M1045	Disabling all output reset. Flag for IST instruction.
M1046	Setting STL status as On. On when any of the steps is On.
M1047	Enabling STL monitoring
D1040	On status of step No. 1
D1041	On status of step No. 2
D1042	On status of step No. 3
D1043	On status of step No. 4
D1044	On status of step No. 5
D1045	On status of step No. 6
D1046	On status of step No. 7
D1047	On status of step No. 8

4.5 Types of Sequences

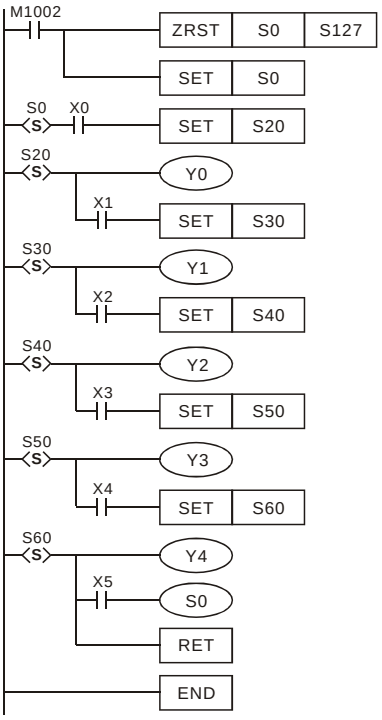
Single Sequence: The basic type of sequences

The first step in a step ladder diagram is called the initial step, which can be S0 ~ S9. The steps following the initial step are general steps, which can be S10 ~ S1023. If you are using IST instruction, S10 ~ S19 will become the steps for zero return.

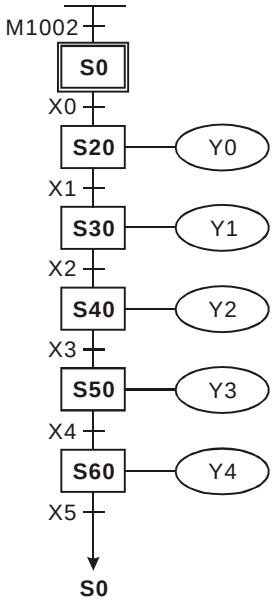
a) Single sequence without divergence and convergence

After a sequence is completed, the control power on the steps will be given to the initial step.

Step ladder diagram

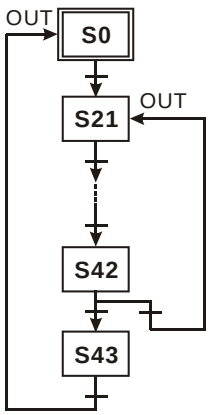


SFC:

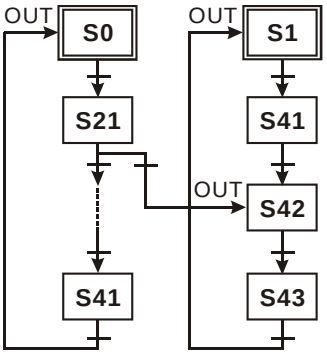


b) Jumping Sequence

1. The control power over the step is transferred to a certain step on top.

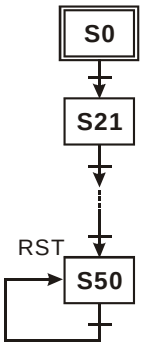


2. The control power over the step is transferred to the step in another sequence.



c) Reset Sequence

See the diagram in the right hand side. When the condition at S50 is true, S50 will be reset and the sequence will be completed at this time.



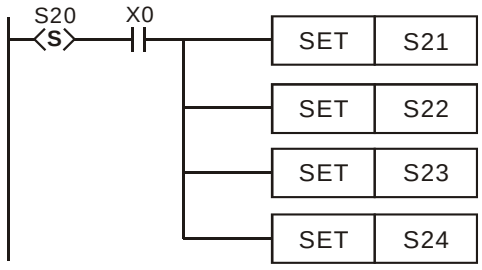
Complicated Single Sequence:

Including simultaneous divergence, alternative divergence, simultaneous convergence and alternative convergence.

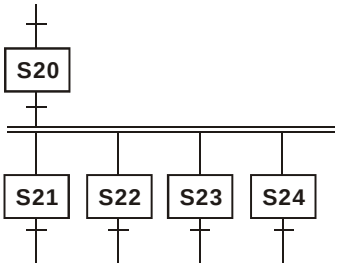
a) Structure of simultaneous divergence

When the condition at the current step is true, the step can be transferred to many steps. See the diagrams below. When X0 = On, S20 will be simultaneously transferred to S21, S22, S23 and S24.

Ladder diagram:



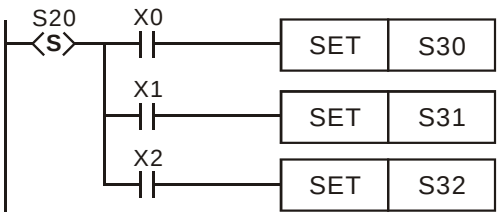
SFC:



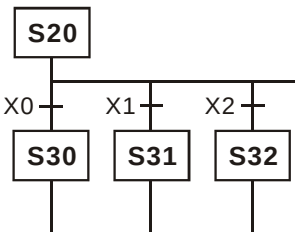
b) Structure of alternative divergence

When the individual condition at the current status is true, the step will be transferred to another individual step. See the diagrams below. When X0 = On, S20 will be transferred to S30; when X1 = On, S20 will be transferred to S31; when X2 = On, S20 will be transferred to S32.

Ladder diagram:



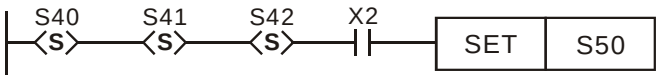
SFC:



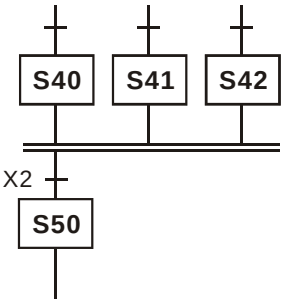
c) Structure of the simultaneous convergence

See the ladder diagram below. A continuous STL instruction represents a simultaneous convergence. When the condition is true after a continuous output, the step will be transferred to the next step. In the simultaneous convergence, only when several conditions are true will the transfer be allowed.

Ladder diagram:



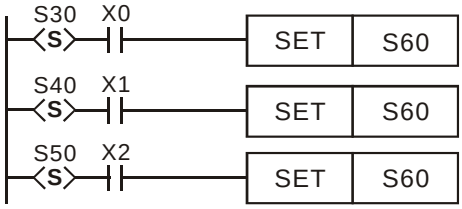
SFC:



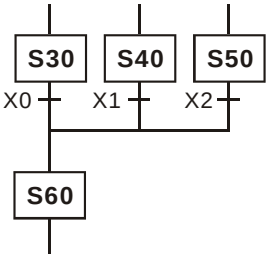
d) Structure of alternative convergence

See the diagrams below. Depending on the condition of the input signal of which of S30, S40 and S50 becomes true first, the first one will be first transferred to S60.

Ladder diagram:

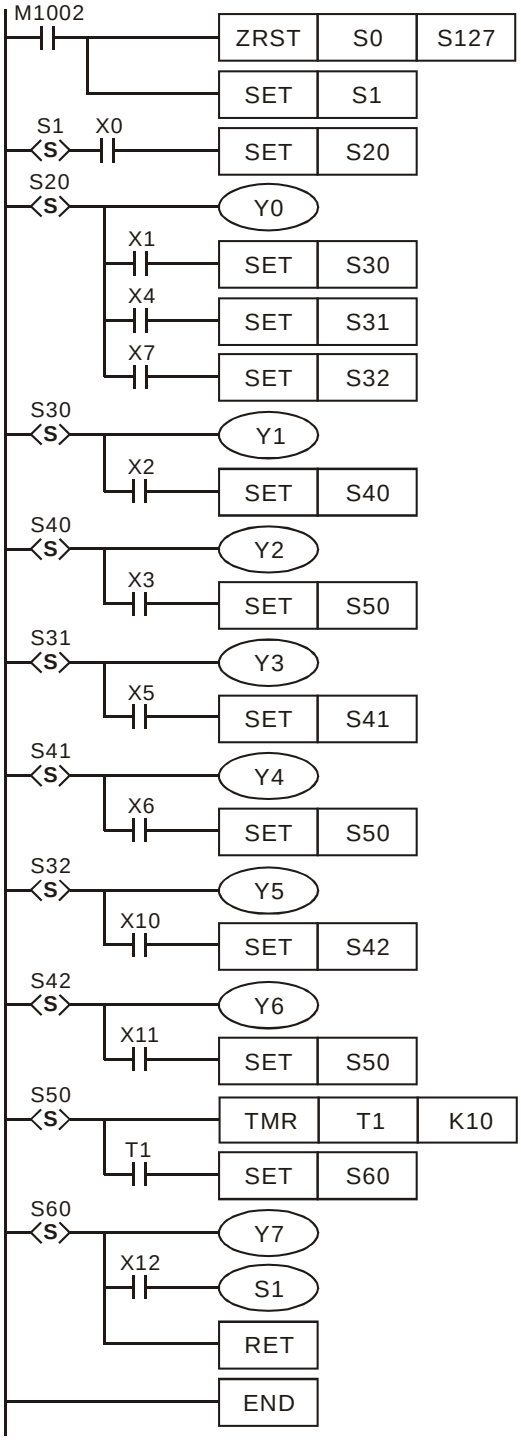


SFC:

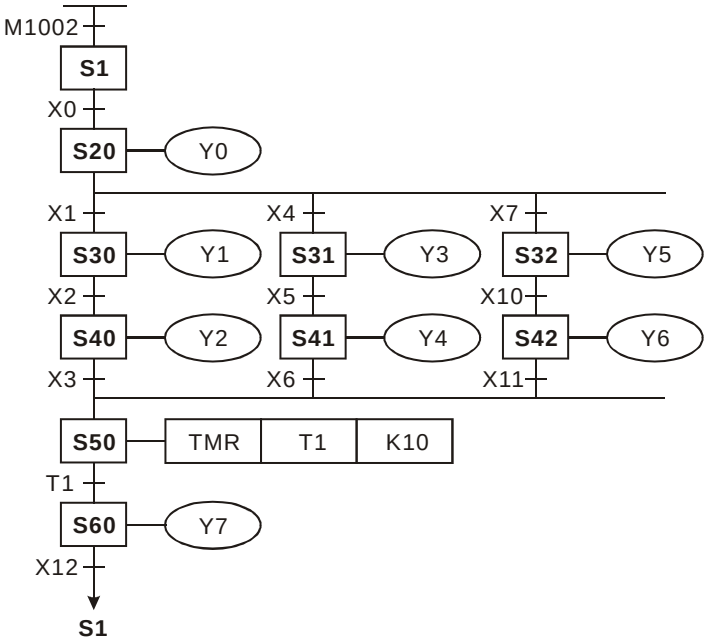


Examples of alternative divergence & alternative convergence:

Ladder diagram:

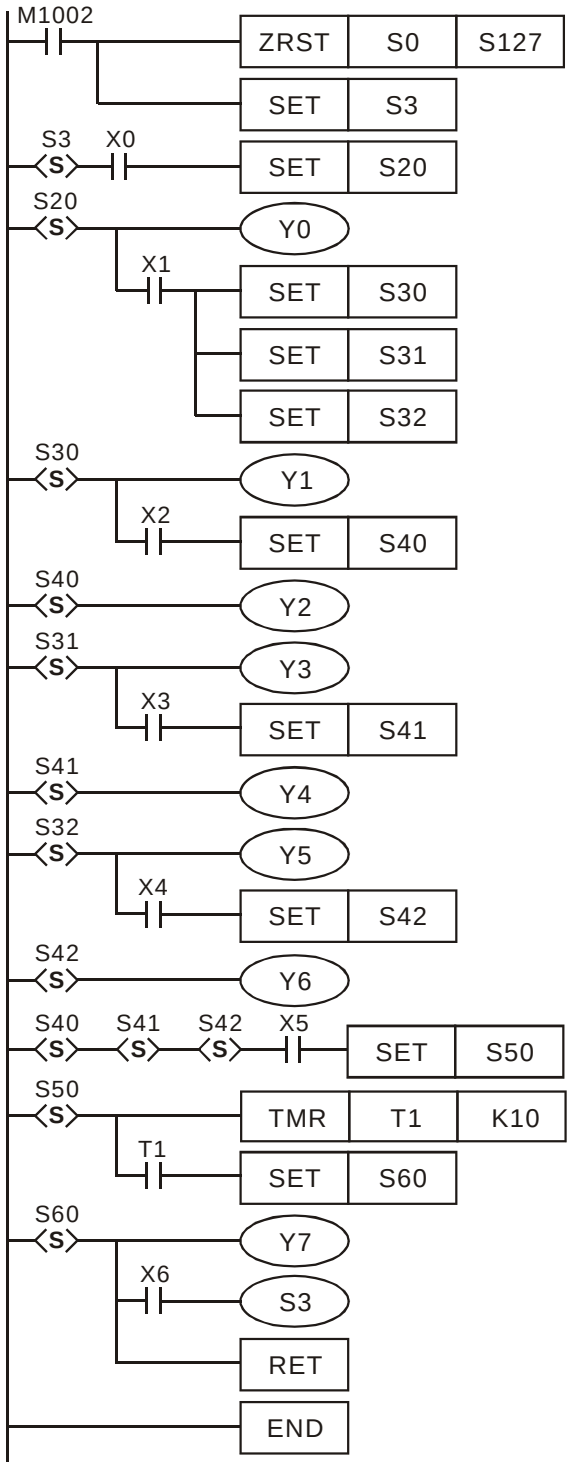


SFC:

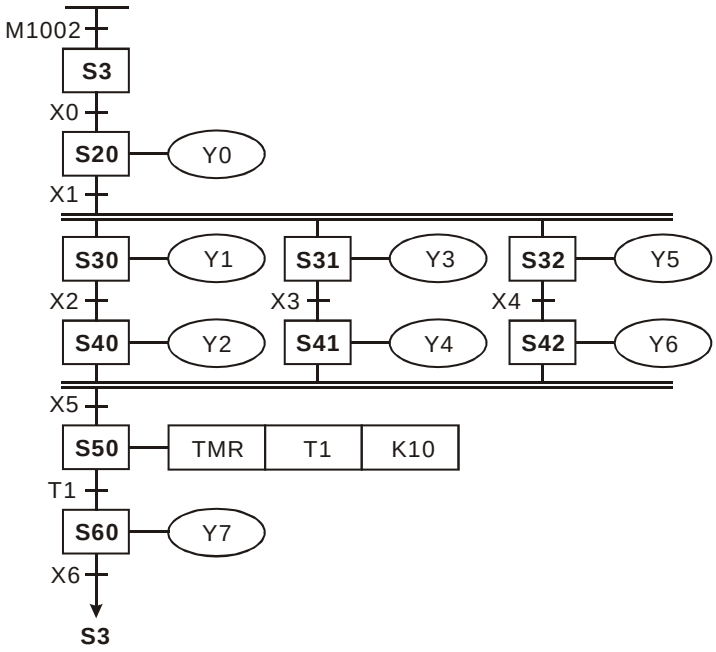


Examples of simultaneous divergence & simultaneous convergence:

Ladder diagram:

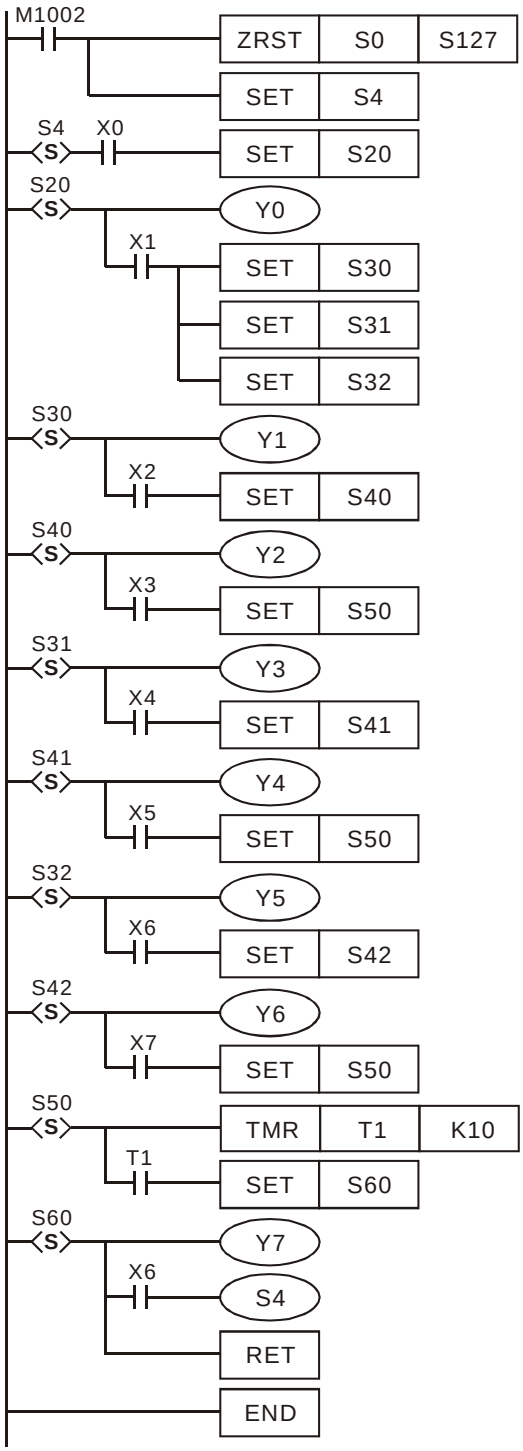


SFC:

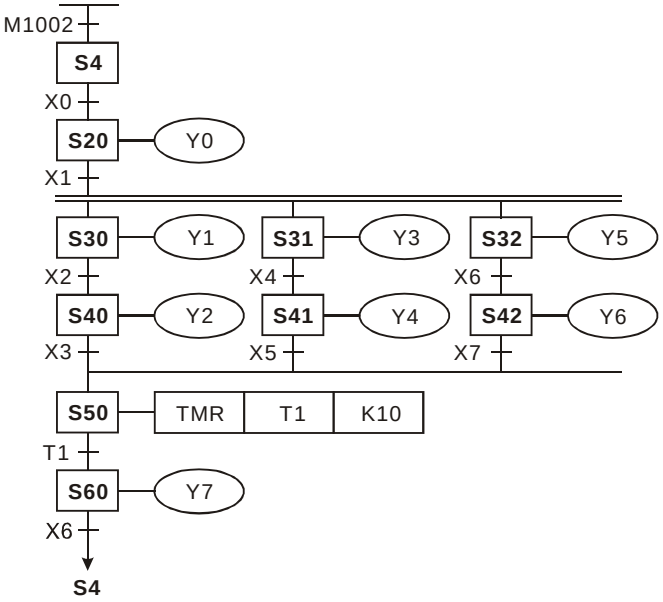


Example of the simultaneous divergence & alternative convergence:

Ladder diagram:



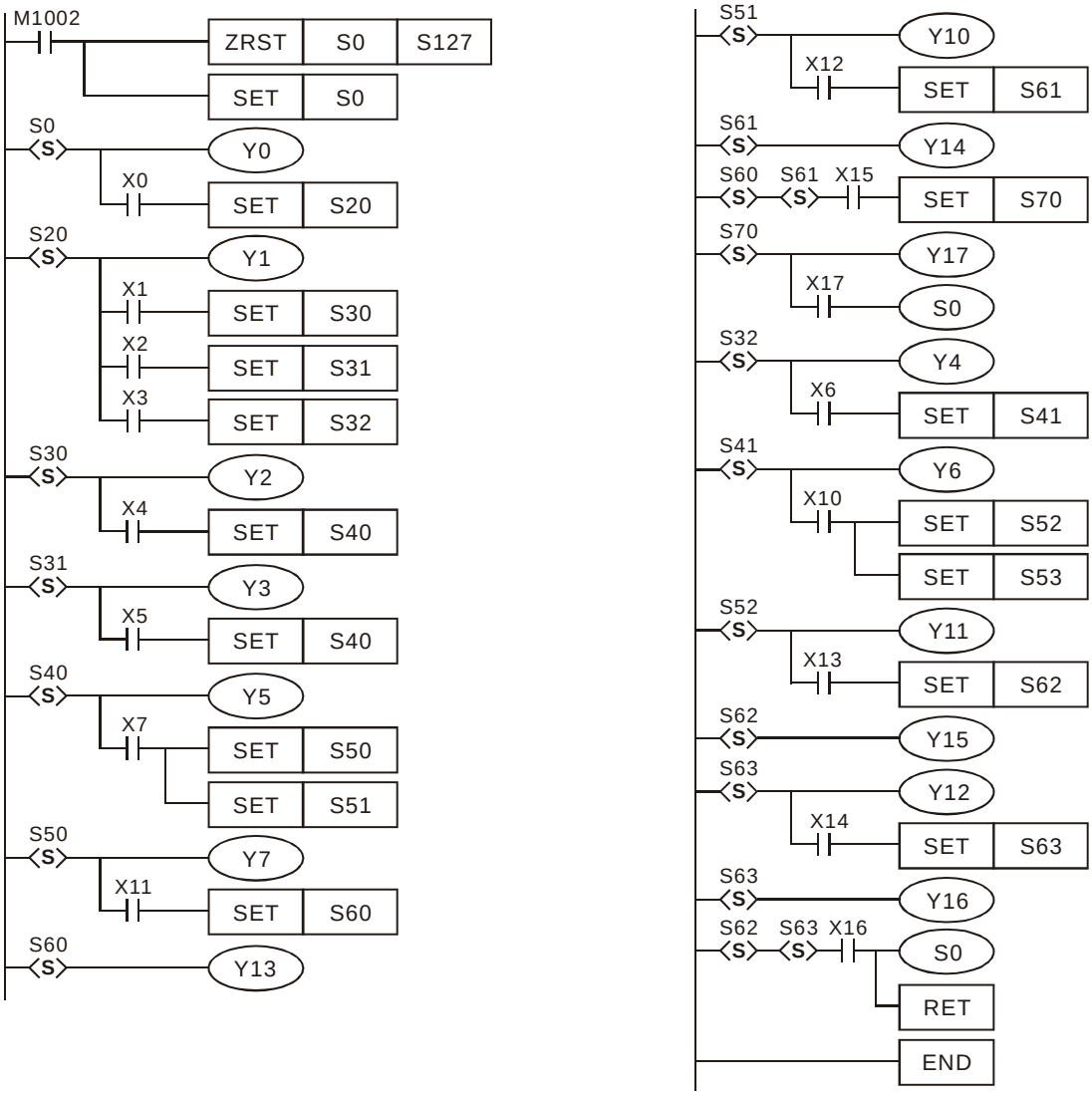
SFC:



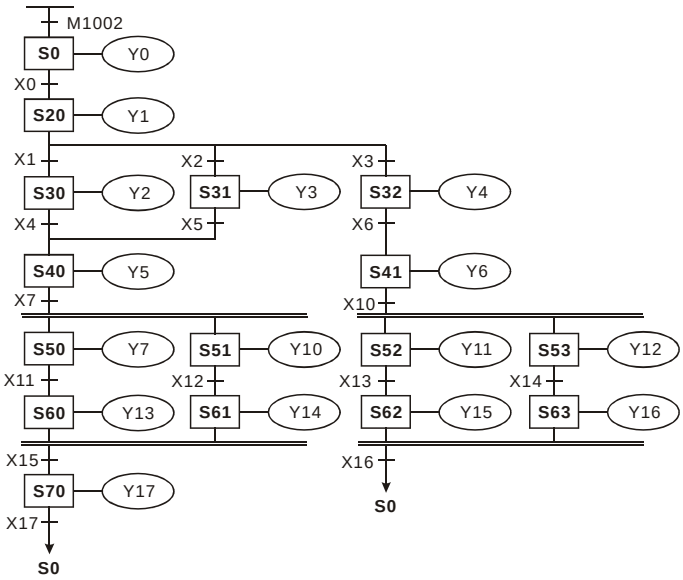
Combination Example 1:

(Including alternative divergence/convergence and simultaneous divergence/convergence)

Ladder diagram:



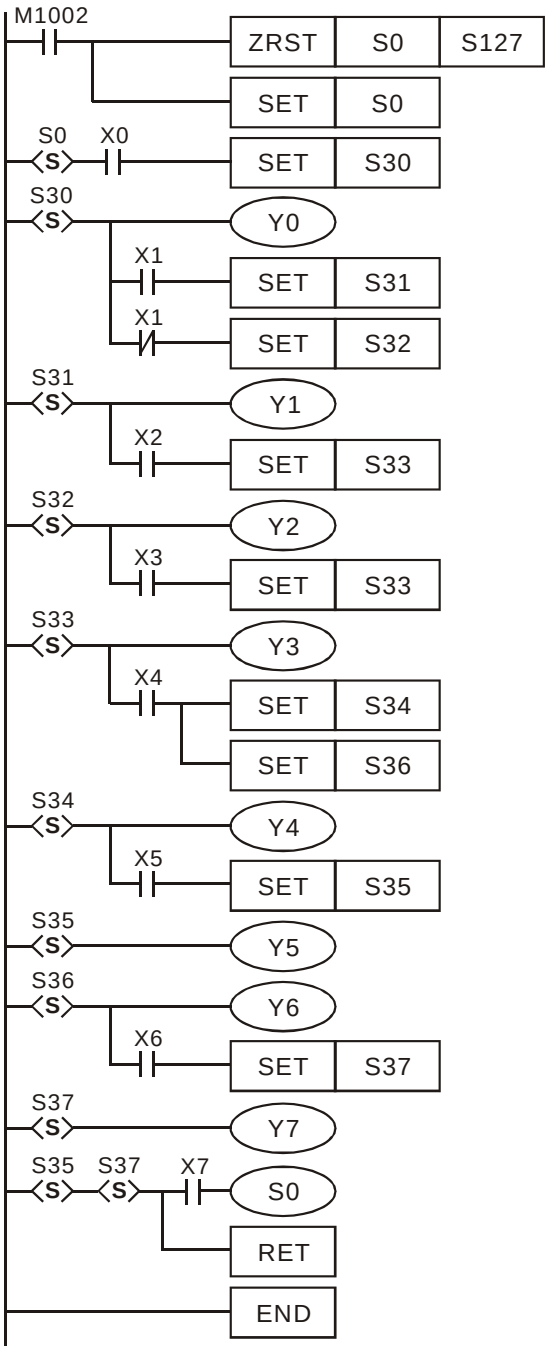
SFC:



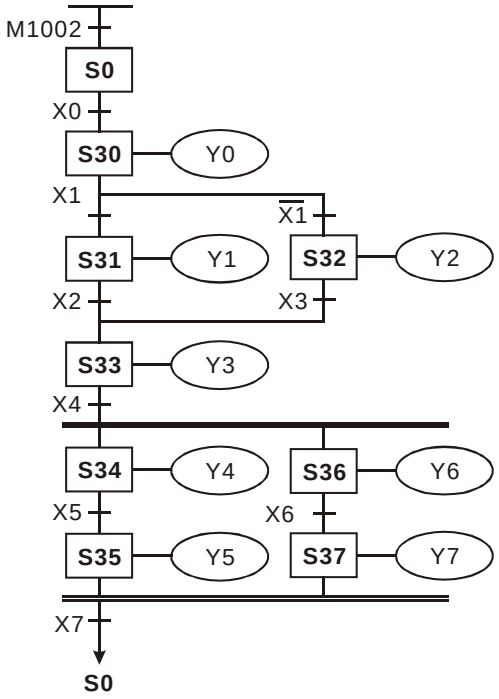
Combination Example 2:

(Including alternative divergence/convergence and simultaneous divergence/convergence)

Ladder diagram:

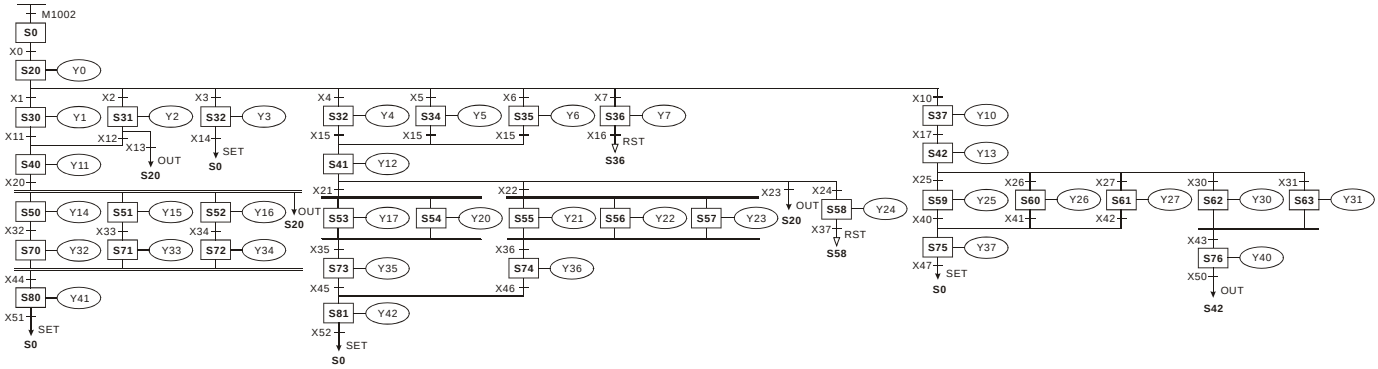


SFC:



Restrictions on Divergence Sequence:

1. You can use maximum 8 divergence steps in a divergence sequence. As the diagram below, there are maximum 8 diverged steps S30 ~ S37 after step S20.
2. You can use maximum 16 loops in multiple divergence sequences or in simultaneous sequences combined into one sequence. As the diagram below, there are 4 steps diverged after S40, 7 steps diverged after S41, and 5 steps diverged after S42. There are maximum 16 loops in this sequence.
3. You can designate a step in the sequence to jump to any step in another sequence.



4.6 IST Instruction

API	Mnemonic	Operands	Function
60	IST	(S) (D1) (D2)	Initial State

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*													IST: 7 steps
D_1				*												
D_2				*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start device in the designated operation mode **D_1**: The smallest No. of designated step in auto mode

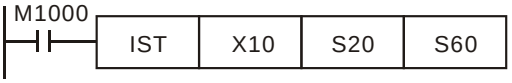
D_2: The biggest No. of designated step in auto mode

Explanations:

1. **S** will occupy 8 consecutive points.
2. Range of **D_1** and **D_2**: for SA/SX/SC/EH2/SV/EH3/SV2 S20 ~ S899; for ES/EX/SS S20 ~ S127; **D_2** > **D_1**.
3. See the specifications of each model for their range of use.
4. IST instruction can only be used once in the program.
5. Flags: M1040 ~ M1047. See remarks for more details.
6. IST instruction is a handy instruction specifically for the initial status of step ladder control procedure to accommodate special auxiliary relay.

Program Example 1:

1. Use of IST instruction

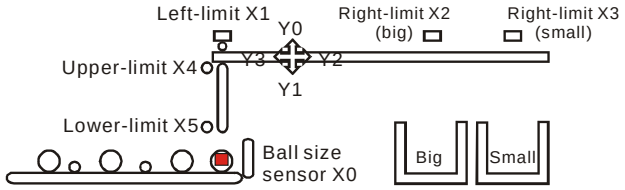


S X10: Individual operation
X11: Zero return
X12: Step operation
X13: One cycle operation
X14: Continuous operation
X15: Zero return enabled switch
X16: Start switch
X17: Stop switch

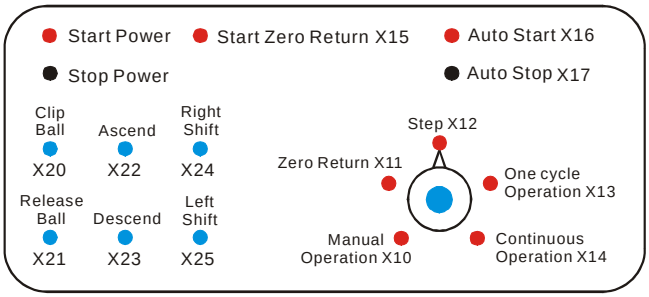
2. When IST instruction is being executed, the following special auxiliary relays will switch automatically.
   M1040: Operation forbidden
   M1041: Operation starts
   M1042: Pulse output enabled
   M1047: STL monitor enabled
   S0: Initiates manual operation
   S1: Initiates zero return
   S2: Initiates auto operation
3. S10 ~ S19 are for zero return and cannot be used as general steps. When S0 ~ S9 are in use, S0 ~ S2 represent manual operation mode, zero return mode and auto operation mode. Therefore, in the program, you have to write the circuit of the three steps in advance.
4. When switched to S1 (zero return) mode, any On in S10 ~ S19 will result in no zero return.
5. When switched to S2 (auto operation) mode, any On of the S in **D_1** ~ **D_2** or M1043 = On will result in no auto operation.

Program Example 2:

1. Robot arm control (by IST instruction):
   a) Motion request: Separate the big ball and small ball and move them to different boxes. Configure the control panel for the control.
   b) Motions of the robot arm: descending, clipping ball, ascending, right shifting, releasing ball, ascending, left shifting.
   c) I/O devices:

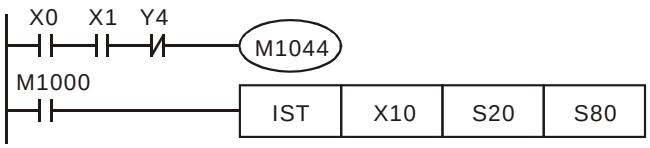


2. Operation modes:
   Manual operation: Turn On/Off of the load by a single button.
   Zero return: Press the zero return button to automatically zero-return the machine.
   Auto operation:
   a) Single step operation: Press “auto start” button for every one step forward.
   b) One cycle operation: Press “auto start” button at the zero point. After a cycle of auto operation, the operation will stops at the zero point. Press “auto stop” button in the middle of the operation to stop the operation and press “auto start” to restart the operation. The operation will resume until it meets the zero point.
   c) Continuous operation: Press “auto start” button at the zero point to resume the operation. Press “auto stop” to operate until it meets the zero point.
3. The control panel:

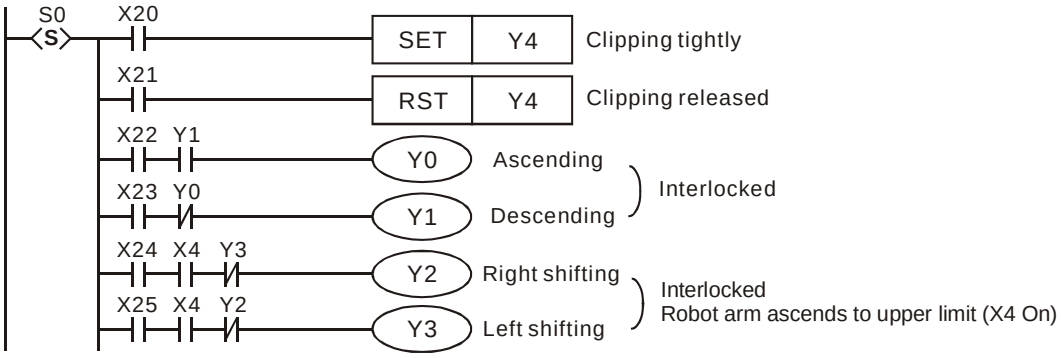


   a) Ball size sensor X0.
   b) Robot arm: left limit X1, big ball right limit X2, small ball right limit X3, upper limit X4, lower limit X5.
   c) Robot arm: ascending Y0, descending Y1, right shifting Y2, left shifting Y3, clipping Y4.

Start Circuit

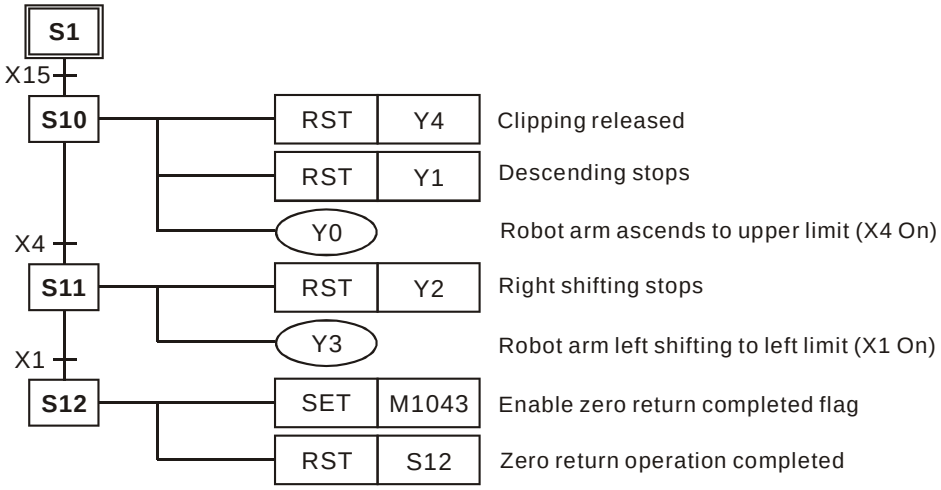


Manual Operation Mode

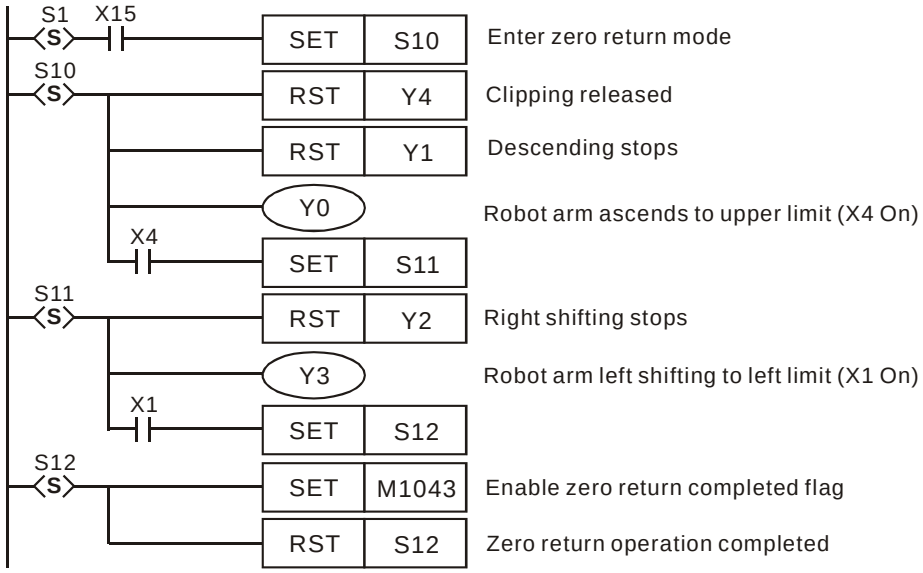


Zero Return Mode

SFC:

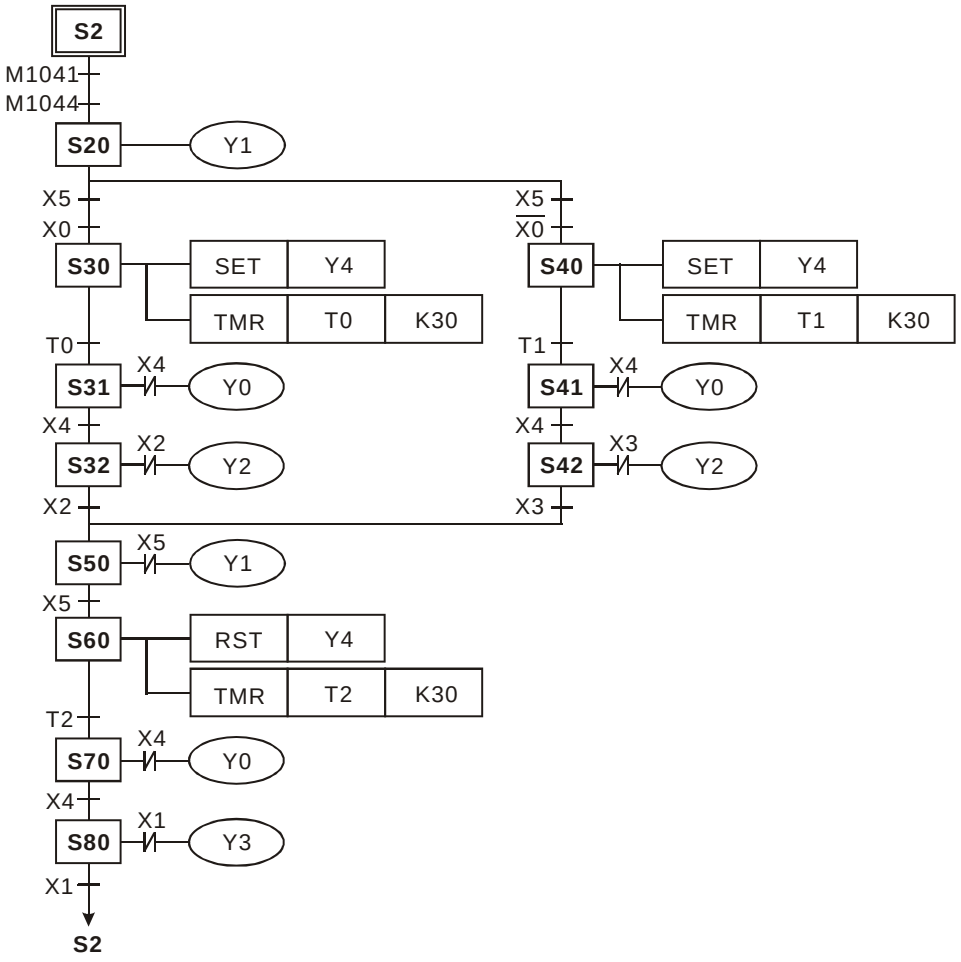


Ladder Diagram:

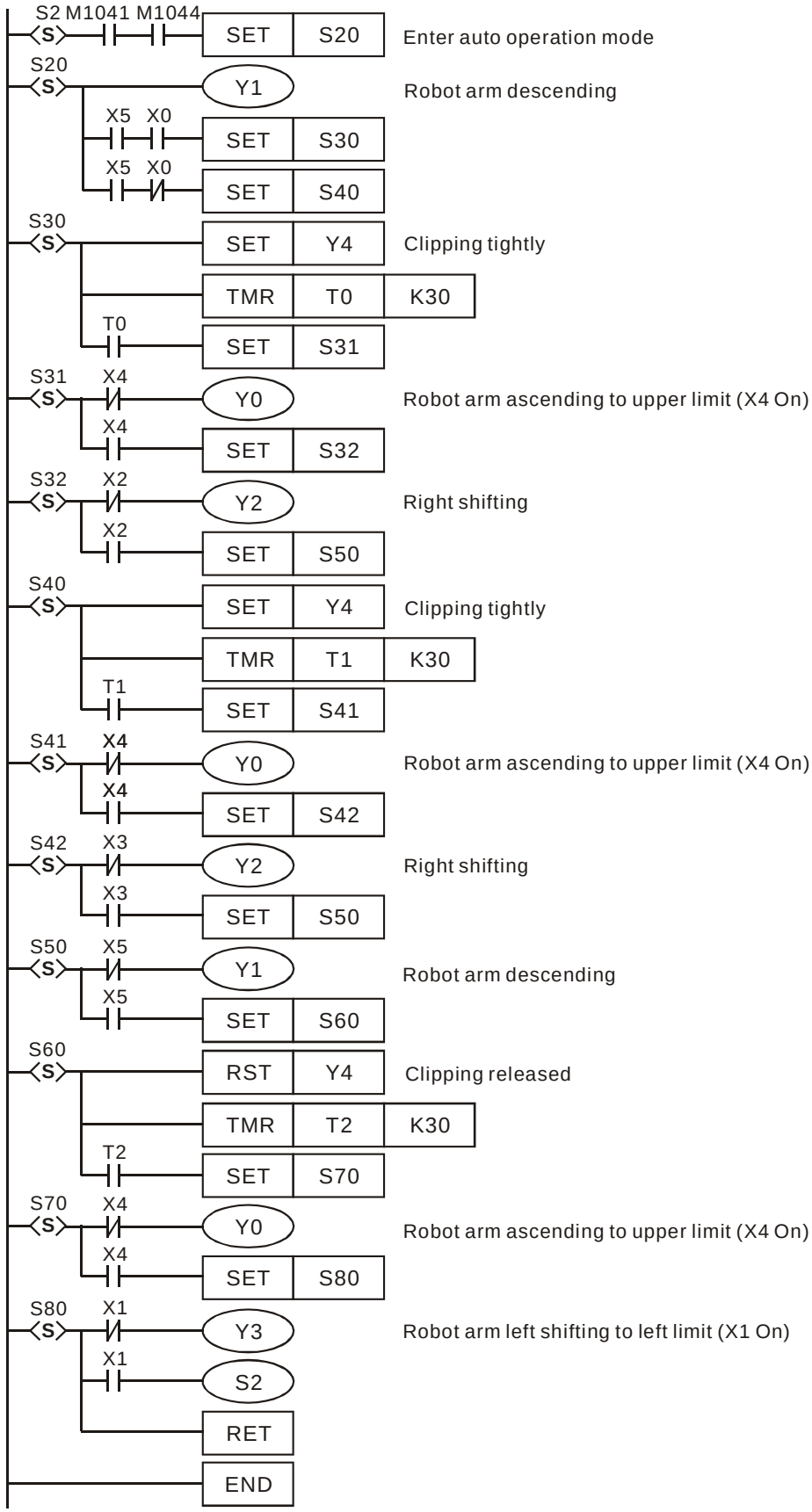


Auto Operation Modes

SFC:



Ladder Diagram:



5.1 Composition of Application Instruction

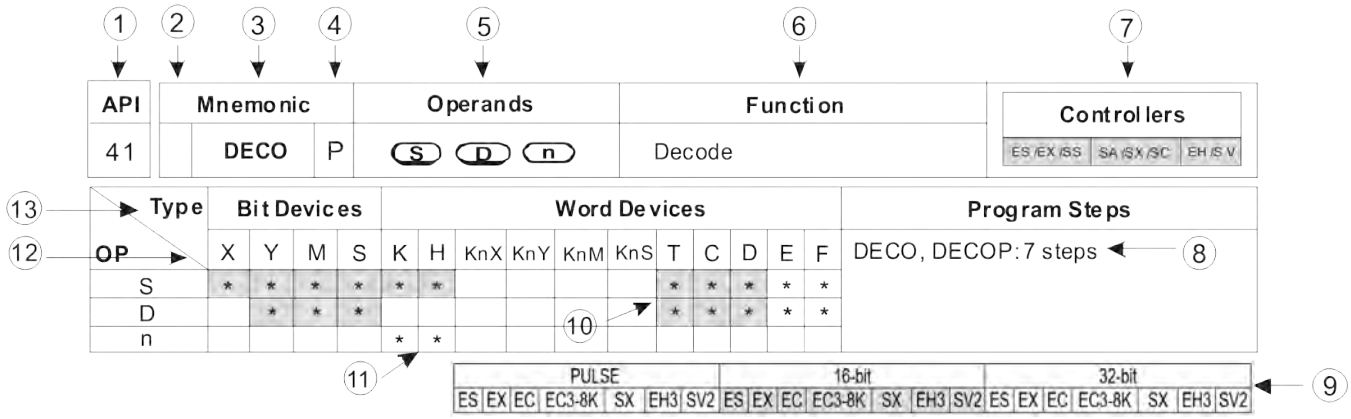
- An application instruction has two parts: the instruction and operands.

  Instruction: The function of the instruction

  Operands: Devices for processing the operations of the instruction

  The instruction part of an application instruction usually occupies 1 step, and one operand occupies 2 or 4 steps depending on the instruction is a 16-bit or 32-bit one.

- Format of an application instruction:



① API No.

② Indication of if there is a 16-bit or 32-bit instruction. If there is a 32-bit instruction, the column will be marked with “D”.

③ Mnemonic of the application instruction

④ Indication of if there is a pulse execution type instruction. If there is a pulse instruction, the column will be marked with “P”.

⑤ Operands

⑥ Function of the application instruction

⑦ DVP-PLC applicable to the application instruction. ES includes ES/EX/EC/EC3-8K (FW V8.60 or later); SX (FW V3.00); EH3 includes EH3/SV2.

⑧ Steps occupied by the 16-bit/32-bit/pulse execution instruction

⑨ DVP-PLC applicable to the pulse/16-bit/32-bit instruction

⑩ Column marked with * and in grey refers to E, F index register modification is applicable.

⑪ Column marked with * is the device applicable for the operand

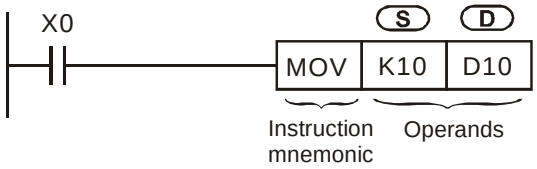
⑫ Device name

⑬ Device type

■ Input of application instruction:

Some application instructions are only composed of the instruction part (mnemonic), e.g. EI, DI, WDT…. Most application instructions are composed of the instruction part and many operands.

The application instructions for DVP-PLC are represented as API 00 ~ API 246. Every application instruction has its own mnemonic. For example, the mnemonic of API 12 is MOV. If you are using the ladder diagram editing software (WPLSoft) to input API 12 into the program, you only have to enter “MOV”. Different application instructions designate different operands. Take MOV instruction for example:



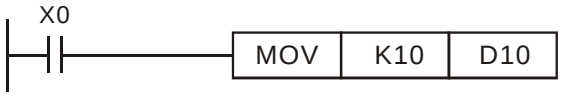
MOV instruction is to move the operand designated in **S** to the operand designated in **D**.

S	Source operand: If there are more than 1 source operands, they will be represented as **S_1**, **S_2**, ….
D	Destination operand: If there are more than 1 destination operands, they will be represented as **D_1**, **D_2**, ….
If the operand can only be constant K/H or a register, it will be represented as **m**, **m_1**, **m_2**, **n**, **n_1**, **n_2**, ….	

■ Length of operand (16-bit instruction or 32-bit instruction)

Depending on the contents in the operand, the length of an operand can be 16-bit or 32-bit. Therefore, a 16-bit instruction is for processing 16-bit operands, and 32-bit instruction is for processing 32-bit operands. The 32-bit instruction is indicated by adding a “D” before the 16-bit instruction.

16-bit MOV instruction



When X0 = On, K10 will be sent to D10.

32-bit DMOV instruction

When X1 = On, the content in (D11, D10) will be sent to (D21, D20).

■ Continuous execution instruction and pulse execution instruction

Continuous execution and pulse execution are the two types of execution for an application instruction. Due to that the execution time required will be shorter when the instruction is not executer, the pulse execution instructions are used more to shorten the scan period. Instructions marked with a “P" following the mnemonic are pulse execution instruction. Some instructions are mostly used as pulse execution type, e.g. INC, DEC, the kind of displacement instructions.

Pulse execution instruction

When X0 goes from Off to On, MOVP instruction will be executed once and the instruction will not be executed again in the scan period.

Continuous execution instruction

In every scan period when X1 = On, MOV instruction will be executed once.

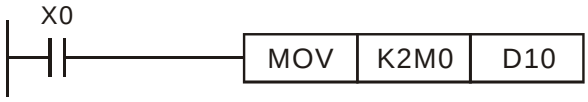
In the two figures, when X0, X1 = Off, the instruction will not be executed, and the content in operand **D** will remain unchanged.

■ Designation of operands

1. Bit devices X, Y, M, and S can be combined into word device, storing values and data for operaions in the form of KnX, KnY, KnM and KnS in an application instruction.
2. Data register D, timer T, counter C and index register E, F are designated by general operands.
3. A data register is usually in 16 bits, i.e. of the length of 1 register D. A designated 32-bit data register refers to 2 consecutive register Ds.
4. If an operand of a 32-bit instruction designates D0, the 32-bit data register composed of (D1, D0) will be occupied. D1 is the higher 16 bits; D0 is the lower 16 bits. The same rule also apply to timer T, 16-bit timers and C0 ~ C199.
5. When the 32-bit counters C200 ~ C255 are used as data registers, they can only be designataed by the operands of 32-bit instructions.

■ Format of operand

1. X, Y, M, and S can only On/Off a single point and are defined as bit devices.
2. 16-bit (or 32-bit) devices T, C, D, and registers E, F are defined as word devices.
3. You can place Kn (n = 1 refers to 4 bits. For 16-bit instruction, n = K1 ~ K4; for 32-bit instruction, n = K1 ~ K8) before bit devices X, Y, M and S to make it a word device for performing word-device operations. For example, K1M0 refers to 8 bits, M0 ~ M7.



When X0 = On, the contents in M0 ~ M7 will be moved to bit0 ~ 7 in D10 and bit8 ~ 15 will be set to “0”.

■ Data processing of word devices combined from bit devices

16-bit instruction	
Designated value: K-32,768 ~ K32,767	
Values for designated K1 ~ K4	
K1 (4 bits)	0 ~ 15
K2 (8 bits)	0 ~ 255
K3 (12 bits)	0 ~ 4,095
K4 (16 bits)	-32,768 ~ +32,767

32-bit instruction	
Designated value: K-2,147,483,648 ~ K2,147,483,647	
Values for designated K1 ~ K8	
K1 (4 bits)	0 ~ 15
K2 (8 bits)	0 ~ 255
K3 (12 bits)	0 ~ 4,095
K4 (16 bits)	0 ~ 65,535
K5 (20 bits)	0 ~ 1,048,575
K6 (24 bits)	0 ~ 167,772,165
K7 (28 bits)	0 ~ 268,435,455
K8 (32 bits)	-2,147,483,648 ~ +2,147,483,647

■ Flags

1. General flags

a) The flags listed below are for indicating the operational result of the application instruction.

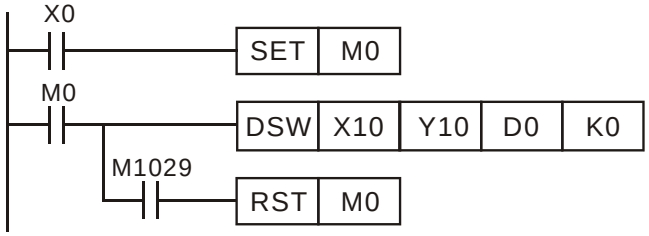
M1020: zero flag M1022: carry flag

M1021: borrow flag M1029: execution of instruction is completed

All flags will turn On or Off according to the operational result of an instruction. For example, the execution result of operation instructions ADD/SUB/MUL/DVI will affect the status of M1020 ~ M1022. When the instruction is not executed, the On/Off status of the flag will be held. The status of the four flags relates to many instructions. See relevant instructions for more details.

b) Example of M1029

When the contact of DSW (Digital Switch) instruction is On, 4 output points will automatically act in cycle at the frequency of 0.1 second in order to read the set value of the digital switch. If the contact goes Off during the execution, the action will be disabled. When it is On again, the disabled action will be re-executed. If you do not wish the action to be disabled, you can take the circuit below as a reference.



When X0 = On, DSW will be enabled.
When X0 = Off, M0 will be Off only when DSW completes a cycle and M1029 = On.

2. Error Operation Flags

Errors occur during the execution of the instruction when the combination of application instructions is incorrect or the devices designated by the operand exceed their range. Other than errors, the flags listed in the table below will be On, and error codes will also appear.

Device	Explanation
M1067 D1067 D1069	When operational errors occur, M1067 will be On. D1067 displays the error code. D1069 displays the step where the error occurs. Other errors occurring will update the contents in D1067 and D1069. M1067 will be Off when the error is eliminated.
M1068 D1068	When operational errors occur, M1068 will be On. D1068 displays the step where the error occurs. Other errors occurring wil not update the content in D1068. You have to use RST instruction to reset M1068 to Off; otherwise M1068 will keep being On.

3. Flags for expanding functions

Some application instructions can use some special flags to expand their functions or complete special functions. For example, the communication instruction RS can use M1161 to switch between 8-bit and 16-bit transmission mode.

■ Times of using instructions

Since it takes time for some instructions to be executed and some instructions shuld work with the corresponding parts of PLC, there are limitation on the times of using some instructions in the program. Refer to the table below for reference.

1. Can be edited or executed only once in the program:

API 58 (PWM) (ES series MPU)	API 60 (IST) (ES/SA/EH2/EH3/SV/SV2 series MPU)
API 74 (SEGL) (ES series MPU)	API 155 (DABSR) (SC/EH2/EH3 series MPU)
API 52 (MTR) (all series MPU)	API 70 (TKY) (all series MPU)
API 69 (SORT) (all series MPU)	API 71 (HKY) (all series MPU)
API 72 (DSW) (all series MPU)	API 74 (SEGL) (all series MPU)
API 151 (PWD) (all series MPU)	API 75 (ARWS) (all series MPU)
API 80 (RS) (all series MPU) (each COM port)	API 100 (MODRD) (all series MPU)
API 101 (MODWR) (all series MPU)	API 102 (FWD) (all series MPU)
API 103 (REV) (all series MPU)	API 104 (STOP) (all series MPU)
API 105 (RDST) (all series MPU)	API 106 (RSTEF) (all series MPU)
API 150 (MODRW) (all series MPU) (each COM port)	

2. Can be edited or executed twice in the program:

API 57 (PLSY) (ES/SX/EC3-8K series MPU)	API 59 (PLSR) (ES/SX/EC3-8K series MPU)
API 74 (SEGL) (EH3/SV2 series MPU)	API 77 (PR) (EC3-8K/SX/EH3/SV2 series MPU)
API 58 (PWM) (SX/EC3-8K series MPU)	API 72 (DSW) (all series MPU)
API 56 (SPD) (ES/SX/EC3-8K series MPU)	

3. Can be edited or executed 4 times in the program:

API 169 (HOUR) (all series MPU)	API 57 (PLSY) (EH2/EH3/SV/SV2 series MPU)
API 58 (PWM) (EH2/EH3/SV/SV2 series MPU)	API 56 (SPD) (EC3-8K/EH3/SV2 series MPU)

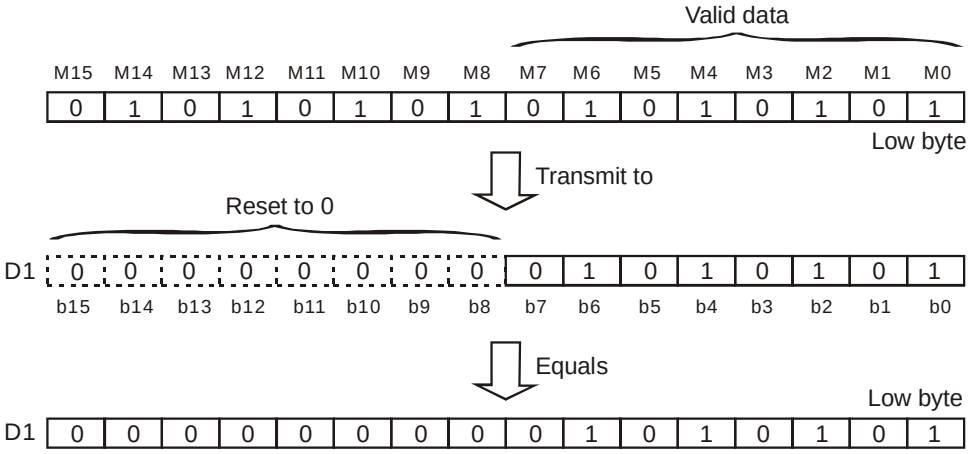
4. Can be edited or executed 8 times in the program:

API 64 (TTMR) (all series MPU)

5. API 53 DHSCS and API 54 DHSCR together can be used only maximum 4 times in the program (ES series MPU).
6. API 53 DHSCS, API 54 DHSCR, and API 55 DHSZ together can be used only maximum 6 times in the program (SX/EC3-8K series MPU).
7. There is on limitation on the times of using the high-speed output instructions PLSY, PWM and PLSR and positioning instructios, but only one high-speed output instruction will be enabled in every scan.
8. In EH2/EH3/SV/SV2 series MPU, there is no limitation on the times of using hardware high-speed counter instructions DHSCS, DHSCR and DHSZ, but when the three instructions are enabled at the same time, DHSCS will occupy 1 memory unit, DHSCR 1 memory unit, and DHSZ 2 memory units. The total memeory units occupied by the three instructions cannot be more than 8 units. If there are more than 8 memory units occupied, the PLC system will execute the instruction that is first scanned and enabled and ignore the rest.

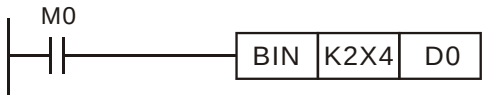
5.2 Handling of Numeric Values

- Devices only with On/Off status are called bit devices, e.g. X, Y, M and S. Devices used exclusively for storing numeric values are called word devices, e.g. T, C, D, E and F. Bit device plus a specific bit device (place a digit before the bit device in Kn) can be used in the operand of an application instruction in the form of numeric value.
- n = K1 ~ K4 for a 16-bit value; n = K1 ~ K8 for a 32-bit value. For example, K2M0 refers to an 8-bit value composed of M0 ~ M7.



- K1M0, K2M0, and K3M0 are transmitted to 16-bit registers and the vacant high bits will be filled in “0”. The same rule applied to when K1M0, K2M0, K3M0, K4M0, K5M0, K6M0, and K7M0 are transmitted to 32-bit registers and the vacant high bits will be filled in “0”.

- In the 16-bit (or 32-bit) operation, if the contents of the operand are designated as bit devices K1 ~ K3 (or K4 ~ K7), the vacant high bits will be regarded as “0”. Therefore, the operation is a positive-value one.



The BCD value composed of X4 ~ X13 will be converted to BIN value and sent to D0.

- You can choose any No. for bit devices, but please make the 1s digit of X and Y “0”, e.g. X0, X10, X20, …Y0, Y10…, and the 1s digit of M and S “8’s multiple” (“0” is still the best choice), e.g. M0, M10, M20….
- Designating continuous device No.

  Take data register D for example, continuous D refers to D0, D1, D2, D3, D4….

  For bit devices with specifically designated digit, continuous No. refers to:

K1X0	K1X4	K1X10	K1X14…
K2Y0	K2Y10	K2Y20	Y2X30…
K3M0	K3M12	K3M24	K3M36…
K4S0	K4S16	K4S32	K4S48…

Please follow the No. in the table and do not skip No. in case confusion may occur. In addition, if you use K4Y0 in the 32-bit operation, the higher 16 bits will be regarded as “0”. For 32-bit data, please use K8Y0.

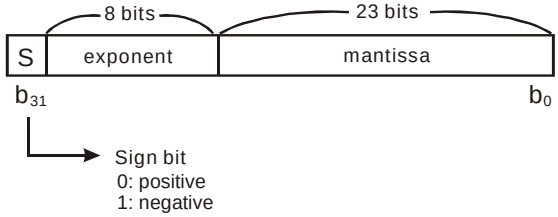
The operations in DVP-PLC are conducted in BIN integers. When the integer performs division, e.g. 40 ÷ 3 = 13 and the remainder is 1. When the integer performs square root operations, the decimal point will be left out. Use decimal point operation instructions to obtain the decimal point.

Application instructions revelant to decimal point:

API 49 (FLT)	API 110 (D ECMP)	API 111 (D EZCP)	API 112 (D MOVR)
API 116 (D RAD)	API 117 (D DEG)	API 118 (D EBCD)	API 119 (D EBIN)
API 120 (D EADD)	API 121 (D ESUB)	API 122 (D EMUL)	API 123 (D EDIV)
API 124 (D EXP)	API 125 (D LN)	API 126 (D LOG)	API 127 (D ESQR)
API 128 (D POW)	API 129 (INT)	API 130 (D SIN)	API 131 (D COS)
API 132 (D TAN)	API 133 (D ASIN)	API 134 (D ACOS)	API 135 (D ATAN)
API 136 (D SINH)	API 137 (D COSH)	API 138 (D TANH)	API 172 (D ADDR)
API 173 (D SUBR)	API 174 (D MULR)	API 175 (D DIVR)	
API 275~280 (FLD※)	API 281~286 (FAND※)	API 287~292 (FOR※)	

Binary Floating Point

DVP-PLC represents floating points in 32 bits, following the IEEE754 standard:



$(-1)^S \times 2^{E-B} \times 1.M$, in which B = 127

Therefore, the range for the 32-bit floating point is $\pm 2^{-126} \sim \pm 2^{+128}$, i.e. $\pm 1.1755 \times 10^{-38} \sim \pm 3.4028 \times 10^{+38}$

Example 1: Representing "23" in 32-bit floating point

Step 1: Convert “23” into a binary value: 23.0 = 10111

Step 2: Normalize the binary value: $10111 = 1.0111 \times 2^4$, in which 0111 is mantissa and 4 is exponent

Step 3: Obtain the exponent: ∵ E – B = 4 → E – 127 = 4 ∴ E = 131 = 1000011_2

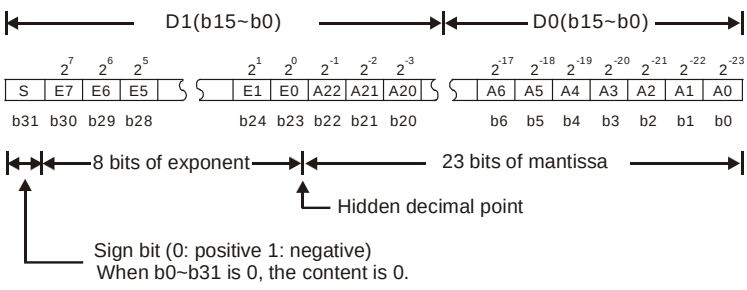
Step 4: Combine the sign bit, exponent and mantissa into a floating point

0 100**0001**1 011**1000**0000**0000**0000**0000**$_2$ = $41B80000_{16}$

Example 2: Representing “-23.0” in 32-bit floating point

The steps required are the same as those in Example 1. The only difference is you have to alter the sign bit into “1”.

DVP-PLC uses registers of 2 continuous No. to combine into a 32-bit floating point. For example, we use registers (D1, D0) for storing a binary floating point as below:



Decimal Floating Point

- Since the binary floating point are not very user-friendly, we can convert it into a decimal floating point for use. Please be noted that the decimal point operation in DVP-PLC is still in binary floating point.
- The decimal floating point is represented by 2 continuous registers. The register of smaller No. is for the constant while the register of bigger No. is for the exponent.

  Example: Storing a decimal floating point in registers (D1, D0)

  Decimal floating point = [constant D0] × $10^{[\text{exponent D1}]}$

  Constant D0 = ±1,000 ~ ±9,999

  Exponent D1 = -41 ~ +35

  The constant 100 does not exist in D0 due to 100 is represented as 1,000 × 10^{-1}. The range of decimal floating point is ±1175 × 10^{-41} ~ ±3402×10^{+35}.
- The decimal floating point can be used in the following instructions:

  D EBCD: Converting binary floating point to decimal floating point

  D EBIN: Converting decimal floating point to binary floating point
- Zero flag (M1020), carry flag (M1021), carry flag (M1022) and the floating point operation instructions:

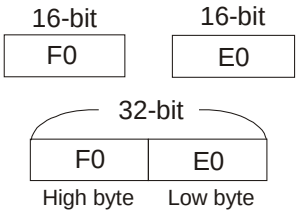
  Zero flag: M1020 = On if the operational result is “0”.

  Borrow flag: M1021 = On if the operational result exceeds the minimum unit.

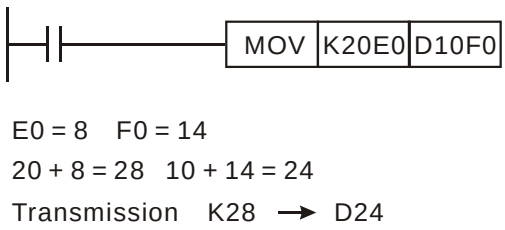
  Carry flag: M1022 = On if the absolute value of the operational result exceeds the range of use.

5.3 E, F Index Register Modification

The index registers are 16-it registers. There are 2 points of E, F in ES/EC, 8 points E0 ~ E3 and F0 ~ F3 in SX, and 16 points E0 ~ E7 and F0 ~ F7 in EH3/SV2/EC3-8K series MPU.



- E and F index registers are 16-bit data registers. They can be read and written.
- If you need a 32-bit register, you have to designate E. In this case, F will be covered up by E and cannot be used; otherwise, the contents in E may become incorrect. (We recommend you use MOVP instruction to reset the contents in D to 0 when the PLC is switched on.)
- Combination of E and F when you designate a 32-bit index register: (E0, F0), (E1, F1), (E2, F2), ... (E7, F7)



See the diagram in the left hand side. E, F index register modification refers to the content in the operand changes with the contents in E and F.

For example, E0 = 8 and K20E0 represents constant K28 (20 + 8). When the condition is true, constant K28 will be transmitted to register D24.

Devices modifiable in ES/EC/EC3-8K/SX series MPU: P, X, Y, M, S, KnX, KnY, KnM, KnS, T, C, D.

Devices modifiable in EH2/SV/EH3/SV2 series MPU: P, I, X, Y, M, S, K, H, KnX, KnY, KnM, KnS, T, C, D

E and F can modify the devices listed above but cannot modify themselves and Kn. K4M0E0 is valid and K0E0M0 is invalid. Grey columns in the table of operand at the beginning page of each application instruction indicate the operands modifiable by E and F.

If you need to modify device P, I, X, Y, M, S, KnX, KnY, KnM, KnS, T, C and D by E, F, you have to select a 16-bit register, i.e. you can designate E or F. To modify constant K and H in a 32-bit instruction, you have to select a 32-bit register, i.e. you have to designate E.

When you use the instruction mode in WPLSoft to modify constant K and H, you have to use @, e.g.

"MOV K10@E0 D0F0"

DVP-PLC applicable to the application instruction. ES includes ES/EX/EC/EC3-8K (FW V8.60 or later) (EC3: FW V8.40 or previous version); SX (FW V3.00); EH3 includes EH3/SV2.

ES/EX/EC series MPU does not support pulse execution type instructions (P instruction).

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	32-bit			ES	EC3-8K	EH2	EH3	16-bit	32-bit
Loop Control	00	CJ	-	✓	Conditional Jump	✓	✓	✓	✓	3	–
	01	CALL	-	✓	Call Subroutine	✓	✓	✓	✓	3	–
	02	SRET	-	-	Subroutine Return	✓	✓	✓	✓	1	–
	03	IRET	-	-	Interrupt Return	✓	✓	✓	✓	1	–
	04	EI	-	-	Enable Interrupts	✓	✓	✓	✓	1	–
	05	DI	-	-	Disable Interrupts	✓	✓	✓	✓	1	–
	06	FEND	-	-	The End of The Main Program (First End)	✓	✓	✓	✓	1	–
	07	WDT	-	✓	Watchdog Timer Refresh	✓	✓	✓	✓	1	–
	08	FOR	-	-	Start of a FOR-NEXT loop	✓	✓	✓	✓	3	–
	09	NEXT	-	-	End of a FOR-NEXT loop	✓	✓	✓	✓	1	–
Transmission Comparison	10	CMP	DCMP	✓	Compare	✓	✓	✓	✓	7	13
	11	ZCP	DZCP	✓	Zone Compare	✓	✓	✓	✓	9	17
	12	MOV	DMOV	✓	Move	✓	✓	✓	✓	5	9
	13	SMOV	-	✓	Shift Move	–	✓	✓	✓	11	–
	14	CML	DCML	✓	Compliment	✓	✓	✓	✓	5	9
	15	BMOV	-	✓	Block Move	✓	✓	✓	✓	7	–
	16	FMOV	DFMOV	✓	Fill Move	✓	✓	✓	✓	7	13
	17	XCH	DXCH	✓	Exchange	✓	✓	✓	✓	5	9
	18	BCD	DBCD	✓	Binary Coded Decimal	✓	✓	✓	✓	5	9
	19	BIN	DBIN	✓	Binary	✓	✓	✓	✓	5	9
Four Arithmetic Operation	20	ADD	DADD	✓	Addition	✓	✓	✓	✓	7	13
	21	SUB	DSUB	✓	Subtraction	✓	✓	✓	✓	7	13
	22	MUL	DMUL	✓	Multiplication	✓	✓	✓	✓	7	13
	23	DIV	DDIV	✓	Division	✓	✓	✓	✓	7	13
	24	INC	DINC	✓	Increment	✓	✓	✓	✓	3	5
	25	DEC	DDEC	✓	Decrement	✓	✓	✓	✓	3	5
	26	WAND	DAND	✓	Logical Word AND	✓	✓	✓	✓	7	13
	27	WOR	DOR	✓	Logical Word OR	✓	✓	✓	✓	7	13
	28	WXOR	DXOR	✓	Logical Exclusive OR	✓	✓	✓	✓	7	13
	29	NEG	DNEG	✓	2's Complement (Negative)	✓	✓	✓	✓	3	5

Category	API	Mnemonic		P	Function	Applicable to				STEPS	
		16-bit	32-bit	instruction		ES	EC3-8K	EH2	EH3	16-bit	32-bit
Rotation & Displacement	30	ROR	DROR	✓	Rotation Right	✓	✓	✓	✓	5	9
	31	ROL	DROL	✓	Rotation Left	✓	✓	✓	✓	5	9
	32	RCR	DRCR	✓	Rotation Right with Carry	✓	✓	✓	✓	5	9
	33	RCL	DRCL	✓	Rotation Left with Carry	✓	✓	✓	✓	5	9
	34	SFTR	-	✓	Bit Shift Right	✓	✓	✓	✓	9	–
	35	SFTL	-	✓	Bit Shift Left	✓	✓	✓	✓	9	–
	36	WSFR	-	✓	Word Shift Right	–	✓	✓	✓	9	–
	37	WSFL	-	✓	Word Shift Left	–	✓	✓	✓	9	–
	38	SFWR	-	✓	Shift Register Write	–	✓	✓	✓	7	–
	39	SFRD	-	✓	Shift Register Read	–	✓	✓	✓	7	–
Data Processing	40	ZRST	-	✓	Zero Reset	✓	✓	✓	✓	5	–
	41	DECO	-	✓	Decode	✓	✓	✓	✓	7	–
	42	ENCO	-	✓	Encode	✓	✓	✓	✓	7	–
	43	SUM	DSUM	✓	Sum of Active Bits	✓	✓	✓	✓	5	9
	44	BON	DBON	✓	Check Specified Bit Status	✓	✓	✓	✓	7	13
	45	MEAN	DMEAN	✓	Mean	✓	✓	✓	✓	7	13
	46	ANS	-	-	Timed Annunciator Set	–	–	✓	✓	7	–
	47	ANR	-	✓	Annunciator Reset	–	–	✓	✓	1	–
	48	SQR	DSQR	✓	Square Root	✓	✓	✓	✓	5	9
	49	FLT	DFLT	✓	Floating Point	✓	✓	✓	✓	5	9

API	Mnemonic		Operands	Function
00	CJ	P	(S)	Conditional Jump

OP	Range	Program Steps
(S)	P0~P255	CJ, CJP: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: The destination pointer of conditional jump

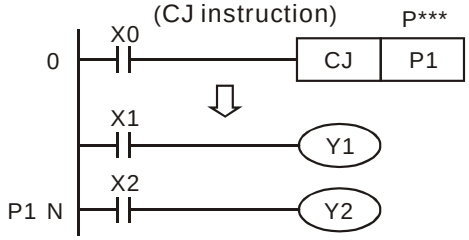
Explanations:

1. Operand **S** can designate P.
2. P can be modified by index register E, F.
3. In ES series models: Operand S can designate P0 ~ P63.
4. In EC3-8K/SX/EH3 series models: Operand S can designate P0 ~ P255.
5. When the user does not wish a particular part of PLC program in order to shorten the scan time and execute dual outputs, CJ instruction or CJP instruction can be adopted.
6. When the program designated by pointer P is prior to CJ instruction, WDT timeout will occur and PLC will stop running. Please use it carefully.
7. CJ instruction can designate the same pointer P repeatedly. However, CJ and CALL cannot designate the same pointer P; otherwise an error will occur.
8. Actions of all devices while conditional jumping is being executed.

a) Y, M and S remain their previous status before the conditional jump takes place.
b) Timer 10ms and 100ms that is executing stops.
c) Timer T192 ~ T199 that execute the subroutine program will continue and the output contact executes normally.
d) The high-speed counter that is executing the counting continues counting and the output contact executes normally.
e) The ordinary counters stop executing.
f) If the “reset instruction” of the timer is executed before the conditional jump, the device will still be in the reset status while conditional jumping is being executed.
g) Ordinary application instructions are not executed.
h) The application instructions that are being executed, i.e. API 53 DHSCS, API 54 DHSCR, API 55 DHSZ, API 56 SPD, API 57 PLSY, API 58 PWM, API 59 PLSR, API 157 PLSV, API 158 DRVI, API 159 DRVA, continue being executed.

Program Example 1:

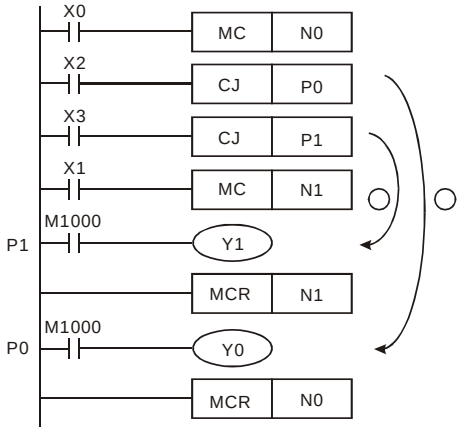
1. When X0 = On, the program automatically jumps from address 0 to N (the designated label P1) and keeps its execution. The addresses between 0 and N will not be executed.
2. When X0 = Off, as an ordinary program, the program keeps on executing from address 0. CJ instruction will not be executed at this time.



Program Example 2:

1. CJ instruction can be used in the following 5 conditions between MC and MCR instructions.
   a) Without MC ~ MCR.
   b) From without MC to within MC. Valid in the loop P1 as shown in the figure below.
   c) In the same level N, inside of MC~MCR.
   d) From within MC to without MCR.
   e) Jumping from this MC ~ MCR to another MC ~ MCR*

   *Note: This function is only available in ES/EC series models V4.9 (and above) and EC3-8K/SX/EH3/SV2 series models.
2. Actions in ES/EC series models V4.7 (and below): When CJ instruction is used between MC and MCR, it can only be applied without MC ~ MCR or in the same N layer of MC ~ MCR. Jumping from this MC ~ MCR to another MC ~ MCR will result in errors, i.e. a) and c) as stated above can ensure correct actions; others will cause errors.
3. When MC instruction is executed, PLC will push the status of the switch contact into the self-defined stack in PLC. The stack will be controlled by the PLC, and the user cannot change it. When MCR instruction is executed, PLC will obtain the previous status of the switch contact from the top layer of the stack. Under the conditions as stated in b), d) and e), the times of pushing-in and obtaining stack may be different. In this case, the maximum stack available to be pushed in is 8 and the obtaining of stacks cannot resume once the stack becomes empty. Thus, when using CALL or CJ instructions, the user has to be aware of the pushing-in and obtaining of stacks.



Program Example 3:

1. The states of each device

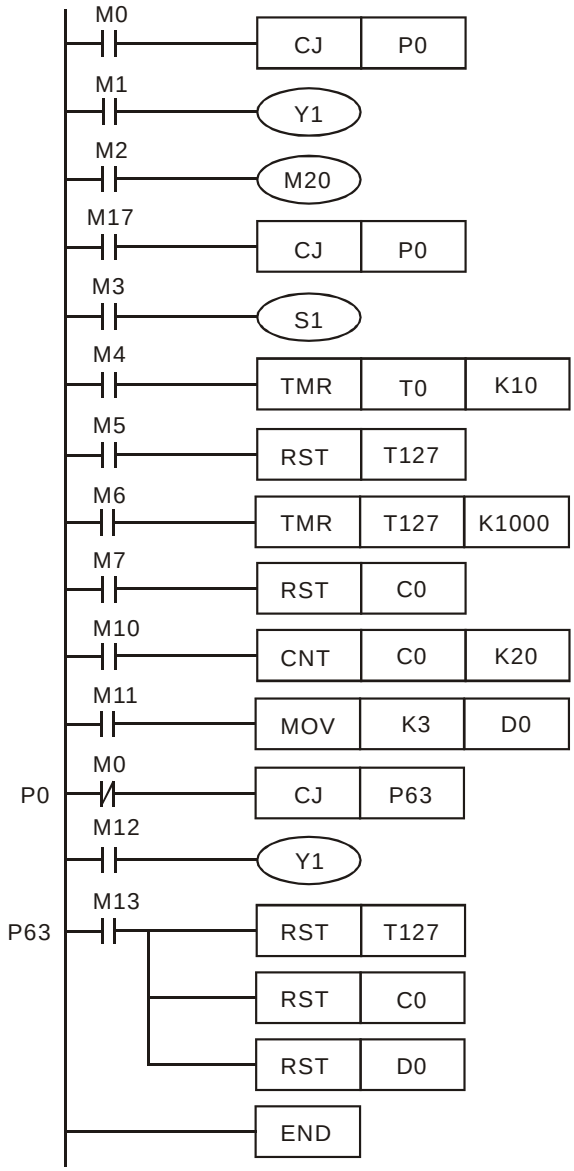
Device	Contact state before CJ is executed	Contact state when CJ is being executed	Output coil state when CJ is being executed
Y, M, S	M1, M2, M3 Off	M1, M2, M3 Off→On	Y1 *1, M20, S1 Off
	M1, M2, M3 On	M1, M2, M3 On→Off	Y1 *1, M20, S1 On
10ms, 100ms Timer ES/SA/EH	M4 Off	M4 Off→On	Timer T0 is not enabled.
	M4 On	M4 On→Off	Timer T0 immediately stops and is latched. M0 On→Off, T0 is reset as 0.
1ms, 10ms, 100ms Timer *2 (accumulative) SA/EH	M6 Off	M6 Off→On	Timer T240 is not enabled.
	M6 On	M6 On→Off	Once the timer function is enabled and when met with CJ instruction, all accumulative timers will stop timing and stay latched. M0 On→Off. T240 remains unchanged.
C0 ~ C234 *3	M7, M10 Off	M10 On/Off trigger	Counter does not count.
	M7 Off, M10 On/Off trigger	M10 On/Off trigger	Counter C0 stops counting and stays latched. After M0 goes Off, C0 resumes its counting.
Application instruction	M11 Off	M11 Off→On	Application instructions are not executed.
	M11 On	M11 On→Off	The skipped application instructions are not executed, but API 53 ~ 59, API 157 ~ 159 keep being executed.

*1: Y1 is a dual output. When M0 is Off, M1 will control Y1. When M0 is On, M12 will control Y1.

*2: When the timers (T184 ~ T199, applicable in EC3-8K; T192 ~ T199 applicable in SX/EH3/SV2 series MPU) used by a subroutine re driven and encounter the execution of CJ instruction, the timing will resume. After the timing target is reached, the output contact of the timer will be On.

*3: When the high-speed counters (C235 ~ C255) are driven and encounter the execution of CJ instruction, the counting will resume, as well as the action of the output points.

2. Y1 is a dual output. When M0 = Off, Y1 is controlled by M1. When M0 = On, Y1 is controlled by M12.



API	Mnemonic		Operands	Function
01	CALL	P	(S)	Call Subroutine

OP	Range	Program Steps
(S)	P0 ~ P255	CALL, CALLP: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: The pointer of call subroutine.

Explanations:

1. Operand S can designate P.
2. P can be modified by index register E, F.
3. In ES series models: Operand S can designate P0 ~ P63.
4. In EC3-8K/SX/EH3 series models: Operand S can designate P0 ~ P255.
5. Edit the subroutine designated by the pointer after FEND instruction.
6. The number of pointer P, when used by CALL, cannot be the same as the number designated by CJ instruction.
7. If only CALL instruction is in use, it can call subroutines of the same pointer number with no limit on times.
8. Subroutine can be nested for 5 levels including the initial CALL instruction. (If entering the sixth level, the subroutine won't be executed.)

API	Mnemonic	Function
02	SRET	Subroutine Return

OP	Descriptions	Program Steps
N/A	Automatically returns to the step immediately following the CALL instruction which activated the subroutine	SRET: 1 steps

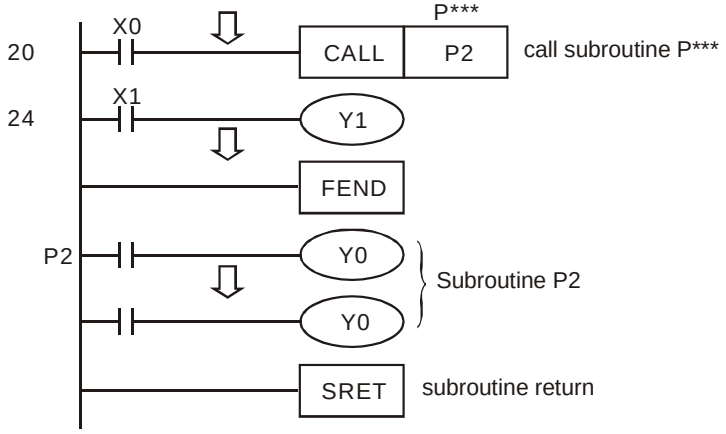
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Explanations:

1. No operand. No contact to drive the instruction is required.
2. The subroutine will return to main program by SRET after the termination of subroutine and execute the sequence program located at the next step to the CALL instruction.

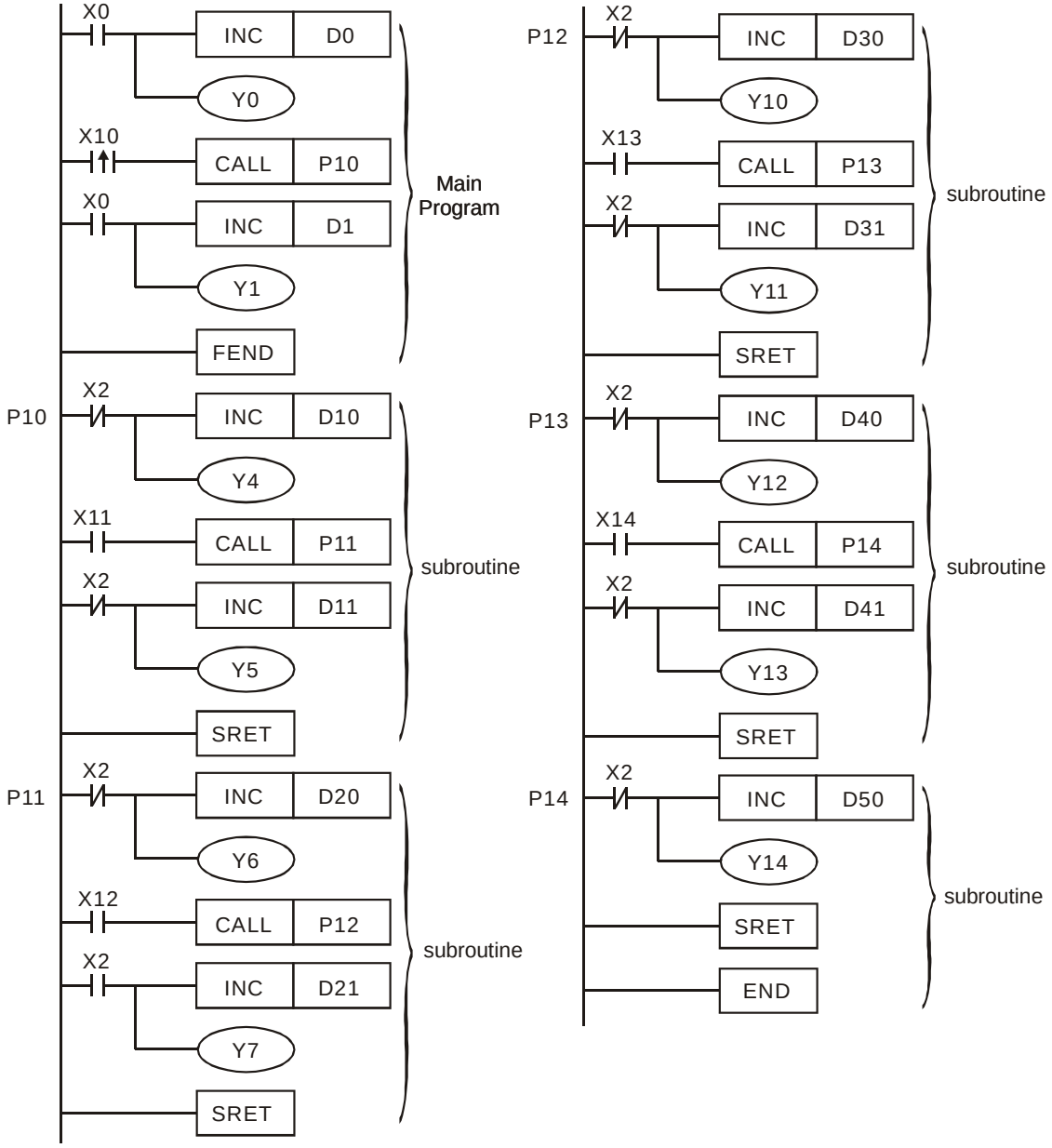
Program Example 1:

When X0 = On, CALL instruction is executed and the program jumps to the subroutine designated by P2. When SRET instruction is executed, the program returns to address 24 and continues its execution.



Program Example 2:

1. When X10 goes from Off to On, its rising-edge trigger executes CALL P10 instruction and the program jumps to the subroutine designated by P10.
2. When X11 is On, CALL P11 is executed and the program jumps to the subroutine designated by P11.
3. When X12 is On, CALL P12 is executed and the program jumps to the subroutine designated by P12.
4. When X13 is On, CALL P13 is executed and the program jumps to the subroutine designated by P13.
5. When X14 is On, CALL P14 is executed and the program jumps to the subroutine designated by P14. When SRET is executed, the program returns to the previous P※ subroutine and continues its execution.
6. After SRET instruction is executed in P10 subroutine, returning to the main program.


X0
INC D0
Y0
X10
CALL P10
X0
INC D1
Y1
FEND
Main Program
P10
X2
INC D10
Y4
X11
CALL P11
X2
INC D11
Y5
SRET
subroutine
P11
X2
INC D20
Y6
X12
CALL P12
X2
INC D21
Y7
SRET
subroutine
P12
X2
INC D30
Y10
X13
CALL P13
X2
INC D31
Y11
SRET
subroutine
P13
X2
INC D40
Y12
X14
CALL P14
X2
INC D41
Y13
SRET
subroutine
P14
X2
INC D50
Y14
SRET
END
subroutine

API	Mnemonic	Function
03	IRET	Interrupt Return

OP	Descriptions	Program Steps
N/A	IRET ends the processing of an interruption subroutine and returns to the execution of the main program.	IRET: 1 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Explanations:

1. No operand. No contact to drive the instruction is required.
2. Interruption return refers to interrupt the subroutine.
3. After the interruption is over, returning to the main program from IRET to execute the next instruction where the program was interrupted.

API	Mnemonic	Function
04	EI	Enable Interrupts

OP	Descriptions	Program Steps
N/A	See more details of the explanation on this instruction in DI (Disable Interruption) instruction.	EI: 1 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Explanations:

1. No operand. No contact to drive the instruction is required.
2. The pulse width of the interruption signal should be >200us.
3. See DI instruction for the range of the No. of I for all models.
4. See DI instruction for more details about M1050 ~ M1059, M1280 ~ M1299.

API	Mnemonic	Function
05	DI	Disable Interrupts

OP	Descriptions	Program Steps
N/A	When the special auxiliary relay M1050 ~ M1059, M1280 ~ M1299 for disabling interruption is driven, the corresponding interruption request will not be executed even in the range allowed for interruptions.	DI: 1 step

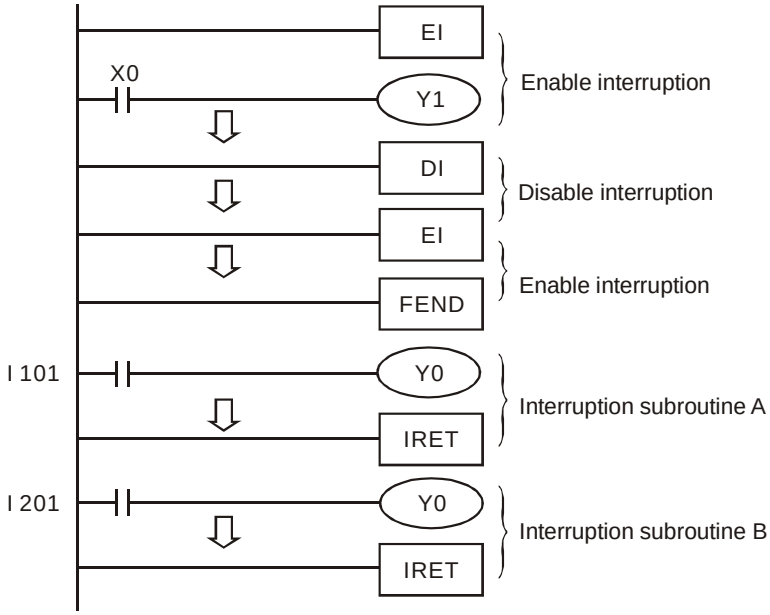
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Explanations:

1. No operand. No contact to drive the instruction is required.
2. EI instruction allows interrupting subroutine in the program, e.g. external interruption, timed interruption, and high-speed counter interruption.
3. In the program, using interruption subroutine between EI and DI instruction is allowed. However, you can choose not to use DI instruction if there is no interruption-disabling section in the program.
4. When M1050 ~ M1059 are the special auxiliary relays to drive disabling interruption in ES/SX/EC3-8K, or M1280 ~ M1299 are the special auxiliary relays to drive disabling interruption in EH3/SV2, the corresponding interruptions will not be executed even in the area allowed for interruptions.
5. Pointer for interruption (I) must be placed after FEND instruction.
6. Other interruptions are not allowed during the execution of interruption subroutine.
7. When many interruptions occur, the priority is given to the firstly executed interruption. If several interruptions occur simultaneously, the priority is given to the interruption with the smaller pointer No.
8. The interruption request occurring between DI and EI instructions that cannot be executed immediately will be memorized and will be executed in the area allowed for interruption.
9. When using the interruption pointer, DO NOT repeatedly use the high-speed counter driven by the same X input contact.
10. When immediate I/O is required during the interruption, write REF instruction in the program to update the status of I/O.

Program Example:

During the operation of PLC, when the program scans to the area between EI and DI instructions and X1 = Off→On or X2 = Off→On, interruption subroutine A or B will be executed. When the subroutine executes to IRET, the program will return to the main program and resumes its execution.



Remarks:

1. No. of interruption pointer I in ES/EX/EC:
   a) External interruptions: (I001, X0), (I101, X1), (I201, X2), (I301, X3) 4 points[1].
   b) Time interruptions: I6□□, 1 point (□□ = 10 ~ 99, time base = 1ms) (support V5.7 and above)
   c) Communication interruption for receiving specific words (I150) (support V5.7 and above)
2. No. of interruption pointer I in SX/EC3-8K:
   a) External interruptions: SX series: (I001, X0), (I101, X1), (I201, X2), (I301, X3), (I401, X4), (I501, X5) 6 points; EC3-8K series: (I001,X0), (I101, X1), (I201, X2), (I301, X3), (I401, X4), (I501, X5), (I601, X6), (I701, X7) 8 points.
   b) Time interruptions: SX series: I6□□, I7□□ 2 points. (□□ = 1 ~ 99ms, time base = 1ms); EC3-8K series: I6□□, I7□□ 2 points。(□□＝2~99, time base =1ms)；I8□□, 1 point(□□=10~99, time base =0.1ms)
   c) SX series: High-speed counter interruptions: I010, I020, I030, I040, 1050, 1060, 6 points. (used with API 53 DHSCS instruction to generate interruption signals)
   d) Communication interruption for receiving specific words .(I150)
   e) The order for execution of interruption pointer I: high-speed counter interruption, external interruption, time interruption and communication interruption for receiving specific words.
   f) Among the following 6 interruption No., (I001, I010), (I101, I020), (I201, I030), (I301, I040), (I401, I050), (I501, I060), the program allows the user to use only one of the two numbers in a pair. If the user uses the two numbers in the pair, grammar check errors may occur when the program is written into PLC.
3. No. of interruption pointer I in EH3/SV2:

[1] Input points occupied by external interruptions cannot be used for inputs of high-speed counters; otherwise grammar check errors may occur when the program is written in PLC.

a) External interruptions: (I00□, X0), (I10□, X1), (I20□, X2), (I30□, X3), (I40□, X4), (I50□, X5) 6 points. (□ = 0 designates interruption in falling-edge, □ = 1 designates interruption in rising-edge)

b) Time interruptions: I6□□, I7□□, 2 points. (□□ = 1~99ms, time base = 1ms)
I8□□ 1 point. (□□ = 1 ~ 99ms, time base = 0.1ms)

c) High-speed counter interruptions: I010, I020, I030, I040, 1050, 1060 6 points. (used with API 53 DHSCS instruction to generate interruption signals)

d) When pulse output interruptions I110, I120 (triggered when pulse output is finished), I130, I140 (triggered when the first pulse output starts) are executed, the currently executed program is interrupted and jumps to the designated interruption subroutine.

e) Communication interruption: I150, I160, I170

f) Frequency measurement card interruption: I180

g) The order for execution of interruption pointer I: external interruption, time interruption, high-speed counter interruption, pulse interruption, communication interruption and frequency measurement card interruption.

4. No. of interruption pointer I in EH3/SV2:

a) External interruptions: (I00□, X0), (I10□, X1), (I20□, X2), (I30□, X3), (I40□, X4), (I50□, X5), (I60□, X6), (I70□, X7), (I90□, X10), (I91□, X11), (I92□, X12), (I93□, X13), (I94□, X14), (I95□, X15), (I96□, X16), (I97□, X17) 16 points. (□ = 0 designates interruption in falling-edge, □ = 1 designates interruption in rising-edge)

b) Time interruptions: I6□□, I7□□, 2 points. (□□ = 2~99ms, time base = 1ms)
I8□□ 1 point. (□□ = 1 ~ 99ms, time base = 0.1ms)

c) High-speed counter interruptions: I010, I020, I030, I040, 1050, 1060 6 points. (used with API 53 DHSCS instruction to generate interruption signals)

d) When pulse output interruptions I110, I120 (triggered when pulse output is finished), I130, I140 (triggered when the first pulse output starts) are executed, the currently executed program is interrupted and jumps to the designated interruption subroutine.

e) Communication interruption: I150, I151, I153、I160, I161, I163, I170

f) The order for execution of interruption pointer I: external interruption, time interruption, high-speed counter interruption, pulse interruption, and communication interruption.

5. “Disable interruption” flags in ES/EX/EC:

Flag	Function
M1050	Disable external interruption I001
M1051	Disable external interruption I101
M1052	Disable external interruption I201
M1053	Disable external interruption I301
M1056	Disable time interruption I6□□

6. “Disable interruption” flags in SX/EC3-8K:

Flag	Function
M1050	Disable external interruption I001
M1051	Disable external interruption I101

Flag	Function
M1052	Disable external interruption I201
M1053	Disable external interruption I301
M1054	Disable external interruption I401
M1055	Disable external interruption I501
M1056	Disable time interruption I6□□ (I601 for EC3-8K included)
M1057	Disable time interruption I7□□ (I701 for EC3-8K included)
M1059	Disable high-speed counter interruption I010 ~ I060

7. “Disable interruption” flags in EH3/SV2:

Flag	Function
M1280	Disable external interruption I00□
M1281	Disable external interruption I10□
M1282	Disable external interruption I20□
M1283	Disable external interruption I30□
M1284	Disable external interruption I40□
M1285	Disable external interruption I50□
M1286	Disable time interruption I6□□
M1287	Disable time interruption I7□□
M1288	Disable time interruption I8□□
M1289	Disable high-speed counter interruption I010
M1290	Disable high-speed counter interruption I020
M1291	Disable high-speed counter interruption I030
M1292	Disable high-speed counter interruption I040
M1293	Disable high-speed counter interruption I050
M1294	Disable high-speed counter interruption I060
M1295	Disable pulse output interruption I110
M1296	Disable pulse output interruption I120
M1297	Disable pulse output interruption I130
M1298	Disable pulse output interruption I140
M1299	Disable communication interruption I150
M1300	Disable communication interruption I160
M1301	Disable communication interruption I170
M1302	Disable frequency measurement card interruption I180
M1340	Generate interruption I110 after CH0 pulse is sent
M1341	Generate interruption I120 after CH1 pulse is sent
M1342	Generate interruption I130 when CH0 pulse is being sent
M1343	Generate interruption I140 when CH1 pulse is being sent

API	Mnemonic	Function
06	FEND	The End of The Main Program (First End)

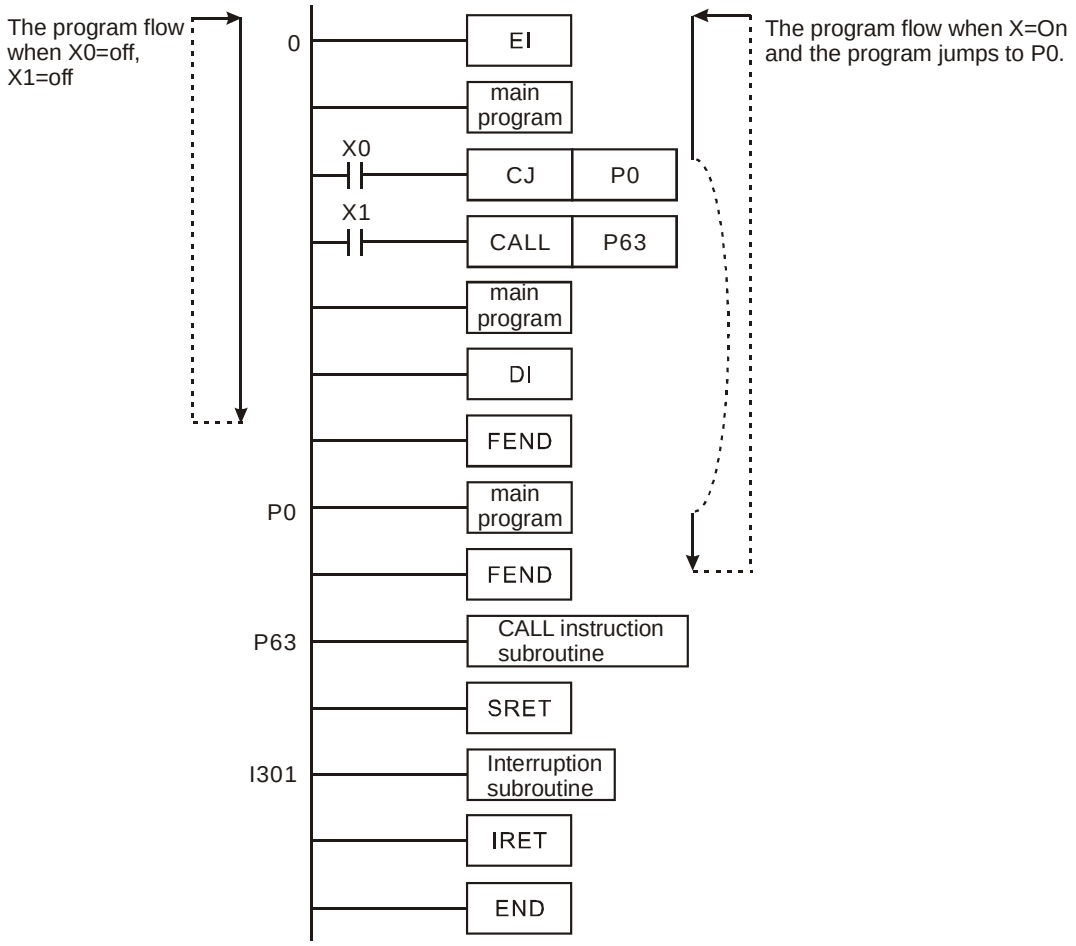
OP	Descriptions	Program Steps
N/A	No contact to drive the instruction is required.	FEND: 1 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

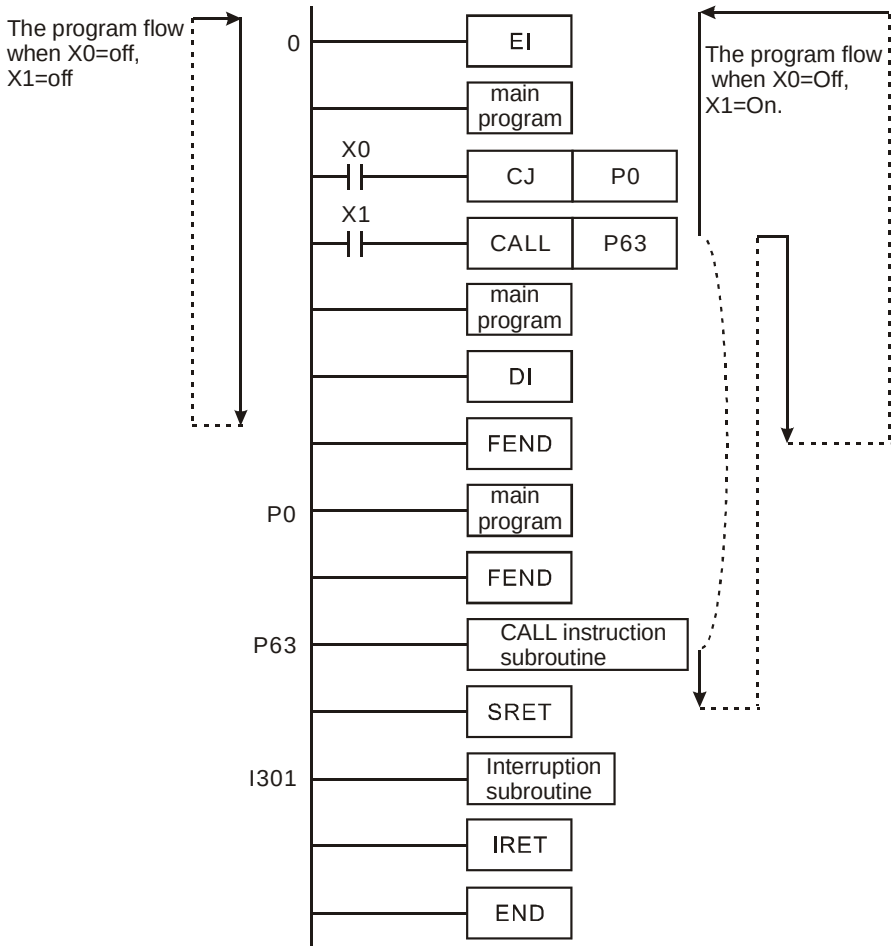
Explanations:

1. This instruction denotes the end of the main program. It has the same function as that of END instruction when being executed by PLC.
2. CALL must be written after FEND instruction and add SRET instruction in the end of its subroutine. Interruption program has to be written after FEND instruction and IRET must be added in the end of the service program.
3. If several FEND instructions are in use, place the subroutine and interruption service programs between the final FEND and END instruction.
4. After CALL instruction is executed, executing FEND before SRET will result in errors in the program.
5. After FOR instruction is executed, executing FEND before NEXT will result in errors in the program.

CJ Instruction Program Flow:



CALL Instruction Program Flow:



API	Mnemonic		Function
07	WDT	P	Watchdog Timer Refresh

OP	Descriptions	Program Steps
N/A		WDT, WDTP: 1 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

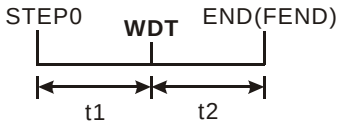
Explanations:

1. No operand.
2. The watchdog timer in DVP series PLCs is used for monitoring the operation of the PLC system.
3. WDT instruction can be used to reset Watch Dog Timer. If the PLC scan time (from step 0 to END or when FEND instruction is executed) exceeds 200ms, PLC ERROR LED will flash. The user will have to turn off PLC and back On again. PLC will determine RUN/STOP status by RUN/STOP switch. If there is no RUN/STOP switch, PLC will return to STOP status automatically.
4. When to use WDT:

a) When errors occur in the PLC system.

b) When the executing time of the program is too long, resulting in the scan time being larger than the content in D1000, the user can improve the problem by the following two methods.

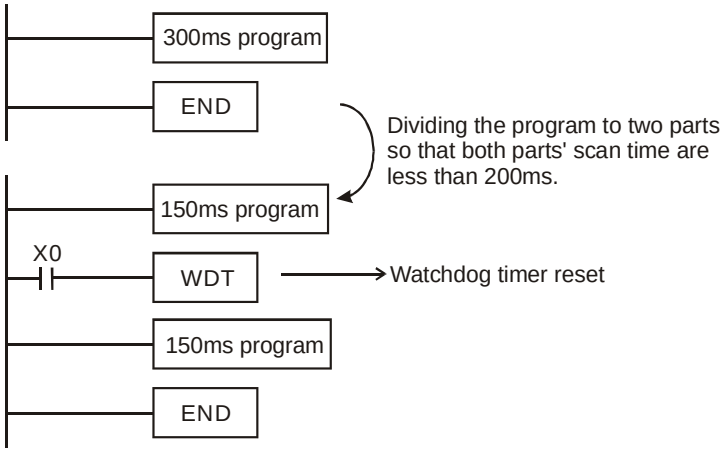
- Using WDT instruction



- Using the set value in D1000 (default value: 200ms) to change the time for watchdog.

Program Example:

Assume the scan time of the program is 300ms, divide the program into two parts and place WDT instruction in the middle of the two parts, making scan time of the first half and second half of the program being less than 200ms.



API	Mnemonic	Operands	Function
08	FOR	(S)	Start of a FOR-NEXT Loop

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	FOR: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: The number of repeated nested loops

Explanations:

1. No contact to drive the instruction is required.
2. See the specifications of each model for their range of use.

API	Mnemonic	Function
09	NEXT	End of a FOR-NEXT Loop

OP	Descriptions	Program Steps
N/A		NEXT: 1 steps

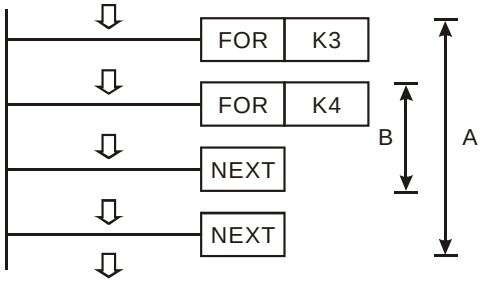
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Explanations:

1. No operand. No contact to drive the instruction is required.
2. FOR instruction indicates FOR ~ NEXT loops executing back and forth N times before escaping for the next execution.
3. N = K1 ~ K32,767. N is regarded as K1 when N ≤ 1.
4. When FOR~NEXT loops are not executed, the user can use the CJ instruction to escape the loops.
5. Error will occur when
   a) NEXT instruction is before FOR instruction.
   b) FOR instruction exists but NEXT instruction does not exist.
   c) There is NEXT instruction after FEND or END instruction.
   d) The number of instructions between FOR ~ NEXT differs.
6. FOR~NEXT loops can be nested for maximum five levels. Be careful that if there are too many loops, the increased PLC scan time may cause timeout of watchdog timer and error. Users can use WDT instruction to modify this problem.

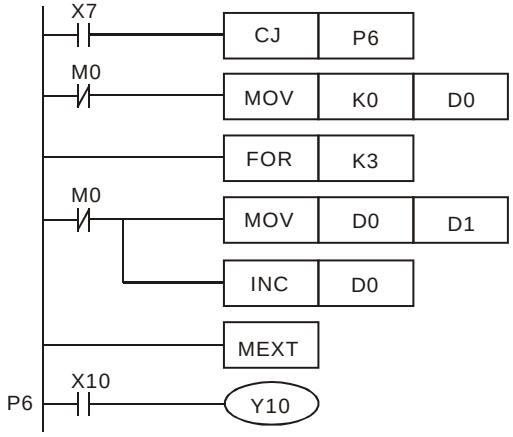
Program Example 1:

After program A has been executed for 3 times, it will resume its execution after NEXT instruction. Program B will be executed for 4 times whenever program A is executed once. Therefore, program B will be executed 3 × 4 = 12 times in total.



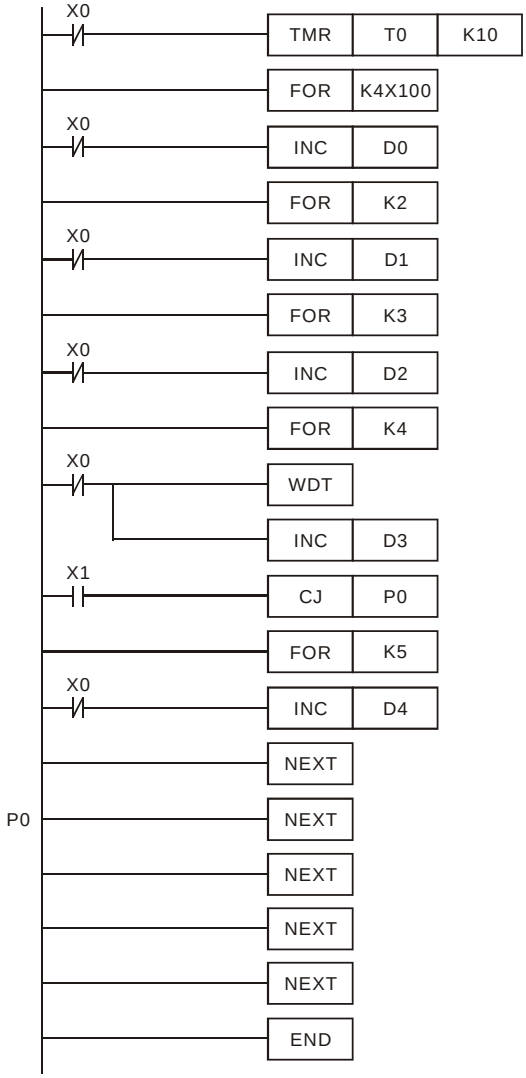
Program Example 2:

When X7 = Off, PLC will execute the program between FOR ~ NEXT. When X7 = On, CJ instruction jumps to P6 and avoids executing the programs between FOR ~ NEXT.



Program Example 3:

When the programs between FOR ~ NEXT are not to be executed, the user can adopt CJ instruction for a jumping. When the most inner FOR ~ NEXT loop is in the status of X1 = On, CJ instruction executes jumping to P0 and skips the execution on P0.



API	Mnemonic			Operands	Function
10	D	CMP	P	(S1) (S2) (D)	Compare

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	CMP, CMPP: 7 steps DCMP, DCMPP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

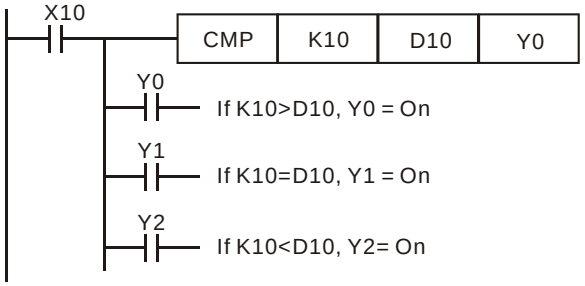
S_1: Comparison Value 1 **S_2**: Comparison Value 2 **D**: Comparison result

Explanations:

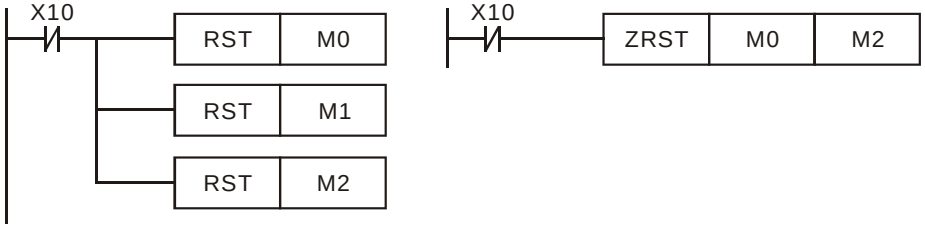
1. If **S_1** and **S_2** are used in device F, only 16-bit instruction is applicable.
2. Operand **D** occupies 3 consecutive devices.
3. See the specifications of each model for their range of use.
4. The contents in **S_1** and **S_2** are compared and the result will be stored in **D**.
5. The two comparison values are compared algebraically and the two values are signed binary values. When b15 = 1 in 16-bit instruction or b31 = 1 in 32-bit instruction, the comparison will regard the value as negative binary values.

Program Example:

1. Designate device Y0, and operand D automatically occupies Y0, Y1, and Y2.
2. When X10 = On, CMP instruction will be executed and one of Y0, Y1, and Y2 will be On. When X10 = Off, CMP instruction will not be executed and Y0, Y1, and Y2 remain their status before X10 = Off.
3. If the user need to obtain a comparison result with ≥ ≤, and ≠, make a series parallel connection between Y0 ~ Y2.



4. To clear the comparison result, use RST or ZRST instruction.



API	Mnemonic			Operands	Function
11	D	ZCP	P	(S1) (S2) (S) (D)	Zone Compare

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ZCP, ZCPP: 9 steps
S_1					*	*	*	*	*	*	*	*	*	*	*	DZCP, DZCPP: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
S					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Lower bound of zone comparison **S_2**: Upper bound of zone comparison **S**: Comparison value

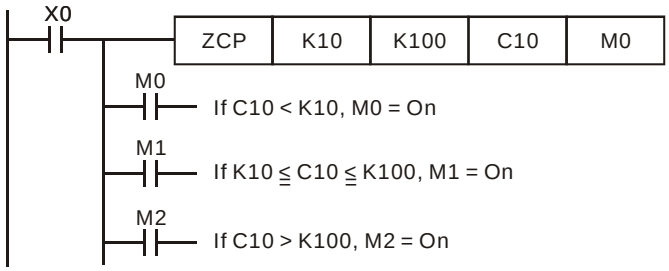
D: Comparison result

Explanations:

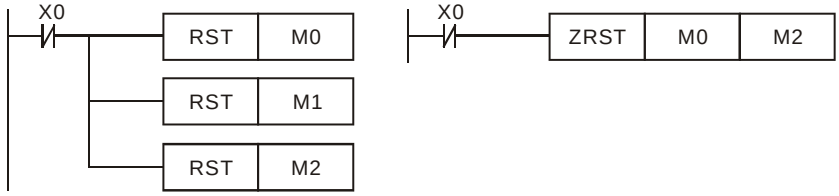
1. If **S_1**, **S_2** and **S** are used in device F, only 16-bit instruction is applicable.
2. The content in **S_1** should be smaller than the content in **S_2**.
3. Operand **D** occupies 3 consecutive devices.
4. See the specifications of each model for their range of use.
5. **S** is compared with its **S_1**, **S_2** and the result is stored in **D**.
6. When **S_1** > **S_2**, the instruction performs comparison by using **S_1** as the lower/upper bound.
7. The two comparison values are compared algebraically and the two values are signed binary values. When b15 = 1 in 16-bit instruction or b31 = 1 in 32-bit instruction, the comparison will regard the value as negative binary values.

Program Example:

1. Designate device M0, and operand D automatically occupies M0, M1 and M2.
2. When X0 = On, ZCP instruction will be executed and one of M0, M1, and M2 will be On. When X0 = Off, ZCP instruction will not be executed and M0, M1, and M2 remain their status before X0 = Off.



3. To clear the comparison result, use RST or ZRST instruction.



API	Mnemonic			Operands	Function
12	D	MOV	P	(S) (D)	Move

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	MOV, MOVP: 5 steps DMOV, DMOVP: 9 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

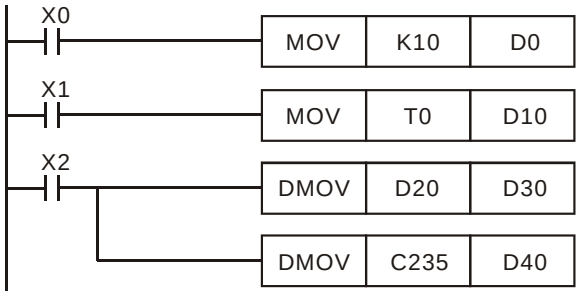
S: Source of data **D**: Destination of data

Explanations:

1. If **S** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. When this instruction is executed, the content of **S** will be moved directly to **D**. When this instruction is not executed, the content of **D** remains unchanged.
4. If the operation result refers to a 32-bit output, (i.e. application instruction MUL and so on), and the user needs to move the present value in the 32-bit high-speed counter, DMOV instruction has to be adopted.

Program Example:

1. MOV instruction has to be adopted in the moving of 16-bit data.
   a) When X0 = Off, the content in D10 will remain unchanged. If X0 = On, the value K10 will be moved to D10 data register.
   b) When X1 = Off, the content in D10 will remain unchanged. If X1 = On, the present value T0 will be moved to D10 data register.
2. DMOV instruction has to be adopted in the moving of 32-bit data.
   When X2 = Off, the content in (D31, D30) and (D41, D40) will remain unchanged. If X2 = On, the present value of (D21, D20) will be sent to (D31, D30) data register. Meanwhile, the present value of C235 will be moved to (D41, D40) data register.



API	Mnemonic		Operands	Function
13	SMOV	P	S m1 m2 D n	Shift Move

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	SMOV, SMOVP: 11 steps
m_1					*	*										
m_2					*	*										
D								*	*	*	*	*	*	*	*	
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

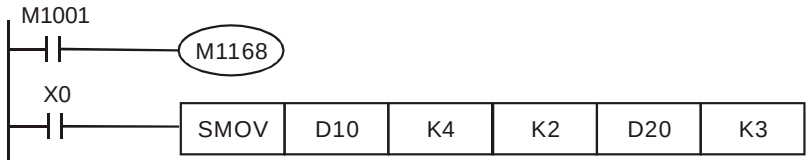
S: Source of data **m_1**: Start digit to be moved of the source data **m_2**: Number of digits (nibbles) to be moved of the source data **D**: Destination device **n**: Start digit of the destination position for the moved digits

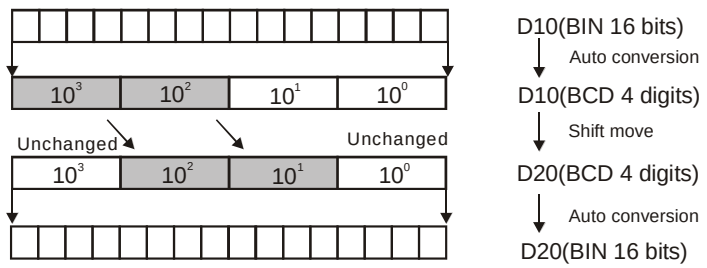
Explanations:

1. This instruction is able to re-allocate or combine data. When the instruction is executed, **m_2** digits of contents starting from digit **m_1** (from high digit to low digit) of **S** will be sent to **m_2** digits starting from digit **n** (from high digit to low digit) of **D**.
2. Range: **m_1** = 1 ~ 4; **m_2** = 1 ~ **m_1**; **n** = **m_2** ~ 4
3. See the specifications of each model for their range of use.
4. M1168 is designated by SMOV working mode. When M1168 = On, the program is in BIN mode. When M1168 = Off, the program is in BCD mode.

Program Example 1:

1. When M1168 = Off (in BCD mode) and X0 = On, the 4th (thousand) and 3rd (hundred) digit of the decimal value in D10 start to move to the 3rd (hundred) and 2nd (ten) digit of the decimal value in D20. 10^3 and 10^0 of D20 remain unchanged after this instruction is executed.
2. When the BCD value exceeds the range of 0 ~ 9,999, PLC will determine an operation error and will not execute the instruction. M1067, M1068 = On and D1067 records the error code OE18 (hex).



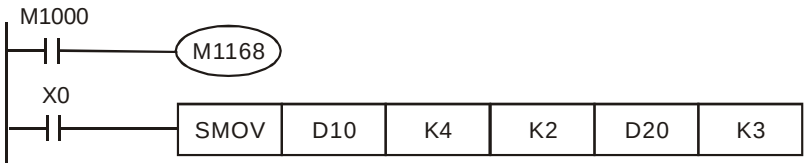


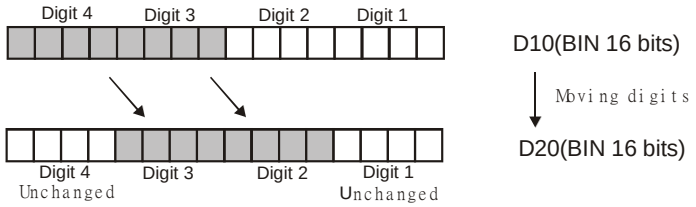
D10(BIN 16 bits)
↓ Auto conversion
D10(BCD 4 digits)
↓ Shift move
D20(BCD 4 digits)
↓ Auto conversion
D20(BIN 16 bits)

Before the execution, assume D10 = K1234 and D20 = K5678. After the execution, D10 will remain unchanged and D20 will become K5128.

Program Example 2:

When M1168 = On (in BIN mode) and SMOV instruction is in use, D10 and D20 will not be converted in BCD format but be moved in BIN format (4 digits as a unit).

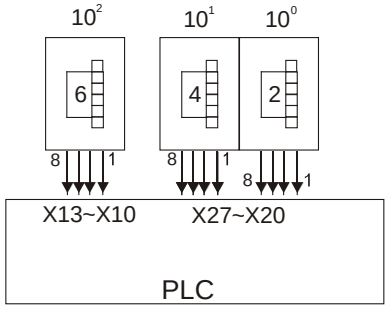


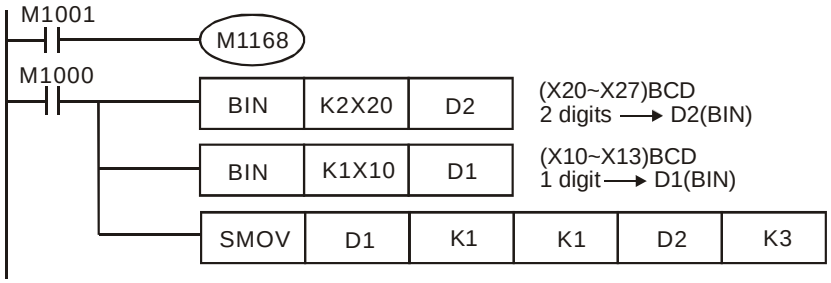


Before the execution, assume D10 = H1234 and D20 = H5678. After the execution, D10 will remain unchanged and D20 will become H5128.

Program Example 3:

1. This instruction can be used to combine the DIP switches connected to the input terminals with interrupted No.
2. Move the 2nd right digit of the DIP switch to the 2nd right digit of D2, and the 1st left digit of the DIP switch to the 1st right digit of D1.
3. Use SMOV instruction to move the 1st digit of D1 to the 3rd digit of D2 and combine the two DIP switches into one.





API	Mnemonic			Operands	Function
14	D	CML	P	(S) (D)	Compliment

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	CML, CMLP: 5 steps DCML, DCMLP: 9 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

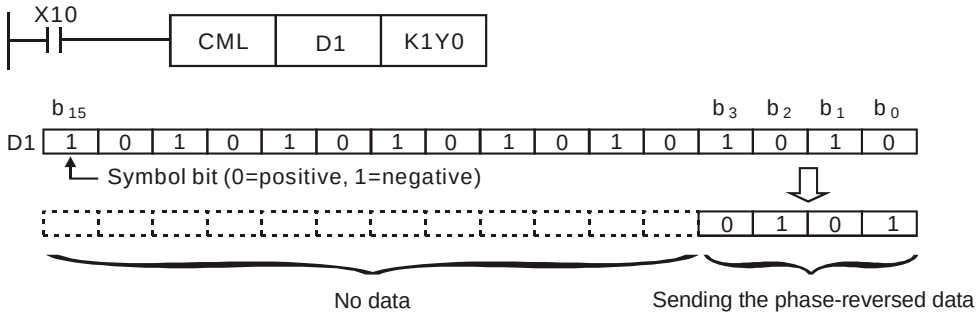
S: Source of data **D**: Destination device

Explanations:

1. If **S** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. This instruction can be used for phase-reversed output.
4. Reverse the phase (0→1, 1→0) of all the contents in **S** and send the contents to **D**. Given that the content is a constant K, K will be automatically converted into a BIN value.

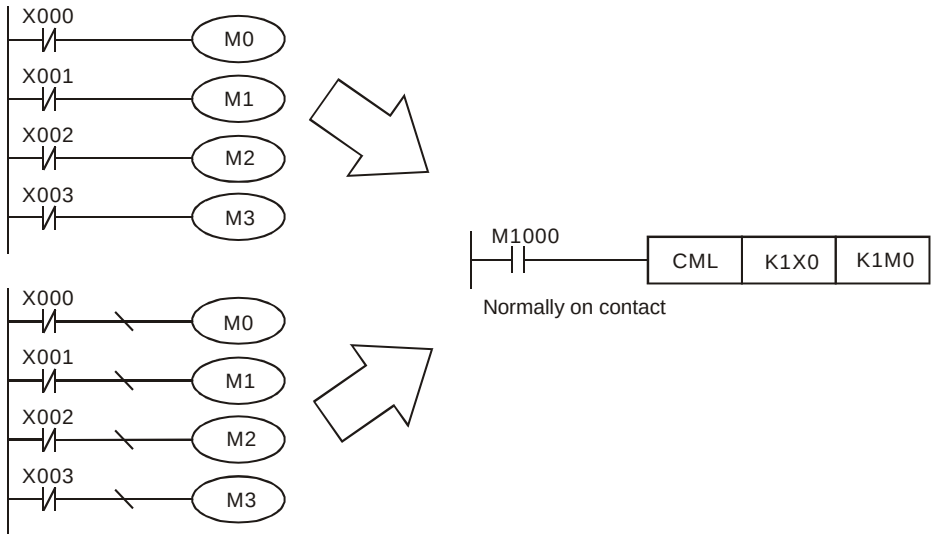
Program Example 1:

1. When X10 = On, b0 ~ b3 in D1 will be phase-reversed and send to Y0 ~ Y3.



Program Example 2:

The loop below can also adopt CML instruction (see right below).



API	Mnemonic		Operands	Function
15	BMOV	P	(S) (D) (n)	Block Move

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			BMOV, BMOVP: 7 steps
D								*	*	*	*	*	*			
n					*	*					*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start of source devices **D**: Start of destination devices **n**: Number of data to be moved

Explanations:

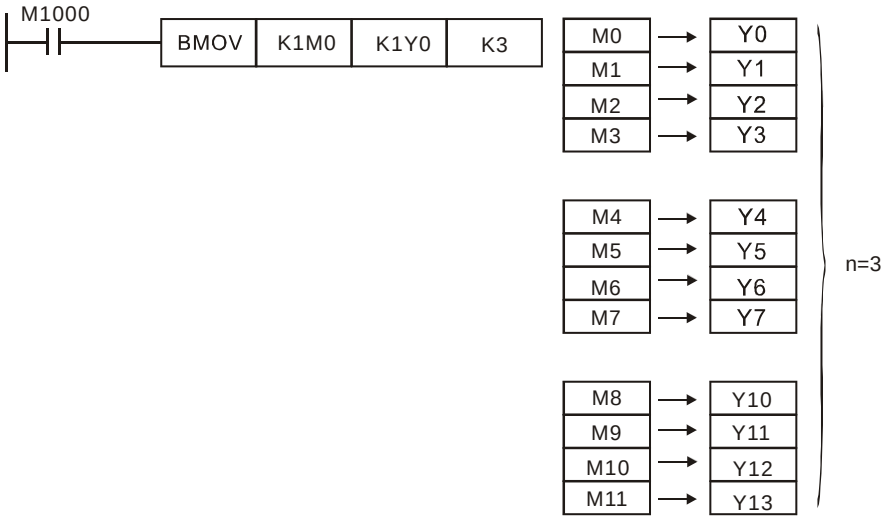
1. Range of **n**: 1 ~ 512
2. See the specifications of each model for their range of use.
3. The contents in n registers starting from the device designated by **S** will be moved to n registers starting from the device designated by **D**. If n exceeds the actual number of available source devices, only the devices that fall within the valid range will be used.

Program Example 1:

When X10 = On, the contents in registers D0 ~ D3 will be moved to the 4 registers D20 ~ D23.

Program Example 2:

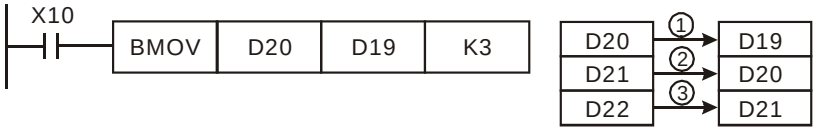
1. Assume the bit devices KnX, KnY, KnM and KnS are designated for moving, the number of digits of **S** and **D** has to be the same, i.e. their n has to be the same.
2. ES/EX/EC do not support the use of KnX, KnY, KnM, KnS and E, F index register modification.



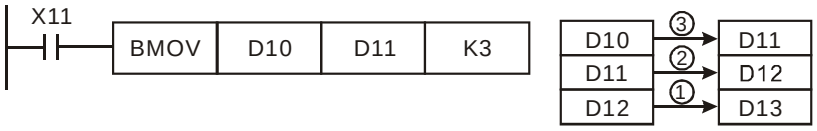
Program Example 3:

To avoid coincidence of the device numbers to be moved designated by the two operands and cause confusion, please be aware of the arrangement on the designated device numbers.

1. When **S** > **D**, the instruction is processed following the order ①→②→③



2. In EH/EH2/SV/EH3/SV2, when **S** < **D**, the instruction is processed following the order ①→②→③



3. In ES/EX/EC/SX, when **S** < **D**, avoid the number difference of "1" and the instruction is processed following the order ③→②→①. If the devices have the number difference of "1", the contents in D11 ~ D13 will all be the content in D10.

API	Mnemonic			Operands	Function
16	D	FMOV	P	(S) (D) (n)	Fill Move

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	FMOV, FMOVP: 7 steps DFMOV, DFMOVP: 13 steps
D								*	*	*	*	*	*			
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

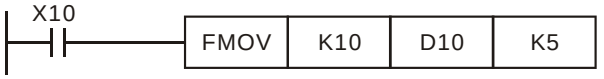
S: Source of data **D**: Destination of data **n**: Number of data to be moved

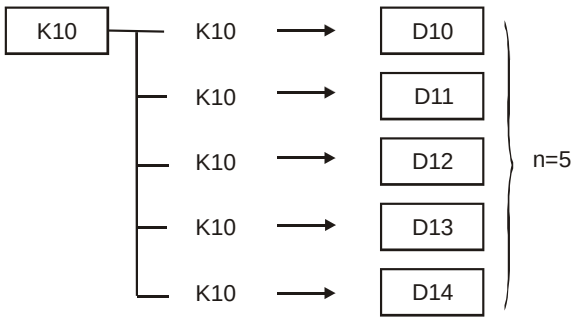
Explanations:

1. If **S** is used in device F, only 16-bit instruction is applicable.
2. Range of **n**: 1~ 512 (16-bit, 32-bit instructions)
3. See the specifications of each model for their range of use.
4. The contents in n registers starting from the device designated by **S** will be moved to n registers starting from the device designated by **D**. If n exceeds the actual number of available source devices, only the devices that fall within the valid range will be used.
5. ES/EX/EC do not support the use of KnX, KnY, KnM, KnS and E, F index register modification.

Program Example:

When X10 = On, K10 will be moved to the 5 consecutive registers starting from D10.





API	Mnemonic			Operands	Function
17	D	XCH	P	D_1 D_2	Exchange

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D_1								*	*	*	*	*	*	*	*	XCH, XCHP: 5 steps DXCH, DXCHP: 9 steps
D_2								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

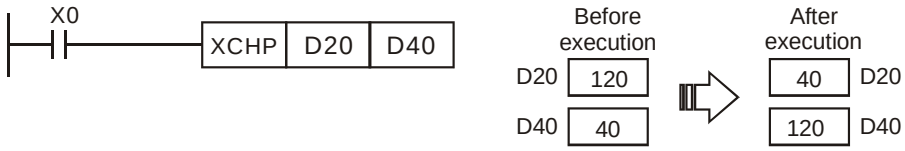
D_1: Data to be exchanged 1 **D_2**: Data to be exchanged 2

Explanations:

1. If **D_1** and **D_2** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. The contents in the devices designated by **D_1** and **D_2** will exchange.
4. Flag: M1303 (designated by XCH working mode).

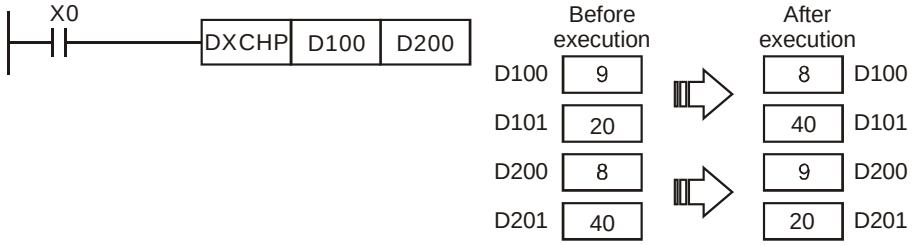
Program Example 1:

When X0 = Off→On, the contents in D20 and D40 exchange with each other.



Program Example 2:

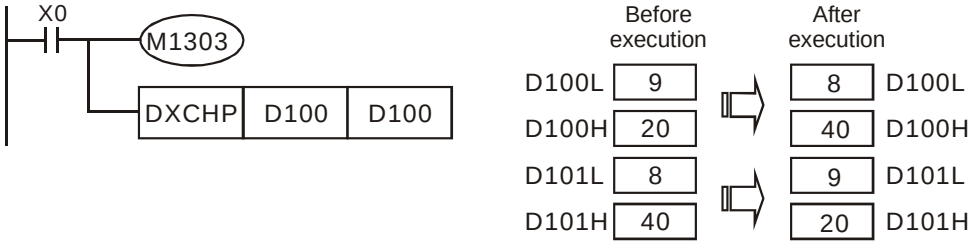
When X0 = Off → On, the contents in D100 and D200 exchange with each other.



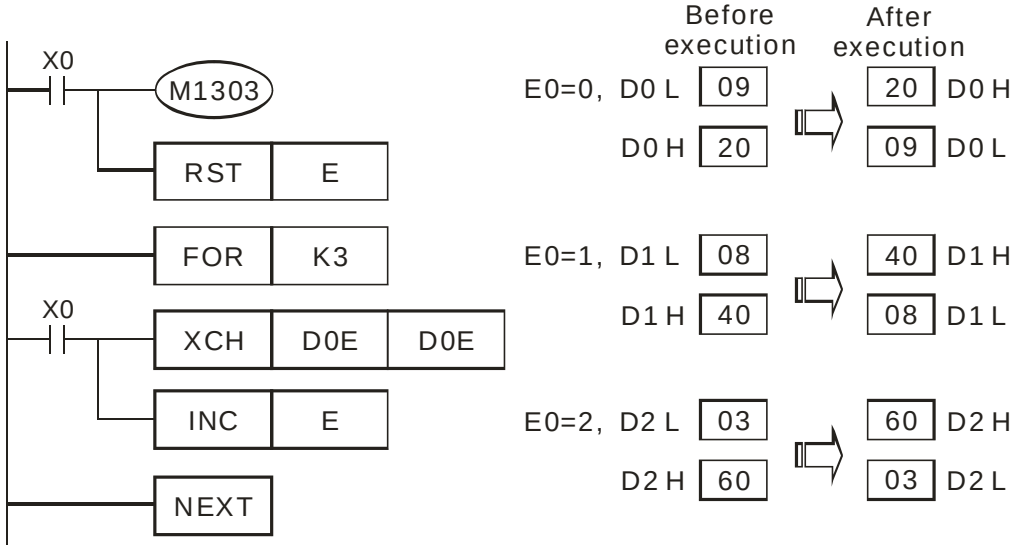
Remarks:

1. ES/EX/EC do not support M1303.
2. As a 16-bit instruction, when the devices designated by **D_1** and **D_2** are the same and M1303 = On, the upper and lower 8 bits of the designated devices exchange with each other.
3. As a 32-bit instruction, when the devices designated by **D_1** and **D_2** are the same and M1303 = On, the upper and lower 16 bits in the individual designated device exchange with each other.

4. When X0 = On and M1303 = On, the 16-bit contents in D100 and those in D101 will exchange with each other.



5. When X0 = ON and M1303 = ON, the high 8 bits and the low 8 bits in D0 are exchanged, the high 8 bits and the low 8 bits in D1 are exchanged., and the high 8 bits and the low 8 bits in D2 are exchanged.



API	Mnemonic			Operands	Function
18	D	BCD	P	(S) (D)	Binary Coded Decimal

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	BCD, BCDP: 5 steps DBCD, DBCDP: 9 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

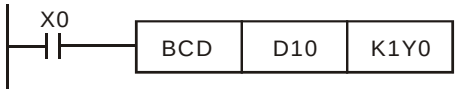
S: Source of data **D**: Conversion result

Explanations:

1. If **S** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. Flags: M1067 (operation error); M1068 (operation error); D1067 (error code)
4. The content in **S** (BIN value) is converted into BCD value and stored in **D**.
5. As a 16-bit (32-bit) instruction, when the conversion result exceeds the range of 0 ~ 9,999 (0 ~ 99,999,999), and M1067, M1068 = On, D1067 will record the error code 0E18 (hex).
6. The four arithmetic operations and applications in PLC and the execution of INC and DEC instructions are performed in BIN format. Therefore, if the user needs to see the decimal value display, simply use this instruction to convert the BIN value into BCD value.

Program Example:

1. When X0 = On, the binary value of D10 will be converted into BCD value, and the 1s digit of the conversion result will be stored in K1Y0 (Y0 ~ Y3, the 4 bit devices).



2. When D10 = 001E (hex) = 0030 (decimal), the execution result will be: Y0 ~ Y3 = 0000(BIN).

API	Mnemonic			Operands	Function
19	D	BIN	P	(S) (D)	Binary

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	BIN, BINP: 5 steps DBIN, DBINP: 9 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

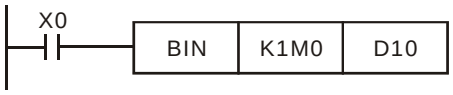
S: Source of data **D**: Conversion result

Explanations:

1. If **S** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. Flags: M1067 (operation error); M1068 (operation error); D1067 (error code)
4. The content in **S** (BCD value) is converted into BIN value and stored in **D**.
5. Valid range of **S** : BCD (0 ~ 9,999), DBCD (0 ~ 99,999,999)
6. Provided the content in S is not a BCD value (in hex and any one of its digits does not fall in the range of 0 ~ 9), an operation error will occur. M1067, M1068 = On and D1067 records the error code 0E18 (hex).
7. Constant K and H will automatically be converted into BIN format. Thus, they do not need to adopt this instruction.

Program Example:

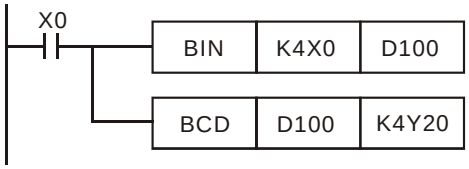
When X0 = On, the BCD value of K1M0 will be converted to BIN value and stored in D10.

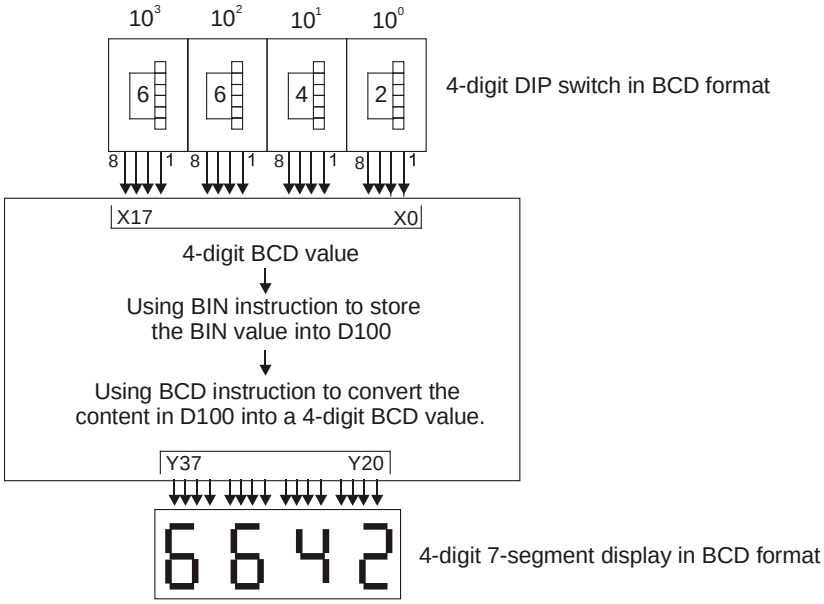


Remarks:

Explanations on BCD and BIN instructions:

1. When PLC needs to read an external DIP switch in BCD format, BIN instruction has to be first adopted to convert the read data into BIN value and store the data in PLC.
2. When PLC needs to display its stored data by a 7-segment display in BCD format, BCD instruction has to be first adopted to convert the data into BCD value and send the data to the 7-segment display.
3. When X0 = On, the BCD value of K4X0 is converted into BIN value and sent it to D100. The BIN value of D100 will then be converted into BCD value and sent to K4Y20.





4-digit DIP switch in BCD format

4-digit 7-segment display in BCD format

API	Mnemonic			Operands	Function
20	D	ADD	P	(S1) (S2) (D)	Addition

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	ADD, ADDP: 7 steps DADD, DADDP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Summand **S_2**: Addend **D**: Sum

Explanations:

1. If **S_1**, **S_2** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
4. This instruction adds **S_1** and **S_2** in BIN format and store the result in **D**.
5. The highest bit is symbolic bit 0 (+) and 1 (-), which is suitable for algebraic addition, e.g. 3 + (-9) = -6.
6. Flag changes in binary addition

   In 16-bit BIN addition,

   a) If the operation result = 0, zero flag M1020 = On.

   b) If the operation result < -32,768, borrow flag M1021 = On.

   c) If the operation result > 32,767, carry flag M1022 = On.

   In 32-bit BIN addition,

   a) If the operation result = 0, zero flag M1020 = On.

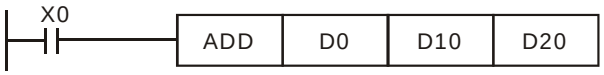
   b) If the operation result < -2,147,483,648, borrow flag M1021 = On.

   c) If the operation result > 2,147,483,647, carry flag M1022 = On.

Program Example 1:

In 16-bit BIN addition:

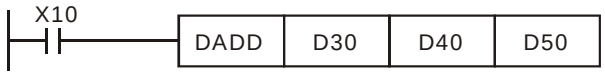
When X0 = On, the content in D0 will plus the content in D10 and the sum will be stored in D20.



Program Example 2:

In 32-bit BIN addition:

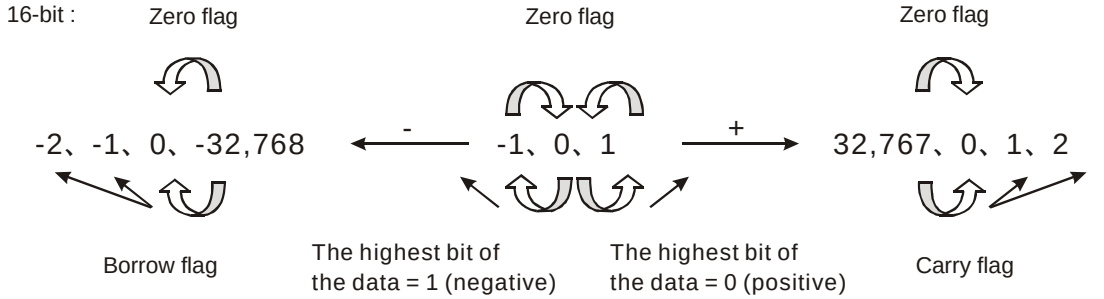
When X0 = On, the content in (D31, D30) will plus the content in (D41, D40) and the sum will be stored in (D51, D50). D30, D40 and D50 are low 16-bit data; D31, D41 and D51 are high 16-bit data.

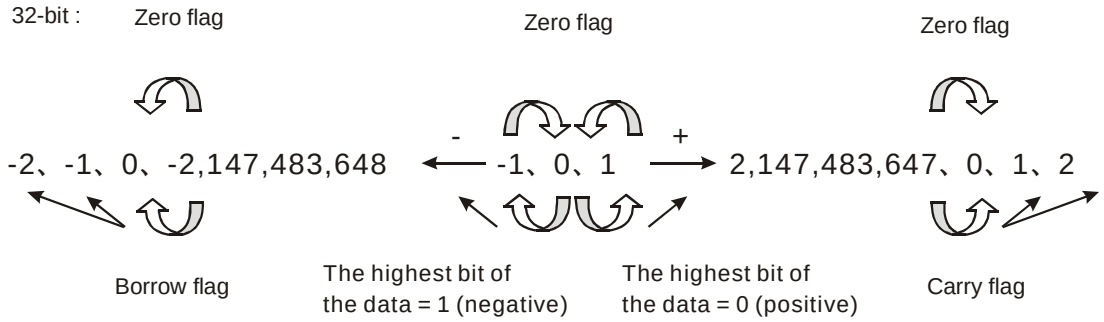


(D31, D30) + (D41, D40) = (D51, D50)

Remarks:

Flags and the positive/negative sign of the values:





API	Mnemonic			Operands	Function
21	D	SUB	P	(S1) (S2) (D)	Subtraction

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*	*	SUB, SUBP: 7 steps DSUB, DSUBP: 13 steps
S2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S1: Minuend **S2**: Subtrahend **D**: Remainder

Explanations:

1. If **S1**, **S2** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
4. This instruction subtracts **S1** and **S2** in BIN format and stores the result in **D**.
5. The highest bit is symbolic bit 0 (+) and 1 (-), which is suitable for algebraic subtraction.
6. Flag changes in binary subtraction

   In 16-bit instruction:

   a) If the operation result = 0, zero flag M1020 = On.

   b) If the operation result < –32,768, borrow flag M1021 = On.

   c) If the operation result > 32,767, carry flag M1022 = On.

   In 32-bit instruction:

   a) If the operation result = 0, zero flag M1020 = On.

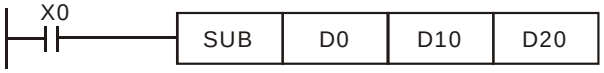
   b) If the operation result < –2,147,483,648, borrow flag M1021 = On.

   c) If the operation result > 2,147,483,647, carry flag M1022 = On.
7. For flag operations of SUB instruction and the positive/negative sign of the value, see the explanations in ADD instruction on the previous page.

Program Example 1:

In 16-bit BIN subtraction:

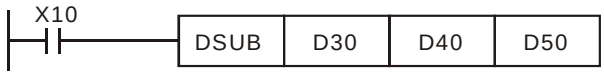
When X0 = On, the content in D0 will minus the content in D10 and the remainder will be stored in D20.



Program Example 2:

In 32-bit BIN subtraction:

When X10 = On, the content in (D31, D30) will minus the content in (D41, D40) and the remainder will be stored in (D51, D50). D30, D40 and D50 are low 16-bit data; D31, D41 and D51 are high 16-bit data.



(D31, D30) − (D41, D40) = (D51, D50)

API	Mnemonic			Operands	Function
22	D	MUL	P	(S1) (S2) (D)	Multiplication

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*	*	MUL, DMULP: 7 steps DMUL, DMULP: 13 steps
S2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*		

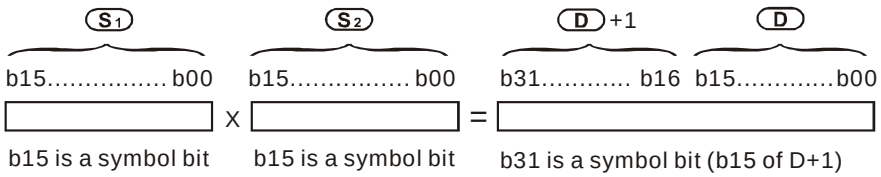
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S1: Multiplicand **S2**: Multiplicator **D**: Product

Explanations:

1. If **S1** and **S2** are used in device F, only 16-bit instruction is applicable.
2. If **D** is used in device E, only 16-bit instruction is applicable.
3. In 16-bit instruction, **D** occupies 2 consecutive devices.
4. In 32-bit instruction, **D** occupies 4 consecutive devices.
5. See the specifications of each model for their range of use.
6. This instruction multiplies **S1** by **S2** in BIN format and stores the result in **D**. Be careful with the positive/negative signs of **S1**, **S2** and **D** when doing 16-bit and 32-bit operations.
7. In 16-bit BIN multiplication,



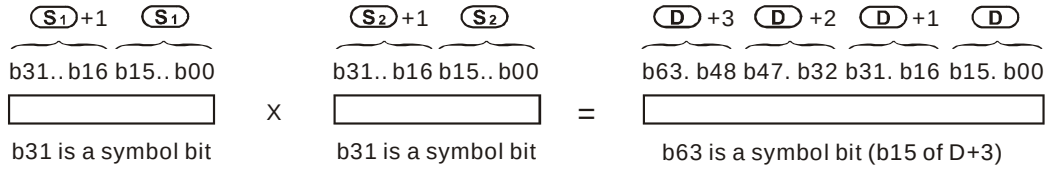
Symbol bit = 0 refers to a positive value.
Symbol bit = 1 refers to a negative value.

16-bit value x 16-bit value = 32-bit value

When D serves as a bit device, it can designate K1 ~ K4 and construct a 16-bit result, occupying consecutive 2 groups of 16-bit data. ES/EX/EC only stores low 16-bit data.

8. If the product of a 16-bit multiplication must be a 16-bit value (16-bit value x 16-bit value = 16-bit value), users have to use API 114 MUL16/MUL16P. Please refer to the explanation of API 114 MUL16/MUL16P for more information.

9. 32-bit BIN multiplication,



Symbol bit = 0 refers to a positive value.
Symbol bit = 1 refers to a negative value.

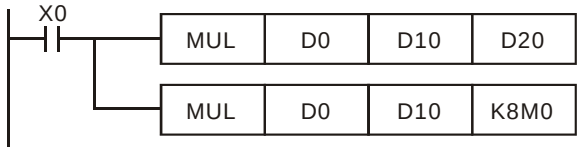
32-bit value x 32-bit value = 64-bit value

When D serves as a bit device, it can designate K1 ~ K8 and construct a 32-bit result, occupying consecutive 2 groups of 32-bit data.

10. If the product of a 32-bit multiplication must be a 32-bit value (32-bit value x 32-bit value = 32-bit value), users have to use API 114 MUL32/MUL32P. Please refer to the explanation of API 114 MUL32/MUL32P for more information.

Program Example:

The 16-bit D0 is multiplied by the 16-bit D10 and brings forth a 32-bit product. The higher 16 bits are stored in D21 and the lower 16-bit are stored in D20. On/Off of the most left bit indicates the positive/negative status of the result value.



API	Mnemonic			Operands	Function
23	D	DIV	P	(S1) (S2) (D)	Division

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*		DIV, DIVP: 7 steps DDIV, DDIVP: 13 steps
S2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

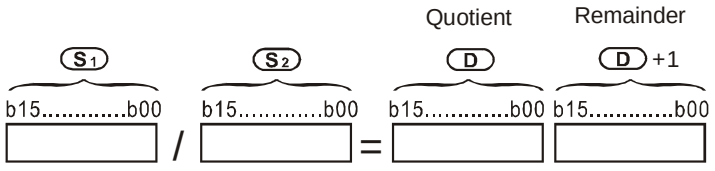
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S1: Dividend **S2**: Divisor **D**: Quotient and remainder

Explanations:

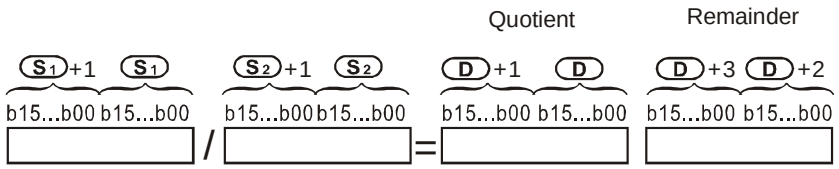
1. If **S1** and **S2** are used in device F, only 16-bit instruction is applicable.
2. If **D** is used in device E, only 16-bit instruction is applicable.
3. In 16-bit instruction, **D** occupies 2 consecutive devices.
4. In 32-bit instruction, **D** occupies 4 consecutive devices.
5. See the specifications of each model for their range of use.
6. This instruction divides **S1** and **S2** in BIN format and stores the result in **D**. Be careful with the positive/negative signs of **S1**, **S2** and **D** when doing 16-bit and 32-bit operations.
7. This instruction will not be executed when the divisor is 0. M1067 and M1068 will be On and D1067 records the error code 0E19 (hex).
8. In 16-bit BIN division,



When D serves as a bit device, it can designate K1 ~ K4 and construct a 16-bit result, occupying consecutive 2 groups of 16-bit data and bringing forth the quotient and remainder. ES/EX/EC is able to bring forth only quotient without the remainder.

If users want to store the quotient of a 16-bit division (leave out the remainder), they have to use AP I115 DIV16/DIV16P. Please refer to the explanation of API 115 DIV16/DIV16P for more information.

9. In 32-bit BIN division,

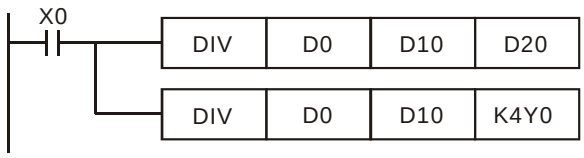


When D serves as a bit device, it can designate K1 ~ K8 and construct a 32-bit result, occupying consecutive 2 groups of 32-bit data and bringing forth the quotient and remainder.

If users want to store the quotient of a 32-bit division (leave out the remainder), they have to use AP I115 DIV32/DIV32P. Please refer to the explanation of API 115 DIV32/DIV32P for more information.

Program Example:

When X0 = On, D0 will be divided by D10 and the quotient will be stored in D20 and remainder in D21. On/Off of the highest bit indicates the positive/negative status of the result value.



API	Mnemonic			Operands	Function
24	D	INC	P	(D)	Increment

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	INC, INCP: 3 steps DINC, DINCP: 5 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

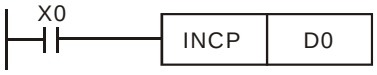
D: Destination device

Explanations:

1. If **D** is used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. If the instruction is not a pulse execution one, the content in the designated device D will plus “1” in every scan period whenever the instruction is executed.
4. This instruction adopts pulse execution instructions (INCP, DINCP).
5. In 16-bit operation, 32,767 pluses 1 and obtains -32,768. In 32-bit operation, 2,147,483,647 pluses 1 and obtains -2,147,483,648.
6. The operation results will not affect M1020 ~ M1022.

Program Example:

When X0 = Off→On, the content in D0 pluses 1 automatically.



API	Mnemonic			Operands	Function
25	D	DEC	P	(D)	Decrement

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	DEC, DECP: 3 steps DDEC, DDECP: 5 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

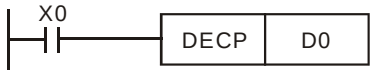
D: Destination device

Explanations:

1. If **D** is used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. If the instruction is not a pulse execution one, the content in the designated device D will minus “1” in every scan period whenever the instruction is executed.
4. This instruction adopts pulse execution instructions (DECP, DDECP).
5. In 16-bit operation, -32,768 minuses 1 and obtains 32,767. In 32-bit operation, -2,147,483,648 minuses 1 and obtains 2,147,483,647.
6. The operation results will not affect M1020 ~ M1022.

Program Example:

When X0 = Off→On, the content in D0 minuses 1 automatically.



API	Mnemonic			Operands	Function
26	W / D	AND	P	(S1) (S2) (D)	Logical Word AND

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	WAND, WANDP: 7 steps DAND, DANDP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

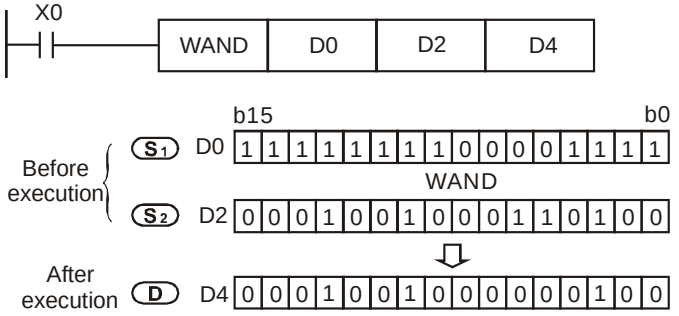
S_1: Source data device 1 **S_2**: Source data device 2 **D**: Operation result

Explanations:

1. If **S_1**, **S_2** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. This instruction conducts logical AND operation of **S_1** and **S_2** and stores the result in **D**.
4. Operation rule: The corresponding bit of the operation result in **D** will be "0" if any of the bits in **S_1** or **S_2** is "0".

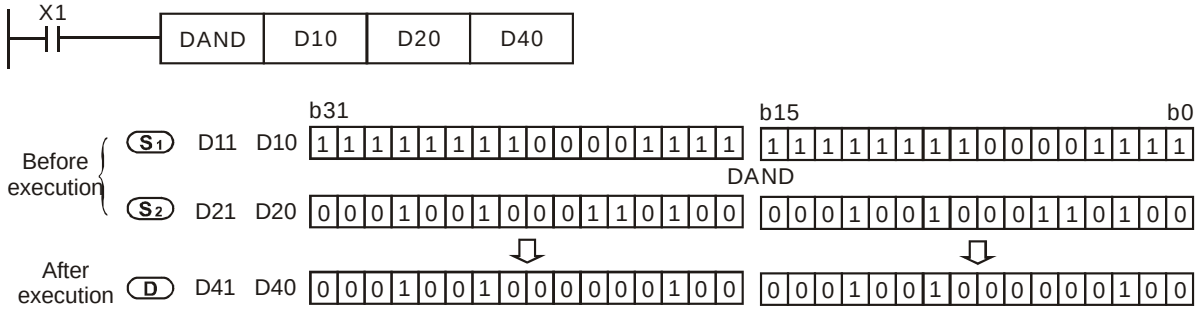
Program Example 1:

When X0 = On, the 16-bit D0 and D2 will perform WAND, logical AND operation, and the result will be stored in D4.



Program Example 2:

When X1 = On, the 32-bit (D11, D10) and (D21, D20) will perform DAND, logical AND operation, and the result will be stored in (D41, D40).



API	Mnemonic			Operands	Function
27	W/D	OR	P	(S1) (S2) (D)	Logical Word OR

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	WOR, WORP: 7 steps DOR, DORP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

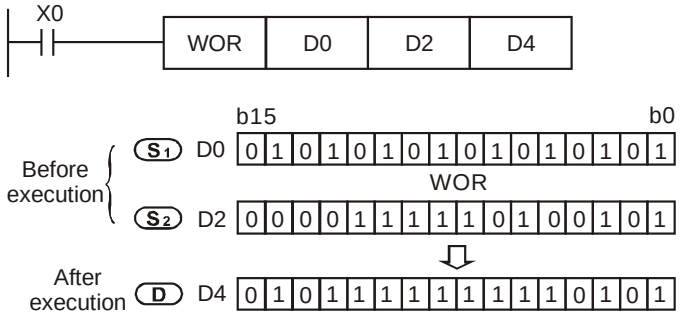
S_1: Source data device 1 **S_2**: Source data device 2 **D**: Operation result

Explanations:

1. If **S_1**, **S_2** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. This instruction conducts logical OR operation of **S_1** and **S_2** and stores the result in **D**.
4. Operation rule: The corresponding bit of the operation result in **D** will be "1" if any of the bits in **S_1** or **S_2** is "1".

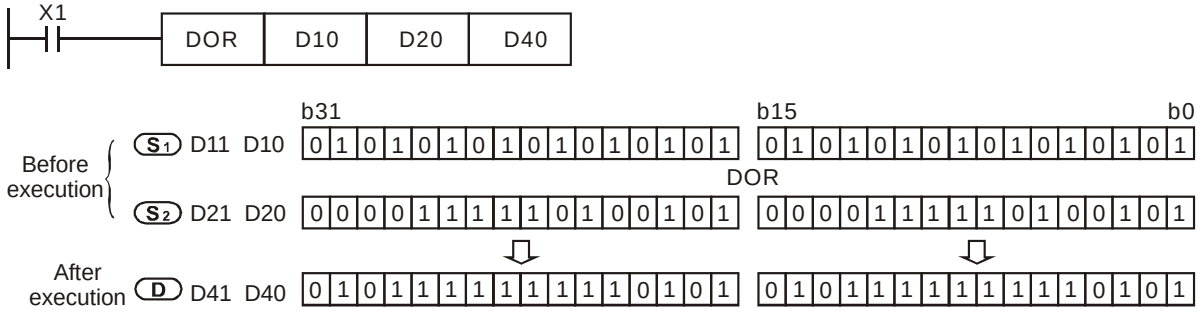
Program Example 1:

When X0 = On, the 16-bit D0 and D2 will perform WOR, logical OR operation, and the result will be stored in D4.



Program Example 2:

When X1 = On, the 32-bit (D11, D10) and (D21, D20) will perform DOR, logical OR operation, and the result will be stored in (D41, D40).



API	Mnemonic			Operands	Function
28	W/D	XOR	P	S1 S2 D	Logical Exclusive OR

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	WXOR, WXORP: 7 steps DXOR, DXORP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

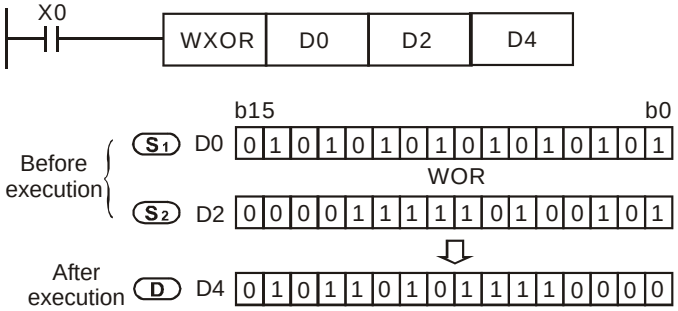
S_1: Source data device 1 **S_2**: Source data device 2 **D**: Operation result

Explanations:

1. If **S_1**, **S_2** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. This instruction conducts logical XOR operation of **S_1** and **S_2** and stores the result in **D**.
4. Operation rule: If the bits in **S_1** and **S_2** are the same, the corresponding bit of the operation result in **D** will be "0"; if the bits in **S_1** and **S_2** are different, the corresponding bit of the operation result in **D** will be "1".

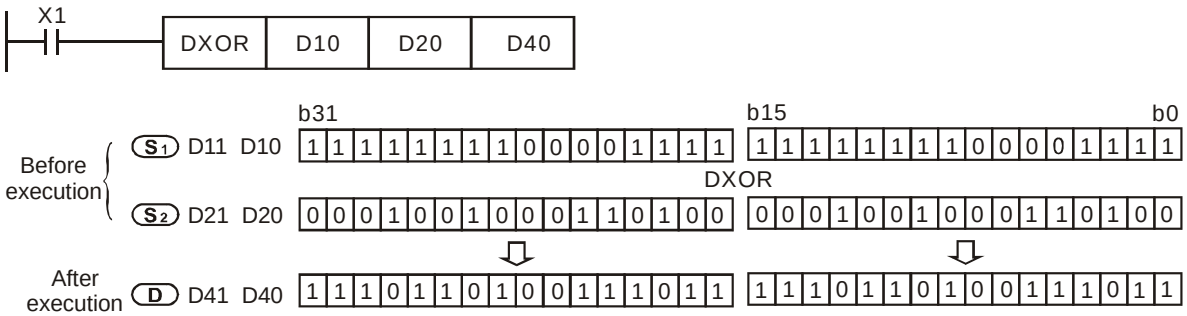
Program Example 1:

When X0 = On, the 16-bit D0 and D2 will perform WXOR, logical XOR operation, and the result will be stored in D4.



Program Example 2:

When X1 = On, the 32-bit (D11, D10) and (D21, D20) will perform DXOR, logical XOR operation, and the result will be stored in (D41, D40).



API	Mnemonic			Operands	Function
29	D	NEG	P	(D)	2's Complement (Negative)

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	NEG, NEGP: 3 steps DNEG, DNEGP: 5 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

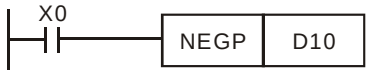
D: Device to store 2's complement

Explanations:

1. If **D** is used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. This instruction converts a negative BIN value into an absolute value.
4. This instruction adopts pulse execution instructions (NEGP, DNEGP).

Program Example 1:

When X0 = Off→On, the phase of every bit of the content in D10 will be reversed (0→1, 1→0) and pluses 1. The result will then be stored in D10.



Program Example 2:

Obtaining the absolute value of a negative value:

a) When the 15th bit of D0 is "1", M0 = On. (D0 is a negative value).

b) When M0 = Off→On, NEG instruction will obtain 2's complement of D0 and further its absolute value.

Program Example 3:

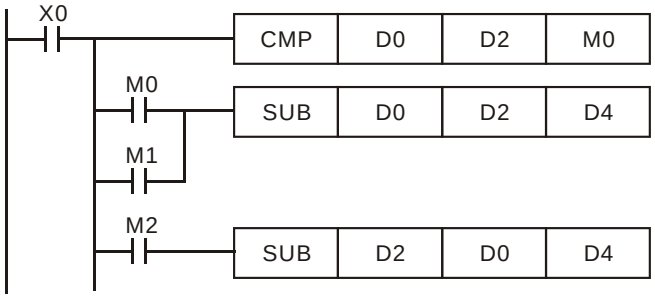
Obtaining the absolute value by the remainder of the subtraction. When X0 = On,

a) If D0 > D2, M0 = On.

b) If D0 = D2, M1 = On.

c) If D0 < D2, M2 = On.

d) D4 is then able to remain positive.

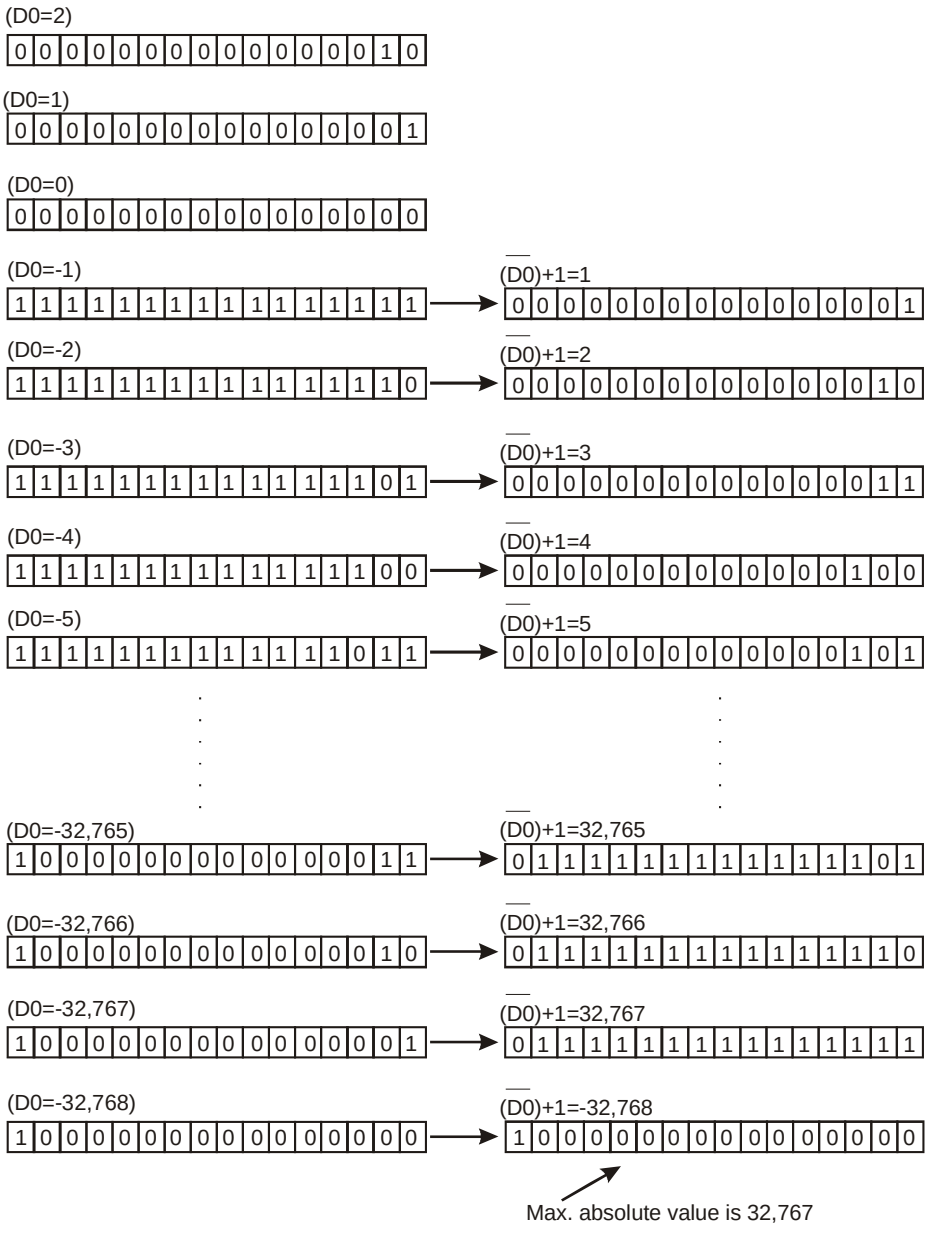


Remarks:

Negative value and its absolute value

a) The sign of a value is indicated by the highest (most left) bit in the register. 0 indicates that the value is a positive one and 1 indicates that the value is a negative one.

b) NEG instruction is able to convert a negative value into its absolute value.



API	Mnemonic			Operands	Function
30	D	ROR	P	(D) (n)	Rotation Right

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	ROR, RORP: 5 steps DROR, DRORP: 9 steps
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

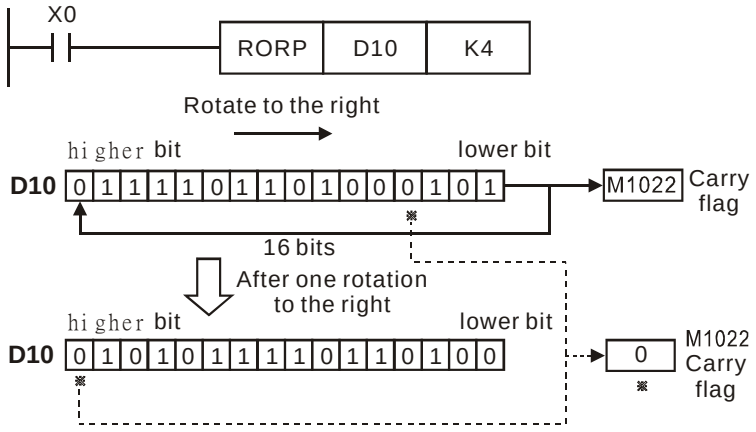
D: Device to be rotated **n**: Number of bits to be rotated in 1 rotation

Explanations:

1. If **D** is used in device F, only 16-bit instruction is applicable.
2. If **D** is designated as KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid.
3. Range of **n**: K1 ~ K16 (16-bit); K1 ~ K32 (32-bit)
4. See the specifications of each model for their range of use.
5. Flag: M1022 (carry flag)
6. This instruction rotates the device content designated by **D** to the right for **n** bits.
7. This instruction adopts pulse execution instructions (RORP, DRORP).

Program Example:

When X0 = Off→On, the 16 bits (4 bits as a group) in D10 will rotate to the right, as shown in the figure below. The bit marked with ※ will be sent to carry flag M1022.



API	Mnemonic			Operands	Function
31	D	ROL	P	(D) (n)	Rotation Left

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	ROL, ROLP: 5 steps DROL, DROLP: 9 steps
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

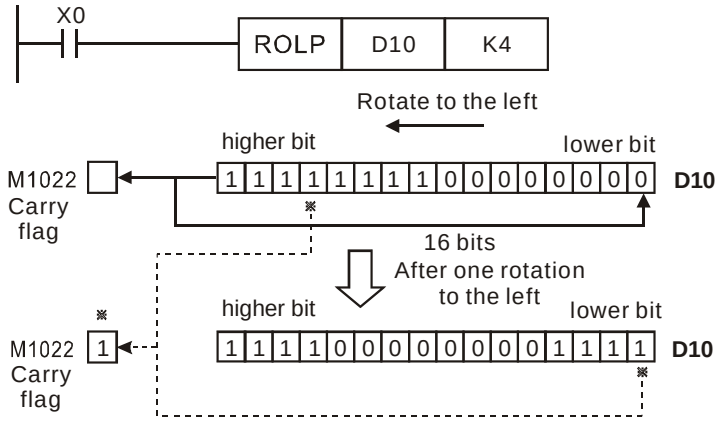
D: Device to be rotated **n**: Number of bits to be rotated in 1 rotation

Explanations:

1. If **D** is used in device F, only 16-bit instruction is applicable.
2. If **D** is designated as KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid.
3. Range of **n**: K1 ~ K16 (16-bit); K1 ~ K32 (32-bit)
4. See the specifications of each model for their range of use.
5. Flag: M1022 (carry flag)
6. This instruction rotates the device content designated by **D** to the left for **n** bits.
7. This instruction adopts pulse execution instructions (ROLP, DROLP).

Program Example:

When X0 = Off→On, the 16 bits (4 bits as a group) in D10 will rotate to the left, as shown in the figure below. The bit marked with ※ will be sent to carry flag M1022.



API	Mnemonic			Operands	Function
32	D	RCR	P	(D) (n)	Rotation Right with Carry

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	RCR, RCRP: 5 steps DRCR, DRCRP: 9 steps
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

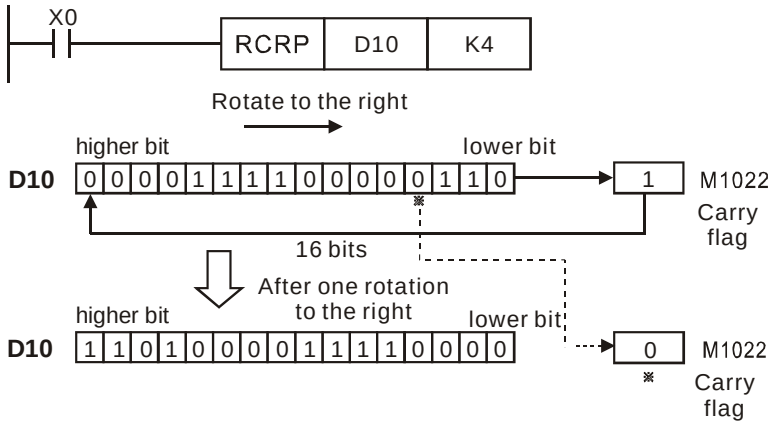
D: Device to be rotated **n**: Number of bits to be rotated in 1 rotation

Explanations:

1. If **D** is used in device F, only 16-bit instruction is applicable.
2. If **D** is designated as KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid.
3. Range of **n**: K1 ~ K16 (16-bit); K1 ~ K32 (32-bit)
4. See the specifications of each model for their range of use.
5. Flag: M1022 (carry flag)
6. This instruction rotates the device content designated by **D** together with carry flag M1022 to the right for **n** bits.
7. This instruction adopts pulse execution instructions (RCRP, DRCRP).

Program Example:

When X0 = Off→On, the 16 bits (4 bits as a group) in D10 together with carry flag M1022 (total 17 bits) will rotate to the right, as shown in the figure below. The bit marked with ※ will be sent to carry flag M1022.



API	Mnemonic			Operands	Function
33	D	RCL	P	(D) (n)	Rotation Left with Carry

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	RCL, RCLP: 5 steps DRCL, DRCLP: 9 steps
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

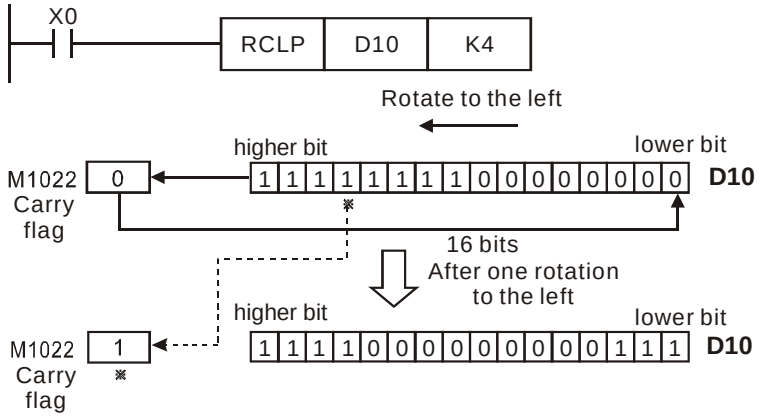
D: Device to be rotated **n**: Number of bits to be rotated in 1 rotation

Explanations:

1. If **D** is used in device F, only 16-bit instruction is applicable.
2. If **D** is designated as KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid.
3. Range of **n**: K1 ~ K16 (16-bit); K1 ~ K32 (32-bit)
4. See the specifications of each model for their range of use.
5. Flag: M1022 (carry flag)
6. This instruction rotates the device content designated by **D** together with carry flag M1022 to the left for **n** bits.
7. This instruction adopts pulse execution instructions (RCLP, DRCLP).

Program Example:

When X0 = Off→On, the 16 bits (4 bits as a group) in D10 together with carry flag M1022 (total 17 bits) will rotate to the left, as shown in the figure below. The bit marked with ※ will be sent to carry flag M1022.



API	Mnemonic		Operands	Function
34	SFTR	P	(S) (D) (n1) (n2)	Bit Shift Right

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												SFTR, SFTRP: 9 steps
D		*	*	*												
n_1					*	*										
n_2					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start No. of the shifted device **D**: Start No. of the device to be shifted **n_1**: Length of data to be shifted

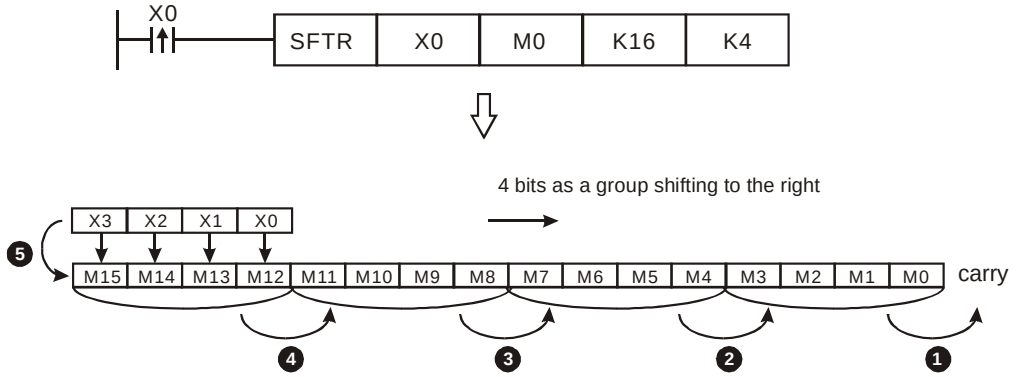
n_2: Number of bits to be shifted in 1 shift

Explanations:

1. Range of **n_1**: 1~ 1,024
2. Range of **n_2**: 1 ~ **n_1**
3. In ES/EX/EC , 1 ≤ **n_2** ≤ **n_1** ≤ 512
4. ES/EX/EC series MPU does not support E, F index register modification.
5. See the specifications of each model for their range of use.
6. This instruction shifts the bit device of **n_1** bits (desired length for shifted register) starting from **D** to the right for **n_2** bits. **S** is shifted into **D** for **n_2** bits to supplement empty bits.
7. This instruction adopts pulse execution instructions (SFTRP).

Program Example:

1. When X0 = Off→On, M0 ~M15 will form 16 bits and shifts to the right (4 bits as a group).
2. The figure below illustrates the right shift of the bits in one scan.
   - ❶ M3 ~ M0 → carry
   - ❷ M7 ~ M4 → M3 ~ M0
   - ❸ M11 ~ M8 → M7 ~ M4
   - ❹ M15 ~ M12 → M11 ~ M8
   - ❺ X3 ~ X0 → M15 ~ M12 completed



API	Mnemonic		Operands	Function
35	SFTL	P	(S) (D) (n1) (n2)	Bit Shift Left

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												SFTL, SFTLP: 9 steps
D		*	*	*												
n_1					*	*										
n_2					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start No. of the shifted device **D**: Start No. of the device to be shifted **n_1**: Length of data to be shifted

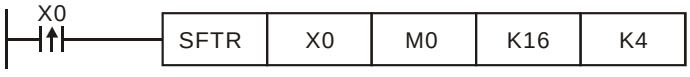
n_2: Number of bits to be shifted in 1 shift

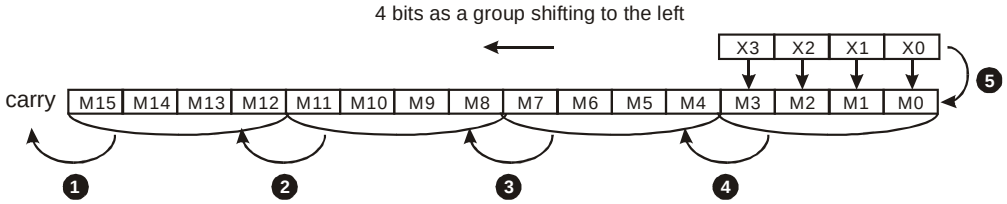
Explanations:

1. Range of **n_1**: 1~ 1,024
2. Range of **n_2**: 1 ~ **n_1**
3. In ES/EX/EC , 1 ≤ **n_2** ≤ **n_1** ≤ 512
4. ES/EX/EC series MPU does not support E, F index register modification.
5. See the specifications of each model for their range of use.
6. This instruction shifts the bit device of **n_1** bits (desired length for shifted register) starting from **D** to the left for **n_2** bits. **S** is shifted into **D** for **n_2** bits to supplement empty bits.
7. This instruction adopts pulse execution instructions (SFTLP).

Program Example:

1. When X0 = Off→On, M0 ~M15 will form 16 bits and shifts to the left (4 bits as a group).
2. The figure below illustrates the left shift of the bits in one scan.
   - ❶ M15 ~ M12 → carry
   - ❷ M11 ~ M8 → M15 ~ M12
   - ❸ M7 ~ M4 → M11 ~ M8
   - ❹ M3 ~ M0 → M7 ~ M4
   - ❺ X3 ~ X0 → M3 ~ M0 completed





API	Mnemonic		Operands	Function
36	WSFR	P	(S) (D) (n1) (n2)	Word Shift Left

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			WSFR, WSFRP: 9 steps
D								*	*	*	*	*	*			
n_1					*	*										
n_2					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start No. of the shifted device **D**: Start No. of the device to be shifted **n_1**: Length of data to be shifted

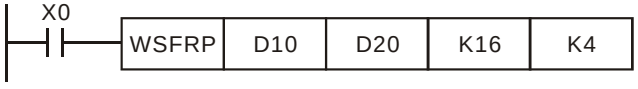
n_2: Number of words to be shifted in 1 shift

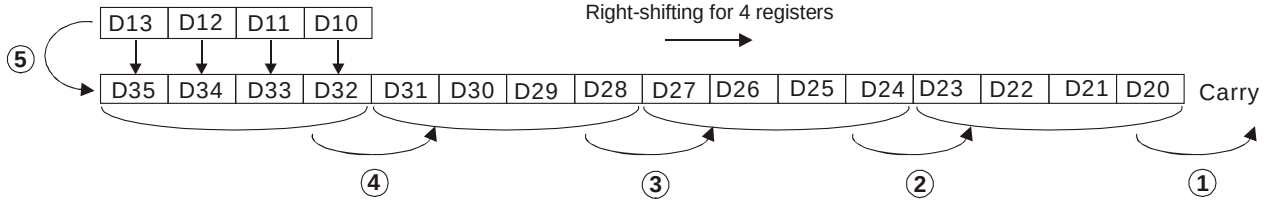
Explanations:

1. The type of devices designated by **S** and **D** has to be the same, e.g. K_nX, K_nY, K_nM, and K_nS as a category and T, C, and D as another category.
2. Provided the devices designated by **S** and **D** belong to K_n type, the number of digits of K_n has to be the same.
3. Range of **n_1**: 1~ 512
4. Range of **n_2**: 1 ~ **n_1**
5. See the specifications of each model for their range of use.
6. This instruction shifts the stack data of **n_1** words starting from **D** to the right for **n_2** words. **S** is shifted into **D** for **n_2** words to supplement empty words.
7. This instruction adopts pulse execution instructions (WSFRP)

Program Example 1:

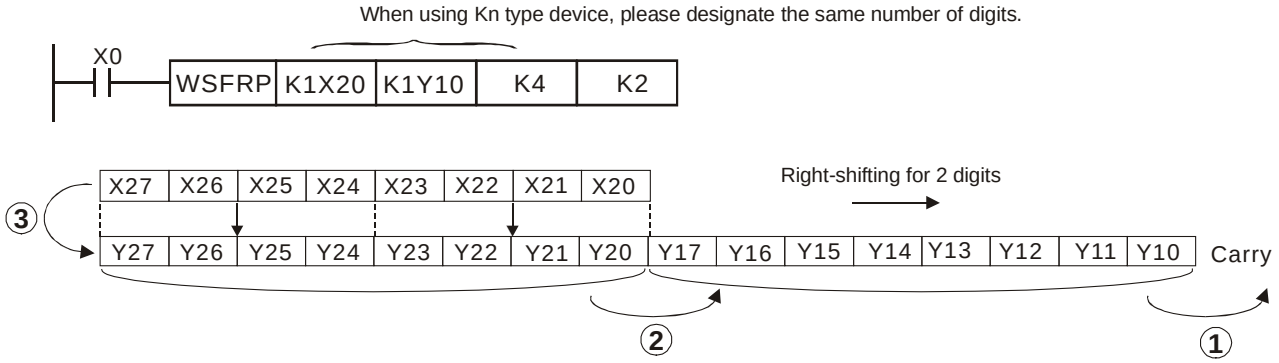
1. When X0 = Off→On, the 16 register stack data composed of D20 ~ D35 will shift to the right for 4 registers.
2. The figure below illustrates the right shift of the words in one scan.
   - ❶ D23 ~ D20 → carry
   - ❷ D27 ~ D24 → D23 ~ D20
   - ❸ D31 ~ D28 → D27 ~ D24
   - ❹ D35 ~ D32 → D31 ~ D28
   - ❺ D13 ~ D10 → D35 ~ D32 completed





Program Example 2:

1. When X0 = Off→On, the bit register stack data composed of Y10 ~ Y27 will shift to the right for 2 digits.
2. The figure below illustrates the right shift of the words in one scan.
   - ❶ Y17 ~ Y10 → carry
   - ❷ Y27 ~ Y20 → Y17 ~ Y10
   - ❸ X27 ~ X20 → Y27 ~ Y20 completed



API	Mnemonic		Operands	Function
37	WSFL	P	(S) (D) (n1) (n2)	Word Shift Left

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			WSFL, WSFLP: 9 steps
D								*	*	*	*	*	*			
n_1					*	*										
n_2					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start No. of the shifted device **D**: Start No. of the device to be shifted **n_1**: Length of data to be shifted

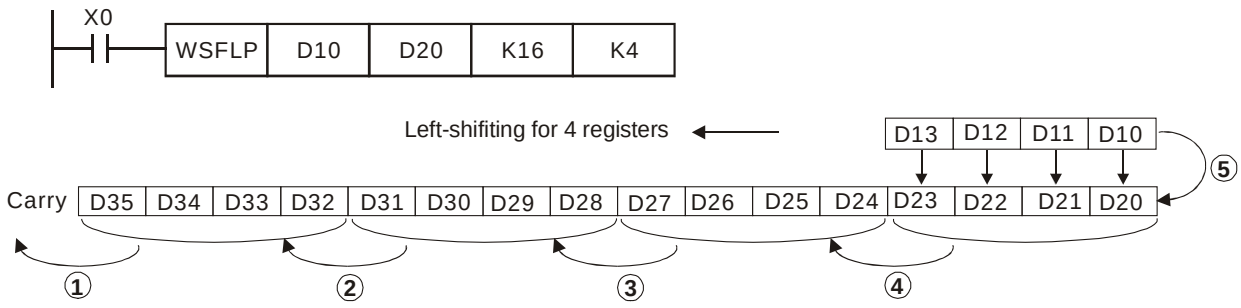
n_2: Number of words to be shifted in 1 shift

Explanations:

1. The type of devices designated by **S** and **D** has to be the same, e.g. K_nX, K_nY, K_nM, and K_nS as a category and T, C, and D as another category.
2. Provided the devices designated by **S** and **D** belong to K_n type, the number of digits of K_n has to be the same.
3. Range of **n_1**: 1~ 512
4. Range of **n_2**: 1 ~ **n_1**
5. See the specifications of each model for their range of use.
6. This instruction shifts the stack data of **n_1** words starting from **D** to the left for **n_2** words. **S** is shifted into **D** for **n_2** words to supplement empty words.
7. This instruction adopts pulse execution instructions (WSFLP)

Program Example:

1. When X0 = Off→On, the 16 register stack data composed of D20 ~ D35 will shift to the left for 4 registers.
2. The figure below illustrates the left shift of the words in one scan.
   - ❶ D35 ~ D32 → carry
   - ❷ D31 ~ D28 → D35 ~ D32
   - ❸ D27 ~ D24 → D31 ~ D28
   - ❹ D23 ~ D20 → D27 ~ D24
   - ❺ D13 ~ D10 → D23 ~ D20 completed



API	Mnemonic			Operands	Function
38		SFWR	P	(S) (D) (n)	Shift Register Write

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SFWR, SFWRP: 7 steps
D								*	*	*	*	*	*			
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Device of stack data written in **D**: Start No. of stack data **n**: Length of stack data

Explanations:

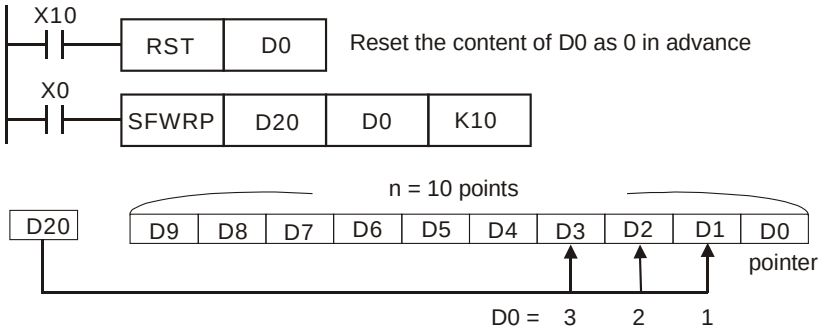
1. Range of **n**: 2 ~ 512
2. See the specifications of each model for their range of use.
3. Flag: M1022 (carry flag)
4. The stack data of **n** words starting from **D** are defined as “first-in, first-out” stack data and designate the first device as the pointer. When the instruction is executed, the content in the pointer pluses 1, and the content in the device designated by **S** will be written into the designated location in the “first-in, first-out” stack data designated by the pointer. When the content in the pointer exceeds **n** - 1, this instruction will not process any new value written in and the carry flag M1022 = On.
5. This instruction adopts pulse execution instructions (SFWRP)

Program Example:

1. Pointer D0 is reset as 0. When X0 = Off→On, the content in D20 will be sent to D1 and the content in pointer D0 becomes 1. After the content in D20 is changed, make X0 = Off→On again, and the content in D2 will be sent to D2 and the content in D0 becomes 2.
2. The figure below illustrates the shift and writing in 1~2 execution of the instruction.

   ❶ The content in D20 is sent to D1.

   ❷ The content in pointer D0 becomes 1.



Remarks:

This instruction can be used together with API 39 SFRD for the reading/writing of “first-in, first-out” stack data.

API	Mnemonic		Operands	Function
39	SFRD	P	(S) (D) (n)	Shift Register Read

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			SFRD, SFRDP: 7 steps
D								*	*	*	*	*	*	*	*	
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start No. of stack data **D**: Device of stack data read out **n**: Length of stack data

Explanations:

1. Range of **n**: 2 ~ 512
2. See the specifications of each model for their range of use.
3. Flag: M1020 (zero flag)
4. The stack data of **n** words starting from **S** are defined as “first-in, first-out” stack data. After the content in **S** minuses 1, the content in the device designated by (**S** + 1) will be written into the location designated by **D**, and (**S** + **n**-1) ~ (**S** + 2) will all right shift for one register while the content in (**S** + **n**-1) remains the same. When the content in **S** equals 0, this instruction will not process any new value read out and the zero flag M1020 = On.
5. This instruction adopts pulse execution instructions (SFRDP)

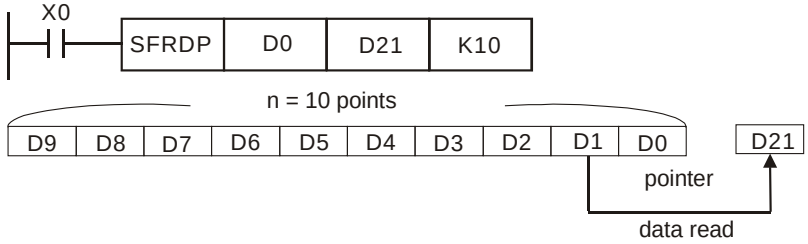
Program Example:

1. When X0 = Off→On, the content in D1 will be sent to D21 and D9~D2 will shift to the right for 1 register (content in D9 remains unchanged) and the content in D0 minus 1.
2. The figure below illustrates the shift and reading in 1~3 execution of the instruction.

   ❶ The content in D1 is sent to D21.

   ❷ D9 ~ D2 shift to the right for 1 register.

   ❸ The content in D0 minuses 1.



Remarks:

This instruction can be used together with API 38 SFWR for the reading/writing of “first-in, first-out” stack data.

API	Mnemonic		Operands	Function
40	ZRST	P	D_1 D_2	Zero Reset

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D_1		*	*	*							*	*	*			ZRST, ZRSTP: 5 steps
D_2		*	*	*							*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

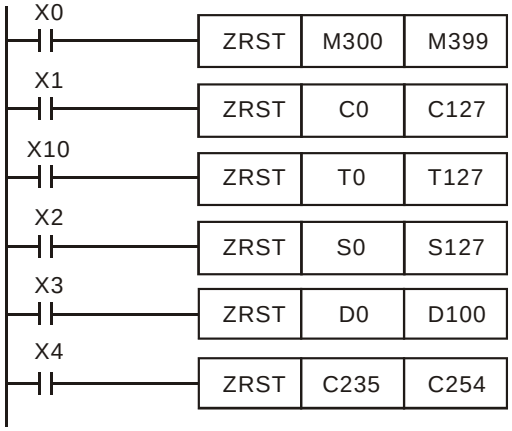
D_1: Start device of the range to be reset **D_2**: End device of the range to be reset

Explanations:

1. No. of operand $D_1 \leq$ No. of operand D_2.
2. D_1 and D_2 have to designate devices of the same type.
3. ES/EX/EC series MPU does not support E, F index register modification.
4. See the specifications of each model for their range of use.
5. When the instruction is executed, area from **D_1** to **D_2** will be cleared.
6. In ES/EX/EC , 16-bit counter and 32-bit counter cannot use ZRST instruction together.
7. In SA/EH, 16-bit counter and 32-bit counter can use ZRST instruction together.
8. When **D_1** > **D_2**, only operands designated by **D_2** will be reset.

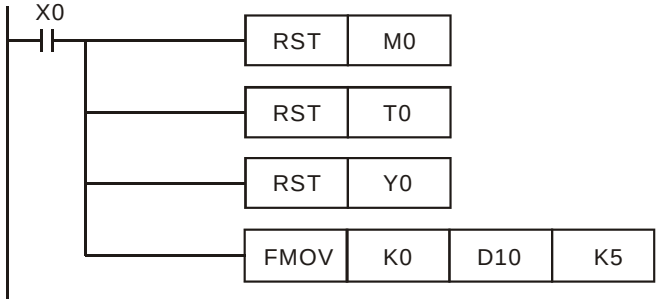
Program Example:

1. When X0 = On, auxiliary relays M300 ~ M399 will be reset to Off.
2. When X1 = On, 16 counters C0 ~ C127 will all be reset (writing in 0; contact and coil being reset to Off).
3. When X10 = On, timers T0 ~ T127 will all be reset (writing in 0; contact and coil being reset to Off).
4. When X2 = On, steps S0 ~ S127 will be reset to Off.
5. When X3 = On, data registers D0 ~ D100 will be reset to 0.
6. When X4 = On, 32-bit counters C235 ~ C254 will all be reset. (writing in 0; contact and coil being reset to Off)



Remarks:

1. Devices, e.g. bit devices Y, M, S and word devices T, C, D, can use RST instruction.
2. API 16 FMOV instruction is also to send K0 to word devices T, C, D or bit registers KnY, KnM, KnS for reset.



API	Mnemonic		Operands	Function
41	DECO	P	(S) (D) (n)	Decode

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*	*	*					*	*	*	*	*	DECO, DECOP: 7 steps
D		*	*	*							*	*	*	*	*	
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

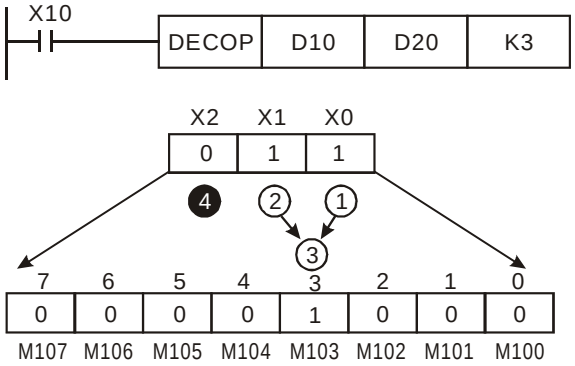
S: Source device to be decoded **D**: Device for storing the decoded result **n**: Length of decoded bits

Explanations:

1. Range of **n** when **D** is a bit device: 1 ~ 8
2. Range of **n** when **D** is a word device: 1 ~ 4
3. ES/EX/EC series MPU does not support E, F index register modification.
4. See the specifications of each model for their range of use.
5. The lower "**n**" bits of **S** are decoded and the results of "2^n" bits are stored in **D**.
6. This instruction adopts pulse execution instructions (DECOP)

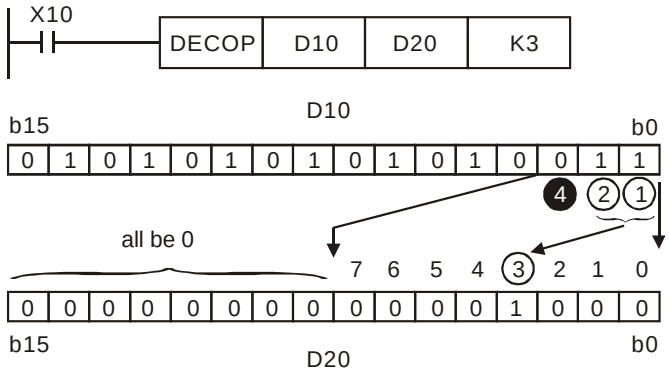
Program Example 1:

1. When **D** is used as a bit device, **n** = 1 ~ 8. Errors will occur if **n** = 0 or **n** > 8.
2. When **n** = 8, the maximum points to decode is 2^8 = 256 points. (Please be aware of the storage range of the devices after the decoding and do not use the devices repeatedly.)
3. When X10 = Off→On, this instruction will decode the content in X0 ~ X2 to M100 ~ M107.
4. When the source of data is 1 + 2 = 3, set M103, the 3rd bit starting from M100, as 1.
5. After the execution of this instruction is completed and X10 turns to Off, the content that has been decoded and output keeps acting.



Program Example 2:

1. When **D** is used as a word device, n = 1 ~ 4. Errors will occur if n = 0 or n > 4.
2. When **n** = 4, the maximum points to decode is 2^4 = 16 points.
3. When X10 = Off→On, this instruction will decode b2 ~ b0 in D10 to b7 ~ b0 in D20. b15 ~ b8 that have not been used in D20 will all become 0.
4. The lower 3 bits of D10 are decoded and stored in the lower 8 bits of D20. The higher 8 bits of D20 are all 0.
5. After the execution of this instruction is completed and X10 turns to Off, the content that has been decoded and output keeps acting.



API	Mnemonic		Operands	Function
42	ENCO	P	(S) (D) (n)	Encode

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*							*	*	*	*	*	ENCO, ENCOP: 7 steps
D											*	*	*	*	*	
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

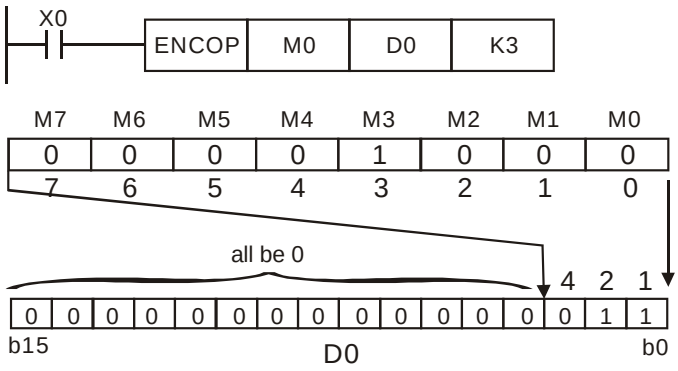
S: Source device to be encoded **D**: Device for storing the encoded result **n**: Length of encoded bits

Explanations:

1. Range of **n** when **S** is a bit device: 1 ~ 8
2. Range of **n** when **S** is a word device: 1 ~ 4
3. ES/EX/EC series MPU does not support E, F index register modification.
4. See the specifications of each model for their range of use.
5. The lower "2^n" bits of **S** are encoded and the result is stored in **D**.
6. If several bits of **S** are 1, the first bit that is 1 will be processed orderly from high bit to low bit.
7. If no bits of S is 1, M1067, M1068 = On and D1067 records the error code 0E1A (hex).
8. This instruction adopts pulse execution instructions (ENCOP)

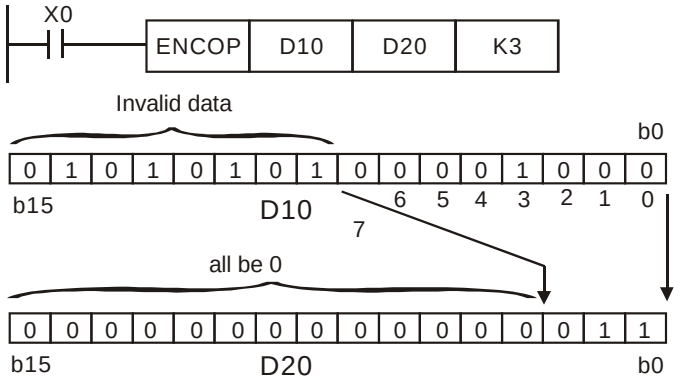
Program Example 1:

1. When **S** is used as a bit device, **n** = 1 ~ 8. Errors will occur if **n** = 0 or **n** > 8.
2. When **n** = 8, the maximum points to encode is 2^8 = 256 points.
3. When X10 = Off→On, this instruction will encode the 2^3 bits data (M0 ~ M7) and store the result in the lower 3 bits (b2 ~ b0) of D0. b15 ~ b3 that have not been used in D0 will all become 0.
4. After the execution of this instruction is completed and X10 turns to Off, the content in D remains unchanged.



Program Example 2:

1. When **S** is used as a word device, n = 1 ~ 4. Errors will occur if n = 0 or n > 4.
2. When **n** = 4, the maximum points to decode is 2^4 = 16 points.
3. When X10 = Off→On, this instruction will encode 2^3 bits (b0 ~ b7) in D10 and stores the result in the lower 3 bits (b2 ~ b0) of D20. b15 ~ b3 that have not been used in D20 will all become 0. b8 ~ b15 of D10 are invalid data.
4. After the execution of this instruction is completed and X10 turns to Off, the content in D remains unchanged.



API	Mnemonic			Operands	Function
43	D	SUM	P	(S) (D)	Sum of Active Bits

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SUM, SUMP: 5 steps DSUM, DSUMP: 9 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

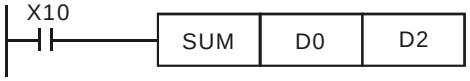
S: Source device **D**: Destination device for storing counted value

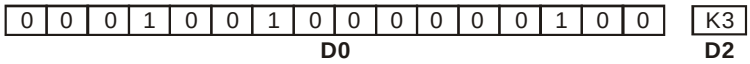
Explanations:

1. If **S** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. Flag: M1020 (zero flag)
4. Among the bits of **S**, the total of bits whose content is “1” will be stored in **D.**
5. When all the 16 bits of **S** are “0”, zero flag M1020 = On.
6. When 32- instruction is in use, **D** will occupy 2 registers.

Program Example:

When X10 = On, among the 16 bits of D0, the total of bits whose content is “1” will be stored in D2.





API	Mnemonic			Operands	Function
44	D	BON	P	(S) (D) (n)	Check Specified Bit Status

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	BON, BONP: 7 steps DBON, DBONP: 13 steps
D		*	*	*												
n					*	*					*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

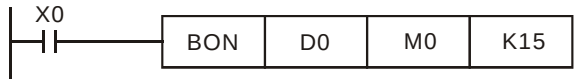
S: Source device **D**: Device for storing check result **n**: Bits specified for check

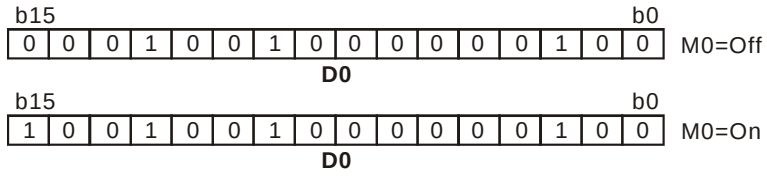
Explanations:

1. If **S** is used in device F, only 16-bit instruction is applicable.
2. Range of **n**: 0 ~ 15 (16-bit instruction); 0 ~ 31 (32-bit instruction)
3. See the specifications of each model for their range of use.
4. When the **n**th bit of **S** is “1”, D = On; when the **n**th bit of **S** is “0”, D = Off.

Program Example:

1. When X0 = On, assume the 15th bit of D0 is “1”, and M0 = On. Assume the 15th bit of D0 is “0”, and M0 = Off.
2. When X0 goes Off, M0 will remains in its previous status.





API	Mnemonic			Operands	Function
45	D	MEAN	P	(S) (D) (n)	Mean

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MEAN, MEANP: 7 steps
D								*	*	*	*	*	*	*	*	DMEAN, DMEANP: 13 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

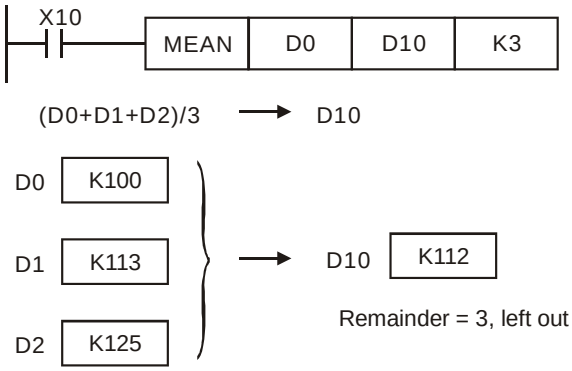
S: Start device to obtain mean value **D**: Destination device for storing mean value **n**: The number of consecutive source devices used

Explanations:

1. If **D** is used in device F, only 16-bit instruction is applicable.
2. Range of **n**: 1 ~ 64
3. In ES/EX/EC series models: Operand S cannot designate KnX, KnY, KnM, KnS.
4. ES/EX/EC series MPU does not support E, F index register modification.
5. See the specifications of each model for their range of use.
6. After the content of **n** devices starting from **S** are added up, the mean value of the result will be stored in **D**.
7. Remainders in the operation will be left out.
8. Provided the No. of designated device exceeds its normal range, only the No. within the normal range can be processed.
9. If **n** falls without the range of 1 ~ 64, PLC will determine it as an “instruction operation error”.

Program Example:

When X10 = On, the contents in 3 (n = 3) registers starting from D0 will be summed and then divided by 3. The obtained mean value will be stored in D10 and the remainder will be left out.



API	Mnemonic	Operands	Function
46	ANS	(S) (m) (D)	Timed Annunciator Set

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*					ANS: 7 steps
m					*											
D				*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

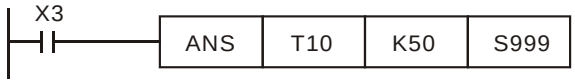
S: Timer for monitoring annunciator **m**: Time setting **D**: Annunciator device

Explanations:

1. Range of **S**: for SX T0 ~ T191; for EH3/SV2 T0 ~ T199.
2. **m** can designate K1 ~ K32,767 (unit: 100ms)
3. Range of **D**: for SX S896 ~ S1023; for EH3/SV2 S900 ~ S1023.
4. See the specifications of each model for their range of use.
5. Flags: M1048 (annunciator in action); M1049 (valid monitoring)
6. This instruction is used for enabling the annunciator.

Program Example:

If X3 = On for more than 5 seconds, annuniciator point S999 = On. Even X3 goes Off afterwards, S999 will still keep On. However, T10 will be reset to Off and the present value = 0.



API	Mnemonic		Operands	Function
47	ANR	P		Annunciator Reset

OP	Descriptions	Program Steps
N/A		ANR, ANRP: 1 steps

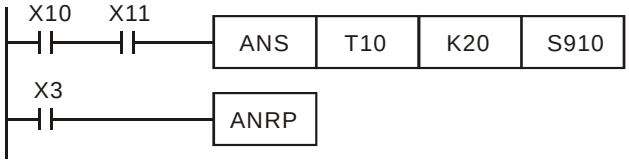
PULSE								16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2		ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Explanations:

1. No operand.
2. This instruction is used for resetting the annunciator.
3. When more than one annuciators are On, the annunciator of smaller number will be reset.
4. This instruction adopts pulse execution instructions (ANRP)

Program Example:

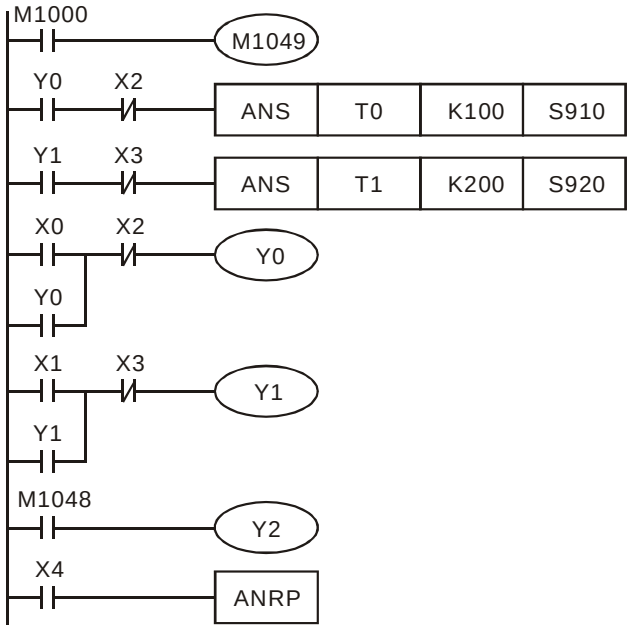
1. If X10 and X11 = On at the same time for more than 2 seconds, annuniciator point S910 = On. Even X10 and X11 go Off afterwards, S910 will still keep On. However, T10 will be reset to Off and the present value = 0.
2. When X10 and X11 are On at the same time for less than 2 seconds, the present value of T10 will be reset to 0.
3. When X3 goes from Off to On,
   S896 ~ S1023 in SA/SX/SX are able to reset the annunciators in action.
   S900 ~ S1023 in EH/EH2/ SV/EH3/SV2 are able to reset the annunciators in action.
4. When X3 goes from Off to On again, the annunciator with secondary smaller No. will be reset.



Remarks:

1. Flag:
   a) M1048 (annunciator in action): When M1049 = On, any of the annunciators among S896 ~ S1023 in SA/SX/SC or S900 ~ S1023 in EH/EH2/SV/EH3/SV2 starts output, M1048 will be On.
   b) M1049 (valid monitoring): When M1049 = On, D1049 will automatically display the annuciator of the smallest number in action.
2. Application of annunciators:
   I/O point configuration:

X0: Forward switch	Y0: Forward	S910: Forward annunciator
X1: Backward switch	Y1: Backward	S920: Backward annunciator
X2: Front position switch	Y2: Annunciator indicator	
X3: Back position switch		
X4: Annunciatro reset button		



(1) M1048 and D1049 are valid only when M1049 = On.

(2) When Y0 = On for more than 10 seconds and the device fails to reach the frong position X2, S910= On.

(3) When Y1 = On for more than 10 seconds and the device fails to reach the back position X3, S920= On.

(4) When backward switch X1 = On and backward device Y1 = On, Y1 will go Off only when the device reaches the back position switch X3.

(5) Y2 will be On when any annunciator is enabled. Whenever X4 is on, 1 annunciator in action will be reset. The reset starts from the annunciator with the smallest No.

API	Mnemonic			Operands	Function
48	D	SQR	P	(S) (D)	Square Root

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			SQR, SQRP: 5 steps DSQR, DSQRP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

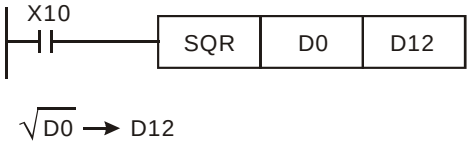
S: Source device **D**: Device for storing the result

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1067 (instruction operation error)
3. This instruction performs a square root operation on **S** and stores the result in **D**.
4. **S** can only be a positive value. If **S** is negative, PLC will regard it as an “instruction operation error” and will not execute this instruction. M1067 and M1068 = On and D1067 records the error code 0E1B (hex).
5. The operation result **D** should be integer only, and the decimal will be left out. Borrow flag M1021 = On.
6. When the operation result **D** = 0, zero flag M1020 = On.

Program Example:

When X10 = On, the instruction performs a square root on D0 and stores the result in D12.



$\sqrt{D0} \rightarrow D12$

API	Mnemonic			Operands	Function
49	D	FLT	P	(S) (D)	Convert BIN integer to binary floating point

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			FLT, FLTP: 5 steps DFLT, DFLTP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source device for conversion **D**: Device for storing the conversion result

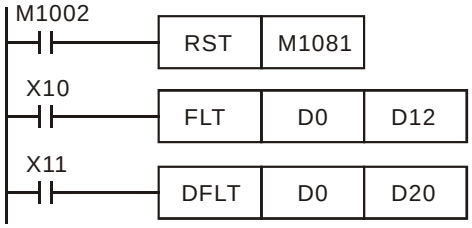
Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1081 (FLT instruction function switch); M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. **D** will occupy 2 consecutive devices
4. When M1081 is Off, BIN integer is converted into binary floating point value. At this time, **S** of the 16-bit instruction, FLT, occupies 1 register and **D** occupies 2 registers.
   a) If the absolute value of the conversion result ＞ max. floating value, carry flag M1022 = On.
   b) If the absolute value of the conversion result ＜ min. floating value, carry flag M1021 = On.
   c) If the conversion result is 0, zero flag M1020 = On.
5. When M1081 is On, binary floating point value is converted into BIN integer (digits after decimal point are left out). At this time, **S** of the 16-bit instruction, FLT, occupies 2 registers and **D** occupies 1 register (action same as that of INT instruction).
   a) If the conversion result exceeds the range of BIN integer available in **D** (for 16-bit: -32,768 ~ 32,767; for 32-bit: -2,147,483,648 ~ 2,147,483,647), D will obtain the maximum or minimum value and carry flag M1022 = On.
   b) If any digits is left out during the conversion, borrow flag M1021 = On.
   c) If **S** = 0, zero flag M1020 = On.
   d) After the conversion, **D** stores the result in 16 bits.

Program Example 1:

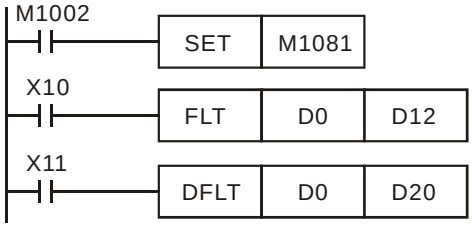
1. When M1081 = Off, the BIN integer is converted into binary floating point value.
2. When X10 = On, D0 (BIN integer) is converted into D13 and D12 (binary floating point value).
3. When X11 = On, D1 and D0 (BIN integer) are converted into D21 and D20 (binary floating point value).
4. If D0 = K10, X10 will be On. The 32-bit value of the converted floating point will be H41200000 and stored in 32-bit register D12 (D13).

5. If 32-bit register D0 (D1) = K100,000, X11 will be On. The 32-bit value of the converted floating point will be H47C35000 and stored in 32-bit register D20 (D21).



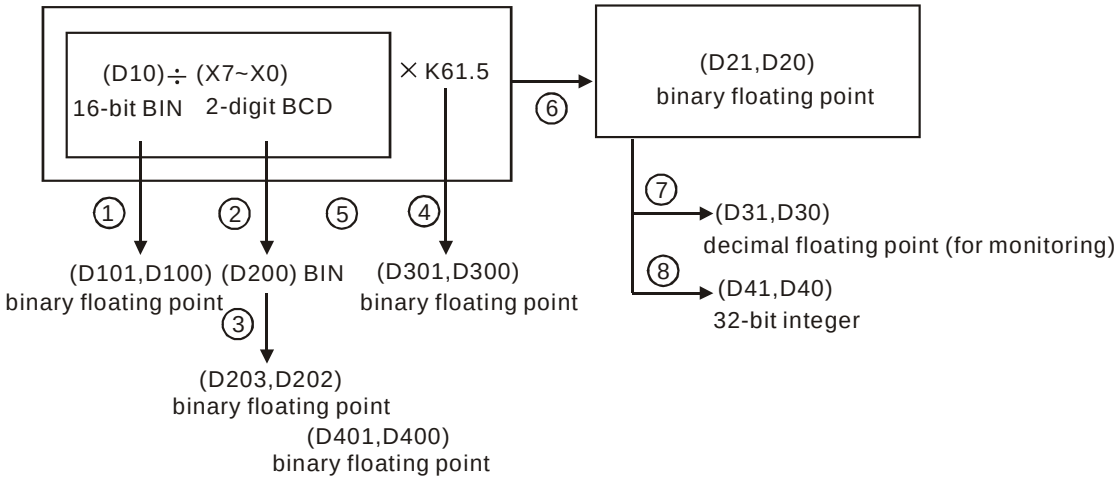
Program Example 2:

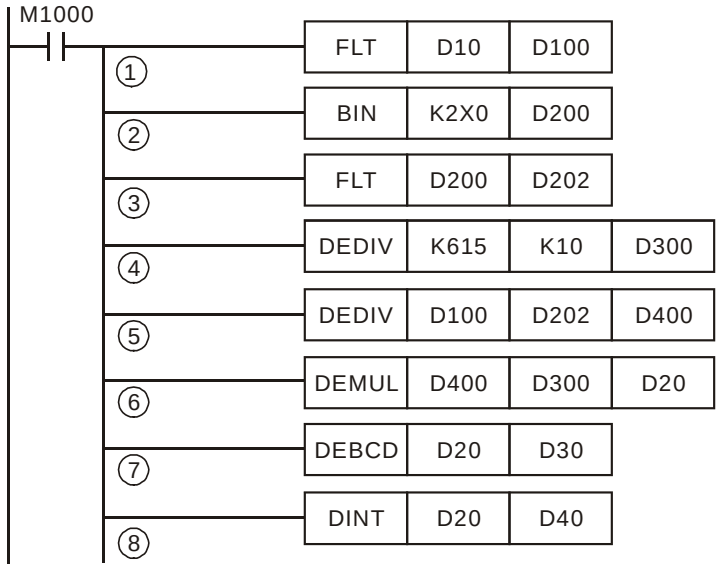
1. When M1081 = On, the binary floating point value is converted into BIN integer (the decimal is left out).
2. When X10 = On, D0 and D1 (binary floating point value) are converted into D12 (BIN integer). If D0 (D1) = H47C35000, the floating point value will be presented as 100,000. Due to that the value is larger than the value presentable by the 16-bit register D12, the result will be D12 = K32, 767 and M1022 = On.
3. When X11 = On, D1 and D0 (binary floating point value) are converted into D21 and D20 (BIN integer). If D0 (D1) = H47C35000, the floating point value will be presented as 100,000. The result will be stored in the 32-bit register D20 (D21).



Program Example 3:

Please use this instruction to complete the following operation.





① D10 (BIN integer) is converted to D101 and D102 (binary floating point value).

② X7 ~ X0 (BCD value) are converted to D200 (BIN value).

③ D200 (BIN integer) is converted to D203 and D202 (binary floating point value).

4The result of K615 ÷ K10 is stored in D301 and D300 (binary floating point value).

⑤ The result of binary decimal division (D101, D100) ÷ (D203, D202) is stored in D401 and D400 (binary floating point value).

⑥ The result of binary decimal multiplication (D401, D400) × (D301, D300) is stored in D21 and D20 (binary floating point value).

⑦ D21 and D20 (binary floating point value) are converted to D31 and D30 (decimal floating point value).

⑧ D21 and D20 (binary floating point value) are converted to D41 and D40 (BIN integer).

DVP-PLC applicable to the application instruction. ES includes ES/EX/EC/EC3-8K (FW V8.60 or later) (EC3: FW V8.40 or previous version); SX (FW V3.00); EH3 includes EH3/SV2.

ES/EX/EC series MPU does not support pulse execution type instructions (P instruction).

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	32-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
High Speed Processing	50	REF	-	✓	Refresh	✓	✓	✓	✓	5	-
	51	REFF	-	✓	Refresh and Filter Adjust	–	✓	✓	✓	3	-
	52	MTR	-	-	Input Matrix	–	–	✓	✓	9	-
	53	-	DHSCS	-	High Speed Counter Set	✓	✓	✓	✓	-	13
	54	-	DHSCR	-	High Speed Counter Reset	✓	✓	✓	✓	-	13
	55	-	DHSZ	-	High Speed Zone Compare	–	–	✓	✓	-	17
	56	SPD	-	-	Speed Detection	✓	✓	✓	✓	7	-
	57	PLSY	DPLSY	-	Pulse Y Output	✓	✓	✓	✓	7	13
	58	PWM	-	-	Pulse Width Modulation	✓	✓	✓	✓	7	-
	59	PLSR	DPLSR	-	Pulse Ramp	✓	✓	✓	✓	9	17
Handy Instructions	60	IST	-	-	Initial State	✓	✓	✓	✓	7	-
	61	SER	DSER	✓	Search a Data Stack	–	✓	✓	✓	9	17
	62	ABSD	DABSD	-	Absolute Drum Sequencer	–	–	✓	✓	9	17
	63	INCD	-	-	Incremental Drum Sequencer	–	–	✓	✓	9	-
	64	TTMR	-	-	Teaching Timer	–	–	✓	✓	5	-
	65	STMR	-	-	Special Timer	–	–	✓	✓	7	-
	66	ALT	-	✓	Alternate State	✓	✓	✓	✓	3	-
	67	RAMP	DRAMP	-	Ramp Variable Value	–	✓	✓	✓	9	17
	68	DTM	-	-	Data Transform and Move	–	✓	–	✓	9	-
	69	SORT	DSORT	-	Sort Tabulated Data	–	✓	✓	✓	11	21
Display of External Settings	70	TKY	DTKY	-	Ten Key Input	–	–	✓	✓	7	13
	71	HKY	DHKY	-	Hexadecimal Key Input	–	–	✓	✓	9	17
	72	DSW	-	-	Digital Switch	–	–	✓	✓	9	-
	73	SEGD	-	✓	Seven Segment Decoder	✓	✓	✓	✓	5	-
	74	SEGL	-	-	Seven Segment with Latch	✓	✓	✓	✓	7	-
	75	ARWS	-	-	Arrow Switch	–	–	✓	✓	9	-
	76	ASC	-	-	ASCII Code Conversion	–	–	✓	✓	11	-
	77	PR	-	-	Print (ASCII Code Output)	–	–	✓	✓	5	-
Serial I/O	78	FROM	DFROM	✓	Read CR Data in Special Modules	–	–	✓	✓	9	17
	79	TO	DTO	✓	Write CR Data into Special Modules	–	–	✓	✓	9	17
	80	RS	-	-	Serial Communication Instruction	✓	✓	✓	✓	9	-
	81	PRUN	DPRUN	✓	Parallel Run	–	–	✓	✓	5	9
	82	ASCI	-	✓	Converts Hex to ASCII	✓	✓	✓	✓	7	-
	83	HEX	-	✓	Converts ASCII to Hex	✓	✓	✓	✓	7	-
	84	CCD	-	✓	Check Code	–	✓	✓	✓	7	-
	85	VRRD	-	✓	Volume Read	–	✓	✓	✓	5	-
	86	VRSC	-	✓	Volume Scale	–	✓	✓	✓	5	-
	87	ABS	DABS	✓	Absolute Value	✓	✓	✓	✓	3	5
	88	PID	DPID	-	PID Control Loop	✓	✓	✓	✓	9	17

API	Mnemonic		Operands	Function
50	REF	P	D n	Refresh

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D	*	*														REF, REFP: 5 steps
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

D: Start device to be I/O refreshed **n**: Number of items to be I/O refreshed

Explanations:

1. **D** must designate X0, X10, Y0, Y10…the points whose 1s digit is "0". See remarks for more details.
2. Range of **n**: 8 ~ 256 (has to be the multiple of 8).
3. See the specifications of each model for their range of use.
4. The status of all PLC input/output terminals will be updated after the program scans to END. When the program starts to scan, the status of the external input terminal is read and stored into the memory of the input point. The output terminal will send the content in the output memory to the output device after END instruction is executed. Therefore, this instruction is applicable when the latest input/output data are needed for the operation.
5. **D** has to be designated to be X0, X10, Y0, Y10…such forms whose 1st digit is "0". Range of **n**: 8 ~ 256 (must be 8's multiple); otherwise it will be regarded as an error. The range varies in different models. See Remarks for more details.
6. EH3/SV2 FW V2.06 or later versions: new function is added to refresh the pulse position immediately.

Output Point	Y0	Y2	Y4	Y6
Flag to refresh the pulse position	M1640	M1641	M1642	M1643
The current value of the the output pulse	D1336/D1337	D1338/D1339	D1375/D1376	D1377/D1378

A. The the pulse position is refreshed, when PLC executes the output pulse instruction. If the program is too large, using this method to refresh the current output position may not be that accurate.

B. Use M1640~1643 to work with the REF instruction can have the pulse position refreshed immediately, not be affected by the scan cycle. (When using the flags with the REF insturciotn, the REF instruciton only reads the pulse positon. The actual input/output point refreshing is not executed.)

Program Example 1:

When X0 = On, PLC will read the status of input points X0 ~ X17 immediately and refresh the input signals without any input delay.

Program Example 2:

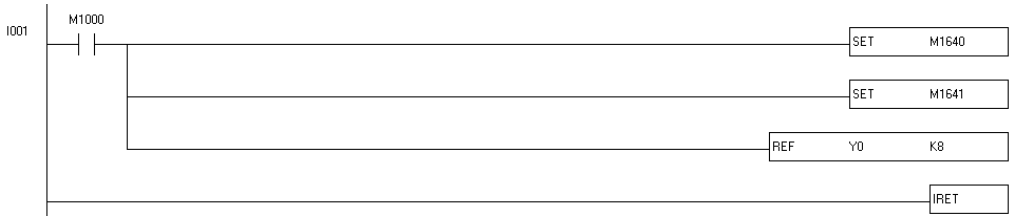
When X0 = On, the 8 output signal from Y0 ~ Y7 will be sent to output terminals and refreshed without having to wait for the END instruction for output.

Remarks:

The instruction only process the I/O points X0 ~ X17 and Y0 ~ Y17 of ES/EX/EC3-8K/SX series MPU, namely **n** = K8 or **n** = K16.

Program Example 3:

EH3/SV2 Series: When there is an external interrup in X0, the output high-speed output position in Y0 and Y2, and D1336/D1337 and D1338/D1339 will be refreshed immediately, not be affected by the scan cycle.



API	Mnemonic		Operands	Function
51	REFF	P	n	Refresh and Filter Adjust

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
n					*	*										REFF, REFFP: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

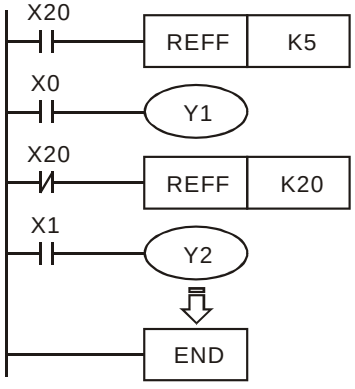
n: Response time (unit: ms)

Explanation:

1. Range of **n**: for SA/SX/SC, **n** = K0 ~ K20; for EH/EH2/SV/EH3/SV2, **n** = K0 ~ K60.
2. To avoid interferences, X0 ~ X17 of EC3-8K/EH3/SV2 series MPU and X0 ~ X7 of SX series MPU are equipped with digital filters on output terminals. Digital filters adjust the response time by REFF instruction. This instruction sets up **n** directly in D1020 (adjusting the response time of X0 ~ X7) and D1021 (adjusting the response time of X10 ~ X17).
3. Rules for adjusting the reponse time of the filter at X0 ~ X17:
   a) When the power of PLC turns from Off to On or the END instruction is being executed, the response time will be determined upon the contents in D1020 and D1021.
   b) You can use MOV instruction in the program to move the time values to D1020 and D1021 and make adjustments in the next scan.
   c) You can use REFF instruction to change the response time during the execution of the program. The changed response time will be move to D1020 and D1021 and you can make adjustments in the next scan.

Program Example:

1. When the power of PLC turns from Off to On, the response time of X0 ~ X17 will be determined by the contents in D1020 and D1021.
2. When X20 = On, REFF K5 will be executed and the response time will be changed to 5ms for the adjustment in the next scan.
3. When X20 = Off, the REFF K20 will be executed and the response time will be changed to 20ms for the adjustment in the next scan.



Remarks:

When inserting an interrupt or a high speed counter or using API 56 SPD instruction in the program, the corresponding signals at the input terminals will not delay and thus no filtering.

API	Mnemonic	Operands	Function
52	MTR	(S) (D1) (D2) (n)	Input Matrix

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															MTR: 9 steps
D_1		*														
D_2		*	*	*												
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start device of matrix input **D_1**: Start device of matrix output **D_2**: Corresponding start device for matrix scan

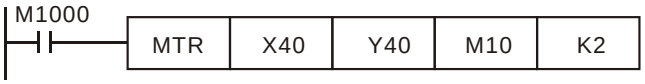
n: Number of arrays in matrix scan

Explanations:

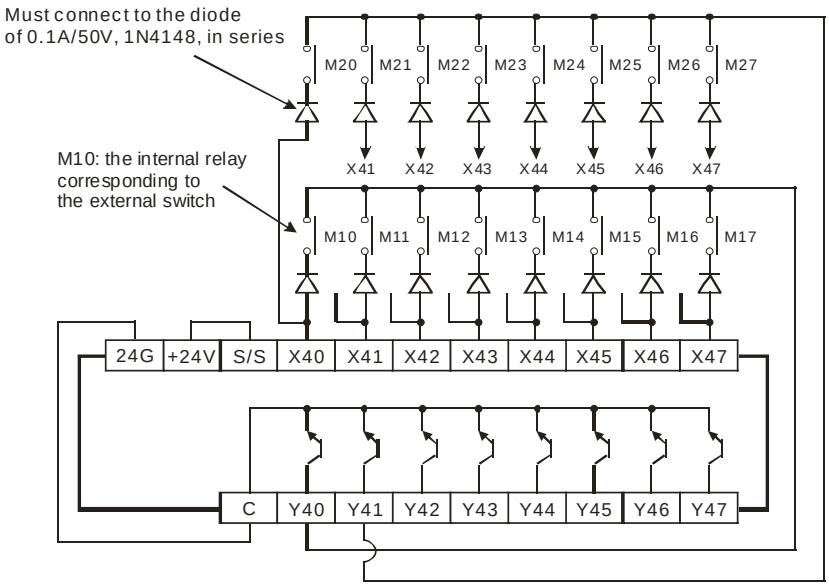
1. **S** must designate X0, X10…the X points whose 1st digit is "0" and occupies 8 consecutive points.
2. **D_1** must designate Y0, Y10…the Y points whose 1st digit is "0" and occupies n consecutive points.
3. **D_2** must designate Y0, M0. S0…the Y, M, S points whose 1st digit is "0".
4. Range of **n**: 2 ~ 8.
5. See the specifications of each model for their range of use.
6. Flag: M1029 (execution of the instruction is completed).
7. **S** is the start device No. of all input terminals connected to the matrix. Once **S** is designated, the 8 points following the No. will be the input terminals in the matrix.
8. **D_1** designate the start device No. of transistor output Y in the matrix scan.
9. This instruction occupies continuous 8 input devices starting from **S**. **n** external output terminals starting from **D_1** read the 8 switches of **n** arrays by matrix scan, obtaining 8 × **n** multiple-matrix input points. The status of scanned switches will be stored in the devices starting from **D_2**.
10. Maximum 8 input switches can be parallelly connected in 8 arrays and obtaining 64 input points (8 × 8 = 64).
11. When the 8-point 8-array matrix inputs are in use, the reading time of each array is approximately 25ms, totaling the reading of 8 arrays 200ms, i.e. the input signals with On/Off speed of over 200ms are not applicable in a matrix input.
12. The drive contact of this instruction uses normally On contact M1000.
13. Whenever this instruction finishes a matrix scan, M1029 will be On for one scan period.
14. There is no limitation on the number of times using the instruction, but only one instruction can be executed in one scan cycle.

Program Example:

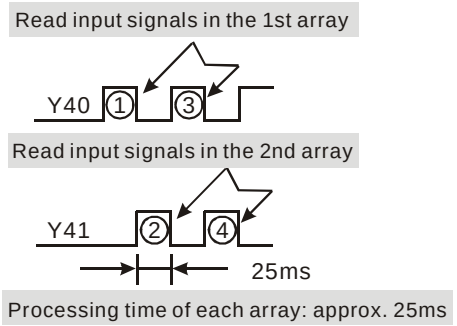
1. When PLC RUN, MRT instruction will start to be executed. The statuses of the external 2 arrays of 16 switches will be read in order and stored in the internal relays M10 ~ M17, M20 ~ M27.



2. The figure below illustrates the external wiring of the 2-array matrix input loop constructed by X40 ~ X47 and Y40 ~ Y41. The 16 switches correponds to the internal relays M10 ~ M17, M20 ~ M27. Should be used with MTR instruction.



3. See the figure above. The 8 points starting from X40 start to perform a matrix scan from Y40 ~ Y41 (n = 2). **D_2** designates that the start device No. of the read results is M10, indicating that the first array is read to M10 ~ M17 and the second array is read to M20 ~ M27.



API	Mnemonic	Operands	Function
53	D HSCS	(S1) (S2) (D)	High Speed Counter Set

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DHSCS: 13 steps
S_2												*				
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

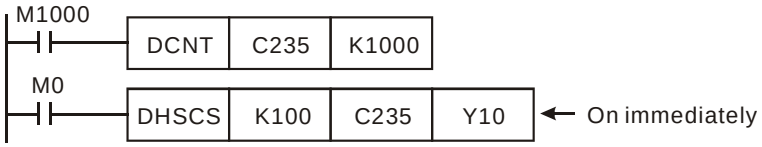
S_1: Comparative value **S_2**: No. of high speed counter **D**: Comparison result

Explanations:

1. **S_2** has to designate the No. of high speed counters C235 ~ C255. See remarks for more details.
2. **D** can designate I0□0; □ = 1 ~ 6. ES/EX/EC/EC3-8K series MPU does not support this.
3. **D** of ES/EX/EC/EC3-8K and SX series MPU does not support E, F index register modification.
4. See the specifications of each model for their range of use.
5. Flags: M1289 ~ M1294 are interruption disability of the high speed counters in EH3/SV2 series MPU. See Program Example 3 for more details.
6. The high speed counter inputs counting pulses from the corresponding external input terminals X0 ~ X17 by inserting an interruption. When the high speed counter designated in **S_2** pluses 1 or minuses 1, DHSCS instruction will perform a comparison immediately. When the present value in the high speed counter equals the comparative value designated in **S_1**, device designated in **D** will turn On. Even the afterward comparison results are unequal, the device will still be On.
7. If the devices specified as the device **D** are Y0 ~ Y17, when the compare value and the present value of the high-speed counter are equal, the comparison result will immediately output to the external inputs Y0 ~ Y17, and other Y devices will be affected by the scan cycle. However, M, S devices are immediate output and will not be affected by the scan cycle.

Program Example 1:

After PLC RUN and M0 = On, DHSCS instruction will be executed. When the present value in C235 changes from 99 to 100 or 101 to 100, Y10 will be On constantly.

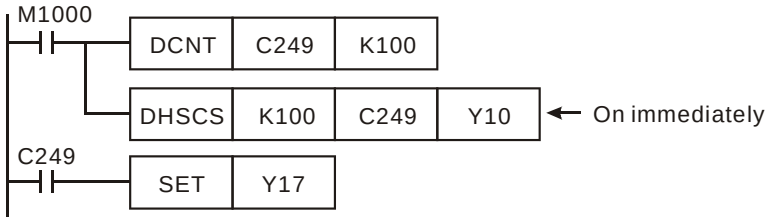


Program Example 2:

Differences between Y output of DHSCS instruction and general Y output:

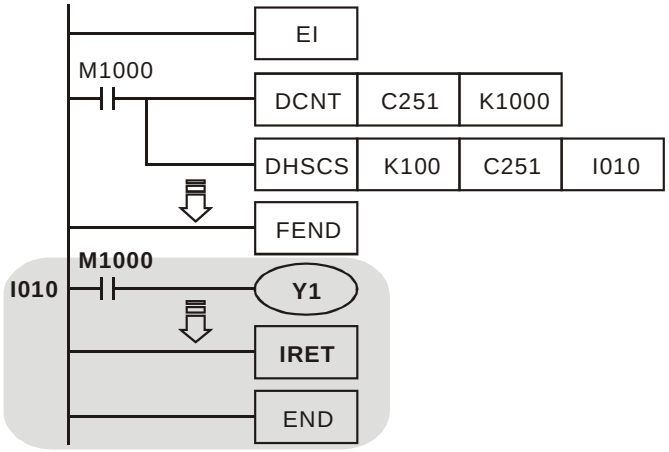
a) When the present value in C249 changes from 99 to 100 or 101 to 100, Y10 outputs immediately to the external output point by interruption and has nothing to do with the PLC scan time. However, the time will still be delayed by the relay (10ms) or transistor (10us) of the output module.

b) When the present value in C249 changes from 99 to 100, the drive contact of C249 will be On immediately. When the execution arrives at SET Y17, Y17 will still be affected by the scan time and will output after END instruction.



Program Example 3:

1. High speed counter interruption:
   a) Operand **D** of DHSCS instruction can designate I0□0, □ = 1 ~ 6, as the timing of interruption when the counting reaches its target.
   b) ES/EX/EC/EC3-8K series MPU does not support high speed counter interruption.
   c) SX series MPU supports high speed counter interruption. However, when DHSCS instruction designates an I interruption, the designated high speed counter cannot be used in DHSCS, DHSCR, DHSZ instructions. Misuse of high speed counter will result in error.
   d) For SX series MPU, when the counting reaches the target, the interruption will occur. X0 is the counter for counting input and the interruption No. is I010 (1 phase 2 inputs and A-B phase counter No. C246 ~ C254 can only designate I010). X1 designates I020; X2 designates I030; X3 designates I040; X4 designates I050; X5 designates I060, totaling 6 points.
   e) When the present value in C251 changes from 99 to 100 or 101 to 100, the program will jump to I010 and execute the interruption service subroutine.



2. In SX series MPU, M1059 is “I010 ~ I060 high speed counter interruption forbidden” flag.
3. In EH3/SV2 series MPU, M1289 ~ M1294 are the respectively for I010 ~ I060 “high speed counter interruption forbidden flags”, i.e. when M1294 = On, I060 interruption will be forbidden.

Interruption pointer I No.	Interruption forbidden flag
I010	M1289
I020	M1290
I030	M1291

Interruption pointer I No.	Interruption forbidden flag
I040	M1292
I050	M1293
I060	M1294

Remarks:

1. The output contact of the high speed counter and the comparative outputs of API 53 DHSCS, API 34 DHSCR and API 55 DHSZ instructions only perform comparison and contact outputs when there is a counting input. When using data operation instructions, e.g. DADD, DMOV, for changing the present value in the high speed counter or making the present value equals the set value, there will not be comparisons or comparative outputs because there is no counting inputs.
2. High speed counters supported by ES/EX/EC/EC3-8K series MPU (total bandwidth: 20kHz):

Type / Input	1-phase 1 input							1-phase 2 inputs			2-phase 2 inputs		
	C235	C236	C237	C238	C241	C242	C244	C246	C247	C249	C251	C252	C254
X0	U/D				U/D		U/D	U	U	U	A	A	A
X1		U/D			R		R	D	D	D	B	B	B
X2			U/D			U/D			R	R		R	R
X3				U/D		R	S			S			S

U: Progressively increasing input A: A phase input S: Input started

D: Progressively decreasing input B: B phase input R: Input cleared

a) Input points X0 and X1 can be planned as counters of higher speed (1 phase input can reach 20kHz). However, the total counting frequency of the two input points has to be smaller or equal 20kHz. Provided the input is a 2-phas input signal, the counting frequency will be approximately 4kHz. The frequency of the input points X2 and X3 (1-phase) can reach 10kHz.

b) For ES/EX/EC/EC3-8K series MPU, the uses of DHSCS instructio with DHSCR instruction cannot be more than 4 times.

3. High speed counters supported by SX series MPU (total bandwidth: 40kHz):

Type / Input	1-phase 1 input									1-phase 2 inputs			2-phase 2 inputs			
	C235	C236	C237	C238	C239	C240	C241	C242	C244	C246	C247	C249	C251	C252	C253	C254
X0	U/D						U/D		U/D	U	U	U	A	A	B	A
X1		U/D					R		R	D	D	D	B	B	A	B
X2			U/D					U/D			R	R		R		R
X3				U/D				R	S			S				S
X4					U/D											
X5						U/D										

U: Progressively increasing input A: A phase input S: Input started

D: Progressively decreasing input B: B phase input R: Input cleared

a) Input points X0 and X1 for 1-phase input can reach a frequency of 20kHz and X2 ~ X5 can reach 10kHz. 2-phase input (X0, X1) C251, C252 and C254 can reach a frequency of 4kHz and C253 reach 4kHz (only supports 4 times frequency counting).

b) Functions of the input point X5:

   i) When M1260 = Off, C240 is the general U/D high speed counuter.

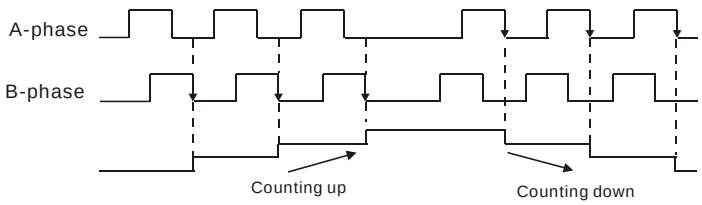
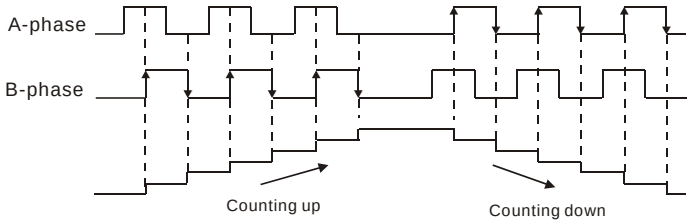
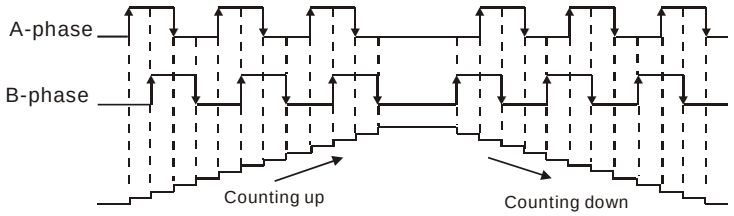
ii) When M1260 = On and C240 is enabled by DCNT instruction, X5 will be the shared reset signal for C235 ~ C239. The counter C240 will still receive the counting input signals from X5.

c) Counting modes (ES/EX/EC/EC3-8K and SA Series):

i) The 2-phase 2 inputs counting mode of the high speed counters in ES/EX/EC/EC3-8K (V5.5 and above) and SX series MPU is set by special D1022 with normal frequency, double frequency and 4 times frequency modes. The contents in D1022 will be loaded in in the first scan when PLC is switched from STOP to RUN.

Device No.	Function
D1022 = K1	Normal frequency mode selected
D1022 = K2 or 0	Double frequency mode selected (default)
D1022 = K4	4 times frequency mode selected

ii) Multiplied frequency mode (↑↓ indicates the occurrence of counting)

Counting mode		Counting wave pattern
2-phase 2 inputs	1 (Normal frequency)	
	2 (Double frequency)	
	4 (4 times frequency)	

4. EH3/SV2 series MPU supports high speed counters. C235 ~ C240 are program-interruption 1-phase high speed counter with a total bandwidth of 20kHz, can be used alone with a counting frequency of up to 10kHz. C241 ~ C254 are hardware high speed counter (HHSC). There are four HHSC in EH3/SV2 series MPU, HHSC0 ~ 3. The pulse input frequency of HHSC0~4 can reach 200kHz (1 phase or A-B phase). The pulse input frequency of HHSC 2 (X10, X11) in the 20 points of the EH3 series MPU can reach 20kHz, among which:

C241, C246 and C251 share HHSC0

C242, C247 and C252 share HHSC1

C243, C248 and C253 share HHSC2

C244, C249 and C254 share HHSC3

- Every HHSC can only be designated to one counter by DCNT instruction.

- There are three counting modes in every HHSC (see the table below):
  i) 1-phase 1 input refers to “pulse/direction” mode.
  ii) 1-phase 2 inputs refers to “clockwise/counterclockwise (CW/CCW)” mode.
  iii) 2-phase 2 inputs refers to “A-B phase” mode.

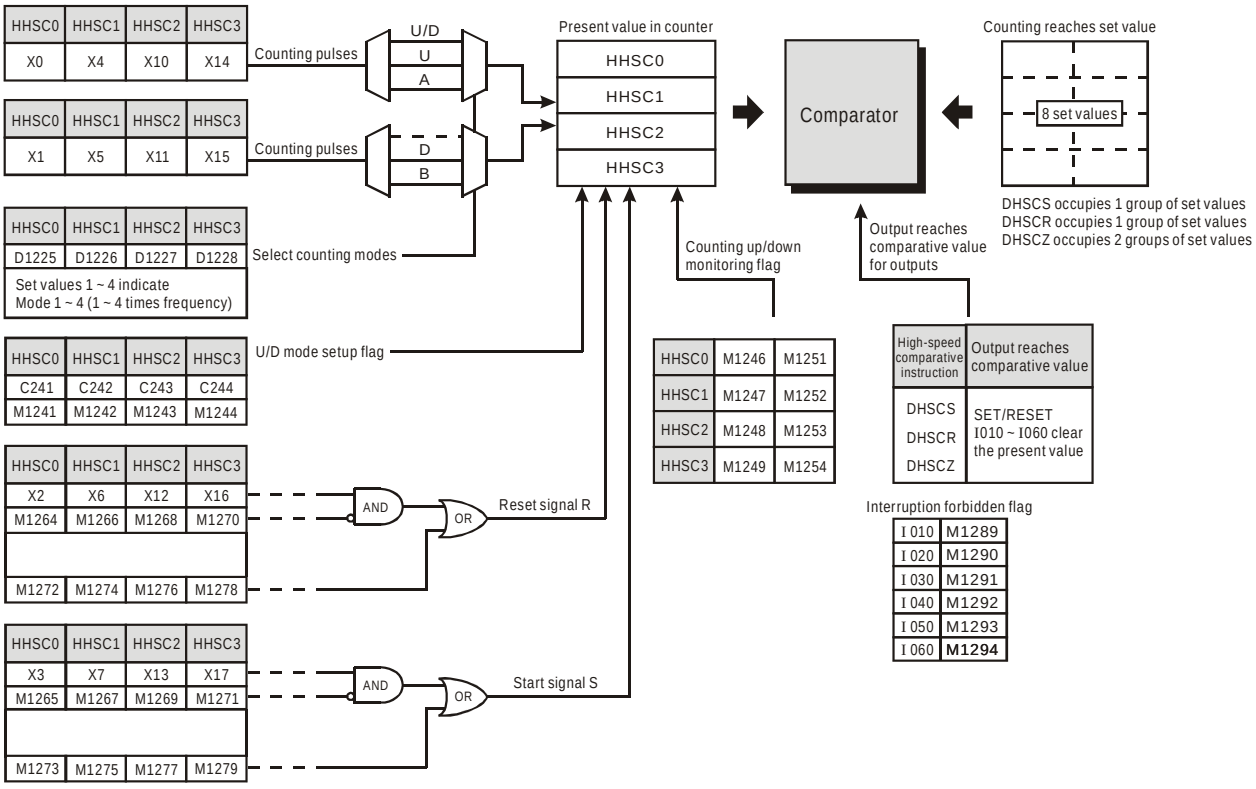
Counter type	Program-interruption high speed counter						Hardware high speed counter											
Type / Input	1-phase 1 input						1-phase 1 input				1-phase 2 inputs				2-phase 2 inputs			
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C246	C247	C248	C249	C251	C252	C253	C254
X0	U/D						U/D				U				A			
X1		U/D									D				B			
X2			U/D				R				R				R			
X3				U/D			S				S				S			
X4					U/D			U/D				U				A		
X5						U/D						D				B		
X6								R				R				R		
X7								S				S				S		
X10									U/D				U				A	
X11													D				B	
X12									R				R				R	
X13									S				S				S	
X14										U/D				U				A
X15														D				B
X16										R				R				R
X17										S				S				S

U: Progressively increasing input A: A phase input S: Input started
B: Progressively decreasing input B: B phase input R: Input cleared

- In EH3/SV2 series MPU, there is no limitation on the times of using the hardware high speed counter related instructions, DHSCS, DHSCR and DHSZ. However, when these instructions are enabled at the same time, there will be some limitations. DHSCS instruction will occupy 1 group of settings, DHSCR 1 group of settings and DHSZ 2 groups of settings. There three instructions cannot occupy 8 groups of settings in total; otherwise the system will ignore the instructions which are not the first scanned and enabled.
- The device used to set or reset a high-speed comparison in EH3/SV2 should not be a special auxiliary relay, Besides, it can not be a device used by another applied instruction. The 20 devices starting upward from the device used by the applied instruction, and the 20 devices starting downward from the device used by the applied instruction also can not be used. For example, if DHSCS uses M100 to set a

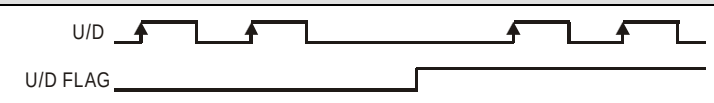
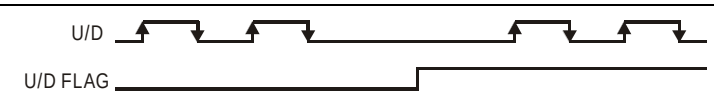
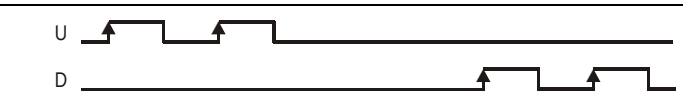
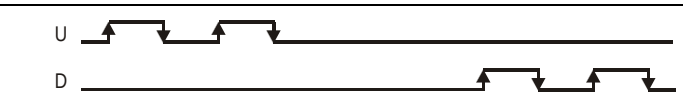
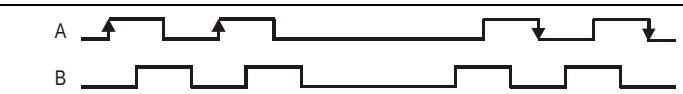
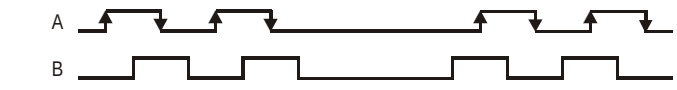
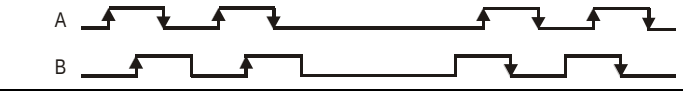
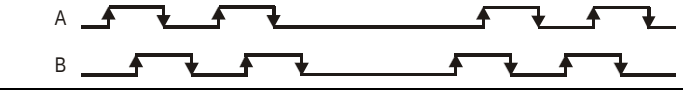
high-speed comparison, CMP should not used M80~M120 to set a comparison.

- System structure of the hardware high speed counters:
  i) HHSC0 ~ 3 have reset signals and start signals from external inputs. Settings in M1272, M1274, M1276 and M1278 are reset signals of HHSC0, HHSC1, HHSC2 and HHSC3. Settings in M1273, M1275, M1277 and M1279 are start signals of HHSC0, HHSC1, HHSC2 and HHSC3.
  ii) If the external control signal inputs of R and S are not in use, you can set M1264/M1266/M1268/M1270 and M1265/M1267/M1269/M1271 as True and disable the input signals. The corresponding external inputs can be used again as general input points (see the figure below).
  iii) When special M is used as a high speed counter, the inputs controlled by START and RESET will be affected by the scan time.



- Counting modes:

  Special D1225 ~ D1228 are for setting up different counting modes of the hardware high speed counters (HHSC0 ~ 3) in EH3/SV2 series MPU. There are normal ~ 4 times frequency for the counting and the default setting is double frequency.

Counting modes		Wave pattern
Type	Set value in special D	Counting up(+1) Counting down(-1)
1-phase 1 input	1 (Normal frequency)	
	2 (Double frequency)	
1-phase 2 inputs	1 (Normal frequency)	
	2 (Double frequency)	
2-phase 2 inputs	1 (Normal frequency)	
	2 (Double frequency)	
	3 (Triple frequency)	
	4 (4 times frequency)	

- Special registers for relevant flags and settings of high speed counters:

Flag	Function
M1150	DHSZ instruction in multiple set values comparison mode
M1151	The execution of DHSZ multiple set values comparison mode is completed.
M1152	Set DHSZ instruction as frequency control mode
M1153	DHSZ frequency control mode has been executed.
M1235 ~ M1245	Designating the counting direction of high speed counters C235 ~ C245 When M12□□ = Off, C2□□ will perform a counting up. When M12□□ = On, C2□□ will perform a counting down.
M1246 ~ M1255	Monitor the counting direction of high speed counters C246 ~ C255 When M12□□ = Off, C2□□ will perform a counting up. When M12□□ = On, C2□□ will perform a counting down.
M1260	X5 as the reset input signal of all high speed counters
M1261	High-speed comparison flag for DHSCR instruction
M1264	Disable the external control signal input point of HHSC0 reset signal point (R)

Flag	Function
M1265	Disable the external control signal input point of HHSC0 start signal point (S)
M1266	Disable the external control signal input point of HHSC1 reset signal point (R)
M1267	Disable the external control signal input point of HHSC1 start signal point (S)
M1268	Disable the external control signal input point of HHSC2 reset signal point (R)
M1269	Disable the external control signal input point of HHSC2 start signal point (S)
M1270	Disable the external control signal input point of HHSC3 reset signal point (R)
M1271	Disable the external control signal input point of HHSC3 start signal point (S)
M1272	Internal control signal input point of HHSC0 reset signal point (R)
M1273	Internal control signal input point of HHSC0 start signal point (S)
M1274	Internal control signal input point of HHSC1 reset signal point (R)
M1275	Internal control signal input point of HHSC1 start signal point (S)
M1276	Internal control signal input point of HHSC2 reset signal point (R)
M1277	Internal control signal input point of HHSC2 start signal point (S)
M1278	Internal control signal input point of HHSC3 reset signal point (R)
M1279	Internal control signal input point of HHSC3 start signal point (S)
M1289	High speed counter I010 interruption forbidden
M1290	High speed counter I020 interruption forbidden
M1291	High speed counter I030 interruption forbidden
M1292	High speed counter I040 interruption forbidden
M1293	High speed counter I050 interruption forbidden
M1294	High speed counter I060 interruption forbidden
M1312	C235 Start input point control
M1313	C236 Start input point control
M1314	C237 Start input point control
M1315	C238 Start input point control
M1316	C239 Start input point control
M1317	C240 Start input point control
M1320	C235 Reset input point control
M1321	C236 Reset input point control
M1322	C237 Reset input point control
M1323	C238 Reset input point control
M1324	C239 Reset input point control
M1325	C240 Reset input point control
M1328	Enable Start/Reset of C235
M1329	Enable Start/Reset of C236
M1330	Enable Start/Reset of C237
M1331	Enable Start/Reset of C238

Flag	Function
M1332	Enable Start/Reset of C239
M1333	Enable Start/Reset of C240

Special D	Function
D1022	Multiplied frequency of A-B phase counters for EX/EX/EC/EC3-8K/SX series MPU
D1150	Table counting register for DHSZ multiple set values comparison mode
D1151	Register for DHSZ instruction frequency control mode (counting by table)
D1152 (low word) D1153 (high word)	In frequency control mode, DHSZ reads the upper and lower limits in the table counting register D1153 and D1152.
D1225	The counting mode of the 1st group counters (C241, C246, C251)
D1226	The counting mode of the 2nd group counters (C242, C247, C252)
D1227	The counting mode of the 3rd group counters (C243, C248, C253)
D1228	The counting mode of the 4th group counters (C244, C249, C254)
D1225 ~ D1228	Counting modes of HHSC0 ~ HHSC3 in EH3/SV2 series MPU (default = 2) 1: Normal frequency counting mode 2: Double frequency counting mode 3: Triple frequency counting mode 4: 4 times frequency counting mode

API	Mnemonic		Operands	Function
54	D	HSCR	(S1) (S2) (D)	High Speed Counter Reset

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DHSCR: 13 steps
S_1					*	*	*	*	*	*	*	*	*	*		
S_2												*				
D		*	*	*								*				

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

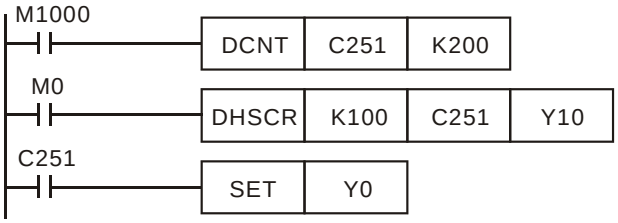
S_1: Comparative value **S_2**: No. of high speed counter **D**: Comparison result

Explanations:

1. **S_2** has to designate the No. of high speed counters C235 ~ C255. See remarks of API 53 DHSCS for more details.
2. **D** of EH3/SV2 series MPU can designate the No. of high speed counters C241 ~ C254 that are the same as the counters designated by **S_2**.
3. **D** of ES/EX/EC/EC3-8K series MPU does not support device C.
4. See the specifications of each model for their range of use.
5. Flags: M1150 ~ M1333; see remarks of API 53 DHSCS for more details. ES/EX/SS/SX series MPU does not support M1261 (high speed counter external reset mode designation); see remarks for more details.
6. The high speed counter inputs counting pulses from the corresponding external input terminals X0 ~ X17 by inserting an interruption. When the No. of high-speed counter designated in **S_2** "+1" or "-1", DHSCR will perform a comparison immediately. When the present value in the high speed counter equals the comparative value designated in **S_1**, the device designated in **D** will turn Off and even the afterward comparison results are unequal, the device will still be Off.
7. If the devices designated in **D** are Y0 ~ Y17, when the comparative value equals the present value in the high speed counter, the comparison result will immediately output to the external output terminals Y0 ~ Y17 (and clear the designated Y output) and the rest of Y devices will be affected by the scan cycle. Devices M and S act immediately without being affected by the scan cycle.

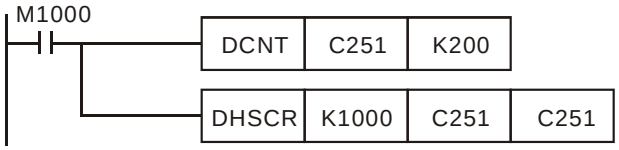
Program Example 1:

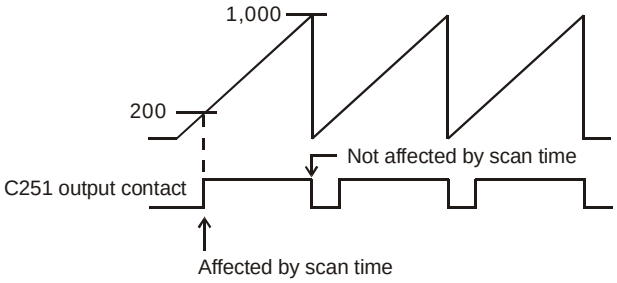
1. When M0 = On and the present value in the high speed counter C251 changes from 99 to 100 or 101 to 100, Y10 will be cleared and Off.
2. When the present value in the high speed counter C251 changes from 199 to 200, the contact of C251 will be On and make Y0 = On. However, the program scan time will delay the output.
3. Y10 will immediately reset the status when the counting reaches its target. **D** is also able to designate high speed counters of the same No. See Program Example 2.



Program Example 2:

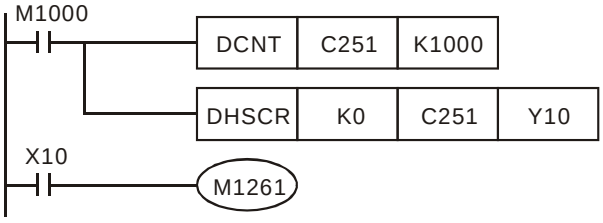
When DHSCR instruction designates the same high speed counter, and the present value in the high speed counter C251 changes from 999 to 1,000 or 1,001 to 1,000, C251 will be reset to Off.





Remarks:

1. DVP all series MPU support high speec counters. For the limitation on the use of instructions, see remarks of API 53 DHSCS for more details.
2. M1261 of EH3/SV2 series MPU designates the external reset modes of the high speed counter. Some high speec counters have input points for external reset; therefore, when the input point is On, the present value in the corresponding high speed counter will be cleared to 0 and the output contact will be Off. If you wish the reset to be executed immediately by the external output, you have to set M1261 to be On.
3. M1261 can only be used in the hardware high speed counter C241 ~ C255.
4. Example:
   a) X2 is the input point for external reset of C251.
   b) Assume Y10 = On.
   c) When M1261 = Off and X2 = On, the present value in C251 will be cleared to 0 and the contact of C251 will be Off. When DHSCR instruction is executed, there will be no counting input and the comparison result will not output. The external output will not execute the reset; therefore Y10 = On will remain unchanged.
   d) When M1261 = On and X2 = On, the present value in C251 will be cleared to 0 and the contact of C251 will be Off. When DHSCR instruction is executed, there will be no counting input but the comparison result will output. Therefore, Y10 will be reset.



API	Mnemonic		Operands	Function
55	D	HSZ	S_1 S_2 S D	High Speed Zone Compare

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DHSZ: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*		
S												*				
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

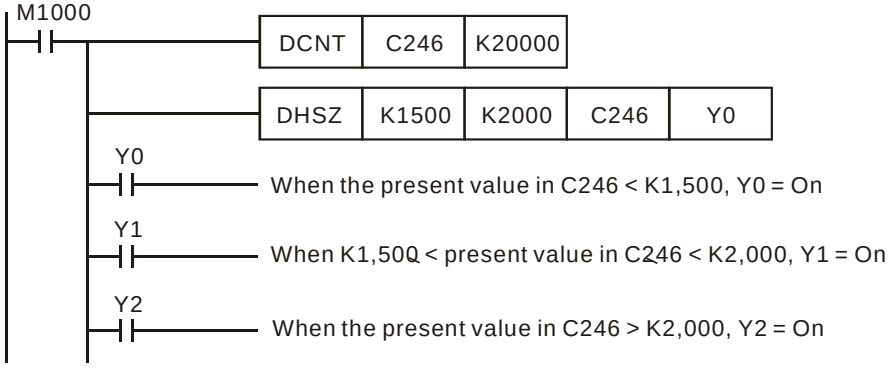
S_1: Lower bound of the comparison zone **S_2**: Upper bound of the comparison zone **S**: No. of high speed counter **D**: Comparison result

Explanations:

1. **S_1** has to be euqal to or smaller than **S_2**. (**$S_1 \leq S_2$**)
2. When **S_1** > **S_2**, the instruction will perform a comparison by using **S_1** as the upper bound and **S_2** as the lower bound.
3. **S** has to designate high speed counters C235 ~ C255, See remarks of API 53 DHSCS for more details.
4. **D** will occupy 3 consecutive devices.
5. Flags: M1150 ~ M1333; see remarks of API 53 DHSCS for more details. M1150, M1151 DHSZ executing multiple points comparison mode; see Program Example 3 for more details; SX series MPU does not support. M1152, M1153 DHSZ as frequency control mode; see Program Example 4 for more details; SX series MPU does not support.
6. The output will not be affected by the scan time.
7. The zone comparisons and outputs are all processed by inserting interruptions.
8. Please refer to the remarks on API 53 DHSCS for more information about the limitation to the use of the instruction in an EH3/SV2 series PLC.

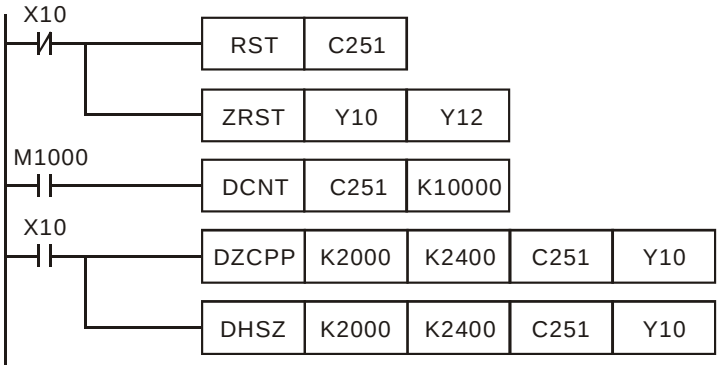
Program Example 1:

1. Designate device Y0 and Y0 ~ Y2 will be automatically occupied.
2. When DHSZ instruction is being executed and the counting of the high speed counter C246 reaches upper and lower bounds, one of Y0 ~ Y2 will be On

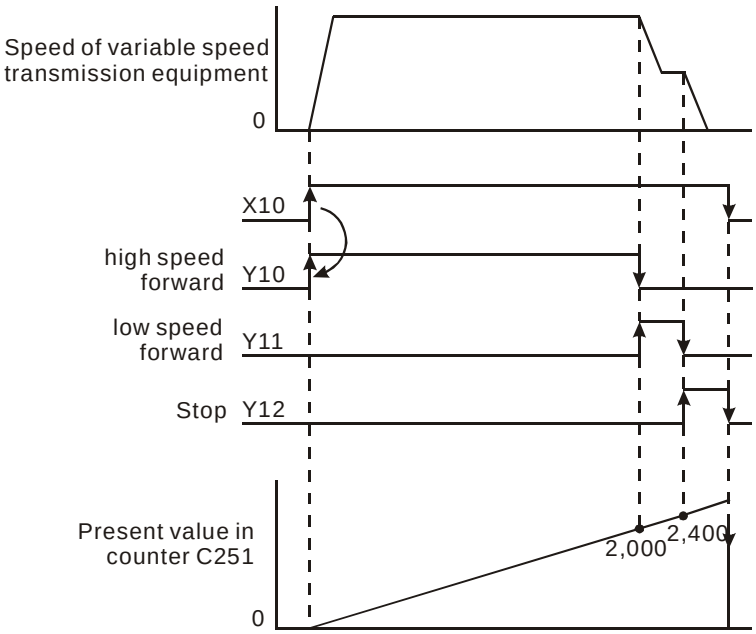


Program Example 2:

1. Use DHSZ instruction for high/low speed stop control. C251 is an A-B phase high speed counter and DHSZ only performs comparison output when there is a C251 counting pulse input. Therefore, even when the present value in the counter is 0, Y10 will not be On.
2. When X10 = On, DHSZ will require that Y10 has to be On when the present value in the counter ≤ K2,000. To solve this requirement, you can execute DZCPP instruction when the program was first RUN and compare C251 with K2,000. When the present value in the counter ≤ K2,000, Y10 will be On. DZCPP instruction is a pulse execution instruction and will only be executed once with Y10 being kept On.
3. When the drive contact X10 = Off, Y10 ~ Y12 will be reset to Off.



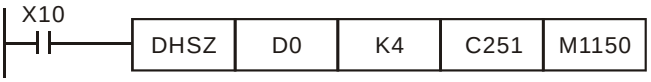
4. The timing diagram



Program Example 3:

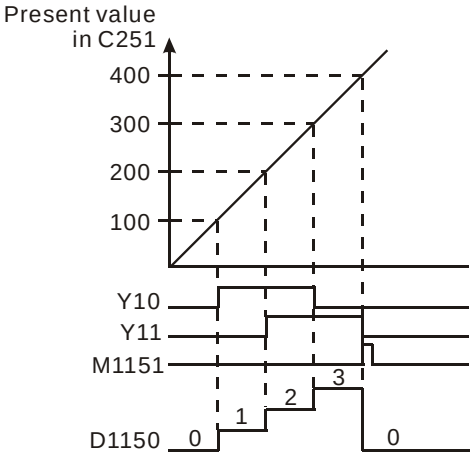
1. Program Example 3 is only applicable to EH3/SV2 series MPU.
2. The multiple set values comparison mode: If **D** of DHSZ instruction designates a special auxiliary relay M1150, the instruction will be able to compare (output) the present value in the high speed counter with many set values.

3. In this mode,
   - **S_1**: start device in the comparison table. **S_1** can only designate data register D and can be modified by E and F. Once this mode is enabled, **S_1** will not be changed even the E and F has been changed.
   - **S_2**: number of group data to be compared. **S_2** can only designate K1 ~ K255 or H1 ~ HFF and can be modified by E and F. Once this mode is enabled, **S_2** cannot be changed. If **S_2** is not within its range, error code 01EA (hex) will display and the instruction will not be executed.
   - **S**: No. of high speed counter (designated as C241 ~ C254).
   - **D**: Designated mode (can only be M1150)
4. The No. of start register designated in **S_1** and the number of rows (groups) designated in **S_2** construct a comparison table. Please enter the set values in every register in the table before executing the instruction.
5. When the present value in the counter C251 designated in **S** equals the set values in D1 and D0, the Y output designated by D2 will be reset to Off (D3 = K0) or On (D3 = K1) and be kept. Output Y will be processed as an interruption. No. of Y output pointss are in decimal (range: 0 ~ 255). If the No. falls without the range, SET/RESET will not be enabled when the comparison reaches its target.
6. When this mode is enabled, PLC will first acquire the set values in D0 and D1 as the target value for the first comparison section. At the same time, the index value displayed in D1150 will be 0, indicating that PLC performs the comparison based on the group 0 data.
7. When the group 0 data in the table have been compared, PLC will first execute the Y output set in group 0 data and determine if the comparison reaches the target number of groups. If the comparison reaches the target, M1151 will be On; if the comparison has not reached the final group, the content in D1150 will plus 1 and continue the comprison for the next group.
8. M1151 is the flag for the completion of one execution of the table, can be Off by the user. Or when the next comparion cycle takes place and the group 0 data has been compared, PLC will automatically reset the flag.
9. When the drive contact of the instruction X10 goes Off, the execution of the instruction will be interrupted and the content in D1150 (table counting register) will be reset to 0. However, the On/Off status of all outputs will be remained.
10. When the instruction is being executed, all set values in the comparison table will be regarded as valid values only when the scan arrives at END instruction for the first time.
11. This mode can only be used once in the program.
12. This mode can only be used on the hardware high speed counters C241 ~ C254.
13. When in this mode, the frequency of the input counting pulses cannot exceed 50kHz or the neighboring two groups of comparative values cannot differ by 1; otherwise there will not be enough time for the PLC to react and result in errors.



The comparison table:

32-bit data for comparison		No. of Y output	On/Off indication	Table counting register D1150
High word	Low word			
D1 (K0)	D0 (K100)	D2 (K10)	D3 (K1)	0
D5 (K0)	D4 (K200)	D6 (K11)	D7 (K1)	1
D9 (K0)	D8 (K300)	D10 (K10)	D11 (K0)	2
D13 (K0)	D12 (K400)	D14 (K11)	D15 (K0)	3
		K10: Y10 K11: Y11	K0: Off K1: On	0→1→2→3→0 Cyclic scan



14. Special registers for flags and relevant settings:

Flag	Function
M1150	DHSZ instruction in multiple set values comparison mode
M1151	The execution of DHSZ multiple set values comparison mode is completed.

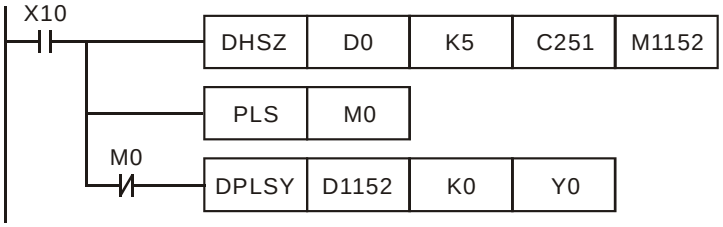
Special D	Function
D1150	Table counting register for DHSZ multiple set values comparison mode

Program Example 4:

1. Program Example 4 is only applicable to EH3/SV2 series MPU.
2. DHSZ and DPLSY instructions are combined for frequency control. If **D** of DHSZ instruction is a special auxiliary relay M1152, the present value in the counter will be able to control the pulse output frequency of DPLSY instruction.
3. In this mode,
   - **S_1**: start device in the comparison table. **S_1** can only designate data register D and can be modified by E and F. Once this mode is enabled, **S_1** will not be changed even the E and F has been changed.
   - **S_2**: number of group data to be compared. **S_2** can only designate K1 ~ K255 or H1 ~ HFF and can be modified by E and F. Once this mode is enabled, **S_2** cannot be changed. If **S_2** is not within its range, error code 01EA (hex) will display and the instruction will not be executed.
   - **S**: No. of high speed counter (designated as C241 ~ C254).

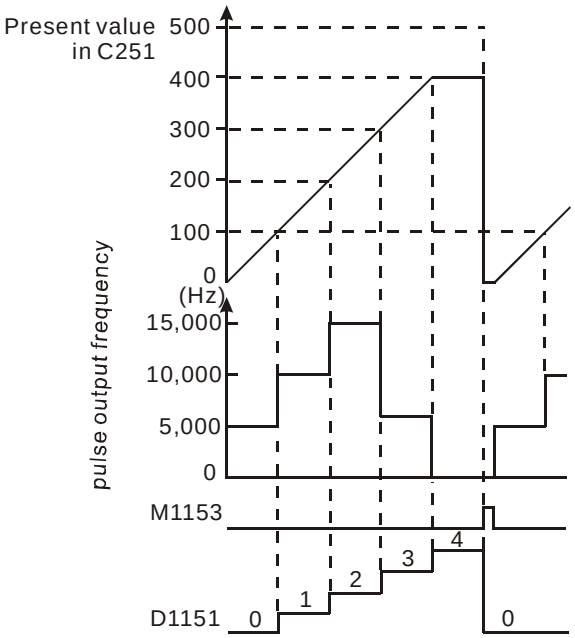
- **D**: Designated mode (can only be M1152)

4. This mode can only be used once. For EH3/SV2 series MPU, this mode can only be used in the hardware high speed counter C241 ~ C254. Please enter the set values in every register in the table before executing the instruction.
5. When this mode is enabled, PLC will first acquire the set values in D0 and D1 as the target value for the first comparison section. At the same time, the index value displayed in D1152 will be 0, indicating that PLC performs the comparison based on the group 0 data.
6. When the group 0 data in the table have been compared, PLC will first execute at the frequency set in group 0 data (D2, D3) and copy the data to D1152 and D1153, determining if the comparison reaches the target number of groups. If the comparison reaches the target, M1153 will be On; if the comparison has not reached the final group, the content in D1151 will plus 1 and continue the comprison for the next group.
7. M1153 is the flag for the completion of one execution of the table, can be Off by the user. Or when the next comparion cycle takes place and the group 0 data has been compared, PLC will automatically reset the flag.
8. If you wish to use this mode with PLSY instruction, please preset the value in D1152.
9. If you wish to stop the execution at the last row, please set the value in the last row K0.
10. When the drive contact of the instruction X10 goes Off, the execution of the instruction will be interrupted and the content in D1151 (table counting register) will be reset to 0.
11. When in this mode, the frequency of the input counting pulses cannot exceed 50kHz or the neighboring two groups of comparative values cannot differ by 1; otherwise there will not be enough time for the PLC to react and result in errors.



The comparison table:

32-bit data for comparison		Pulse output frequency 0 ~ 200kHz	Table counting register D1151
High word	Low word		
D1 (K0)	D0 (K0)	D3, D2 (K5,000)	0
D5 (K0)	D4 (K100)	D7, D6 (K10,000)	1
D9 (K0)	D8 (K200)	D11, D10 (K15,000)	2
D13 (K0)	D12 (K300)	D15, D14 (K6,000)	3
D17 (K0)	D16 (K400)	D19, D18 (K0)	4
			0→1→2→3→4 Cyclic scan

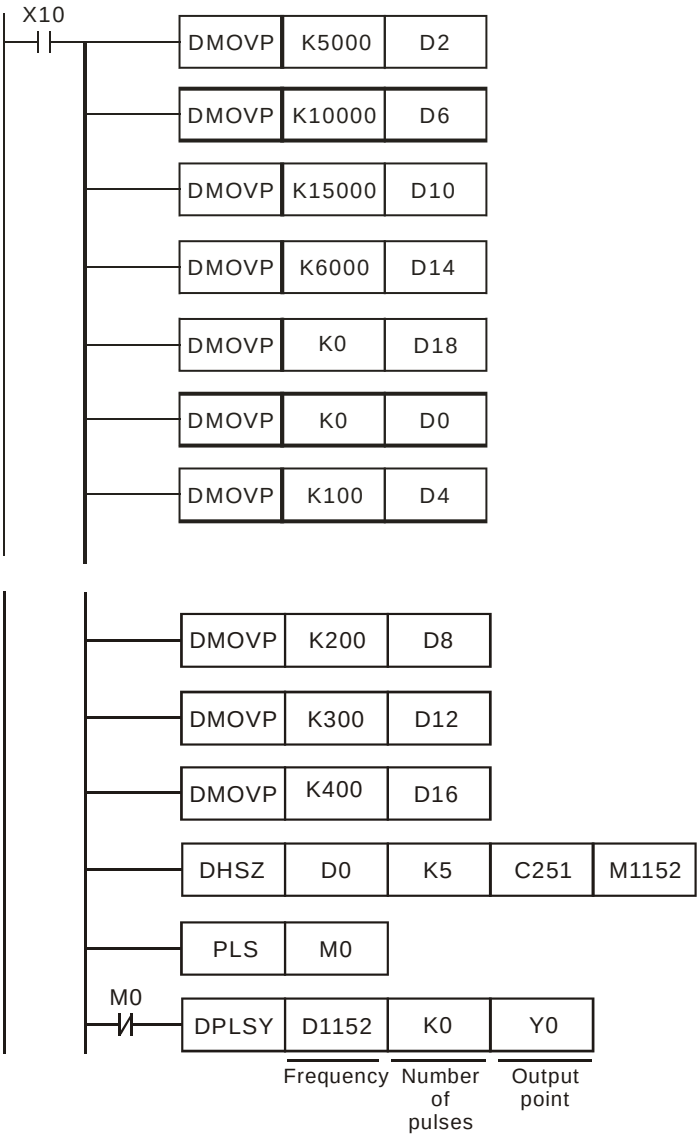


12. Special registers for flags and relevant settings:

Flag	Function
M1152	DHSZ instruction in frequency control mode
M1153	The execution of DHSZ frequency control mode is completed.

Special D	Function
D1151	Table counting register for DHSZ multiple set values comparison mode
D1152 (low word) D1153 (high word)	In frequency control mode, DHSZ reads the upper and lower limits in the table counting register D1153 and D1152.
D1336 (low word) D1337 (high word)	Current number of pulses output by DPLSY instruction

13. The complete program:



14. During the execution of DHSZ instruction, do not modify the set values in the comparison table.
15. The designated data will be arranged into the the above program diagram when the program executes to END instruction. Therefore, PLSY instruction has to be executed after DHSZ instruction has been executed once.

API	Mnemonic	Operands	Function
56	SPD	(S1) (S2) (D)	Speed Detection

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1	*															SPD: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D											*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: External pulse input terminal **S_2**: Pulse receiving time (ms) **D**: Detected result

Explanations:

1. See the specifications of each model for their range of use.
2. Flag: M1100 (SPD instruction performs sampling for one time)
3. External pulse input terminals designated in **S_1** for all series MPU:

MPU / Input	ES/EX/EC/EC3-8K	SX	EH3/SV2
Available input points	single-phase: X1, X2	double-phase: X0 / X1 single-phase: X1, X2	double-phase: X0 / X1, X10 / X11, X4 / X5, X14 / X15, single-phase: X1 ~ X3, X11 ~ X13, X5 ~ X7, X15 ~ X17

4. For SX series MPU, the X0 and X1 can be used together with A-B phase input points. When “A ahead of B” detection result is a positive value and “B ahead of A” detection result is a negative value, the multiplied frequency of the counter can be set by D1022.
5. EH3/SV2 V1.86 (and above) can only detect the speed of one input (X0/X1, X1 ~ X3). EH3/SV2 V1.88 (and above) can detect the speeds of four inputs at most. If the X0, X4, X10, or X14 is selected, the speed of an A/B-pahse input will be detected. If the input terminal slected is in the range of X1 ~ X3, X5 ~ X7, X11 ~ X13, or X15 ~ X17, the speed of a single-pahse input will be detected. For example, when selecting X0/X1 of the HHSC0 (X0~X3) as the speed detection of the single-phase inputs, the remaining inputs of HHSC0, such as X2 and X3 cannot be used to detect speed. The same rule applies to HHSC1 (X4~X7), HHSC2 (X10~X13) and HHSC3 (X14~X17).
6. The received number of pulses of the input terminal designated in **S_1** is calculated within the time (in ms) designated in **S_2**. The result is stored in the register designated in **D**.
7. **D** will occupy 5 consecutive devices. **D** + 1 and **D** are the detected value obtained from the previous pulses; **D** +3 and **D** + 2 are the current accumulated number of values; **D** + 4 is the counting time remaining (max. 32,767ms).

8. Pulse frequency detection for all series:

MPU	Max. frequency
ES/EX/EC/EC3-8K	X1 (20kHz), X2 (10kHz)
SX	X0/X1 (4kHz), X1 (20kHz), X2 (10kHz)
EH3/SV2	X0/X1 (200kHz), X1 (200kHz), X2 ~ X3 (10kHz), X4/X5 (200kHz), X5 (200kHz), X6 ~ X7 (10kHz), X10/X11 (200kHz), X11 (200kHz), X12 ~ X13 (10kHz), X14/X15 (200kHz), X15 (200kHz), X16 ~ X17 (10kHz) Note: X10/X11 in the 20 points of EH3: 10kHz

9. This instruction is mainly used for obtaining a proportional value of rotation speed. The result **D** and rotation speed will be in proportion. The following equation is for obtaining the rotation speed of motor.

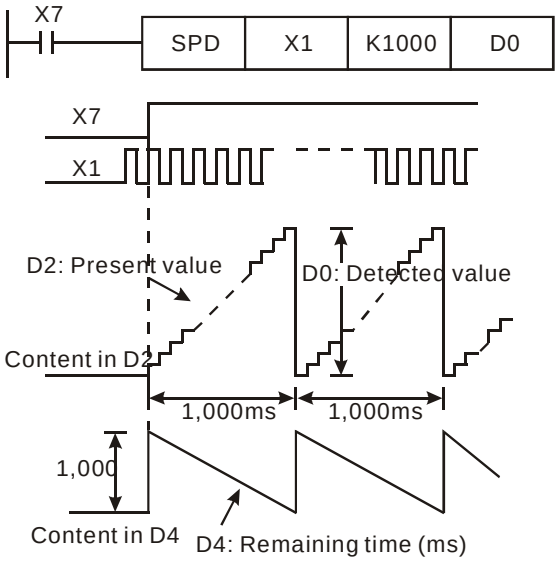
$$N=\frac{60(D0)}{nt}\times 10^{3}(rpm)$$

N: Rotation speed
n: The number of pulses produced per rotation
t: Detecting time designated in **S_2** (ms)

10. The X input point designated by this instruction cannot be used again as the pulse input terminal of the high speed counter or as an external interruption signal.
11. There is no limitation on the times of using this instruction in the program, but only one instruction will be executed at a time.
12. When SPD instruction is enabled and M1100 = On, SPD instruction will perform a sampling at the moment when M1100 goes from Off to On and stop the sampling. If you wish to resume the sampling, you have to turn Off M1100 and re-enable SPD instruction.

Program Example:

1. When X7 = On, D2 will calculate the high-speed pulses input by X1 and stop the calculation automatically after 1,000ms. The result will be stored in D0.
2. When the 1,000ms counting is completed, D2 will be cleared to 0. When X7 is On again, D2 will start the calculation again.



Remarks:

1. When ES/EX/EC/EC3-8K and SX series MPU use X1 or X2, the relevant high speed counters or external interruptions I101 and I201 cannot be used.

API	Mnemonic			Operands	Function
57	D	PLSY		(S1) (S2) (D)	Pulse Y Output

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	PLSY: 7 steps DPLSY: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Pulse output frequency **S_2**: Number of output pulses **D**: Pulse output device (please use transistor output module)

Explanations:

1. The program of ES/EX/EC/EC3-8K series MPU can use PLSY instruction two times but cannot designate the same Y device.
2. Flags: M1010 ~ M1345. See remarks for more details.
3. **S_1** designates the pulse output frequency. For SX series, you can directly use the instruciton to output pulses at 50kHz. EH3/SV2 series MPU of V1.4 and later versions use M1190 ~ M1191, and Y0 and Y2 are able to output 0.01 ~ 500Hz.

   Range of output frequency for all series:

MPU	ES/EX/EC	EC3-8K	SX	EH3/SV2
Frequency range	Y0: 0 ~10kHz Y2: 0 ~10kHz	Y0 : 0 ~20kHz Y2 : 0 ~20kHz Y1 : 0~10kHz Y3 : 0~10kHz	Y0: 0 ~ 50kHz Y1: 0 ~10kHz	Y0：0~200kHz Y2：0~200kHz Y4：0~200kHz Y6：0~200kHz Y10：0~10KHz (Note) Y12：0~10KHz (Note)

   Only SV2 series with 24 points supports Y10 and Y12.

4. **S_2** designates the number of output pulses. The 16-bit instruction can designate 1 ~ 32,767 pulses and the 32-bit instruction can designate 1 ~ 2,147,483,647 pulses.

   Number of continuous pulses for all series:

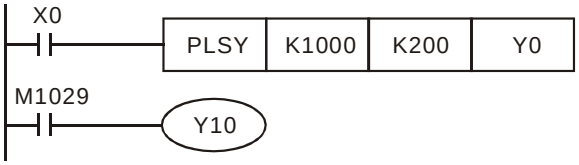
MPU	ES/EX/EC/EC3-8K/SX	EH3/SV2
How to designate continuous pulses	M1010 (Y0) On M1023 (Y1) On	The number of output pulses designated for Y0, Y2, Y4, Y6, Y10 and Y12 to be set to K0.

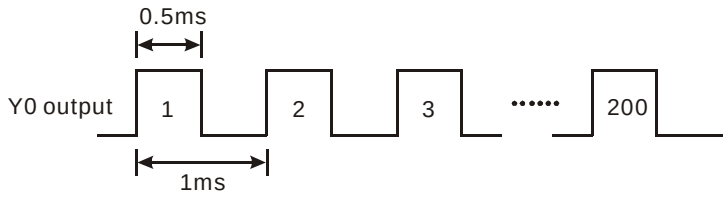
5. For EH3/SV2 series MPU, when the number of output pulses is set to 0, there will be continuous pulse output with no limitation on the number of pulses. For ES/EX/EC3-8K/SX series MPU, you have to set M1010 (Y0) or M1023 (Y1) ON to allow a continuous pulse output with no limitation on the number of pulses. For EC3-8K series, when output value in Y0~Y3 is set to 0, there is no limitation on the number of outputs.

6. For the pulse output device designated in D, EH3/SV2 series MPU can designate Y0, Y2, Y4 and Y6, 24SV2 series MPU can designate Y0, Y2, Y4, Y6, Y10 and Y12. ES/EX/EC/EC3-8Kseries MPU can designate Y0 and Y1.
7. EH3/SV2 series MPU has four groups of A-B phase pulse output from CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4, Y5) and CH3 (Y6, Y7). See 2.3 and remarks for how to set up.
8. When PLSY instruction is executed, it will designate the number of output pulses (**S_2**) output from the output device (**D**) at a pulse output frequency (**S_1**).
9. 24SV2 series MPU has 2 groups of single-phase pulse output from CH4 (Y10, Y11), CH5 (Y12, Y13).
10. When PLSY instruction is used in the program, its outputs cannot be the same as those in API 58 PWM and API 59 PLSR.
11. See remarks for more details on pulse output completed flags for all series.
12. For ES/EX/EC/EC3-8K series MPU, when PLSY and DPLSY instruction is disabled, the pulse output completed flags will all be Off automatically.
13. For EH3/SV2 series MPU, when PLSY and DPLSY instruction is disabled, the user will have to reset the pulse output completed flags.
14. The user has to reset the pulse output completed flags after the pulse output is completed.
15. After PLSY instruction starts to be executed, Y will start a pulse output. Modifying **S_2** at this moment will not affect the current output. If you wish to modify the number of output pulses, you have to first stop the execution of PLSY instruction and modify the number.
16. **S_1** can be modified when the program executes to PLSY instruction.
17. Off time : On time of the pulse output = 1 : 1.
18. When the program executes to PLSY instruction, the current number of output pulses will be stored in the special data registers D1336 ~ D1339. See remarks for more details.
19. There is no limitation on the times using this instruction. For SX series MPU, the program allows two instructions being executed on different outputs at the same time. For EC3-8K/EH3/SV2 series MPU, the program allows four instructions being executed on different outputs at the same time. For 24SV2 series MPU, the program allows six instructions being executed on different outputs at the same time.

Program Example:

1. When X0 = On, there will be 200 pulses output from Y0 at 1kHz. When the pulse output is completed, M1029 will be On and Y10 will be On.
2. When X0 = Off, the pulse output from Y0 will stop immediately. When X0 is On again, the output will start again ffrom the first pulse.





Remarks:

1. Flags and special registers for ES/EX/EC/EC3-8K series MPU:

   M1010: When On, Y0 output will be continuous with no limitation on the number of pulses. When Off, the number of output pulses from Y0 will be decided by **S_2**.

   M1023: When On, Y1 output will be continuous with no limitation on the number of pulses. When Off, the number of output pulses from Y1 will be decided by **S_2**.

   M1029: On when Y0 pulse output is completed.

   M1030: On when Y1 pulse output is completed.

   M1102: On when Y2 pulse output is completed. (for EC3-8K)

   M1103: On when Y3 pulse output is completed. (for EC3-8K)

   M1078: Y0 output pauses.

   M1079: Y1 output pauses.

   M1104： Y0 output pauses. (for EC3-8K)

   M1105： Y1 output pauses. (for EC3-8K)

   M1347： Y0 output completion auto-reset (for EC3-8K)

   M1348： Y1 output completion auto-reset (for EC3-8K)

   M1524： Y2 output completion auto-reset (for EC3-8K)

   M1525： Y3 output completion auto-reset (for EC3-8K)

   D1030: Low word of the current number of output pulses from Y0 (for EC3-8K, retainable)

   D1031: High word of the current number of output pulses from Y0

   D1032: Low word of the current number of output pulses from Y1

   D1033: High word of the current number of output pulses from Y1

   D1336： Low word of the current number of output pulses from Y2 (for EC3-8K, retainable)

   D1337： High word of the current number of output pulses from Y2

   D1338： Low word of the current number of output pulses from Y3 (for EC3-8K)

   D1339： High word of the current number of output pulses from Y3

2. Flags and special registers for SX series MPU:

   M1010: When On, Y0 output will be continuous with no limitation on the number of pulses. When Off, the number of output pulses from Y0 will be decided by **S_2**.

   M1023: When On, Y1 output will be continuous with no limitation on the number of pulses. When Off, the number of output pulses from Y1 will be decided by **S_2**.

   M1029: On when Y0 pulse output is completed.

   M1030: On when Y1 pulse output is completed.

   M1078: Y0 output pauses.

M1079: Y1 output pauses.

D1030: (SX series) Low word of the current number of output pulses from Y0

D1031: (SX series) High word of the current number of output pulses from Y0

D1032: (SX series) Low word of the current number of output pulses from Y1

D1033: (SX series) High word of the current number of output pulses from Y1

D1220 (SX V3.0 and above) Setting the phase of CH0 (Y0, Y1): Users can judge the phase of CH0 (Y0, Y1) by the last two bits in D1220. The other bits are invalid.

1. K0: Y0
2. K2: Y0 is an A-phase output, and Y1 is a B-phase output. The A-phase output is ahead of the B-phase output.

3. Flags and special registers for EH3/SV2 series MPU:

M1010: When On, CH0, CH1, CH2 and CH3 will output pulses at END instruction. Off when the output starts.

M1029: On when CH0 pulse output is completed.

M1030: On when CH1 pulse output is completed.

M1036: On when CH2 pulse output is completed.

M1037: On when CH3 pulse output is completed.

M1078 CH4 pulse output pauses. (for 24SV2 series)

M1104 CH5 pulse output pauses. (for 24SV2 series)

M1190: Able to output 0.01 ~ 500Hz when PLSY Y0 high-speed output is enabled.

M1191: Able to output 0.01 ~ 500Hz when PLSY Y2 high-speed output is enabled.

M1326 On when CH4 pulse output is completed. (for 24SV2 series)

M1327 On when CH5 pulse output is completed. (for 24SV2 series)

M1334: CH0 pulse output pauses.

M1335: CH1 pulse output pauses.

M1520: CH2 pulse output pauses.

M1521: CH3 pulse output pauses.

M1336: CH0 pulse output has been sent.

M1337: CH1 pulse output has been sent.

M1522: CH2 pulse output has been sent.

M1523: CH3 pulse output has been sent.

M1614 CH3 pulse output has been sent. (for 24SV2 series)

M1615 CH4 pulse output has been sent. (for 24SV2 series)

M1340: I110 interruption occurs after CH0 pulse output is completed.

M1341: I120 interruption after occurs CH1 pulse output is completed.

M1342: I130 interruption occurs when CH0 pulse output is sending.

M1343: I140 interruption occurs when CH1 pulse output is sending.

M1347: CH0 pulse output reset flag

M1348: CH1 pulse output reset flag

M1524: CH2 pulse output reset flag

M1525: CH3 pulse output reset flag

D1030: Low word of the current number of output pulses from CH4 (for 24SV2 series)

D1031: High word of the current number of output pulses from CH4 (for 24SV2 series)

D1032: Low word of the current number of output pulses from CH5 (for 24SV2 series)

D1033: High word of the current number of output pulses from CH5 (for 24SV2 series)

D1220: Setting the phase of CH0 (Y0, Y1): Users can judge the phase of CH0 (Y0, Y1) by the last two bits in D1220. The other bits are invalid.

1. K0: Y0
2. K1: Y0 is an A-phase output, and Y1 is a B-phase output. The A-phase output is ahead of the B-phase output.
3. K2: Y0 is an A-phase output, and Y1 is a B-phase output. The B-phase output is ahead of the B-phase output.
4. Y1

D1221: Phase setting of CH1 (Y2, Y3): D1221 determines the phase by the last two bits; other bits are invalid.

1. K0: Y2 output
2. K1: Y2, Y3 AB-phase output; A ahead of B.
3. K2: Y2, Y3 AB-phase output; B ahead of A.
4. K3: Y3 output

D1229: Phase setting of CH2 (Y4, Y5): D1229 determines the phase by the last two bits; other bits are invalid.

1. K0: Y4 output
2. K1: Y4, Y5 AB-phase output; A ahead of B.
3. K2: Y4, Y5 AB-phase output; B ahead of A.
4. K3: Y5 output

D1230: Phase setting of CH3 (Y6, Y7): D1230 determines the phase by the last two bits; other bits are invalid.

1. K0: Y6 output
2. K1: Y6, Y7 AB-phase output; A ahead of B.
3. K2: Y6, Y7 AB-phase output; B ahead of A.
4. K3: Y7 output

D1332: Low word of the number of remaining pulses at CH0

D1333: High word of the number of remaining pulses at CH0

D1334: Low word of the number of remaining pulses at CH1

D1335: High word of the number of remaining pulses at CH1

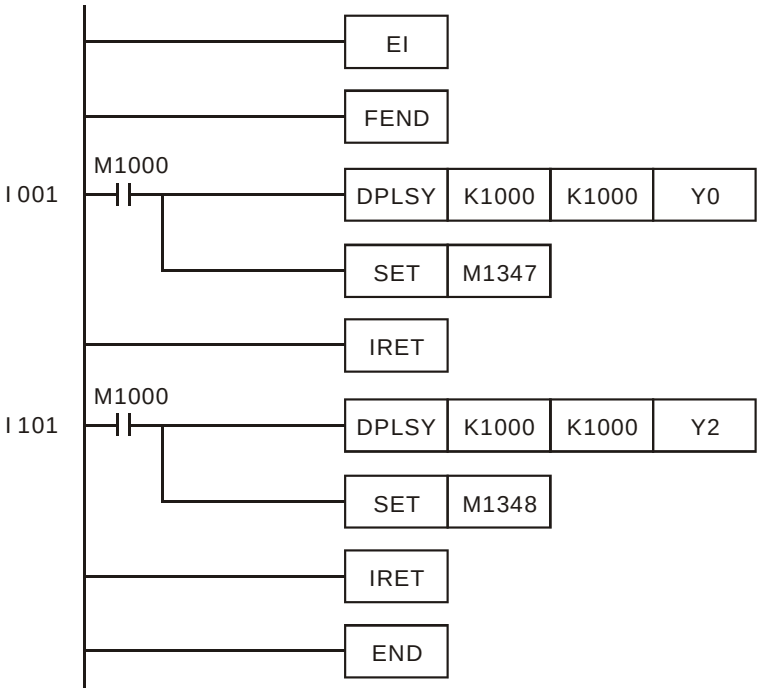
D1336: Low word of the current number of output pulses at CH0

D1337: High word of the current number of output pulses at CH0

D1338: Low word of the current number of output pulses at CH1
D1339: High word of the current number of output pulses at CH1
D1375: Low word of the current number of output pulses at CH2
D1376: High word of the current number of output pulses at CH2
D1377: Low word of the current number of output pulses at CH3
D1378: High word of the current number of output pulses at CH3

4. When there are many high speed output instructions (PLSY, PWM, PLSR) for Y0 output in a program, PLC will only execute the settings and outputs of the instruction that is first enabled.
5. More explanations on M1347, M1348, M1524 and M1525
   If the execution of PLSY instruction is complete and it will not be scanned or executed again (when it is used in an interrupt), PLC will detect the status of M1347 to see if it will be reset automatically, when END instruction is being executed. After confirming all the pulses output is complete, PLC releases the control and PLSY instruction can have control over output.

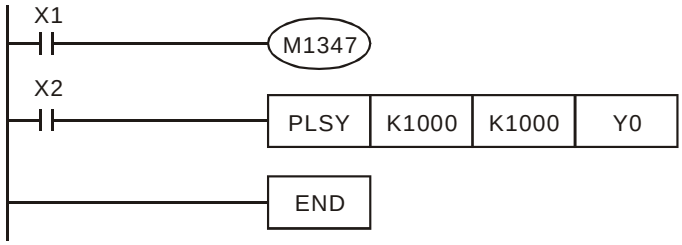
Program Example 1:



Explanations:

a) Whenever X0 is triggered, Y0 will output 1,000 pulses; whenever X1 is triggered, Y2 will output 1,000 pulses.
b) When X triggers Y pulse output, there should be an interval of at least one scan time between the end of Y pulse output and the next X-triggered output.

Program Example 2:



Explanations:

When both X1 and X2 are On, Y0 pulse output will keep operating. However, there will be a short pause (approx. 1 scan time) every 1,000 pulses before the output of the next 1,000 pulses.

API	Mnemonic	Operands	Function
58	PWM	(S1) (S2) (D)	Pulse Width Modulation

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	PWM: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Pulse output width **S_2**: Pulse output period **D**: Pulse output device (please use transistor output module)

Explanations:

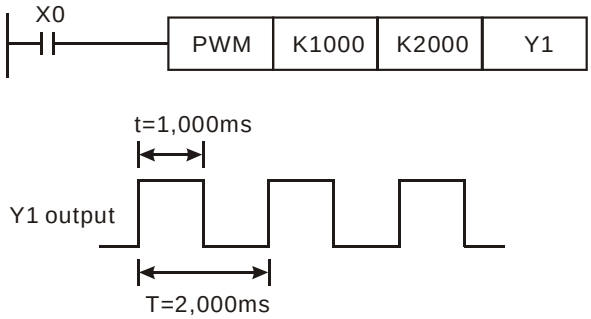
1. **$S_1 \leq S_2$**.
2. See the specifications of each model for their range of use.
3. In ES/EX/EC/EC3-8K series MPU, PWM instruction can only be used once in the program.
4. Flags: See remarks for more details.
5. Range of **S_1**: (t) 0 ~ 32,767ms. (refer to the remarks for more information about the time unit settings.)
6. Range of **S_2**: (T) 1 ~ 32,767ms (but **$S_1 \leq S_2$**).
7. **D** for all series MPU: (24SV2 does NOT support Y10 and Y12.)

MPU	ES/EX/EC	EC3-8K	SX	EH3/SV2
Output point	Y1	Y0, Y1, Y2, Y3	Y0, Y2	Y0, Y2, Y4, Y6

8. When PWM instruction is used in the program, its outputs cannot be the same as those of API 57 PLSY, API 59 PLSR or other positioning instructions.
9. PWM instruction designates the pulse output width in **S_1** and pulse output period in **S_2** and outputs from output device **D**.
10. For SX series MPU, When, **$S_1 \leq 0$** or **$S_2 \leq 0$** or **$S_1 > S_2$**, there will be operational errors (M1067 and M1068 will not be On), and there will be no output from the pulse output device. When **$S_1 = S_2$**, the pulse output device will keep being On.
11. For EC3-8K/EH3/SV2 series MPU, When, **$S_1 < 0$** or **$S_2 \leq 0$** or **$S_1 > S_2$**, there will be operational errors (M1067 and M1068 will be On), and there will be no output from the pulse output device. When **$S_1 = 0$**, M1067 and M1068 will not be On and there will be no output from the pulse output device. When **$S_1 = S_2$**, the the pulse output device will keep being On.
12. **S_1** and **S_2** can be changed when PWM instruction is being executed.
13. There is no limitation on the times using this instruction in the program. However, for SX series MPU, two instructions are allowed to be executed at different output points at the same time; for EC3-8K/EH3/SV2 series MPU, four instructions are allowed to be executed at different output points at the same time.

Program Example:

When X0 = On, Y1 will output the pulses as below. When X0 = Off, Y1 output will also be Off.



Remarks:

1. Flags for ES/EX/EC/SX series MPU:
   - M1070: Y1 pulse output time unit switch. When Off: 1ms; when On: 100us
   - D1032: Low word of the current number of output pulses from Y1
   - D1033: High word of the current number of output pulses from Y1

2. Flags for EC3-8K series MPU:
   - M1112: Y0 pulse output time unit switch. When Off: 1ms; when On: 100us
   - M1070: Y1 pulse output time unit switch. When Off: 1ms; when On: 100us
   - M1113: Y2 pulse output time unit switch. When Off: 1ms; when On: 100us
   - M1071: Y3 pulse output time unit switch. When Off: 1ms; when On: 100us
   - D1030: Low word of the current number of output pulses from Y0
   - D1031: High word of the current number of output pulses from Y0
   - D1032: Low word of the current number of output pulses from Y1
   - D1033: High word of the current number of output pulses from Y1
   - D1336: Low word of the current number of output pulses from Y2
   - D1337: High word of the current number of output pulses from Y2
   - D1338: Low word of the current number of output pulses from Y3
   - D1339: High word of the current number of output pulses from Y3

3. Flags and special registers for EH3/SV2 series MPU:

M1010:	When On, CH0, CH1, CH2 and CH3 will output pulses when END instruction is executed. Off when the output starts.
M1070:	The setting of time unit of Y0 has to work with D1371.
M1071:	The setting of time unit of Y2 has to work with D1372.
M1530:	The setting of time unit of Y4 has to work with D1373.
M1531:	The setting of time unit of Y6 has to work with D1374.
M1258:	Y0 pulse output signals reverse.
M1259:	Y2 pulse output signals reverse.

M1526:	Y4 pulse output signals reverse.
M1527:	Y6 pulse output signals reverse.
M1334:	CH0 pulse output pauses.
M1335:	CH1 pulse output pauses.
M1336:	CH0 pulse output has been sent.
M1337:	CH1 pulse output has been sent.
M1520:	CH2 pulse output pauses.
M1521:	CH3 pulse output pauses.
M1522:	CH2 pulse output has been sent.
M1523:	CH3 pulse output has been sent.
D1336:	Low word of the current number of output pulses from CH0.
D1337:	High word of the current number of output pulses from CH0.
D1338:	Low word of the current number of output pulses from CH1.
D1339:	High word of the current number of output pulses from CH1.
D1371:	Time unit of Y0 output pulses when M1070 = On.
D1372:	Time unit of Y2 output pulses when M1071 = On.
D1373:	Time unit of Y4 output pulses when M1530 = On.
D1374:	Time unit of Y6 output pulses when M1531 = On.
D1375:	Low word of the current number of output pulses from CH2.
D1376:	High word of the current number of output pulses from CH2.
D1377:	Low word of the current number of output pulses from CH3.
D1378:	High word of the current number of output pulses from CH3.

4. Time unit settings for EH3/SV2 series MPU:

   You cannot modify special M and D devices during the execution of the instruction in the program.

   D1371, D1372, D1373 and D1374 determine the time unit of the output pulses from CH0, CH1, CH2 and CH3 and the default setting is K1. If your set value is not within the range, the default value will be adopted.

D1371, D1372, D1373, D1374	K0	K1	K2	K3
Time unit	10us	100us	1ms	10ms

5. For the limitation on the times using this instruction in the program, refer to PLSY instruction for more information.

API	Mnemonic	Operands	Function
59	D PLSR	(S1) (S2) (S3) (D)	Pulse Ramp

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	PLSR: 9 steps DPLSR: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
S_3					*	*	*	*	*	*	*	*	*	*	*	
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Maximum speed of pulse output **S_2**: Total number of output pulses **S_3**: Acceleration/deceleration time (ms)

D: Pulse output device (please use transistor output module PLC)

Explanations:

1. See the specifications of each model for their range of use.
2. For ES/EX/EC series MPU, PLSR instruction can be used twice in the program but the outputs cannot be overlapped.
3. Flags: See remarks of API 57 PLSY.
4. Range of **S_1**: 10 ~ 32,767Hz (16-bit); 10 ~ 200,000Hz (32-bit). The maximum speed has to be 10's multiple; if not, the 1s digit will be left out. 1/10 of the maximum speed is the variation of one acceleration or deleration. Please be aware if the variation reponds to the acceleration/deceleration demand from the step motor, in case the step motor may crash.
5. Range of **S_2**: 110 ~ 32,767 (16-bit); 110 ~ 2,147,483,647 (32-bit). If **S_2** is less than 110, the pulet output will be abnormal.
6. Range of **S_3**: below 5,000ms. The acceleration time and deceleration time have to be the same. For DVP-ES/EX/EC series PLC, see below for more information.
   a) The acceleration/deceleration time in a DVP-ES/EX/EC series PLC has to be 10 times longer than the maximum scan time (D1012). If not, the slope of accleration and deceleration will be incorrect.
   b) The minimum set value of acceleration/deceleration time can be obtained from the following equation:

$$S_3 \geqslant \frac{90{,}000}{S_1}$$

   If the set value is less than the result obtained from the equation, the acceleration/deceleration time will be longer. If the set value is less than 90,000/**S_1**, use the result of 90,000/**S_1** as the set value.
   c) The maximum set value of acceleration/deceleration time can be obtained from the following equation:

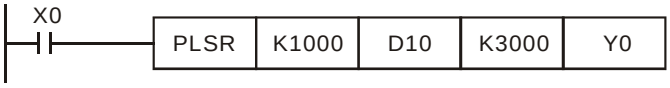
$$S_3 \leqslant \frac{S_2}{S_1} \times 818$$

   d) The speed variation is fixed to 10 steps. If the input acceleration/deceleration time is longer than the maximum set value, the acceleration/deceleration time will follow the maximum set time. If shorter than the minimum set value, the accleration/deceleration time will follow the minimum set time.

7. Refer to the related section in explanation of PLSY instruction for **D** devices and maximum frequency,.
8. EH3/SV2 series MPU has four groups pf A-B phase pulse output CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4, Y5) and CH3 (Y6, Y7). See remarks of API 57 PLSY for how to set up.
9. 24SV2 series MPU: CH4 (Y10, Y11), CH5 (Y12, Y13); Y10 and Y12 are single-phase pulse output.
10. PLSR instruction is a pulse output instruction with acclerating and decelerating functions. The pulses accelerate from the static status to target speed and decelerates when the target distance is nearly reached. The pulse output will stop when the target distance is reached.
11. For EX/EX/EC series MPU: when PLSR instruction is executed, after $\mathbf{S_1}$, $\mathbf{S_2}$ and $\mathbf{S_3}$ are set, the pulses will output from **D**. The output starts at the frequency of increasing $\mathbf{S_1}$/10 at a time. The time forf every frequency is fixed at $\mathbf{S_3}$/9.
12. $\mathbf{S_1}$, $\mathbf{S_2}$ and $\mathbf{S_3}$ can be changed when PLSR instruction is being executed.
13. For ES/EX/EC/SX series MPU, when all the Y0 pulses have been sent, M1029 will be On; when all the Y1 pulses have been sent, M1030 will be On. Next time when PLSR instruction is enabled, M1029 or M1030 will be 0 again and after the pulse output is completed, it will become 1 again.
14. For EH3/SV2 series MPU, when all the CH0 (Y0, Y1) pulses have been sent, M1029 will be On; when all the CH1 (Y2, Y3) pulses have been sent, M1030 will be On; when CH2 (Y4, Y5) pulses have been sent, M1036 will be On; when CH3 (Y6, Y7) pulses have been sent, M1037 will be On. When all the CH4 (Y10, Y11) pulses have been sent, M1326 will be On. When all the CH5 (Y12, Y13) pulses have been sent, M1327 will be On. Next time when PLSR instruction is enabled, M1029, M1030, M1036, M1037, M1326 and M1327 will be 0 again and after the pulse output is completed, they will become 1 again.
15. For EC3-8K/EH3/SV2 series MPU, when the instruction designate incorrect parameters, the default output will become the maximum value or mininum value.
16. During every acceleration section, the number of pulses (frequency × time) may not all be integers. PLC will round up the number to an integer before the output. Therefore, the acceleration time of every section may not be exactly the same. The offset is determined upon the frequency and the decimal after rounding up. In order to ensure the correct number of output pulses, PLC will supplement insufficient pulses in the last section.
17. For the limitation on the times using this instruction in the program, refer to PLSY instruction for more information.

Acceleration/deceleration in EC3-8K/SX/EH3/SV2 series MPU:

1. Example:

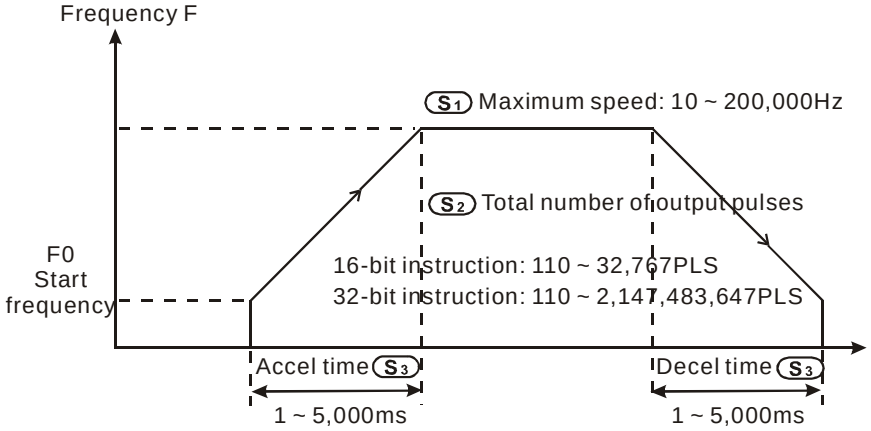


2. The range of pulse speed for this instruction is 10 ~ 200,000Hz. If the set values of maximum speed and acceleration/deceleration time exceed the range, PLC will operate by the default value that is within the range.

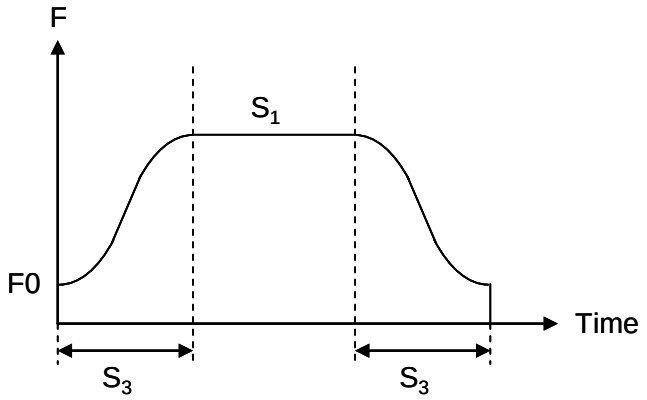
Operand		S_1	S_2	S_3	D
Explanation		Max. frequency	Total number of pulses	Accel/Decel time	Output point
Range	16-bit	10 ~ 32,767Hz	110 ~ 32,767	1 ~ 5,000ms	Refer to PLSY instruction for more information.
	32-bit	10 ~ 200kHz	110 ~ 2,147,483,647		
Definition		K0: No output Kn: Designated frequency	Kn: Designated number	Flag: M1067, M1068	See settings of D1220, D1221

Refer to PLSY instruction for more information on the usage of special M and D devices.

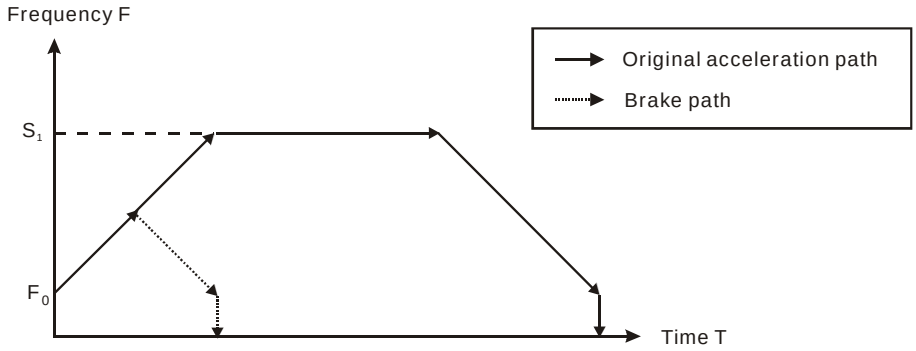
When M1257 is Off, the acceleration/deceleration slope is a straight-line curve, as shown below.



When M1257 is On, the acceleration/deceleration slope is an S curve, as shown below.



3. The acceleration/deceleration of EC3-8K/EH3/SV2 series MPU is based on the number of pulses. If the output cannot reach the maximum acceleration frequency within the acceleration/deceleration time offered, the instruction will automatically adjust the acceleration/deceleration time and the maximum frequency.
4. The operands have to be set before the execution of the instruction PLSR. You cannot change the acceleration/deceleration during the instruciton execution.
5. All acceleration/deceleration instructions are included with the brake function. The brake function will be enabled when PLC is performing acceleration and the switch contact is suddenly Off. The deceleration will operate at the slope of the acceleration.



API	Mnemonic	Operands	Function
60	IST	(S) (D1) (D2)	Initial State

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	IST: 7 steps
S	*	*	*													
D_1				*												
D_2				*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start device in the designated operation mode **D_1**: The smallest No. of designated step in auto mode

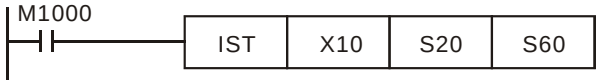
D_2: The biggest No. of designated step in auto mode

Explanations:

1. **S** will occupy 8 consecutive points.
2. Range of **D_1** and **D_2**: for SX/EH3 /SV2: S20 ~ S899; for ES/EX/EC/EC3-8K: S20 ~ S127; **D_2** > **D_1**.
3. See the specifications of each model for their range of use.
4. ES/SX/EC3-8K series MPU does not support E, F index register modification.
5. IST instruction can only be used once in the program.
6. Flags: M1040 ~ M1047. See remarks for more details.
7. IST instruction is a handy instruction specifically for the initial status of step ladder control procedure to accommodate special auxiliary relay.

Program Example 1:

1. Use of IST instruction

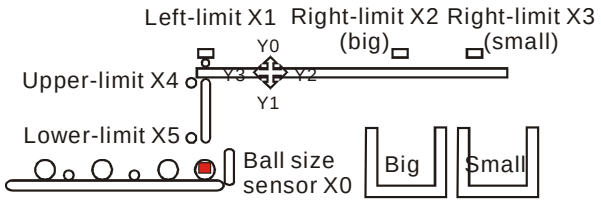


S X10: Individual operation
X11: Zero return
X12: Step operation
X13: One cycle operation
X14: Continuous operation
X15: Zero return enabled switch
X16: Start switch
X17: Stop switch

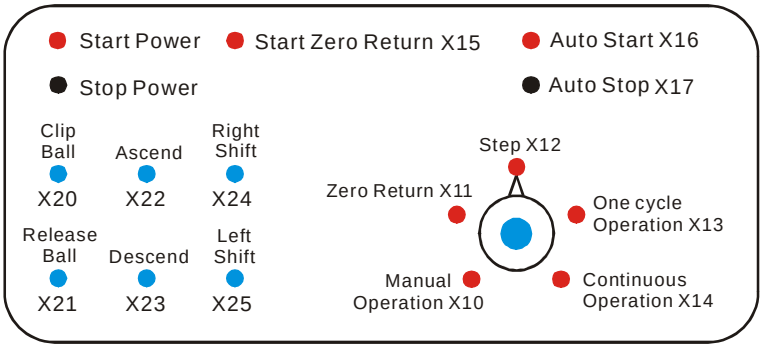
2. When IST instruction is being executed, the following special auxiliary relays will switch automatically.
   M1040: Operation forbidden
   M1041: Operation starts
   M1042: Pulse output enabled
   M1047: STL monitor enabled
   S0: Initiates manual operation
   S1: Initiates zero return
   S2: Initiates auto operation
3. S10 ~ S19 are for zero return and cannot be used as general steps. When S0 ~ S9 are in use, S0 ~ S2 represent manual operation mode, zero return mode and auto operation mode. Therefore, in the program, you have to write the circuit of the three steps in advance.
4. When switched to S1 (zero return) mode, any On in S10 ~ S19 will result in no zero return.
5. When switched to S2 (auto operation) mode, any On of the S in **D_1** ~ **D_2** or M1043 = On will result in no auto operation.

Program Example 2:

1. Robot arm control (by IST instruction):
   a) Motion request: Separate the big ball and small ball and move them to different boxes. Configure the control panel for the control.
   b) Motions of the robot arm: descending, clipping ball, ascending, right shifting, releasing ball, ascending, left shifting.
   c) I/O devices:

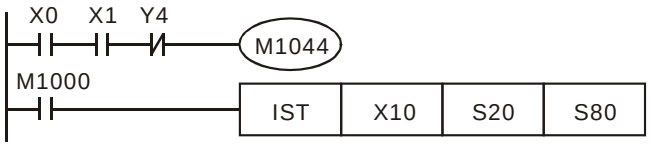


2. Operation modes:
   Manual operation: Turn On/Off of the load by a single button.
   Zero return: Press the zero return button to automatically zero-return the machine.
   Auto operation:
   a) Single step operation: Press “auto start” button for every one step forward.
   b) One cycle operation: Press “auto start” button at the zero point. After a cycle of auto operation, the operation will stops at the zero point. Press “auto stop” button in the middle of the operation to stop the operation and press “auto start” to restart the operation. The operation will resume until it meets the zero point.
   c) Continuous operation: Press “auto start” button at the zero point to resume the operation. Press “auto stop” to operate until it meets the zero point.
3. The control panel:

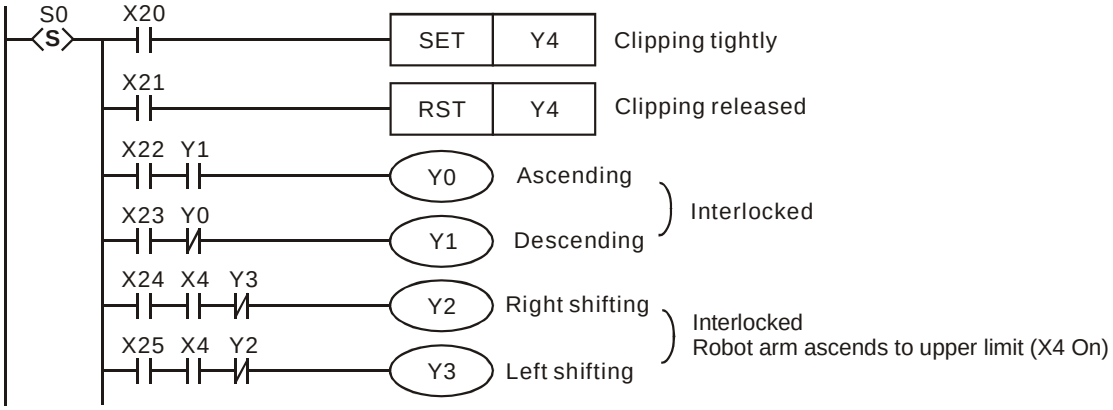


   a) Ball size sensor X0.
   b) Robot arm: left limit X1, big ball right limit X2, small ball right limit X3, upper limit X4, lower limit X5.
   c) Robot arm: ascending Y0, descending Y1, right shifting Y2, left shifting Y3, clipping Y4.

Start Circuit

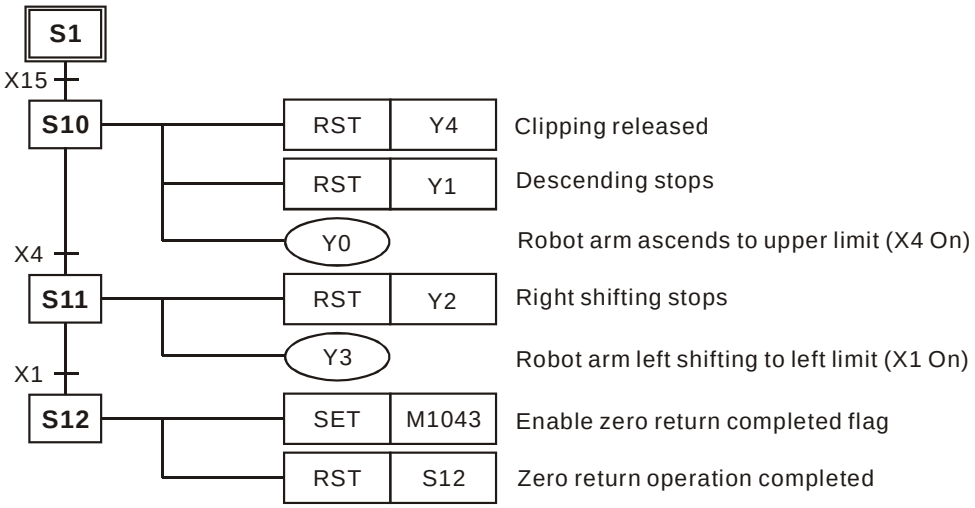


Manual Operation Mode

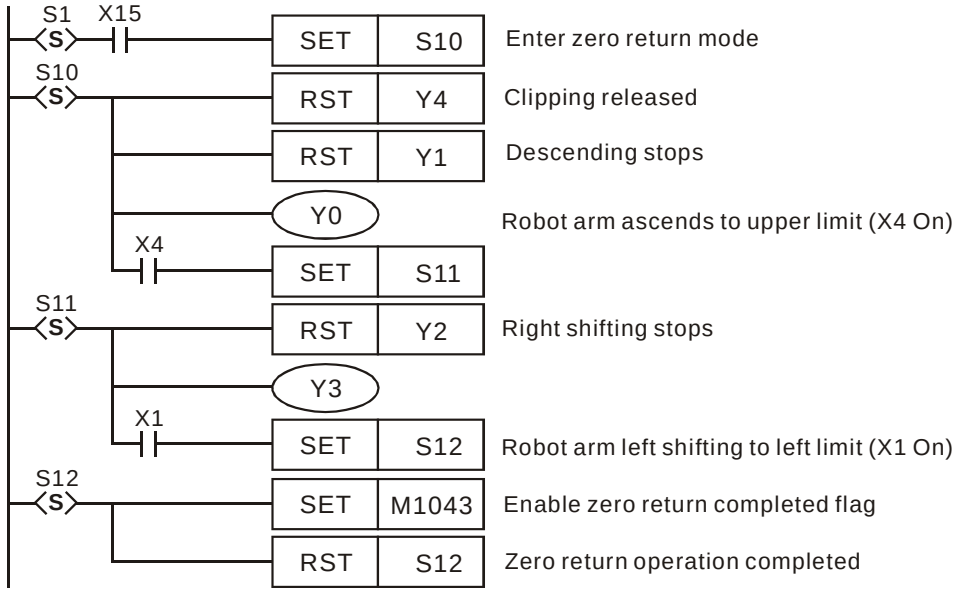


Zero Return Mode

SFC:

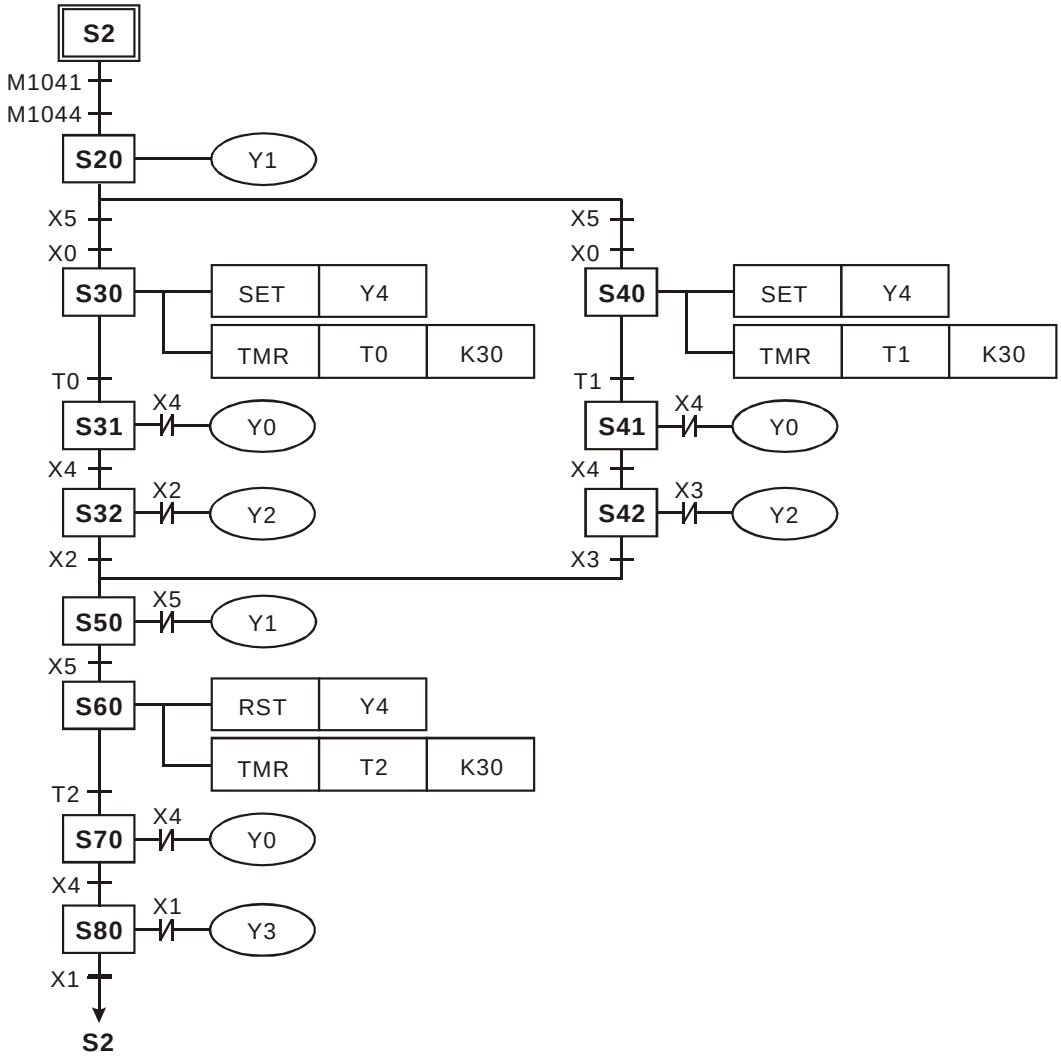


Ladder Diagram:

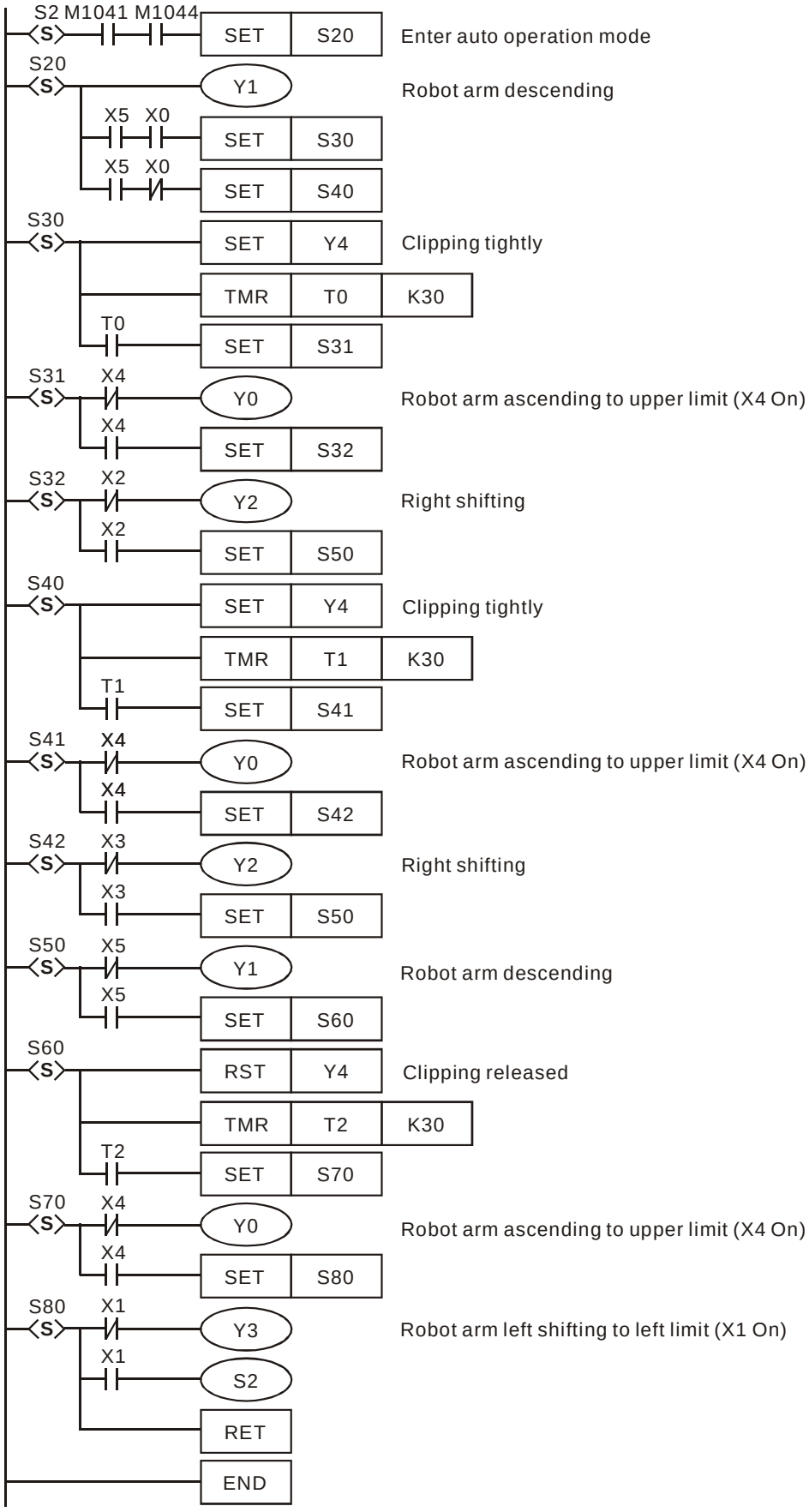


Auto Operation Modes

SFC:



Ladder Diagram:



Remarks:

Flag explanations:

M1040: When On, all step operations are forbidden.

1. Manual mode: M1040 keeps being On
2. Zero return/one cycle operation mode: Between the timing of pressing “auto stop” and “auto start” buttons, M1040 will keep being On.
3. Step mode: M1040 keeps being On until “auto start” button is pressed.
4. Continuous operation mode: When PLC goes from STOP to RUN, M1040 will keep being On and turn Off when “auto start” button is pressed.

M1041: Step operation starts. Special M for initial S2 to move to the next step.

1. Manual/zero return mode: M1041 keeps being Off.
2. Step/one cycle operation mode: M1041 will only be On when “auto start” button is pressed.
3. Continuous operation mode: M1041 keeps On when “auto start” button is pressed; Off when “auto stop” button is pressed.

M1042: Enabling pulse output. Sending pulses once when “auto start” button is pressed.

M1043: On when zero return is completed.

M1044: In continuous operation mode, M1044 has to be On to more S2 to the next step.

M1045: All output resets are forbidden.

If the machine (not at the zero point) goes

- from manual (S0) to zero return (S1)
- from auto (S2) to manual (S0)
- from auto (S2) to zero return (S1)

1. When M1045 is Off, and any of the S among $\mathbf{D_1}$ ~ $\mathbf{D_2}$ is On, SET Y output and the step in action will be reset to Off.
2. When M1045 is On, SET Y output will be remained but the step in action will be reset to Off

If the machine executes zero return (at the zero point) and goes from zero return (S1) to manual (S0), no matter M1045 is On or Off, SET Y output will be remained but the step in action will be reset to Off.

M1046: STL state setting. On when any of the steps is On. When M1047 is forced On, On of any S will result in On of M1046. D1040 ~ D1047 will record the No. of the previous 8 points before On of S.

M1047: On for enabling STL monitor. When IST instruction starts to be executed, M1047 will be forced On. In every scan time, as long as IST instruction is still On, M1047 will be forced On. M1047 monitors all the S.

D1040 ~ D1047: On status of step No. 1 ~ 8

API	Mnemonic			Operands	Function
61	D	SER	P	S_1 S_2 D n	Search a Data Stack

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			SER, SERP: 9 steps DSER, DSERP: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

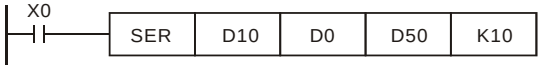
S_1: Start device for data stack comparison **S_2**: Data to be compared **D**: Start device for storing comparison result **n**: Length of data to be compared

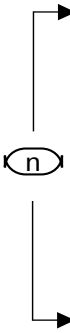
Explanations:

1. When **S_2** are used in device F, only 16-bit instruction is applicable.
2. **D** will occupy 5 consecutive points.
3. Range of **n**: for 16-bit instruction 1 ~ 256; for 32-bit instruction 1 ~ 128.
4. See the specifications of each model for their range of use.
5. The **n** data in the registers starting from **S_1** are compared with **S_2** and the results are stored in the registers starting from **D**.
6. In the 32-bit instruction, **S_1**, **S_2**, **D** and **n** will designate 32-bit registers.
7. For **D**, the 16-bit counters and 32-bit counters cannot be mixed when being used.

Program Example:

1. When X0 = On, the data stack consist of D10 ~ D19 will be compared against D0 and the result will be stored in D50 ~ D52. If there are equivalent values appearing during the comparison, D50 ~ D52 will all be 0.
2. The data are compared algebraically. (-10 < 2).
3. The No. of the register with the smallest value among the compared data will be recorded in D53; the biggest will be recorded in D54. When there are more than one smallest value or biggest value, device D will record the No. of the register with bigger value.





S_1	Content	Data to be compared	Data No.	Result
D10	88	S_2	0	
D11	100		1	Equal
D12	110		2	
D13	150		3	
D14	100	D0 = K100	4	Equal
D15	300		5	
D16	100		6	Equal
D17	5		7	Smallest
D18	100		8	Equal
D19	500		9	Biggest

D	Content	Description
D50	4	Total number of data with equivalent values
D51	1	No. of the first equivalent value
D52	8	No. of the last equivalent value
D53	7	No. of the smallest value
D54	9	No. of the biggest value

API	Mnemonic		Operands	Function
62	D	ABSD	(S1) (S2) (D) (n)	Absolute Drum Sequencer

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			ABSD: 9 steps DABSD: 17 steps
S_2											*	*	*			
D		*	*	*												
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Start device in the data table **S_2**: No. of counter **D**: Start No. of the devices for the comparison results

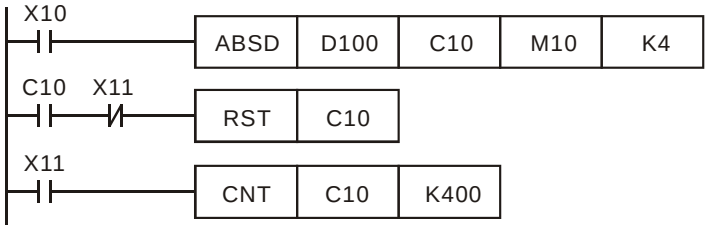
n: Number of data for comparison

Explanations:

1. When **S_1** designates KnX, KnY, KnM and KnS, the 16-bit instruction has to designate K4 and 32-bit instruction has to designate K8.
2. For SX series MPU, **S_2** only supports C device.
3. Range of **n**: 1 ~ 64
4. See the specifications of each model for their range of use.
5. ABSD instruction is for the absolute control of the multiple output pulses generated by the present value in the counter.
6. **S_2** of DABSD instruction can designate high speed counters. However, when the present value in the high speed counter is compared with the target value, the result cannot output immediately owing to the scan time. If an immediate output is required, please use DHSZ instruction that is exclusively for high speed counters.

Program Example:

1. Before the execution of ABSD instruction, use MOV instruction to write all the set values into D100 ~ D107 in advance. The even-number D is for lower bound value and the odd-number D is for upper bound value.
2. When X10 = On, the present value in counter C10 will be compared with the four groups of lower and upper bound values in D100 ~ D107. The comprison results will be stored in M10 ~ M13.
3. When X10 = Off, the original On/Off status of M10 ~ M13 will be remained.

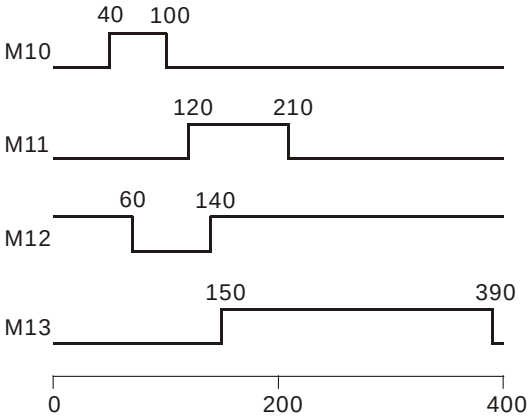


4. M10~ M13 will be On when the present value in C10 ≦ upper bound value or ≧ lower bound value.

Lower bound value	Upper bound value	Present value in C10	Output
D100 = 40	D101 = 100	40 ≦ C10 ≦ 100	M10 = On
D102 = 120	D103 = 210	120 ≦ C10 ≦ 210	M11 = On
Lower bound value	Upper bound value	Present value in C10	Output
D104 = 140	D105 = 170	140 ≦ C10 ≦ 170	M12 = On
D106 = 150	D107 = 390	150 ≦ C10 ≦ 390	M13 = On

5. If the lower bound value > upper bound value, when C10 < upper bound value (60) or > upper bound value (140), M12 will be On.

Lower bound value	Upper bound value	Present value in C10	Output
D100 = 40	D101 = 100	40 ≦ C10 ≦ 100	M10 = On
D102 = 120	D103 = 210	120 ≦ C10 ≦ 210	M11 = On
D104 = 140	D105 = 60	60 ≦ C10 ≦ 140	M12 = On
D106 = 150	D107 = 390	150 ≦ C10 ≦ 390	M13 = On



API	Mnemonic	Operands	Function
63	INCD	S_1 S_2 D n	Incremental Drum Sequencer

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			INCD: 9 steps
S_2												*				
D		*	*	*												
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Start device in the data table **S_2**: No. of counter **D**: Start No. of the devices for the comparison results

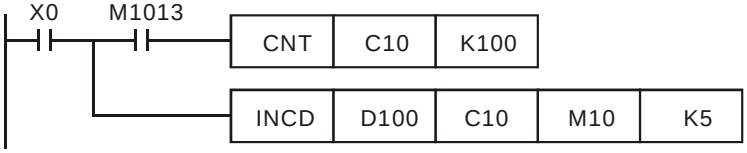
n: Number of data for comparison

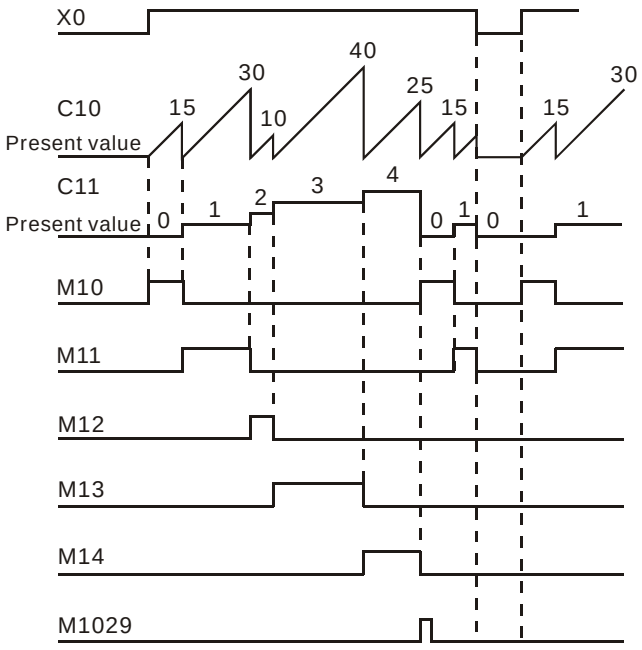
Explanations:

1. When **S_1** designates KnX, KnY, KnM and KnS, it has to designate K4.
2. In the 16-bit instruction, **S_2** has to designate C0 ~ C198 and will occupy 2 consecutive No. of counters.
3. Range of **n**: 1 ~ 64
4. See the specifications of each model for their range of use.
5. Flag: M1029 (instruciton execution completed)
6. INCD instruction is for the relative control of the multiple output pulses generated by the present value in the counter.
7. The present value in **S_2** is compared with **S_1**. **S_2** will be reset to 0 whenever a comparison is completed. The current number of data processed in temporarily stored in **S_2** + 1.
8. When n data have been processed, M1029 will be On for one scan period.

Program Example:

1. Before the execution of INCD instruction, use MOV instruction to write all the set values into D100 ~ D104 in advance. D100 = 15, D101 = 30, D102 = 10, D103 = 40, D104 = 25.
2. The present value in C10 is compared against the set values in D100 ~ D104. The present value will be reset to 0 whenever a comparison is completed.
3. The current number of data having been processed is temporarily stored in C11.
4. The number of times of reset is temporarily stored in C11.
5. Whenever the content in C11 pluses 1, M10 ~ M14 will also correspondingly change. See the timing diagram below.
6. After the 5 groups of data have been compared, M1029 will be On for one scan period.
7. When X0 goes from On to Off, C10 and C11 will both be reset to 0 and M10 ~ M14 will all be Off. When X0 is On again, the instruction will start its execution again from the beginning.




X0
C10
Present value
15
30
10
40
25
15
15
30
C11
Present value
0
1
2
3
4
0
1
0
1
M10
M11
M12
M13
M14
M1029

API	Mnemonic	Operands	Function
64	TTMR	(D) (n)	Teaching Timer

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D													*			TTMR: 5 steps
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

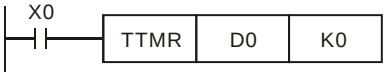
D: Device No. for storing the “On” time of button switch **n**: Multiple setting

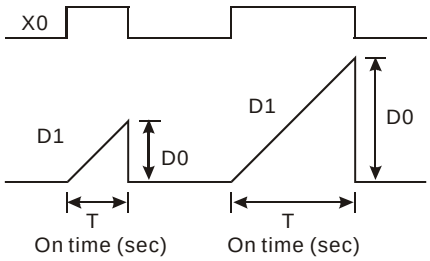
Explanations:

1. **D** will occupy 2 consecutive devices.
2. Range of **n**: 0 ~ 2
3. See the specifications of each model for their range of use.
4. TTMR instruction can be used 8 times in the program.
5. The “On” time (unit: 100ms) of the external button switch is stored in device No. **D** + 1. The “On” time (unit: second) of the switch is multiplied by **n** and stored in **D**.
6. Multiple setting:
   When n = 0, unit of **D** = second
   When n = 1, unit of **D** = 100ms (D × 10)
   When n = 2, unit of **D** = 10ms (D × 100)

Program Example 1:

1. The “On” (being pressed) time of button switch X0 is stored in D1. The setting of n is stored in D0. Therefore, the button switch will be able to adjust the set value in the timer.
2. When X0 goes Off, the content in D1 will be cleared to 0, but the content in D0 will remain.



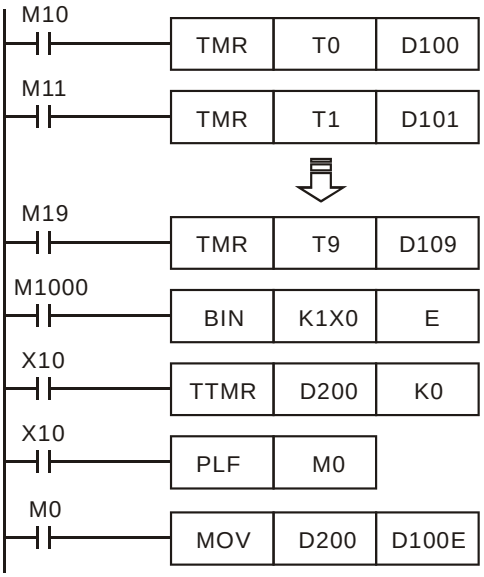


3. Assume the “On” time of X0 is T (sec.), see the relation between D0, D1 and n in the table below.

n	D0	D1 (unit: 100ms)
K0 (unit: s)	1 × T	D1 = D0 × 10
K1 (unit: 100 ms)	10 × T	D1 = D0
K2 (unit: 10 ms)	100 × T	D1 = D0/10

Program Example 2:

1. Use TMR instruction to write in 10 groups of set time.
2. Write the set values into D100 ~ D109 in advance.
3. The timing unit for timer T0 ~ T9 is 0.1 sec. The timing unit for the teaching timer is 1 sec.
4. Connect the 1-bit DIP switch to X0 ~ X3 and use BIN instruction to convert the set value of the switch into a bin value and store it in E.
5. Store the “On” time (sec.) of X10 in D200.
6. M0 refers to the pulses generated from one scan period after the button switch of the teaching timer X10 is released.
7. Use the set number of the DIP switch as the indirectly designated pointer and send the content in D200 to D100E (D100 ~ D109).



Remarks:

1. There is no limitation on the times using this instruction in the program and 8 instructions can be executed at the same time.

API	Mnemonic	Operands	Function
65	STMR	(S) (m) (D)	Special Timer

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*					STMR: 7 steps
m					*	*										
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

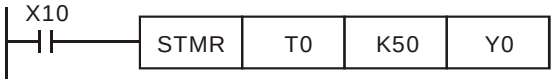
S: No. of timer **m**: Set value in timer (unit: 100ms) **D**: No. of start output device

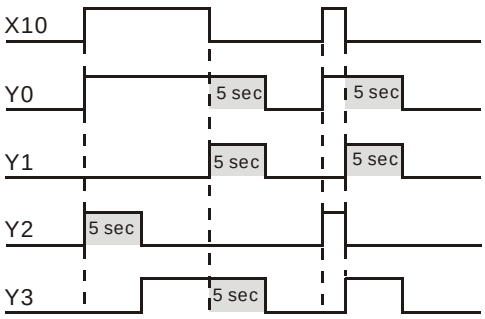
Explanations:

1. Range of **S**: for SX T0 ~ T191; for EH/EH2/SV T0 ~ T199; for EH3/SV2 T0 ~ T183
2. Range of **m**: 1 ~ 32,767
3. **D** will occupy 4 consecutive devices.
4. See the specifications of each model for their range of use.
5. STMR instruction is used for Off-delay, one shot timer and flashing sequence.
6. The No. of timers designated by STMR instructions can be used only once.

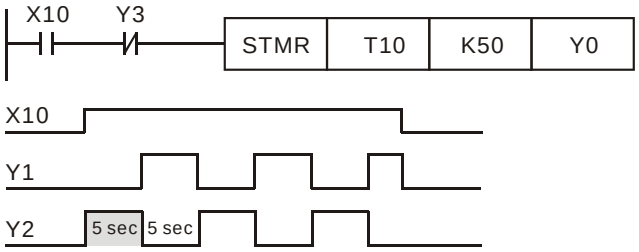
Program Example:

1. When X10 = On, STMR instruction will designate timer T0 and set the set value in T0 as 5 seconds.
2. Y0 is the contact of Off-delay. When X10 goes from Off to On, Y0 will be On. When X10 goes from On to Off, Y0 will be Off after a five seconds of delay.
3. When X10 goes from On to Off, there will be a five seconds of Y1 = On output.
4. When X10 goes from Off to On, there will be a five seconds of Y2 = On output.
5. When X10 goes from Off to On, Y3 will be On after a five seconds of delay. When X10 goes from On to Off, Y3 will be Off after a five seconds of delay.





6. Add a b contact of Y3 after X10, and Y1 and Y2 can operate for flashing sequence output. When X10 goes Off, Y0, Y1 and Y3 will be Off and the content in T10 will be reset to 0.



API	Mnemonic			Operands	Function
66		ALT	P	(D)	Alternate State

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D		*	*	*												ALT, ALTP: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

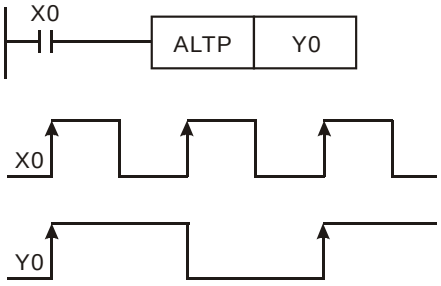
D: Destination device

Explanations:

1. See the specifications of each model for their range of use.
2. When ALT instruction is executed, “On” and “Off” of **D** will switch.
3. This instruction adopts pulse execution instructions (ATLP).

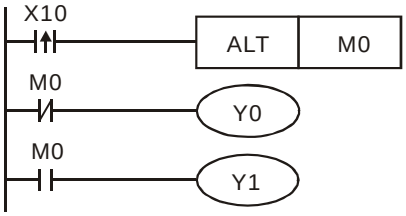
Program Example 1:

When X0 goes from Off to On, Y0 will be On. When X0 goes from Off to On for the second time, Y0 will be Off.



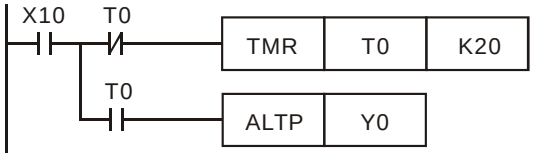
Program Example 2:

Using a single switch to enable and disable control. At the beginning, M0 = Off, so Y0 = On and Y1 = Off. When X10 switches between On/Off for the first time, M0 will be On, so Y1 = On and Y0 = Off. For the second time of On/Off switching, M0 will be Off, so Y0 = On and Y1 = Off.



Program Example 3:

Generate flashing. When X10 = On, T0 will generate a pulse every 2 seconds and Y0 output will switch between On and Off following the T0 pulses.



API	Mnemonic	Operands	Function
67	RAMP	S1 S2 D n	Ramp Variable Value

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			RAMP: 9 steps
S_2													*			DRAMP: 17 steps
D													*			
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

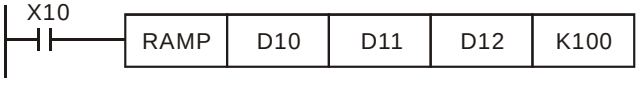
Operands:

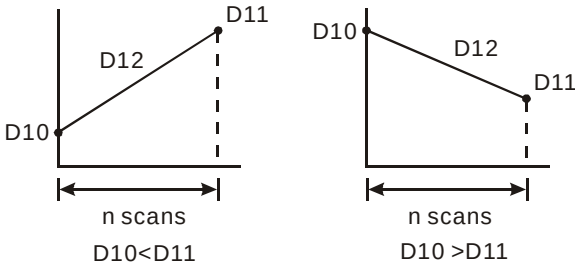
S_1: Start of ramp signal **S_2**: End of ramp signal **D**: Duration of ramp signal **n**: Scan times

Explanations:

1. Range of **n**: 1 ~ 32,767
2. **D** will occupy 2 consecutive points.
3. See the specifications of each model for their range of use.
4. Flags: M1026 (enabling RAMP; see remarks for more details); M1029 (RAMP execution completed).
5. This instruction is for obtaining slope (the relation between linearity and scan time). Before using this instruction, you have to preset the scan time.
6. The set value of start ramp signal is pre-written in D10 and set value of end ramp signal in D11. When X10 = On, D10 increases towards D11 through n (= 100) scans (the duration is stored in D12). The times of scans are stored in D13.
7. In the program, first drive M1039 = On to fix the scan time. Use MOV instruction to write the fixed scan time to the special data register D1039. Assume the scan time is 30ms and take the above program for example, n = K100, the time for D10 to increase to D11 will be 3 seconds (30ms × 100).
8. When X10 goes Off, the instruction will stop its execution. When X10 goes On again, the content in D12 will be reset to 0 for recalculation.
9. When M1026 = Off, M1029 will be On and the content in D12 will be reset to the set value in D10.
10. When this instruction is used with analog signal outputs, it will be able to buffer START and STOP.
11. DRAMP only supports SX V3.0, and above.

Program Example:

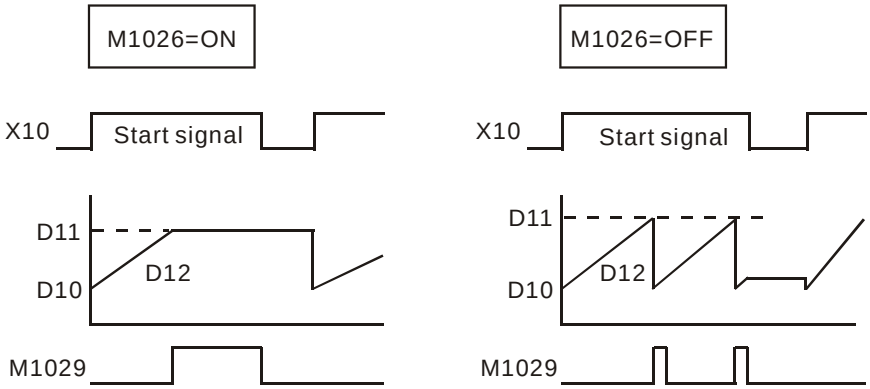




The scan times is stored in D13

Remarks:

D12 for enabling On/Off of M1026:



Please be noted that the content in D can only be modified when the instruction stops executing. Modification cannot be made in the execution of the instruction.

API	Mnemonic	Operands	Function
68	DTM	(S1) (D) (m) (n)	Data Transform and Move

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DTM: 9 steps
S_1													*			
S_2													*			
m					*	*							*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

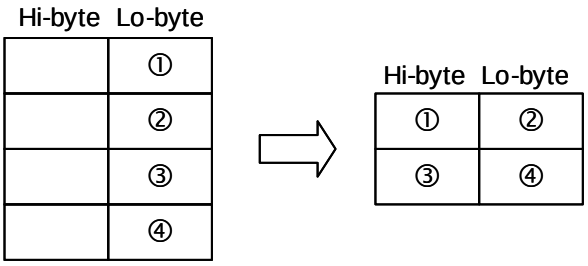
S_1: Start device of the source data stack **D**: Start device of the destination data stack **m**: Transformation mode

n: Length of source data stack

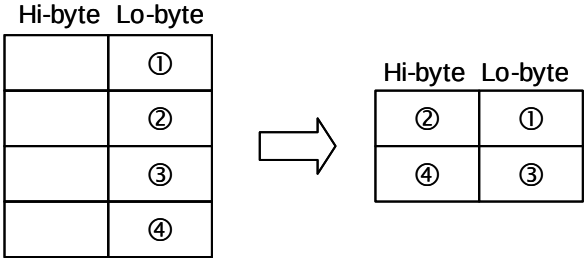
Explanations:

1. For parameter settings of operand **m**, please refer to the following description. K, H, D devices can be specified by operand **m.** If the set value is not in the available range, no transformation or move operation will be executed and no error will be detected.
2. K, H, D devices can be specified by operand **n,** which indicates the length of the source data stack. The available range for **n** is 1~256. If the set value falls out of available range, PLC will take the max value (256) or the min value (1) as the set value automatically.
3. Explanations on parameter settings of **m** operand:

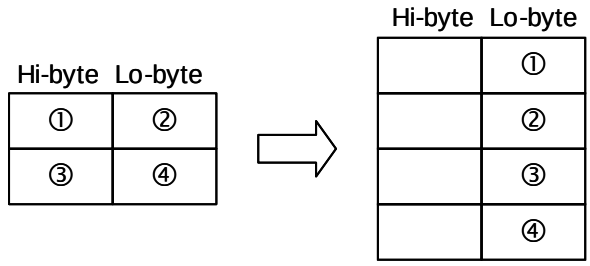
   K0: With n = 4, transform 8-bit data into 16-bit data (Hi-byte, Lo-byte) in the following rule:



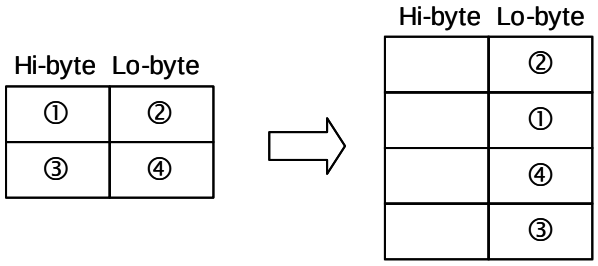
   K1: With n = 4, transform 8-bit data into 16-bit data (Lo-byte, Hi-byte) in the following rule:



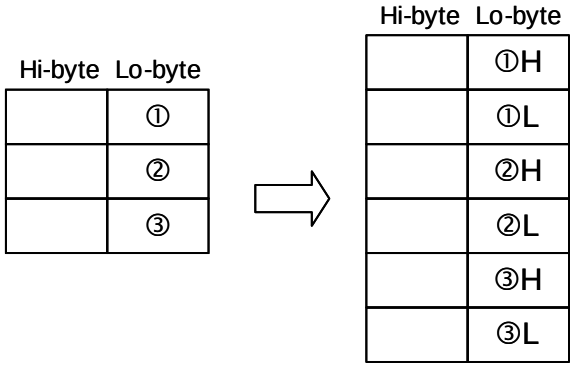
K2: With n = 2, transform 16-bit data (Hi-byte, Lo-byte) into 8-bit data in the following rule:



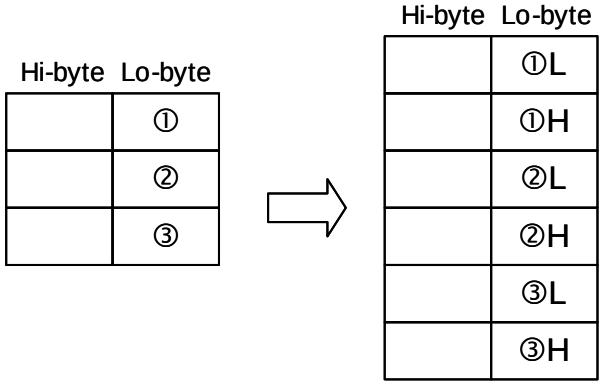
K3: With n = 2, transform 16-bit data (Lo-byte, Hi-byte) into 8-bit data in the following rule:



K4: With n = 3, transform 8-bit HEX data into ASCII data (higher 4 bits, lower 4 bits) in the following rule:

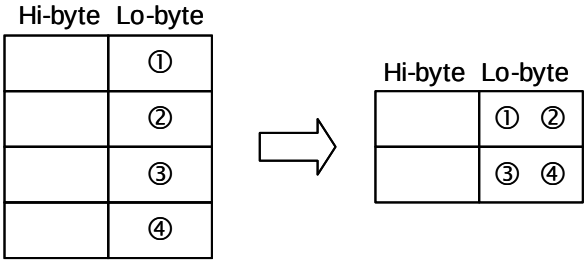


K5: With n = 3, transform 8-bit HEX data into ASCII data (lower 4 bits, higher 4 bits) in the following rule:

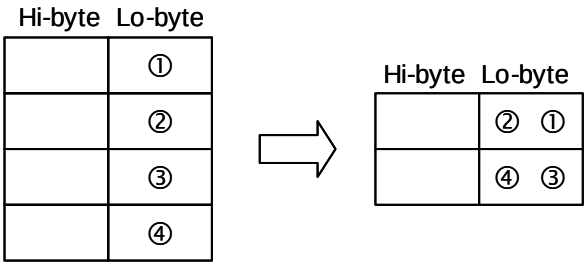


K6: When n = 4, transform 8-bit ASCII data (higher 4 bits, lower 4 bits) into HEX data in the following rule:

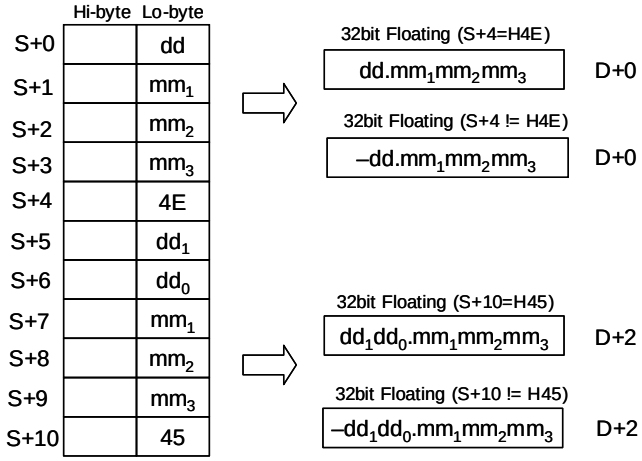
(ASCII value to be transformed includes 0 ~ 9 (0x30~0x39), A ~ F (0x41~0x46), and a ~ f (0x61~0x66).)



K7: When n = 4, transform 8-bit ASCII data (lower 4 bits, higher 4 bits) into HEX data in the following rule:



K8: Transform 8-bit GPS data into 32-bit floating point data in the following rule:



K9: Calculate the optimal frequency for positioning instructions with ramp up/ down function.

Users only need to set up the total number of pulses for positioning and the total time for positioning first, DTM instruction will automatically calculate the optimal max output frequency as well as the optimal start frequency for positioning instructions with ramp-up/down function such as PLSR, DDRVI and DCLLM.

Points to note:

1. When the calculation results exceed the max frequency of PLC, the output frequency will be set as 0.
2. When the total of ramp-up and ramp-down time exceeds the total time for operation, PLC will change the total time for operation (**S**+2) into “ramp-up time (**S**+3) + ramp-down time (**S**+4) + 1” automatically.

Explanation on operands: (For DVP-EH3 series PLCs whose version is 1.60 or below)

S+0, **S**+1: Total number of pulses for operation (32-bit)

S+2: Total time for operation (unit: ms)
D1343: Ramp-up time (unit: ms)
D1348: Ramp-down time (unit: ms)
D+0, **D**+1: Optimal max output frequency (unit: Hz) (32-bit)
D+2: Optimal start frequency (Unit: Hz)
n: Reserved

Whether the ramp-up time is equal to the ramp-down time depends on the setting of M1534. If the ramp-up time is not equal to the ramp-down time, there will be 30 sections of acceleration, and 30 sections of deceleration. If the ramp-up time is equal to the ramp-down time, there will be 60 sections of acceleration, and 60 sections of decelerations.

Explanation on operands: (For DVP-SV2 series PLCs whose version is 1.40 or below)
S+0, **S**+1: Total number of pulses for operation (32-bit)
S+2: Total time for operation (unit: ms)
S+3: Ramp-up time (unit: ms)
S+4: Ramp-down time (unit: ms)
D+0, **D**+1: Optimal max output frequency (unit: Hz) (32-bit)
D+2: Optimal start frequency (Unit: Hz)
n: Reserved

The ramp-up time is equal to the ramp-down time. There are 30 sections of acceleration, and 30 sections of deceleration.

Explanation on operands: (For DVP-EH3 series PLCs whose version is 1.62 (or above) and DVP-SV2 series PLCs whose version is 1.40 (or above)
S+0, **S**+1: Total number of pulses for operation (32-bit)
S+2: Total time for operation (unit: ms)
S+3: Ramp-up time (unit: ms)
S+4: Ramp-down time (unit: ms)
D+0, **D**+1: Optimal max output frequency (unit: Hz) (32-bit)
D+2: Optimal start frequency (Unit: Hz)
n: Reserved

If **S**+3 is equal to **S**+4, the ramp-up time is equal to the ramp-down time, and there are 60 sections of acceleration, and 60 sections of deceleration. If **S**+3 is not equal to **S**+4, the ramp-up time is not equal to the ramp-down time, and there are 30 sections of acceleration, and 60 sections of deceleration.

K11: Conversion from Local Time to Local Sidereal Time

Unlike the common local time defined by time zones, local sidereal time is calculated based on actual longitude. The conversion helps the user obtain the more accurate time difference of each location within the same time zone.

Explanation on operands:

S+0, **S**+1: Longitude (32-bit floating point value; East: positive, West: negative)

S+2: Time zone (16-bit integer; unit: hour)

S+3~ **S**+8: Year, Month, Day, Hour, Minute, Second of local time (16-bit integer)

D+0~**D**+5: Year, Month, Day, Hour, Minute, Second of the converted local sidereal time (16-bit integer)

n: Reserved

Example:

Input: Longitude F121.55, Time zone: +8, Local time: AM 8:00:00, Jan/6/2011

Conversion results: AM 8:06:12, Jan/6/2011

K12: Ramp value for multiple points (16-bit)

SV2/EH3: V1.88 or later version support K12~14.

Explanation on operands (16 bits):

S: input value

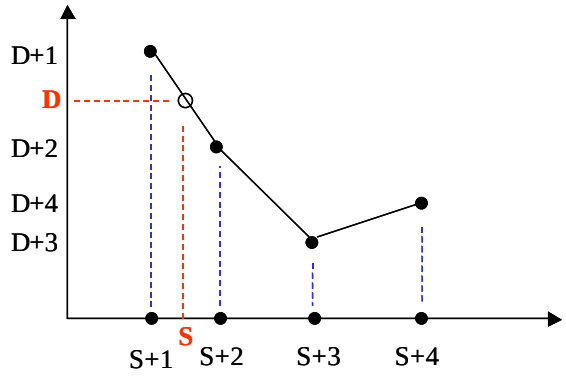
S+1, **S**+2….. **S**+n: input values for multiple points. The value of these values must be set by the following rule, **S**+1 must be the smallest value, **S**+2 must be larger than **S**+1 and so on. Therefore, **S**+n must be the largest value.

D: ramp value

D+1, **D** +2 … **D**+n: Range of ramp values for multiple points

n: Setting value for multiple points. The setting value is within the range between K2 ~ K50. If the setting value exceeds the range, the instruction is not executed.

The example of a curve is as follows. (n=K4)



Explanation of the example:

1. If **S** is larger than **S**+1 (S_1) and is less than **S**+2 (S_2), **D**+1 (D_1) and **D**+2 (D_2). D= ((**S** – S_1) x (D_2 – D_1) / (S_2 – S_1)) + D_1.
2. If **S** is less than **S**+1, **D** = D+1. If **S** is larger than **S**+**n**, **D** = **D**+**n**.
3. The floating-point numbers are involved in the operation. The output value is rounded down to the nearest whole digit, and then the 16-bit integer is output.

K13: Ramp value for multiple points (32-bit)

Please refer to parameter K12 for more information about the operands. The source device and the destination device are represented by 32-bit values.

K14: Floating-point value for multiple points

Please refer to parameter K12 for more information about the operands. The source device and the destination device are represented by 32-bit floating-point values.

K15: Sunrise and sunset times

SV2/EH3: V2.24 or later versions

Explanation on operands:

S1:

S1+0, **S**1+1: Local longitude (floating-points)

S1+1+2, **S**1+1+3: Local latitude (floating-point)

S1+1+4: Time zone (integer)

S1+1+5: If the day light saving time is enabled, it is in the integer format. 0: disable; others: enabled.

S1+6, **S**1+7, **S**1+8: Year, Month, Day, Hour, Minute, Second of local time (integer); the value used in year should be 2000 or later.

D:

D+0, **D**+1, **D**+2: Hour (24-hour format), Minute, Second of the converted local sunrise time (integer)

D+3, **D**+4, **D**+5: Hour (24-hour format), Minute, Second of the converted local sunset time (integer)

m: function code 15

n: Reserved

K16: String combination

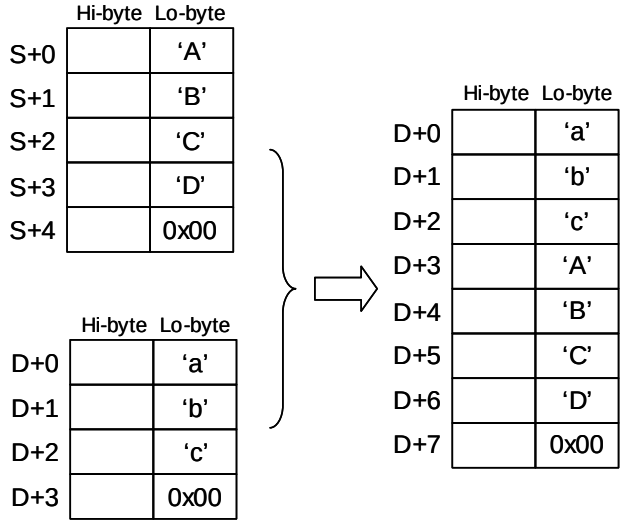
Explanations:

The system searches for the location of ETX (value 0x00) of the destination data string (lower 8 bits), then copies the data string starting of the source register (lower 8 bits) to the end of the destination data string. The source data string will be copied in byte order until the ETX (value 0x00) is reached.

Points to note:

The operand **n** sets the **max** data length after the string combination (max 256). If the ETX is not reached after the combination, the location indicated by **n** will be the ETX and filled with 0x00.

The combination will be **performed** in the following rule:

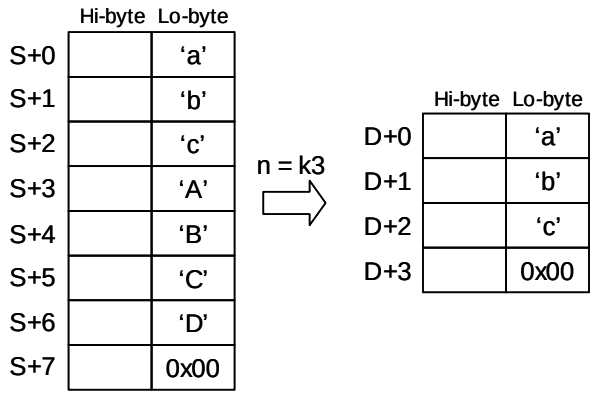


K17: String capture

Explanations:

The system copies the source data string (lower 8 bits) with the data length specified by operand n to the destination registers, where the n+1 register will be filled with 0x00. If value 0x00 is reached before the specified capture length n is completed, the capture will also be ended.

The capture will be performed in the following rule:



K18: Convert data string to floating point value

Explanations:

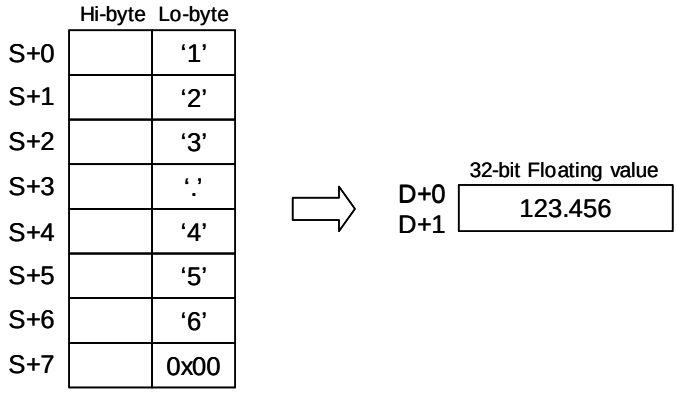
The system converts **n** words (lower 8 bits) of the source data string (decimal point is not included) to floating point value and stores the converted value in the destination device.

Points to note:

1. Operand **n** sets the number of total digits for the converted floating value. Max 8 digits are applicable and the value over **n** digit will be omitted. For example, **n** = K6, data string "123.45678" will be converted to "123.456".

2. When there are characters other than numbers 1~9 or the decimal point in the source data string, the character before the decimal point will be regarded as 0, and the value after the decimal point will be regarded as the ETX.
3. If the source data string contains no decimal point, the converted value will be displayed by an **n**-digit floating point value automatically.

The conversion will be performed in the following rule:



K19: Convert floating point value to data string

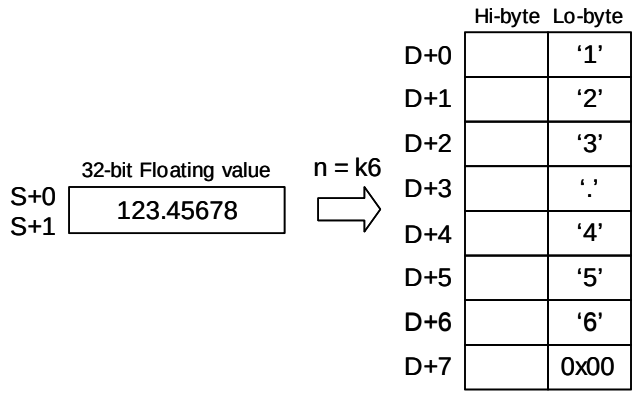
Explanations:

The system converts the floating point value in the source device S to data string with specified length **n** (decimal point is not included).

Points to note:

1. Operand **n** sets the number of total digits for the floating point value to be converted. Max 8 digits are applicable and the value over **n** digit will be omitted. For example, **n** = K6, floating value F123.45678 will be converted to data string "123.456".
2. When the digits of source value are more than the specified **n** digits, only the **n** digits from the left will be converted. For example, source value F123456.78 with **n**=K4 will be converted as data string "1234".
3. If the source value is a decimal value without integers, e.g. 0.1234, the converted data string will be ".1234" where the first digit is the decimal point.

The conversion will be performed in the following rule:



K30: Bit swap 16-bit data (for EC3-8K)

Use K30 to swap data in S1~S1+(N-1) and store the swapped data in S2~S2+(N-1).

Bit Swap action: BIT15⇔BIT0, BIT14⇔BIT1, BIT13⇔BIT2, and so forth.

Explanation: DTM D0 D10 K30 K8

D0 = 0x0001	D10 = 0x8000
D1 = 0x0002	D11 = 0x4000
D2 = 0x0004	D12 = 0x2000
D3 = 0x0008	D13 = 0x1000
D4 = 0x0010	D14 = 0x0800
D5 = 0x0020	D15 = 0x0400
D6 = 0x0040	D16 = 0x0200
D7 = 0x0080	D17 = 0x0100

K31: Copy WORD data to registers of PLC in a consecutive order

SV2/EH3: V1.88 or later versions supports K31~K36

Explanation on operands:

S: the first source data of the device D

D: the first target data of the device D

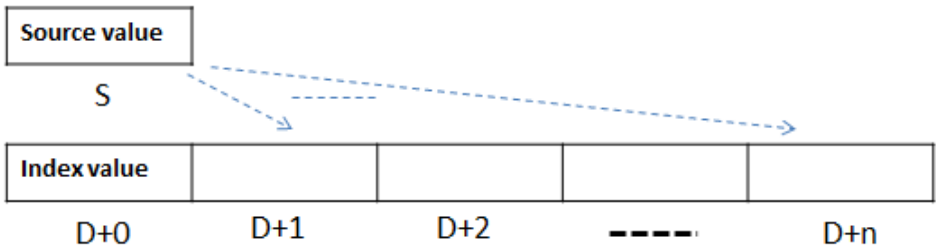
m: parameters setting

n: length of the source data (**n**=1~256)。

Note: D1000~D1999 are not for use

Explanation:

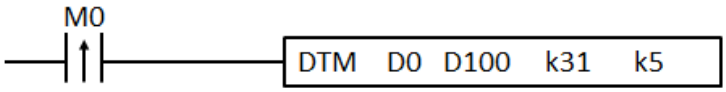
Copy the source value to the D target device of the index value D+0; once complete, add 1 to the value in D+0.



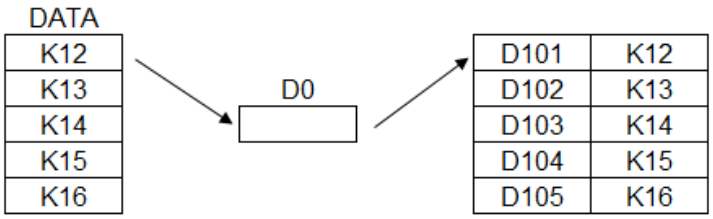
Note: When the value in D+0 is less than 1 (default), copy the value in D+0 and add 1 to the value in D+0.

When the value in D+0 is bigger than n (default: n+1), do not copy the value.

Explanation of the example:



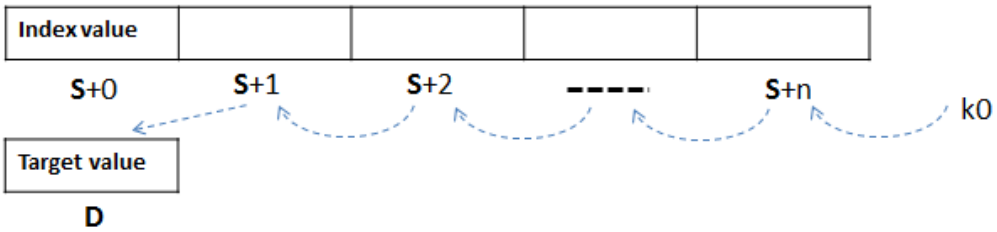
If M0 is OFF -> ON for 5 times in a row, DTM instruction copies values in D0 to D101~D105 in a consecutive order. After the DTM instruciotn is exected, add 1 to the value of D100. See the example below for reference.



K32: Move the registers of PLC in a consecutive order (from the first to the last) to some specified registers.

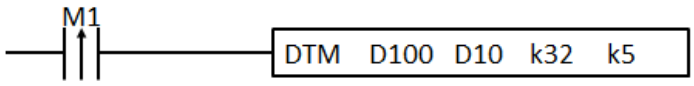
Explanation:

Move the value in **S**+1 (the source data) to the target device D and move the value in **S**+2 to **S**+1 till all S source data of the index value are moved forward and insert K0 in **S**+n. Subtract 1 from the index value of **S**+0.



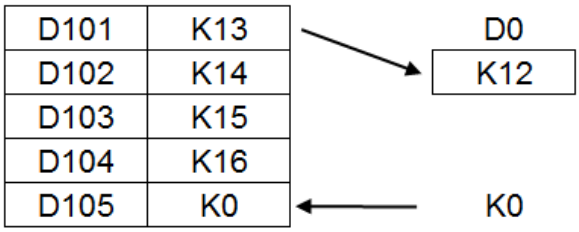
Note: When the value in **S**+0 is less than 2, it means there is no data and no action is required. When the value in **S**+0 is larger than n+1, it means data is full and no action is required, no error message and the index value of **S**+0 is unchanged. See the example below for reference.

Explanation of the example:

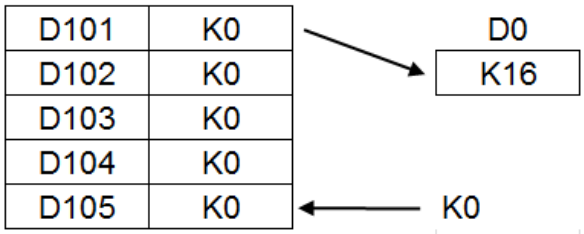


When M1 is OFF-> ON, the DTM instruction moves the value in D101 to D0. After the DTM instruciotn is exected, Subtract 1 from the value of D100. See the example below for reference.

After one time executionof DTM instruciton, move the value (K12) in D101 to D0 and insert K0 to the value of D105.



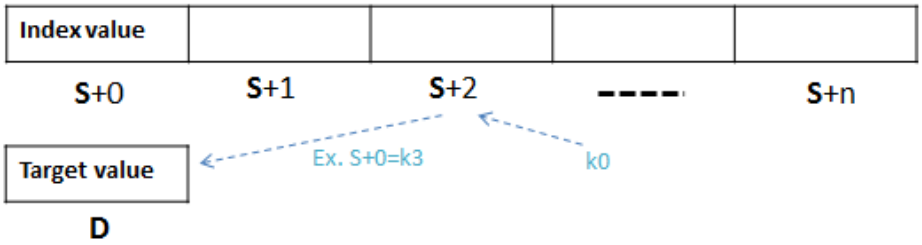
After five-time executions of DTM instruciton, D0 is K16 and the value in D101~D105 are all K0.



K33: Move the registers of PLC in a consecutive order (from the last to the first) to some specified registers.

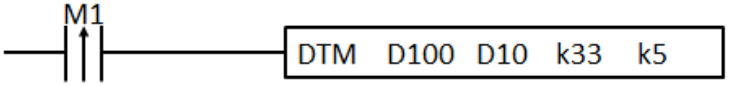
Explanation:

Subtract 1 from the index value of **S**+0. Move the value in S+[**S**+0] (the source data) to the target device D and insert K0 in **S**+[**S**+0].



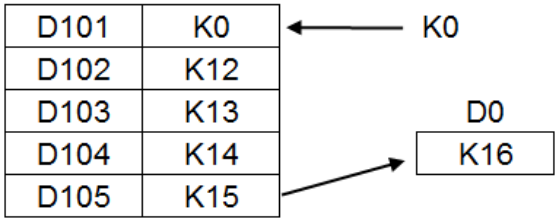
Note: When the value in **S**+0 is less than 2, it means there is no data and no action is required. When the value in **S**+0 is larger than n+1, it means data is full and no action is required, no error message and the index value of **S**+0 is unchanged. See the example below for reference.

Explanation of the example:

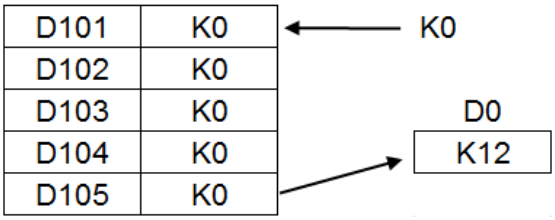


When M1 is OFF-> ON, the DTM instruction moves the value in D105 to D0. After the DTM instruciotn is exected, subtract 1 from the value of D100. See the example below for reference.

After one time executionof DTM instruciton, move the value (K16) in D105 to D0 and insert K0 to the value of D101.



After five-time executions of DTM instruciton, D0 is K12 and the value in D101~D105 are all K0.



K34: Copy BIT data to registers of PLC in a consecutive order

Explanation on operands:

S: the first source data of the device M

D: the first target data of the device M

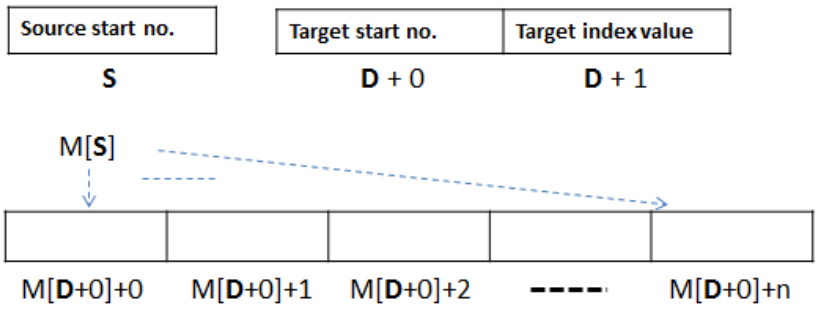
m: parameters setting

n: length of the source data (**n**=1~256)。

Note: M1000~M1999 are not for use

Explanation:

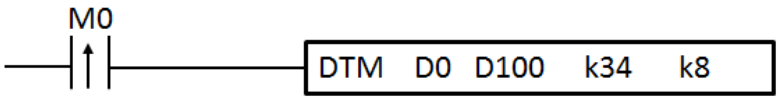
Copy the M status of the source value to the M[**D**+0] target device of the index value D+1; once complete, add 1 to the value in D+1.



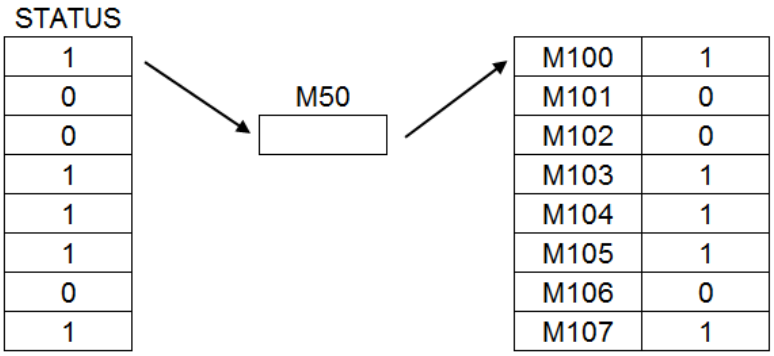
Note: When the value in **D**+1 is less than 0, edit the value to 0 and copy the status of M[**S**] to M[**D**+0]+0 and add 1 to the value in **D**+1.

When the value in **D**+1 is bigger than n-1, edit the value in **D**+1 to n but do not copy the value.

Explanation of the example:



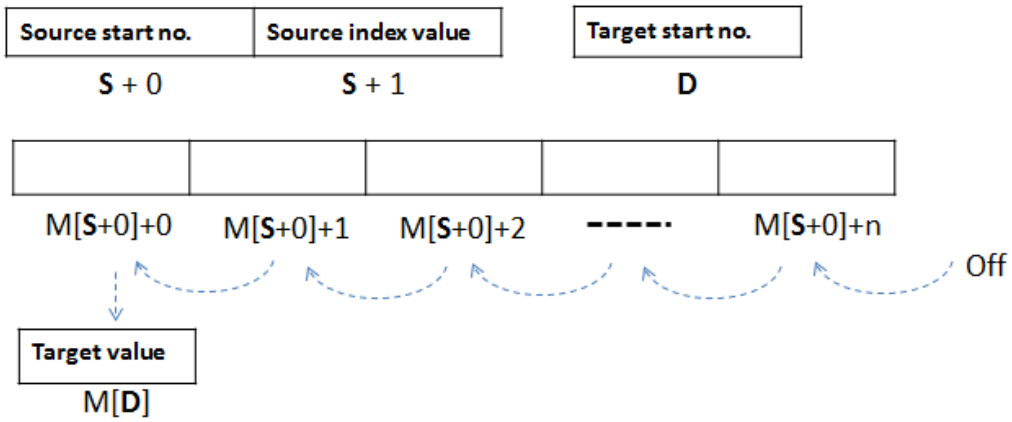
Set D0=K50, D100=K100, and DTM copy the status of M50 to M100~107. After each execution of DTM instruction, add 1 to the value of D101. See the example below for reference.



K35: Move the bit data in a consecutive order (from the first to the last) to specified bit registers.

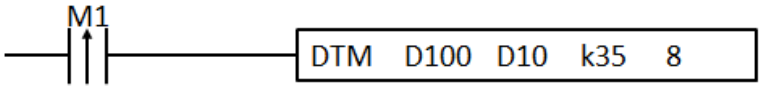
Explanation:

Move the value in M[**S**+0]+0 (the source data) to the target device M[**D**] and move the value in M[**S**+0]+1 to M[**S**+0]+0 till all M[**S**+0]+**S**+1 source data are moved forward and the state changed to OFF. Subtract 1 from the index value of **S**+1.

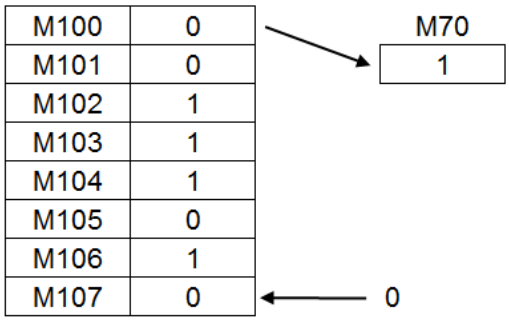


Note: When the value in **S**+1 is less than 1, it means there is no data and no action is required. When the value in **S**+1 is larger than n, it means data is full and no action is required, no error message and the index value of **S**+1 is unchanged. See the example below for reference.

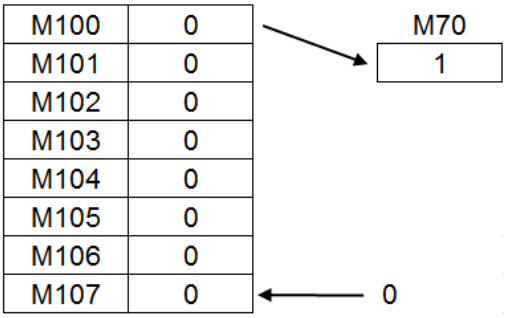
Explanation of the example:



When D100 = K100, D10 = K70, the DTM instruction moves the states of M100~107 to M70. After the DTM instruciotn is exected, subtract 1 from the value of D101. See the example below for reference.
After one time executionof DTM instruciton, move the value (1) in M100 to M70 and insert 0 to the value of M107.



After 8-time executions of DTM instruciton, the state of M70 is 1 and the states in M100~M107 are all 0.

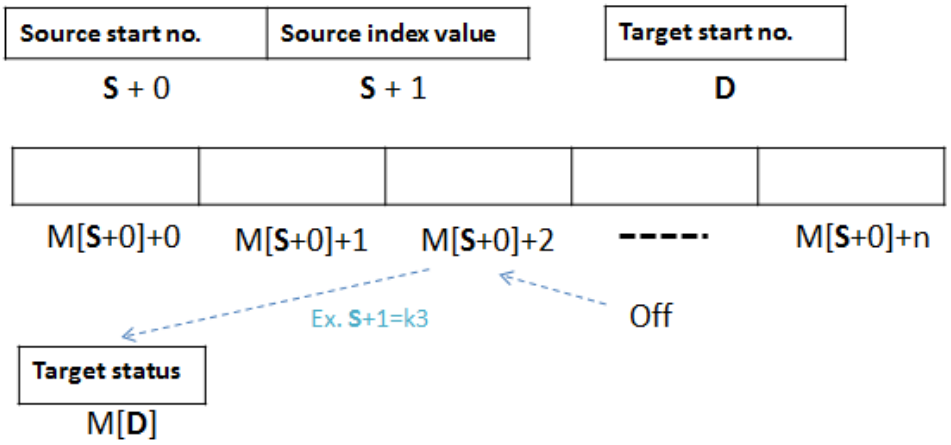


K36: Move the bit data in a consecutive order (from the last to the first) to specified bit registers.

Explanation:

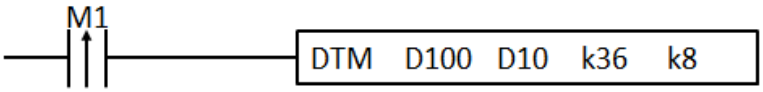
Move the value in M[**S**+0]+0 (the source data) to the target device M[**D**] and move the value in M[**S**+0]+1 to M[**S**+0]+0 till all M[**S**+0]+**S**+1 source data are moved forward and the state changed to Off. Subtract 1 from the index value of **S**+1.

Subtract 1 from the index value of **S**+1. Move the state in M[**S**+0]+**S**+1 (the source data) to the target device M[**D**] and change the state of M[**S**+0]+**S**+1 to OFF.

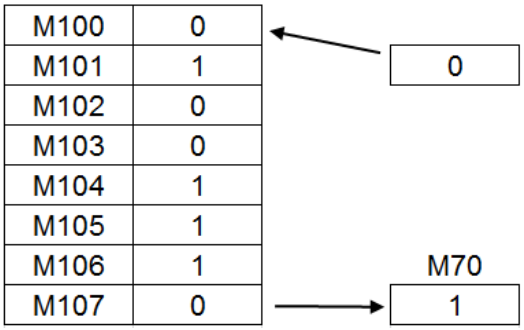


Note: When the value in **S**+1 is less than 1, it means there is no data and no action is required. When the value in **S**+1 is larger than n, it means data is full and no action is required, no error message and the index value of **S**+1 is unchanged. See the example below for reference.

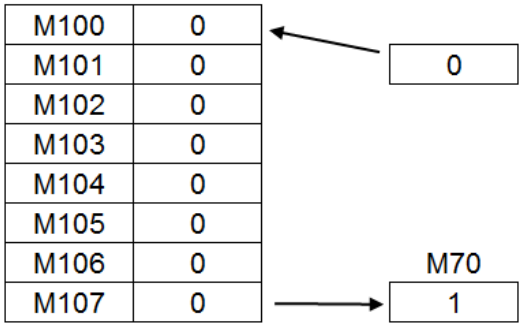
Explanation of the example:



When D100 = K100, D10 = K70, the DTM instruction moves the states of M100~107 to M70. After the DTM instruciotn is exected, subtract 1 from the value of D101. See the example below for reference.
After one time executionof DTM instruciton, move the value (1) in M100 to M70 and insert 0 to the value of M107.



After 8-time executions of DTM instruciton, the state of M70 is 1 and the states in M100~M107 are all 0.



K39: Read the PLC serial number (the PLC serial number contains 17 characters and that takes 9 words)

EH3/SV2: V2.06 or later

Explanation on operands:

S: the first source data of the device D (of no use)

D: the first target data of the device D

m: parameters setting

n: length of the source data (of no use)

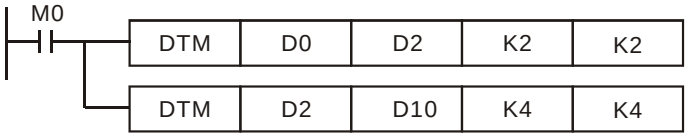
Note: D1000~D1999 are not for use

Example:

The PLC serial number is 32EH00T3W17010001 and if **D** device is D10, the readings will be stored as the followings, D10=H3332, D11=H4548, D12=H3030，D18=H3031.

Program Example 1: K2, K4

1. When M0 = ON, transform 16-bit data in D0, D1 into ASCII data in the following order: H byte - L byte - H byte - Low byte, and store the results in D10 ~ D17.



2. Value of source devices D0, D1:

Register	D0	D1
Value	H1234	H5678

3. When the 1st DTM instruction executes (m=K2), ELC transforms the 16-bit data (Hi-byte, Lo-byte) into 8-bit data and move to registers D2~D5.

Register	D2	D3	D4	D5
Value	H12	H34	H56	H78

4. When the 2nd DTM instruction executes (m=K4), ELC transforms the 8-bit HEX data into ASCII data and move to registers D10~D17.

Register	D10	D11	D12	D13	D14	D15	D16	D17
Value	H0031	H0032	H0033	H0034	H0035	H0036	H0037	H0038

Program Example 2: K9

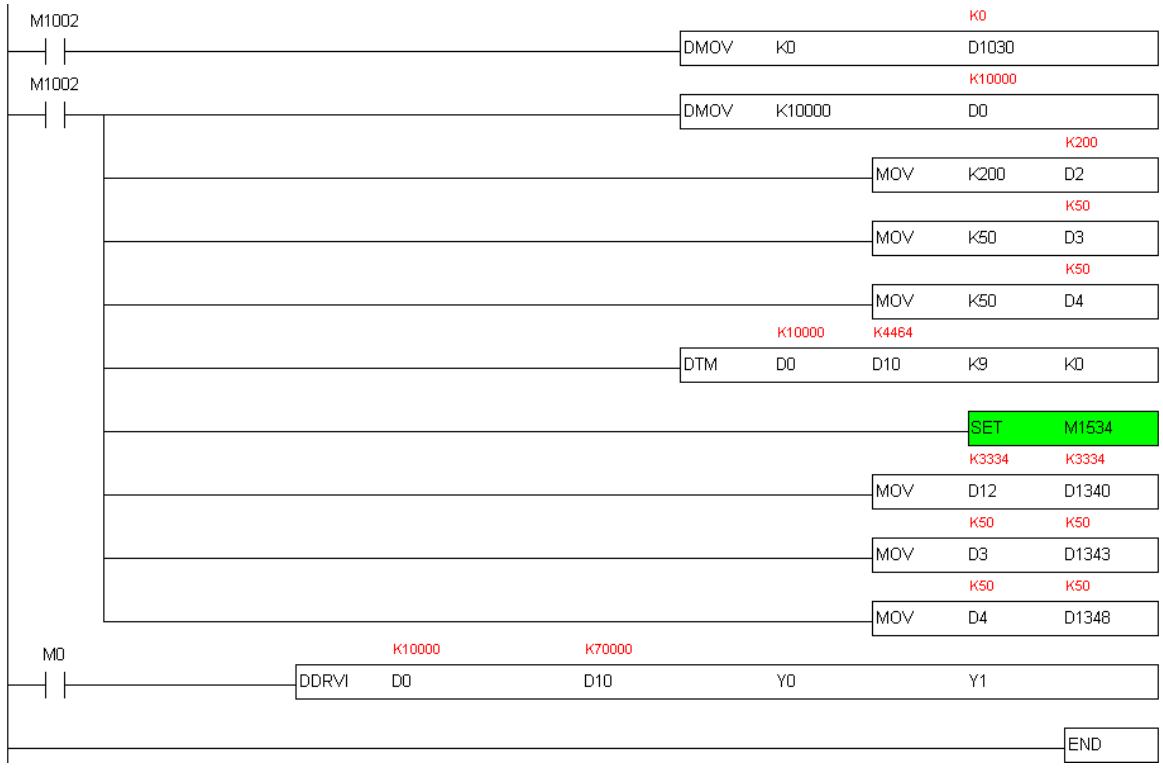
m = K9

1. Set up total number of pulses, total time, ramp-up time and ramp-down time in source device starting with D0. Execute DTM instruction and the optimal max frequency as well as optimal start frequency can be obtained and executed by positioning instructions.
2. Assume the data of source device is set up as below:

Total Pulses	Total Time	Ramp-up Time	Ramp-down Time
D0, D1	D2	D3	D4
K10000	K200	K50	K50

3. The optimal positioning results can be obtained as below:

Optimal max frequency	Optimal start frequency
D10, D11	D12
K70000	K3334



API	Mnemonic			Operands	Function
69	D	SORT		S m1 m2 D n	Sort Tabulated Data

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			SORT: 11 steps
m_1					*	*										DSORT: 21 steps
m_2					*	*										
D													*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start device for the original data **m_1**: Groups of data to be sorted **m_2**: Number of columns of data

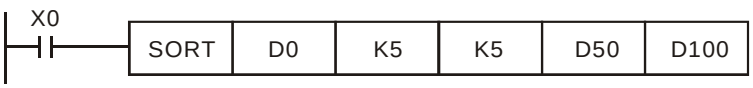
D: Start device for the sorted data **n**: Reference value for data sorting

Explanations:

1. Range of **m_1**: 1 ~ 32.
2. Range of **m_2**: 1 ~ 6
3. Range of **n**: 1 ~ **m_2**
4. See the specifications of each model for their range of use.
5. Flag: M1029 (SORT execution completed).
6. The sorted result is stored in **m_1** × **m_2** registers starting from the device designated in **D**. Therefore, if **S** and **D** designate the same register, the sorted result will be the same as the data designated in **S**.
7. It is better that the start No. designated in **S** is 0.
8. The sorting will be completed after **m_1** times of scans. After the sorting is completed, M1029 will be On.
9. There is no limitation on the times of using this instruction. However, only one instruction can be executed at a time.
10. The function of sorting one-dimensional data is added. If users set **m_1** and **m_2** to 1, the function will be enabled. The operand **n** represents the number of data. It must be in the range of 1 to 32. The data in the **n** devices starting from **S** is sorted. The sorting result is stored in the devices starting from **D**. This function only needs one scan time. After data is sorted. M1029 will be ON. This function supports DVP-EH3 series PLCs whose version is 1.62, DVP-SV2 series PLCs whose version is 1.62, DVP-SX series PLCs whose version is 3.0, and above.
11. The 32-bit instruction DSORT is added. It supports DVP-EH3 series PLCs whose version is 1.62, DVP-SV2 series PLCs whose version is 1.62, DVP-SX seires PLCs whose version is 3.0, DVP-EC3-8K whose version is V8.60 and above.

Program Example:

1. When X0 = On, the sorting will start. When the sorting is completed, M1029 will be On. DO NOT change the data to be sorted during the execution of the instruction. If you wish to change the data, please make X0 go from Off to On again.



2. Example table of data sorting

Columns of data: m_2

Groups of data: m_1

Data Column					
Column / Row	1	2	3	4	5
	Students No.	Physics	English	Math	Chemistry
1	(D0) 1	(D5) 90	(D10) 75	(D15) 66	(D20) 79
2	(D1) 2	(D6) 55	(D11) 65	(D16) 54	(D21) 63
3	(D2) 3	(D7) 80	(D12) 98	(D17) 89	(D22) 90
4	(D3) 4	(D8) 70	(D13) 60	(D18) 99	(D23) 50
5	(D4) 5	(D9) 95	(D14) 79	(D19) 75	(D24) 69

Sorted data when D100 = K3.

Columns of data: m_2

Groups of data: m_1

Data Column					
Column / Row	1	2	3	4	5
	Students No.	Physics	English	Math	Chemistry
1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
3	(D52) 1	(D57) 90	(D62) 75	(D67) 66	(D72) 79
4	(D53) 5	(D58) 95	(D63) 79	(D68) 75	(D73) 69
5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

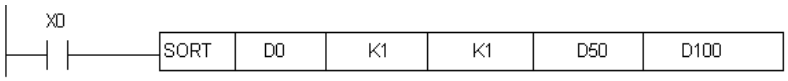
Sorted data when D100 = K5.

Columns of data: m_2

Groups of data: m_1

Data Column					
Column / Row	1	2	3	4	5
	Students No.	Physics	English	Math	Chemistry
1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
3	(D52) 5	(D57) 95	(D62) 79	(D67) 75	(D72) 69
4	(D53) 1	(D58) 90	(D63) 75	(D68) 66	(D73) 79
5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

3. Example of sorting one-dimensional data: If X0 is ON, the data specified will be sorted. After the data is sorted, M1092 will be ON.



Owing to the fact that **m_1** and **m_2** are set to K1, one-dimensional data will be sorted. 5 values will be sorted (D100=K5). The values in D0~D4 are shown below.

(a) The values in D0~D4 are shown below.

Data source	D0	D1	D2	D3	D4
Data	75	65	98	60	79

(b) The sorting result is stored in D50~D54.

Sorting result	D50	D51	D52	D53	D54
Data	60	65	75	79	98

API	Mnemonic	Operands	Function
70	D TKY	S D1 D2	Ten Key Input

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												TKY: 7 steps DTKY: 13 steps
D_1								*	*	*	*	*	*	*	*	
D_2		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

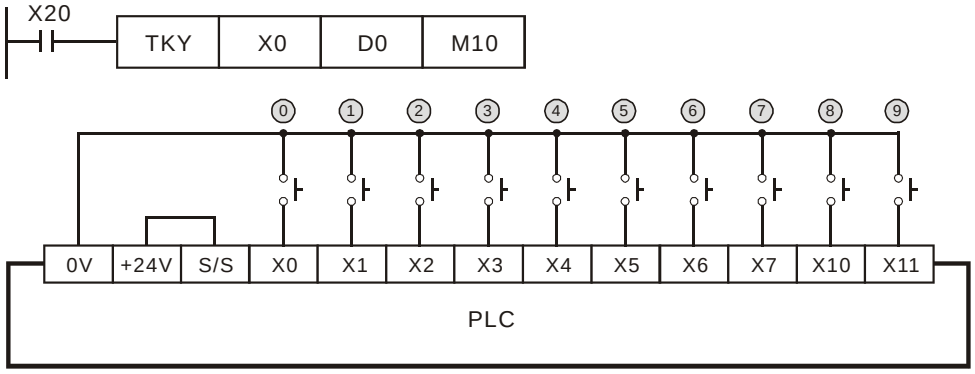
S: Start device for key input **D_1**: Device for storing keyed-in value **D_2**: Key output signal

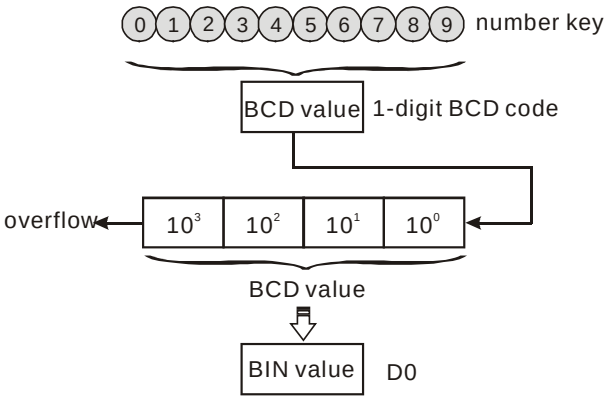
Explanations:

1. **S** will occupy 10 consecutive points; **D_2** will occupy 11 consecutive points.
2. See the specifications of each model for their range of use.
3. For SA series MPU, **S** and **D_2** do not support E, F index register modification.
4. This instruction designates 10 external input points (representing decimal numbers 0 ~ 9) starting from **S**. The 10 points are respectively connected to 10 keys. By pressing the keys, you can enter a 4-digit decimal figure 0 ~ 9,999 (16-bit instruction) or a 8-digit figure 0 ~ 99,999,999 (32-bit instruction) and store the figure in **D_1**. **D_2** is used for storing key status.
5. There is no limitation on the times of using this instruction. However, only one instruction can be executed at a time.

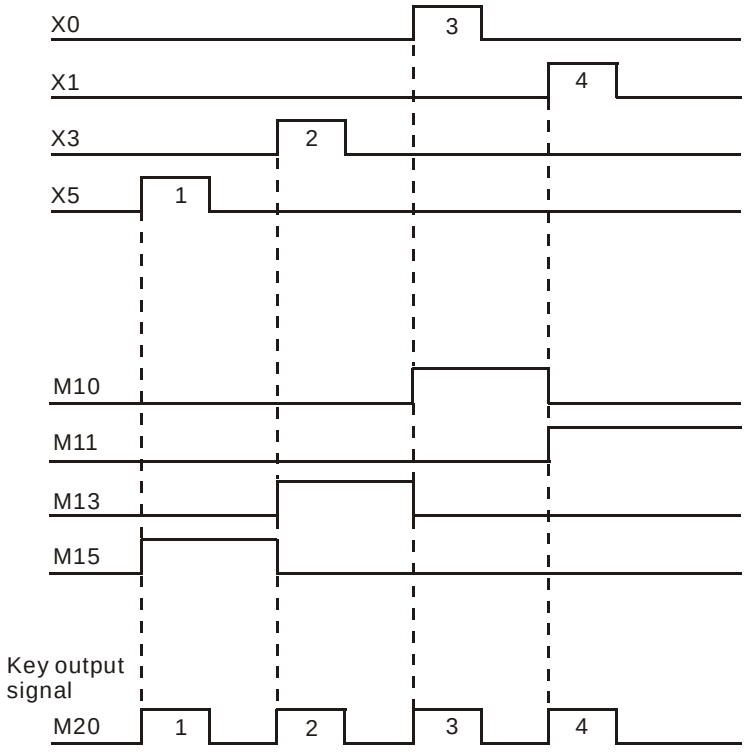
Program Example:

1. Connect the 10 input points starting from X0 to the 10 keys (0 ~ 9). When X20 = On, the instruction will be executed and the keyed-in values will be stored in D0 in bin form. The key status will be stored in M10 ~ M19.





2. As shown in the timing chart below, the 4 points X5, X3, X0, and X1 connected to the keys are entered in order and you can obtain the result 5,301. Store the result in D0. 9,999 is the maximum value allowed to stored in D0. Once the value exceeds 4 digits, the highest digit will overflow.
3. M12 = On when from X2 is pressed to the other key is pressed. Same to other keys.
4. When any of the keys in X0 ~ X11 is pressed, one of M10 ~ 19 will be On correspondingly.
5. M20 = On when any of the keys is pressed.
6. When X20 goes Off, the keyed-in value prior to D0 will remain unchanged, but M10 ~ M20 will all be Off.



API	Mnemonic		Operands	Function
71	D	HKY	S D1 D2 D3	Hexadecimal Key Input

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	HKY: 9 steps DHKY: 17 steps
S	*															
D_1		*														
D_2											*	*	*	*	*	
D_3		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

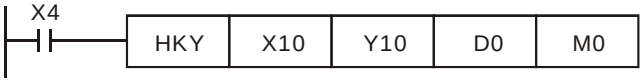
S: Start device for key scan input **D_1**: Start device for key scan output **D_2**: Device for storing keyed-in value **D_3**: Key output signal

Explanations:

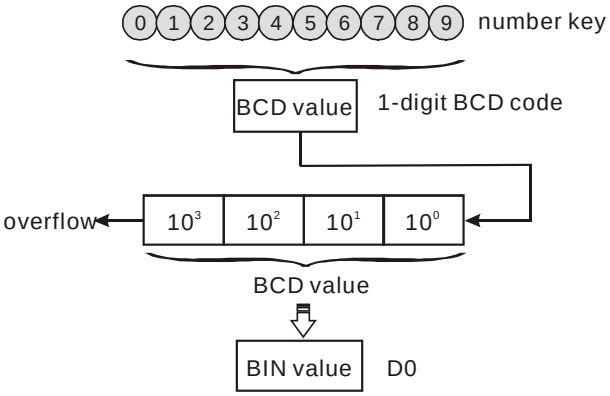
1. **S** will occupy 4 consecutive points.
2. **D_1** will occupy 4 consecutive points.
3. **D_3** will occupy 8 consecutive points.
4. See the specifications of each model for their range of use.
5. For SA series MPU, **S**, **D_1** and **D_3** do not support E, F index register modification.
6. Flags: M1029 (On whenever a matrix scan period is completed); M1167 (HKY input modes switch). See remarks for more details.
7. This instruction designates 4 continuous external input points starting from **S** and 4 continuous external input points starting from **D_1** to construct a 16-key keyboard by a matrix scan. The keyed-in value will be stored in **D_2** and **D_3** is used for storing key status. If several keys are pressed at the same time, the first key pressed has the priority.
8. The keyed-in value is termporarily stored in D0. When the 16-bit instruction HKY is in use, 9,999 is the maximum value D0 is able to store. When the value exceeds 4 digits, the highest digit will overflow. When the 32-bit instruction DHKY is in use, 99,999,999 is the maximum value D0 is able to store. When the value exceeds 8 digits, the highest digit will overflow.
9. There is no limitation on the times of using this instruction. However, only one instruction can be executed at a time.

Program Example:

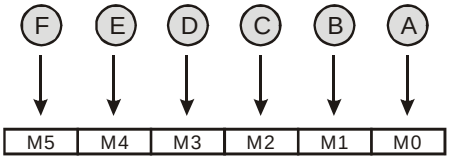
1. Designate 4 input points X10 ~ X13 and the other 4 input points Y10 ~ Y13 to construct a 16-key keyboard. When X4 = On, the instruction will be executed and the keyed-in value will be stored in D0 in bin form. The key status will be stored in M0 ~ M7.



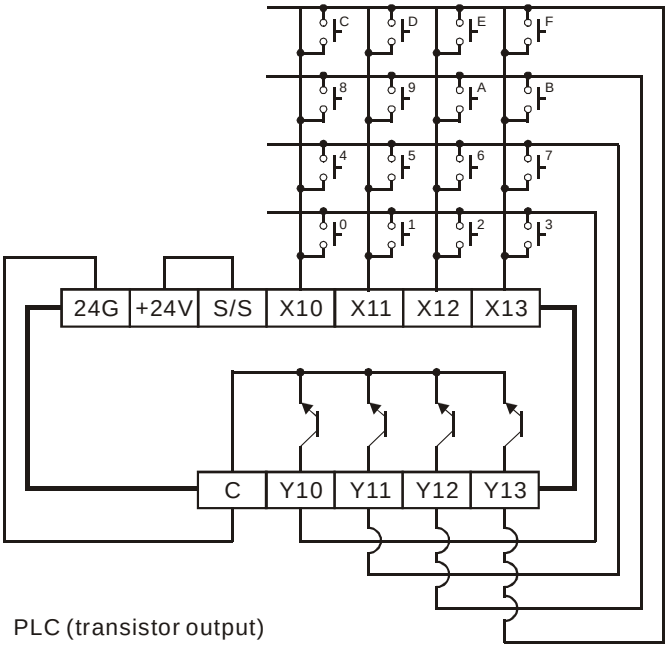
2. Key in numbers:



3. Function keys input:
   a) When A is pressed, M0 will be On and retained. When D is pressed next, M0 will be Off, M3 will be On and retained.
   b) When many keys are pressed at the same time, the first key pressed has the priority.



4. Key output signal:
   a) When any of A ~ F is pressed, M6 will be On for once.
   b) When any of 0 ~ 9 is pressed, M7 will be On for once.
5. When X4 goes Off, the keyed-in value prior to D0 will remain unchanged, but M0 ~ M7 will all be Off.
6. External wiring:



Remarks:

1. When this instruction is being executed, it will require 8 scans to obtain one valid keyed-in value. A scan period that is too long or too short may result in poor keyed-in effect, which can be avoided by the following methods:
   a) If the scan period is too short, I/O may not be able to respond in time, resulting in not being able to read the keyed-in value correctly. In this case, please fix the scan time.
   b) If the scan period is too long, the key may respond slowly. In this case, write this instruction into the time interruption subroutine to fix the time for the execution of this instruction.
2. Functions of M1167:
   a) When M1167 = On, HKY instruction will be able to input the hexadecimal value of 0 ~ F.
   b) When M1167 = Off, HKY instruction will see A ~ F as function keys.
3. Functions of D1037 (only supports EH3/SV2 series MPU):
   Write D1037 to set the overlapping time for keys (unit: ms). The overlapping time will vary upon different program scan time and the settings in D1037.

API	Mnemonic	Operands	Function
72	DSW	(S) (D1) (D2) (n)	Digital Switch

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															DSW: 9 steps
D_1		*														
D_2											*	*	*			
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

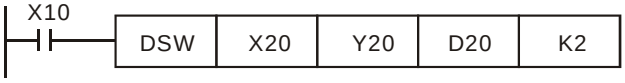
S: Start device for switch scan input **D_1**: Start device for switch scan output **D_2**: Device for storing the set value of switch **n**: Groups of switches

Explanations:

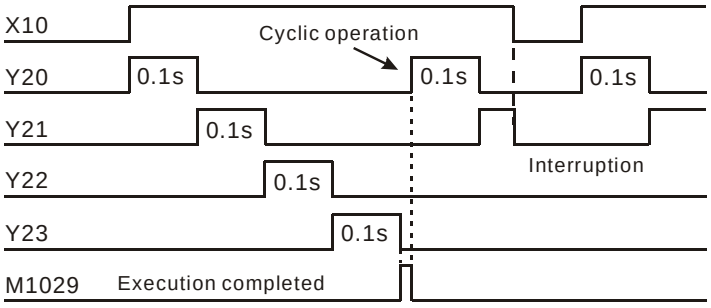
1. Range of **n**: 1 ~ 2
2. **S** and **D_1** in SX series MPU do not support E, F index register modification.
3. See the specifications of each model for their range of use.
4. Flag: M1029 (DSW execution completed)
5. This instruction designates 4 or 8 consecutive external input points starting from **S** and 4 consecutive external input points starting from **D_1** to scan read 1 or 2 4-digit DIP switches.The set values of DIP switches are stored in **D_2**. **n** decides to read 1 or 2 4-digit DIP switches.
6. There is no limitation on the times of using this instruction in the program. However, for SA series MPU, only one instruction can be executed at a time. For EH series MPU, two instructions are allowed to be executed at a time.

Program Example:

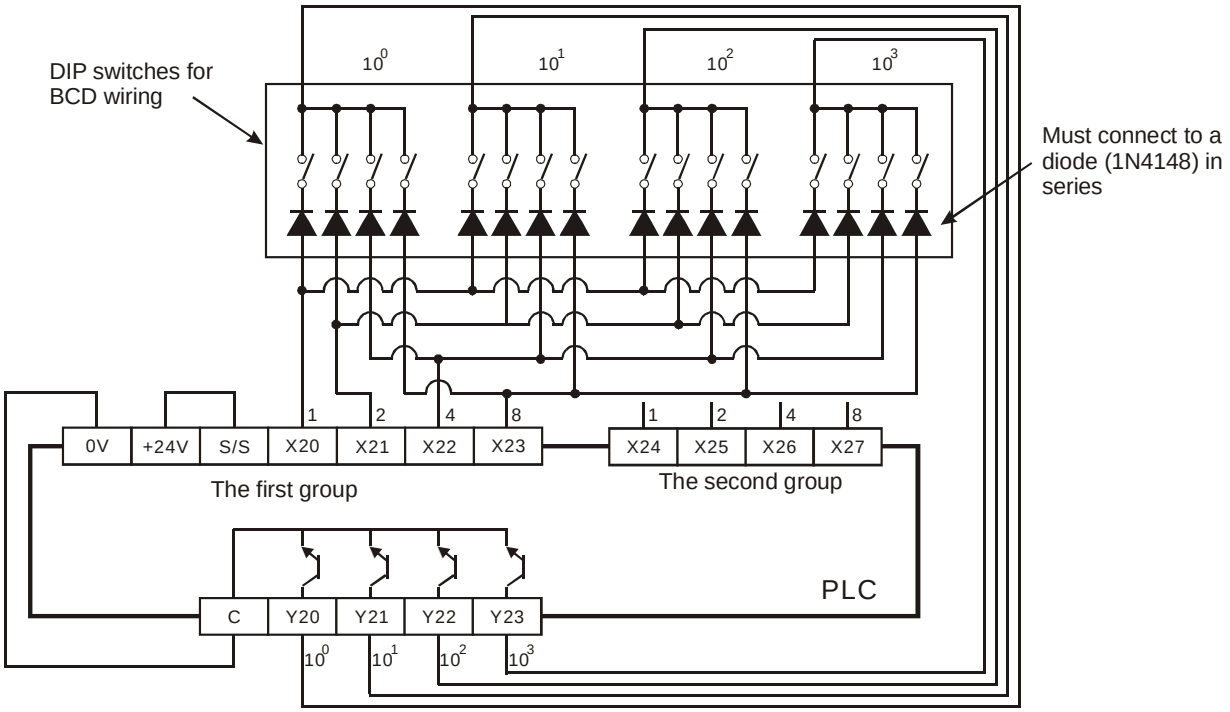
1. The first group of DIP switches consist of X20 ~ X23 and Y20 ~ Y23. The second group of switches consist of X24 ~ X27 and Y20 ~ Y23. When X10 = On, the instruction will be executed and the set values of the first group switches will be read and converted into bin values before being stored in D20. The set values of the second group switches will be read, converted into bin values and stored in D21.



2. When X10 = On, the Y20 ~ Y23 auto scan cycle will be On. Whenever a scan cycle is completed, M1029 will be On for a scan period.
3. Please use transistor output for Y20 ~ Y23. Every pin 1, 2, 4, 8 shall be connected to a diode (0.1A/50V) before connecting to the input terminals on PLC.

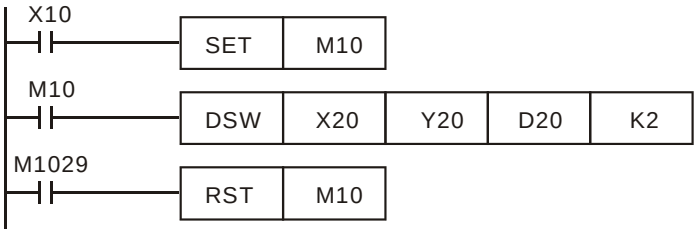


4. Wiring for DIP swich input:



Remarks:

1. When **n** = K1, $\mathbf{D_2}$ will occupy one register. When **n** = K2, $\mathbf{D_2}$ will occupy 2 consecutive registers.
2. Follow the methods below for the transistor scan output:
   a) When X10 = On, DSW instruction will be executed. When X10 goes Off, M10 will keep being On until the scan output completes a scan cycle and go Off.
   b) When X10 is used as a button switch, whenever X10 is pressed once, M10 will be reset to Off when the scan output designated by DSW instruction completes a scan cycle. The DIP switch data will be read completely and the scan output will only operate during the time when the button switch is pressed. Therefore, even the scan output is a transistor type, the life span of the transistor can be extended because it does not operate too frequently.



API	Mnemonic		Operands	Function
73	SEGD	P	(S) (D)	Seven Segment Decoder

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SEGD, SEGDP: 5 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source device to be decoded **D**: Output device after the decoding

Explanations:

See the specifications of each model for their range of use.

Program Example:

When X10 = On, the contents (0 ~ F in hex) of the lower 4 bits (b0 ~ b3) of D10 will be decoded into a 7-segment display for output. The decoded results will be stored in Y10 ~ Y17. If the content exceeds 4 bits, the lower 4 bits are still used for the decoding.

Decoding table of the 7-segment display:

Hex	Bit combination	Composition of the 7-segment display	Status of each segment							Data displayed
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)	
0	0000		ON	ON	ON	ON	ON	ON	OFF	
1	0001		OFF	ON	ON	OFF	OFF	OFF	OFF	
2	0010		ON	ON	OFF	ON	ON	OFF	ON	
3	0011		ON	ON	ON	ON	OFF	OFF	ON	
4	0100		OFF	ON	ON	OFF	OFF	ON	ON	
5	0101		ON	OFF	ON	ON	OFF	ON	ON	
6	0110		ON	OFF	ON	ON	ON	ON	ON	
7	0111		ON	ON	ON	OFF	OFF	ON	OFF	
8	1000		ON	ON	ON	ON	ON	ON	ON	
9	1001		ON	ON	ON	ON	OFF	ON	ON	
A	1010		ON	ON	ON	OFF	ON	ON	ON	
B	1011		OFF	OFF	ON	ON	ON	ON	ON	
C	1100		ON	OFF	OFF	ON	ON	ON	OFF	
D	1101		OFF	ON	ON	ON	ON	OFF	ON	
E	1110		ON	OFF	OFF	ON	ON	ON	ON	
F	1111		ON	OFF	OFF	OFF	ON	ON	ON	

API	Mnemonic	Operands	Function
74	SEGL	(S) (D) (n)	Seven Segment with Latch

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SEGL: 7 steps
D		*														
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source device to be displayed in 7-segment display **D**: Start device for 7-segment display scan output

n: Polarity setting of output signal and scan signal

Explanations:

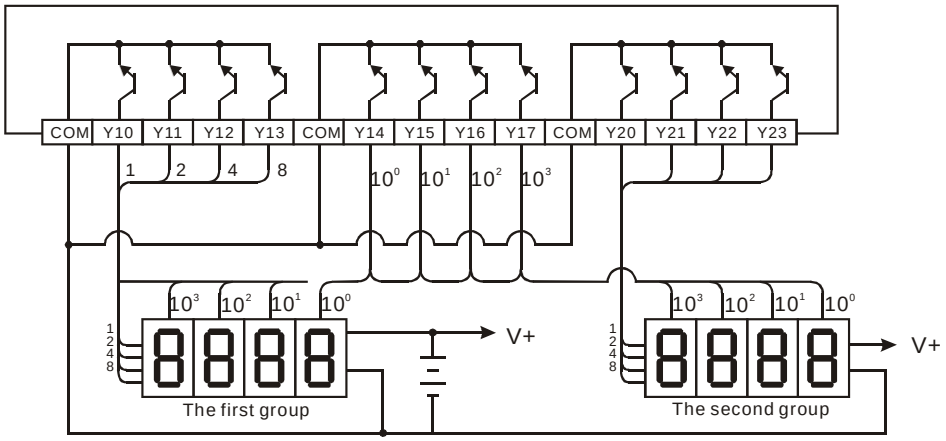
1. Range of **n**: 0 ~ 7. See remarks for more details.
2. For ES/EX/EC series MPU, the instruction can only be used once in the program. For EH3/SV3 series MPU, the instruction can be used twice in the program. For EC3-8K/SX series MPU, there is no limitation on the times of using the instruction, but only one instruction can be executed at a time.
3. For ES/EX/EC/SX/EC3-8K series MPU, the last digit of **D** should be 0 and it does not support E, F index register modification.
4. Flag: M1029 (SEGL execution completed)
5. This instruction occupies 8 or 12 continuous external input points starting from **D** for displaying 1 or 2 4-digit 7-segment display data and outputs of scanned signals. Every digit carries a 7-segment display drive (to convert the BCD codes into 7-segment display signal). The drive also carries latch control signals to retain the 7-segment display.
6. **n** decides there be 1 group or 2 groups of 4-digit 7-segment display and designates the polarity for the output.
7. When there is 1 group of 4-digit output, 8 output points will be occupied. When there are 2 groups of 4-digit output, 12 output points will be occupied.
8. When this instruction is being executed, the scan output terminals will circulate the scan in sequence. When the drive contact of the instruction goes from Off to On again, the scan output terminal will restart the scan again.

Program Example:

1. When X10 = On, this instruction starts to be executed, Y10 ~ Y17 construct a 7-segment display scan circuit. The value in D10 will be converted into BCD codes and sent to the first group 7-segment display. The value in D11 will be converted into BCD codes as well and sent to the second group 7-segment display. If the values in D10 and D11 exceed 9,999, operational error will occur.

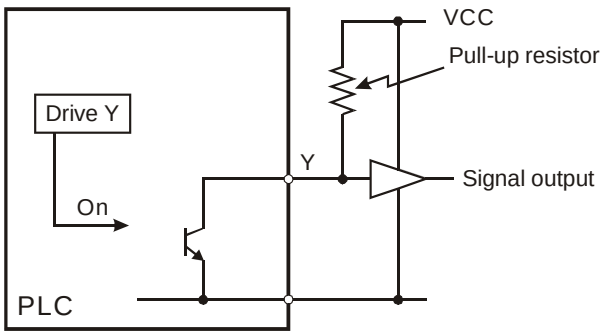
2. When X10 = On, Y14 ~ Y17 will circulate the scan automatically. Every cycle requires 12 scan period. Whenever a cycle is completed, M1029 will be On for a scan period.

3. When there is 1 group of 4-digit 7-segment display, **n** = 0 ~ 3.
   a) Connect the already decoded 7-segment display terminals 1, 2, 4, 8 in parallel an connect them to Y10 ~ Y13 on the PLC. Connect the latch terminals of each digit to Y14 ~ Y17 on the PLC.
   b) When X10 = On, the instruction will be executed and the content in D10 will be sent to the 7-segment displays in sequence by the circulation of Y14 ~ Y17.
4. When there is 2 groups of 4-digit 7-segment display, **n** = 4 ~ 7.
   a) Connect the already decoded 7-segment display terminals 1, 2, 4, 8 in parallel an connect them to Y20 ~ Y23 on the PLC. Connect the latch terminals of each digit to Y14 ~ Y17 on the PLC.
   b) The contents in D10 are sent to the first group 7-segment display. The contents in D11 are sent the the second group 7-segment display. If D10 = K1234 and D11 = K4321, the first group will display 1 2 3 4, and the second group will display 4 3 2 1.
5. Wiring of the 7-segment display scan output:



Remarks:

1. ES/EX/EC series MPU (V4.9 and above) supports this instruction but only supports 1 group of 4-digit 7-segment display and 8 points of output. This instruction can only be used once in the program. Range of **n**: 0 ~ 3.
2. **D** of ES/EX/EC series MPU can only designate Y0.
3. When this instruction is executed, the scan time has to be longer than 10ms. If the scan time is shorter than 10ms, please fix the scan time at 10ms.
4. **n** is for setting up the polarity of the transistor output and the number of groups of the 4-digit 7-segment display.
5. The output point must be a transistor module of NPN output type with open collector outputs. The output has to connect to a pull-up resistor to VCC (less than 30VDC). Therefore, when output point Y is On, the signal output will be in low voltage.



6. Positive logic (negative polarity) output of BCD code

BCD value				Y output (BCDcode)				Signal output			
b_3	b_2	b_1	b_0	8	4	2	1	A	B	C	D
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1	0

7. Negative logic (positive polarity) output of BCD code

BCD value				Y output (BCDcode)				Signal output			
b_3	b_2	b_1	b_0	8	4	2	1	A	B	C	D
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	0	1	0	0
0	1	0	1	1	0	1	0	0	1	0	1
0	1	1	0	1	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0	0	0
1	0	0	1	0	1	1	0	1	0	0	1

8. Scan latched signal display

Positive logic (negative polarity)		Negative logic (positive polarity)	
Y output (latch)	Output signal	Y output (latch)	Output signal
1	0	0	1

9. Settings of **n**:

Groups of 7-segment display	1 group				2 groups			
Y output of BCD code	＋		－		＋		－	
Scan latched signal display	＋	－	＋	－	＋	－	＋	－
n	0	1	2	3	4	5	6	7

＋: Positive logic (negative polarity) output　　－: Negative logic (positive polarity) output

10. The polarity of transistor output and the polarity of the 7-segment display input can be the same or different by the setting of **n**.

API	Mnemonic	Operands	Function
75	ARWS	S D1 D2 n	Arrow Switch

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												ARWS: 9 steps
D_1											*	*	*	*	*	
D_2		*														
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

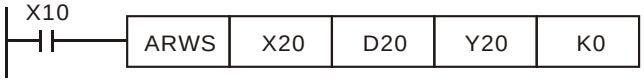
S: Start device for key input **D_1**: Device to be displayed in 7-segment display **D_2**: Start device for 7-segment display scan output **n**: Polarity setting of output signal and scan signal

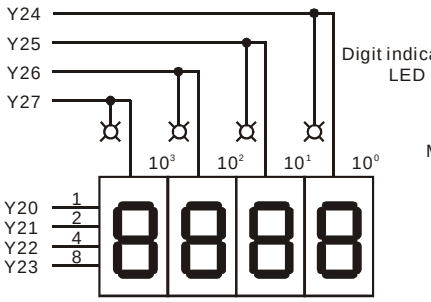
Explanations:

1. **S** will occupy 4 consecutive points.
2. Range of **n**: 0 ~ 3. See remarks of API 74 SEGL for more details.
3. There no limitation on the times of using this instruction in the program. However, only one instruction is allowed to be executed at a time.
4. **S** and **D_2** of SX series MPU do not support E, F index register modification, and **D_2** can only designate the devices whose last digit is 0 (e.g. Y0, Y10….)
5. See the specifications of each model for their range of use.
6. The output points designated by this instruction shall be transistor output.
7. When using this instruction, please fix the scan time, or place this instruction in the time interruption subroutine (I6□□ ~ I8□□).

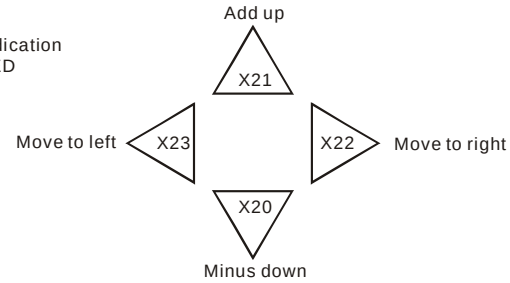
Program Example:

1. When this instruction is executed, X20 is defined as down key, X21 is defined as up key, X22 is defined as right key and X23 is defined as left key. The keys are used for setting up and displaying external set values. The set values (range: 0 ~ 9,999) are stored in D20.
2. When X10 = On, digit 10^3 will be the valid digit for setup. If you press the left key at this time, the valid digit will circulate as $10^3 \rightarrow 10^0 \rightarrow 10^1 \rightarrow 10^2 \rightarrow 10^3 \rightarrow 10^0$.
3. If you press the right key at this time, the valid digit will circulate as $10^3 \rightarrow 10^2 \rightarrow 10^1 \rightarrow 10^0 \rightarrow 10^3 \rightarrow 10^2$. During the circulation, the digit indicators connected Y24 ~ Y27 will also be On interchangeably following the circulation.
4. If you press the up key at this time, the valid digit will change as $0 \rightarrow 1 \rightarrow 2 \ldots \rightarrow 8 \rightarrow 9 \rightarrow 0 \rightarrow 1$. If you press the down key, the valid digit will change as $0 \rightarrow 9 \rightarrow 8 \ldots \rightarrow 1 \rightarrow 0 \rightarrow 9$. The changed value will also be displayed in the 7-segment display.





7-segment display for the 4-digit set value



The 4 switches are used for moving the digits and increasing/decreasing set values.

API	Mnemonic	Operands	Function
76	ASC	(S) (D)	ASCII Code Conversion

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ASC: 11 steps
S																
D											*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

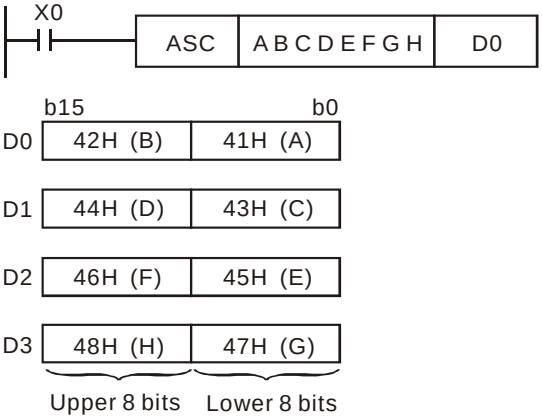
S: English letter to be converted into ASCII code **D**: Device for storing ASCII code

Explanations:

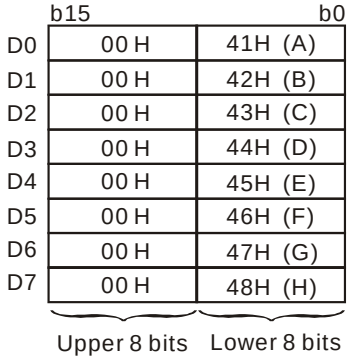
1. **S**: enter 8 Engligh letters by using WPLSoft on computer or enter ASCII code by HPP.
2. **S** in SX series MPU only accepts A, B, C, D, E, F, G, H, the 8 English character.
3. See the specifications of each model for their range of use.
4. Flag: M1161 (8/16 bit mode switch)
5. If the execution of this instruction is connected to a 7-segment display, the error message can be displayed by English letters.

Program Example:

1. When X0 = On, convert A ~ H into ASCII code and stored it in D0 ~ D3.



2. When M1161 = On, every ASCII code converted from the letters will occupy the lower 8 bits (b7 ~ b0) of a register. The upper 8 bits are invalid (filled by 0). One register stores a letter.



API	Mnemonic	Operands	Function
77	PR	(S) (D)	Print (ASCII Code Output)

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			PR: 5 steps
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

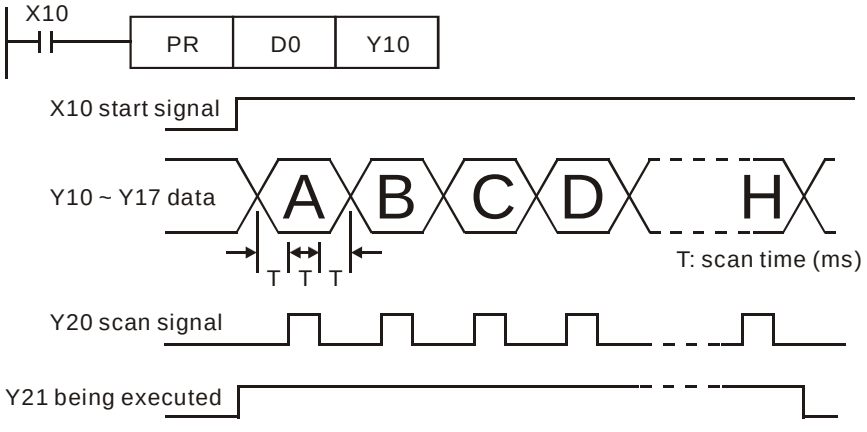
S: Device for storing ASCII code **D**: External ASCII code output points

Explanations:

1. **S** will occupy 4 consecutive points.
2. **D** will occupy 10 consecutive points.
3. This instruction can only be used twice in the program.
4. **D** in SX series MPU does not support E, F, index register modification.
5. See the specifications of each model for their range of use.
6. Flags: M1029 (PR execution completed); M1027 (number of PR outputs)
7. This instruction will output the ASCII codes in the 4 registers starting from **S** from the output devices in the order designated in **D**.

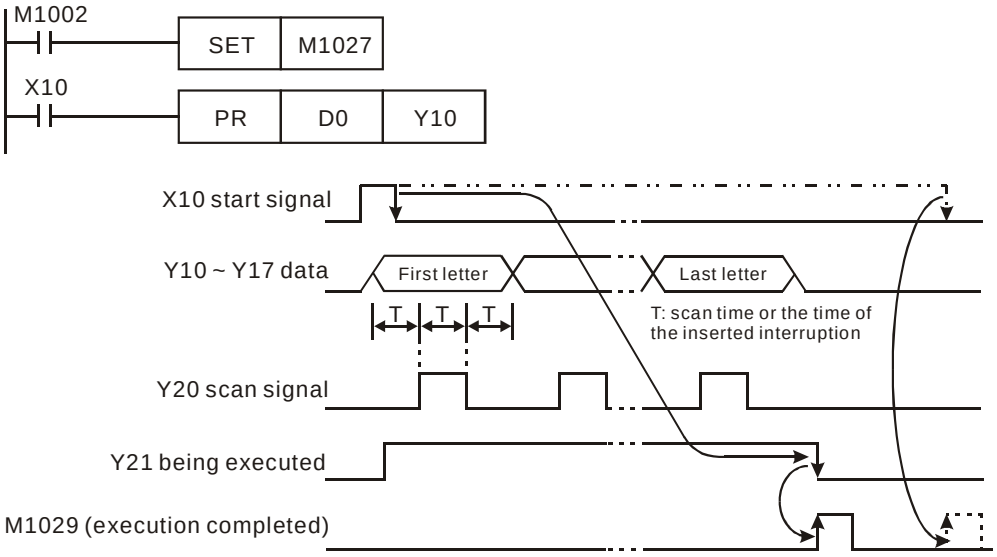
Program Example 1:

1. Use API 76 ASC to convert A ~ H into ASCII codes and store them in D0 ~ D3 and use this instruction to output the codes in sequence.
2. When M1027 = Off and X10 goes On, the instruction will be executed. Designate Y10 (low bits) ~ Y17 (high bits) as the data output points and Y20 for scan signals. Designate Y21 for the monitor signals during the execution. In this mode, you can execute an output for 8 letters in sequence. During the output, if the drive contact goes Off, the data output will stop immediately and all the outputs will go Off.
3. During the execution of the instruction, when X10 goes Off, all the data output will be interrupted. When X10 is On again, the output will be restarted.



Program Example 2:

1. PR instruction is for outputing a string of 8 bits. When the special auxiliary relay M1027 = Off, PR is able to execute an output of maximum 8 letters in string. When M1027 = On, PR is able to execute an output of 1 ~ 16 letters in string.
2. When M1027 = On and X10 goes from Off to On, the instruction will be executed. Designate Y10 (low bits) ~ Y17 (high bits) as the data output points and Y20 for scan signals. Designate Y21 for the monitor signals during the execution. In this mode, you can execute an output for 16 letters in sequence. During the output, if the drive contact goes Off, the data output will stop after it is completed.
3. When the string encounters 00H (NUL), the string output will finish. The letters coming after it will not be processed.
4. When X10 goes from On to Off, the data output will automatically stop after one cycle. If X10 keeps being On, M1029 will not be enabled.



Remarks:

1. Please use transistor output for the output designated by this instruction.
2. When using this instruction, please fix the scan time or place this instruction in a timed interruption subroutine.

API	Mnemonic			Operands	Function
78	D	FROM	P	m1 m2 D n	Read CR Data in Special Modules

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
m_1					*	*							*			FROM, FROMP: 9 steps
m_2					*	*							*			DFROM, DFROMP: 17 steps
D								*	*	*	*	*	*	*	*	
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

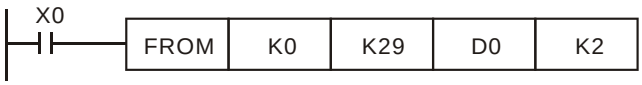
m_1: No. of special module **m_2**: CR# in special module to be read **D**: Device for storing read data **n**: Number of data to be read at a time

Explanations:

1. Range of **m_1** (16-bit and 32-bit): for SX: 0 ~ 7, for EH3/SV2 :0 ~ 255, for SV2: 0 ~ 107.
2. Range of **m_2** (16-bit and 32-bit): for SX: 0 ~ 48, for EH3/SV2: 0 ~ 499.
3. Range of **n**:
   a) 16-bit: for SX: 1 ~ (49 – **m_2**), for EH3/SV2: 1 ~ (500 – **m_2**).
   b) 32-bit: for SX: 1 ~ (49 – **m_2**)/2, for EH3/SV2: 1 ~ (500 – **m_2**)/2.
4. This instruction is for reading the data in the CR in special modules.
5. The 16-bit instruction can designate **D** = K1 ~ K4; the 32-bit instruction can designate **D** = K1 ~ K8.
6. See application examples in API 79 TO insitruction for how to calculate the No. where the special module is located.

Program Example:

1. Read CR#29 of special module No.0 into D0 and CR#30 into D1. Only 2 groups of data is read at a time (n = 2).
2. When X0 = On, the instruction will be executed. When X0 = Off, the instruction will not be executed and the data read will not be changed.



API	Mnemonic			Operands	Function
79	D	TO	P	m1 m2 S n	Write CR Data into Special Modules

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
m_1					*	*							*			TO, TOP: 9 steps
m_2					*	*							*			DTO, DTOP: 17 steps
S					*	*	*	*	*	*	*	*	*	*	*	
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

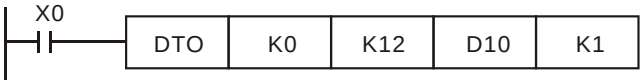
m_1: No. of special module **m_2**: CR# in special module to be written **S**: Data to be written in CR **n**: Number of data to be written at a time

Explanations:

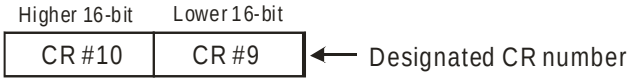
1. Range of **m_1** (16-bit and 32-bit): for SX: 0 ~ 7, for EH3/SV2: 0 ~ 255, for SV: 0 ~ 107.
2. Range of **m_2** (16-bit and 32-bit): for SX: 0 ~ 48, for EH3/SV2: 0 ~ 499.
3. Range of **n**:
   a) 16-bit: for SX: 1 ~ (49 – **m_2**), for EH3/SV2: 1 ~ (500 – **m_2**).
   b) 32-bit: for SX: 1 ~ (49 – **m_2**)/2, for EH3/SV2: 1 ~ (500 – **m_2**)/2.
4. This instruction is for writing the data into the CR in special modules.
5. The 16-bit instruction can designate **S** = K1 ~ K4; the 32-bit instruction can designate **S** = K1 ~ K8.

Program Example:

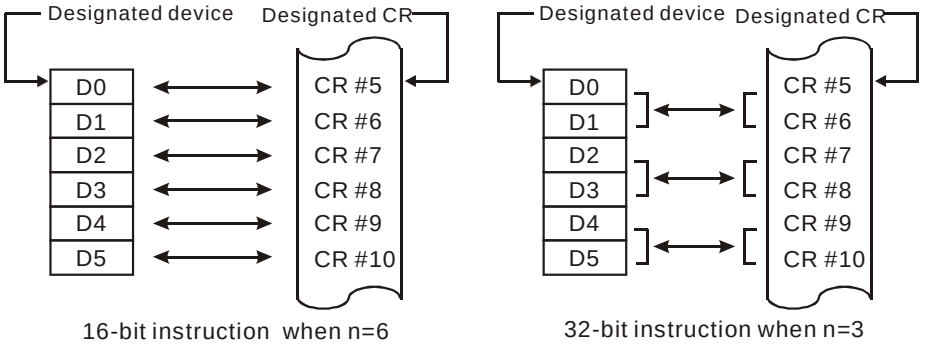
1. Use 32-bit instruction DTO to write the content in D11 and D10 into CR#13 and CR#12 of special module No.0. Only 1 group of data is written in at a time (n = 1).
2. When X0 = On, the instruction will be executed. When X0 = Off, the instruction will not be executed and the data written will not be changed.



3. Operand rules
   a) **m_1**: The No. of special modules connected to PLC MPU. No. 0 is the module closest to te MPU. Maximum 8 modules are allowed to connected to a PLC MPU and they will not occupy any I/O points.
   b) **m_2**: CR#. CR (control register) is the n 16-bit memories built in the special module, numbered in decimal as #0 ~ #n. All operation status and settings of the special module are contained in the CR.
   c) FROM/TO instruction is for reading/writing 1 CR at a time. DFROM/DTO instruction is for reading/writing 2 CRs at a time.



d) Number of groups “n” to be transmitted: n = 2 in 16-bit instructions and n = 1 in 32-bit instructions mean the same.



FROM/TO Application Example 1:

Adjust the A/D conversion curve of DVP04AD. Set the OFFSET value of CH1 as 0V (= $K0_{LSB}$) and GAIN value as 2.5V (= $K2{,}000_{LSB}$).

1. Write H0 to CR#1 of anlog input module No. 0 and set CH1 as mode 0 (voltage input: -10V ~ +10V).
2. Write H0 to CR#33 and allow OFFSET/GAIN tuning in CH1 ~ CH4.
3. When X0 goes from Off to On, write the OFFSET value $K0_{LSB}$ into CR#18 and the GAIN value $K2{,}000_{LSB}$ into CR#24.

FROM/TO Application Example 2:

Adjust the A/D conversion curve of DVP04AD. Set the OFFSET value of CH2 as 2mA (= $K400_{LSB}$) and GAIN value as 18mA (= $K3{,}600_{LSB}$).

1. Write H18 to CR#1 of anlog input module No. 0 and set CH2 as mode 3 (current input: -20mA ~ +20mA).
2. Write H0 to CR#33 and allow OFFSET/GAIN tuning in CH1 ~ CH4.

3. When X0 goes from Off to On, write the OFFSET value K400 $_{LSB}$ into CR#19 and the GAIN value K3,600 $_{LSB}$ into CR#25.

FROM/TO Application Example 3:

Adjust the D/A conversion curve of DVP02DA. Set the OFFSET value of CH2 as 0mA (= K0$_{LSB}$) and GAIN value as 10mA (= K1,000$_{LSB}$).

1. Write H18 to CR#1 of anlog output module No. 1 and set CH2 as mode 3 (current output: 0mA ~ +20mA).
2. Write H0 to CR#33 and allow OFFSET/GAIN tuning in CH1 and CH2.
3. When X0 goes from Off to On, write the OFFSET value K0 $_{LSB}$ into CR#22 and the GAIN value K1,000 $_{LSB}$ into CR#28.

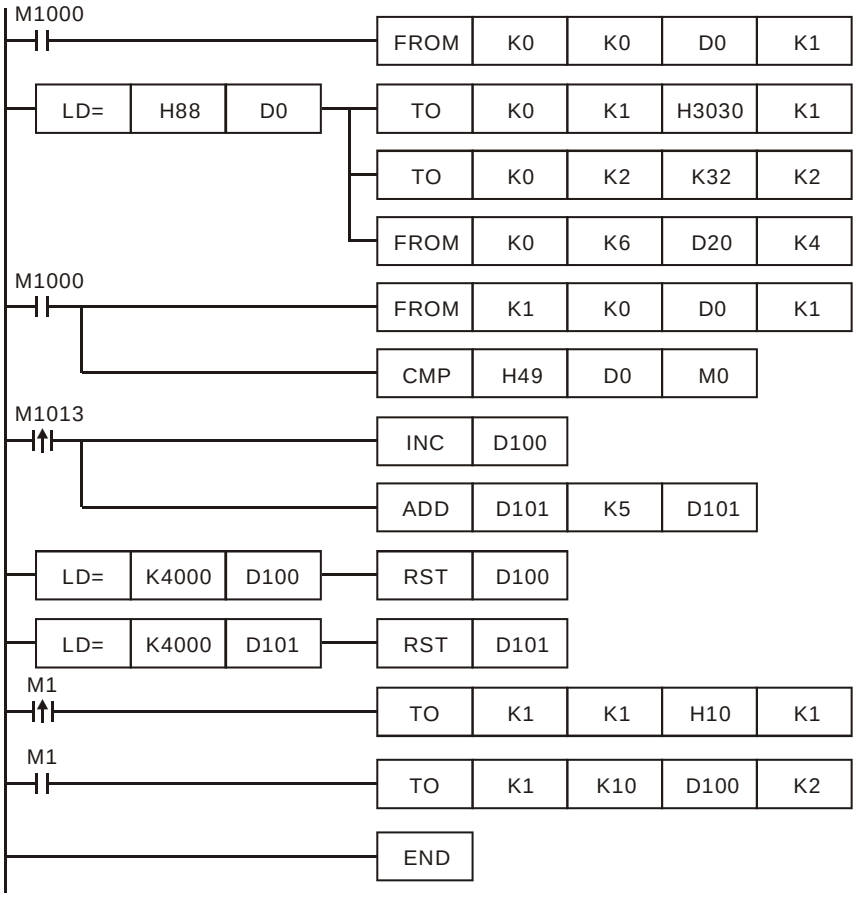
FROM/TO Application Example 4:

Adjust the D/A conversion curve of DVP02DA. Set the OFFSET value of CH2 as 2mA (= K400$_{LSB}$) and GAIN value as 18mA (= K3,600$_{LSB}$).

1. Write H10 to CR#1 of anlog output module No. 1 and set CH2 as mode 2 (current output: +4mA ~ +20mA).
2. Write H0 to CR#33 and allow OFFSET/GAIN tuning in CH1 and CH2.
3. When X0 goes from Off to On, write the OFFSET value K400 $_{LSB}$ into CR#23 and the GAIN value K3,600 $_{LSB}$ into CR#29.

FROM/TO Application Example 5:

When DVP04AD is used with DVP02DA



1. Read CR#0 of the extension module No. 0 and see if it is DVP04AD: H88.
2. If D0 = H88, set the input modes: (CH1, CH3) mode 0, (CH2, CH4) mode 3.
3. Set the average times in CH1 and CH2 from CR#2 and CR#3 as K32.
4. Read the average of input signals at CH1 ~ CH4 from CR#6 ~ CR#9 and store the 4 data in D20 ~ D23.
5. Read CR#0 of the extension module No. 1 and see if it is DVP02DA-S: H49.
6. D100 increases K1 and D101 increases K5 every second.
7. When D100 and D101 reach K4,000, they will be cleared as 0.
8. See if the model is DVP02DA-S when M1 = On. If so, set up output mode: CH1 in mode 0 and CH2 is mode 2.
9. Write the output settings of D100 and D101 into CR#10 and CR#11. The analog output will change by the changes in D100 and D101.

API	Mnemonic	Operands	Function
80	RS	S m D n	Serial Communication Instruction

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			RS: 9 steps
m					*	*							*			
D													*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Start device for the data to be transmitted **m**: Length of data to be transmitted **D**: Start device for receiving data **n**: Length of data to be received

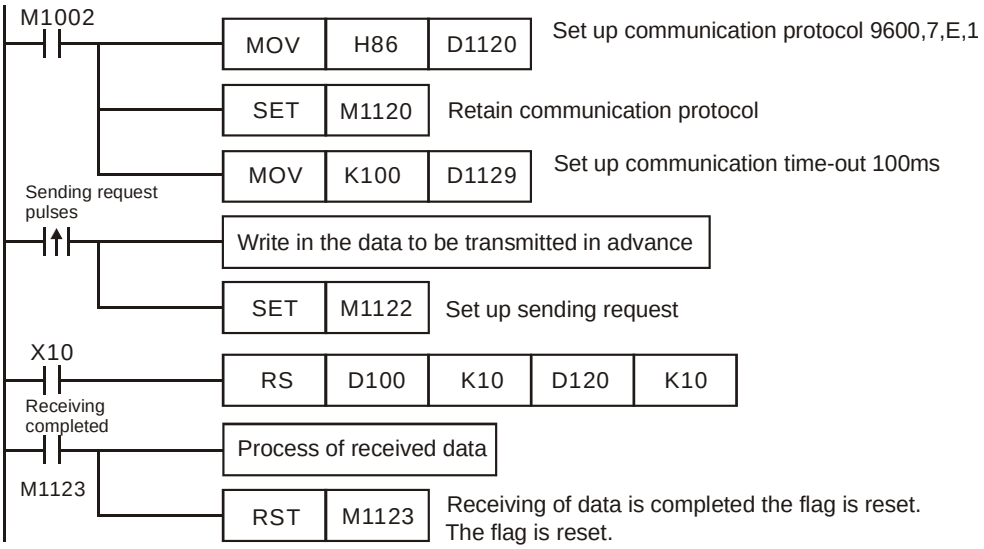
Explanations:

1. Range of **m**: 0 ~ 256
2. Range of **n**: 0 ~ 256
3. See the specifications of each model for their range of use.
4. ES series MPU does not support E, F index register modification.
5. The instruction RS supports COM1 (RS-232), COM2 (RS-485), and COM3 (a communication card) in a DVP-EH3/SV2 series PLC. (COM1 only supports DVP-EH3/SV2 series PLCs. COM3 in DVP-EH3 series PLCs is only applicable to the communication cards DVP-F232, DVP-F485 and DVP-F422.)
6. This instruction is a handy instruction exclusively for MPU to use RS-485 serial communication interface. The user has to pre-store word data in **S** data register, set up data length **m** and the data receiving register **D** and received data length **n**. If E, F index registers are used to modify **S** and **D**, the user cannot change the set values of E and F when the instruction is being executed; otherwise errors may cause in data writing or reading.
7. Designate **m** as K0 if you do not need to send data. Designate **n** as K0 if you do not need to receive data.
8. There is no limitation on the times of using this instruction in the program, but only one instruction is allowed to be executed at a time.
9. During the execution of RS instruction, changing the data to be transmitted will be invalid.
10. If the peripheral devices, e.g. AC motor drive, are equipped with RS-485 serial communication and its communication format is open, you can use RS instruction to design the program for the data transmission between PLC and the peripheral device.
11. If the communication format of the peripheral device is Modbus, DVP series PLC offers handy communication instructions API 100 MODRD, API 101 MODWR, and API 150 MODRW, to work with the device. See explanations of the instructions in this application manual.
12. If a Delta VFD series AC motor drive is used, the PLC provides the convenience instructions API 102 FWD, API 103 REV, API 104 STOP, API 105 RDST, and API 106 RSTEF. If a Delta ASD series servo drive is used, the PLC provides the convenience instruction API 206 ASDRW. If a Delta DMV series product is used, the PLC provides the convenience instruction API 295 DMVRW.
13. For the special auxiliary relays M1120 ~ M1161 and special data registers D1120 ~ D1131 relevant to RS-485

communication, see remarks for more details.

Program Example 1:

1. Use COM2 on the PLC to carry out RS-485 communication.
2. Write the data to be transmitted in advance into registers starting from D100 and set M1122 (sending request flag) as On.
3. When X10 = On, RS instruction will be executed and PLC will start to wait for the sending and receiving of data. D100 starts to continuousl send out 10 data and when the sending is over, M1122 will be automatically reset to Off (DO NOT use the program to execute RST M1122). After 1ms of waiting, PLC will start to receive the 10 data. Store the data in consecutive registers starting from D120.
4. When the receiving of data is completed, M1123 will automatically be On. After the program finishes processing the received data, M1123 has to be reset to Off and the PLC will start to wait for the sending and receiving of data again. DO NOT use the program to continuously execute RST M1123.



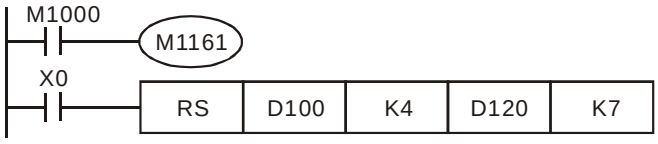
Program Example 2:

Use COM2 on the PLC to carry out RS-485 communication.

Switching between 8-bit mode (M1161 = On) and 16-bit mode (M1161 = Off)

1. 8-bit mode:

   The head code and tail code of the data are set up by M1126 and M1130 together with D1124 ~ D1126. When PLC is executing RS instruction, the head code and tail code set up by the user will be sent out automatically. M1161 = On indicates PLC in 8-bit conversion mode. The 16-bit data will be divided into the higher 8 bits and lower 8 bits. The higher 8 bits are ignored and only the lower 8 bits are valid for data transmission.



Sending data: (PLC -> external equipment)

STX	D100L	D101L	D102L	D103L	ETX1	ETX2

Head code

(S) source data register, starting from the lower 8 bits of D100

(m) length = 4

Tail code 1

Tail code 2

Receiving data: (External equipment -> PLC)

D120L	D121L	D122L	D123L	D124L	D125L	D126L

Head code

(S) received data register, starting from the lower 8 bits of D120

(n) length = 7

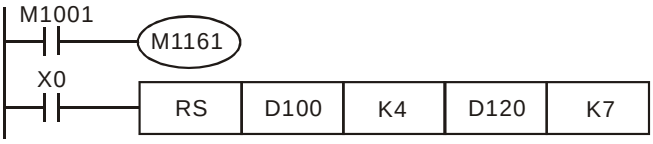
Tail code 1

Tail code 2

When receiving data, PLC will receive the head code and tail code of the data from the external equipment; therefore, the user has to be aware of the setting of data length **n**.

2. 16-bit mode:

The head code and tail code of the data are set up by M1126 and M1130 together with D1124 ~ D1126. When PLC is executing RS instruction, the head code and tail code set up by the user will be sent out automatically. M1161 = Off indicates PLC in 16-bit conversion mode. The 16-bit data will be divided into the higher 8 bits and lower 8 bits for data transmission.



Sending data: (PLC -> external equipment)

STX	D100L	D100L	D101L	D101L	ETX1	ETX2

Head code

(S) source data register, starting from the lower 8 bits of D100

(m) length = 4

Tail code 1

Tail code 2

Receiving data: (External equipment -> PLC)

D120L	D120H	D121L	D121H	D122L	D122H	D123L

Head code

(D) received data register, starting from the lower 8 bits of D120

(n) length = 7

Tail code 1

Tail code 2

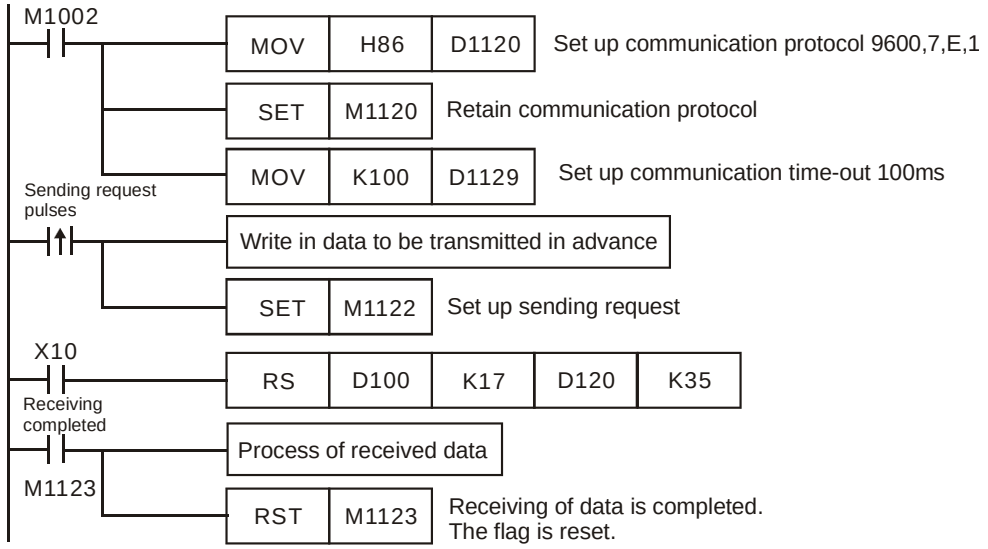
When receiving data, PLC will receive the head code and tail code of the data from the external equipment; therefore, the user has to be aware of the setting of data length **n**.

Program Example 3:

Use COM2 on the PLC to carry out RS-485 communication.

Connect PLC to VFD-B series AC motor drives (AC motor drive in ASCII Mode; PLC in 16-bit mode and M1161 = Off).

Write in the 6 data starting from parameter address H2101 in VFD-B in advance as the data to be transmitted.



PLC ⇨ VFD-B, PLC sends **": 01 03 2101 0006 D4 CR LF "**

VFD-B ⇨ PLC, PLC receives **": 01 03 0C 0100 1766 0000 0000 0136 0000 3B CR LF "**

Registers for sent data (PLC sends out message)

Register	Data		Explanation	
D100 low	': '	3A H	STX	
D100 high	'0'	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D101 low	'1'	31 H	ADR 0	
D101 high	'0'	30 H	CMD 1	Instruction code: CMD (1,0)
D102 low	'3'	33 H	CMD 0	
D102 high	'2'	32 H	Start data address	
D103 low	'1'	31 H		
D103 high	'0'	30 H		
D104 low	'1'	31 H		
D104 high	'0'	30 H	Number of data (counted by words)	
D105 low	'0'	30 H		
D105 high	'0'	30 H		
D106 low	'6'	36 H		
D106 high	'D'	44 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D107 low	'4'	34 H	LRC CHK 0	
D107 high	CR	D H	END	
D108 low	LF	A H		

Registers for received data (VFD-B responds with messages)

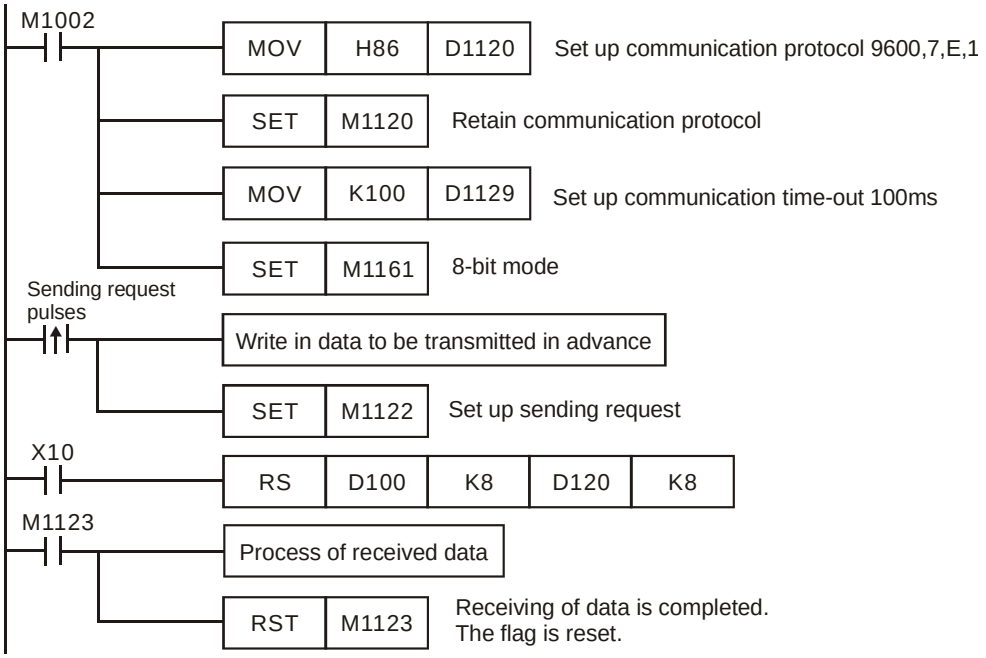
Register	Data		Explanation
D120 low	': '	3A H	STX
D120 high	'0'	30 H	ADR 1
D121 low	'1'	31 H	ADR 0
D121 high	'0'	30 H	CMD 1
D122 low	'3'	33 H	CMD 0
D122 high	'0'	30 H	Number of data (counted by byte)
D123 low	'C'	43 H	
D123 high	'0'	30 H	Content of address 2101 H
D124 low	'1'	31 H	
D124 high	'0'	30 H	
D125 low	'0'	30 H	
D125 high	'1'	31 H	Content of address 2102 H
D126 low	'7'	37 H	
D126 high	'6'	36 H	
D127 low	'6'	36 H	
D127 high	'0'	30 H	Content of address 2103 H
D128 low	'0'	30 H	
D128 high	'0'	30 H	
D129 low	'0'	30 H	
D129 high	'0'	30 H	Content of address 2104 H
D130 low	'0'	30 H	
D130 high	'0'	30 H	
D131 low	'0'	30 H	
D131 high	'0'	30 H	Content of address 2105 H
D132 low	'1'	31 H	
D132 high	'3'	33 H	
D133 low	'6'	36 H	
D133 high	'0'	30 H	Content of address 2106 H
D134 low	'0'	30 H	
D134 high	'0'	30 H	
D135 low	'0'	30 H	
D135 high	'3'	33 H	LRC CHK 1
D136 low	'B'	42 H	LRC CHK 0
D136 high	CR	D H	END
D137 low	LF	A H	

Program Example 4:

Use COM2 on the PLC to carry out RS-485 communication.

Connect PLC to VFD-B series AC motor drives (AC motor drive in RTU Mode; PLC in 16-bit mode and M1161 = On).

Write in H12 to parameter address H2000 in VFD-B in advance as the data to be transmitted.



PLC ⇨ VFD-B, PLC sends: **01 06 2000 0012 02 07**

VFD-B ⇨ PLC, PLC receives: **01 06 2000 0012 02 07**

Registers for sent data (PLC sends out messages)

Register	Data	Explanation
D100 low	01 H	Address
D101 low	06 H	Function
D102 low	20 H	Data address
D103 low	00 H	
D104 low	00 H	Data content
D105 low	12 H	
D106 low	02 H	CRC CHK Low
D107 low	07 H	CRC CHK High

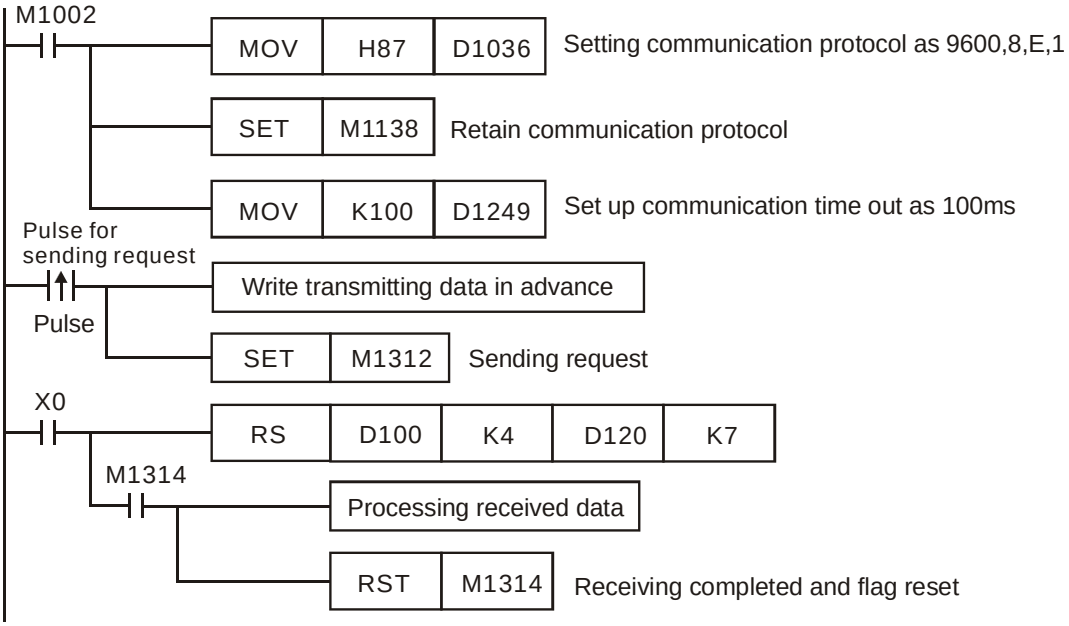
Registers for received data (VFD-B responds with messages)

Register	Data	Explanation
D120 low	01 H	Address
D121 low	06 H	Function
D122 low	20 H	Data address
D123 low	00 H	
D124 low	00 H	Data content
D125 low	12 H	
D126 low	02 H	CRC CHK Low
D127 low	07 H	CRC CHK High

Program Example 5: COM1 RS-232

1. Only 8-bit mode is supported. Communication format and speed are specified by lower 8 bits of D1036.

2. STX/ETX setting function (M1126/M1130/D1124~D1126) is not supported.
3. High byte of 16-bit data is not available. Only low byte is valid for data communication.
4. Write the data to be transmitted in advance into registers starting from D100 and set M1312 (COM1 sending request) as ON
5. When X0 = ON, RS instruction executes and PLC is ready for communication. D100 will then start to send out 4 data continuously. When data sending is over, M1312 will be automatically reset. (DO NOT apply RST M1312 in program). After approximate 1ms, PLC will start to receive 7 data and store the data in 7 consecutive registers starting from D120.
6. When data receiving is completed, M1314 will automatically be ON. When data processing on the received data is completed, M1314 has to be reset (OFF) and the PLC will be ready for communication again. However, DO NOT continuously execute RST M1314.



Sending data: (PLC→External equipment)

D100L	D101L	D102L	D103L

(S) Source data register, starting from lower 8 bits of D100

(m) Length = 4

Receving data: (External equipment→PLC)

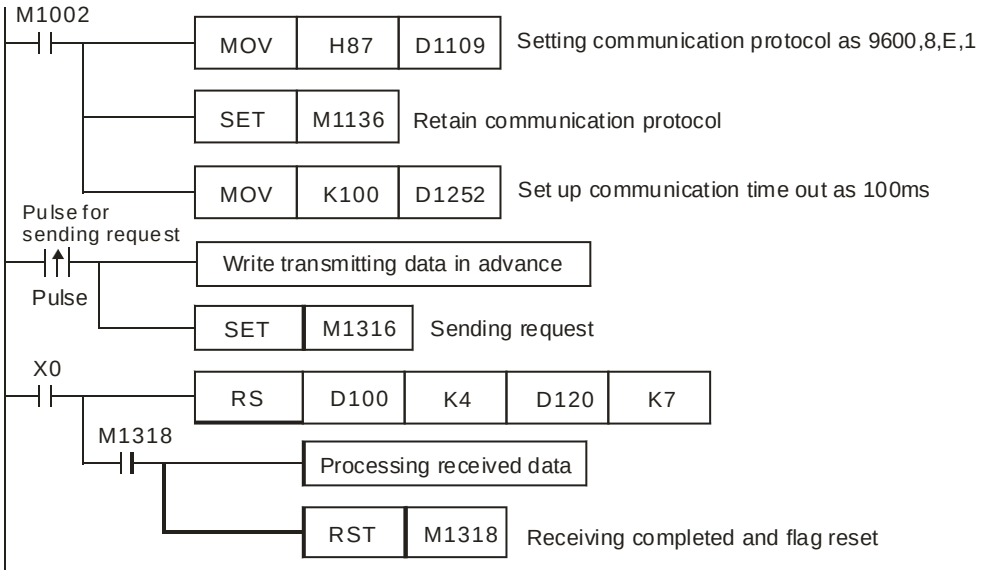
D120L	D121L	D122L	D123L	D124L	D125L	D126L

(S) Registers for received data, starting from lower 8 bits of D120

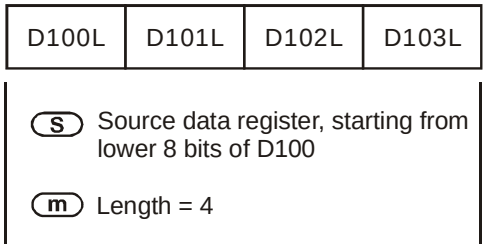
(n) Length = 7

Program Example 6: COM3 RS-485 or RS-232

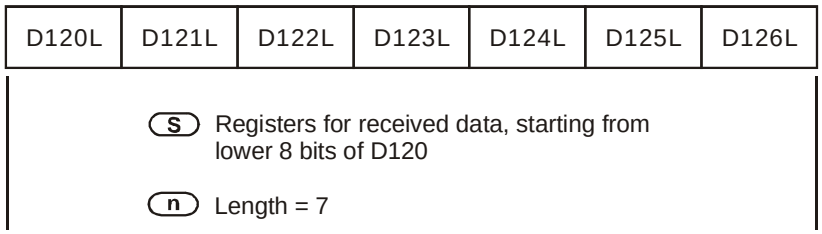
1. Only 8-bit mode is supported. Communication format and speed are specified by lower 8 bits of D1109.
2. STX/ETX setting function (M1126/M1130/D1124~D1126) is not supported.
3. High byte of 16-bit data is not available. Only low byte is valid for data communication.
4. Write the data to be transmitted in advance into registers starting from D100 and set M1316 (COM3 sending request) as ON
5. When X0 = ON, RS instruction executes and PLC is ready for communication. D100 will then start to send out 4 data continuously. When data sending is over, M1316 will be automatically reset. (DO NOT apply RST M1316 in program). After approximate 1ms, PLC will start to receive 7 data and store the data in 7 consecutive registers starting from D120.
6. When data receiving is completed, M1318 will automatically be ON. When data processing on the received data is completed, M1318 has to be reset (OFF) and the PLC will be ready for communication again. However, DO NOT continuously execute RST M1318.



Sending data: (PLC→External equipment)



Receving data: (External equipment→PLC)



Remarks:

1. **PLC COM1 RS-232:** Associated flags (Auxiliary relays) and special registers (Special D) for communication instructions RS / MODRD

Flag	Function	Action
M1138	COM1 retain communication settings. Communication settings will be reset (changed) according to the content in D1036 after every scan cycle. Users can set ON M1138 if the communication protocol requires to be retained. When M1138 = ON, communication settings will not be reset (changed) when communication instructions are being processed, even if the content in D1036 is changed.	User sets and resets
M1139	COM1 ASCII / RTU mode selection, ON: RTU mode, OFF: ASCII mode.	User sets and resets
M1312	COM1 sending request. Before executing communication instructions, users need to set M1312 to ON by trigger pulse, so that the data sending and receiving will be started. When the communication is completed, PLC will reset M1312 automatically.	User sets and system resets
M1313	COM1 data receiving ready. When M1313 is ON, PLC is ready for data receiving	System
M1314	COM1 Data receiving completed.	System sets and user resets
M1315	COM1 receiving error. M1315 will be set ON when errors occur and the error code will be stored in D1250.	System sets and user resets

Special register	Function
D1036	COM1 (RS-232) communication protocol. Refer to the following table in point 4 for protocol setting.
D1167	The specific end word to be detected for RS instruction to execute an interruption request (I140) on COM1 (RS-232). When the character received is equal to the low byte of D1167, the interrupt I140 is triggered. Supported communication instructions: RS
D1121	COM1 (RS-232)/COM2 (RS-485) communication address when COM1/COM2 functions as a slave station.

Special register	Function
D1249	COM1 (RS-232) communication time-out setting (unit: ms). If users set up time-out value in D1249 and the data receiving time exceeds the time-out value, M1315 will be set ON and the error code K1 will be stored in D1250. M1315 has to be reset manually when time-out status is cleared.
D1250	COM1 (RS-232) communication error code. Supported communication instructions: MODRW

2. **PLC COM2 RS-485:** Associated flags (Auxiliary relays) and special registers (Special D) for communication instructions RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW.

Flag	Function	Action
M1120	For retain the communication setting. After the first program scan is completed, the communication setting will be reset according to the setting in the special data register D1120. When the second program scan starts and RS instruction is being executed, the communication settings will all be reset according to the settings in D1120. If your communication protocol is fixed, you can set M1120 to On and the communication protocol will not be reset whenever RS/MODRD/MODWR/FWD/REV/STOP/RDST/RSTEF/MODRW instruction is executed. In this case, even the settings in D1120 are modified, the communication protocol will not be changed.	Set up and reset by the user.
M1121	Off when the RS-485 communication data is being transmitted.	By the system.
M1122	Sending request. When you need to send out or receive data by RS/MODRD/MODWR/FWD/REV/STOP/RDST/RSTEF/MODRW instructions, you have to set M1122 to On by a pulse instruction. When these instructions start to execute, PLC will start to send out or receive data. When the data transmission is completed, M1122 will be reset automatically.	Set up by the user; reset automatically by the system.
M1123	Receiving is completed. When the execution of RS/MODRD/MODWR/FWD/REV/STOP/RDST/RSTEF/MODRW instructions is completed, M1123 will be set to On. You can process the data received when M1123 is On in the program. You have to reset M1123 to Off when the process of received data is completed.	Set up automatically by the system; reset by the user.
M1124	Waiting for receiving. On when PLC is waiting for receiving data.	By the system.

Flag	Function	Action
M1125	Receiving status cleared. When M1125 = On, the waiting for receiving status of PLC will be cleared. You have to reset M1125 to Off after the staus is cleared.	Set up and reset by the user.
M1126	User/system defined STX/ETX selection of RS instruction (see the next table for details.)	
M1130	User/system defined STX/ETX selection of RS instruction (see the next table for details.)	
M1127	Data transmission is completed for communication instructions (RS instruction not included)	Set up automatically by the system; reset by the user.
M1128	Data being sent/received indication	By the systme.
M1129	Receiving time-out. If you already set up a communication time-out in D1129 and the data have not been received completey when the time-out set is reached, M1129 will be On. You have to reset M1129 to Off after the problem is solved.	Set up automatically by the system; reset by the user.
M1131	On when the data are converted into hex of MODRD/RDST/MODRW instructions when in ASCII mode; otherwise, M1131 is Off.	By the system
M1140	Data receiving error of MODRD/MODWR/MODRW instructions	
M1141	Parameter error of MODRD/MODWR/MODRW instructions	
M1142	Data receiving error of VFD-A handy commands	
M1143	ASCII/RTU mode selection (used with MODRD/MODWR/MODRW instructions). On = RTU; Off = ASCII	Set up and reset by the user.
M1161	8/16-bit mode selection. On = 8-bit; Off = 16-bit	

Special register	Function
D1038	Delay time of data response when PLC is SLAVE in RS-485 communication, Range: 0~10,000. (Unit: 0.1ms).
D1050~D1055	After MODRD / RDST is executed, the PLC will automatically convert the ASCII data in D1070~D1085 into Hex data and stores the 16-bit Hex data into D1050~D1055
D1070~D1085	When the PLC's RS-485 communication instruction receives feedback signals, the data will be saved in the registers D1070~D1085 and then converted into Hex in other registers. The RS instruction is not supported.
D1089~D1099	When the PLC's RS-485 communication instruction sends out data, the data will be stored in D1089~D1099. Users can check the sent data in these registers. RS instruction is not supported
D1120	RS-485 communication protocol. Refer to the following table in point 4 for protocol setting.
D1121	PLC communication address when PLC is a slave.

Special register	Function
D1122	Residual number of words of transmitting data.
D1123	Residual number of words of the receiving data.
D1124	Definition of start character (STX) Refer to the following table in point 3 for the setting.
D1125	Definition of first ending character (ETX1) in the RS instruction Refer to the following table in point 3 for the setting.
D1126	Definition of second ending character (ETX2) in the RS instruction Refer to the following table in point 3 for the setting.
D1129	Communication time-out setting (unit: ms) If the value in D1129 is 0, there is no communication timeout. If the setting value in D1129 is greater than 0, and the first character is not received when RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW is executed, or the time interval between two characters is greater than the setting value, the PLC will set M1129 to ON. Users can deal with the communication timeout by means of the flag. After the users deal with the communication timeout, they have to clear M1129.
D1130	Error code returning from Modbus.
D1168	The specific end word to be detected for RS instruction to execute an interruption request (I150) If the character received is equal to the low byte in D1168, the interrupt I150 will be triggered.
D1169	The specific communication length to be detected for RS instruction to execute an interruption request (I160) If the length of the data received is equal to the low byte in D1169, the interrupt I160 will be triggered. If the value in D1169 is 0, the interrupt is not triggered.
D1256~D1295	When the RS-485 communication instruction MODRW is executed, the command sent is stored in D1256 ~ D1295. Users can see the contents of the registers to check whether the command is correct.
D1296~D1311	The PLC automatically converts the received ASCII data into hex data. Supported communication instruction: MODRW

3. **PLC COM3 RS-485:** Associated flags (Auxiliary relays) and special registers (Special D) for communication instructions RS / MODRW and FWD / REV / STOP / RDST / RSTEF when M1177 = ON.

Flag	Function	Action
M1136	COM3 retain communication settings. Communication settings will be reset (changed) according to the content in D1109 after every scan cycle. Users can set ON M1136 if the communication protocol requires to be retained. When M1136 = ON, communication settings will not be reset (changed) when communication instructions are being processed, even if the content in D1109 is changed	User sets and resets
M1320	COM3 ASCII / RTU mode selection. ON : RTU mode, OFF: ASCII mode.	
M1316	COM3 sending request. Before executing communication instructions, users need to set M1316 to ON by trigger pulse, so that the data sending and receiving will be started. When the communication is completed, PLC will reset M1316 automatically.	User sets, system resets
M1317	Data receiving ready. When M1317 is ON, PLC is ready for data receiving.	System sets
M1318	COM3 data receiving completed.	System sets, user resets
M1319	COM3 data receiving error. M1319 will be set ON when errors occur and the error code will be stored in D1253	System sets, user resets

Special register	Function
D1038	Delay time of data response when PLC is SLAVE in COM3 RS-485 communication, Range: 0~10,000. (unit: 0.1ms).
D1109	COM3 (RS-485) communication protocol. Refer to the following table in point 4 for protocol setting.
D1169	The specific end word to be detected for RS instruction to execute an interruption request (I160) on COM3 (RS-485) If the character received is equal to the low byte in D1169, the interrupt I160 will be triggered. Supported communication instructions: RS
D1252	COM3 (RS-485) Communication time-out setting (ms). If users set up time-out value in D1252 and the data receiving time exceeds the time-out value, M1319 will be set ON and the error code K1 will be stored in D1253. M1319 has to be reset manually when time-out status is cleared.
D1253	COM3 (RS-485) communication error code
D1255	COM3 (RS-485) PLC communication address when PLC is a slave.

4. Corresponding table between COM ports and communication settings/status.

	COM1	COM2	COM3	Function Description
Protocol setting	M1138	M1120	M1136	Retain communication setting
	M1139	M1143	M1320	ASCII/RTU mode selection
	D1036	D1120	D1109	Communication protocol
	D1121	D1121	D1255	PLC communication address
Sending request	-	M1161	-	8/16 bit mode selection
	-	M1121	-	Indicate transmission status
	M1312	M1122	M1316	Sending request
	-	M1126	-	Set STX/ETX as user/system defined. (RS)
	-	M1130	-	Set STX/ETX as user/system defined. (RS)
	-	D1124	-	Definition of STX (RS)
Sending request	-	D1125	-	Definition of ETX1 (RS)
	-	D1126	-	Definition of ETX2 (RS)
	D1249	D1129	D1252	Communication timeout setting (ms)
	-	D1122	-	Residual number of words of transmitting data
	-	D1256 ~ D1295	-	Store the sent data of MODRW instruction.
	-	D1089 ~ D1099	-	Store the sent data of MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF instruction
Data Receiving	M1313	M1124	M1317	Data receiving ready
	-	M1125	-	Communication ready status reset
	-	M1128	-	Transmitting/Receiving status Indication
	-	D1123	-	Residual number of words of the receiving data
	-	D1070 ~ D1085	-	Store the feedback data of Modbus communication. RS instruction is not supported.
	D1167	D1168	D1169	Store the specific end word to be detected for executing interrupts I140/I150/I160 (RS)
Receiving completed	M1314	M1123	M1318	RS communication data receiving completed
	M1314	M1127	M1318	The sending / receiving of communication data is complete.
	-	M1131	-	ON when data is being converted from ASCII to Hex
	-	D1296 ~ D1311	-	Store the converted HEX data of MODRW instruction.

	COM1	COM2	COM3	Function Description
Receiving completed	-	D1050 ~ D1055	-	Store the converted HEX data of MODRD instruction
Errors	M1315	-	M1319	Data receiving error
	D1250	-	D1253	Communication error code
	-	M1129	-	Receiving time out
	-	M1140	-	MODRD/MODWR/MODRW data receiving error
	-	M1141	-	MODRD/MODWR/MODRW parameter error (Exception Code exists in received data) Exception Code is stored in D1130
	-	M1142	-	Data receiving error of VFD-A handy instructions (FWD/REV/STOP/RDST/RSTEF)
	-	D1130	-	Error code returning from Modbus communication

5. How to set up RS-485 communication protocol in D1120

	Content	0	1
b0	Data length	7	8
b1 **b2**	Parity bits	00: None 01: Odd 11: Even	
b3	Stop bits	1 bit	2 bits
b4 **b5** **b6** **b7**	0001 (H1) : 110 0010 (H2) : 150 0011 (H3) : 300 0100 (H4) : 600 0101 (H5) : 1200 0110 (H6) : 2400 0111 (H7) : 4800 1000 (H8) : 9600 1001 (H9) : 19200 1010 (HA) : 38400 1011 (HB) : 57600 (does not support ES/SS V5.8 and below) 1100 (HC) : 115200 (does not support ES/SS V5.8 and below)		
b8	Start word	None	D1124
b9	First end word	None	D1125
b10	Second end word	None	D1126
b15 ~ b11	Not defined		

6. When RS instruction is in use, the frequently used communication format in the peripheral device will define the start word and end word of the control string. Therefore, you can set up the start word and end word in D1124 ~ D1126 for COM2 or use the start word and end word defined by the PLC. When you use M1126, M1130 and D1124 ~ D1126 to set up the start word and end word, b8 ~ b10 of D1120 have to be set as 1 to make valid the RS-485 communication protocol. See the table below for how to set up.

		M1130	
		0	1
M1126	0	D1124: user defined D1125: user defined D1126: user defined	D1124: H 0002 D1125: H 0003 D1126: H 0000 (no setting)
	1	D1124: user defined D1125: user defined D1126: user defined	D1124: H 003A (':') D1125: H 000D (CR) D1126: H 000A (LF)

7. Example of how to set up the communication format of COM2:

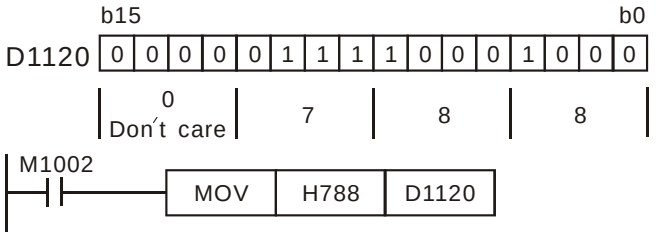
   Assume there is a communication format: Baud rate 9600 7, N, 2

   STX : ":"

   ETX1 : "CR"

   ETX2 : "LF"

   Check the table and obtain the communication format H788 and write it into D1120.



   When STX, ETX1 and EXT2 are in use, please be aware of the On and Off of the special auxiliary relays M1126 and M1130.

8. D1250 (COM1)/D1253 (COM3) communication error code:

Value	Error Description
H0001	Communication time-out
H0002	Checksum error
H0003	Exception code exists
H0004	Command code error / data error
H0005	Communication data length error

9. The relation between special data registers and interrupts is described below. (Only lower 8 bits are valid.) Three communication interrupts at most can be enabled by the program in a DVP-EH3/SV2 series PLC. Users have to note the interrupt numbers used. (DVP-SV2 series PLCs do not support COM3.)

Communication port	Interrupt number	Special data register
COM1	I151	D1397
	I161	D1398
COM2	I150	D1168
	D160	D1169
	I170	-
COM3	I153	D1242
	I163	D1243

The interrupts I151, I161, I153, and I163 only support EH3/EH3-L/SV2 V2.00 (and above). EH2 and SV only support COM2 communucation interrupts.

10. M1143 is for the selection of ASCII mode or RTU mode. On = RTU mode; Off = ASCII mode.
Take the standard Modbus format for example:
In ASCII mode (M1143 = Off)

STX	Start word = ':' (3AH)
Address Hi	Communication address:
Address Lo	The 8-bit address consists of 2 ASCII codes
Function Hi	Function code:
Function Lo	The 8-bit function code consists of 2 ASCII codes
DATA (n-1)	Data:
…….	The n × 8-bit data consists of 2n ASCll codes
DATA 0	
LRC CHK Hi	LRC checksum:
LRC CHK Lo	The 8-bit checksum consists of 2 ASCll code
END Hi	End word:
END Lo	END Hi = CR (0DH), END Lo = LF(0AH)

The communication protocol is in Modbus ASCII mode, i.e. every byte is composed of 2 ASCII characters. For example, 64Hex is '64' in ASCII, composed by '6' (36Hex) and '4' (34Hex). Every hex '0'…'9', 'A'…'F' corresponds to an ASCII code.

Character	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'
ASCII code	30H	31H	32H	33H	34H	35H	36H	37H

Character	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'
ASCII code	38H	39H	41H	42H	43H	44H	45H	46H

Start word (STX):

Fixed as ':' (3AH)

Address:

'0' '0': Broadcasting to all drivers

'0' '1': To the driver at address 01

'0' 'F': To the driver at address 15

'1' '0': To the driver at address 16

….and so on, maximum to the driver at address 254 ('F' 'E')

Function code:

'0' '1': Reading several bit devices

'0' '2': Reading several bit devices (read-only devices)

'0' '3': Reading several word devices

'0' '4': Reading several word devices (read-only devices)

'0' '5': Writing a state in a single bit device

'0' '6': Writing data in a single word device

'0' 'F': Writing states in bit devices

'1' '0': Writing data in word devices

'1' '7': Reading word devices and writing data in word devices

Data characters: The data sent by the user.

LRC checksum:

LCR checksum is 2's complement of the value added from Address to Data Content.

For example: 01H + 03H + 21H + 02H + 00H + 02H = 29H. 2's complement of 29H = D7H

End word (END):

Fixed as END Hi = CR (0DH), END Lo = LF (0AH)

For example: Read 2 continuous data stored in the registers of the driver at address 01H (see the table below).

The start register is at address 2102H.

Inquiry message:

STX	': '
Slave station address	'0'
	'1'
Function code	'0'
	'3'
Start address	'2'
	'1'
	'0'
	'2'
Number of data (counted by words)	'0'
	'0'
	'0'
	'2'
LRC checksum	'D'
	'7'
END	CR
	LF

Responding message:

STX	': '
Slave station address	'0'
	'1'
Function code	'0'
	'3'
Number of data (counted by byte)	'0'
	'4'
Content in start address 2102H	'1'
	'7'
	'7'
	'0'
Content of address 2103H	'0'
	'0'
	'0'
	'0'
LRC check	'7'
	'1'
END	CR
	LF

In RTU mode (M1143 = On)

Name	Data (hexadecimal system)
START	See the following explanation
Address	Communication address: In 8-bit binary
Function	Function code: In 8-bit binary
DATA (n-1)	Data: n × 8-bit data
…….	
DATA 0	
CRC CHK Low	CRC checksum: 16-bit CRC consists of 2 8-bit binary
CRC CHK High	
END	See the following explanation

Address:

00H: Broadcasting to all drivers

01H: To the driver at address 01

0FH: To the driver at address 15

10H: To the driver at address 16…. And so on, maximum to the driver at address 254 (FE H)

Function code:

02H: Reading several bit devices

03H: Reading several word devices

04H: Reading several word devices (read-only devices)

05H: Writing a state in a single bit device

06H: Writing data in a single word device

0FH: Writing states in bit devices

10H: Writing data in word devices

17H: Reading word devices and writing data in word devices

Data characters: The data sent by the user.

CRC checksum: Starting from Address and ending at Data Content.

Step 1: Make the 16-bit register (CRC register) = FFFFH

Step 2: Exclusive OR the first 8-bit message and the low 16-bit CRC register. Store the result in the CRC register.

Step 3: Right shift CRC register for a bit and fill “0” into the high bit.

Step 4: Check the value shifted to the right. If it is 0, fill in the new value obtained in step 3 and store the value in CRC register; otherwise, Exclusive OR A001H and CRC register and store the result in the CRC register.

Step 5: Repeat step 3 – 4 and finish operations of all the 8 bits.

Step 6: Repeat step 2 – 5 for obtaining the next 8-bit message until the operation of all the messages are completed. The final value obtained in the CRC register is the CRC checksum. The CRC checksum has to be placed interchangeably in the checksum of the message.

START and END:

For ES/EX/EC V5.8 (and below) and SX V1.1 (and below) series MPU, keep no input signal be ≥ 10ms.

See the table below for EC3-8K/EH3/SV2 series MPU:

Baud rate(bps)	RTU timeout timer (ms)	Baud rate (bps)	RTU timeout timer (ms)
300	40	9,600	2
600	21	19,200	1
1,200	10	38,400	1
2,400	5	57,600	1
4,800	3	115,200	1

For example: Read 2 continuous data stored in the registers of the driver at address 01H (see the table below).

The start register is at address 2102H.

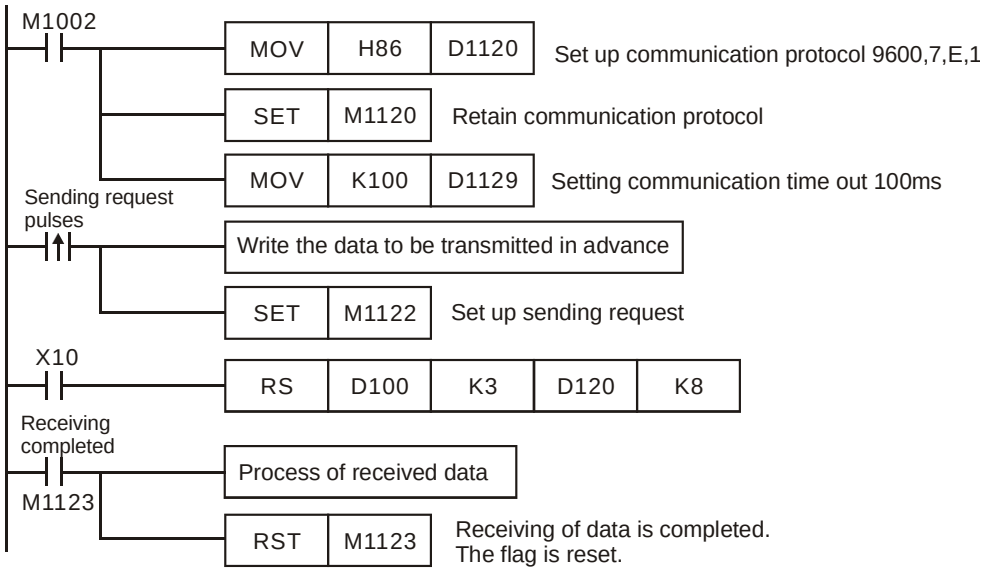
Inquiry message:

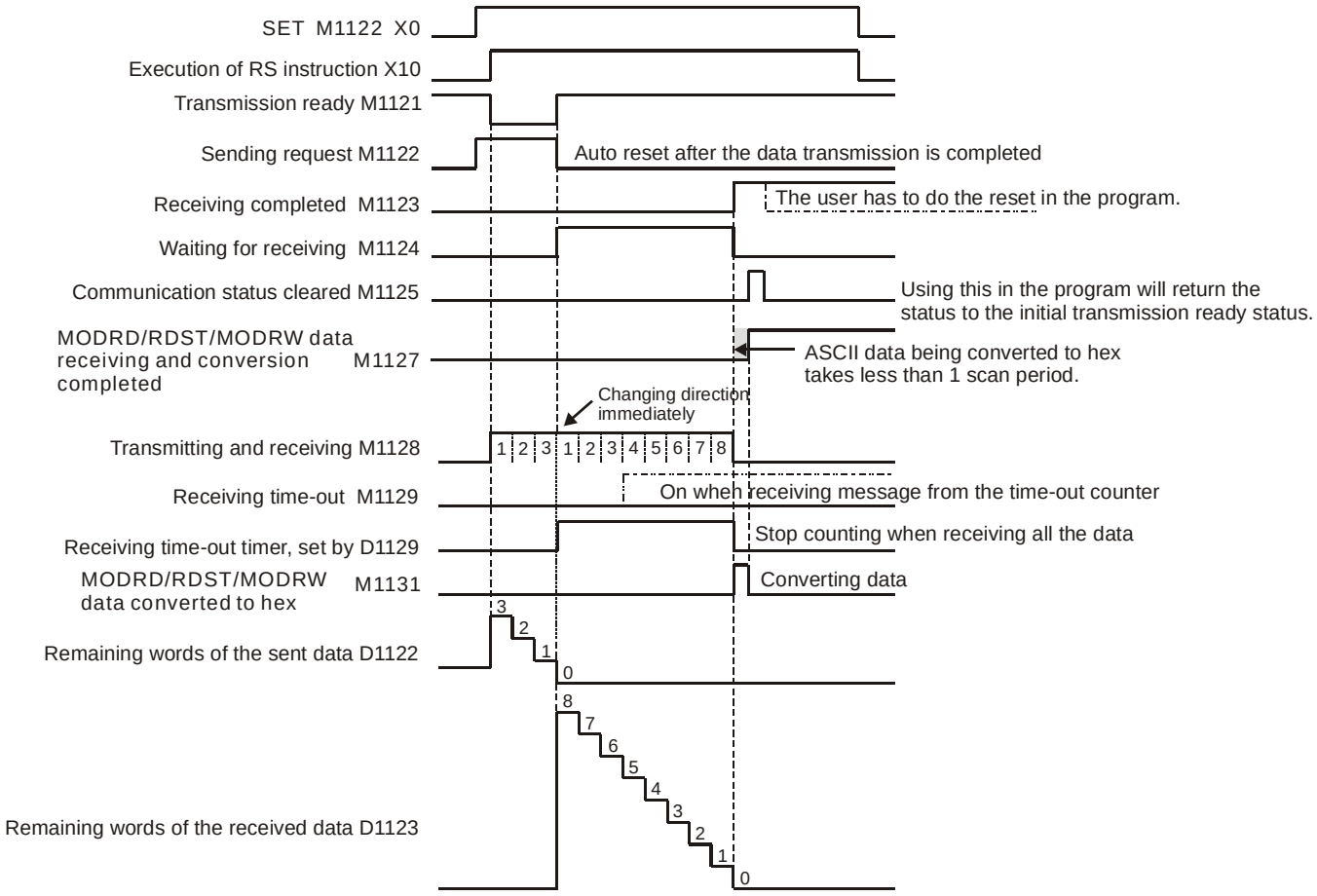
Name	Data (Hexadecimal value)
Address	01 H
Function code	03 H
Start data address	21 H
	02 H
Number of data (counted by words)	00 H
	02 H
CRC CHK Low	6F H
CRC CHK High	F7 H

Responding message:

Name	Data (Hexadecimal value)
Address	01 H
Function	03 H
Number of data (counted by byte)	04 H
Content in data address 2102H	17 H
	70 H
Content in data address 2103H	00 H
	00 H
CRC CHK Low	FE H
CRC CHK High	5C H

11. Timing diagram of the RS-485 communication flag for COM2:




SET M1122 X0
Execution of RS instruction X10
Transmission ready M1121
Sending request M1122
Auto reset after the data transmission is completed
Receiving completed M1123
The user has to do the reset in the program.
Waiting for receiving M1124
Communication status cleared M1125
Using this in the program will return the status to the initial transmission ready status.
MODRD/RDST/MODRW data receiving and conversion completed M1127
ASCII data being converted to hex takes less than 1 scan period.
Changing direction immediately
Transmitting and receiving M1128
1 2 3 1 2 3 4 5 6 7 8
Receiving time-out M1129
On when receiving message from the time-out counter
Receiving time-out timer, set by D1129
Stop counting when receiving all the data
MODRD/RDST/MODRW data converted to hex M1131
Converting data
Remaining words of the sent data D1122
3 2 1 0
Remaining words of the received data D1123
8 7 6 5 4 3 2 1 0

API	Mnemonic			Operands	Function
81	D	PRUN	P	(S) (D)	Parallel Run

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*		*							PRUN, PRUNP: 5 steps DPRUN, DPRUNP: 9 steps
D								*	*							

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

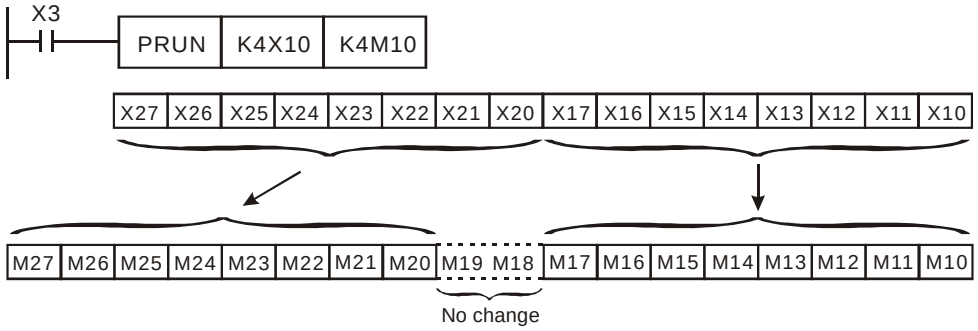
S: Source device **D**: Destination device

Explanations:

1. The most right digit of X, Y and M of KnX, KnY and KnM has to be 0.
2. When **S** designates KnX, **D** has to designate KnM; when **S** designates KnM, **D** has to designate KnY.
3. See the specifications of each model for their range of use.
4. This instruction sends the content in **S** to **D** in the form of octal system.

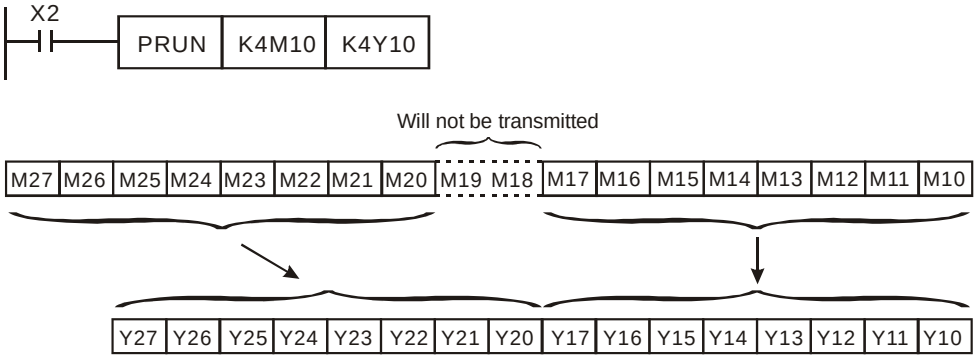
Program Example 1:

When X3 = On, the content in K4X10 will be sent to K4M10 in octal form.



Program Example 2:

When X2 = On, the content in K4M10 will be sent to K4Y10 in octal form.



API	Mnemonic			Operands	Function
82		ASCI	P	S D n	Converts Hex to ASCII

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ASCI, ASCIP: 7 steps
S					*	*	*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

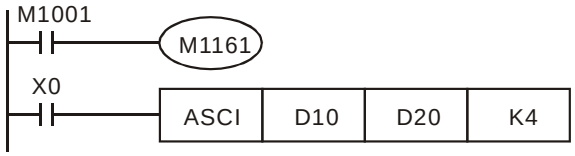
S: Start device for source data **D**: Start device for storing the converted result **n**: Number of bits to be converted

Explanations:

1. Range of **n**: 1 ~ 256
2. See the specifications of each model for their range of use.
3. Flag: M1161 (8/16 bit mode switch)
4. 16-bit conversion mode: When M1161 = Off, the instruction converts every bit of the hex data in **S** into ASCII codes and send them to the 8 high bits and 8 low bits of **D**. **n** = the converted number of bits.
5. 8-bit conversion mode: When M1161 = On, the instruction converts every bit of the hex data in **S** into ASCII codes and send them to the 8 low bits of **D**. **n** = the number of converted bits. (All 8 high bits of **D** = 0)

Program Example 1:

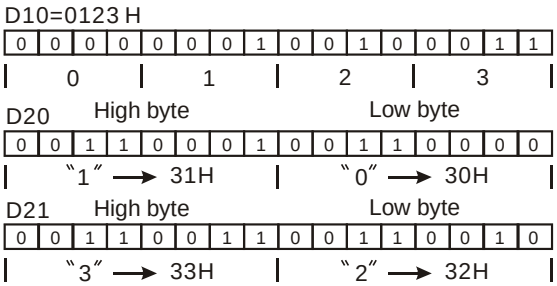
1. M1161 = Off: The 16-bit conversion mode
2. When X0 = On, convert the 4 hex values in D10 into ASCII codes and send the result to registers starting from D20.



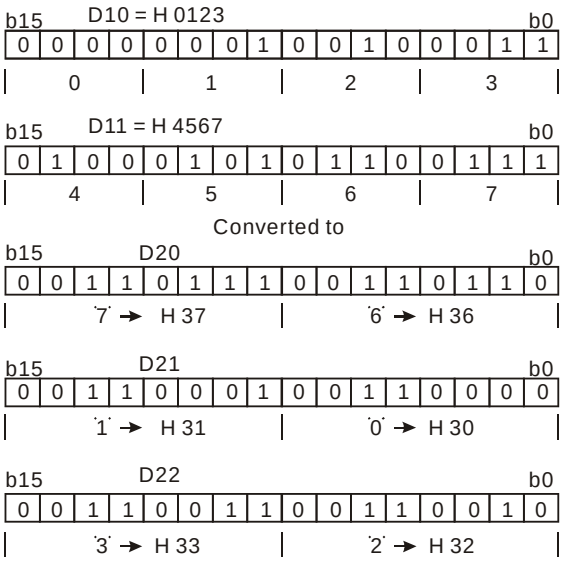
3. Assume

(D10) = 0123 H	'0' = 30H	'4' = 34H	'8' = 38H
(D11) = 4567 H	'1' = 31H	'5' = 35H	'9' = 39H
(D12) = 89AB H	'2' = 32H	'6' = 36H	'A' = 41H
(D13) = CDEF H	'3' = 33H	'7' = 37H	'B' = 42H

4. When **n** = 4, the bit structure will be as:



5. When **n** = 6, the bit structure will be as:



6. When **n** = 1 ~ 16:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20 Low byte	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D20 High byte		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D21 Low byte			"3"	"2"	"1"	"0"	"7"	"6"
D21 High byte				"3"	"2"	"1"	"0"	"7"
D22 Low byte					"3"	"2"	"1"	"0"
D22 High byte						"3"	"2"	"1"
D23 Llow byte							"3"	"2"
D23 High byte								"3"
D24 Low byte								
D24 High byte								
D25 Low byte								
D25 High byte								
D26 Low byte								
D26 High byte								
D27 Low byte								
D27 High byte								

no change

D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20 Low byte	“B”	“A”	“9”	“8”	“F”	“E”	“D”	“C”
D20 High byte	“4”	“B”	“A”	“9”	“8”	“F”	“E”	“D”
D21 Low byte	“5”	“4”	“B”	“A”	“9”	“8”	“F”	“E”
D21 High byte	“6”	“5”	“4”	“B”	“A”	“9”	“8”	“F”
D22 Low byte	“7”	“6”	“5”	“4”	“B”	“A”	“9”	“8”
D22 High byte	“0”	“7”	“6”	“5”	“4”	“B”	“A”	“9”
D23 Llow byte	“1”	“0”	“7”	“6”	“5”	“4”	“B”	“A”
D23 High byte	“2”	“1”	“0”	“7”	“6”	“5”	“4”	“B”
D24 Low byte	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D24 High byte		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D25 Low byte			“3”	“2”	“1”	“0”	“7”	“6”
D25 High byte				“3”	“2”	“1”	“0”	“7”
D26 Low byte					“3”	“2”	“1”	“0”
D26 High byte			no change			“3”	“2”	“1”
D27 Low byte							“3”	“2”
D27 High byte								“3”

Program Example 2:

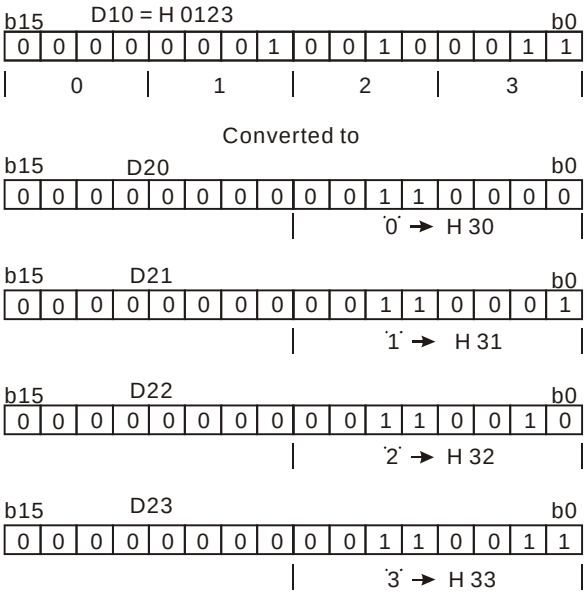
1. M1161 = On: The 8-bit conversion mode
2. When X0 = On, convert the 4 hex values in D10 into ASCII codes and send the result to registers starting from D20.

3. Assume

(D10) = 0123 H	‘0’ = 30H	‘4’ = 34H	‘8’ = 38H
(D11) = 4567 H	‘1’ = 31H	‘5’ = 35H	‘9’ = 39H
(D12) = 89AB H	‘2’ = 32H	‘6’ = 36H	‘A’ = 41H
(D13) = CDEFH	‘3’ = 33H	‘7’ = 37H	‘B’ = 42H

4. When **n** = 2, the bit structure will be as:

5. When **n** = 4, the bit structure will be as:



6. When **n** = 1 ~ 16:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D21		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D22			“3”	“2”	“1”	“0”	“7”	“6”
D23				“3”	“2”	“1”	“0”	“7”
D24					“3”	“2”	“1”	“0”
D25						“3”	“2”	“1”
D26							“3”	“2”
D27								“3”
D28								
D29				no change				
D30								
D31								
D32								
D33								
D34								
D35								

n / D	K9	K10	K11	K12	K13	K14	K15	K16
D20	“B”	“A”	“9”	“8”	“F”	“E”	“D”	“C”
D21	“4”	“B”	“A”	“9”	“8”	“F”	“E”	“D”
D22	“5”	“4”	“B”	“A”	“9”	“8”	“F”	“E”
D23	“6”	“5”	“4”	“B”	“A”	“9”	“8”	“F”
D24	“7”	“6”	“5”	“4”	“B”	“A”	“9”	“8”
D25	“0”	“7”	“6”	“5”	“4”	“B”	“A”	“9”
D26	“1”	“0”	“7”	“6”	“5”	“4”	“B”	“A”
D27	“2”	“1”	“0”	“7”	“6”	“5”	“4”	“B”
D28	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D29		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D30			“3”	“2”	“1”	“0”	“7”	“6”
D31				“3”	“2”	“1”	“0”	“7”
D32					“3”	“2”	“1”	“0”
D33			no change			“3”	“2”	“1”
D34							“3”	“2”
D35								“3”

API	Mnemonic	Operands	Function
83	HEX P	S D n	Converts ASCII to Hex

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			HEX, HEXP: 7 steps
D								*	*	*	*	*	*			
n					*	*										

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

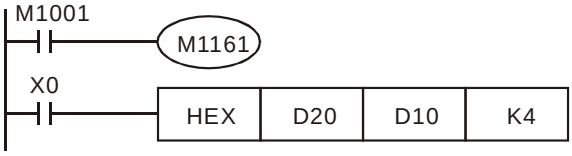
S: Start device for source data **D**: Start device for storing the converted result **n**: Number of bits to be converted

Explanations:

1. Range of **n**: 1 ~ 256
2. See the specifications of each model for their range of use.
3. Flag: M1161 (8/16 bit mode switch)
4. 16-bit conversion mode: When M1161 = Off, the instruction is in 16-bit conversion mode. ASCII codes of the 8 high bits and 8 low bits of the hex data in **S** are converted into hex value and sent to **D** (every 4 bits as a group). **n** = the number of bits converted into ASCII codes.
5. 8-bit conversion mode: When M1161 = On, the instruction is in 8-bit conversion mode. Every bit of the hex data in **S** are converted into ASCII codes and sent to the 8 low bits of **D**. **n** = the number of converted bits. (All 8 high bits of **D** = 0)
6. If the ASCII code is not in the range of H30~H39 (0~9) or is not in the range H41~H46 (A~F), HEX will set M1067, and the conversion of the ASCII code into a hexadecimal value will stop.

Program Example 1:

1. M1161 = Off: The 16-bit conversion mode
2. When X0 = On, convert the ASCII codes stored in the registers starting from D20 into hex value and send the result (every 4 bits as a group) to registers starting from D10. **n** = 4.

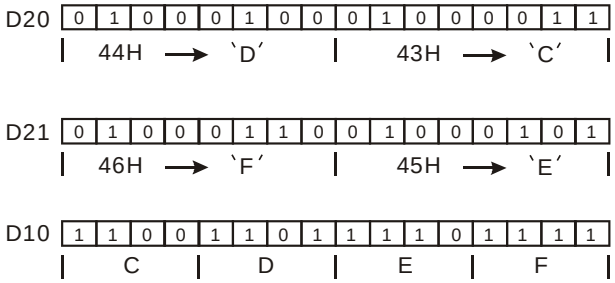


3. Assume

S	ASCII code	Converted to hex	S	ASCII code	Converted to hex
D20 low byte	H 43	“C”	D24 low byte	H 34	“4”
D20 high byte	H 44	“D”	D24 high byte	H 35	“5”
D21 low byte	H 45	“E”	D25 low byte	H 36	“6”
D21 high byte	H 46	“F”	D25 high byte	H 37	“7”
D22 low byte	H 38	“8”	D26 low byte	H 30	“0”

S	ASCII code	Converted to hex	S	ASCII code	Converted to hex
D22 high byte	H 39	“9”	D26 high byte	H 31	“1”
D23 low byte	H 41	“A”	D27 low byte	H 32	“2”
D23 high byte	H 42	“B”	D27 high byte	H 33	“3”

4. When **n** = 4, the bit structure will be as:

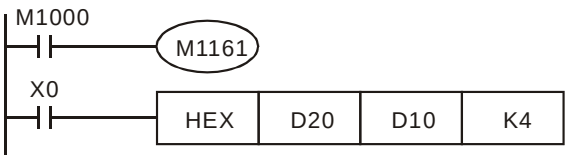


5. When **n** = 1 ~ 16:

n \ D	D13	D12	D11	D10
1	The undesignated parts in the registers in use are all 0.			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

Program Example 2:

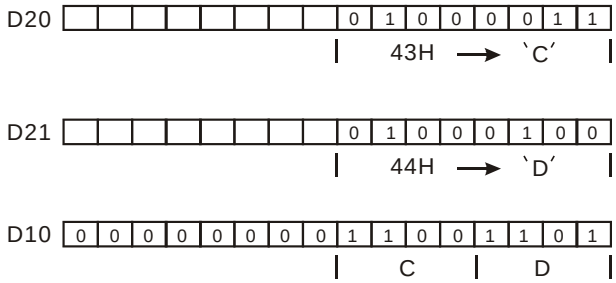
1. M1161 = On: The 8-bit converstion mode



2. Assume

S	ASCII code	Converted to hex	S	ASCII code	Converted to hex
D20	H 43	“C”	D28	H 34	“4”
D21	H 44	“D”	D29	H 35	“5”
D22	H 45	“E”	D30	H 36	“6”
D23	H 46	“F”	D31	H 37	“7”
D24	H 38	“8”	D32	H 30	“0”
D25	H 39	“9”	D33	H 31	“1”
D26	H 41	“A”	D34	H 32	“2”
D27	H 42	“B”	D35	H 33	“3”

3. When **n** = 2, the bit structure will be as:



4. When **n** = 1 ~ 16:

n \ D	D13	D12	D11	D10
1	The used registers which are not specified are all 0			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

API	Mnemonic		Operands	Function
84	CCD	P	S D n	Check Code

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CCD, CCDP: 7 steps
S							*	*	*	*	*	*	*			
D									*	*	*	*	*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

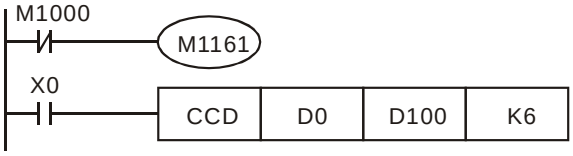
S: Start device for source data **D**: Device for storing the sum check result **n**: Number of data

Explanations:

1. Range of **n**: 1 ~ 256
2. See the specifications of each model for their range of use.
3. Flag: M1161 (8/16 bit mode switch)
4. The sum check is used for ensuring the correctness of the data transmission.
5. 16-bit conversion mode: When M1161 = Off, the instruction is in 16-bit conversion mode. The instruction sums up **n** data (8 bits as a unit) from the start register designated in **S** and stores the results in the registers designated in **D**. The parity bits are stored in **D** + 1.
6. 8-bit conversion mode: When M1161 = On, the instruction is in 8-bit conversion mode. The instruction sums up **n** data (8 bits as a unit; only 8 low bits are valid) from the start register designated in **S** and stores the results in the registers designated in **D**. The parity bits are stored in **D** + 1.

Program Example 1:

1. M1161 = Off: The 16-bit conversion mode
2. When X0 = On, the instruction will sum up 6 data stored in the register designated in D0 (8 bits as a unit; **n** = 6 indicates D0 ~ D2 are designated) and store the result in the register designated in D100. The parity bits are stored in D101.



(S)	Content of data
D0 low byte	K100 = 0 1 1 0 0 1 0 0
D0 high byte	K111 = 0 1 1 0 1 1 1 ①
D1 low byte	K120 = 0 1 1 1 1 0 0 0
D1 high byte	K202 = 1 1 0 0 1 0 1 0
D2 low byte	K123 = 0 1 1 1 1 0 1 ①
D2 high byte	K211 = 1 1 0 1 0 0 1 ①
D100	K867
D101	0 0 0 1 0 0 0 ①

Total

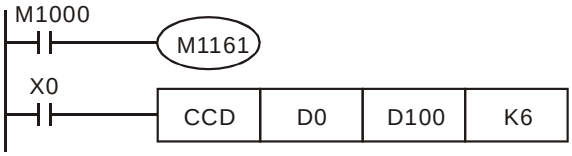
The parity is 1 when there is a odd number of 1.
The parity is 0 when there is a even number of 1.

D100	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1
D101	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

⟸ Parity

Program Example 2:

1. M1161 = On: The 8-bit conversion mode
2. When X0 = On, the instruction will sum up 6 data stored in the register designated in D0 (8 bits as a unit; **n** = 6 indicates D0 ~ D5 are designated) and store the result in the register designated in D100. The parity bits are stored in D101.



(S)	Content of data
D0 low byte	K100 = 0 1 1 0 0 1 0 0
D1 low byte	K111 = 0 1 1 0 1 1 1 ①
D2 low byte	K120 = 0 1 1 1 1 0 0 0
D3 low byte	K202 = 1 1 0 0 1 0 1 0
D4 low byte	K123 = 0 1 1 1 1 0 1 ①
D5 low byte	K211 = 1 1 0 1 0 0 1 ①
D100	K867
D101	0 0 0 1 0 0 0 ①

Total

The parity is 1 when there is a odd number of 1.
The parity is 0 when there is a even number of 1.

D100	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1
D101	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

⟸ Parity

API	Mnemonic			Operands	Function
85		VRRD	P	(S) (D)	Volume Read

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										VRRD, VRRDP: 5 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

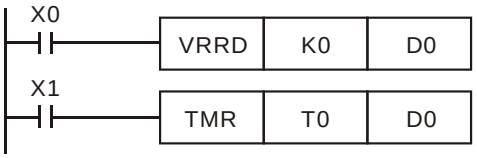
S: No. of VR **D**: Device for storing the volume of VR

Explanations:

1. Range of **S**: 0 ~ 7; without function card: 0 ~ 1.
2. See the specifications of each model for their range of use.
3. Flags: M1178, M1179. See remarks for more details.
4. VRRD instruction is used for reading 2 points (No.0, No.1) of PLC or the VR rotary switch volume change in the 6 points of the function cards (No.2 ~ No.7) and converting the data into values 0 ~ 255 (stored in **D**).
5. If you are to set up the timer by the VR volume, simply rotate the VR to modify the set time in the timer. If you are to acquire a value larger than 255, multiply **D** by a constant.

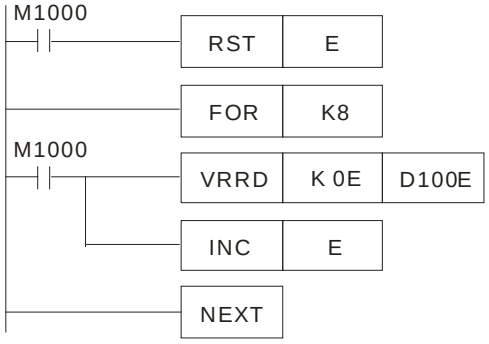
Program Example 1:

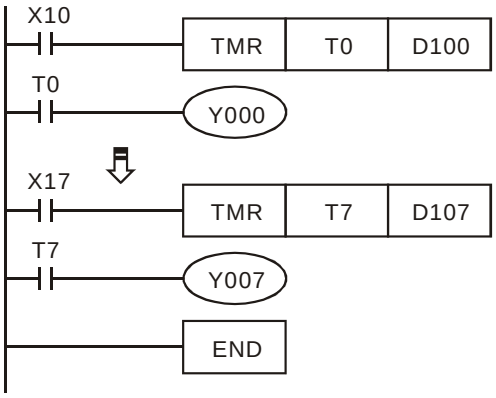
1. When X0 = On, VR0 volume changed will be converted into an 8-bit BIN value (0 ~ 255) and stored in D0.
2. When X1 = On, the imer T0 will start to time with the content in D0 as the set value in the timer.



Program Example 2:

1. Read the VR volume in order: The VR0 ~ VR7 rotary switches on the PLC correspond to **S** = K0 ~ K7 of VRRD instruction. E index register modification is used in the example below, K0E = K0 ~ K7.
2. The timer converts the scale 0 ~ 10 on the rotary switch into 0 ~ 255. The timing unit of T0 ~ T7 is 0.1 second; therefore, the set time in the timer will be 0 ~ 25.5 seconds.





3. Operation of FOR ~ NEXT instruction:
   a) In the area between FOR ~ NEXT instruction, FOR designating K8 indicates the loop between FOR ~ NEXT will be executed repeatedly for 8 times before the next instruction is executed.
   b) Between FOR ~ NEXT (INC E), E will be 0, 1, 2, …7 plusing 1. Therefore, the 8 VR rotary switch volumes will be VR0→D100, VR1→D101, VR2→D102…VR7→D107 and be read to designated registers in order.

Remarks:

1. VR refers to **V**ariable **R**esister.
2. The 2 points of VR rotary switch built in EC3-8K/SX/EH3/SV2 series MPU can be used together with special D and special M.

Device	Function
M1178	Enabling VR0
M1179	Enabling VR1
D1178	VR0 value
D1179	VR1 value

3. If there is no VR extension card inserted in the PLC, setting up the No. of rotary switches as K2 ~ K7 in VRRD and VRSC instruction in the program will result in errors in grammar check.

API	Mnemonic			Operands	Function
86		VRSC	P	(S) (D)	Volume Scale

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	VRSC, VRSCP: 5 steps
S					*	*										
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: No. of VR **D**: Device for storing the scale of VR

Explanations:

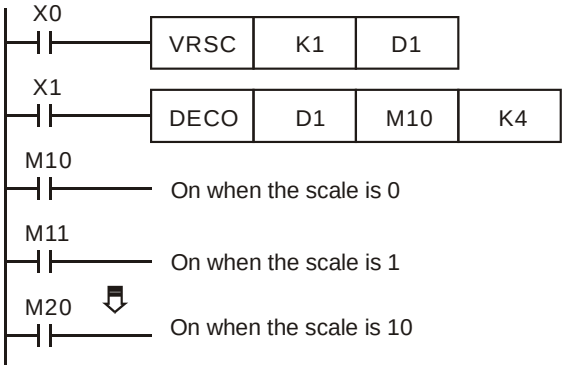
1. Range of **S**: 0 ~ 7; without function card: 0 ~ 1
2. See the specifications of each model for their range of use.
3. VRSC instruction is used for reading 2 points (No.0, No.1) of PLC or the VR rotary switch scale (0 ~ 10) in the 6 points of the function cards (No.2 ~ No.7) and storing the data in **D**. If the position of the VR falls in the middle of two scales, VRSC will round up the value into an integer of 0 ~ 10.

Program Example 1:

When X0 = On, the scale of VR0 (0 ~ 10) will be stored in D10.

Program Example 2:

1. When the VR is used as DIP switch, they will correspond to scale 0 ~ 10 and only one of M10 ~ M20 will be On. Use API 41 DECO instruction to decode the scales into M10 ~ M25.
2. When X0 = On, store the scale (0 ~ 10) of VR1 into D1.
3. When X1 = On, use API 41 DECO to decode the scales into M10 ~ M25.



Remarks:

If the MPU is not inserted with a VR extension card, and the No. of the rotary switches inVRRD or VRSC instruction in the program are set as K2 ~ K7, errors will occur in the execution of grammar check.

API	Mnemonic			Operands	Function
87	D	ABS	P	(D)	Absolute Value

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	ABS, ABSP: 3 steps DABS, DABSP: 5 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

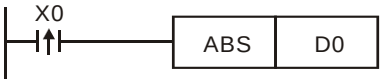
D: Device of the absolute value

Explanations:

1. See the specifications of each model for their range of use.
2. This instruction obtains the absolute value of the content in the designated in **D**.
3. This instruction adopts pulse execution instructions (ABSP, DABSP).

Program Example:

When X0 = Off→On, obtain the absolute value of the content in D0.



API	Mnemonic	Operands	Function
88	D PID	(S1) (S2) (S3) (D)	PID Control Loop

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1													*			PID : 9 steps
S2													*			DPID: 17 steps
S3													*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

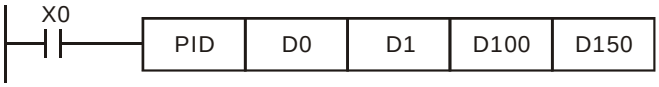
S1: Set value (SV) **S2**: Present value (PV) **S3**: Parameter **D**: Output value (MV)

Explanations:

1. In the 16-bit instruction, **S3** will occupy 20 consecutive devices; in the 32-bit instruction, **S3** will occupy 21 consecutive devices.
2. See the specifications of each model for their range of use.
3. See the Remarks below for the times of using PID instruction allowed in the program.
4. This instruction is specifically for PID control. PID operation will be executed by the scan only when the sampling time is reached. PID refers to “proportion, integration and differential”. PID control is widely applied to many machines, pneumatic and electronic equipments.
5. For the 16-bit instruction, the parameters are **S3** ~ **S3**+19; for the 32-bit instruction, the parameters are **S3** ~ **S3**+20. After all the parameters are set up, PID instruction will start to be executed and the results will be stored in **D**. **D** has to be the data register area without latched function. (If you wish to designate a latched data register area, place the data register in the latched area at the beginning of the program and clear it as 0.)

Program Example:

1. Complete the parameter setting before executing PID instruction.
2. When X0 = On, the instruction will be executed and the result will be stored in D150. When X0 goes Off, the instruction will not be executed and the data prior to the instruction will stay intact.



Remarks:

1. ES/EX/EC series MPU V5.7 (and above) supports PID instruction.
2. There is no limitation on the times of using this instruction. However, the register No. designated in **S3** cannot be repeated.
3. For the 16-bit instruction, **S3** will occupy 20 registers. In the program example above, the area designated in **S3** is D100 ~ D119. Before the execution of PID instruction, you have to transmit the setting value to the designated register area by MOV instruction, If the designated registers are latched, use MOVP instruction to transmit all setting value at a time.
4. Settings of **S3** in the 16-bit instruction

Device No.	Function	Setup Range	Explanation
S_3:	Sampling time (T_S) (unit: 10ms)	1 ~ 2,000 (unit: 10ms)	If T_S is less than 1 program scan time, PID instruction will be executed for 1 program scan time. If T_S= 0, PID instruction will not be enabled. The minimum T_S has to be longer than the program scan time.
S_3 +1:	Proportional gain (K_P)	0 ~ 30,000 (%)	The magnified error proportional value between SV – PV.
S_3 +2:	Integral gain (K_I)	0 ~ 30,000 (%)	For control mode K0~K8
S_3 +3:	Differential gain (K_D)	-30,000 ~ 30,000 (%)	For control mode K0~K8
S_3 +4:	Control mode	0: automatic control 1: forward control (E = SV - PV) 2: inverse control (E = PV - SV) 3: Auto-tuning of parameter exclusively for the temperature control. The device will automatically become K4 when the auto-tuning is completed and be filled in with the appropriate parameter K_P, K_I and K_D (not avaliable in the 32-bit instruction). 4: Exclusively for the adjusted temperature control (not avaliable in the 32-bit instruction). 5: Auto direction control (limited integrall upper/lower limit) P.S. K5 mode is only available in SV/EH2/EH3/SV2 V1.2, SX V1.8 and SC V1.6, and EH3/SV2 V1.0. 7: Manual control 1: Users set an MV. The accumulated integral value increases according to the error. It is suggested that the control mode should be used in a control environment which changes more slowly. EH3 V1.62 and SV2 V1.62 are supported. 8: Manual control 2: Users set an MV. The accumulated integral value will stop increasing. When the control mode becomes the automatic mode (the control mode K5 is used), the instruction PID outputs an appropriate accumulated integral value according to the last MV. EH3 V1.62 and SV2 V1.62 are supported.	
S_3 +5:	The range that error value (E) doesn't work	0 ~ 32,767	E = the error of SV – PV. When **S_3** +5 = K0, the function will not be enabled, e.g. when **S_3** +5 is set as 5, MV of E between -5 and 5 will be 0.
S_3 +6:	Upper bound of output value (MV)	-32,768 ~ 32,767	Ex: if **S_3** +6 is set as 1,000, the output will be 1,000 when MV is bigger than 1,000. **S_3** +6 has to be bigger or equal **S_3** +7; otherwise the upper bound and lower bound will switch.
S_3 +7:	Lower bound of output	-32,768 ~ 32,767	Ex: if **S_3** +7 is set as -1,000, the output will be

Device No.	Function	Setup Range	Explanation
	value (MV)		-1,000 when MV is smaller than -1,000.
S_3 +8:	Upper bound of integral value	-32,768~32,767	Ex: if **S_3** +8 is set as 1,000, the output will be 1,000 when the integral value is bigger than 1,000 and the integration will stop. **S_3** +8 has to be bigger or equal **S_3** +9; otherwier the upper bound and lower bound will switch.
S_3 +9:	Lower bound of integral value	-32,768 ~ 32,767	Ex: if **S_3** +9 is set as -1,000, the output will be -1,000 when the integral value is smaller than -1,000 and the integration will stop. If **S_3**+8 and **S_3**+9 are set to 0, there will be no upper limit for integration.
S_3 +10,11:	Accumulated integral value	32-bit floating point	The accumulated integral value is only for reference. You can still clear or modify it (in 32-bit floating point) according to your need.
S_3 +12:	The previous PV	-32,768~32,767	The previous PV is only for reference. You can still modify it according to your need.
S_3 +13: ~ **S_3** +19:	For system use only.		

5. When parameter setting exceeds its range, the upper bound and lower bound will become the setting value. However, if the motion direction (DIR) exceeds the range, it will be set to 0.
6. PID instruction can be used in interruption subroutines, step points and CJ instruction.
7. The maximum error of sampling time T_S = - (1 scan time + 1ms) ~ + (1 scan time). When the error affects the output, please fix the scan time or execute PID instruction in the interruption subroutine of the timer.
8. PV of PID instruction has to be stable before the execution of PID instruction. If you are to extract the input value of DVP04AD/04DA/06XA/04PT/04TC for PID operation, please be aware of the A/D conversion time of these modules.
9. For the 32-bit instruction, If **S_3** designates the parameter setting area of PID instruction as D100 ~ D120, **S_3** occupies 21 registers. Before the execution of PID instruction, you have to use MOV instrction first to send the setting value to the register area for setup. If the designated registers are latched one, use MOVP instruction to send all the setting value at a time.
10. Settings of **S_3** in the 32-bit instruction

Device No.	Function	Setup range	Explanation
S_3:	Sampling time (T_S) (unit: 10ms)	1 ~ 2,000 (unit: 10ms)	If T_S is less than 1 program scan time, PID instruction will be executed for 1 program scan time. If T_S= 0, PID instruction will not be enabled. The minimum T_S has to be longer than the program scan time.
S_3 +1:	Proportional gain (K_P)	0 ~ 30,000 (%)	The magnified error proportional value between SV – PV.

Device No.	Function	Setup range	Explanation
S3 +2:	Integral gain (K_I)	0 ~ 30,000 (%)	For control mode K0~K2, K5
S3 +3:	Differential gain (K_D)	-30,000 ~ 30,000 (%)	For control mode K0~K2, K5.
S3 +4:	Control direction (DIR)	0: automatic control 1: forward control (E = SV – PV) 2: inverse control (E = PV – SV) 5: Automatic mode with MV upper/lower bound control. When MV reaches upper/lower bound, the accumulation of integral value stops.	
S3 +5, 6:	The range that 32-bit error value (E) doesn't work	0 ~ 2,147,483,647	E = the error of SV – PV. When **S3** +5,6 = K0, the function will not be enabled, e.g. when **S3** +5,6 is set as 5, MV of E between -5 and 5 will be 0.
S3 +7, 8:	Upper bound of 32-bit output value (MV)	-2,147,483,648 ~ 2,147,483,647	Ex: if **S3** +7,8 is set as 1,000, the output will be 1,000 when MV is bigger than 1,000. **S3** +7,8 has to be bigger or equal **S3** +9,10; otherwise the upper bound and lower bound will switch.
S3 +9, 10:	Lower bound of 32-bit output value (MV)	-2,147,483,648 ~ 2,147,483,647	Ex: if **S3** +9,10 is set as -1,000, the output will be -1,000 when MV is smaller than -1,000.
S3 +11, 12:	Upper bound of 32-bit integral value	-2,147,483,648 ~ 2,147,483,647	Ex: if **S3** +11,12 is set as 1,000, the output will be 1,000 when the integral value is bigger than 1,000 and the integration will stop. **S3** +11,12 has to be bigger or equal **S3** +13,14; otherwier the upper bound and lower bound will switch.
S3 +13, 14:	Lower bound of 32-bit integral value	-2,147,483,648 ~ 2,147,483,647	Ex: if **S3** +13,14 is set as -1,000, the output will be -1,000 when the integral value is smaller than -1,000 and the integration will stop.
S3 +15, 16:	32-bit accumulated integral value	32-bit floating point	The accumulated integral value is only for reference. You can still clear or modify it (in 32-bit floating point) according to your need.
S3 +17, 18:	32-bit previous PV	-	The previous PV is only for reference. You can still modify it according to your need.
S3 +19: ≀ **S3** +20:	For system use only.		

11. The explanation of 32-bit **S3** and 16-bit **S3** are almost the same. The difference is the capacity of **S3**+5 ~ **S3**+20.

PID Equations:

1. The PID operation is conducted according to the speed and the differential PV.
2. The PID operation has three control directions: automatic, foreward and inverse. Forward or inverse are designated in **S3** +4. Other relevant settings of PID operation are set by the registers designated in **S3** ~ **S3** +5.
3. Basic PID equation:

$$MV = K_P * E(t) + K_I * E(t)\frac{1}{S} + K_D * PV(t)S$$

Control direction	PID equation
Forward, automatic	E(t) = SV – PV
Inverse	E(t) = PV – SV

$PV(t)S$ is the differential value of $PV(t)$; $E(t)\frac{1}{S}$ is the integral value of $E(t)$. When E(t) is less than 0 as

the control direction is selected as forward or inverse, $E(t)$ will be regarded as "0".

The equation above illustrates that this instruction is different from a general PID instruction by the variable use of the differential value. To avoid the flaw that the transient differential value is too big when a general PID instruction is executed for the first time, our PID instruction monitors the differentiation status of the PV. When the variation of PV is too big, this instruction will reduce the output of MV.

4. Symbol explanation:

MV : Output value

K_P : Proprotional gain

$E(t)$: Error value

PV : Present measured value

SV : Target value

K_D : Differential gain

$PV(t)S$: Differential value of PV(t)

K_I : Integral gain

$E(t)\frac{1}{S}$: Integral value of E(t)

5. Temperature Control Equation:

When **S_3** +4 is K3 and K4, the equation used in diagram 2 (see below) will be changed as:

$$MV = \frac{1}{K_P}\left[E(t) + \frac{1}{K_I}\left(E(t)\frac{1}{S} \right) + K_D * PV(t)S \right]$$

In which the error value is fixed as E(t) = SV – PV

This equation is exclusively designed for temperature control. Therefore, when the sampling time (T_S) is set as 4 seconds (K400), the range of output value (MV) will be K0 ~ K4,000 and the cycle time of GPWM instruction used together has to be set as 4 seconds (K4000) as well.

If you have no idea how to adjust the parameters, you can select K3 (auto-tuning) and after all the parameters are adjusted (the control direction will be automatically set as K4), you can modify your parameters to better ones according to the result of the control.

6. Control diagrams:

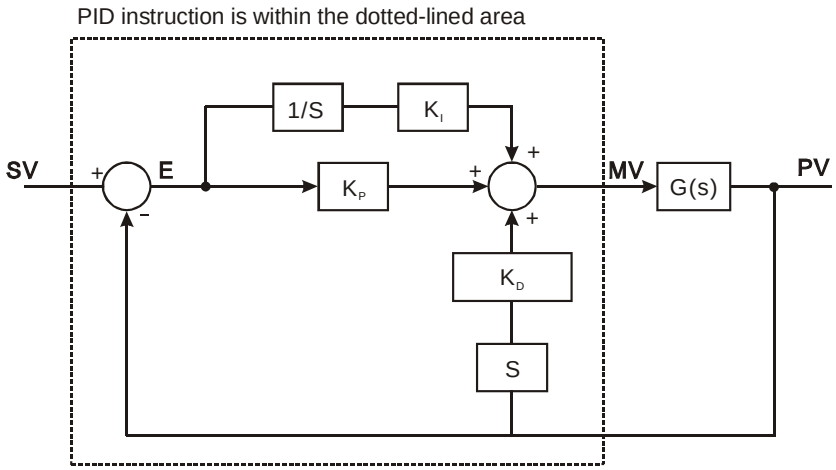


Diagram 1:**S**$_3$ + 4 = K0 ~ K2

In Diagram 1, S is differentiation, referring to “PV – previous PV / sampling time”. 1 / S is integration, referring to “(previous integral value + error value) × sampling time”. G(S) refers to the device being controlled.

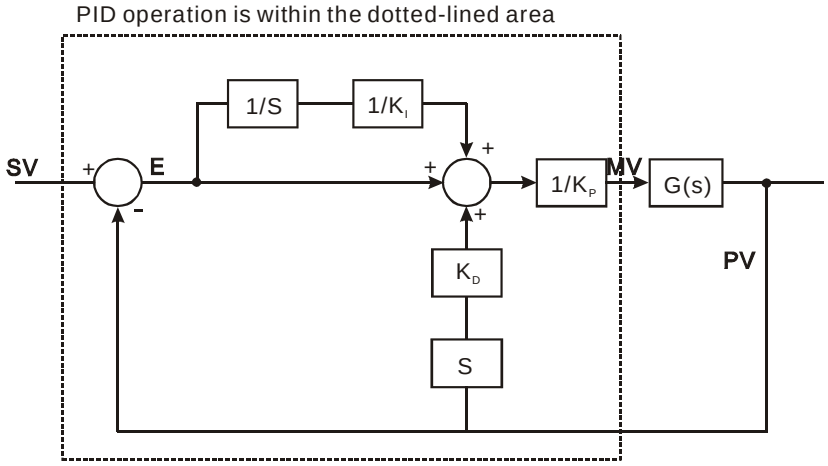


Diagram 2: **S**$_3$ + 4 = K3 ~ K4

In Diagram 2, $1/K_I$ and $1/K_P$ refer to “divided by K_I” and “divided by K_P”. Due to that this is exclusively for temperature control, you have to use PID instruction together with GPWM instruction. See Application 3 for more details.

7. Notes:

a) **S**$_3$ + 6 ~ **S**$_3$ + 13 are only available in SA/SX/SC/EH/EH2/SV series, and ES/EX/EC/EC3-8K (v5.7 and above) series MPU.

b) PID instruction can only be used once in ES/EX/EC/EC3-8K (v5.6 and below) series MPU. There is no limitation on the times of using PID instruction in ES/EX/EC/EC3-8K (v5.7 and above) series and SA/SX/SC/EH/EH2/SV/EH3/SV2 series MPU.

c) **S**$_3$ + 3 of ES/EX/EC/EC3-8K (v5.7 and below), SX (v1.1 and below) and EH (v1.0 and below) series MPU can only be the value within 0 ~ 30,000.

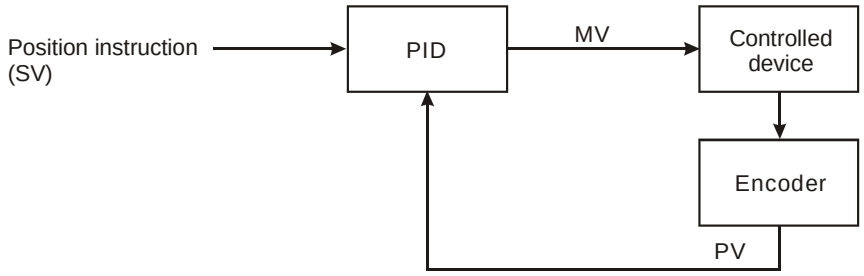
d) There are a lot of circumstances where PID instruction can be applied; therefore, please choose the control functions appropriately. For example, when you select parameter auto-tuning for the temperature (**S**$_3$ + 4 = K3), you cannot use it in a motor control environment in case improper control may occur.

e) When you adjust the three main parameters, K_P, K_I and K_D (**S**$_3$ + 4 = K0 ~ K2), you have to adjust K_P first (according to your experiences) and set K_I and K_D as 0. When you can roughly handle the control, you then adjust K_I (increasingly) and K_D (increasingly) (see example 4 below for how to adjust). K_P = 100 refers to 100%, i.e. the gain of the error is 1. K_P < 100% will decrease the error and K_P > 100% will increase the error.

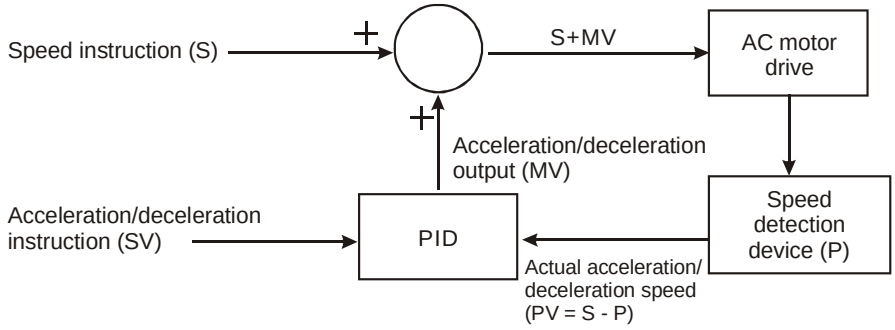
f) When you select the parameter exclusively for temperature control (**S**$_3$ + 4 = K3, K4), it is suggested that you store the parameter in D register in the latched area in case the automatically adjusted parameter will disappear after the power is cut off. There is no guarantee that the adjusted parameter is suitable for every control. Therefore, you can modify the adjusted parameter according to your actual need, but it is suggested that you modify only K_I or K_D.

g) PID instruction can to work with many parameters; therefore please do not randomly modify the parameters in case the control cannot be executed normally.

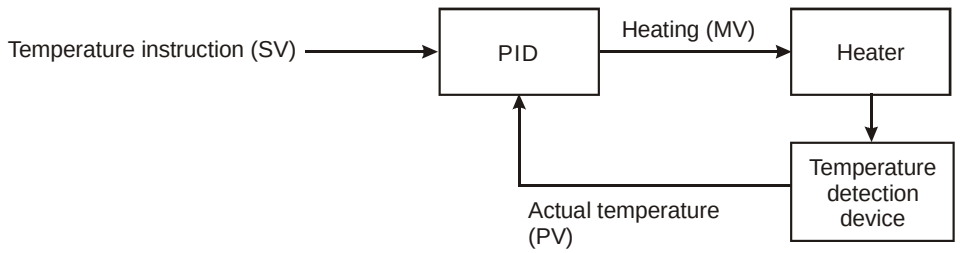
Example 1: Diagram of using PID instruction in position control (**S**$_3$ + 4 = 0)



Example 2: Diagram of using PID instruction with AC motor drive on the control (**S**$_3$ + 4 = 0)



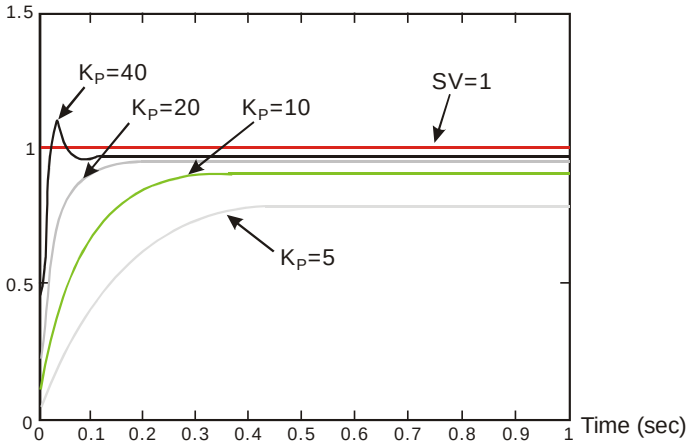
Example 3: Diagram of using PID instruction in temperature control (**S**$_3$ + 4 = 1)



Example 4: How to adjust PID parameters

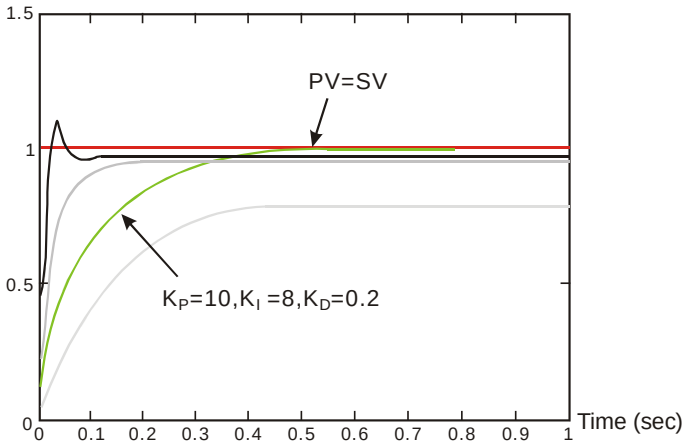
Assume that the transfer function of the controlled device G(S) in a control system is a first-order function $G(s)=\frac{b}{s+a}$ (most models of motors are first-order function), SV = 1, and sampling time (T_S) = 10ms, we suggest you to follow the steps below for adjusting the parameters.

Step 1: Set K_I and K_D as 0 and K_P as 5, 10, 20 and 40. Record the SV and PV respectively and the results are as the figure below.



Step 2: From the figure, we can see that when K_P = 40, there will be over-reaction, so we will not select it. When K_P = 20, the PV reaction curve will be close to SV and there will not be over-reaction, but due to its fast start-up with big transient MV, we will consider to put it aside. When K_P = 10, the PV reaction curve will get close to SV value more smoothly, so we will use it. Finally when K_P = 5, we will not consider it due to the slow reaction.

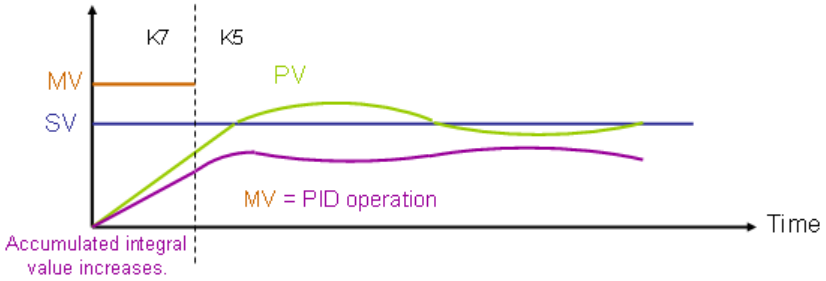
Step 3: Select K_P = 10 and adjust K_I from small to big (e.g. 1, 2, 4 to 8). K_I should not be bigger than K_P. Adjust K_D from small to big (e.g. 0.01, 0.05, 0.1 and 0.2). K_D should not exceed 10% of K_P. Finally we obtain the figure of PV and SV below.



Note: This example is only for your reference. Please adjust your parameters to proper ones according to your actual condition of the control system.

Example 5: Switching between the manual mode (K7) and the automatic mode (K5)

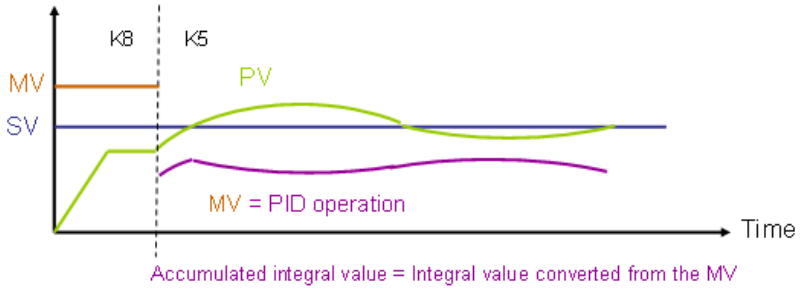
If the setting of the PID parameters is complete, and the control mode is the manual mode (K7), the control curve will be as shown below.



If the control mode becomes the automatic mode (K5), the output value MV changes from the output value set by users to the output value of the PID operation.

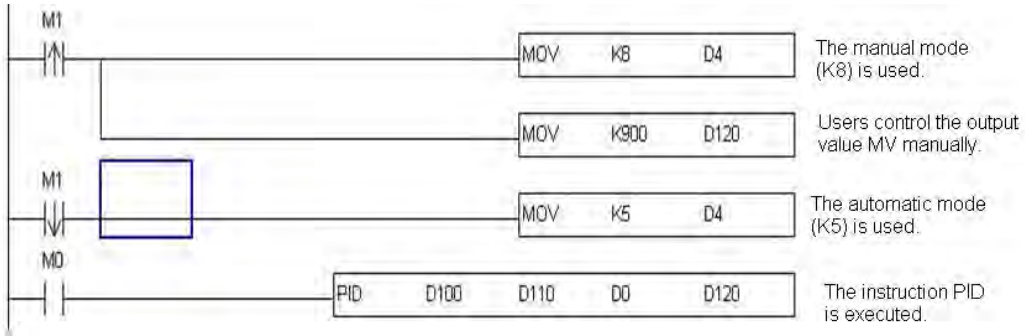
Example 6: Switching between the manual mode (K8) and the automatic mode (K5)

If the setting of the PID parameters is complete, and the control mode is the manual mode (K8), the control curve will be as shown below.



If the control mode becomes the automatic mode (K5), the accumulated integral value will be the integral value converted from the last MV, and the accumulated integral value will be converted into the output value of the PID operation.

The program for example 5 and program 6 are shown below. In the figure below, M0 is a flag for enabling the instruction PID. When M1 is On, the manual mode is used. When M1 is Off, the automatic mode is used.



Application Examples:

Application 1 Using PID instruction in the pressure control system (use the diagram of Example 1).

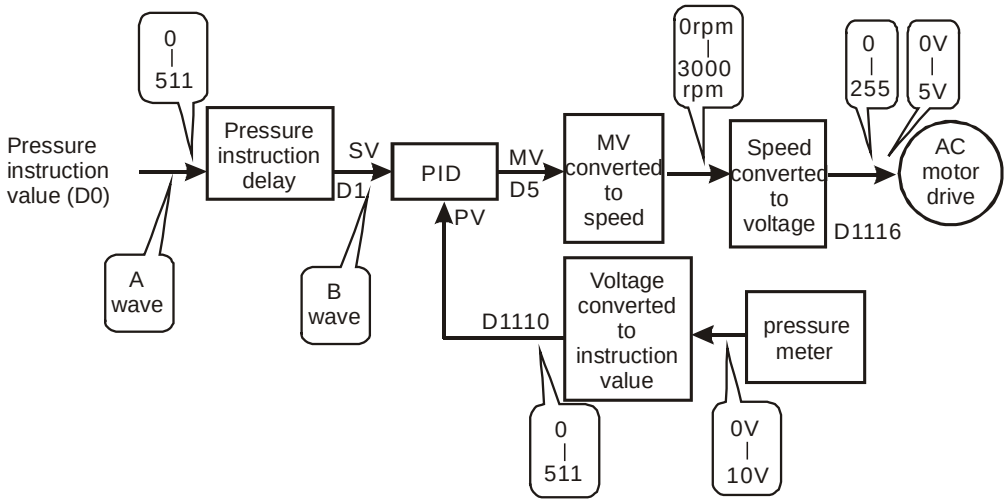
Purpose: Enabling the control system to reach the target pressure.

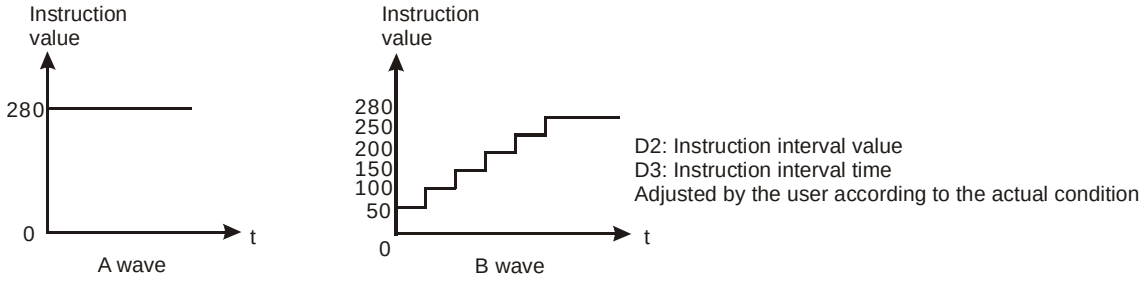
Explanation: The system requires a gradual control. Therefore, the system will be overloaded or out of control if the process progresses too fast.

Suggested solution:

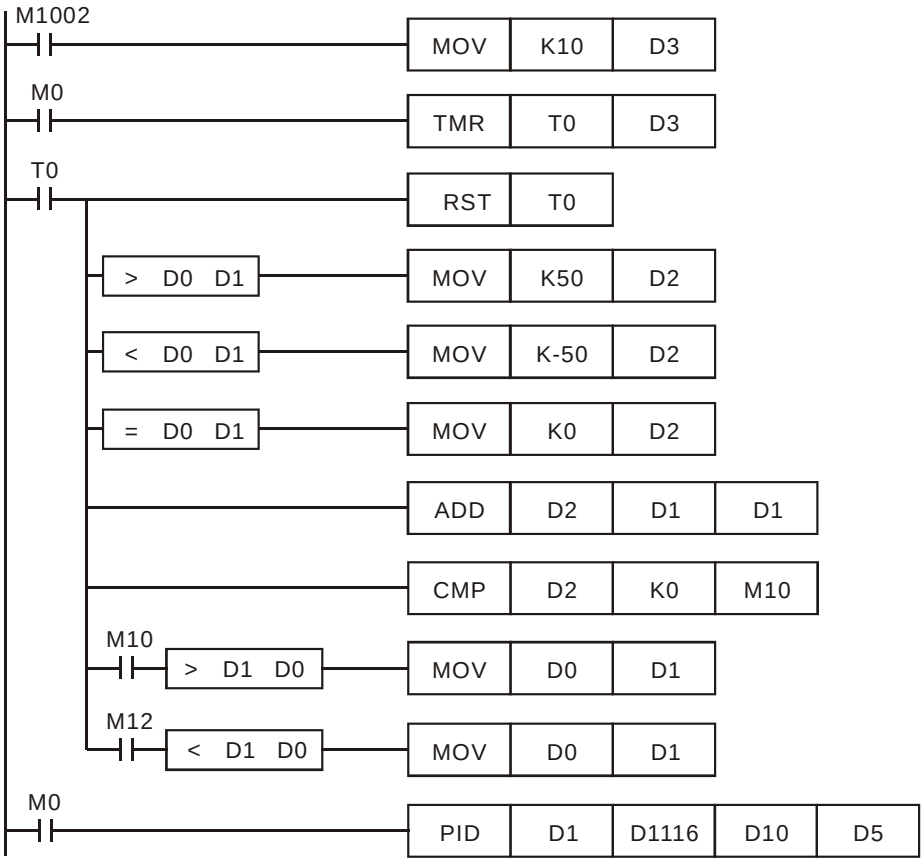
Solution 1: Longer sampling time

Solution 2: Using delay instruction. See the figure below.





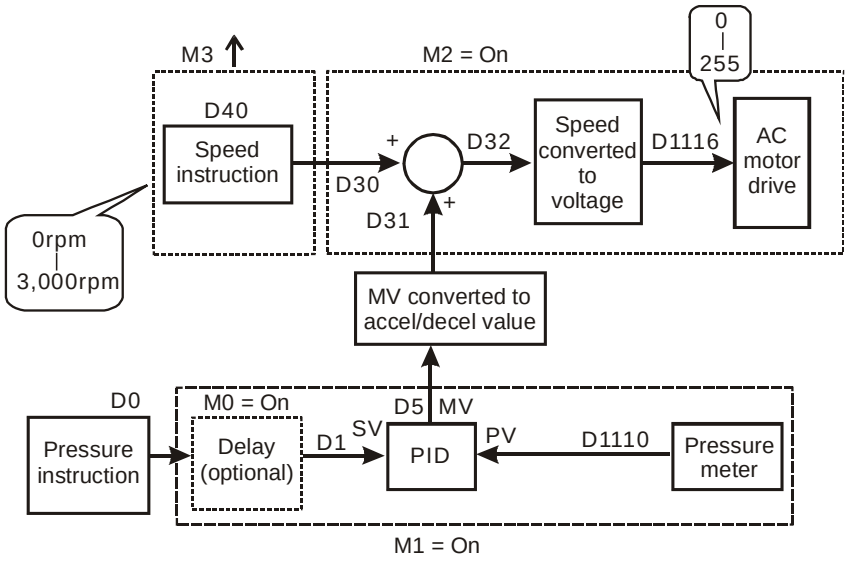
The example program of the instruction delay:



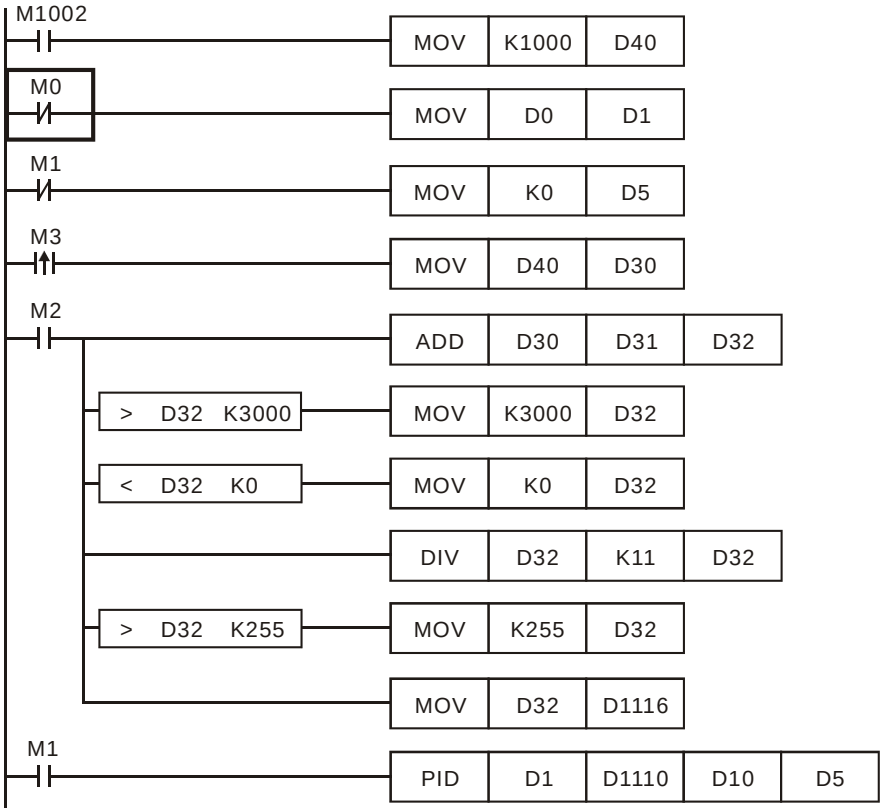
Application 2 Speed control system and pressure control system work individually (use diagram of Example 2).

Purpose: After the speed control operates in open loop for a period of time, adding into it the pressure control system (PID instruction) for close loop control.

Explanation: Since the speed and pressure control systems are not interrelated, we have to structure an open loop for speed control first following by a close loop pressure control. If you fear that the control instruction of the pressure control system changes too fast, you can consider adding the instruction delay illustrated in Application 1 into the control. See the control diagram below.



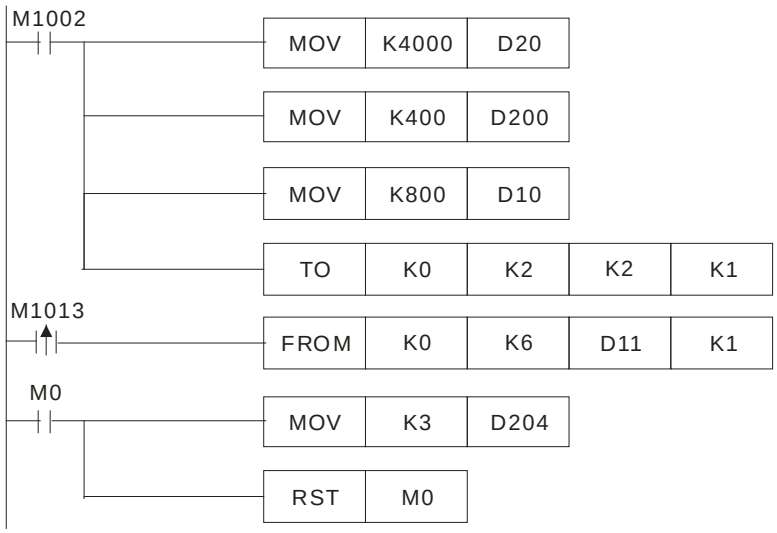
Part of the example program:

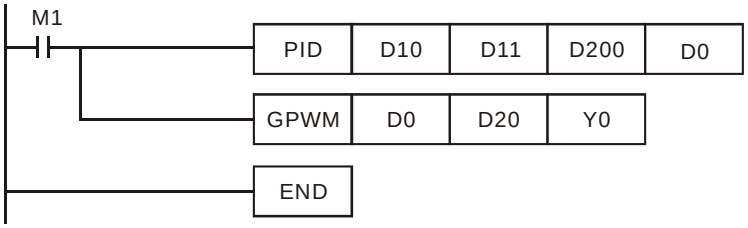


Application 3 Using auto-tuning on the parameter for the temperature control.

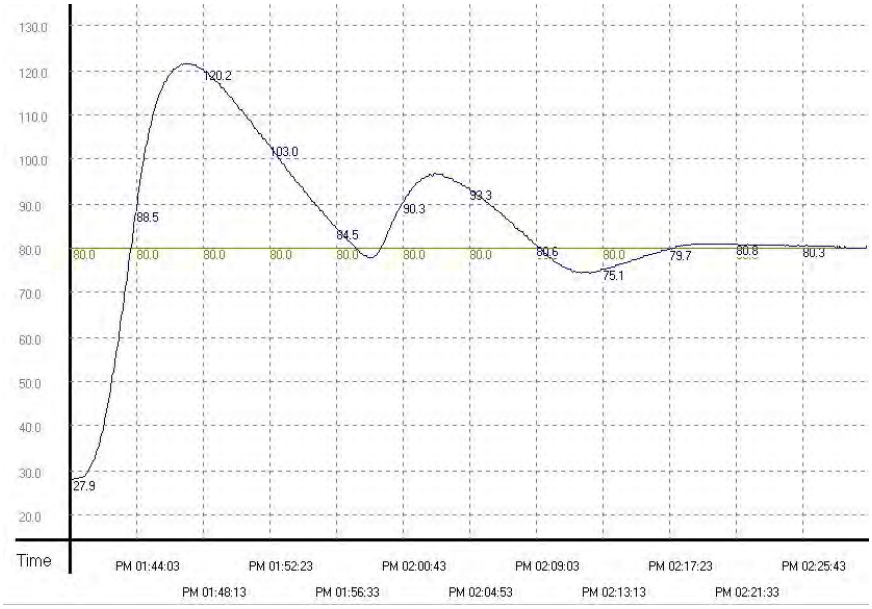
Purpose: Using auto-tuning to calculate the most suitable parameters for PID temperature control.

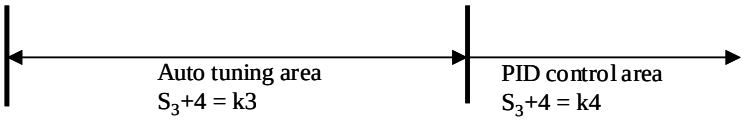
Explanation: You may not be familiar with the temperature environment for the first time, so you can use auto-tuning (**S_3** + 4 = K3) for an initial adjustment. After this, PID instruction will become exclusively for temperature control (**S_3** + 4 = K4). In this example, the control environment is an oven. See the example program below.



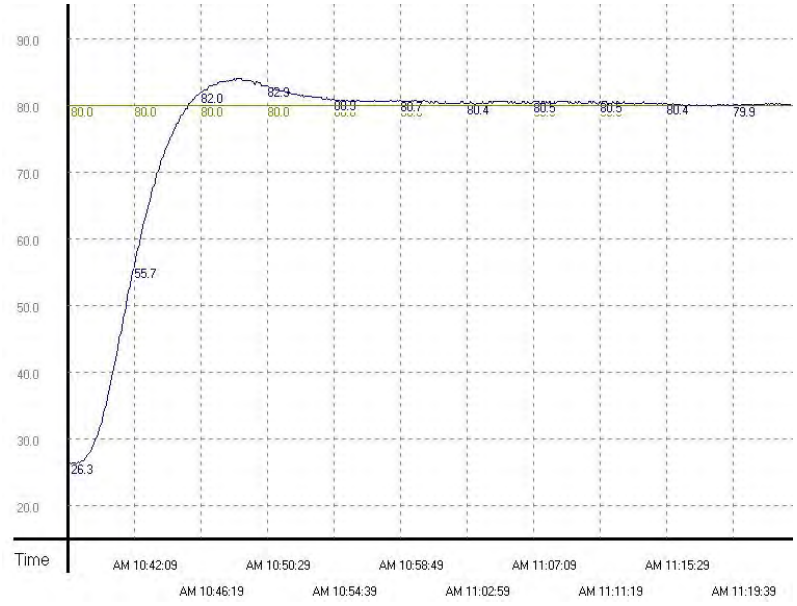


The experiment result of auto-tuning:

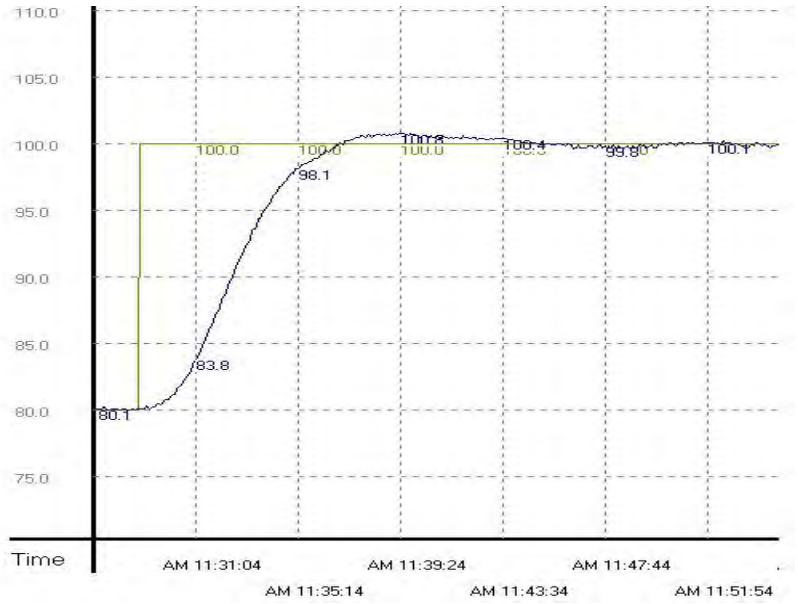




The experiment result of using the adjusted parameter exclusively for temperature control after auto-tuning:



From the figure above, we can see that the temperature control after auto-tuning is working fine and we use only approximately 20 minutes for the control. Next, we modify the target temperature from 80°C to 100°C and obtain the result below.



From the result above, we can see that when the parameter is 100°C, we can still control the temperature without spending too much time.

DVP-PLC applicable to the application instruction. ES includes ES/EX/EC/EC3-8K (FW V8.60 or later) (EC3: FW V8.40 or previous version); SX (FW V3.00); EH3 includes EH3/SV2.

ES/EX/EC series MPU does not support pulse execution type instructions (P instruction).

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	32-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
Communication Instructions	100	MODRD	–	–	Read Modbus Data	✓	✓	✓	✓	7	–
	101	MODWR	–	–	Write Modbus Data	✓	✓	✓	✓	7	–
	102	FWD	–	–	Forward Running of VFD-A	✓	✓	✓	✓	7	–
	103	REV	–	–	Reverse Running of VFD-A	✓	✓	✓	✓	7	–
	104	STOP	–	–	Stop VFD-A	✓	✓	✓	✓	7	–
	105	RDST	–	–	Read VFD-A Status	✓	✓	✓	✓	5	–
	106	RSTEF	–	–	Reset Abnormal VFD-A	✓	✓	✓	✓	5	–
	107	LRC	–	✓	Checksum LRC Mode	✓	✓	✓	✓	7	–
	108	CRC	–	✓	Checksum CRC Mode	✓	✓	✓	✓	7	–
	113	ETHRW	–	–	Reading/Writing through Ethernet	–	–	–	✓	9	–
Arithmetic Instructions	114	MUL16	MUL32	✓	Multiplying binary numbers for 16/32-bit	–	✓	✓	✓	7	13
	115	DIV16	DIV32	✓	Dividing binary numbers for 16/32-bit	–	✓	✓	✓	7	13
Floating Point Operation Instructions	110	–	DECMP	✓	Floating Point Compare	✓	✓	✓	✓	–	13
	111	–	DEZCP	✓	Floating Point Zone Compare	✓	✓	✓	✓	–	17
	112	–	DMOVR	✓	Move Floating Point Data	✓	✓	✓	✓	–	9
	116	–	DRAD	✓	Angle → Radian	–	✓	✓	✓	–	9
	117	–	DDEG	✓	Radian → Angle	–	✓	✓	✓	–	9
	118	–	DEBCD	✓	Float to Scientific Conversion	✓	✓	✓	✓	–	9
	119	–	DEBIN	✓	Scientific to Float Conversion	✓	✓	✓	✓	–	9
	120	–	DEADD	✓	Floating Point Addition	✓	✓	✓	✓	–	13
	121	–	DESUB	✓	Floating Point Subtraction	✓	✓	✓	✓	–	13
	122	–	DEMUL	✓	Floating Point Multiplication	✓	✓	✓	✓	–	13
	123	–	DEDIV	✓	Floating Point Division	✓	✓	✓	✓	–	13
	124	–	DEXP	✓	Exponent of Binary Floating Point	✓	✓	✓	✓	–	9
	125	–	DLN	✓	Natural Logarithm of Binary Floating Point	✓	✓	✓	✓	–	9
	126	–	DLOG	✓	Logarithm of Binary Floating Point	✓	✓	✓	✓	–	13
	127	–	DESQR	✓	Floating Point Square Root	✓	✓	✓	✓	–	9
	128	–	DPOW	✓	Floating Point Power Operation	✓	✓	✓	✓	–	13
	129	INT	DINT	✓	Float to Integer	✓	✓	✓	✓	5	9

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	32-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
Trigonometric function Instructions	130	–	DSIN	✓	Sine	✓	✓	✓	✓	–	9
	131	–	DCOS	✓	Cosine	✓	✓	✓	✓	–	9
	132	–	DTAN	✓	Tangent	✓	✓	✓	✓	–	9
	133	–	DASIN	✓	Arc Sine	–	✓	✓	✓	–	9
	134	–	DACOS	✓	Arc Cosine	–	✓	✓	✓	–	9
	135	–	DATAN	✓	Arc Tangent	–	✓	✓	✓	–	9
	136	–	DSINH	✓	Hyperbolic Sine	–	–	–	✓	–	9
	137	–	DCOSH	✓	Hyperbolic Cosine	–	–	–	✓	–	9
	138	–	DTANH	✓	Hyperbolic Tangent	–	–	–	✓	–	9
Others	109	SWRD	–	✓	Read Digital Switch	–	–	–	✓	3	–
	143	DELAY	–	✓	Delay Instruction	–	✓	✓	✓	3	–
	144	GPWM	–	–	General PWM Output	–	✓	✓	✓	7	–
	145	FTC	–	–	Fuzzy Temperature Control	–	–	✓	✓	9	–
	146	CVM	–	–	Valve Control	–	–	–	✓	7	–
	147	SWAP	DSWAP	✓	Byte Swap	✓	✓	✓	✓	3	5
	148	MEMR	DMEMR	✓	Read File Register	–	–	✓	✓	7	13
	149	MEMW	DMEMW	✓	Write File Register	–	–	✓	✓	7	13

API	Mnemonic	Operands	Function
100	MODRD	(S1) (S2) (n)	Read Modbus Data

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODRD: 7 steps
S_2					*	*							*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

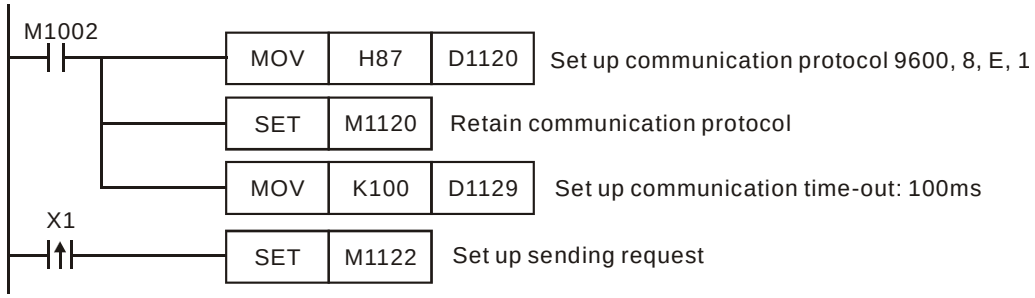
S_1: Address of communication device **S_2**: Address of data to be read **n**: Length of read data

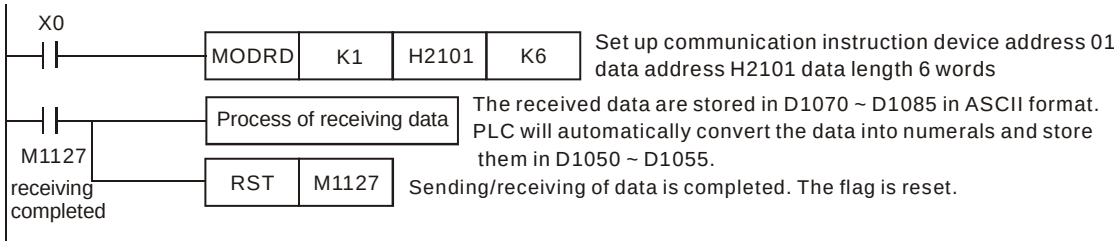
Explanations:

1. Range of **S_1**: K0 ~ K254
2. Range of **n**: K1 ≤ n ≤ K6
3. See the specifications of each model for their range of use.
4. ES/EX/SS series MPU does not support E, F index register modification.
5. Flags: See API 80 RS for explanations on M1120 ~ M1131, M1140 ~ M1143
6. MODRD is a drive instruction exclusively for peripheral communication equipment in MODBUS ASCII mode /RTU mode. The built-in RS-485 communication ports in Delta VFD drives (except for VFD-A series) are all compatible with MODBUS communication format. MODRD can be used for controlling communication (read data) of Delta drives.
7. If the address of **S_2** is illegal to the designed communication device, the device will respond with an error, PLC will records the error code in D1130 and M1141 will be On.
8. The feedback (returned) data from the peripheral equipment will be stored in D1070 ~ D1085. After receiving the feedback data is completed, PLC will auto-check if all data are correct. If there is an error, M1140 will be On.
9. In ASCII mode, due to that the feedback data are all in ASCII, PLC will convert the feedback data into numerals and store them in D1050 ~ D1055. D1050 ~ D1055 will be invalid in RTU mode, but it is available for EH3/SV2 V1.88 or later.
10. After M1140 or M1141 turn On, the program will send a correct datum to the peripheral equipment. If the feedback datum is correct, M1140 and M1141 will be reset.

Program Example 1:

Communication between PLC and VFD-S series AC motor drives (ASCII Mode, M1143 = Off)





PLC ⇨ VFD-S, PLC sends: "**01 03 2101 0006 D4**"

VFD-S ⇨ PLC , PLC receives: "**01 03 0C 0100 1766 0000 0000 0136 0000 3B**"

Registers for sent data (sending messages)

Register	DATA		Explanation	
D1089 low	'0'	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D1089 high	'1'	31 H	ADR 0	
D1090 low	'0'	30 H	CMD 1	Instruction code: CMD (1,0)
D1090 high	'3'	33 H	CMD 0	
D1091 low	'2'	32 H	Starting data address	
D1091 high	'1'	31 H		
D1092 low	'0'	30 H		
D1092 high	'1'	31 H		
D1093 low	'0'	30 H	Number of data (counted by words)	
D1093 high	'0'	30 H		
D1094 low	'0'	30 H		
D1094 high	'6'	36 H		
D1095 low	'D'	44 H	LRC CHK 1	Checksum: LRC CHK (0,1)
D1095 high	'4'	34 H	LRC CHK 0	

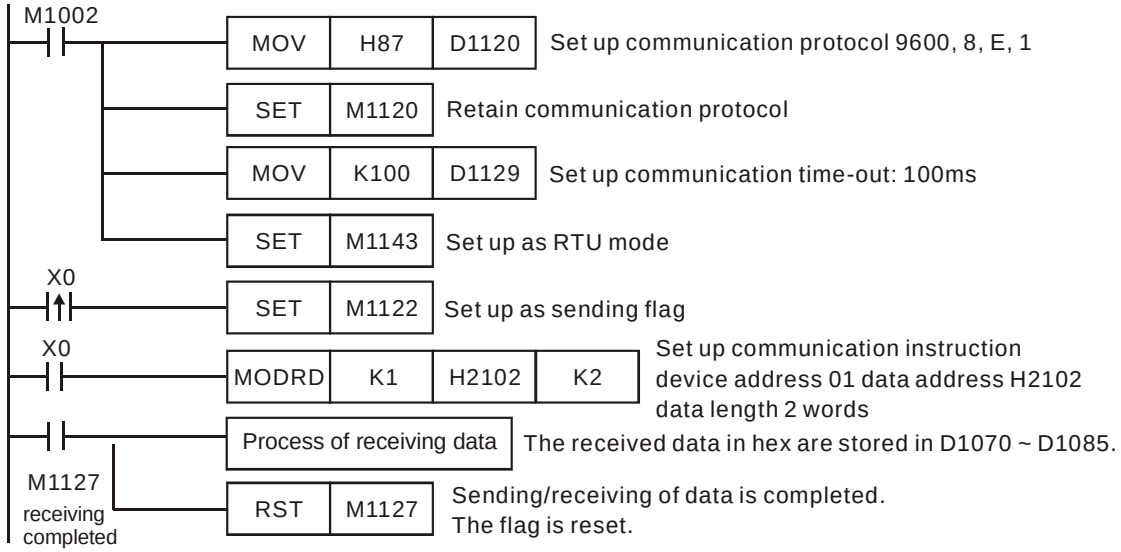
Registers for received data (responding messages)

Register	DATA		Explanation	
D1070 low	'0'	30 H	ADR 1	
D1070 high	'1'	31 H	ADR 0	
D1071 low	'0'	30 H	CMD 1	
D1071 high	'3'	33 H	CMD 0	
D1072 low	'0'	30 H	Number of data (counted by byte)	
D1072 high	'C'	43 H		
D1073 low	'0'	30 H	Content of address 2101 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1050 = 0100 H
D1073 high	'1'	31 H		
D1074 low	'0'	30 H		
D1074 high	'0'	30 H		
D1075 low	'1'	31 H	Content of address 2102 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1051 = 1766 H
D1075 high	'7'	37 H		
D1076 low	'6'	36 H		
D1076 high	'6'	36 H		
D1077 low	'0'	30 H	Content of address 2103 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1052 = 0000 H
D1077 high	'0'	30 H		
D1078 low	'0'	30 H		
D1078 high	'0'	30 H		
D1079 low	'0'	30 H	Content of address 2104 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1053 = 0000 H
D1079 high	'0'	30 H		
D1080 low	'0'	30 H		
D1080 high	'0'	30 H		
D1081 low	'0'	30 H	Content of address 2105 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1054 = 0136 H
D1081 high	'1'	31 H		
D1082 low	'3'	33 H		
D1082 high	'6'	36 H		

Register	DATA		Explanation	
D1083 low	'0'	30 H	Content of address 2106 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1055 = 0000 H
D1083 high	'0'	30 H		
D1084 low	'0'	30 H		
D1084 high	'0'	30 H		
D1085 low	'3'	33 H	LRC CHK 1	
D1085 high	'B'	42 H	LRC CHK 0	

Program Example 2:

Communication between PLC and VFD-S series AC motor drives (RTU Mode, M1143 = On)



PLC ⇨ VFD-S, PLC sends: **01 03 2102 0002 6F F7**

VFD-S ⇨ PLC, PLC receives: **01 03 04 1770 0000 FE 5C**

Registers for sent data (sending messages)

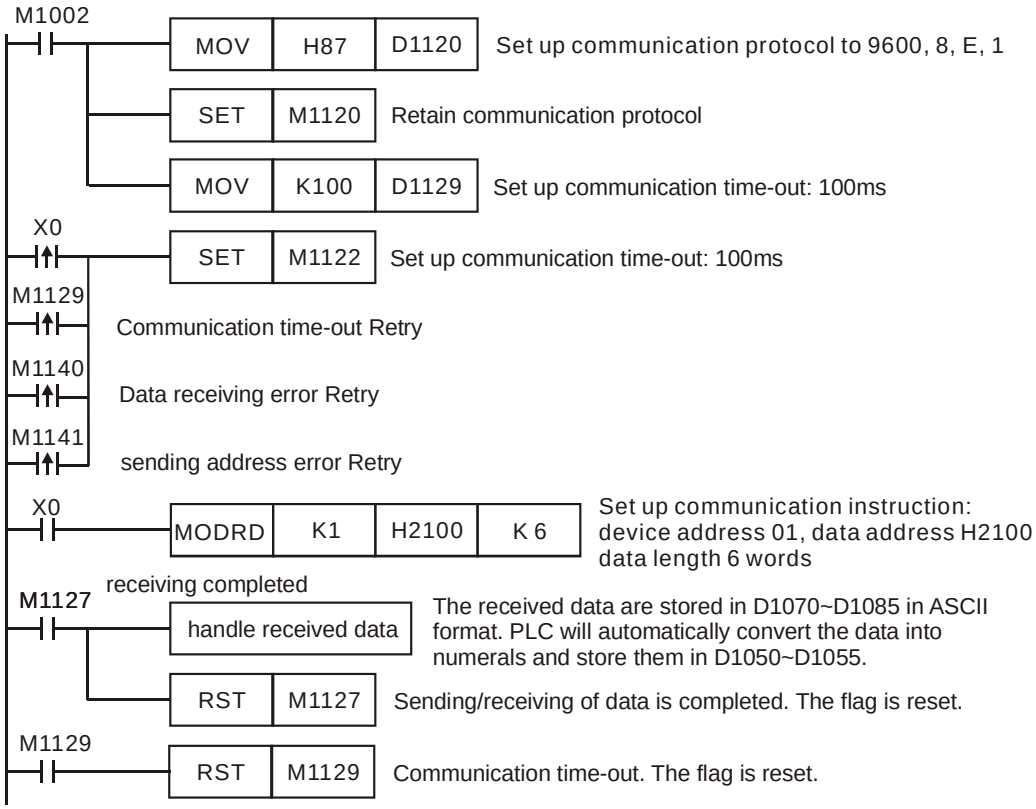
Register	DATA	Explanation
D1089 low	01 H	Address
D1090 low	03 H	Function
D1091 low	21 H	Starting data address
D1092 low	02 H	
D1093 low	00 H	Number of data (counted by words)
D1094 low	02 H	
D1095 low	6F H	CRC CHK Low
D1096 low	F7 H	CRC CHK High

Registers for received data (responding messages)

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	03 H	Function
D1072 low	04 H	Number of data (counted by bytes)
D1073 low	17 H	Content of address 2102 H
D1074 low	70 H	
D1075 low	00 H	Content of address 2103 H
D1076 low	00 H	
D1077 low	FE H	CRC CHK Low
D1078 low	5C H	CRC CHK High

Program Example 3:

1. In the communication between PLC and VFD-S series AC motor drive (ASCII Mode, M1143 = Off), retry when communication time-out, data receiving error and sending address error occur.
2. When X0=On, PLC will read the data in VFFD-S data adress H2100 of device 01 and stores the data in ASCII format in D1070 ~ D1085. PLC will automatically convert the data into numerals and stores them in D1050 ~ D1055.
3. M1129 will be On when communication time-out occurs. The program will trigger M1129 and send request to M1122 for reading the data again.
4. M1140 will be On when data receiving error occurs. The program will trigger M1140 and send request to M1122 for reading the data again.
5. M1141 will be On when sending address error occurs. The program will trigger M1141 and send request to M1122 for reading the data again.



Remarks:

1. The activation criteria placed before the three instructions, API 100 MODRD, API 105 RDST, and API 150 MODRW (Function Code H03), cannot use rising-edge contacts (LDP, ANDP ORP) and falling-edge contacts (LDF, ANDF, ORF); otherwise, the data stores in the receiving registers will be incorrect.
2. M1127 for MODRD instruction stands for the response of data is completed. M1127 will only be On if the responded data are correct. M1123 will be On no matter the responded data are correct or wrong.
3. There is no limitation on the times of using this instruction in the program, but only one instruction is allowed to be executed at a time.

API	Mnemonic	Operands	Function
101	MODWR	(S1) (S2) (n)	Write Modbus Data

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MODWR: 7 steps
S_1					*	*							*			
S_2					*	*							*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

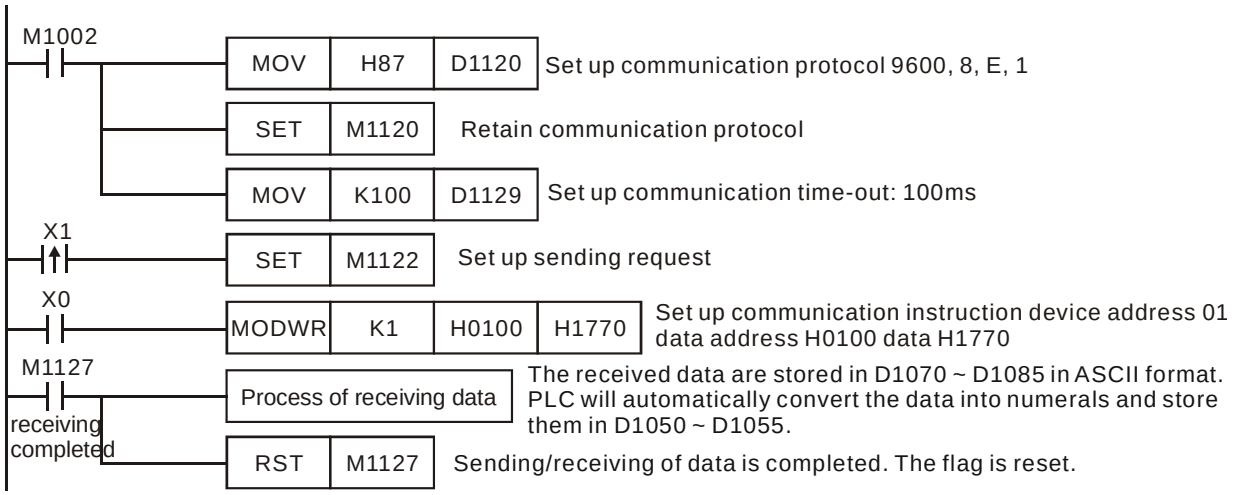
S_1: Address of communication device **S_2**: Address of data to be read **n**: Data to be written

Explanations:

1. Range of **S_1**: K0 ~ K254
2. See the specifications of each model for their range of use.
3. ES/EX/SS series MPU does not support E, F index register modification.
4. Flags: See API 80 RS for explanations on M1120 ~ M1131, M1140 ~ M1143
5. MODWR is a drive instruction exclusively for peripheral communication equipment in MODBUS ASCII mode/RTU mode. The built-in RS-485 communication ports in Delta VFD drives (except for VFD-A series) are all compatible with MODBUS communication format. MODRD can be used for controlling communication (write data) of Delta drives.
6. If the address of **S_2** is illegal to the designed communication device, the device will respond with an error, PLC will records the error code in D1130 and M1140 will be On. For example, if 8000H is illegal to VFD-S, M1141 will be On and D1130 = 2. For error codes, see the user manual of VFD-S.
7. The feedback (returned) data from the peripheral equipment will be stored in D1070 ~ D1076. After receiving the feedback data is completed, PLC will auto-check if all data are correct. If there is an error, M1140 will be On.
8. After M1140 or M1141 turn On, the program will send a correct datum to the peripheral equipment. If the feedback datum is correct, M1140 and M1141 will be reset.

Program Example 1:

Communication between PLC and VFD-S series AC motor drives (ASCII Mode, M1143 = Off)



PLC ⇨ VFD-B, PLC sends: " **01 06 0100 1770 71** "

VFD-B ⇨ PLC, PLC receives: " **01 06 0100 1770 71** "

Registers for sent data (sending messages)

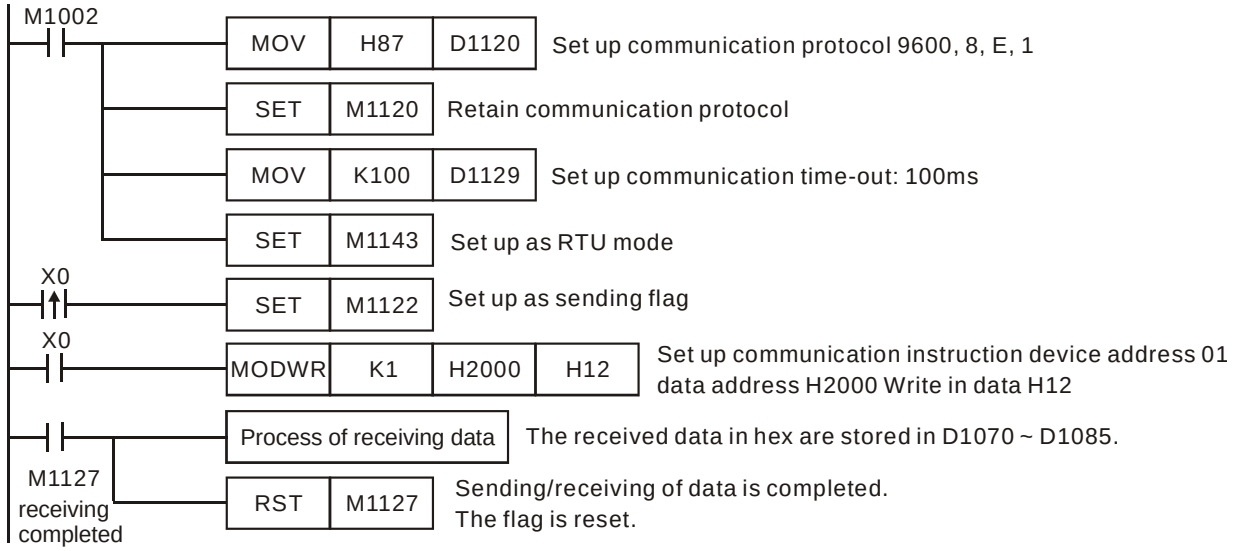
Register	DATA		Explanation	
D1089 low	'0'	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D1089 high	'1'	31 H	ADR 0	
D1090 low	'0'	30 H	CMD 1	Instruction code: CMD (1,0)
D1090 high	'6'	36 H	CMD 0	
D1091 low	'0'	30 H	Data address	
D1091 high	'1'	31 H		
D1092 low	'0'	30 H		
D1092 high	'0'	30 H		
D1093 low	'1'	31 H	Data contents	
D1093 high	'7'	37 H		
D1094 low	'7'	37 H		
D1094 high	'0'	30 H		
D1095 low	'7'	37 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1095 high	'1'	31 H	LRC CHK 0	

PLC receiving data register (response messages)

Register	DATA		Explanation
D1070 low	'0'	30 H	ADR 1
D1070 high	'1'	31 H	ADR 0
D1071 low	'0'	30 H	CMD 1
D1071 high	'6'	36 H	CMD 0
D1072 low	'0'	30 H	Data address
D1072 high	'1'	31 H	
D1073 low	'0'	30 H	
D1073 high	'0'	30 H	
D1074 low	'1'	31 H	Data content
D1074 high	'7'	37 H	
D1075 low	'7'	37 H	
D1075 high	'0'	30 H	
D1076 low	'7'	37 H	LRC CHK 1
D1076 high	'1'	31 H	LRC CHK 0

Program Example 2:

Communication between PLC and VFD-S series AC motor drives (RTU Mode, M1143 = On)



PLC ⇨ VFD-S, PLC sends: **01 06 2000 0012 02 07**

VFD-S ⇨ PLC, PLC receives: **01 06 2000 0012 02 07**

Registers for sent data (sending messages)

Register	DATA	Explanation
D1089 low	01 H	Address
D1090 low	06 H	Function
D1091 low	20 H	Data address
D1092 low	00 H	
D1093 low	00 H	Data contents
D1094 low	12 H	
D1095 low	02 H	CRC CHK Low
D1096 low	07 H	CRC CHK High

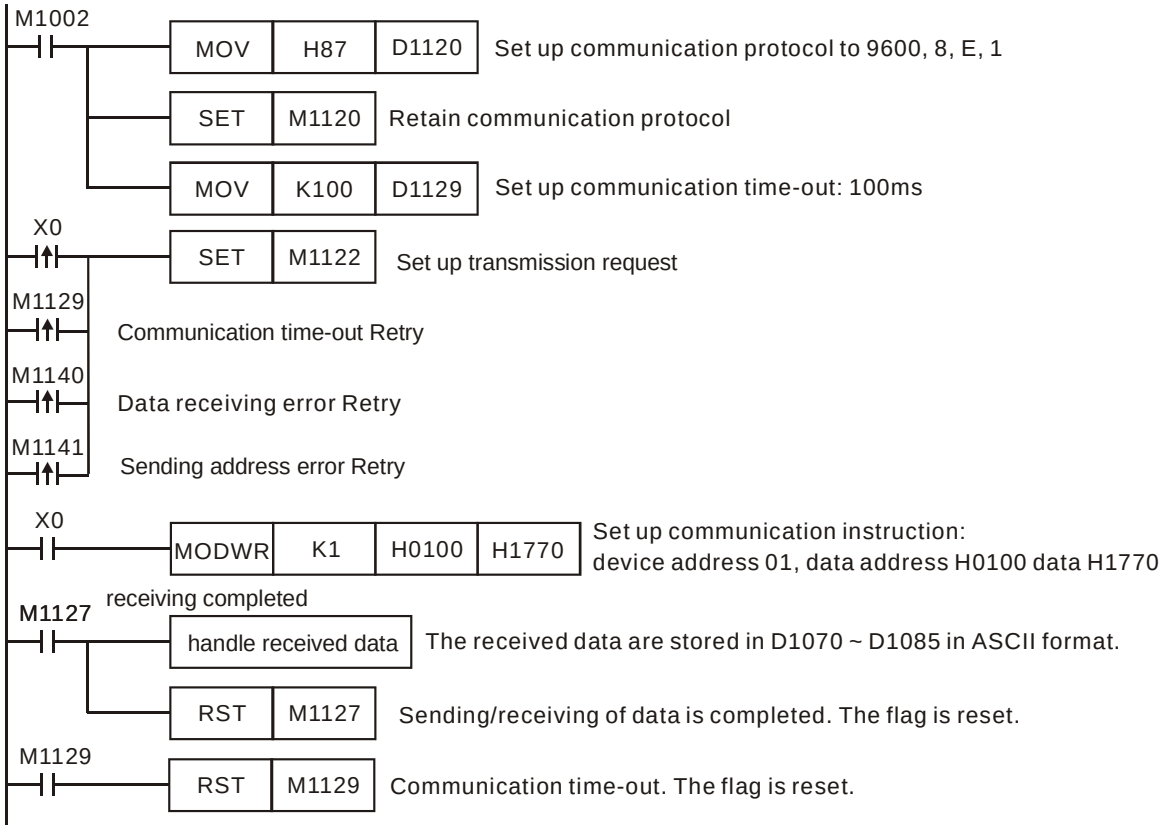
Registers for received data (responding messages)

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	06 H	Function
D1072 low	20 H	Data address
D1073 low	00 H	
D1074 low	00 H	Data contents
D1075 low	12 H	
D1076 low	02 H	CRC CHK Low
D1077 low	07 H	CRC CHK High

Program Example 3:

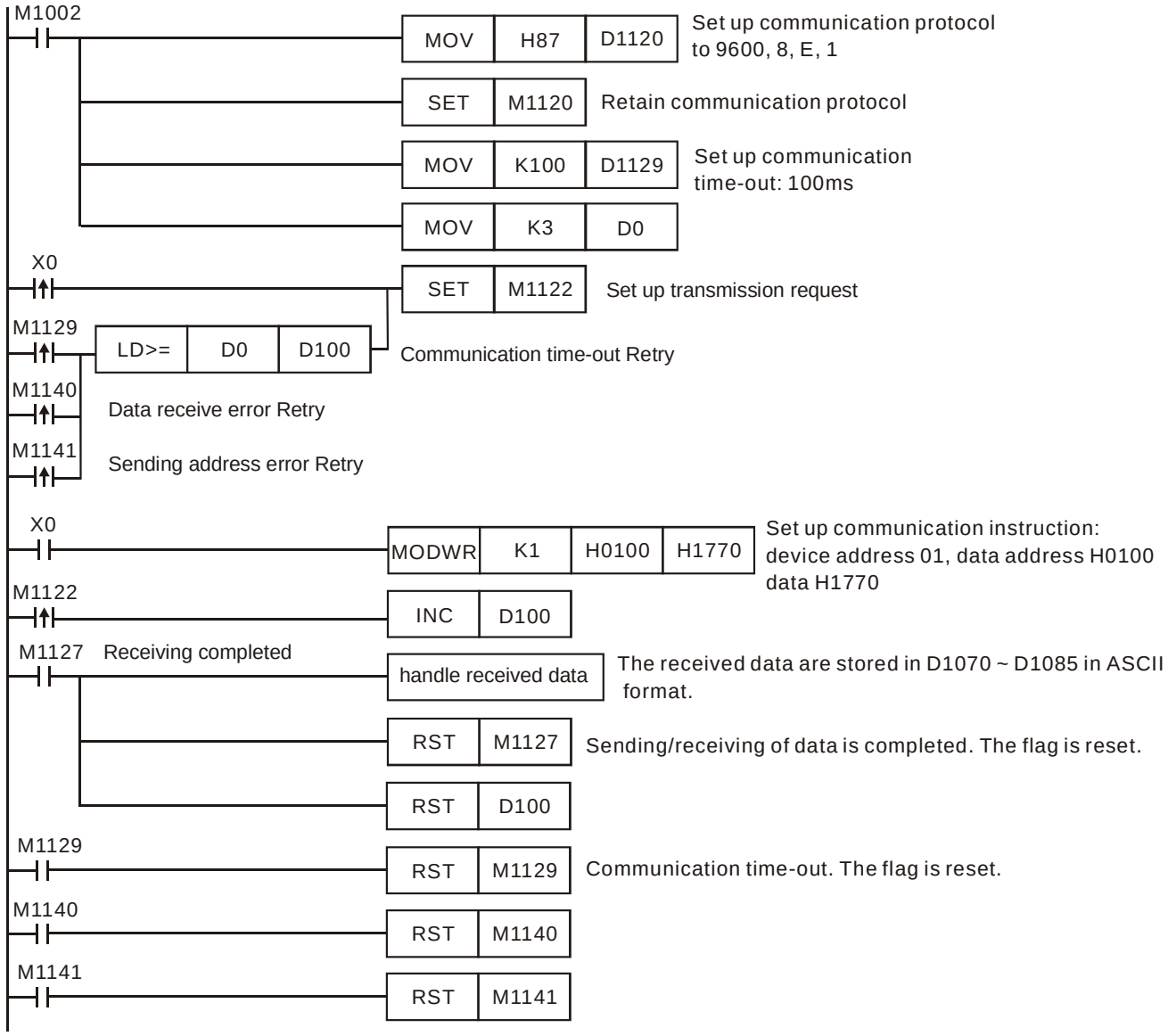
1. In the communication between PLC and VFD-S series AC motor drive (ASCII Mode, M1143 = Off), retry when communication time-out, data receiving error and sending address error occur.
2. When X0 = On, PLC will write H1770(K6000) into VFD-S data adress H0100 of device 01.
3. M1129 will be On when communication time-out occurs. The program will trigger M1129 and send request to M1122 for writing the data again.
4. M1140 will be On when data receiving error occurs. The program will trigger M1140 and send request to M1122 for writing the data again.

5. M1141 will be On when sending address error occurs. The program will trigger M1141 and send request to M1122 for writing the data again.



Program Example 4:

1. In the communication between PLC and VFD-S series AC motor drive (ASCII Mode, M1143 = Off), retry when communication time-out, data receiving error and sending address error occur. Times of retry = D0 (default = 3). When communication Retry is successful, the user can return to controlling by triggering criteria.
2. When X0 = On, PLC will write H1770(K6000) into VFD-S data adress H0100 of device 01.
3. M1129 will be On when communication time-out occurs. The program will trigger M1129 and send request to M1122 for writing the data again. Times of Retry = D0 (default = 3)
4. M1140 will be On when data receiving error occurs. The program will trigger M1140 and send request to M1122 for writing the data again. Times of Retry = D0 (default = 3)
5. M1141 will be On when sending address error occurs. The program will trigger M1141 and send request to M1122 for writing the data again. Times of Retry = D0 (default = 3)



Remarks:

1. For the registers for flag settings, see explanations in API 80 RS.
2. M1127 for MODWR instruction stands for the response of data is completed. M1127 will only be On if the responded data are correct. M1123 will be On no matter the responded data are correct or wrong.
3. There is no limitation on the times of using this instruction in the program, but only one instruction is allowed to be executed at a time.

API	Mnemonic	Operands	Function
102	FWD	(S1) (S2) (n)	Forward Running of VFD-A

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			FWD: 7 steps
S_2					*	*							*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

API	Mnemonic	Operands	Function	Controllers		
103	REV	(S1) (S2) (n)	Reverse Running of VFD-A	ES/EX/SS	SX	EH/SV

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			REV: 7 steps
S_2					*	*							*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

API	Mnemonic	Operands	Function	Controllers		
104	STOP	(S1) (S2) (n)	Stop VFD-A	ES/EX/SS	SX	EH/SV

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			STOP: 7 steps
S_2					*	*							*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Address of communication device **S_2**: Rotation frequency of AC motor drive **n**: Target to be instructed

Explanations:

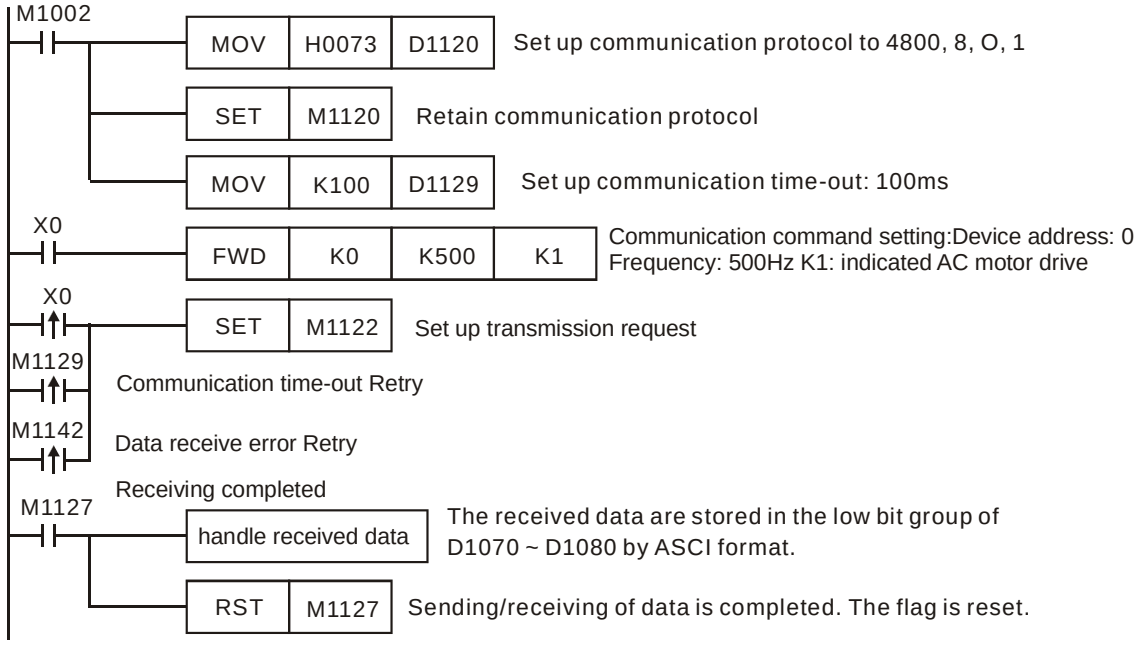
1. Range of **S_1**: K0 ~ K31
2. Range of **n**: K1 or K2
3. See the specifications of each model for their range of use.
4. ES series MPU does not support E, F index register modification.
5. Flags: See API 80 RS for explanations on M1120 ~ M1131, M1140 ~ M1143
6. FWD/REV/STOP are handy instructions exclusively for Delta VFD-A/H series AC motor drive to perform forward running/reverse running/stop. Be sure to set up communication time-out (D1129) when executing this instruction.
7. **S_2** = operation frequency of AC motor drive. Set frequency in A-series AC motor drive: K0 ~ K4,000 (0.0Hz ~

400.0Hz). Set frequency in H-series: K0 ~ K1,500 (0Hz ~ 1,500Hz).

8. **n** = instructed target. **n**=1: AC motor drive at designated address. **n**=2: all connected AC motor drives.
9. The feedback (returned) data from the peripheral equipment will be stored in D1070 ~ D1080. After receiving the feedback data is completed, PLC will auto-check if all data are correct. If there is an error, M1142 will be On. When **n** = 2, PLC will not receive any data.

Program Example:

Communication between PLC and VFD-A series AC drives, retry for communication time-out and received data error.



PLC ⇨ VFD-A, PLC sends: "**C ♥ ☺ 0001 0500** "

VFD-A ⇨ PLC, PLC sends: "**C ♥ ♠ 0001 0500** "

Registers for sent data (sending messages)

Register	DATA		Explanation
D1089 low	'C'	43 H	Start word of instruction
D1090 low	'♥'	03 H	Checksum
D1091 low	'☺'	01 H	Instructed target
D1092 low	'0'	30 H	Communication address
D1093 low	'0'	30 H	
D1094 low	'0'	30 H	
D1095 low	'1'	31 H	
D1096 low	'0'	30 H	Running instruction
D1097 low	'5'	35 H	
D1098 low	'0'	30 H	
D1099 low	'0'	30 H	

Registers for received data (responding messages)

Register	DATA		Explanation
D1070 low	'C'	43 H	Start word of instruction
D1071 low	'♥'	03 H	Checksum
D1072 low	'♠'	06 H	Reply authorization (correct: 06H, incorrect: 07 H)
D1073 low	'0'	30 H	Communication address
D1074 low	'0'	30 H	

Register	DATA		Explanation
D1075 low	‘0’	30 H	
D1076 low	‘1’	31 H	
D1077 low	‘0’	30 H	Running instruction
D1078 low	‘5’	35 H	
D1079 low	‘0’	30 H	
D1080 low	‘0’	30 H	

Remarks:

There is no limitation on the times of using this instruction in the program, but only one instruction is allowed to be executed at a time.

API	Mnemonic	Operands	Function
105	RDST	(S) (n)	Read VFD-A Status

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	RDST: 5 steps
S					*	*							*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Address of communicatino device **n**: Target to be instructed

Explanations:

1. Range of **S**: K0 ~ K31
2. Range of **n**: K0 ~ K3
3. See the specifications of each model for their range of use.
4. ES series MPU does not support E, F index register modification.
5. Flags: See API 80 RS for explanations on M1120 ~ M1131, M1140 ~ M1143
6. **n**: Instructed target (to be read) in AC motor drive
   n=0, frequency
   n=1, output frequency
   n=2, output current
   n=3, running instruction
7. Data sent back (feedback) from AC motor drive (11 bytes, see VFD-A user manual) are stored in the low bytes of D1070 ~ D1080.

   ”Q, S, B, Uu, Nn, ABCD”

Feedback	Explanation		Data storage
Q	Start word: ’Q’ (51H).		D1070 low
S	Checksum code: 03H.		D0171 low
B	Instruction authorization. correct: 06H, incorrect: 07H.		D1072 low
U	Communication address (address: 00~31). ”Uu” = (“00” ~ “31”) indicated in ASCII format.		D1073 low
U			D1074 low
N	Instructed target (00 ~ 03).”Nn” = (“00 ~ 03”) indicated in ASCII format.		D1075 low
N			D1076 low
A	Instructed data. The content of “ABCD” differs upon the instructed targets (00 ~ 03). 00 ~ 03 indicate frequency, current and running mode respectively. Please refer to the explanations below for details.		D1077 low
B			D1078 low
C			D1079 low
D			D1080 low
	Nn = “00”	Frequency instruction = ABC.D (Hz)	
	Nn = “01”	Output instruction = ABC.D (Hz)	
	Nn = “02”	Output current = ABC.D (A)	
	PLC will automatically convert the ASCII characters of “ABCD” into numerals and store the numeral in D1050. For example, assume “ABCD” = “0600”, PLC will convert ABCD into K0600 (0258 H) and store it in the special register D1050.		

Feedback	Explanation									Data storage
	Nn = “03”	Running instruction								
	‘A’ =	‘0’	Stop,		‘5’	JOG (forward)				
		‘1’	Forward running		‘6’	JOG (reverse)				
		‘2’	Stop,		‘7’	JOG (reverse)				
		‘3’	Reverse running		‘8’	Abnormal				
		‘4’	JOG (forward),							
		ES series PLCs will convert the ASCII characters of “A” into a numeral and store the numeral in D1051. For example, assume “A” = “3”, PLC will convert A into K3 and store it in the special register D1051. SA/EH series PLCs will store the numeral in low bytes of D1051.								
	‘B’ =	b7	b6	b5	b4	Source of running instruction				
		0	0	0	0	Digital keypad				
		0	0	0	1	1st Step Speed				
		0	0	1	0	2nd Step Speed				
		0	0	1	1	3rd Step Speed				
		0	1	0	0	4th Step Speed				
		0	1	0	1	5th Step Speed				
		0	1	1	0	6th Step Speed				
		0	1	1	1	7th Step Speed				
		1	0	0	0	JOG frequency				
		1	0	0	1	Analog signal frequency instruction				
		1	0	1	0	RS-485 communication interface				
		1	0	1	1	Up/Down control				
		b3	=	0	No DC braking stop		1	DC braking stop		
		b2	=	0	No DC braking startup		1	DC braking startup		
		b1	=	0	Forward running		1	Reverse running		
		b0	=	0	Stop		1	Running		
		ES series PLCs will store “B” in special auxiliary relay M1168 (b0) ~ M1175 (b7). SA/EH series PLCs will store “B” (in hex) in the high bytes of special register D1051.								
	“CD” =	“00”	No abnormal record			“10”	OcA			
		“01”	oc			“11”	Ocd			
		“02”	ov			“12”	Ocn			
		“03”	oH			“13”	GFF			
		“04”	oL			“14”	Lv			
		“05”	oL1			“15”	Lv1			
		“06”	EF			“16”	cF2			
		“07”	cF1			“17”	bb			
		“08”	cF3			“18”	oL2			
		“09”	HPF			“19”				
		ES/SA/EH series PLCs will convert the ASCII characters of “CD” into a numerals and store the numeral in D1052. For example, assume “CD” = “16”, PLC will convert CD into K16 and store it in the special register D1052.								

Remarks:

1. The activation criteria placed before the three instructions, API 100 MODRD, API 105 RDST and API 150 MODRW (Function Code 03), cannot use rising-edge contacts (LDP, ANDP ORP) and falling-edge contacts (LDF, ANDF, ORF); otherwise, the data stores in the receiving registers will be incorrect.
2. For the registers for flag settings, see explanations in API 80 RS.
3. There is no limitation on the times of using this instruction in the program, but only one instruction is allowed to be executed at a time.

API	Mnemonic	Operands	Function
106	RSTEF	(S) (n)	Reset Abnormal VFD-A

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			RSTEF: 5 steps
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Address of communication device **n**: Target to be instructed

Explanations:

1. Range of **S**: K0 ~ K31
2. Range of **n**: K1 or K2
3. See the specifications of each model for their range of use.
4. Flags: See API 80 RS for explanations on M1120 ~ M1131, M1140 ~ M1143
5. RSTEF is a handy communication instruction exclusively for Delta VFD-A series AC motor drives and is used for reset when the AC motor drive operates abnormally.
6. **n**: instructed target. **n**=1: AC motor drive at assigned address. **n**=2: all connected AC motor drives.
7. The feedback (returned) data from the peripheral equipment will be stored in D1070 ~ D1089. If **n** = 2, there will be no feedback data.

Remarks:

1. The activation criteria placed before the three instructions, API 100 MODRD, API 105 RDST and API 150 MODRW (Function Code 03), cannot use rising-edge contacts (LDP, ANDP ORP) and falling-edge contacts (LDF, ANDF, ORF); otherwise, the data stores in the receiving registers will be incorrect.
2. For the registers for flag settings, see explanations in API 80 RS.
3. There is no limitation on the times of using this instruction in the program, but only one instruction is allowed to be executed at a time.

API	Mnemonic			Operands	Function
107		LRC	P	(S) (n) (D)	Checksum LRC Mode

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			LRC, LRCP: 7 steps
n					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

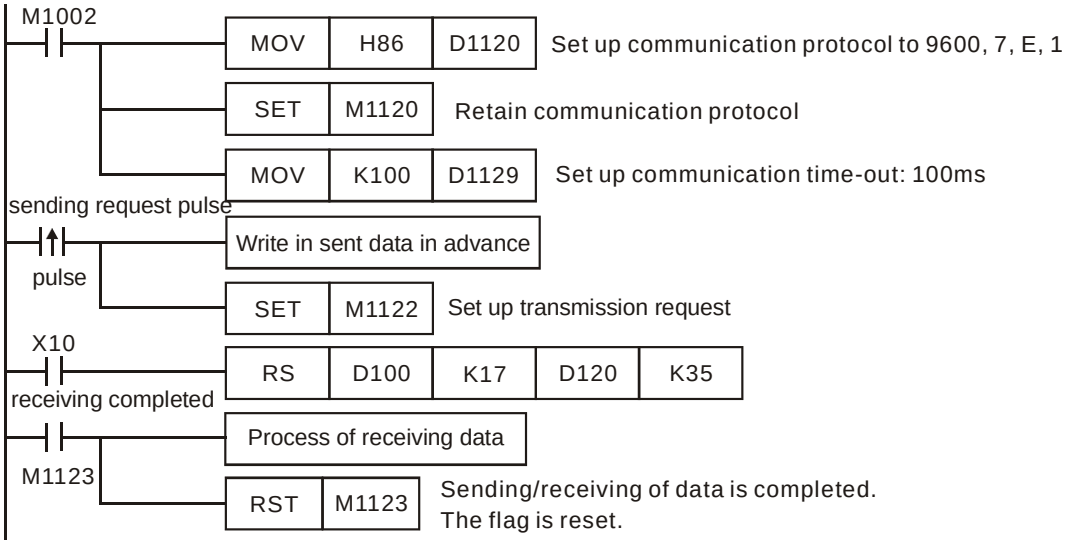
S: Start operation device for ASCII mode checksum **n**: Number of calculated bits **D**: Start device for storing the operation result LRC checksum: See remarks.

Explanations:

1. Range of **n**: K1 ~ K256
2. See the specifications of each model for their range of use.
3. Flag: M1161 (switching between 8/16 bit modes)
4. **n** has to be even. If **n** does not fall within its range, an operation error will occur, the instruction will not be executed, M1067, M1068 = On and D1067 will record the error code H'0E1A.
5. In 16-bit conversion mode: When M1161 = Off, **S** divides its hex data area into higher 8 bits and lower 8 bits and performs LRC checksum operation on each bit. The data will be sent to the higher 8 bits and lower 8 bits in **D**. **n** = the number of calculated bits.
6. In 8-bit conversion mode: When M1161 = On, **S** divides its hex data area into higher 8 bits (invalid data) and lower 8 bits and performs LRC checksum operation on each bit. The data will be sent to the lower 8 bits in **D** and occupy 2 registers. **n** = the number of calculated bits. (All higher bits in **D** are "0".)

Program Example:

When PLC communicates with VFD-S series AC motor drives (In ASCII mode, M1143 = Off), (In 8-bit mode, M1161 = On), the sent data write in advance the 6 data read starting from H2101 of VFD-S.

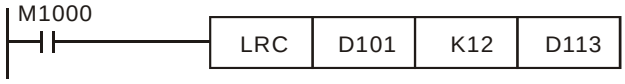


PLC ⇨ VFD-S, PLC sends: **“: 01 03 2101 0006 D4 CR LF ”**

Registers for sent data (sending messages)

Register	DATA		Explanation	
D100 low	‘: ’	3A H	STX	
D101 low	‘0’	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D102 low	‘1’	31 H	ADR 0	
D103 low	‘0’	30 H	CMD 1	Instruction code: CMD (1,0)
D104 low	‘3’	33 H	CMD 0	
D105 low	‘2’	32 H	Starting data address	
D106 low	‘1’	31 H		
D107 high	‘0’	30 H		
D108 low	‘1’	31 H		
D109 low	‘0’	30 H	Number of data (counted by words)	
D110 low	‘0’	30 H		
D111 low	‘0’	30 H		
D112 low	‘6’	36 H		
D113 low	‘D’	44 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D114 low	‘4’	34 H	LRC CHK 0	
D115 low	CR	A H	END	
D116 low	LF	D H		

The error checksum LRC CHK (0,1) can be calculated by LRC instruction (in 8-bit mode, M1161 = On).



LRC checksum: 01 H + 03 H + 21 H + 01 H + 00 H + 06 H = 2C H. Obtain 2’s complement, D4H, and store ‘D’(44H) in the lower 8 bits of D113 and ‘4’(34H) in the lower 8 bits of D114.

Remarks:

1. The format of ASCII mode with a communication datum

STX	‘: ’	Start word = ‘: ’ (3AH)
Address Hi	‘ 0 ’	Communication:
Address Lo	‘ 1 ’	8-bit address consists of 2 ASCll codes
Function Hi	‘ 0 ’	Function code:
Function Lo	‘ 3 ’	8-bit function consists of 2 ASCll codes
DATA (n-1)	‘ 2 ’	Data content:
…….	‘ 1 ’	n × 8-bit data consists of 2n ASCll codes
DATA 0	‘ 0 ’	
	‘ 2 ’	
	‘ 0 ’	
	‘ 0 ’	
	‘ 0 ’	
	‘ 2 ’	
LRC CHK Hi	‘ D ’	LRC checksum:
LRC CHK Lo	‘ 7 ’	8-bit checksum consists of 2 ASCll codes
END Hi	CR	End word:
END Lo	LF	END Hi = CR (0DH), END Lo = LF(0AH)

2. LRC checksum: 2’s complement of the summed up value of communication address and data. For example, 01 H + 03 H + 21 H + 02 H + 00 H + 02 H = 29 H. Obtain 2’s complement = D7H.

API	Mnemonic		Operands	Function
108	CRC	P	(S) (n) (D)	Checksum CRC Mode

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			CRC, CRCP: 7 steps
n					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

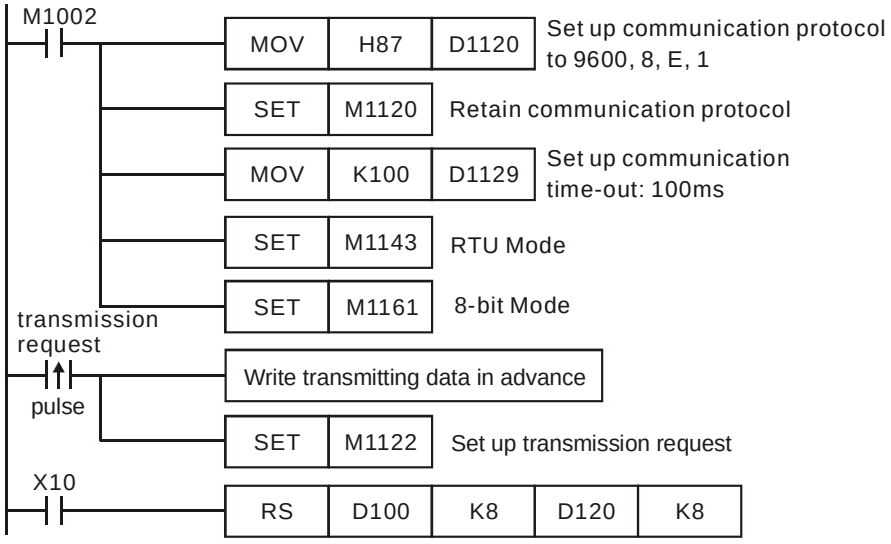
S: Start operation device for RTU mode checksum **n**: Number of calculated bits **D**: Start device for storing the operation result CRC checksum: See remarks.

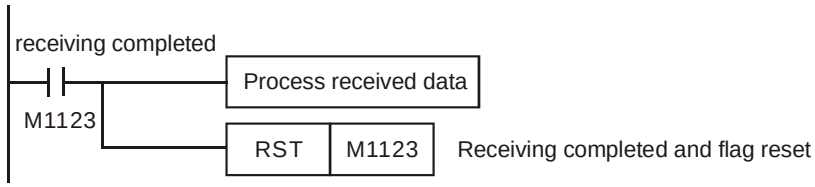
Explanations:

1. Range of **n**: K1 ~ K256
2. Flags: M1161 (switching between 8/16-bit modes)
3. If **n** does not fall within its range, an operation error will occur, the instruction will not be executed, M1067, M1068 = On and D1067 will record the error code H'0E1A.
4. In 16-bit conversion mode: When M1161 = Off, **S** divides its hex data area into higher 8 bits and lower 8 bits and performs CRC checksum operation on each bit. The data will be sent to the higher 8 bits and lower 8 bits in **D**. **n** = the number of calculated bits.
5. In 8-bit conversion mode: When M1161 = On, **S** divides its hex data area into higher 8 bits (invalid data) and lower 8 bits and performs CRC checksum operation on each bit. The data will be sent to the lower 8 bits in **D** and occupy 2 registers. **n** = the number of calculated bits. (All higher 8 bits in **D** are "0".)

Program Example:

When PLC communicates with VFD-S series AC motor drives (In RTU mode, M1143 = On), (In 16-bit mode, M1161 = On), the sent data write in advance H12 into H2000 of VFD-S.



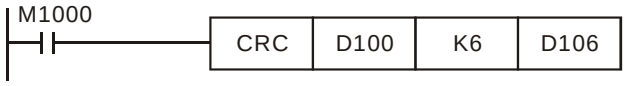


PLC ⇨ VFD-S, PLC sends: **01 06 2000 0012 02 07**

Registers for sent data (sending messages)

Register	DATA	Explanation
D100 low	01 H	Address
D101 low	06 H	Function
D102 low	20 H	Data address
D103 low	00 H	
D104 low	00 H	Data content
D105 low	12 H	
D106 low	02 H	CRC CHK 0
D107 low	07 H	CRC CHK 1

The error checksum CRC CHK (0,1) can be calculated by CRC instruction (in 8-bit mode, M1161 = On).



CRC checksum: 02 H is stored in the lower 8 bits of D106 and 07 H in the lower 8 bits of D107,

Remarks:

1. The format of RTU mode with a communication datum

START	Time interval
Address	Communication address: 8-bit binary
Function	Function code: 8-bit binary
DATA (n-1)	Data content: n × 8-bit data
…….	
DATA 0	
CRC CHK Low	CRC checksum: 16-bit CRC checksum consists of 2 8-bit binaries
CRC CHK High	
END	Time interval

2. CRC checksum starts from Address and ends at Data content.

   The operation of CRC checksum:

   Step 1: Make the 16-bit register (CRC register) = FFFFH

   Step 2: Exclusive OR the first 8-bit byte message instruction and the low-bit 16-bit CRC register. Store the result in CRC register.

   Step 3: Shift the CRC register one bit to the right and fill 0 in the higher bit.

   Step 4: Check the value that shifts to the right. If it is 0, store the new value from Step 3 into the CRC register, otherwise, Exclusive OR A001H and the CRC register, and store the result in the CRC register.

   Step 5: Repeat Step 3 ~ 4 and finish calculating the 8 bits.

   Step 6: Repeat Steps 2 ~ 5 for obtaining the next 8-bit message instruction until all the message instructions are calculated. In the end, the obtained CRC register value is the CRC checksum. Be aware that CRC checksum should be placed in the checksum of the message instruction.

API	Mnemonic	Operands	Function
109	SWRD P	D	Read Digital Switch

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SWRD, SWRDP: 3 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

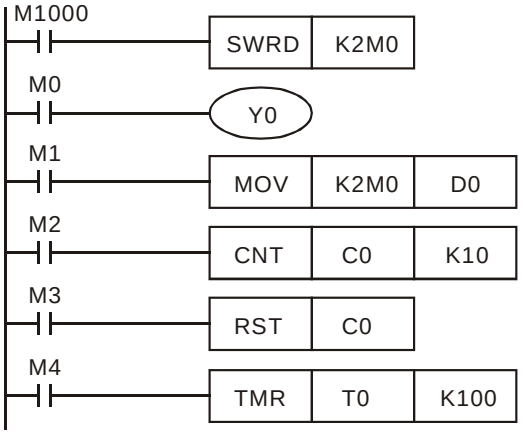
D: Device for storing the read value

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1104 ~ M1111 (status of digital switch)
3. This instruction stores the value read from digital switch function card into **D**.
4. The read value is stored in the low byte in D. Every switch has a corresponding bit.
5. When there is no digital function card inserted, the error message C400 (hex) will appear in grammar check.

Program Example:

1. There are l 8 DIP switches on the digital switch function card. After the switches are read by SWRD instruction, the status of each switch will correspond to M0 ~ M7.



2. The status of M0 ~ M7 can be executed by each contact instruction.
3. The execution of END instruction indicates that the process of input is completed. REF (I/O refresh) instruction will be invalid.
4. When SWRD instruction uses the data in digital switch function card, it can read minimum 4 bits (K1Y*, K1M* or K1S*).

Remarks:

When digital switch function card is inserted, the status of the 8 DIP switches will correspond to M1104 ~ M1111.

API	Mnemonic			Operands	Function
110	D	ECMP	P	(S1) (S2) (D)	Floating Point Compare

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DECMP, DECMPP: 13 steps
S_1					*	*							*			
S_2					*	*							*			
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

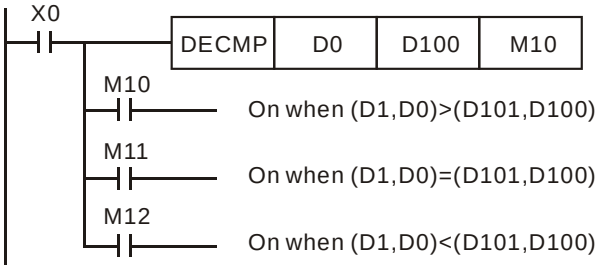
S_1: Binary floating point comparison value 1 **S_2**: Binary floating point comparison value 2 **D**: Comparison result

Explanations:

1. **D** occupies 3 consecutive devices.
2. See the specifications of each model for their range of use.
3. The binary floating point values **S_1** and **S_2** are compared with each other. The comparison result (>, =, <) is stored in **D**.
4. If **S_1** or **S_2** is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the comparison.

Program Example:

1. Designated device M10 and M10 ~ M12 are automatically occupied.
2. When X0 = On. DECMP instruction will be executed and one of M10 ~ M12 will be On. When X0 = Off, DECMP instruction will not be executed and M10 ~ M12 will remain their status before X0 = Off.
3. To obtain results ≧, ≦ ≠serial-parallel M10 ~ M12.
4. Use RST or ZRST instruction to clear the result.



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
111	D	EZCP	P	(S1) (S2) (S) (D)	Floating Point Zone Compare

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEZCP, DEZCPP: 17 steps
S_2					*	*							*			
S					*	*							*			
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

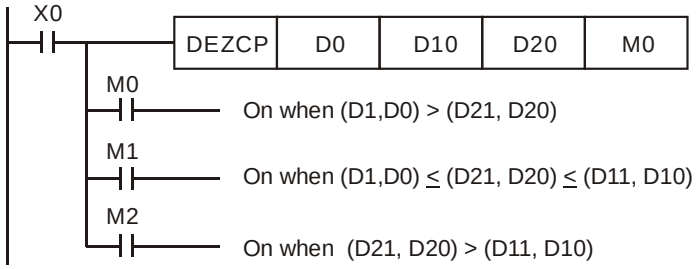
S_1: Lower bound of binary floating point **S_2**: Upper bound of binary floating point **S**: Binary floating point comparison result **D**: Comparison result

Explanations:

1. **D** occupied 3 consecutive devices.
2. **$S_1 \le S_2$**. See the specifications of each model for their range of use.
3. **S** is compared with **S_1** and **S_2** and the result (>, =, <) is stored in **D**.
4. If **S_1** or **S_2** is andesignated constant K or H, the instruction will convert the constant into a binary floating point value before the comparison.
5. When **S_1** > **S_2**, **S_1** will be used as upper/lower bound for the comparison.

Program Example:

1. Designated device M0 and M0 ~ M2 are automatically occupied.
2. When X0 = On. DEZCP instruction will be executed and one of M0 ~ M2 will be On. When X0 = Off, EZCP instruction will not be executed and M0 ~ M2 will remain their status before X0 = Off.
3. Use RST or ZRST instruction to clear the result.



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
112	D	MOVR	P	(S) (D)	Move Floating Point Data

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S																DMOVR, DMOVRP: 9 steps
D								*	*	*	*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

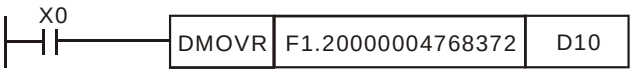
S：Source floating point data **D**：Destination device

Explanations:

1. **S** can only be a floating-point constant value.
2. See the specifications of each model for their range of use.
3. This instruction is able to enter floating point values directly in **S**.
4. When the instruction is executed, the content in **S** is moved directly into **D**. When the instruction is not executed, the content in **D** will not be modified.
5. If users want to move the floating-point value in registers, they have to use DMOV.

Program Example:

1. User DMOVR instruction to move 32-bit floating point data.
2. When X0 = Off, the content in (D11、D10) remains unchanged. When X0 = On, the present value F1.20000004768372 will be moved to data registers (D11, D10).



Remarks:

This instruction only supports ES/EX/EC V6.0 and above versions.

API	Mnemonic	Operands	Function
113	ETHRW	(S1) (S2) (D) (n)	Ethernet communication

OP \ Type	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1													*			ETHRW: 9 steps
S2					*	*							*			
D													*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S1: IP address, communication port number, and read/write mode **S2**: Device address **D**: Source/Destination data register **n**: Data length (Unit: Word; Range: K1~K96)

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: D1395 and D1396
3. **S1**: IP address, communication port number, and read/write mode

   The operand **S1** occupies five consecutive data registers. The functions are as follows.

   - IP address: Two data registers are occupied, that is, **S1**+0 and **S1**+1.

     IP address→IP3.IP2.IP1.IP0→192.168.0.2

     If **S1** is D100, the values in D100 and D101 are H'0002 and H'C0A8 respectively.

D100 (**S1**+0)		D101 (**S1**+1)	
High	Low	High	Low
IP1	IP0	IP3	IP2
0	2	192	168
H'0002		H'C0A8	

   - **S1**+2: Communication port number

     The communication port number of the Ethernet communication card installed in DVP-EH3 is K108. The communication ports on the left-side Ethernet modules connected to a CPU module are numbered according to their distances from the CPU module. The numbers start from K100 to K107.
   - **S1**+3: Station address of a slave
   - **S1**+4: Read/Write mode

     The definition is the same as Modbus. The function codes supported are H'03, H'04, H'06, and H'10.
4. **S2**: Device address

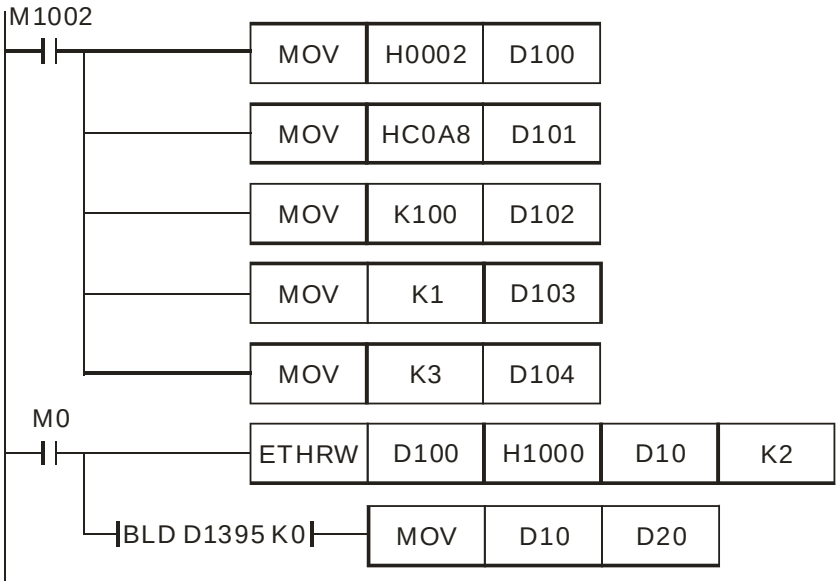
   The definition is the same as Modbus.
5. The operand **D** specifies a source data register or a destination data register. If the operand **D** specifies D10, and two pieces of communication data is read by means of the function code H'03, the communication data will be stored in D10 and D11.

6. **n**: Length of data (Unit: word)
   The setting range is K1~K96. If **n** exceeds the range, it will be taken as the maximum value or the minimum value.
7. Whenever the instruction is executed, the communication command is sent. Users do not need to enable a special flag to send the communication command.
8. The instruction can be used several times. However, if an ETHRW instruction specifies a module, other ETHRW instructions can not send communication commands to the module. The next communication command can not be sent until the reception is complete or the module replies that an error occurs.
9. If a communication command is being received, the reception stops when the execution of the instruction stops. Besides, the flag related to the command's having being received and the error flag are not ON.
10. The communication timeout is stored in D1394. The default timeout is 3000 milliseconds. The range of digital values is 1~32767. If the communication timeout exceeds the range, it will be taken as 3000 milliseconds.
11. The values of bit0~bit8 in D1395 indicate which communication port has received a command. Bit8 represents an Ethernet communication card. For example, if the Ethernet communication card installed in DVP-EH3 has received a command, "BLD D1395 K8" is satisfied.
12. The values of bit0~bit8 in D1396 indicate which module does not receive a command correctly. For example, if a reception error occurs in the first left-side module DVP-EN01, "BLD D1396 K0" is satisfied.
13. When the instruction is executed, user can not use the online editing function. Otherwise, the data received will not be stored correctly.
14. The instruction supports EH3 version 1.20 (and above), and SV2 version 1.00 (and above).

Program Example:

(The command is sent and received through the first left-side module (DVP-EN01) connected to DVP-EH3-L.)

The IP address stored in D100 and D101 is 192.168.0.2, the communication port number stored in D102 is K100, the station address stored in D103 is K1, and the function code stored in D104 is H'03. The device address is H'1000, and two pieces of data are read. When M0 is ON, ETHRW is executed. After the reception of the communication command is complete, bit0 in D1395 is ON. The data received is stored in D10 and D11.



API	Mnemonic			Operands	Function
114		MUL16 MUL32	P	(S1) (S2) (D)	16-bit Multiplication 32-bit Multiplication

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*		MUL, DMULP: 7 steps DMUL, DMULP: 13 steps
S2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

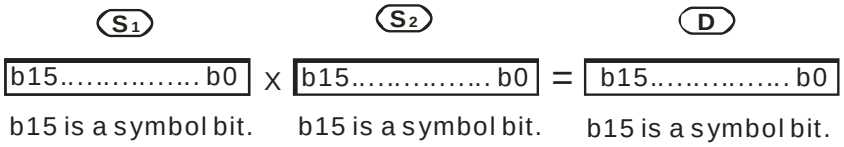
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S1: Multiplicand **S2**: Multiplicator **D**: Product

Explanations:

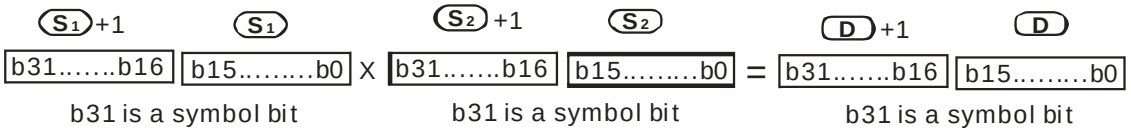
1. In 16-bit instruction, **D** occupies one device.
2. In 32-bit instruction, **D** occupies 2 consecutive devices.
3. See the specifications of each model for their range of use.
4. Flag: M1022 (carry flag)
5. This instruction multiplies **S1** by **S2** in BIN format and stores the result in **D**. Be careful with the positive/negative signs of **S1**, **S2** and **D** when doing 16-bit and 32-bit operations.
6. The instruction supports EH3/EH3L/SV2 series PLCs whose version is 1.82 or above.
7. In 16-bit BIN multiplication,



16 bits x 16 bits = 16 bits
Symbol bit = 0 refers to a positive value.
Symbol bit = 1 refers to a negative value.

When D serves as a bit device, it can designate K1 ~ K4 and construct a 16-bit result, occupying 16-bit data.

8. In 32-bit BIN multiplication,

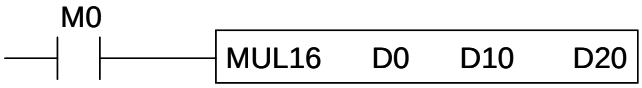


32 bits x 32 bits = 32 bits
Symbol bit = 0 refers to a positive value.
Symbol bit = 1 refers to a negative value.

When D serves as a bit device, it can designate K1 ~ K8 and construct a 32-bit result, occupying consecutive 32-bit data.

Program Example 1:

If M0 is On, the 16-bit D0 is multiplied by the 16-bit D10 and a 16-bit product is produced. The 16-bit data is stored in D20. On/Off of the most left bit indicates the positive/negative status of the result value.



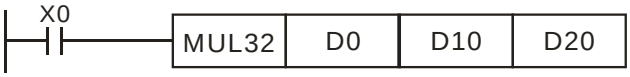
16 bits × 16 bits = 16 bits

⇨ D0 × D10 = D20

⇨ D0=K100, D10=K200, D20=K10,000

Program Example 2:

If X0 is On, the 32-bit value K10,00 in (D1, D0) is multiplied by the 32-bit value K20,000 in (D11, D10) and a 32-bit product is produced. The 32-bit data is stored in (D21, D20). On/Off of the most left bit indicates the positive/negative status of the result value.



32 bits × 32 bits = 32 bits

⇨ (D1,D0) ×(D11,D10) = (D21,D20)

⇨ (D1,D0)=K10,000, (D11,D10)=K20,000, (D21, D20)=K200,000,000

Remarks:

1. If the value gotten from the 16-bit multiplication can not be represented by a 16-bit signed value, and is greater than the maximum 16-bit positive value K32767 or less than the minimum 16-bit negative value K-32768, M1022 is On, and the low 16-bit data is stored.
2. If users need to get a complete value (32-bit value) from a 16-bit multiplication, they have to use API22 MUL/MULP. Please refer to the explanation of API22 MUL/MULP for more information.
3. If the value gotten from the 32-bit multiplication can not be represented by a 32-bit signed value, and is greater than the maximum 32-bit positive value K2147483647 or less than the minimum 16-bit negative value K-2147483648, M1022 is On, and the low 32-bit data is stored.
4. If users need to get a complete value (64-bit value) from a 32bit multiplication, they have to use API22 DMUL/DMULP. Please refer to the explanation of API22 DMUL/DMULP for more information.

API	Mnemonic			Operands	Function
115		DIV16 DIV32	P	(S1) (S2) (D)	16-bit Division 32-bit Division

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DIV, DIVP: 7 steps DDIV, DDIVP: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

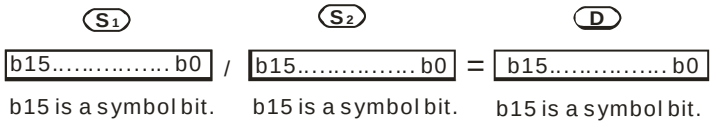
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Dividend **S_2**: Divisor **D**: Quotient and remainder

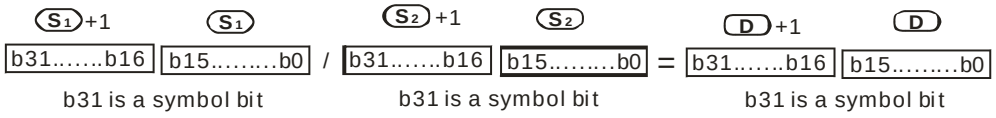
Explanations:

1. In 16-bit instruction, **D** occupies one device.
2. In 32-bit instruction, **D** occupies 2 consecutive devices.
3. See the specifications of each model for their range of use.
4. This instruction divides **S_1** and **S_2** in BIN format and stores the result in **D**. Be careful with the positive/negative signs of **S_1**, **S_2** and **D** when doing 16-bit and 32-bit operations.
5. The instruction supports EH3/EH3L/SV2 series PLCs whose version is 1.82 or above.
6. This instruction will not be executed when the divisor is 0. M1067 and M1068 will be On and D1067 records the error code 0E19 (hex).
7. In 16-bit BIN division,



When D serves as a bit device, it can designate K1 ~ K4 and construct a 16-bit result, occupying a 16-bit quotient.

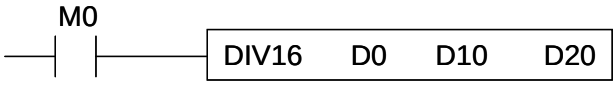
8. In 32-bit BIN division,



When D serves as a bit device, it can designate K1 ~ K8 and construct a 32-bit result, occupying a 32-bit quotient.

Program Example 1:

If M0 = On, the value in D0 (K103) will be divided by the value in D10 (K5) and the quotient will be stored in D20. On/Off of the highest bit indicates the positive/negative status of the result value.



D0/D10=D20

⇨ K103/K5=K20. The remainder is K3.

⇨ D20=K20 (The remainder is left out.)

Program Example 2:

If M0 = On, the value in (D1, D0) (K81,000) will be divided by the value in (D11, D10) (K40,000) and the quotient will be stored in (D21, D20). On/Off of the highest bit indicates the positive/negative status of the result value.

(D1,D0)/(D11,D10)=(D21,D20)

⇨ K81,000/K40,000=K2. The remainder is K1,000.

⇨ (D21,D20)=K2 (The remainder is left out.)

Remarks:

1. If users need to record a remainder by a 16-bit division, they have to use API23 DIV/DIVP. Please refer to the explanation of API23 DIV/DIVP for more information.
2. If users need to record a remainder by a 32-bit division, they have to use API23 DDIV/DDIVP. Please refer to the explanation of API23 DDIV/DDIVP for more information.

API	Mnemonic			Operands	Function
116	D	RAD	P	(S) (D)	Angle → Radian

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DRAD, DRADP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

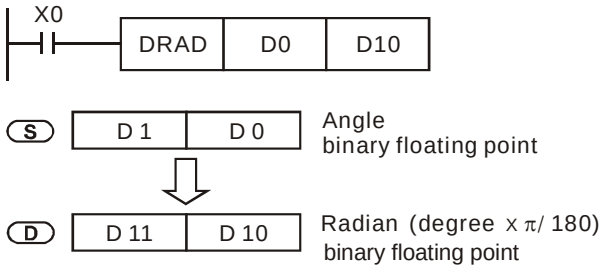
S: Source (angle) **D**: Result (radian)

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. Radian = degree × (π /180)
4. If the absolute value of the result $>$ maximum floating point available, the carry flag M1022 = On.
5. If the absolute value of the result $<$ minimum floating point available, the borrow flag M1021 = On.
6. If the result = 0, the zero flag M1020 = On.

Program Example:

When X0 = On, designate the degree of binary floating point (D1, D0). Convert the angle into radian and store the result in binary floating point in (D11, D10).



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
117	D	DEG	P	(S) (D)	Radian → Angle

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DDEG, DDEGP: 9 steps
S					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

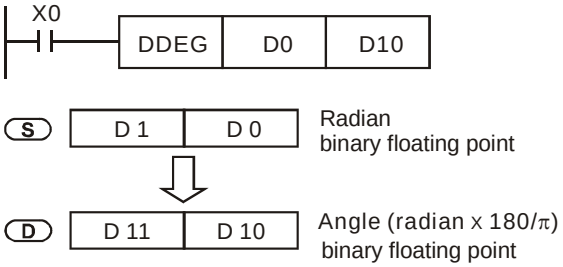
S: Source (radian) **D**: Result (angle)

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. Degree = radian × (180/π)
4. If the absolute value of the result $>$ maximum floating point available, the carry flag M1022 = On.
5. If the absolute value of the result $<$ minimum floating point available, the borrow flag M1021 = On.
6. If the result = 0, the zero flag M1020 = On.

Program Example:

When X0 = On, designate the angle of binary floating point (D1, D0). Convert the radian into angle and store the result in binary floating point in (D11, D10).



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
118	D	EBCD	P	(S) (D)	Float to Scientific Conversion

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DEBCD, DEBCDP: 9 steps
S													*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

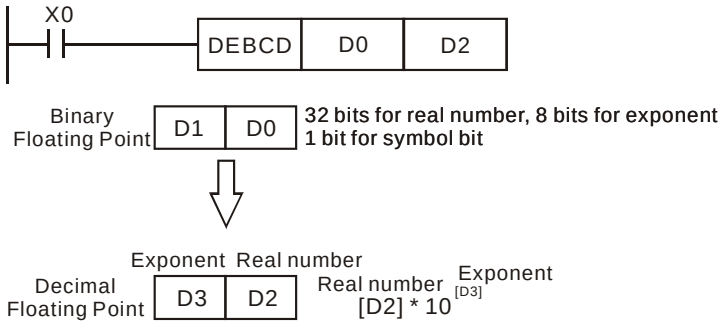
S: Source **D**: Result

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. This instruction converts binary floating point value in the register designated by **S** into decimal floating point value and stores it in the register designated by **D**.
4. PLC conducts floating point operation in binary format. DEBCD instruction is exclusively for converting floating points from binary to decimal.
5. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
6. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
7. If the result = 0, the zero flag M1020 = On.

Program Example:

When X0 = On, the binary floating points in D1 and D0 will be converted into decimal floating points and stored in D3 and D2.



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
119	D	EBIN	P	(S) (D)	Scientific to Float Conversion

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			DEBIN, DEBINP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

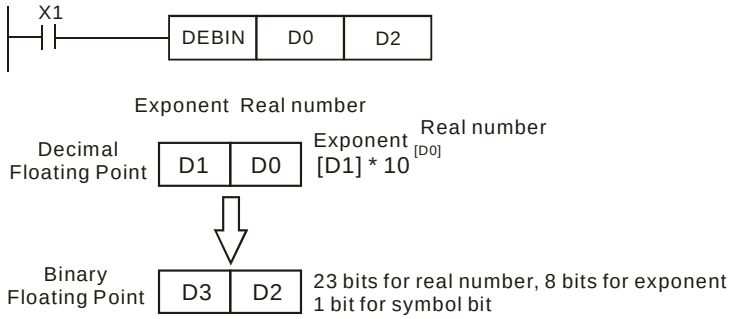
S: Source **D**: Result

Explanations:

1. See the specifications of each model for their range of use.
2. Flag: M1020 (zero flag)
3. This instruction converts decimal floating point value in the register designated by **S** into binary floating point value and stores it in the register designated by **D**.
4. DEBIN instruction is exclusively for converting floating points from decimal to binary.
5. Range of decimal floating point real numbers: -9.999 ~ +9,999. Range of exponants: -41 ~ +35. Range of PLC decimal floating points: $\pm 1{,}175 \times 10^{-41} \sim \pm 3{,}402 \times 10^{+35}$.
6. If the result = 0, the zero flag M1020 = On.

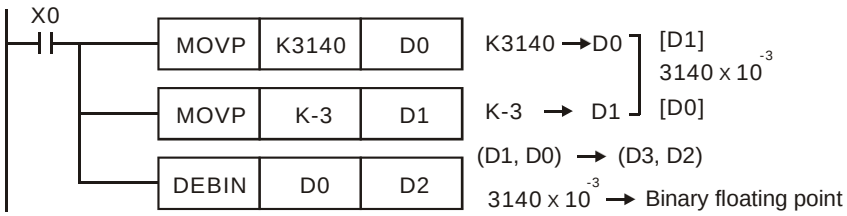
Program Example 1:

When X1 = On, the decimal floating points in D1 and D0 will be converted into binary floating points and stored in D3 and D2.



Program Example 2:

1. Use FLT instruction (API 149) to convert BIN integer into binary floating point before performing floating point operation. The value to be converted must be BIN integer and use DEBIN instruction to convert the floating point into a binary one.
2. When X0 = On, move K3,140 to D0 and K-3 to D1 to generate decimal floating point ($3.14 = 3140 \times 10^{-3}$).



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
120	D	EADD	P	(S1) (S2) (D)	Floating Point Addition

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DEADD, DEADDP: 13 steps
S_1					*	*							*			
S_2					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Summand **S_2**: Addend **D**: Sum

Explanations:

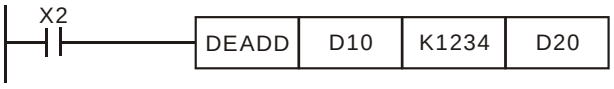
1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. **S_1** + **S_2** = **D**. The floating point value in the register designated by **S_1** and **S_2** are added up and the result is stored in the register designated by **D**. The addition is conducted in binary floating point system.
4. If **S_1** or **S_2** is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
5. **S_1** and **S_2** can designate the same register. In this case, if the "continuous execution" instruction is in use, during the period when the criteria contact in On, the register will be added once in every scan by pulse execution instruction DEADDP.
6. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
7. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
8. If the result = 0, the zero flag M1020 = On.

Program Example 1:

When X0 = On, binary floating point (D1, D0) + binary floating point (D3, D2) and the result is stored in (D11, D10).

Program Example 2:

When X2 = On, binary floating point (D11, D10) + K1234 (automatically converted into binary floating point) and the result is stored in (D21, D20).



Remarks:

For floating point operations, see "5.2 Handling of Numeric Values".

API	Mnemonic			Operands	Function
121	D	ESUB	P	(S1) (S2) (D)	Floating Point Subtraction

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DESUB, DESUBP: 13 steps
S_1					*	*							*			
S_2					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Minuend **S_2**: Subtrahend **D**: Remainder

Explanations:

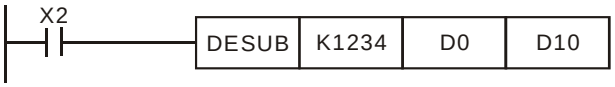
1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. **S_1** − **S_2** = **D**. The floating point value in the register designated by **S_2** is subtracted from the floating point value in the register assigned by **S_1** and the result is stored in the register designated by **D**. The subtraction is conducted in binary floating point system.
4. If **S_1** or **S_2** is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
5. **S_1** and **S_2** can designate the same register. In this case, if the “continuous execution” instruction is in use, during the period when the criteria contact in On, the register will be subtracted once in every scan by pulse execution instruction DESUBP.
6. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
7. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
8. If the result = 0, the zero flag M1020 = On.

Program Example 1:

When X0 = On, binary floating point (D1, D0) － binary floating point (D3, D2) and the result is stored in (D11, D10).

Program Example 2:

When X2 = On, K1234 (automatically converted into binary floating point) － binary floating point (D1, D0) and the result is stored in (D11, D10).



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
122	D	EMUL	P	(S1) (S2) (D)	Floating Point Multiplication

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEMUL, DEMULP: 13 steps
S_2					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Multiplicand **S_2**: Multiplicator **D**: Product

Explanations:

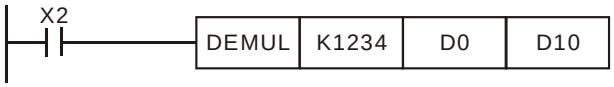
1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. **S_1** × **S_2** = **D**. The floating point value in the register assigned by **S_1** is multiplied with the floating point value in the register designated by **S_2** and the result is stored in the register designated by **D**. The multiplication is conducted in binary floating point system.
4. If **S_1** or **S_2** is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
5. **S_1** and **S_2** can designate the same register. In this case, if the "continuous execution" instruction is in use, during the period when the criteria contact in On, the register will be multiplied once in every scan by pulse execution instruction DEMULP.
6. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
7. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
8. If the result = 0, the zero flag M1020 = On.

Program Example 1:

When X1 = On, binary floating point (D1, D0) × binary floating point (D11, D10) and the result is stored in (D21, D20).

Program Example 2:

When X2 = On, K1234 (automatically converted into binary floating point) × binary floating point (D1, D0) and the result is stored in (D11, D10).



Remarks:

For floating point operations, see "5.2 Handling of Numeric Values".

API	Mnemonic			Operands	Function
123	D	EDIV	P	(S1) (S2) (D)	Floating Point Division

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEDIV, DEDIVP: 13 steps
S_2					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

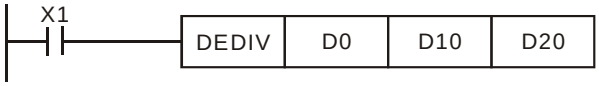
S_1: Dividend **S_2**: Divisor **D**: Quotient and remainder

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. **S_1** ÷ **S_2** = **D**. The floating point value in the register designated by **S_1** is divided by the floating point value in the register assigned by **S_2** and the result is stored in the register designated by **D**. The division is conducted in binary floating point system.
4. If **S_1** or **S_2** is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
5. If **S_2** = 0, operation error will occur, the instruction will not be executed, M1067, M1068 = On and D1067 will recorded the error code H'0E19.
6. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
7. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
8. If the result = 0, the zero flag M1020 = On.

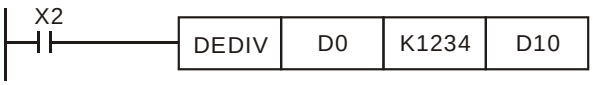
Program Example 1:

When X1 = On, binary floating point (D1, D0) ÷ binary floating point (D11, D10) and the quotient is stored in (D21, D20).



Program Example 2:

When X2 = On, binary floating point (D1, D0) ÷ K1234 (automatically converted into binary floating point) and the result is stored in (D11, D10).



Remarks:

For floating point operations, see "5.2 Handling of Numeric Values".

API	Mnemonic			Operands	Function
124	D	EXP	P	(S) (D)	Exponent of Binary Floating Point

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DEXP, DEXPP: 13 steps
S					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

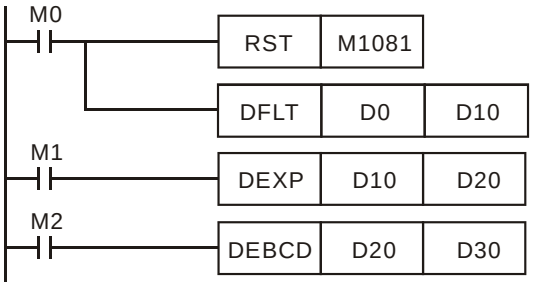
S: Device for operation source **D**: Device for operation result

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. e = 2.71828 as the base and **S** as exponent for EXP operation: EXP〔**D** +1, **D**〕= 〔**S** +1, **S**〕
4. Both positive and negative values are valid for **S**. When designating **D** registers, the data should be 32-bit and the operation should be performed in floating point system. Therefore, **S** should be converted into a floating point value.
5. The content in **D** = e^{S}; e = 2.71828, ^{S} = designated source data
6. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
7. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
8. If the result = 0, the zero flag M1020 = On.

Program Example:

1. When M0 = On, convert (D1, D0) into binary floating point and store it in register (D11, D10).
2. When M1= On, use (D11, D10) as the exponent for EXP operation and store the binary floating point result in register (D21, D20).
3. When M2 = On, convert the binary floating point (D21, D20) into decimal floating point (D30 × $10^{[D31]}$) and store it in register (D31, D30).



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
125	D	LN	P	(S) (D)	Natural Logarithm of Binary Floating Point

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DLN, DLNP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

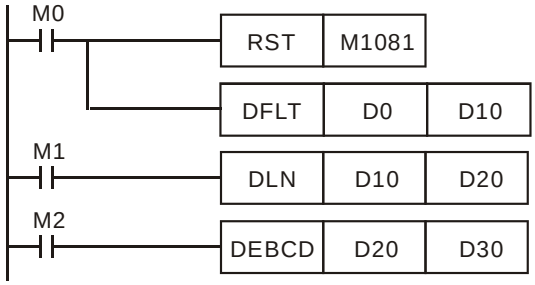
S: Device for operation source **D**: Device for operation result

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. This instruction performs natural logarithm "LN" operation by **S**: LN〔**S** + 1, **S**〕=〔**D** + 1, **D**〕
4. Only positive values are valid for **S**. When designating **D** registers, the data should be 32-bit and the operation should be performed in floating point system. Therefore, **S** should be converted into a floating point value.
5. e^{D} = **S** .The content in **D** = ln**S**; **S** = designated source data.
6. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
7. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
8. If the result = 0, the zero flag M1020 = On.

Program Example:

1. When M0 = On, convert (D1, D0) into binary floating point and store it in register (D11, D10).
2. When M1= On, use register (D11, D10) as the real number for LN operation and store the binary floating point result in register (D21, D20).
3. When M2 = On, convert the binary floating point (D21, D20) into decimal floating point (D30 × $10^{[D31]}$) and store it in register (D31, D30).



Remarks:

For floating point operations, see "5.2 Handling of Numeric Values".

API	Mnemonic			Operands	Function
126	D	LOG	P	(S1) (S2) (D)	Logarithm of Binary Floating Point

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DLOG, DLOGP: 13 steps
S_2					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Device for base **S_2**: Device for operation source **D**: Device for operation result

Explanations:

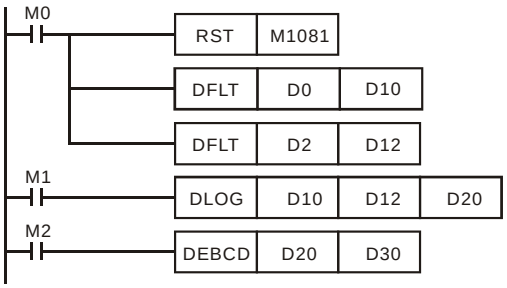
1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. This instruction performs "log" operation of the content in **S_1** and **S_2** and stores the result in **D**.
4. Only positives are valid for the content in **S_1** and **S_2**. When designating **D** registers, the data should be 32-bit and the operation should be performed in floating point system. Therefore, **S_1** and **S_2** should be converted into floating point values.
5. $\mathbf{S_1}^{D} = \mathbf{S_2}$, **D** = ? → $\mathrm{Log}_{\mathbf{S_1}}{}^{\mathbf{S_2}} = \mathbf{D}$

   Example: Assume $\mathbf{S_1} = 5$, $\mathbf{S_2} = 125$, $\mathbf{D} = \log_5 125 = ?$

   $\mathbf{S_1}^{\mathbf{D}} = \mathbf{S_2} \rightarrow 5^{\mathbf{D}} = 125 \rightarrow \mathbf{D} = \log_5 125 = 3$
6. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
7. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
8. If the result = 0, the zero flag M1020 = On.

Program Example:

1. When M0 = On, convert (D1, D0) and (D3, D2) into binary floating points and store them in the 32-bit registers (D11, D10) and (D13, D12).
2. When M1= On, perform log operation on the binary floting points in 32-bit registers (D11, D10) and (D13, D12) and store the result in the 32-bit register (D21, D20).
3. When M2 = On, convert the binary floating point (D21, D20) into decimal floating point ($D30 \times 10^{[D31]}$) and store it in register (D31, D30).



Remarks: For floating point operations, see "5.2 Handling of Numeric Values".

API	Mnemonic			Operands	Function
127	D	ESQR	P	(S) (D)	Floating Point Square Root

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DESQR, DESQRP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

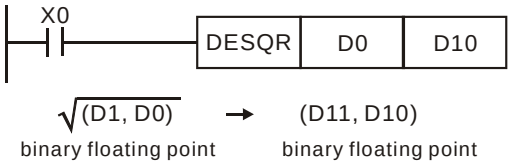
S: Source device **D**: Operation result

Explanations:

1. Range of **S**: ≥ 0
2. See the specifications of each model for their range of use.
3. Flags: M1020 (zero flag); M1067 (operation error)
4. This instruction performs a square root operation on the content in the register designated by **S** and stores the result in the register designated by **D**. The square root operation is performed in floating point system.
5. If **S** is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
6. If the result of the operation = 0, the zero flag M1020 = On.
7. **S** can only be a positive value. Performing any square root operation on a negative value will result in an "operation error" and this instruction will not be executed. M1067 and M1068 will be On and D1067 will record the error code H'0E1B.

Program Example 1:

When M0 = On, calculate the square root of the binary floating point (D1, D0) and store the result in register (D11, D10).



Program Example 2:

When M2 = On, calculate the square root of K1,234 (automatically converted into binary floating point) and store the result in register (D11, D10).

Remarks:

For floating point operations, see "5.2 Handling of Numeric Values".

API	Mnemonic			Operands	Function
128	D	POW	P	(S1) (S2) (D)	Floating Point Power Operation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DPOW, DPOWP: 13 steps
S_2					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

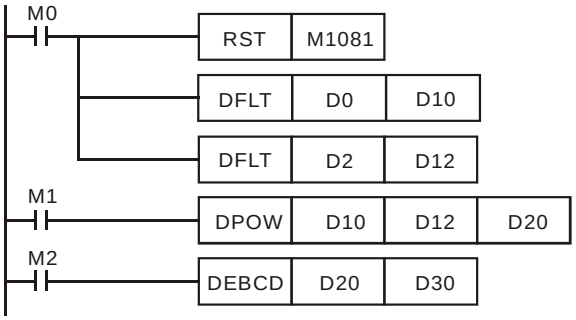
S_1: Device for base. **S_2**: Device for exponent. **D**: Device for operation result

Explanations:

1. See the specifications of each model for their range of use.
2. This instruction performs power multiplication of binary floating point **S_1** and **S_2** and stores the result in **D**.
   D = POW〔**S_1** + 1, **S_1**〕^〔**S_2** + 1, **S_2**〕
3. Only positives are valid for the content in **S_1**. Both positives and negatives are valid for the content in **S_2**. When designating **D** registers, the data should be 32-bit and the operation should be performed in floating point system. Therefore, **S_1** and **S_2** should be converted into floating point values.
   Example: When **S_1**^{S2} = **D**, **D** = ?
   Assume **S_1** = 5, **S_2** = 3, **D** = 5^3 =125
4. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
5. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
6. If the result = 0, the zero flag M1020 = On.

Program Example:

1. When M0 = On, convert (D1, D0) and (D3, D2) into binary floating points and store them in the 32-bit registers (D11, D10) and (D13, D12).
2. When M1= On, perform POW operation on the binary floting points in 32-bit registers (D11, D10) and (D13, D12) and store the result in the 32-bit register (D21, D20).
3. When M2 = On, convert the binary floating point (D21, D20) into decimal floating point (D30 × 10$^{[D31]}$) and store it in register (D31, D30).



Remarks: For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
129	D	INT	P	(S) (D)	Float to Integer

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			INT, INTP: 5 steps DINT, DINTP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

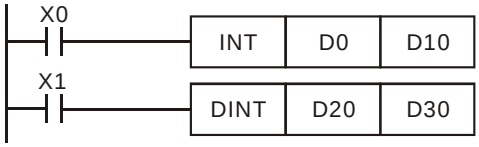
S: Source device **D**: Converted result

Explanations:

1. **S** occupies 2 consecutive devices. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. The binary floating point value of the register designated by **S** is converted to BIN integer and stored in the register designated by **D**. The decimal of BIN integer is left out.
4. This instruction is the inverse operation of API 49 FLT instruction.
5. If the converstion result = 0, the zero flag M1020 = On
   If there is any decimal left out, the borrow flag M1021 = On.
   If the result exceeds the range listed below, the carry flag M1022 = On.
   16-bit instruction: -32,768 ~ 32,767
   32-bit instruction: -2,147,483,648 ~ 2,147,483,647

Program Example:

1. When X0 = On, the binary floating point (D1, D0) will be converted into BIN integer and the result will be stored in (D10). The decimal of BIN integer will be left out.
2. When X1 = On, the binary floating point (D21, D20) will be converted into BIN integer and the result will be stored in (D31, D30). The decimal of BIN integer will be left out.



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
130	D	SIN	P	(S) (D)	Sine

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DSIN, DSINP: 9 steps
S					*	*							*			
D													*			

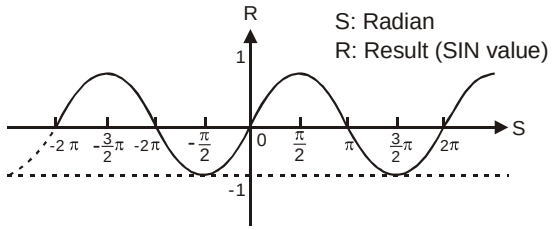
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source value **D**: SIN result

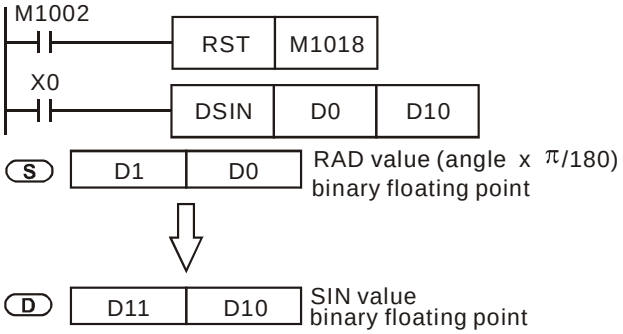
Explanations:

1. 0° ≤ **S** < 360°. See the specifications of each model for their range of use.
2. Flags: M1018 (angle or radian); M1020 (zero flag)
3. **S** can be an angle or radian, decided by M1018.
4. When M1018 = Off, the program will be in radian mode and the RAD value = angle × π /180
5. When M1018 = On, the program will be in angle mode and the range of angle should be "0° ≤ angle < 360°"
6. If the result = On, M1020 = On.
7. The SIN value obtained by **S** is calculated and stored in the register designated by **D**. The figure below offers the relation between radian and the result.



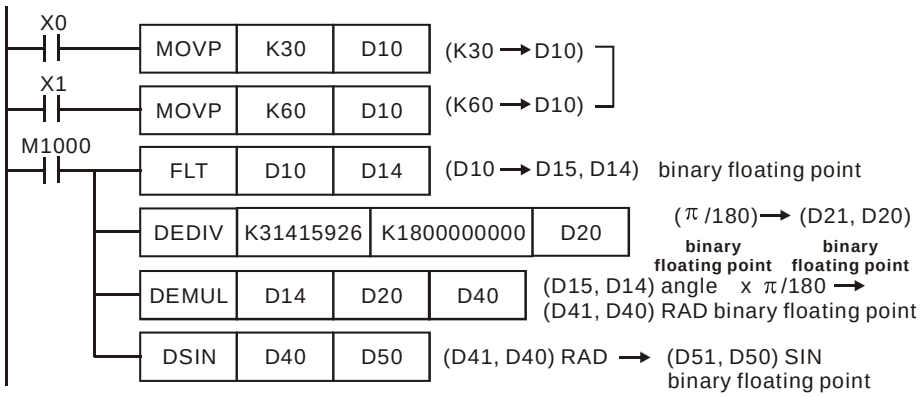
Program Example 1:

When M1018 = Off, the program is in radian mode. When X0 = On, use the RAD value of binary floating point (D1, D0) and obtain its SIN value. The binary floating point result will be stored in (D11, D10).



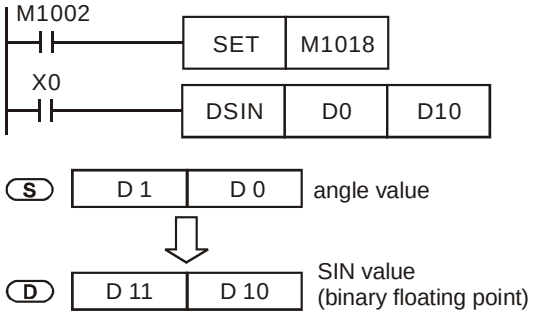
Program Example 2:

When M1018 = Off, the program is in radian mode. Input terminals X0 and X1 select the angle. The angles are converted into RAD value for calculating the SIN value.



Program Example 3:

When M1018 = On, the program is in angle mode. When X0 = On, use the angle of (D1, D0) to obtain SIN value and store the binary floating point result in (D11, D10). (0° ≤ angle < 360°)



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
131	D	COS	P	(S) (D)	Cosine

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DCOS, DCOSP: 9 steps
S					*	*							*			
D													*			

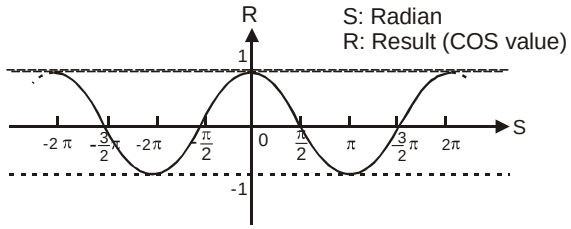
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source value **D**: COS result

Explanations:

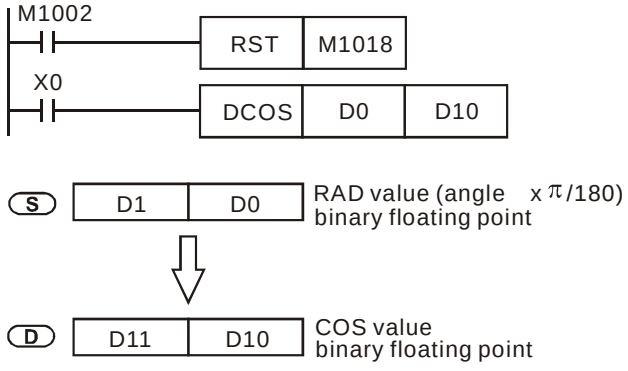
1. 0° ≤ **S** < 360°. See the specifications of each model for their range of use.
2. Flags: M1018 (angle or radian); M1020 (zero flag)
3. **S** can be an angle or radian, decided by M1018.
4. When M1018 = Off, the program will be in radian mode and the RAD value = angle × π /180
5. When M1018 = On, the program will be in angle mode and the range of angle should be "0° ≤ angle < 360°"
6. If the result = On, M1020 = On.
7. The COS value obtained by **S** is calculated and stored in the register designated by **D**. The figure below offers the relation between radian and the result.



8. Switch between radian and angle by M1018: When M1018 = Off, **S** will be a RAD value; when M1018 = On, **S** will be an angle (0° ~ 360°).

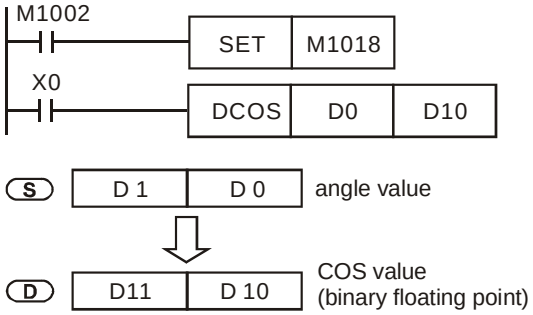
Program Example 1:

When M1018 = Off, the program is in radian mode. When X0 = On, use the RAD value of binary floating point (D1, D0) and obtain its COS value. The binary floating point result will be stored in (D11, D10).



Program Example 2:

When M1018 = On, the program is in angle mode. When X0 = On, use the angle of (D1, D0) to obtain COS value and store the binary floating point result in (D11, D10). (0° ≤ angle < 360°)



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
132	D	TAN	P	(S) (D)	Tangent

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DTAN, DTANP: 9 steps
D													*			

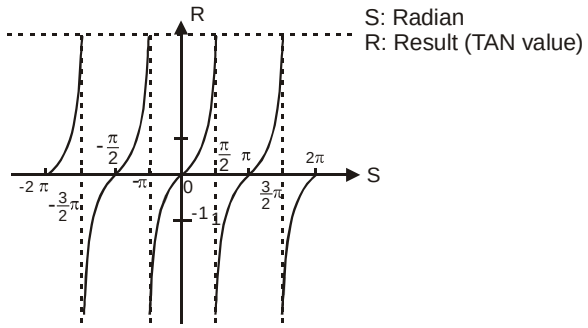
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source value **D**: TAN result

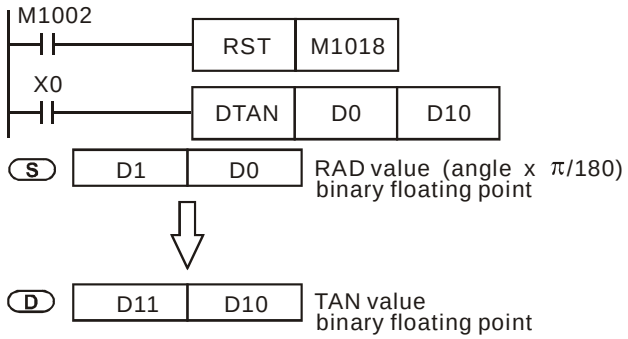
Explanations:

1. 0° ≤ **S** < 360°. See the specifications of each model for their range of use.
2. Flags: M1018 (angle or radian); M1020 (zero flag)
3. **S** can be an angle or radian, decided by M1018.
4. When M1018 = Off, the program will be in radian mode and the RAD value = angle × π /180
5. When M1018 = On, the program will be in angle mode and the range of angle should be "0° ≤ angle < 360°"
6. If the result = On, M1020 = On.
7. The TAN value obtained by S is calculated and stored in the register designated by D. The figure below offers the relation between radian and the result.



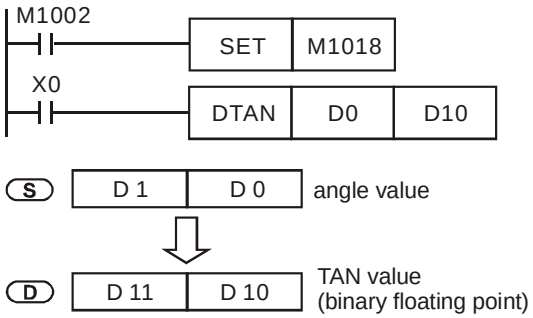
Program Example 1:

When M1018 = Off, the program is in radian mode. When X0 = On, use the RAD value of binary floating point (D1, D0) and obtain its TAN value. The binary floating point result will be stored in (D11, D10).



Program Example 2:

When M1018 = On, the program is in angle mode. When X0 = On, use the angle of (D1, D0) to obtain TAN value and store the binary floating point result in (D11, D10). (0° ≤ angle < 360°)



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
133	D	ASIN	P	(S) (D)	Arc Sine

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DASIN, DASINP: 9 steps
S					*	*							*			
D													*			

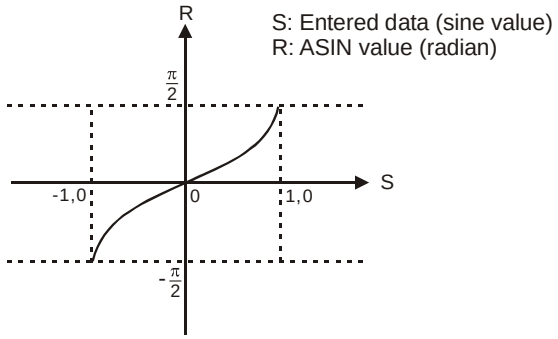
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source value (binary floating point) **D**: ASIN result

Explanations:

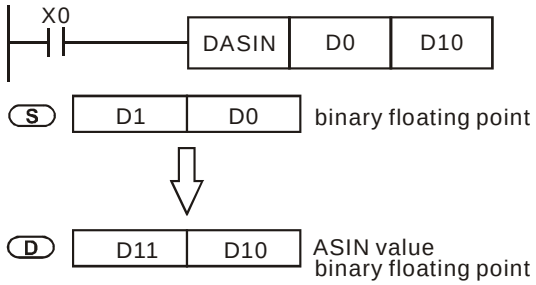
1. See the specifications of each model for their range of use.
2. Flag: M1020 (zero flag)
3. ASIN value=$\sin^{-1}$. The figure below offers the relation between the entered sin value and the result.



4. The decimal floating point of the SIN value designated by **S** should be within -1.0 ~ +1.0. If the value falls without the range, M1067 and M1068 will be On without performing any action.
5. If the result = 0, M1020 = On.

Program Example:

When X0 = On, obtain the ASIN value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
134	D	ACOS	P	(S) (D)	Arc Cosine

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DACOS, DACOSP: 9 steps
D													*			

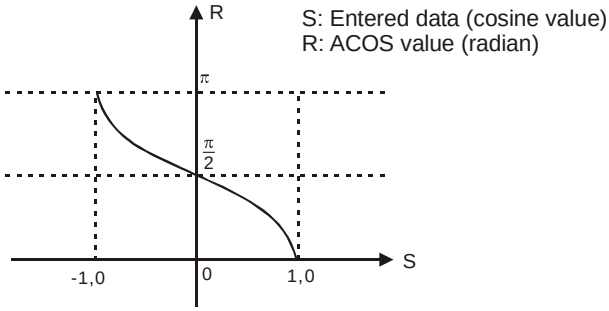
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source value (binary floating point) **D**: ACOS result

Explanations:

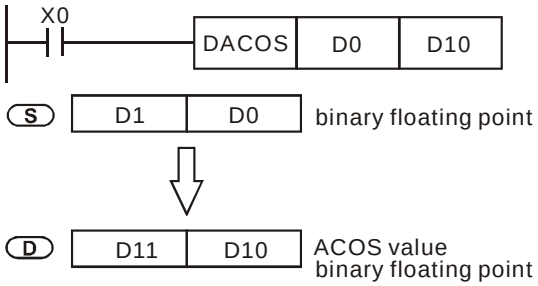
1. See the specifications of each model for their range of use.
2. Flag: M1020 (zero flag)
3. ACOS value=$\cos^{-1}$. The figure below offers the relation between the entered cos value and the result.



4. The decimal floating point of the COS value designated by **S** should be within -1.0 ~ +1.0. If the value falls without the range, M1067 and M1068 will be On without performing any action.
5. If the result = 0, M1020 = On.

Program Example:

When X0 = On, obtain the ACOS value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
135	D	ATAN	P	(S) (D)	Arc Tangent

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DATAN, DATANP: 9 steps
D													*			

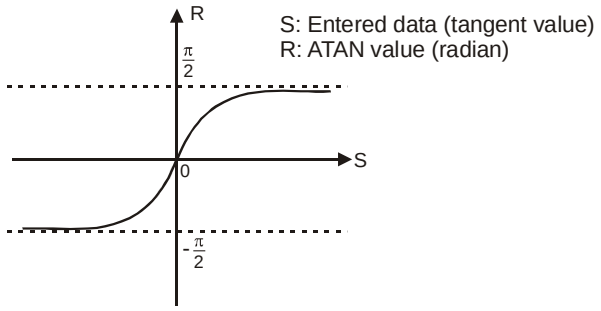
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source value (binary floating point) **D**: ATAN value

Explanations:

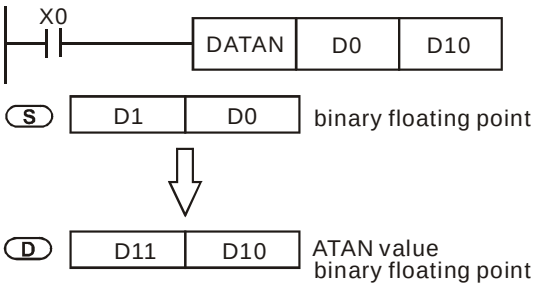
1. See the specifications of each model for their range of use.
2. Flag: M1020 (zero flag)
3. ATAN value=$\tan^{-1}$. The figure below offers the relation between the entered tan value and the result.



4. If the result =0, M1020 = On.

Program Example:

When X0 = On, obtain the ATAN value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
136	D	SINH	P	(S) (D)	Hyperbolic Sine

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DSINH, DSINHP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

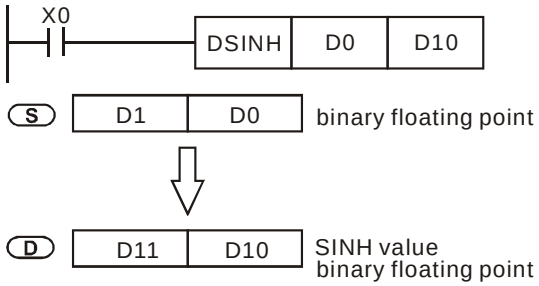
S: Source value (binary floating point) **D**: SINH value

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. SINH value=$(e^{s}-e^{-s})/2$. The result is stored in **D**.

Program Example:

1. When X0 = On, obtain the SINH value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



2. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
3. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
4. If the result = 0, the zero flag M1020 = On.

Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
137	D	COSH	P	(S) (D)	Hyperbolic Cosine

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DCOSH, DCOSHP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

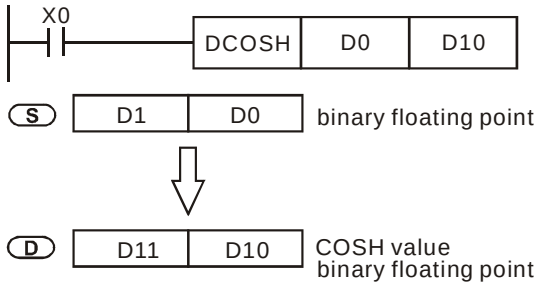
S: Source value (binary floating point) **D**: COSH value

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. COSH value=$(e^{s}+e^{-s})/2$. The result is stored in **D**.

Program Example:

1. When X0 = On, obtain the COSH value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



5. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
6. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
7. If the result = 0, the zero flag M1020 = On.

Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic			Operands	Function
138	D	TANH	P	(S) (D)	Hyperbolic Tangent

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DTANH, DTANHP: 9 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

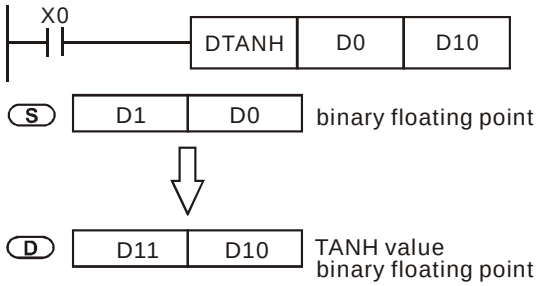
S: Source value (binary floating point) **D**: TANH result

Explanations:

1. See the specifications of each model for their range of use.
2. Flags: M1020 (zero flag); M1021 (borrow flag); M1022 (carry flag)
3. TANH value=$(e^{s}-e^{-s})/(e^{s}+e^{-s})$. The result is stored in **D**.

Program Example:

1. When X0 = On, obtain the TANH value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



2. If the absolute value of the result ＞ maximum floating point available, the carry flag M1022 = On.
3. If the absolute value of the result ＜ minimum floating point available, the borrow flag M1021 = On.
4. If the result = 0, the zero flag M1020 = On.

Remarks:

For floating point operations, see “5.2 Handling of Numeric Values”.

API	Mnemonic	Operands	Function
143	DELAY	(S)	Delay Instruction

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DELAY, DELAYP: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

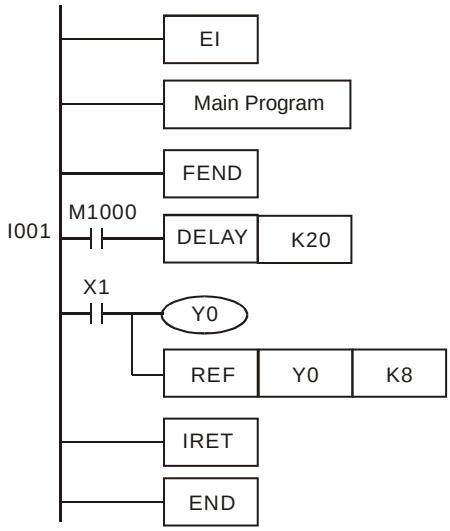
S: delay time (unit: 100ms)

Explanations:

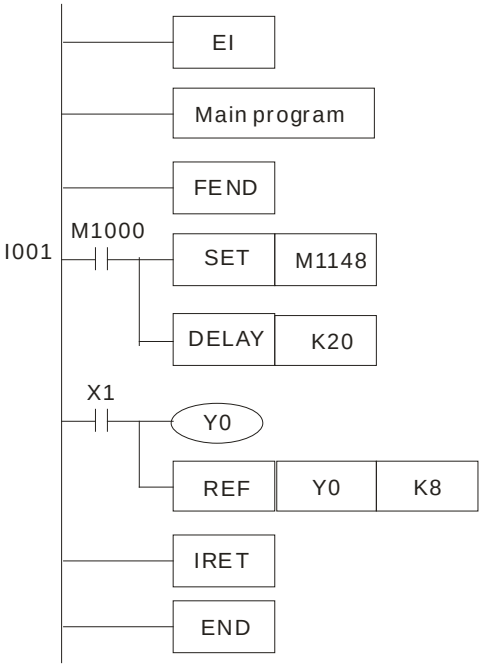
1. Range of **S**: K1 ~ K1,000. See the specifications of each model for their range of use.
2. After DELAY instruction is executed, the program after DELAY in every scan period will execute delay outputs according to the delay time designated by the user.
3. If M1148 is ON, the delay unit will be 5us. If M1148 is ON when the instruction DELAY is executed, the delay unit will change from 100us to 5us. After the instruction DELAY is executed, M1148 will be set to Off. EH3 V1.62 (and above) and SV2 V1.00 (and above) are supported.

Program Example:

If X0 is turned from Off to On, the external interruption will be generated. DELAY in the interrupt subroutine will be execute for 2 ms before the next step (X1 = On and Y0 = On) is executed.



If X0 is turned from Off to On, the external interruption will be generated. Owing to the fact that M1148 is ON, DELAY in the interrupt subroutine will be execute for 100 ms before the next step (X1 = On and Y0 = On) is executed.



Remarks:

1. User can define the delay time based on their needs.
2. The delay time may increase due do the influences from communication, high-speed counters and high-speed pulse output instructions.
3. The delay time of designated external output (transistor or relay) will increase due to the delay on the transistor or relay itself. See 2.3 for more information.

API	Mnemonic	Operands	Function
144	GPWM	(S1) (S2) (D)	General PWM Output

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			GPWM: 7 steps
S_2													*			
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

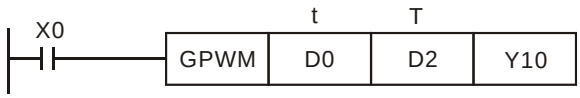
S_1: Width of output pulse **S_2**: Pulse output cycle **D**: Pulse output device

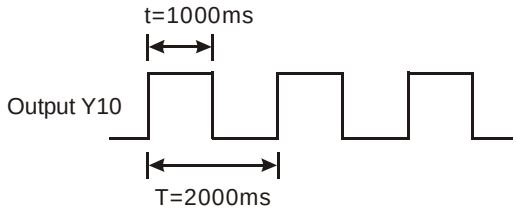
Explanations:

1. **S_2** occupies 3 consecutive devices.
2. **$S_1 \leq S_2$**. See the specifications of each model for their range of use.
3. Range of **S_1**: t = 0 ~ 32,767ms.
4. Range of **S_2**: t = 1 ~ 32,767ms.
5. **S_2** +1 and **S_2** +2 are parameters for the system. Do not occupy them.
6. Pulse output devices **D**: Y, M, S.
7. When being executed, GPWM instruction designates **S_1** and **S_2** and that pulses output will be from device **D**.
8. When **$S_1 \leq 0$**, there will be no pulse output. When **$S_1 \geq S_2$**, the pulse output device will keep being On.
9. **S_1** and **S_2** can be modified when GPWM instruction is being executed.

Program Example:

When X0 = On, D0 = K1,000, D2 = K2,000, and Y10 will output the pulse illustrated below. When X0 = Off, Y10 output will be Off.





Explanations:

1. This instruction counts by the scan cycle; therefore the maximum offset will be one PLC scan cycle. **S_1**, **S_2** and (**S_2** - **S_1**) should > PLC scan cycle; otherwise, errors will occur during GPWM outputs.
2. Please note that placing this instruction in a subroutine or interruption will cause inaccurate GPWM outputs.

API	Mnemonic	Operands	Function
145	FTC	(S1) (S2) (S3) (D)	Fuzzy Temperature Control

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FTC: 9 steps
S_1					*	*							*			
S_2					*	*							*			
S_3													*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

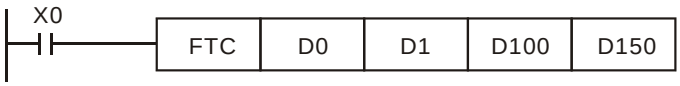
S_1: Set value (SV) **S_2**: Present value (PV) **S_3**: Parameter (sampling time) **D**: Output value (MV)

Explanations:

1. Range of **S_1** : 1 ~ 5000 (shown as 0.1°C ~ 500°C). Unit: 0.1°. If (**S_3** +1) is set as K0, the range will be 0.1°C ~ 500°C.
2. Range of **S_2** : 1 ~ 5000 (shown as 0.1°C ~ 500°C). Unit: 0.1°. If (**S_3** +1) is set as bit0 = 0, the range will be 0.1°C ~ 500°C. Therefore, when the user obtain an A/D value from the temperature sensor, the value has to be converted into a value between 1 ~ 5,000 by four arithmetic operation instructions.
3. If **S_3** < K1, the instruction will not be executed. If **S_3** > K200, S3 will adopt K200. **S_3** will occupy 7 consecutive devices.
4. See the specifications of each model for their range of use.
5. Settings of parameter **S_3** +1: bit0 = 0 ->°C; bit0 = 1 ->°F; bit1 = 0 -> no filter function; bit1 = 1 -> with filter function; bit2 ~ bit5 -> 4 kinds of heating environments; bit6 ~ bit15 -> reserved. See remarks for more information.
6. **D** is the value between 0 ~ sampling time × 100. When using this instruction, the user has to adopt other instructions according to the types of the heater. For example, FTC can be used with GPWM for output pulse control. “Sampling time × 100” is the cycle of GPWM pulse output; MV is the width of GPWM pulse. See the example 1.
7. There is no limit on the times of using FTC instruction, but Do not repeatedly use a designated operand in case an error may occur.

Program Example:

1. Set up the parameter before executing FTC instruction.
2. When X0 = On, the instruction will be executed and and result will be stored in D150. When X0 = Off, the instruction will not be executed and the previous data remain unchanged.

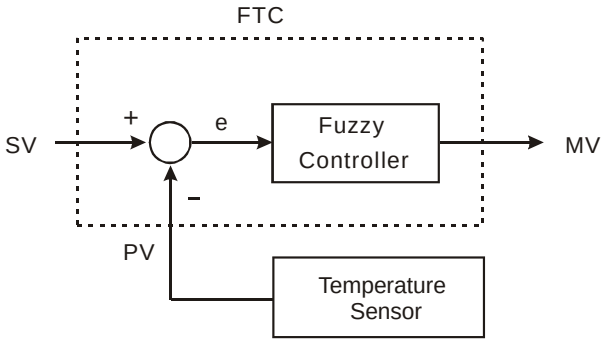


Remarks:

1. Setting of **S₃**:

Device No.	Function	Range	Explanation
S3	Sampling time (T_S) (unit: 100ms)	1 ~ 200 (unit: 100ms)	If T_S is less than a scan time, PID instruction will be executed for a scan time. If T_S= 0, PID instruction will not be enabled. The minimum T_S must be greater than a scan time.
S3 +1	b0: temperature unit b1: filter function b2 ~ b5: heating environnment b6 ~ b15: reserved	b0 =0 means °C b0 =1 means °F	When the value exceeds the upper bound, use the upper bound.
		b1=0 means without fileter function b1=1 means with filter function	When without filter function, PV = currently measured value. When with filter function, PV = (currently measured value + previous PV)/2
		b2=1	Slow heating environment
		b3=1	General heating environment
		b4=1	Fast heating environment
		b5=1	High-speed heating environment
S3 +2 ~ S3 +6	Parameters for system use only. Do not use them.		

2. Control Diagram:



3. Notes and suggestion:

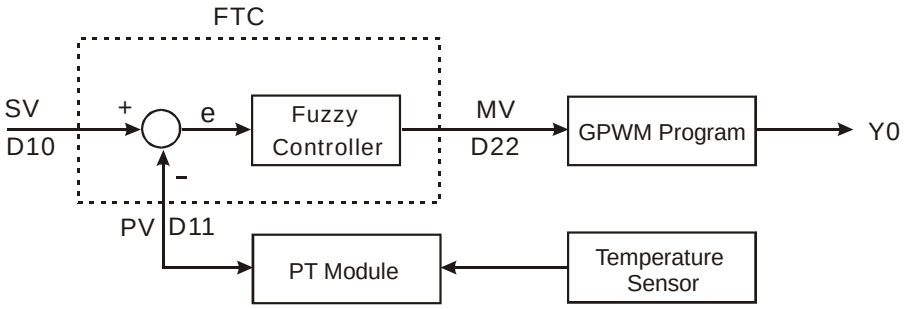
It is recommended that the sampling time be set to 2 times more than the sampling time of the temperature sensor for better temperature control.

bit2 ~ bit5 of **S₃**+1 are for the control speed. If the user does not set up the parameter, FTC will automatically activate “general heating environment”. When the user finds that the control is too slow to reach SV, select “slow heating environment” to enhance the speed to reach SV. On the contrary, when the user finds that the control is

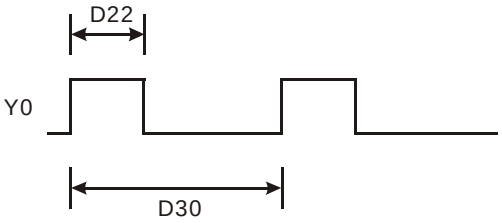
too fast or with too many fluctuations, select “fast heating environment” to slow down the control speed.

When bit2 ~ bit5 of **S_3**+1 are all set as 1 or more than 1 environments are designated, FTC instruction will check from bit2 to bit 5 in order and enable the function that has been set as 1. The parameter can be modified during the control.

4. **Example 1:** control diagram

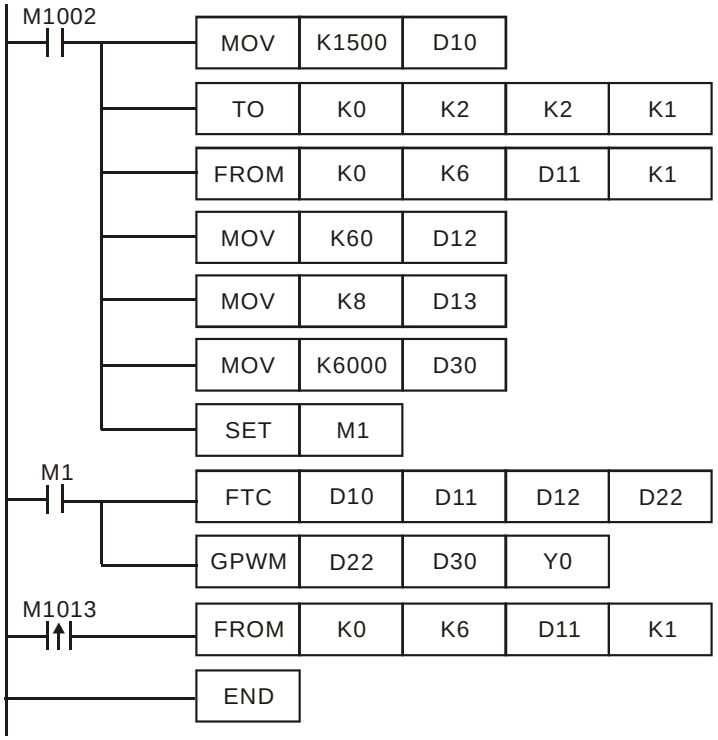


Output D22 (MV) of FTC instruction is the input D22 of GPWM instruction, as the duty cycle of ajustable pulses. D30 is the fixed cycle time of pulses. See below for the timing diagram of Y0 output.

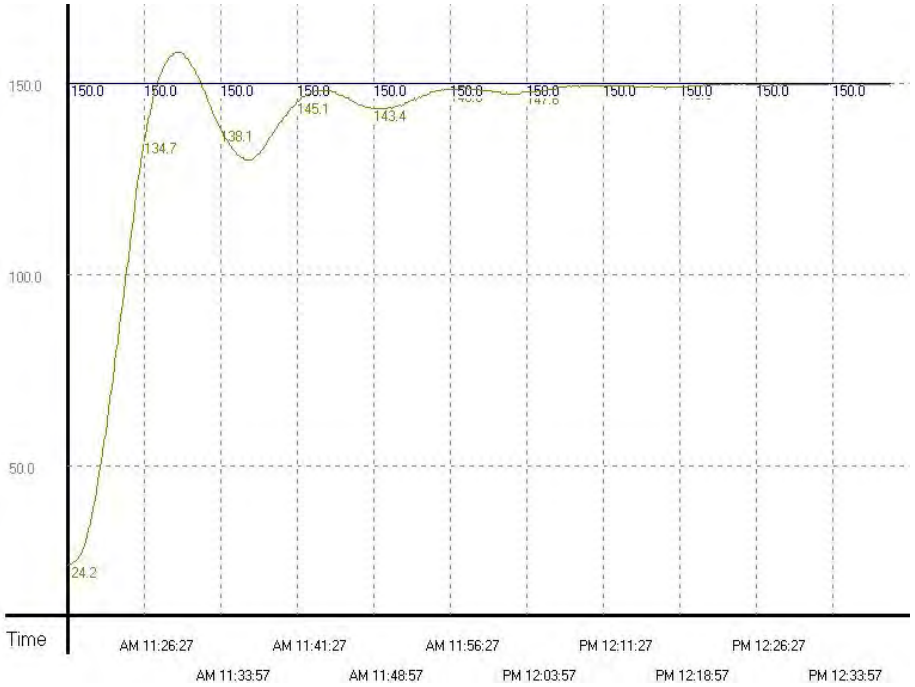


Assume parameter settings: D10 = K1,500 (target temperature), D12 = K60 (sampling time: 6 secs.), D13 = K8 (bit3=1), D30 = K6,000 (=D12*100)

The example control program is indicated as:

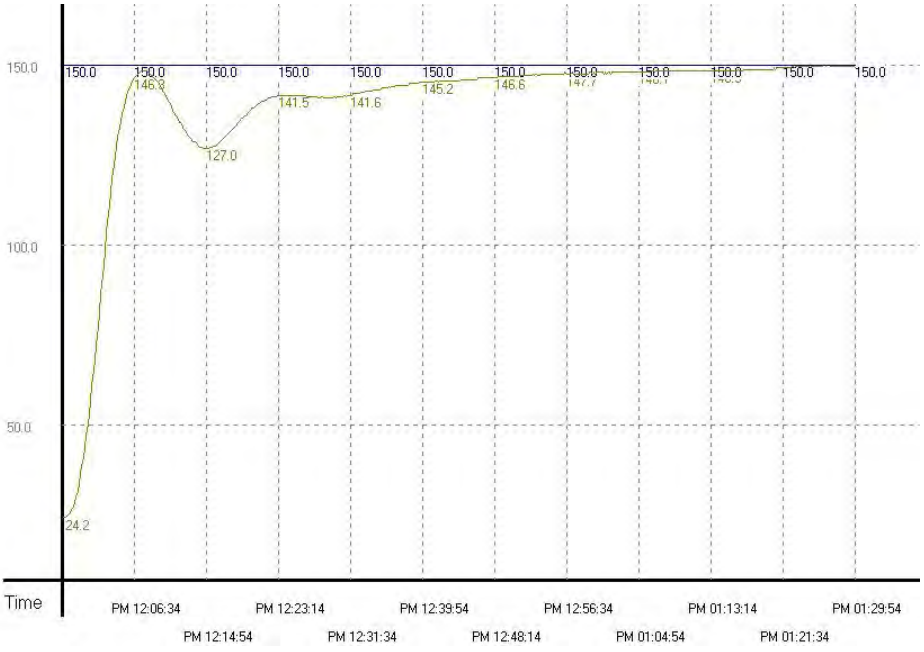


Experiment in an oven which can be heated up to 250°C. See below for the records of target and present temperatures. As shown in the diagram below, we can see that after 48 minutes, the temperature is able to reach the target temperature with ±1ºC inaccuracy and exceed approx. 10°C of the target temperature.



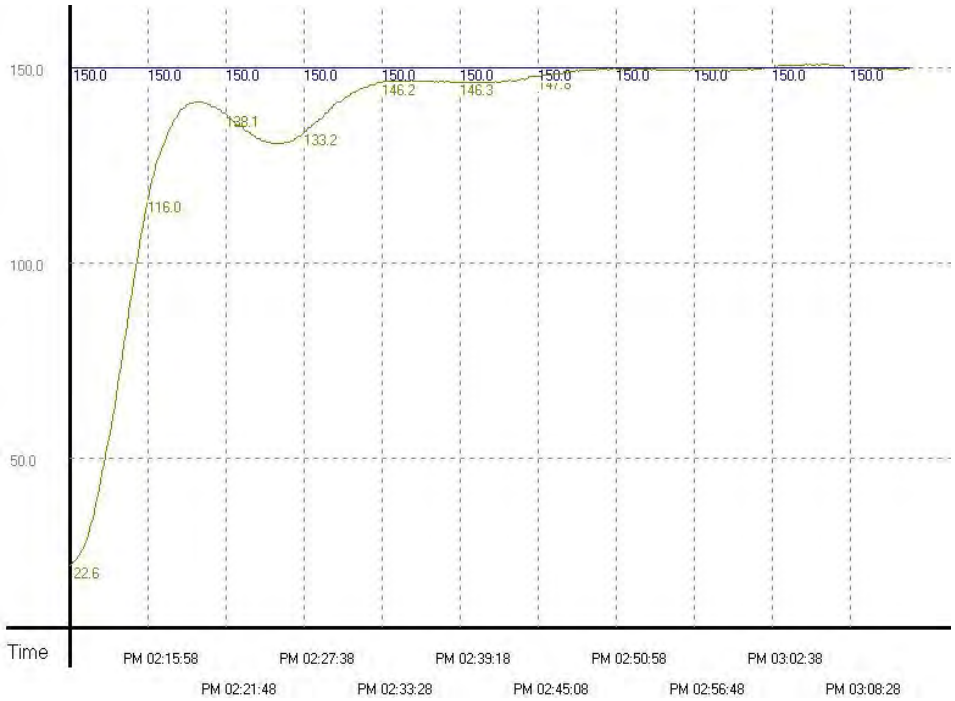
Example 2: Due to that the temperature once exceeds the target temperature, we modify the heating environment into “fast heating environment” (D13 = K16). The results are shown in the diagram below.

From the diagram below, we see that though the temperature no longer exceeds the target temperature, it still needs to take more than 1 hour and 15 minutes to reach the target temperature with ±1ºC inaccuracy. It seems that we have chosen the right environment, but the sampling time is too long, resulting in the extension of heating time.



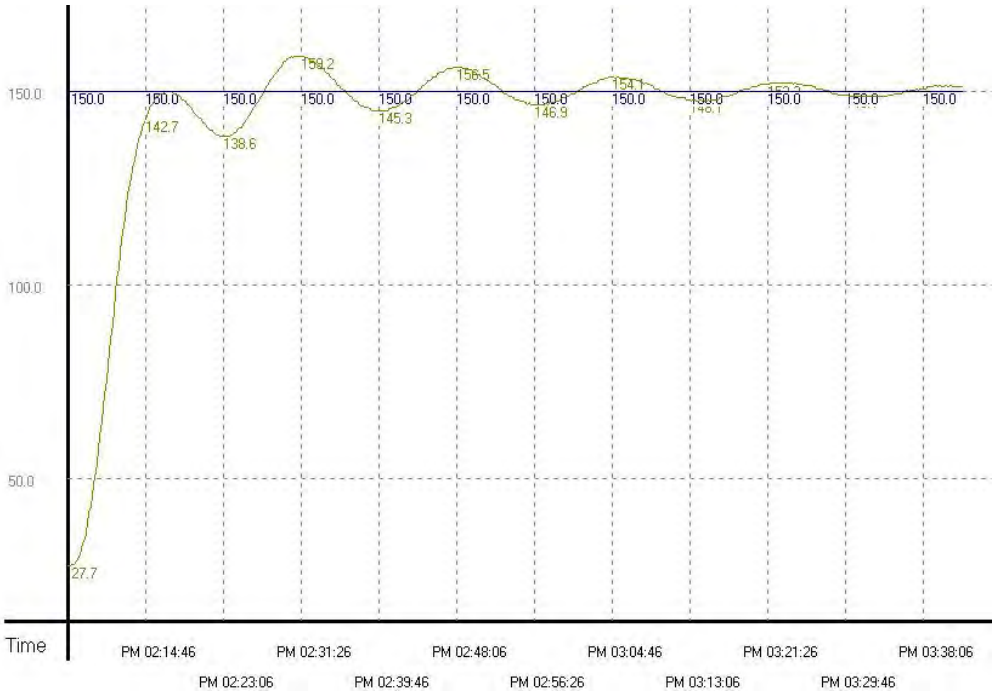
Example 3: To speed up the speed to reach the target temperature, we correct the sampling time as 4 seconds (D12 = K40, D30 = K4,000). The results are shown in the diagram below.

From the diagram below, we see that the overall control time has been shortened as 37 minutes. Therefore, we find out that modifying the sampling time can speed up the time for reaching the target temperature.



Example 4: To see if we can reach the target temperature faster, we modify the sampling time frim example 3 into 2 seconds (D12 = K20, D30 = K2,000). The results are shown in the diagram below.

From the diagram below, we see that the sampling time that is too short will cause the control system to become too sensitive and lead to up and down fluctuations.



API	Mnemonic	Operands	Function
146	CVM	(S1) (S2) (D)	Valve Control

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			CVM: 7 steps
S_2					*	*							*			
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Target time of valve (absolute position) **S_2**: Time from fully-closed to fully-open of valve (destination)

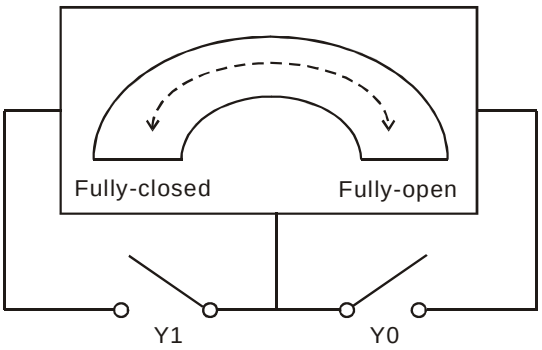
D: Output device

Explanations:

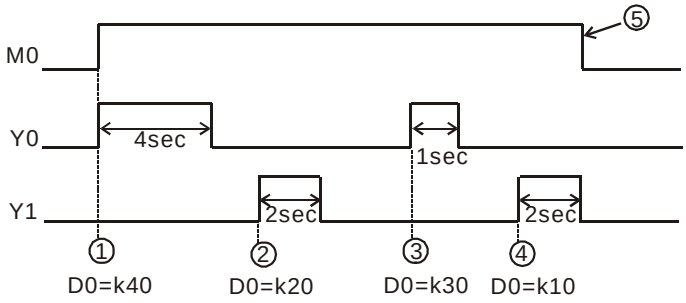
1. **S_1** occupies 3 consecutive registers when in use. **S_1** + 0 are for the user to store the designated value; **S_1** + 1 (the current position of the valve) and **S_1** + 2 are for storing the parameters recorded in the instruction and please DO NOT use and alter these two registers.
2. **D** occupies 2 consecutive output devices when in use. **D** + 0 is the “open” contact and **D** + 1 is the “close” contact.
3. The unit of time: 0.1 second. When the scan time of the program exceeds 0.1 second, DO NOT use this instruction to adjust the position of the valve.
4. Frequency of the output device: 10Hz.
5. When the time of **S_1** + 0 ＞ the fully-opened time set in **S_2**, **D** + 0 will keep being On and **D** + 1 being Off. When the time of **S_1** + 0 ＜ 0, **D** + 0 will keep being Off and **D** + 1 being On.
6. When the instruction is enabled, the instruction will start to control the valve from “0” time position. Therefore, if the user cannot be sure whether the valve is at “0” before executing the instruction, please designate **S_1** + 0 as less than 0 and execute the instruction for **S_2** (time) before sending in the correct target control time.

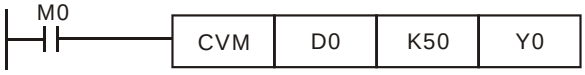
Program Example 1:

1. The control valve



2. Definitions of the control valve:
   a) When Y0 and Y1 = Off: No valve action
   b) When Y0 = On and Y1 = Off: Valve “open”
   c) When Y0 = Off and Y1 = On: Valve “closed”
   d) When Y0 and Y1 = On: The action is prohibited.
3. Timing diagram and program of the control:

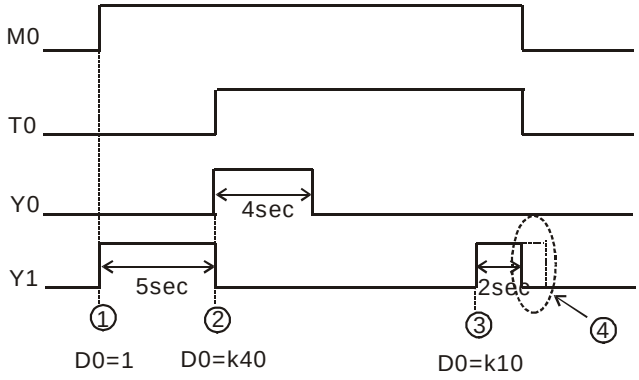


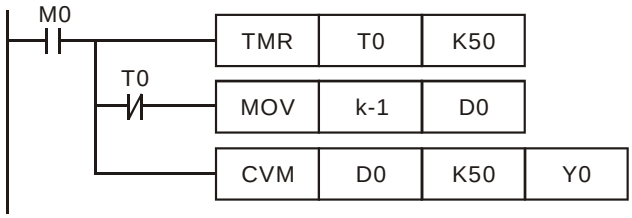


4. Control phases:
   1) Phase ①: When M0 = On, D0 = K40 refers to the valve shall be open (Y0 = On, Y1 = Off) till the position of 4 seconds.
   2) Phase ②: Change the position of the valve and D0 = K20. Due to that the previous position was at 4 seconds, the valve shall be closed (Y0 = Off, Y1 = On) for 2 seconds, moving the valve to the position of 2 seconds.
   3) Phase ③: Change the position of the valve and D0 = K30. Due to that the previous position was at 2 seconds, the valve shall be open (Y0 = On, Y1 = Off) for 1 second, moving the valve to the position of 3 seconds.
   4) Phase ④: Change the position of the valve and D0 = K10. Due to that the previous position was at 2 seconds, the valve shall be closed (Y0 = Off, Y1 = On) for 2 seconds, moving the valve to the position of 1 second.
   5) Phase ⑤: Switch off X0 and no actions at the valve (Y0 = Off, Y1 = Off).

Program Example 2:

1. Timing diagram and program of the control:





2. Control phases:
   1) Phase ①: When M0 = On, due to that we are not sure about there the valve is, set D0 = K-1 to deliberately close the valve (Y0 = Off, Y1 = On) for 5 seconds and make sure the valve is at the position of 0 second before moving on to the next step.
   2) Phase ②: When T0 = On, allow D0 = K40 to start is action. Open the valve (Y0 = On, Y1 = Off) for 4 seconds, moving the valve to the position of 4 seconds.
   3) Phase ③: Change the position of the valve and D0 = K10. Due to that the previous position was at 4 seconds, the valve shall be closed (Y0 = Off, Y1 = On) for 3 seconds, moving the valve to the position of 1 second.
   4) Phase ④: Switch off M0 and the valve will no longer move (Y0 = Off, Y1 = Off).

API	Mnemonic			Operands	Function
147	D	SWAP	P	(S)	Byte Swap

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*	*	*	SWAP, SWAPP: 3 steps DSWAP, DSWAPP: 5 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

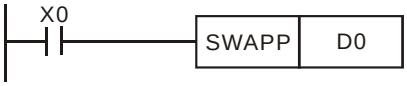
S: Device for swapping 8 high/low byte.

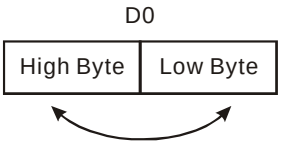
Explanations:

1. If **D** is used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. As 16-bit instruction: the contents in the 8 high bytes and 8 low bytes are swapped.
4. As 32-bit instruction: the 8 high bytes and 8 low bytes in the two registers swap with each other respectively.
5. This instruction adopts pulse execution instructions (SWAPP, DSWAPP).

Program Example 1:

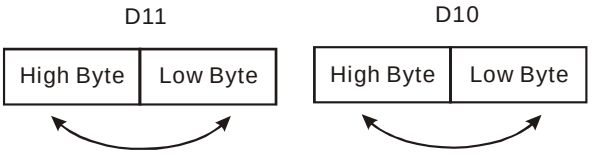
When X0 = On, the high 8 bytes and low 8 bytes in D0 will swap with each other.





Program Example 2:

When X0 = On, the high 8 bytes and low 8 bytes in D11 will swap with each other and the high 8 bytes and low 8 bytes in D10 will swap with each other.



API	Mnemonic			Operands	Function
148	D	MEMR	P	(m) (D) (n)	Read File Register

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
m					*	*							*			MEMR, MEMRP: 7 steps
D													*			DMEMR, DMEMRP: 13 steps
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

m: Address in the file register to be read **D**: Device for storing the read data (starting from the designated D)

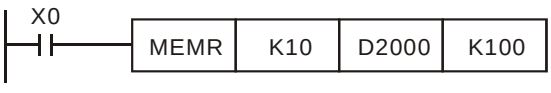
n: Number of data read at a time

Explanations:

1. Range of **m**: K0 ~ K1,599 (SX); K0~49999 (EH3 V1.4/SV2 V1.2 (and above))
2. Range of **D**: D2000 ~ D9999 (SX V3.0 and above); D2000~D11999 (EH3/SV2)
3. Range of **n**: For 16-bit instruction K1 ~ K1,600 (SX), K1 ~ K8,000 (EH3/SV2); For 32-bit instruction K1 ~ K800 (SX), K1 ~ K4,000 (EH3/SV2)
4. See the specifications of each model for their range of use.
5. Flag: M1101. See explanations below.
6. Use this instruction to read the data in file registers and store them into data registers.
7. SX offers 1,600 16-bit file registers.
8. **m** and **n** of SX do not support E and F index register modification.
9. EH3 V1.40 abd above/SV2 V1.20 and above are equipped with 50,000 file registers. The 40,000 file registers added are number 10,000~number 49,999. The 40,000 file registers are stored in the flash ROM. It is suggested that data should be written into the 40,000 file registers by means of WPLSoft or ISPSoft.
10. If **m**, **D** and **n** fall without their range, operation error will occur. M1067, M1068 = On and D1067 will record the error code H'0E1A.

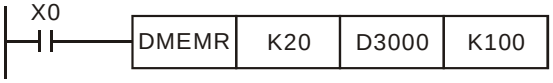
Program Example 1:

1. The 16-bit instruction MEMR reads 100 data at address 10 in the file register and store the read data in register D starting from D2000.
2. When X0 = On, the instruction will be executed. When X0 = Off, the instruction will not be executed and the previously read data will remain unchanged.



Program Example 2:

1. The 32-bit instruction DMEMR reads 100 data at address 20 in the file register and store the read data in register D starting from D3000.
2. When X0 = On, the instruction will be executed. When X0 = Off, the instruction will not be executed and the previously read data will remain unchanged.



API	Mnemonic			Operands	Function
149	D	MEMW	P	(S) (m) (n)	Write File Register

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			MEMW, MEMWP: 7 steps DMEMW, DMEMWP: 13 steps
m					*	*							*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Device for storing the written data (starting from the designated D) **m**: Address in the file register to be written

n: Number of data to be written at a time

Explanations:

1. Range of **S**: D2000 ~ D9999 (SX); D2000~D11999 (EH3/SV2)
2. Range of **m**: K0 ~ K1,599 (SX); K0 ~ K49999 (EH3/SV2)
3. Range of **n**: For 16-bit instruction K1 ~ K1,600 (SX), K1 ~ K8,000 (EH3/SV2); For 32-bit instruction K1 ~ K800 (SX), K1 ~ K4,000 (EH3/SV2)
4. See the specifications of each model for their range of use.
5. Flag: M1101. See explanations below.
6. Use this instruction to read the data in data registers and write them into file registers.
7. SX offers 1,600 16-bit file registers.
8. **m** and **n** of SX do not suppot E and F index register modification.
9. EH3 V1.40 abd above/SV2 V1.20 and above are equipped with 50,000 file registers. The 40,000 file registers added are number 10,000~number 49,999. The 40,000 file registers are stored in the flash ROM. It is suggested that data should be written into the 40,000 file registers by means of WPLSoft or ISPSoft.
10. A DVP-EH3/SV2 series PLC whose version is 1.86 (or above) supports the use of MEMW to write data into the file registers. (DMEMW is not supported. If a memory card is installed, MEMW is not supported.) The number of times users can write data into the file registers should be less than 100000. The writing of data can not be executed continuously, and only one MEMW instruction can be enabled in a scan cycle.
11. **m** should be a value in the table below. (The value is an unsigned value. It is suggested that users should use a hexadecimal value.) **n** should represent 2048 words. If one of the two conditions is not met, data will not written into the PLC, and an operation error will occur. There are 1088 words (48912 ~ 49999) in section 20, but **n** still must be 2048. The PLC will prevent the writing of data from exceeding the range.

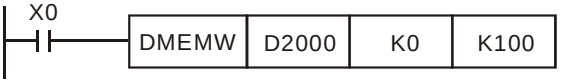
Section number	File register nunmber	Section number	File register nunmber
1	K10000 (H2710)	2	K12048 (H2F10)
3	K14096 (H3710)	4	K16144 (H3F10)
5	K18192 (H4710)	6	K20240 (H4F10)
7	K22288 (H5710)	8	K24336 (H5F10)
9	K26384 (H6710)	10	K28432 (H6F10)

Section number	File register nunmber	Section number	File register nunmber
11	K30480 (H7710)	12	K32528 (H7F10)
13	K34576 (H8710)	14	K36624 (H8F10)
15	K38672 (H9710)	16	K40720 (H9F10)
17	K42768 (HA710)	18	K44816 (HAF10)
19	K46864 (HB710)	20	K48912 (HBF10)

12. It takes about 84 milliseconds for 2048 words to be written into the file registers 10000~49999. It is suggested that the writing of data should be executed when the PLC does not need to operate rapidly (including executing external interrupts).
13. If **S**, **m** and **n** fall without their range, operation error will occur. M1067, M1068 = On and D1067 will record the error code H'0E1A.

Program Example:

1. When X0 = On, the 32-bit instruction DMEMW writes 100 32-bit data starting from D2001 and D2000 into address 0 ~ 199 in the file register.
2. When X0 = On, the instruction will be executed. When X0 = Off, the instruction will not be executed and the previously data written in will remain unchanged.



File Register:

1. EH3/SV2: When the PLC is powered, it will decide whether to automatically send the data in the file register to the designated data register by M1101 (whether to enable the function of file register), D1101 (start address in file register K0 ~ K9,999), D1102 (number of data to be read in file register K1 ~ k8,000), and D1103 (device for storing read data, starting from designated D, K2,000 ~ K9,999).
2. In EH3/SV2, the reading of data from file register to data register D will not be executed if D1101 < 0, D1101 > K9,999, D1103 < K2,000 or D1103 > K9,999.
3. SX: When the PLC is powered, it will decide whether to automatically send the data in the file register to the designated data register by M1101 (whether to enable the function of file register), D1101 (start address in file register K0 ~ K1,599), D1102 (number of data to be read in file register K1 ~ k1,600), and D1103 (device for storing read data, starting from designated D, K2,000 ~ K4,999).
4. In SX, the reading of data from file register to data register D will not be executed if D1101 < 0, D1101 > K1,599, D1103 < K2,000 or D1103 > K4,999.
5. When the reading of data from file register to data register D starts, PLC will stop the reading if the address of file register or data register exceed their range.
6. In PLC program, only API 148 MEMR and API 149 MEMW can be used to read or write the file register. See 2.8.3 for more information on file registers.
7. File registers do not have actual addresses in it. Reading and writing of file registers can only be done through API 148 MEMR, API 149 MEMW or the software WPLSoft or ISPSoft.

8. If the address in the file register to be read exceeds its range, the read value will be 0.
9. Special relays of file register and other relevant special registers:

Flag	Function
M1101	Whether to enable the function of file register; latched; default = off

Special D	Function
D1101	Start address in file register. SX: K0 ~ K1,599; EH3/SV2: K0 ~ K9,999; latched; default = 0
D1102	Number of data to be read in file register. SX: K1 ~ K1,600; EH3/SV2: K1 ~ K8,000; latched; default = 0
D1103	Device for storing read data, starting from designated D. SX: K2,000 ~ K9,999; EH3/SV2: K2,000 ~ K9,999; latched; default = 2,000

MEMO

DVP-PLC applicable to the application instruction. ES includes ES/EX/EC/EC3-8K (FW V8.60 or later) (EC3: FW V8.40 or previous version); SX (FW V3.00); EH3 includes EH3/SV2.

ES/EX/EC series MPU does not support pulse execution type instructions (P instruction).

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	32-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
Communication	150	MODRW	–	–	Read/Write MODBUS Data	✓	✓	✓	✓	11	–
Others	151	PWD	–	–	Detection of Input Pulse Width	–	–	–	✓	5	–
	152	RTMU	–	–	Start of the Measurement of Execution Time of I Interruption	–	–	–	✓	5	–
	153	RTMD	–	–	End of the Measurement of the Execution Time of I Interruption	–	–	–	✓	3	–
	154	RAND	DRAND	✓	Random Number	–	✓	✓	✓	7	13
	168	MVM	DMVM	✓	Move the Designated Bit	–	✓	✓	✓	7	13
	176	MMOV	–	✓	Magnify Move	–	✓	✓	✓	5	–
	177	GPS	–	–	GPS data receiving	–	–	–	✓	5	–
	179	WSUM	DWSUM	✓	Get the Sum	–	✓	✓	✓	7	13
	196	HST	–	✓	High Speed Timer	–	–	–	✓	3	–
Floating Point Operation	172	–	DADDR	✓	Addition of Floating-point Numbers	✓	✓	✓	✓	–	13
	173	–	DSUBR	✓	Subtraction of Floating-point Numbers	✓	✓	✓	✓	–	13
	174	–	DMULR	✓	Multiplication of Floating-point Numbers	✓	✓	✓	✓	–	13
	175	–	DDIVR	✓	Division of Floating-point Numbers	✓	✓	✓	✓	–	13
Position Control	155	–	DABSR	–	Read the Absolute Position from a Servo Motor	–	–	✓	✓	–	13
	156	ZRN	DZRN	–	Zero Return	–	✓	–	✓	9	17
	157	PLSV	DPLSV	–	Adjustable Speed Pulse Output	–	✓	–	✓	7	13
	158	DRVI	DDRVI	–	Drive to Increment	–	✓	–	✓	9	17
	159	DRVA	DDRVA	–	Drive to Absolute	–	✓	–	✓	9	17
	191	–	DPPMR	–	2-Axis Relative Point to Point Motion	–	–	–	✓	–	17
	192	–	DPPMA	–	2-Axis Absolute Point to Point Motion	–	–	–	✓	–	17
	193	–	DCIMR	–	2-Axis Relative Position Arc Interpolation	–	–	–	✓	–	17

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	32-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
	194	–	DCIMA	–	2-Axis Absolute Position Arc Interpolation	–	–	–	✓	–	17
	195	–	DPTPO	–	Single-Axis Pulse Output by Table	–	–	–	✓	–	13
	197	–	DCLLM	–	Close Loop Position Control	–	–	–	✓	–	17
	198	–	DVSPO	–	Variable Speed Pulse Output	–	–	✓	✓	–	17
	199	–	DICF	✓	Immediately Change Frequency	–	–	✓	✓	–	13
Real Time Calendar	160	TCMP	–	✓	Time Compare	–	–	✓	✓	11	–
	161	TZCP	–	✓	Time Zone Compare	–	–	✓	✓	9	–
	162	TADD	–	✓	Time Addition	–	–	✓	✓	7	–
	163	TSUB	–	✓	Time Subtraction	–	–	✓	✓	7	–
	166	TRD	–	✓	Time Read	–	–	✓	✓	3	–
	167	TWR	–	✓	Time Write	–	–	✓	✓	3	–
	169	HOUR	DHOUR	–	Time Compare	–	–	✓	✓	7	13
Gray code	170	GRY	DGRY	✓	BIN → Gray Code	–	✓	✓	✓	5	9
	171	GBIN	DGBIN	✓	Gray Code → BIN	–	✓	✓	✓	5	9
Matrix	180	MAND	–	✓	Matrix 'AND' Operation	–	–	✓	✓	9	–
	181	MOR	–	✓	Matrix 'OR' Operation	–	–	✓	✓	9	–
	182	MXOR	–	✓	Matrix 'XOR' Operation	–	–	✓	✓	9	–
	183	MXNR	–	✓	Matrix 'XNR' Operation	–	–	✓	✓	9	–
	184	MINV	–	✓	Matrix Inverse Operation	–	–	✓	✓	7	–
	185	MCMP	–	✓	Matrix Compare	–	–	✓	✓	9	–
	186	MBRD	–	✓	Read Matrix Bit	–	–	✓	✓	7	–
	187	MBWR	–	✓	Write Matrix Bit	–	–	✓	✓	7	–
	188	MBS	–	✓	Matrix Bit Displacement	–	–	✓	✓	7	–
	189	MBR	–	✓	Matrix Bit Rotation	–	–	✓	✓	7	–
	190	MBC	–	✓	Matrix Bit Status Counting	–	–	✓	✓	7	–

API	Mnemonic	Operands	Function
150	MODRW	S_1 S_2 S_3 S n	Read/Write MODBUS Data

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODRW: 11 steps
S_2					*	*							*			
S_3					*	*							*			
S													*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Address of communication device **S_2**: Function code **S_3**: Device address of data to be read/written

S: Register for storing read/written data (source or destination) **n**: Length of read/written data

Explanations:

1. The content of **S_2** shall only be: K2(H02), K3(H03), K4(H04), K5(H05), K6(H06), K15(H0F), K16(H10), K23(H17).
2. ES/EX/SS series MPU does not support E, F index register modification.
3. The instruction MODRW supports COM1 (RS-232), COM2 (RS-485), and COM3 (a communication card). (COM1 only supports DVP-EH3/SV2 series PLCs. COM3 in DVP-EH3 series PLCs is only applicable to the communication cards DVP-F232, DVP-F485 and DVP-F422.)
4. Flags: M1120 ~ M1131, M1140 ~ M1143. See remarks for more details.
5. **S_1**: Address of communication device; **S_2**: Function code; **S_3**, **S**, **n**: Their functions vary with the function code used. (Please refer to the description below for more information.)
6. **S_1**: must be in the range of K0 to K254. If the function code K2/K3/K4/K23 is used, the address specified can not be K0.
7. **S_2**: Funcation code. Only these function codes are available currently; other function codes are still not executable. See program examples for more information.

Code	Function	Applicable models
H02	Reading several bit devices	EC3-8K, SX V1.8 and EH3/SV2 V1.0 and later versions
H03	Reading several word devices	All series MPU
H04	Reading several word devices (read-only devices)	EH3/SV2 V1.0, SX V3.0, and later versions
H05	Writing a state in a single bit device	EH3/SV2 V1.4, SX V3.0, and later versions
H06	Writing data in a single word device	All series MPU
H0F	Writing states in bit devices	EC3-8K, SX V1.8 and EH3/SV2 V1.0 and later versions
H10	Writing data in word devices	All series MPU
H17	Reading word devices and writing data in word devices	EH3 V1.4, SV2 V1.2, SX V3.0, and later versions

8. **S_3**: Device address of data to be read/written. The device address inside the communication device. If the address is illegal to a designated communication device, the communication device will respond with an error message and PLC will store the error code, and an error flag will be On. If the function code H17 is used, **S_3** can only be a D device, **S_3** is the device from which data will be read, and **S_3**+1 is the device into which data will be written.

   - Error flags and error codes for COM1 and COM2:

PLC COM	COM1	COM2	COM3
Error flag	M1315	M1141	M1319
Error code	D1250	D1130	D1253

   - Exampel: 8000H is an illegal address in a DVP series PLC. If a communication is sent through COM2, M1141 will be On, and the value in D1130 will be 2. If a communication is sent through COM1, M1315 will be On, and the value in D1250 will be 3. If a communication is sent through COM3, M1319 will be On, and the value in D1253 will be 3.

9. **S**: Register for storing read/written data. The user sets up a register and stores the data to be written in the register in advance. The register can be register for storing the read data. If the function code K23 is used, **S** is a D device index which indicates the device in which the communication data string received will be stored, and **S**+1 is a D device index which indicates the device in which the data which will be written is stored. If a reading function code (K2, K3, K4, or K23) is sent through COM2, the communication data string received will be stored in the register indicated by **S**, and the conversion data will be stored in D1296~D1311. Please refer to program example 1 and program example 3 for more information. If a reading function code (K2, K3, K4, or K23) is sent through COM1 or COM3, the conversion data will be stored in the register indicated by **S**. Please refer to program example 2 and program example 4 for more information. Users can refer to program example 13 and program example 14 for more information about the function code K23. If COM2 is used, the communication data which will be sent is stored in D1256~D1295.

10. **n**: Length of read/written data.

    In Modbus function code H05 (force On/Off), n=0: Off, n=1: On.

    In Modbus function code H02, H03, H04, H0F, H10, H17 (data length), the range = K1 ~ Km. See the table below for m upon different models and communication modes, in which the unit of H02 and H0F is a bit, and the unit of H03, H04, H10, and H17 is a word. If function code H17 is used, **n** can only be a D device, **n** represents the number of data which will be read, and **n** +1 represents the number of data which will be written.

Communication mode	Model	H02	H03	H04	H0F	H10	H17
COM1 RTU Mode (M1139 On)	EH3 SV2	K256	K24	K24	K256	K24	K24
COM1 ASCII Mode (M1139 Off)	EH3 SV2	K256	K24	K24	K256	K24	K24
COM2 RTU Mode (M1143 On)	ES	Not supported	K16	Not supported	Not supported	K16	Not supported
	EC3-8K, SX	K64	K16	K16	K64	K6	K16
	EH3/SV2	K256	K16	K16	K256	K16	K16

Communication mode	Model	H02	H03	H04	H0F	H10	H17
COM2 ASCII Mode (M1143 Off)	ES	Not supported	K8	Not supported	Not supported	K8	Not supported
	EC3-8K, SX	K64	K8	K16	K64	K8	K16
	EH3/SV2	K256	K16	K16	K256	K16	K16

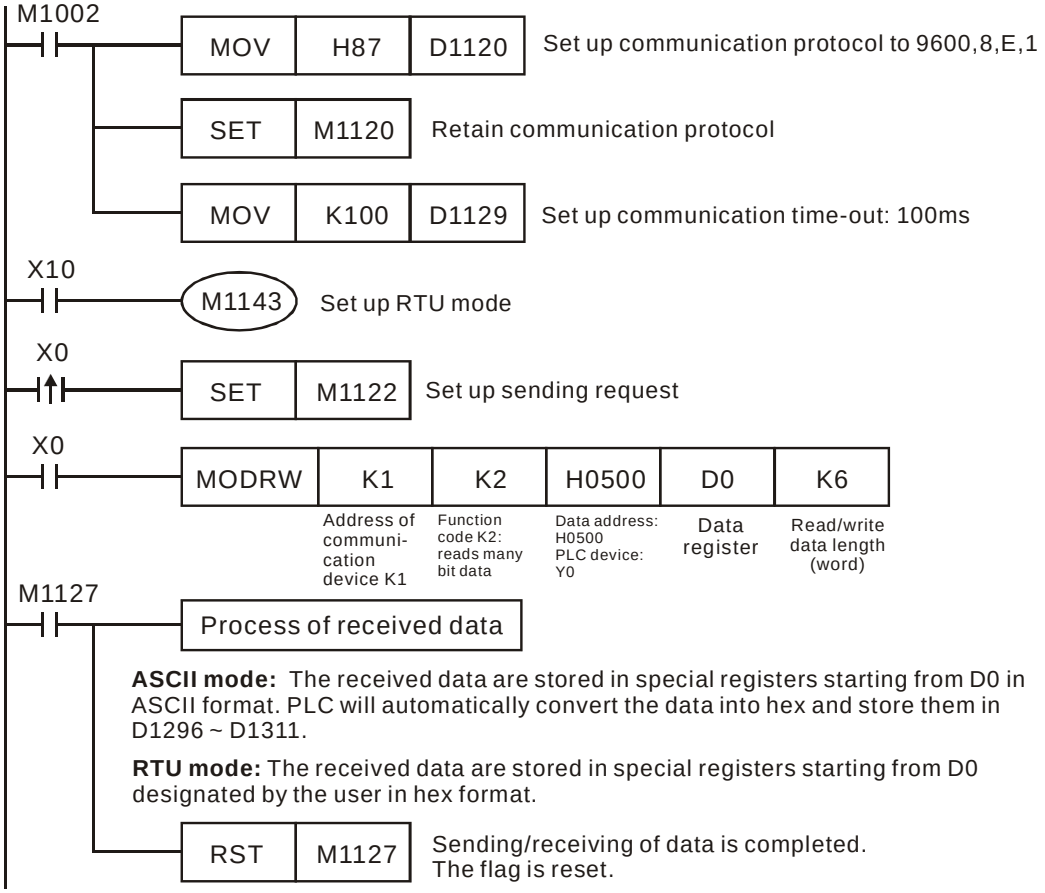
11. The functions of **S_3**, **S**,and **n** vary with the function code used.

Function code	**S_3**	**S**	**n**
H02	Address from which data is read	Register where data read is stored	Number of data read
H03	Address from which data is read	Register where data read is stored	Number of data read
H04	Address from which data is read	Register where data read is stored	Number of data read
H05	Address into which data is written	None	State value written
H06	Address into which data is written	Data register where data written is stored	None
H0F	Address into which data is written	Data register where data written is stored	Number of data written
H10	Address into which data is written	Data register where data written is stored	Number of data written
H17	**S_3**: Address from which data is read **S_3**+1:Address into which data is written	**S**: Register where data read is stored **S**+1: Data register where data written is stored	**n**: Number of data read **n**+1: Number of data written

12. There is no limitation on the times of using this instruction. However, only one instruction can be executed at a time.
13. Rising-edge contact (LDP, ANDP, ORP) and falling-edge contact (LDF, ANDF, ORF) can not be used as drive contact of MODRW (Function code H02, H03, H04, H17) instruction, otherwise the data stored in the receiving registers will be incorrect.
14. MODRW instruction determines the COM port according to the communication request. The COM port determination is made following the order: COM1→COM3→COM2. Therefore, please insert every MODRW instruction right after the sending request instruction for avoiding errors on the target location for data access.
15. For detailed explanation of the associated flags and special registers, please refer to the remarks on API 80 RS instruction.

Program Example 1:

1. Sending the function code K2 (H02) through COM2 (RS-485): Read many bit devices. The read communication code will be placed in the register designated by the 4th operand of the instruction. In the example below, K6 refers to the data length (bit). Assume Y2=Y4=Y5=Y11=Y14=On for Y0 ~ Y16 status.



2. ASCII Mode: When PLC1 is connected to PLC2

When X0 = On, function code 02 of MODRW instrruction will start to be executed.

PLC1⇨ PLC2, PLC1 sends: "**01 02 0500 0010 E8**"

PLC2 ⇨ PLC1, PLC1 receives: "**01 02 02 34 12 B5**"

Registers for PLC1 sent data (sending messages)

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1256 High	'1'	31 H		
D1257 Low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 High	'2'	32 H		
D1258 Low	'0'	30 H	Starting Data Address	
D1258 High	'5'	35 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'0'	30 H	Number of Data (counted by bits)	
D1260 High	'0'	30 H		
D1261 Low	'1'	31 H		
D1261 High	'0'	30 H		
D1262 Low	'E'	45 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1262 High	'8'	38 H	LRC CHK 0	

Register (D0) for PLC1 received data (responding messages):

Register	DATA		Explanation	
D0 Low	'0'	30 H	ADR 1 ADR 0	
D0 High	'1'	31 H		
D1 Low	'0'	30 H	CMD 1 CMD 0	
D1 High	'2'	33 H		
D2 Low	'0'	30 H	Number of data (counted by bytes)	
D2 High	'2'	32 H		
D3 Low	'3'	33 H	Content in address 0500 ~ 0505	PLC automatically convert ASCII words and store the result in D1296 = H1234 (b0 ~ b5 are valid)
D3 High	'4'	34 H		
D4 Low	'1'	31H		
D4 High	'2'	32H		
D5 Low	'B'	52H	LRC CHK 1	
D5 High	'5'	35 H	LRC CHK 0	

3. RTU Mode: When PLC1 is connected to PLC2

When X10 = On, function code 02 of MODRW instruction will start to be executed.

PLC1⇨ PLC2, PLC1 sends: "**01 02 0500 0010 79 0A**"

PLC2 ⇨ PLC1, PLC1 receives: "**01 02 02 34 12 2F 75**"

Registers for PLC sent data (sending messages):

Register	DATA	Explanation
D1256 low	1 H	Address
D1257 low	2 H	Function
D1258 low	5 H	Starting data address
D1259 low	0 H	
D1260 low	0 H	Number of data (counted by words)
D1261 low	10 H	
D1262 low	79 H	CRC CHK Low
D1263 low	0A H	CRC CHK High

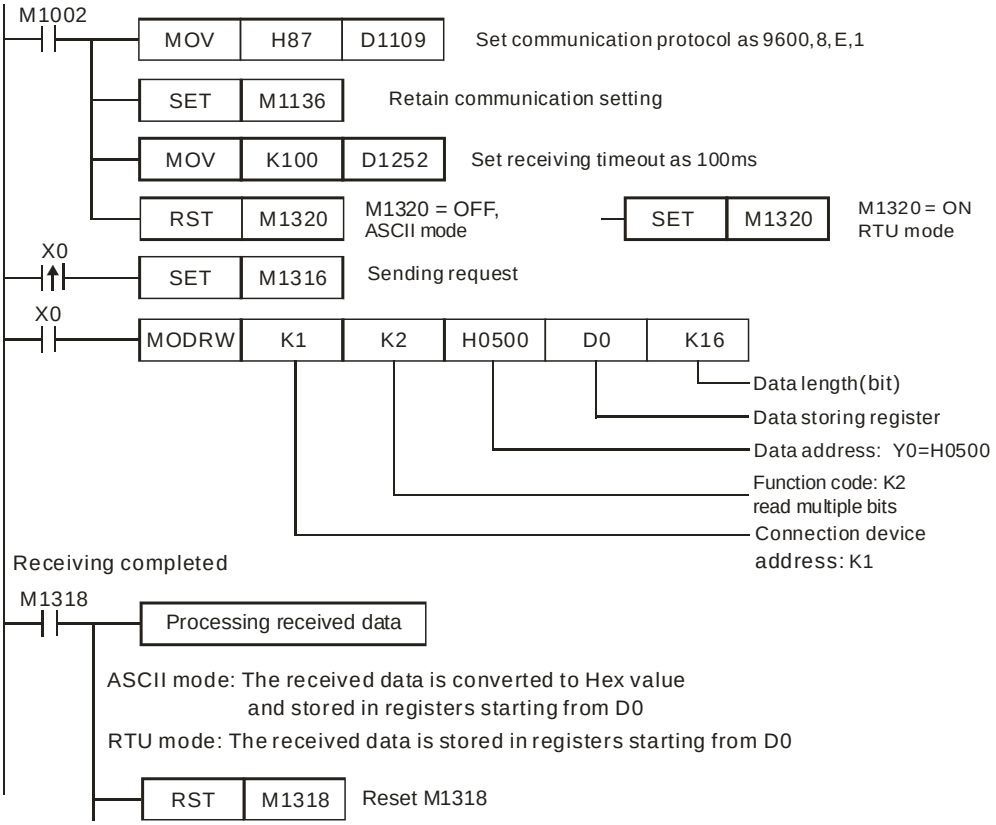
Register (D0) for PLC received data (responding messages):

Register	DATA	Explanation	
D0 low	01 H	Address	
D1 low	02 H	Function	
D2 low	02 H	Number of data (byte)	
D3 low	34 H	Content in address 0500H	PLC automatically stores the value in D1296=H1234 (b0 ~ b5 are valid)
D4 low	12 H		
D5 low	2F H	CRC CHK Low	
D6 low	75 H	CRC CHK High	

Program Example 2: COM1(RS-232) / COM3(RS-485), Function Code H02

1. Function code K2 (H02): read multiple bit devices. Up to 64 bits can be read.
2. PLC1 connects to PLC2: (M1320 = OFF, ASCII mode), (M1320 = ON, RTU mode)
3. For both ASCII and RTU modes, PLC COM1/COM3 only stores the received data in registers starting from **S**, and will not store the data to be sent. The stored data can be transformed and moved by using DTM instruction for applications of other purposes.
4. Take the connection between PLC1 (PLC COM3) and PLC2(PLC COM1) for example, the tables below explains the status when PLC1 reads Y0~Y17 of PLC2
   - If PLC1 applies COM1 for communication, the below program can be usable by changing:
     1. D1109→D1036: communication protocol

2. M1136→M1138: retain communication setting
3. D1252→D1249: Set value for data receiving timeout
4. M1320→M1139: ASCII/RTU mode selection
5. M1316→M1312: sending request
6. M1318→M1314: receiving completed flag



- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):

  When X0 = ON, MODRW instruction executes the function specified by Function Code H02

  PLC1⇨ PLC2, PLC1 sends: "**01 02 0500 0010 E8**"

  PLC2 ⇨PLC1, PLC1 receives: "**01 02 02 3412 B5**"

  PLC1 data receiving register D0

Register	Data	Descriptions
D0	1234H	PLC converts the ASCII data in address 0500H~0515H and stores the converted data automatically.

Analysis of the read status of PLC2 Y0~Y17: 1234H

Device	Status	Device	Status	Device	Status	Device	Status
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

- RTU mode (COM3: M1320 = ON, COM1: M1139 = ON):

  When X0 = ON, MODRW instruction executes the function specified by Function Code H02

  PLC1 ⇨ PLC2, PLC1 sends: "**01 02 0500 0010 79 0A**"

  PLC2 ⇨ PLC1, PLC1 receives: "**01 02 02 34 12 2F 75**"

  PLC data receiving register:

Register	Data	Descriptions
D0	1234 H	PLC converts the data in address 0500H ~ 0515H and stores the converted data automatically.

  Analysis of the read status of PLC2 Y0~Y17: 1234H

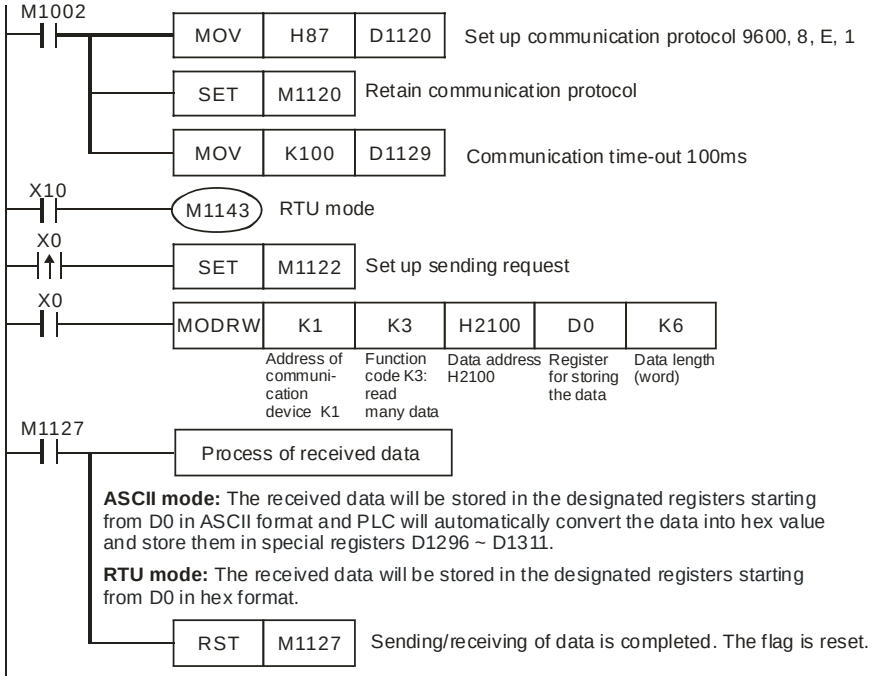
Device	Status	Device	Status	Device	Status	Device	Status
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	On	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

5. Relative flags and data registers when COM1 / COM2 / COM3 works as Master:

	COM2	COM1	COM3	Function
COM setting	M1120	M1138	M1136	Retain communication setting
	M1143	M1139	M1320	ASCII/RTU mode selection
	D1120	D1036	D1109	Communication protocol
	D1121	D1121	D1255	PLC communication address
Sending request	M1122	M1312	M1316	Sending request
	D1129	D1249	D1252	Set value for data receiving timeout (ms)
Receiving completed	M1127	M1314	M1318	Data receiving completed
Errors	-	M1315	M1319	Data receiving error
	-	D1250	D1253	Communication error code
	M1129	-	-	Receiving timeout
	M1140	-	-	Data receiving error
	M1141	-	-	Parameter error. Exception Code is stored in D1130
	D1130	-	-	Error code (Exception code) returning from Modbus communication

Program Example 3:

1. Sending the function code K3(H03) through COM2 (RS-485) (H04 is used the same as H03.): For reading many data in register
   When PLC is connected to VFD-S AC motor drive: M1143 = Off, in ASCII mode
   When PLC is connected to VFD-S AC motor drive: M1143 = On, in RTU mode
2. When in ASCII mode, the received data will be stored in the designated registers starting from D0 in ASCII format and PLC will automatically convert the data into hex value and store them in special registers D1296 ~ D1311. When the conversion into hex value starts, M1131 will be On and turn Off when the conversion is completed.
3. If necessary, the user can move the hex values stored in D1296 ~ D1131 to other general registers by using MOV, DMOV or BMOV instruction. Other instructions of ES/EX/SS do not function on the data in D1296 ~ D1311.
4. When in RTU mode, the received data will be stored in the designated registers starting from D0 in hex format.
5. When In ASCII mode or RTU mode, PLC will store the data to be sent in D1256 ~ D1295. If necessary, the user can move the data to other general registers by using MOV, DMOV or BMOV instruction. Other instructions of ES/EX/EC do not function on the data in D1256 ~ D1295.
6. The data sent back from AC motor drive are stored in the registers designated by the user. After the transmission is completed, PLC will auto-check if the received data are incorrect. M1140 will be On if there is an error.
7. If the device address is illegal to a designated communication device, the communication device will respond with an error message and PLC will store the error code in D1130 and M1141 = On. For example, if 8000H is illegal to VFD-S, M1141 will be On and D1130 = 2. See user manual of VFD-S for error codes.
8. After M1140 = On or M1141 = On, PLC will send another correct datum to AC motor drive. If the data sent back from AC motor drive is correct, M1140 and M1141 will be reset.



9. ASCII Mode: When PLC is connected to VFD-S AC motor drive.

PLC ⇨ VFD-S, PLC sends: **"01 03 2100 0006 D5"**

VFD-S ⇨ PLC, PLC receives: **"01 03 0C 0100 1766 0000 0000 0136 0000 3B"**

Registers for sent data (sending messages)

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1 ADR 0	Address of AC motor drive: ADR (1,0)
D1256 High	'1'	31 H		
D1257 Low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 High	'3'	33 H		
D1258 Low	'2'	32 H	Starting Data Address	
D1258 High	'1'	31 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'0'	30 H	Number of Data (counted by words)	
D1260 High	'0'	30 H		
D1261 Low	'0'	30 H		
D1261 High	'6'	36 H		
D1262 Low	'D'	44 H	LRC CHK 1 LRC CHK 0	Error checksum: LRC CHK (0,1)
D1262 High	'5'	35 H		

Registers for received data D0 (responding messages)

Register	DATA		Explanation	
D0 Low	'0'	30 H	ADR 1	
D0 High	'1'	31 H	ADR 0	
D1 Low	'0'	30 H	CMD 1	
D1 High	'3'	33 H	CMD 0	
D2 Low	'0'	30 H	Number of Data (counted by byte)	
D2 High	'C'	43 H		
D3 Low	'0'	30 H	Content of address 2100H	PLC automatically convert ASCII codes to numerals and store the numeral in D1296 = H0100
D3 High	'1'	31 H		
D4 Low	'0'	30 H		
D4 High	'0'	30 H		
D5 Low	'1'	31 H	Content of address 2101H	PLC automatically convert ASCII codes to numerals and store the numeral in D1297 = H1766
D5 High	'7'	37 H		
D6 Low	'6'	36 H		
D6 High	'6'	36 H		
D7 Low	'0'	30 H	Content of address 2102H	PLC automatically convert ASCII codes to numerals and store the numeral in D1298 = H0000
D7 High	'0'	30 H		
D8 Low	'0'	30 H		
D8 High	'0'	30 H		
D9 Low	'0'	30 H	Content of address 2103H	PLC automatically convert ASCII codes to numerals and store the numeral in D1299 = H0000
D9 High	'0'	30 H		
D10 Low	'0'	30 H		
D10 High	'0'	30 H		
D11 Low	'0'	30 H	Content of address 2104H	PLC automatically convert ASCII codes to numerals and store the numeral in D1300 = H0136
D11 High	'1'	31 H		
D12 Low	'3'	33 H		
D12 High	'6'	36 H		
D13 Low	'0'	30 H	Content of address 2105H	PLC automatically convert ASCII codes to numerals and store the numeral in D1301 = H0000
D13 High	'0'	30 H		
D14 Low	'0'	30 H		
D14 High	'0'	30 H		
D15 Low	'3'	33 H	LRC CHK 1	
D15 High	'B'	42 H	LRC CHK 0	

10. RTU Mode: When PLC is connected to VFD-S AC motor drive

    PLC ⇨ VFD-S, PLC sends: **"01 03 2100 0006 CF F4"**

    VFD-S ⇨ PLC, PLC receives: **"01 03 0C 0000 0503 0BB8 0BB8 0000 012D 8E C5"**

    Registers for sent data (sending messages)

Register	DATA	Explanation
D1256 Low	01 H	Address
D1257 Low	03 H	Function
D1258 Low	21 H	Starting Data Address
D1259 Low	00 H	
D1260 Low	00 H	Number of Data (counted by words)
D1261 Low	06 H	
D1262 Low	CF H	CRC CHK Low
D1263 Low	F4 H	CRC CHK High

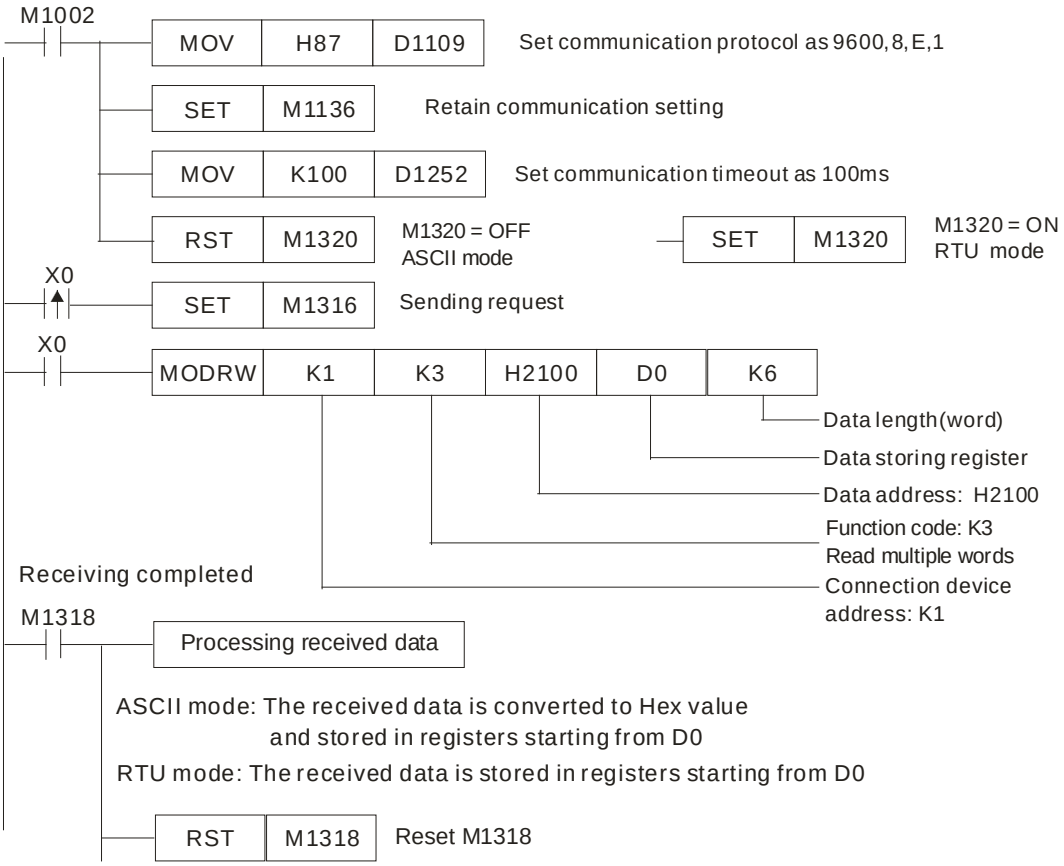
    Registers for received data D0 (responding messages)

Register	DATA	Explanation	
D0 Low	01 H	Address	
D1 Low	03 H	Function	
D2 Low	0C H	Number of Data (byte)	
D3 Low	00 H	Content of address 2100H	PLC automatically convert ASCII codes to numerals and store the numeral in D1296 = H0000
D4 Low	00 H		
D5 Low	05 H	Content of address 2101H	PLC automatically convert ASCII codes to numerals and store the numeral in D1297 = H0503
D6 Low	03 H		
D7 Low	0B H	Content of address 2102H	PLC automatically convert ASCII codes to numerals and store the numeral in D1298 = H0BB8
D8 Low	B8 H		
D9 Low	0B H	Content of address 2103H	PLC automatically convert ASCII codes to numerals and store the numeral in D1299 = H0BB8
D10 Low	B8 H		
D11 Low	00 H	Content of address 2104H	PLC automatically convert ASCII codes to numerals and store the numeral in D1300 = H0000
D12 Low	00 H		
D13 Low	01 H	Content of address 2105H	PLC automatically convert ASCII codes to numerals and store the numeral in D1301 = H012D
D14 Low	2D H		
D15 Low	8E H	CRC CHK Low	
D16 Low	C5 H	CRC CHK High	

Program example 4: COM1(RS-232) / COM3(RS-485), Function Code H03 (The function code H04 is the same as the function code H03.)

1. Function code K3 (H03): read multiple word devices, up to 16 words can be read. For COM2 ASCII mode, only 8 words can be read.
2. PLC COM1 / COM3 stores the received data in registers starting from **S**, and the stored data can be transformed and moved by using DTM instruction for applications of other purposes.
3. Take the connection between PLC and VFD-B for example, the tables below explains the status when PLC reads VFD-B status. (M1320 = OFF, ASCII Mode), (M1320 = ON, RTU Mode)

   - If PLC applies COM1 for communication, the below program can be usable by changing:
     1. D1109→D1036: communication protocol
     2. M1136→M1138: retain communication setting
     3. D1252→D1249: Set value for data receiving timeout
     4. M1320→M1139: ASCII/RTU mode selection
     5. M1316→M1312: sending request
     6. M1318→M1314: receiving completed flag



- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):

  When X0 = ON, MODRW instruction executes the function specified by Function Code H03

  PLC ⇨ VFD-B, PLC sends: "**01 03 2100 0006 D5**"

  VFD-B ⇨ PLC, PLC receives: "**01 03 0C 0100 1766 0000 0000 0136 0000 3B**"

  Registers for received data (responding messages)

Register	Data	Descriptions
D0	0100 H	PLC converts ASCII codes in 2100 H and stores the converted data automatically.
D1	1766 H	PLC converts ASCII codes in 2101 H and stores the converted data automatically.
D2	0000 H	PLC converts ASCII codes in 2102 H and stores the converted data automatically.
D3	0000 H	PLC converts ASCII codes in 2103 H and stores the converted data automatically.
D4	0136 H	PLC converts ASCII codes in 2104 H and stores the converted data automatically.
D5	0000 H	PLC converts ASCII codes in 2105 H and stores the converted data automatically.

- RTU mode (COM3: M1320 = ON COM1: M1139 = ON):

  When X0 = ON, MODRW instruction executes the function specified by Function Code H03

  PLC ⇨ VFD-B, PLC sends: " **01 03 2100 0006 CF F4**"

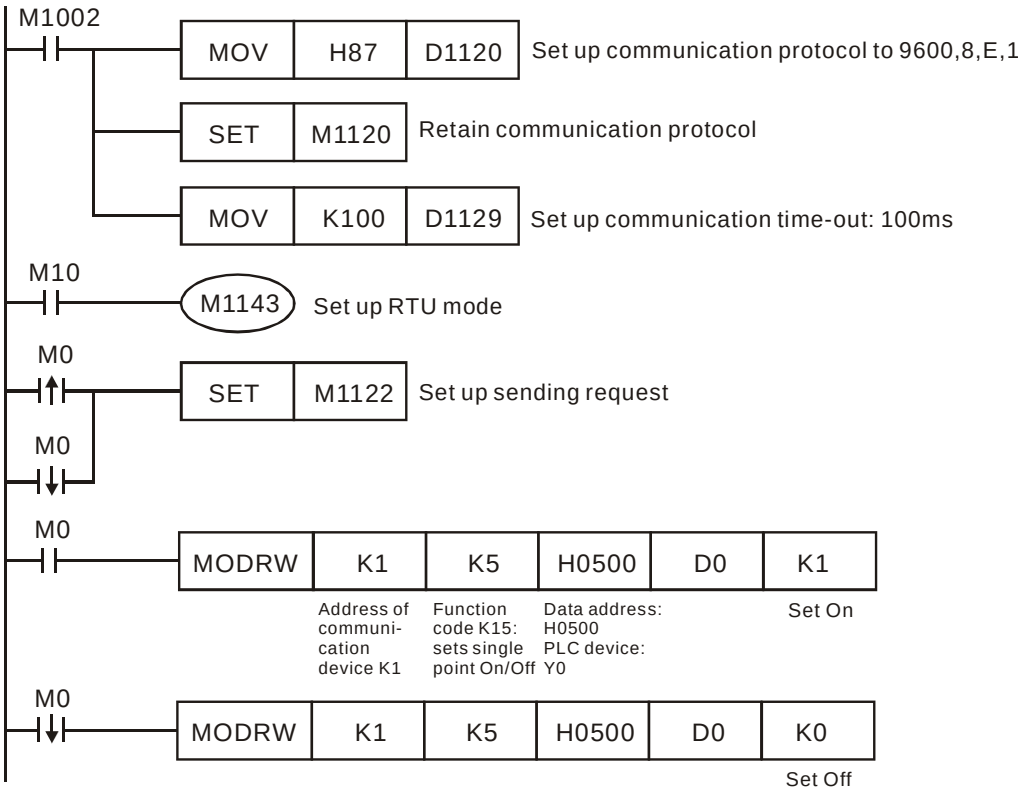
  VFD-B ⇨ PLC, PLC receives: "**01 03 0C 0000 0503 0BB8 0BB8 0000 012D 8E C5**"

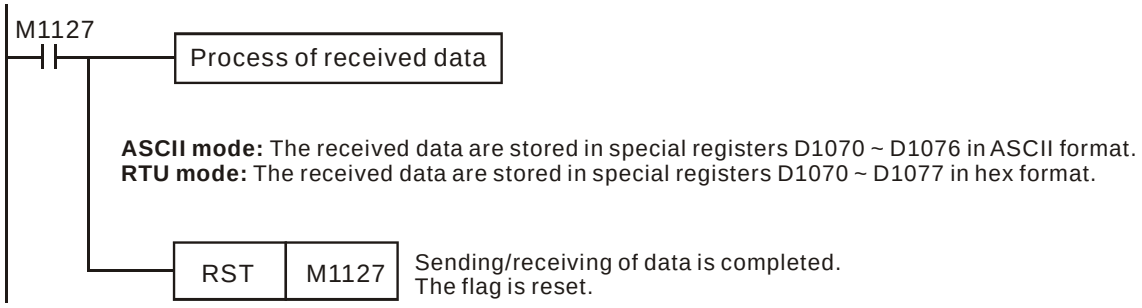
  Registers for received data (responding messages)

Register	Data	Descriptions
D0	0000 H	PLC converts data in 2100 H and stores the converted data automatically.
D1	0503 H	PLC converts data in 2101 H and stores the converted data automatically.
D2	0BB8 H	PLC converts data in 2102 H and stores the converted data automatically.
D3	0BB8 H	PLC converts data in 2103 H and stores the converted data automatically.
D4	0136 H	PLC converts data in 2104 H and stores the converted data automatically.
D5	012D H	PLC converts data in 2105 H and stores the converted data automatically.

Program Example 5:

1. Sending the function code K5(H05) through COM2 (RS-485): Write status of single bit device. In the example below, Set K1 to bit On, K0 to bit Off.





2. ASCII Mode: When PLC1 is connected to PLC2

When M0 = On, function code 05 (bit On) of MODRW instruction will start to be executed.

PLC1⇨ PLC2, PLC1 sends: "**01 05 0500 FF00 F6**"

PLC2 ⇨ PLC1, PLC1 receives: "**01 05 0500 FF00 F6**"

Registers for PLC1 sent data (sending messages):

Register	DATA		Explanation	
D1256 low	'0'	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1256 high	'1'	31 H		
D1257 low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 high	'5'	35 H		
D1258 low	'0'	30 H	Starting data address	
D1258 high	'5'	35 H		
D1259 low	'0'	30 H		
D1259 high	'0'	30 H		
D1260 low	'F'	46 H	Request bit On/Off	
D1260 high	'F'	46 H		
D1261 low	'0'	30 H		
D1261 high	'0'	30 H		
D1262 low	'F'	46 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1262 high	'6'	36 H	LRC CHK 0	

Registers (D0) for PLC1 received data (responding messages):

Register	DATA		Explanation	
D1070 low	'0'	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1070 high	'1'	31 H		
D1071 low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1071 high	'5'	35 H		
D1072 low	'0'	30 H	Starting data address	
D1072 high	'5'	35 H		
D1073 low	'0'	30 H		
D1073 high	'0'	30 H		
D1074 low	'F'	46 H	Request bit On/Off	
D1074 high	'F'	46 H		
D1075 low	'0'	30 H		
D1075 high	'0'	30 H		
D1076 low	'F'	46 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1076 high	'6'	36 H	LRC CHK 0	

When M0 = Off, function code 05 (bit Off) will start to be executed.

PLC1⇨ PLC2, PLC1 sends: "**01 05 0500 FF00 F6**"

PLC2 ⇨ PLC1, PLC1 receives: "**01 05 0500 FF00 F6**"

Registers for PLC1 sent data (sending messages):

Register	DATA		Explanation	
D1256 low	‘0’	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1256 high	‘1’	31 H		
D1257 low	‘0’	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 high	‘5’	35 H		
D1258 low	‘0’	30 H	Starting data address	
D1258 high	‘5’	35 H		
D1259 low	‘0’	30 H		
D1259 high	‘0’	30 H		
D1260 low	‘0’	30 H	Request bit On/Off	
D1260 high	‘0’	30 H		
D1261 low	‘0’	30 H		
D1261 high	‘0’	30 H		
D1262 low	‘F’	46 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1262 high	‘5’	35 H	LRC CHK 0	

Registers (D0) for PLC1 received data (responding messages):

Register	DATA		Explanation	
D1070 low	‘0’	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1070 high	‘1’	31 H		
D1071 low	‘0’	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1071 high	‘5’	35 H		
D1072 low	‘0’	30 H	Starting data address	
D1072 high	‘5’	35 H		
D1073 low	‘0’	30 H		
D1073 high	‘0’	30 H		
D1074 low	‘0’	30 H	Request bit On/Off	
D1074 high	‘0’	30 H		
D1075 low	‘0’	30 H		
D1075 high	‘0’	30 H		
D1076 low	‘F’	46 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1076 high	‘5’	35 H	LRC CHK 0	

3. RTU Mode: When PLC1 is connected to PLC2

When M0 = On, function code 05 (bit On) of MODRW instruction will start to be executed.

PLC1⇨ PLC2, PLC1 sends: “**01 05 0500 FF00 8C F6**”

PLC2 ⇨ PLC1, PLC1 receives: “**01 05 0500 FF00 8C F6**”

Registers for PLC sent data (sending messages):

Register	DATA	Explanation
D1256 low	01 H	Address
D1257 low	05 H	Function
D1258 low	05 H	Starting data address
D1259 low	00 H	
D1260 low	FF H	Set bit On/Off Request bit ON/OFF
D1261 low	00 H	
D1262 low	8C H	CRC CHK Low
D1263 low	F6 H	CRC CHK High

Registers (D0) for PLC received data (responding messages):

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	05 H	Function

Register	DATA	Explanation
D1072 low	05 H	Starting data address
D1073 low	00 H	
D1074 low	FF H	Set bit On/Off
D1075 low	00 H	Request bit ON/OFF
D1076 low	8C H	CRC CHK Low
D1077 low	F6 H	CRC CHK High

When M10 = Off, function code 05 (bit Off) of MODRW instruction will start to be executed.

PLC1⇨ PLC2, PLC1 sends: "**01 05 0500 0000 CD 06**"

PLC2 ⇨ PLC1, PLC1 receives: "**01 05 0500 0000 CD 06**"

Registers for PLC sent data (sending messages):

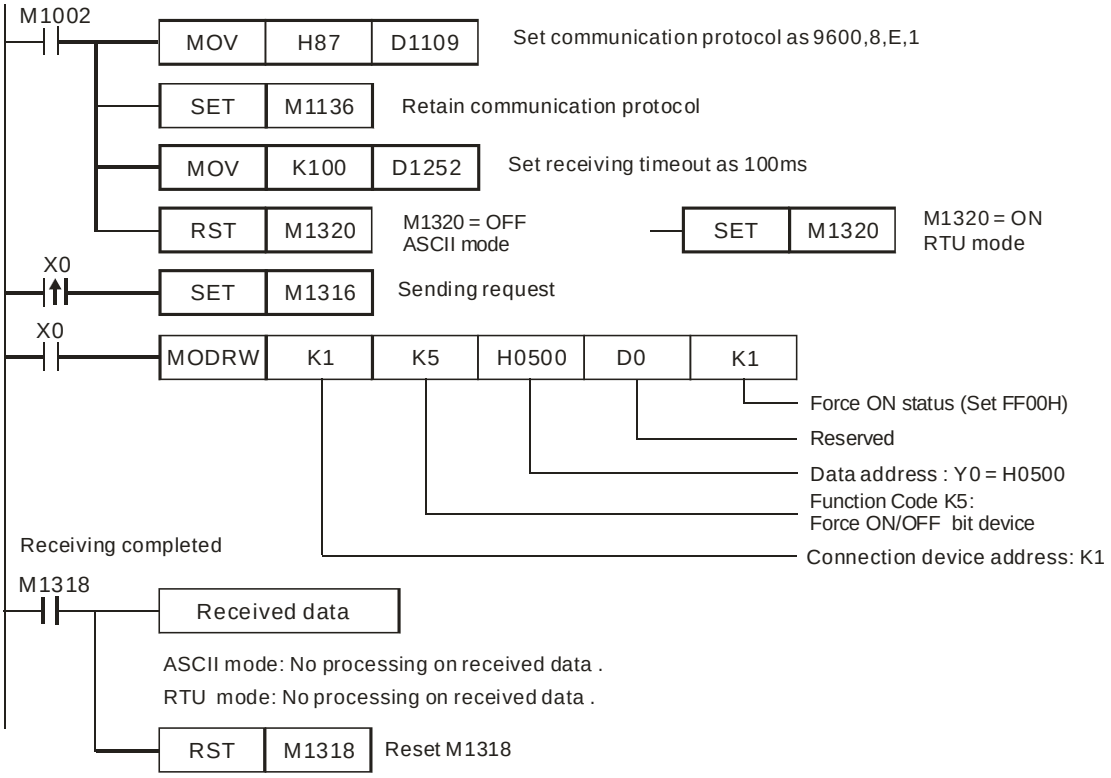
Register	DATA	Explanation
D1256 low	01 H	Address
D1257 low	05 H	Function
D1258 low	05 H	Starting data address
D1259 low	00 H	
D1260 low	00 H	Set bit On/Off
D1261 low	00 H	Request bit ON/OFF
D1262 low	CD H	CRC CHK Low
D1263 low	06 H	CRC CHK High

Registers (D0) for PLC received data (responding messages):

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	05 H	Function
D1072 low	05 H	Starting data address
D1073 low	00 H	
D1074 low	00 H	Set bit On/Off
D1075 low	00 H	Request bit ON/OFF
D1076 low	CD H	CRC CHK Low
D1077 low	06 H	CRC CHK High

Program example 6: COM1(RS-232) / COM3(RS-485), Function Code H05

1. Function Code K5 (H05): Force ON/OFF bit device.
2. PLC1 connects PLC2: (M1320 = OFF, ASCII Mode), (M1320 = ON, RTU Mode)
3. **n** = 1 indicates Force ON (set FF00H) and **n** = 0 indicates Force OFF (set 0000H)
4. PLC COM1/COM3 will not process the received data.
5. Take the connection between PLC1 (PLC COM3) and PLC2(PLC COM1) for example, the tables below explains the status when PLC1 reads Y0~Y17 of PLC2
   - If PLC1 applies COM1 for communication, the below program can be usable by changing:
     1. D1109→D1036: communication protocol
     2. M1136→M1138: retain communication setting
     3. D1252→D1249: Set value for data receiving timeout
     4. M1320→M1139: ASCII/RTU mode selection
     5. M1316→M1312: sending request
     6. M1318→M1314: receiving completed flag

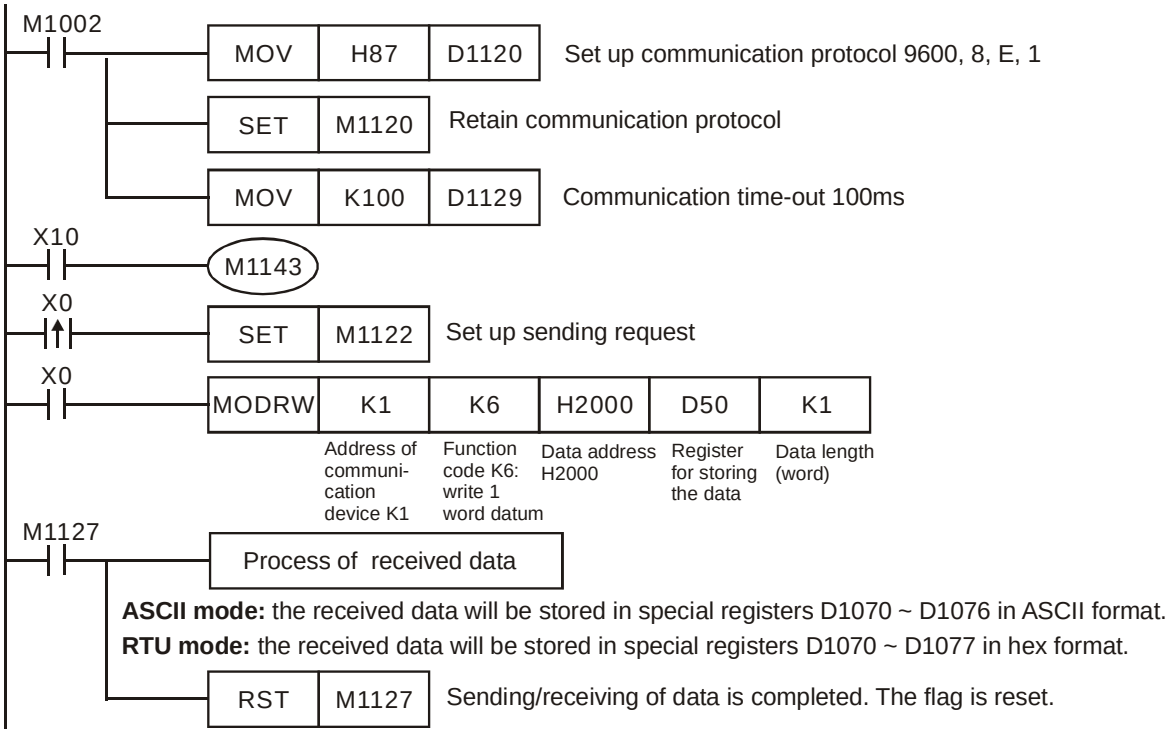


- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):
  When X0 = ON, MODRW instruction executes the function specified by Function Code H05
  PLC1 ⇨ PLC2, PLC sends: "**01 05 0500 FF00 6F**"
  PLC2 ⇨ PLC1, PLC receives: "**01 05 0500 FF00 6F**"
  (No data processing on received data)
- RTU mode (COM3: M1320 = ON, COM1: M1139 = ON):
  When X0 = ON, MODRW instruction executes the function specified by Function Code H05
  PLC1⇨ PLC2, PLC1 sends: "**01 05 0500 FF00 8C F6**"
  PLC2 ⇨PLC1, PLC1 receives: "**01 05 0500 FF00 8C F6**"
  (No data processing on received data)

Program Example 7:

1. Sending the function code K6(H06) through COM2 (RS-485): For writing a word data to a register
   When PLC is connected to VFD-S AC motor drive: M1143 = Off, in ASCII mode
   When PLC is connected to VFD-S AC motor drive: M1143 = On, in RTU mode
2. When in ASCII mode, the user stores the data to be written in the designated register D50 in hex format. The data sent back from AC motor drive are stored in D1070 ~ D1076.
3. When in RTU mode, the user stores the data to be written in the designated register D50 in hex format. The data sent back from AC motor drive are stored in D1070 ~ D1077.
4. When In ASCII mode or RTU mode, PLC will store the data to be sent in D1256 ~ D1295. If necessary, the user can move the data to other general registers by using MOV, DMOV or BMOV instruction. Other instructions of ES/EX/SS do not function on the data in D1256 ~ D1295.

5. After receiving the data sent back from AC motor drive is completed, PLC will auto-check if the received data are incorrect. M1140 will be On if there is an error.
6. If the device address is illegal to a designated communication device, the communication device will respond with an error message and PLC will store the error code in D1130 and M1141 = On. For example, if 8000H is illegal to VFD-S, M1141 will be On and D1130 = 2. See user manual of VFD-S for error codes.
7. After M1140 = On or M1141 = On, PLC will send another correct datum to AC motor drive. If the data sent back from AC motor drive is correct, M1140 and M1141 will be reset.



8. ASCII Mode: When PLC is connected to VFD-S AC motor drive.

   PLC ⇨ VFD-S, PLC sends: **"01 06 0100 1770 71"**

   VFD-S ⇨ PLC, PLC receives: **"01 06 0100 1770 71"**

   Registers for sent data (sending messages)

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1 ADR 0	Address of AC motor drive: ADR (1,0)
D1256 High	'1'	31 H		
D1257 Low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 High	'6'	36 H		
D1258 Low	'0'	30 H	Data Address	
D1258 High	'1'	31 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'1'	31 H	Data content	The content of register D50 (H1770 = K6,000)
D1260 High	'7'	37 H		
D1261 Low	'7'	37 H		
D1261 High	'0'	30 H		
D1262 Low	'7'	37 H	LRC CHK 1 LRC CHK 0	LRC CHK (0,1) is error check
D1262 High	'1'	31 H		

Registers for received data (responding messages)

Register	DATA		Explanation
D1070 Low	'0'	30 H	ADR 1
D1070 High	'1'	31 H	ADR 0
D1071 Low	'0'	30 H	CMD 1
D1071 High	'6'	36 H	CMD 0
D1072 Low	'0'	30 H	Data Address
D1072 High	'1'	31 H	
D1073 Low	'0'	30 H	
D1073 High	'0'	30 H	
D1074 Low	'1'	31 H	Data content
D1074 High	'7'	37 H	
D1075 Low	'7'	37 H	
D1075 High	'0'	30 H	
D1076 Low	'7'	37 H	LRC CHK 1
D1076 High	'1'	31 H	LRC CHK 0

9. RTU Mode: When PLC is connected to VFD-S AC motor drive

PLC ⇨ VFD-S, PLC sends: **"01 06 2000 0012 02 07"**

VFD-S ⇨ PLC, PLC receives: **"01 06 2000 0012 02 07"**

Registers for sent data (sending message)

Register	DATA	Explanation	
D1256 Low	01 H	Address	
D1257 Low	06 H	Function	
D1258 Low	20 H	Data Address	
D1259 Low	00 H		
D1260 Low	00 H	Data content	The content of register D50 (H12)
D1261 Low	12 H		
D1262 Low	02 H	CRC CHK Low	
D1263 Low	07 H	CRC CHK High	

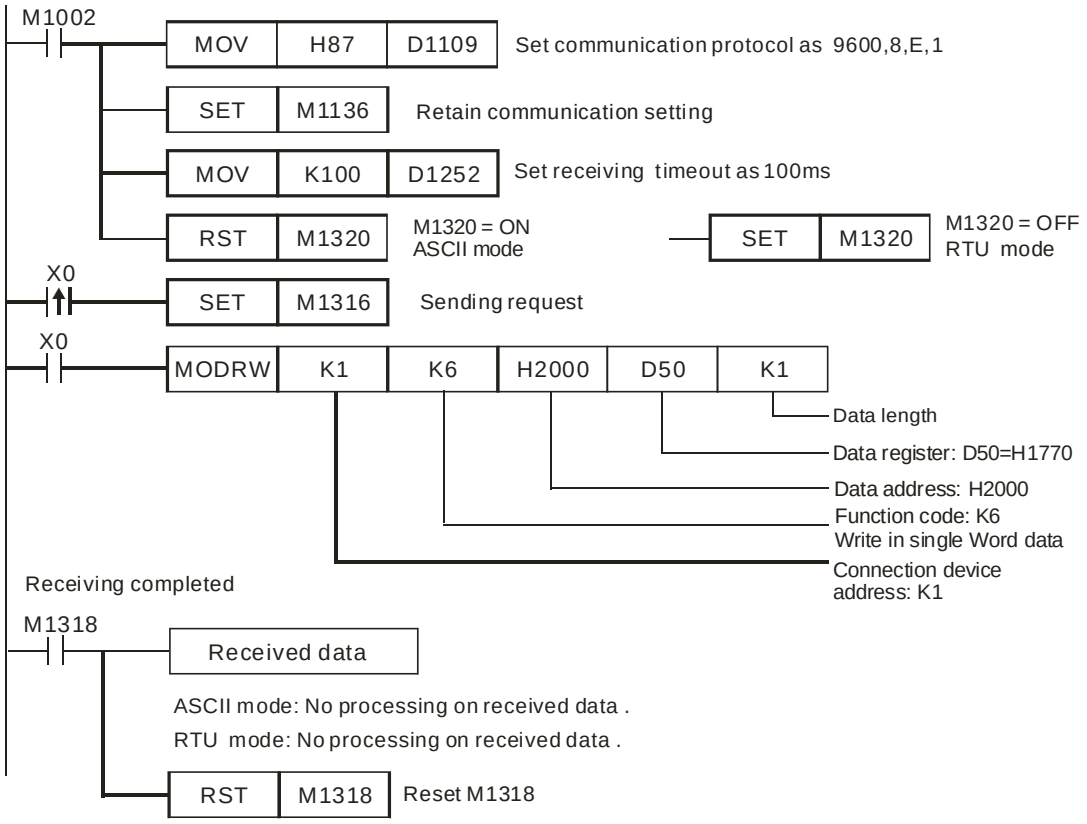
Registers for received data (responding message)

Register	DATA	Explanation
D1070 Low	01 H	Address
D1071 Low	06 H	Function
D1072 Low	20 H	Data Address
D1073 Low	00 H	
D1074 Low	00 H	Data content
D1075 Low	12 H	
D1076 Low	02 H	CRC CHK Low
D1077 Low	07 H	CRC CHK High

Program example 8: COM1 (RS-232) / COM3 (RS-485), Function Code H06

1. Function code K6 (H06): Write in single Word device.
2. Set the value to be written into VFD-B in the register specified by operand **S**.
3. PLC COM1/COM3 will not process the received data.
4. Take the connection between PLC (PLC COM3) and VFD-B for example, the tables below explains the status when PLC COM3 writes in single Word device in VFD-B (M1320 = OFF, ASCII Mode), (M1320 = ON, RTU Mode)
   - If PLC applies COM1 for communication, the below program can be usable by changing:
     1. D1109→D1036: communication protocol

2. M1136→M1138: retain communication setting
3. D1252→D1249: Set value for data receiving timeout
4. M1320→M1139: ASCII/RTU mode selection
5. M1316→M1312: sending request
6. M1318→M1314: receiving completed flag



- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):

  When X0 = ON, MODRW instruction executes the function specified by Function Code H06

  PLC ⇨ VFD-B, PLC sends: **"01 06 2000 1770 52"**

  VFD-B ⇨ PLC, PLC receives: **"01 06 2000 1770 52"**

  (No data processing on received data)

- RTU mode (COM3: M1320 = ON, COM1: M1139 = ON)

  When X0 = ON, MODRW instruction executes the function specified by Function Code H06

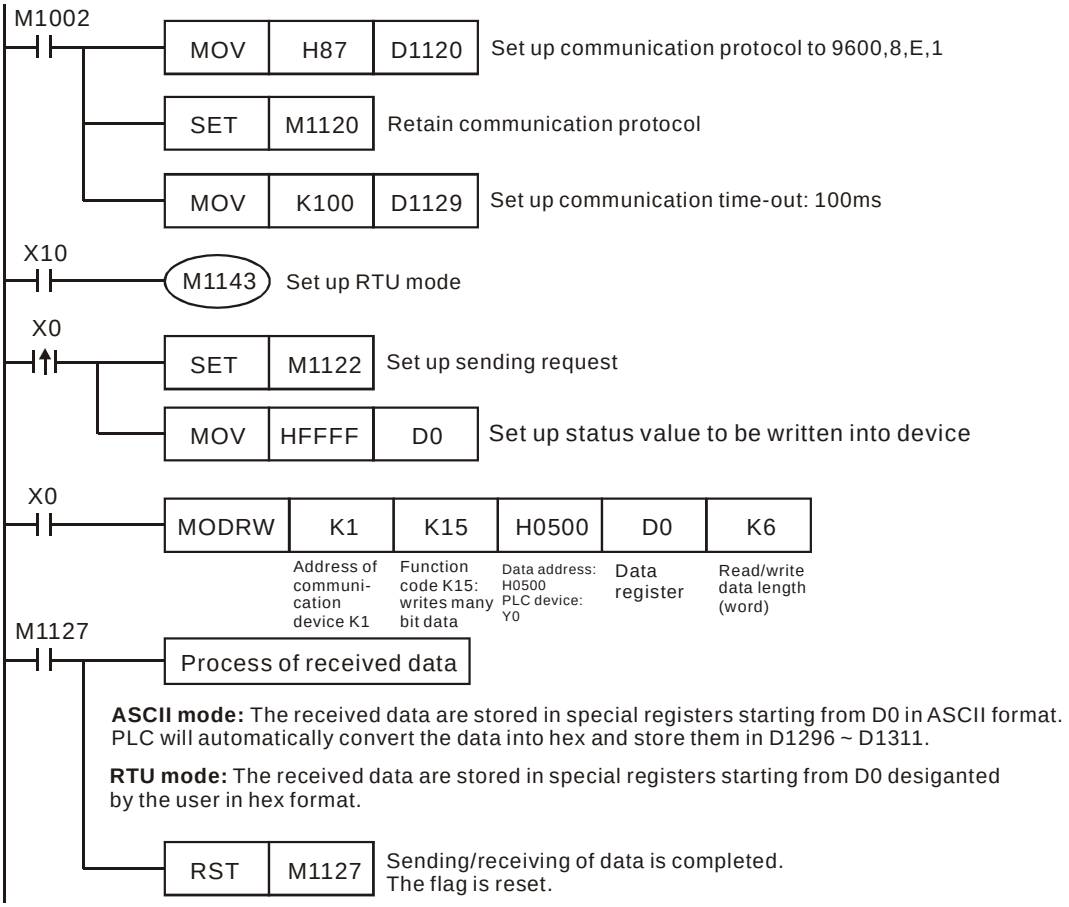
  PLC ⇨ VFD-B, PLC sends: **"01 06 2000 1770 8C 1E"**

  VFD-B → PLC, PLC receives: "**01 06 2000 1770 8C 1E**"

  (No data processing on received data)

Program Example 9:

1. Sending the function code K15 (H0F) through COM2 (RS-485): Write many bit devices. The preset bit status has to be placed in the register designated by the 4th operand of the instruction in b0 ~ b15 order. 1 word is able to contain 16 bit status data.



2. ASCII Mode: When PLC1 is connected to PLC2

   When X0 = On, function code 0F of MODRW instruction will start to be executed.

   PLC1 ⇨ PLC2, PLC sends: " **1 0F 0500 0006 01 3F A5**"

   PLC2 ⇨ PLC1, PLC receives: " **1 0F 0500 0006 E5**"

   Registers for PLC1 sent data (sending messages):

Register	DATA		Explanation	
D1256 low	'0'	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1256 high	'1'	31 H		
D1257 low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 high	'F'	46 H		
D1258 low	'0'	30 H	Data address	
D1258 high	'5'	35 H		
D1259 low	'0'	30 H		
D1259 high	'0'	30 H		
D1260 low	'0'	30 H	Number of data (counted by bits)	
D1260 high	'0'	30 H		
D1261 low	'0'	30H		
D1261 high	'6'	36 H		
D1262 low	'0'	30 H	Byte Count	
D1262 high	'1'	31 H		

Register	DATA		Explanation	
D1263 low	'3'	33 H	Data content 1	Content in D0 register (H3F)
D1263 high	'F'	46 H		
D1264 low	'A'	41 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1264 high	'5'	35 H	LRC CHK 0	

Registers for PLC1 received data (responding messages):

Register	DATA		Explanation
D1070 low	'0'	30 H	ADR 1
D1070 high	'1'	31 H	ADR 0
D1071 low	'0'	31 H	CMD 1
D1071 high	'F'	46 H	CMD 0
D1072 low	'0'	30 H	Data address
D1072 high	'5'	35 H	
D1073 low	'0'	30 H	
D1073 high	'0'	30 H	
D1074 low	'0'	30 H	Number of registers
D1074 high	'0'	30 H	
D1075 low	'0'	30 H	
D1075 high	'6'	36 H	
D1076 low	'E'	45 H	LRC CHK 1
D1076 high	'5'	35 H	LRC CHK 0

3. RTU Mode: When PLC1 is connected to PLC2

When X10 = On, function code 15 of MODRW instruction will start to be executed.

PLC1⇨ PLC2, PLC1 sends: "**01 0F 0500 0006 01 3F**"

PLC2 ⇨ PLC1, PLC1 receives: "**01 0F 0500 0006 D5 05**"

Registers for PLC sent data (sending messages):

Register	DATA	Explanation	
D1256 low	01 H	Address	
D1257 low	0F H	Function	
D1258 low	05 H	Data address	
D1259 low	00 H		
D1260 low	00 H	Data content	Content in D0 register (H3F)
D1261 low	06 H		
D1262 low	01 H	CRC CHK Low	
D1263 low	3F H	CRC CHK High	

Registers for PLC received data (responding messages):

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	0F H	Function
D1072 low	05 H	Data address
D1073 low	00 H	
D1074 low	00 H	Data content
D1075 low	06H	
D1076 low	D5H	CRC CHK Low
D1077 low	05 H	CRC CHK High

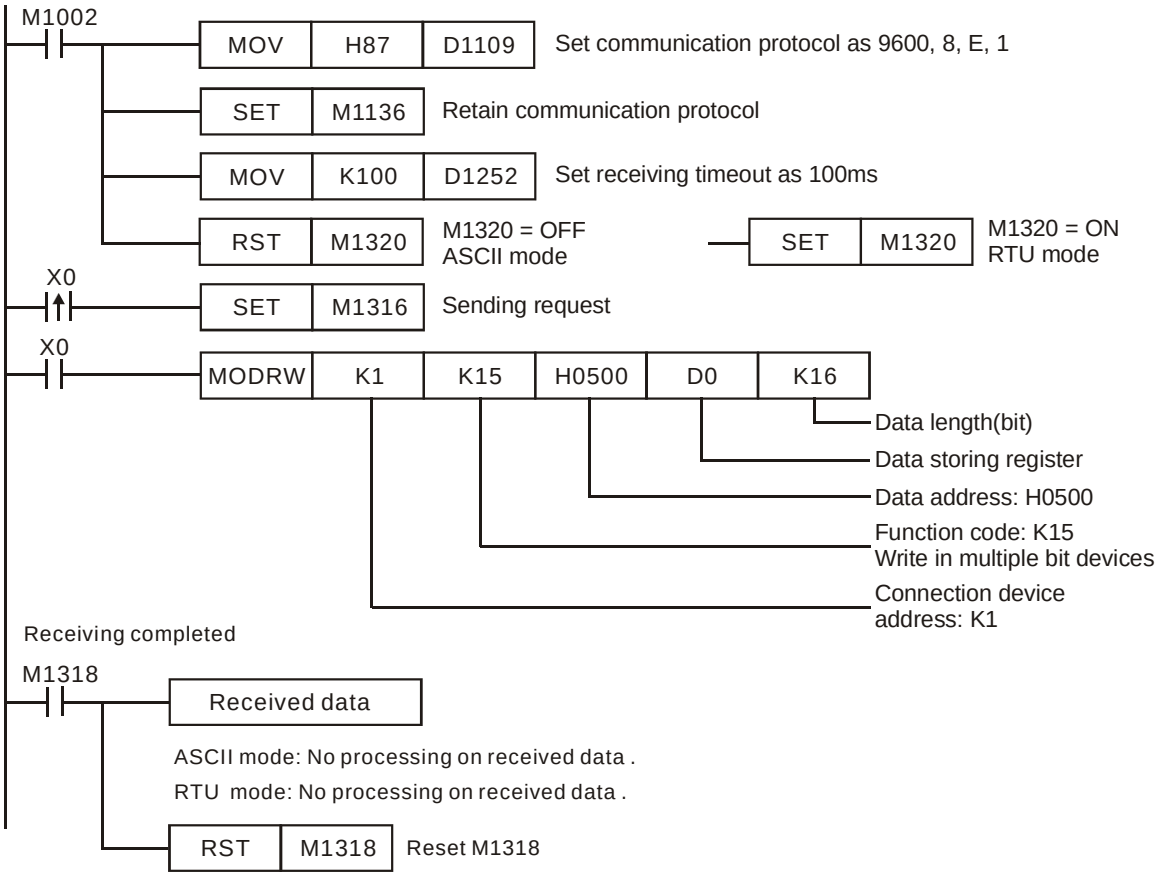
Program example 10: COM1 (RS-232) / COM3 (RS-485), Function Code H0F

1. Function code K15 (H0F): write in multiple bit devices. Up to 64 bits can be written
2. PLC1 connects to PLC2: (M1143 = OFF, ASCII mode), (M1143 = ON, RTU mode)
3. PLC COM1/COM3 will not process the received data.
4. Take the connection between PLC1 (PLC COM3) and PLC2 (PLC COM1) for example, the tables below explain the status when PLC1 force ON/OFF Y0~Y17 of PLC2.

   Set value: K4Y0=1234H

Device	Status	Device	Status	Device	Status	Device	Status
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

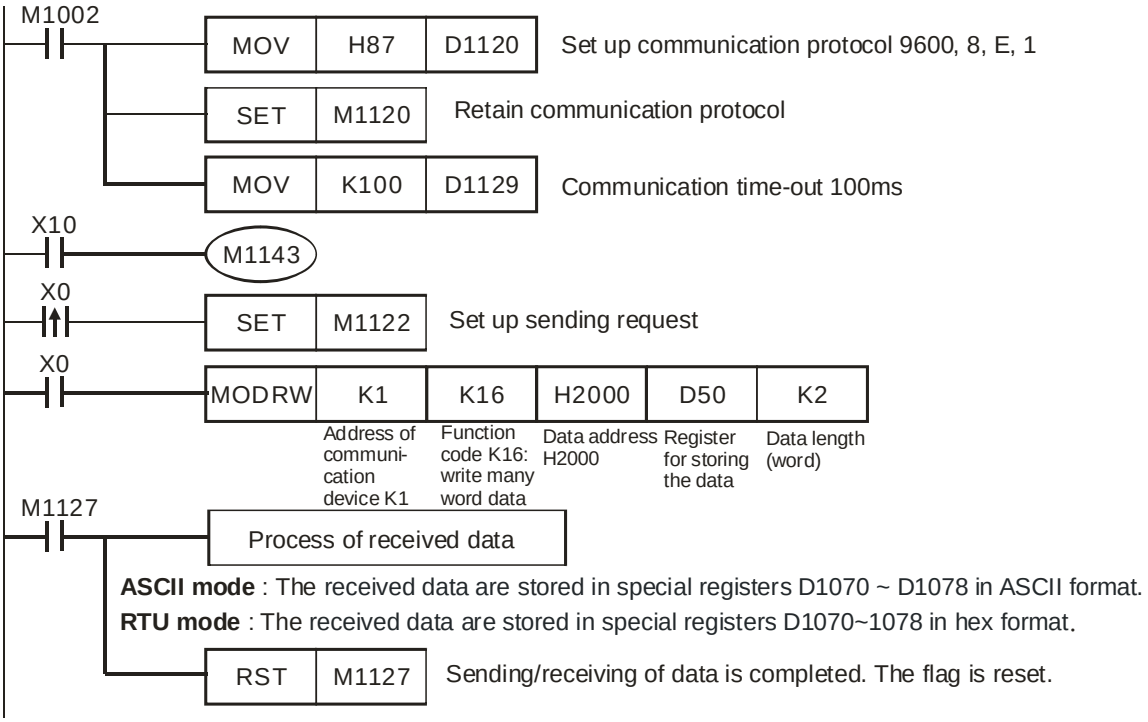
- If PLC applies COM1 for communication, the below program can be usable by changing:
  1. D1109→D1036: communication protocol
  2. M1136→M1138: retain communication setting
  3. D1252→D1249: Set value for data receiving timeout
  4. M1320→M1139: ASCII/RTU mode selection
  5. M1316→M1312: sending request
  6. M1318→M1314: receiving completed flag



- ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):
  When X0 = ON, MODRW executes the function specified by Function Code H0F
  PLC1 ⇨ PLC2, PLC sends: " **01 0F 0500 0010 02 3412 93** "
  PLC2 ⇨ PLC1, PLC receives: " **01 0F 0500 0010 DB** "
  (No data processing on received data)
- RTU mode (COM3: M1320 = ON, COM1: M1139 = ON):
  When X0 = ON, MODRW executes the function specified by Function Code H0F
  PLC1 ⇨ PLC2, PLC1 sends: "**01 0F 0500 0010 02 34 12 21 ED**"
  PLC2 ⇨ PLC1, PLC1 receives: "**01 0F 0500 0010 54 CB**",
  (No data processing on received data)

Program Example 11:

1. Sending the function code K16 (H10) through COM2 (RS-485): For writing many word data into a register.
   When PLC is connected to VFD-S AC motor drive: M1143 = Off, in ASCII mode
   When PLC is connected to VFD-S AC motor drive: M1143 = On, in RTU mode
2. When in ASCII mode, the user stores the data to be written in the designated register D50 in hex format. The data sent back from AC motor drive are stored in D1070 ~ D1076.
3. When in RTU mode, the user stores the data to be written in the designated register D50 in hex format. The data sent back from AC motor drive are stored in D1070 ~ D1077.
4. When In ASCII mode or RTU mode, PLC will store the data to be sent in D1256 ~ D1295. If necessary, the user can move the data to other general registers by using MOV, DMOV or BMOV instruction. Other instructions of ES/EX/SS do not function on the data in D1256 ~ D1295.
5. After receiving the data sent back from AC motor drive is completed, PLC will auto-check if the received data are incorrect. M1140 will be On if there is an error.
6. If the device address is illegal to a designated communication device, the communication device will respond with an error message and PLC will store the error code in D1130 and M1141 = On. For example, if 8000H is illegal to VFD-S, M1141 will be On and D1130 = 2. See user manual of VFD-S for error codes.
7. After M1140 = On or M1141 = On, PLC will send another correct datum to AC motor drive. If the data sent back from AC motor drive is correct, M1140 and M1141 will be reset.



8. ASCII Mode: When PLC is connected to VFD-S AC motor drive.

PLC ⇨ VFD-S, PLC sends: "**01 10 2000 0002 04 0012 1770 30**"

VFD-S ⇨ PLC, PLC receives: "**01 10 2000 0002 CD**"

Registers for sent data (sending messages)

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1 ADR 0	Address of AC motor drive: ADR (1,0)
D1256 High	'1'	31 H		
D1257 Low	'1'	31 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 High	'0'	30 H		
D1258 Low	'2'	32 H	Data Address	
D1258 High	'0'	30 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'0'	30 H	Number of Registers	
D1260 High	'0'	30 H		
D1261 Low	'0'	30 H		
D1261 High	'2'	32 H		
D1262 Low	'0'	30 H	Byte Count	
D1262 High	'4'	34 H		
D1263 Low	'0'	30 H	Data contents 1	The content of register D50 (H12)
D1263 High	'0'	30 H		
D1264 Low	'1'	31 H		
D1264 High	'2'	32 H		
D1265 Low	'1'	31 H	Data contents 2	The content of register D51 (H1770 = K6,000)
D1265 High	'7'	37 H		
D1266 Low	'7'	37 H		
D1266 High	'0'	30 H		
D1267 Low	'3'	33 H	LRC CHK 1 LRC CHK 0	Error checksum: LRC CHK (0,1)
D1267 High	'0'	30 H		

Registers for received data (responding messages)

Register	DATA		Explanation
D1070 Low	'0'	30 H	ADR 1
D1070 High	'1'	31 H	ADR 0
D1071 Low	'1'	31 H	CMD 1
D1071 High	'0'	30 H	CMD 0
D1072 Low	'2'	32 H	Data Address
D1072 High	'0'	30 H	
D1073 Low	'0'	30 H	
D1073 High	'0'	30 H	
D1074 Low	'0'	30 H	Number of Registers
D1074 High	'0'	30 H	
D1075 Low	'0'	30 H	
D1075 High	'2'	32 H	
D1076 Low	'C'	43 H	LRC CHK 1
D1076 High	'D'	44 H	LRC CHK 0

9. RTU Mode: When PLC is connected to VFD-S AC motor drives

PLC ⇨ VFD-S, PLC sends: "**01 10 2000 0002 04 0012 1770 C4 7F**"

VFD-S ⇨ PLC, PLC receives: "**01 10 2000 0002 4A 08**"

Registers for sent data (sending messages)

Register	DATA	Explanation	
D1256 Low	01 H	Address	
D1257 Low	10 H	Function	
D1258 Low	20 H	Data Address	
D1259 Low	00 H		
D1260 Low	00 H	Number of Registers	
D1261 Low	02 H		
D1262 Low	04 H	Byte Count	
D1263 Low	00 H	Data content 1	The content of register D50 (H12)
D1264 Low	12 H		
D1265 Low	17 H	Data content 2	The content of register D51 (H1770 = K6,000)
D1266 Low	70 H		
D1267 Low	C4 H	CRC CHK Low	
D1268 Low	7F H	CRC CHK High	

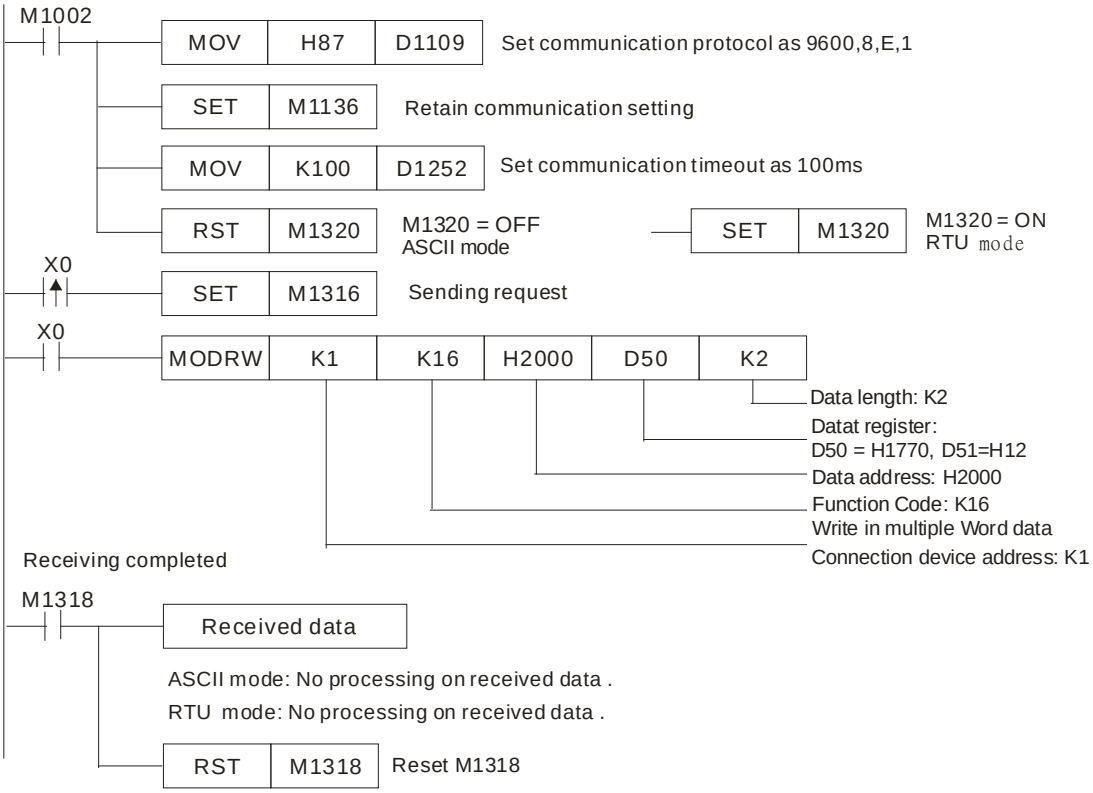
Registers for received data (responding messages)

Register	DATA	Explanation
D1070 Low	01 H	Address
D1071 Low	10 H	Function
D1072 Low	20 H	Data Address
D1073 Low	00 H	
D1074 Low	00 H	Number of Registers
D1075 Low	02 H	
D1076 Low	4A H	CRC CHK Low
D1077 Low	08 H	CRC CHK High

Program example 12: COM1 (RS-232) / COM3 (RS-485), Function Code H10

1. Function code K16 (H10): Write in multiple Word devices. Up to 16 Words can be written. For PLC COM2 ASCII mode, only 8 words can be written.
2. PLC COM1/COM3 will not process the received data
3. Take the connection between PLC COM3 and VFD-B for example, the tables below explain the status when PLC COM3 writes multiple Words in VFD-B. (M1320 = OFF, ASCII mode) (M1320 = ON, RTU mode)

   - If PLC applies COM1 for communication, the below program can be usable by changing:
     1. D1109→D1036: communication protocol
     2. M1136→M1138: retain communication setting
     3. D1252→D1249: Set value for data receiving timeout
     4. M1320→M1139: ASCII/RTU mode selection
     5. M1316→M1312: sending request
     6. M1318→M1314: receiving completed flag



   - ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):

     When X0 = ON, MODRW executes the function specified by Function Code H10

     PLC ⇨VFD-B, PLC sends: **"01 10 2000 0002 04 1770 0012 30"**

     VFD⇨PLC, PLC receives: **"01 10 2000 0002 CD"**

     (No processing on received data)

   - RTU Mode (COM3: M1320=On, COM1: M1139=On):

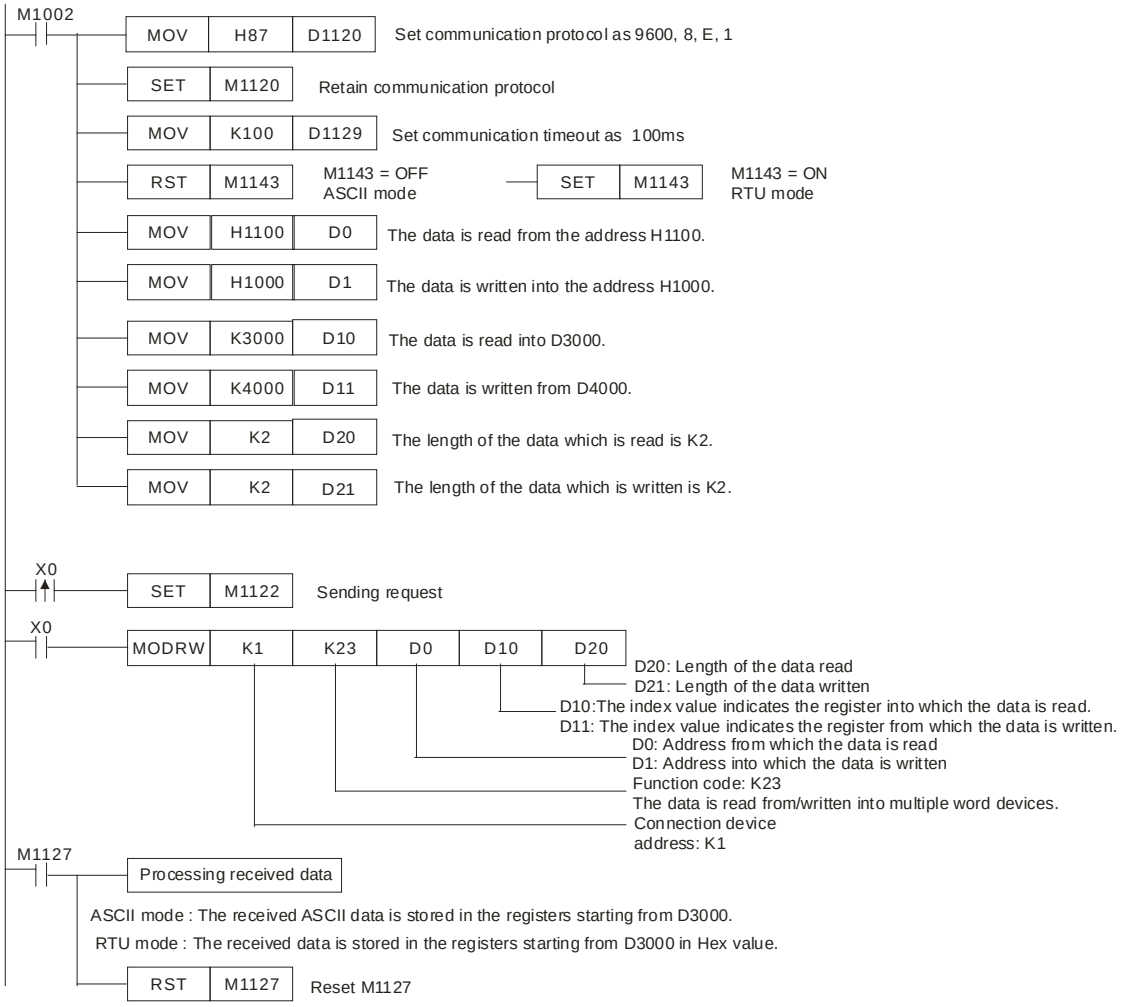
     When X0 = ON, MODRW executes the function specified by Function Code H10

     PLC ⇨VFD-B,PLC sends: "0**1 10 2000 0002 04 1770 0012 EE 0C"**

     VFD-B⇨PLC, PLC receives :" 01 10 2000 0002 4A08" (No processing on received data)

Program Example 13:

1. Sending the function code K23 (H17) through COM2 (RS-485): Read/Write many word devices



2. ASCII Mode: (M1143=OFF)

   When X0=ON, MODRW executes the function specified by the function code H17.

   PLC-A ⇨PLC-B, PLC-A sends: **“01 17 1100 0002 1000 0002 04 1770 0012 06”**

   PLC-B⇨PLC-A, PLC-A receives: **“01 17 04 0100 1766 66”**

   Registers in PLC-A for received data (responding messages)

Register	Data		Description
D3000 Low byte	‘0’	30 H	ADR 1
D3000 High byte	‘1’	31 H	ADR 0
D3001 Low byte	‘1’	31 H	CMD 1
D3001 High byte	‘7’	37 H	CMD 0
D3002 Low byte	‘0’	30 H	Number of data (bytes)
D3002 High byte	‘4’	34 H	
D3003 Low byte	‘0’	30 H	Contents of the address 1100H
D3003 High byte	‘1’	31 H	
D3004 Low byte	‘0’	30 H	
D3004 High byte	‘0’	30 H	
D3005 Low byte	‘1’	31 H	Contents of the address 1101H
D3005 High byte	‘7’	37 H	
D3006 Low byte	‘6’	36H	
D3006 High byte	‘6’	36H	

Register	Data		Description
D3007 Low byte	'6'	36H	LRC CHK 1
D3007 High byte	'6'	36H	LRC CHK 0

3. RTU Mode (M1143=ON)

   When X0=ON, MODRW executes the function specified by the function ode H17.

   PLC-A ⇨PLC-B,PLC-A sends: **"01 17 1100 0002 1000 0002 04 1770 0012 A702"**

   PLC-B⇨PLC-A, PLC-A receives: "**01 17 04 0100 1766 7701**"

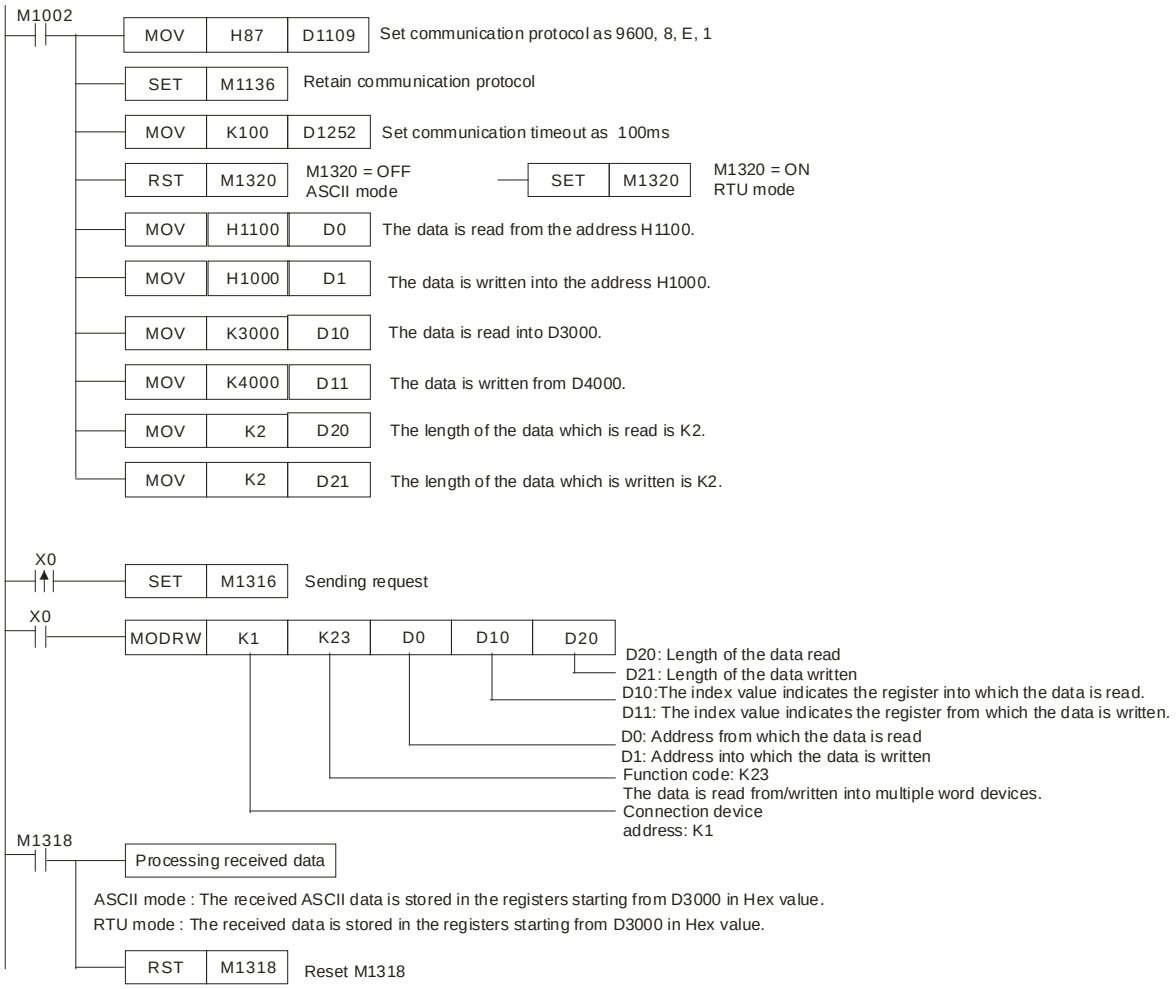
   Registers in PLC-A for received data (responding messages)

   Registers (D0) for PLC received data (responding messages):

Register	Data	Description
D3000 Low byte	01 H	Address
D3001 Low byte	17 H	Function
D3002 Low byte	04 H	Number of data (bytes)
D3003 Low byte	01 H	Contents of the address 1100H
D3004 Low byte	00 H	
D3005 Low byte	17 H	Contents of the address 1101H
D3006 Low byte	66 H	
D3007 Low byte	77 H	CRC CHK Low
D3008 Low byte	01 H	CRC CHK High

Program example 14: COM1 (RS-232)/ COM3 (RS-485), Function Code H17

1. Function code K23 (H17): Data is read from multiple word devices and data is written into multiple word devices. Data can be read from 16 word devices at most, and data can be written into 16 word devices at most.
2. In the ASCII or RTU mode, the data received through COM1/COM3 on the PLC is stored in the registers starting from the register indicated by the index value in **S**+1. Users can use the instruction DTM to transform and move the data.
3. The connection between PLC-A (PLC COM3) and PLC-B:
   - Data is written into multiple word devices in PLC-B from PLC-A. (M1320=OFF, ASCII Mode) (M1320=ON, RTU Mode)
   - If COM1 on PLC-A is connected, the program can be modified as shown below.
     1. D1109→D1036: Communication protocol
     2. M1136→M1138: The communication setting is retained.
     3. D1252→D1249: Communication timeout
     4. M1320→M1139: Choice between the ASCII mode and the RTU mode
     5. M1316→M1312: The sending of the data though the communication instruction is requested.
     6. M1318→M1314: The receiving of the data through the communication instruction is complete.



- ASCII Mode (COM3: M1320=OFF; COM1: M1139=OFF):

  When X0=ON, MODRW executes the function specified by the function ode H17.

  PLC-A ⇨PLC-B, PLC-A sends: **"01 17 1100 0002 1000 0002 04 1770 0012 06"**

  PLC-B⇨PLC-A, PLC-A receives: **"01 17 04 0100 1766 66"**

  Registers in PLC-A for received data (responding messages)

Register	Data	Description
D3000	0100H	PLC-A converts ASCII codes in 1100H and stores the converted data automatically.
D3001	1766H	PLC-A converts ASCII codes in 1101H and stores the converted data automatically.

- RTU Mode (COM3: M1320=ON; COM1: M1139=ON):

  When X0=ON, MODRW executes the function specified by the function code H17.

  PLC-A ⇨PLC-B,PLC-A sends: **"01 17 2100 0002 2000 0002 04 1770 0012 A702"**

  PLC-B⇨PLC-A, PLC-A receives: "**01 17 04 0100 1766 7701**"

  Registers in PLC-A for received data (responding messages)

Register	Data	Description
D3000	0100 H	PLC-A converts data in 1100H and stores the converted data automatically.
D3001	1766 H	PLC-A converts data in 1101H and stores the converted data automatically.

Remarks:

1. The activation condition placed before MODRD, RDST and MODRW instructions cannot use rising-edge or falling-edge contacts; otherwise the data stored in the registers for received data will encounter errors.
2. PLC COM1 ~ COM3: Please refer to API 80 RS for more information about the associated flags (Auxiliary relays) and special registers (Special D) for the communication instruction MODRW.
3. PLC COM2 RS-485: Associated flags (Auxiliary relays) and special registers (Special D) for the communication instruction MODRW

Flags	Function
M1120	For retaining communication setups. After the setup is made, changes in D1120 will be invalid.
M1121	When Off, RS-485 is sending data.
M1122	Sending request
M1123	Receiving is completed
M1124	Waiting for receiving data
M1125	Disable receiving status
M1126	Selecting STX/ETX system
M1127	Sending/receiving data through MODRD / RDST / MODRW instructions is completed.
M1128	Sending data…/receiving data…
M1129	Receiving data time-out
M1130	User/system defined STX/ETX
M1131	On when MODRD / MODWR / MODRW is converting data to hex
M1140	MODRD / MODWR / MODRW data receiving error
M1141	MODRD / MODWR / MODRW parameter error
M1142	VFD-A handy instruction data receiving error
M1143	ASCII/RTU mode selection (used with MODRD/MODWR/MODRW) (Off = ASCII mode; On = RTU mode)
D1070 ~ D1085	When the built-in RS-485 communication instruction is executed and sends out data, the receiving end will respond with a message and the message will be stored in D1070 ~ D1085. The user can check the registers for the messages.
D1120	RS-485 communication protocol
D1121	PLC communication address (saving PLC communication address; latched)
D1122	Remaining words of the sent data
D1123	Remaining words of the received data
D1124	Start text definition (STX)
D1125	Definition of end text 1 (ETX1)
D1126	Definition of end text 2 (ETX2)
D1129	Abnormal communication time-out. Unit: ms
D1130	Records of error codes sent back from MODBUS

Flags	Function
D1256 ~ D1295	When the built-in RS-485 communication instruction MODRW is executed, the sent out data will be stored in D1256 ~ D1295. The user can check whether the instruction is correct by the contents in the registers.
D1296 ~ D1311	PLC will automatically convert the ASCII data stored in the register designated by the user into hex format.

API	Mnemonic	Operands	Function
151	PWD	(S) (D)	Detection of Input Pulse Width

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															PWD: 5 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source device **D**: Destination device for storing the detected result

Explanations: (For SV2 and EH3 V1.40 (and below))

1. Range of **S**: X10 ~ X17
2. **D** must be in the range of D0 to D999., it occupies two consecutive devices.
3. PWD instruction is for detecting the interval between the input signals; the valid frequency range is 1 ~1kHz. If M1169 = Off, the instruction will continuously detect the intervals between the rising edges of the input signals and the falling edges of the input signals (time unit: 100us). If M1169 = On, the instruction will continuously detect the intervals between rising edges of the input signals (time unit: 1us). It cannot designate the same X10 ~ X17 as DCNT and ZRN instructions.
4. **D** occupies two consecutive devices. The longest detection time is 21,474.83647 seconds, about 357.9139 minutes or 5.9652 hours.
5. There is no limitation on the times of using this instruction. However, only one instruction can be executed at a time.

Explanations: (For EH3 V1.60, SV2 V1.60 and later)

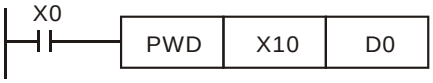
1. **S** must be X10 , X11, X14, X15. Neither of the inputs can be used more than once.
2. The detection result, the number of input pulses, and the number by which an amount is divided are stored in **D**. **D** must be in the range of D0 to D999., it occupies five consecutive devices at most.
3. PWD instruction is for detecting the intervals between the input signals of the frequency of input signals; the valid frequency range is 1 ~1kHz. If M1169 = Off, the instruction will continuously detect the time intervals between the rising edges and the falling edges of the input signals (time unit: 100us). If M1169 = On, the instruction will continuously detect the frequency intervals between rising edges of the input signals (frequency unit: 0.001 Hz). It cannot designate the same input as DCNT and ZRN instructions do.
4. If PWD is executed for the first time, the detection mode of PWD will be set according to the state of M1169. After the instruction is executed, the detection mode can not be changed.
5. If M1169 is ON, M1154 will be the flag for the dectection of the width of the duty-off/duty-on pulse. If M1154 is Off, the width of the duty-off pulse will be detected. If M1154 is On, the width of the duty-on pulse will be detected. If the instruction is used more than once in a program, the same M1154 will be used. The state of M1154 can be changed after the instruciton is executed.
6. If M1169 is On, M1263 will be a averaging mechanism flag. If M1263 is On, the frequencies of input signals will be averaged according to the number set, and the number of input pulses will be stored. If the instruction is

used more than once in a program, the same M1263 will be used. The state of M1263 can be changed after the instruciton is executed.

7. The functions of **D** and **D**+1 depend on the detection mode used. If the width of the duty-off/duty-on pulse is detected, the width of the duty-off/duty-on pulse will be stored in **D** and **D**+1, and the longest detection time will be 21,474.83647 seconds, about 357.9139 minutes or 5.9652 hours. If the frequency of input pulses is detected, it will be stored in **D** and **D**+1. If the frequency of input pules is detected, and the averaging mechanism is enabled, **D**+2, **D**+3, **D**+4 wil be used. The number of input pulses is stored in D+2 and D+3. The number by which an amount is divided is stored in **D**+4, and must be in the range of K1 to K20. If the number by which an amount is divided exceeds the upper limit, or the lower limit, the upper limit or the lower limit will be the setting value.
8. The instruction can be used three times at most in a program.

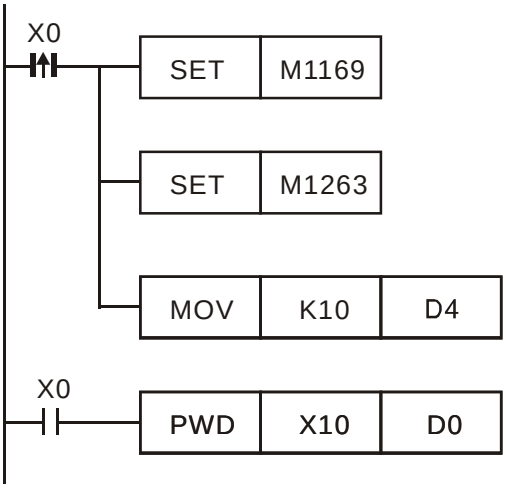
Program Example: (For SV2, and EH3 V1.40 (and below))

When X0 = On, record the time span of X10 = On and store it in D1 and D0.



Program Example: (For EH3/SV2, and EH3 V1.60 (and later))

If X0 is On, ten frequencies of pulses sent to X10 will be averaged, and the result will be stored in D0 and D1. Besides, the number of pulses sent to X10 will be stored in D2 and D3.



API	Mnemonic	Operands	Function
152	RTMU	(D) (n)	Start of the Measurement of Execution Time of I Interruption

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D					*	*							*			RTMU: 5 steps
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

D: Device for storing the measuring time (unit: 1us) **n**: Measurement time base. Parameter range: K10 ~ K500 (time unit: 1us)

Explanations:

1. Range of **D**: K0 ~ K9
2. Range of **n**: K10 ~ K500
3. The designated special D registers (D1156 ~ D1165) can measure up to 10 interruption subroutines. For example, when **D** = K5, the designated D register will be D1161.
4. When RTMU is executed, if the **D** and **n** entered by the user are legal, interruption of the timer will be enabled and the counting starts and the special D designated by **D** is cleared as 0. When RTMD is executed, interruption of the timer is disabled and the calculated time will be assigned to special D designated by RTMD.
5. With API 153 RTMD, RTMU can measure the execution time of “I” interruption service subroutine, which can be reference for dealing with the high-speed response when the user is at the initial stage of developing the program.

API	Mnemonic	Operands	Function
153	RTMD	(D)	End of the Measurement of the Execution Time of I Interruption

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D					*	*							*			RTMD: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

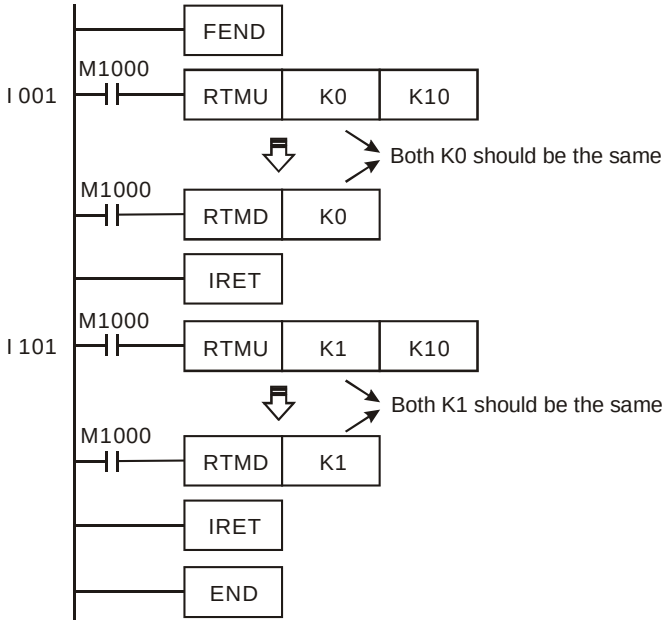
D: Device to store the measuring time (unit: 1us).

Explanations:

1. Range of **D**: K0 ~ K9. The No. of **D** has to be the same as that designated by **D** in API 152 RTMU; otherwise the result of the measurement may be unexpectable.

Program Example:

When X0 goes from Off to On, the program will enter I001 interruption subroutine. RTMU will activate an 8-bit timer (unit: 10us) and RTMD (when D = K0) will shut down the timer and store the time in the timer in special D registers (D1156 ~ D1165, designated by K0 ~ K9).



Remarks:

1. We suggest you remove this instruction after you finish developing your PLC program.
2. Due to the lower priority of the interruption enabled by RTMU, when RTMU is enabled, other high-speed pulse input counting or high-speed pulse output may result in failure to trigger the timer.
3. If you activate RTMU but do not activate RTMD before the end of the interruption, the interruption will not be shut down.
4. RTMU instruction activates 1 timer interruption in PLC. Therefore, if many RTMU or RTMD are executed at the same time, confusion in the timer may occur. Please be aware of the situation.

API	Mnemonic		Operands	Function
154	RAND	P	(S1) (S2) (D)	Random Number

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	RAND, RANDP: 7 steps DRAND: 13 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

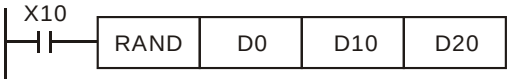
S_1: Lower bound of the random number **S_2**: Upper bound of the random number **D**: The random number produced

Explanations:

1. **$S_1 \leqq S_2$**; K0 $\leqq$ S_1 , S_2 $\leqq$K32,767
2. See the specifications of each model for their range of use.
3. Entering $S_1 > S_2$ will result in operation error. The instruction will not be executed at this time, M1067, M1068 = On and D1067 records the error code 0E1A (hex).

Program Example:

When X10 = On, RAND will produce the random number between the lower bound D0 and upper bound D10 and store the result in D20.



API	Mnemonic	Operands	Function
155	D ABSR	(S) (D1) (D2)	Read the Absolute Position from a Servo Motor

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												DABSR: 13 steps
D_1		*	*	*												
D_2								*	*	*	*	*	*	*		

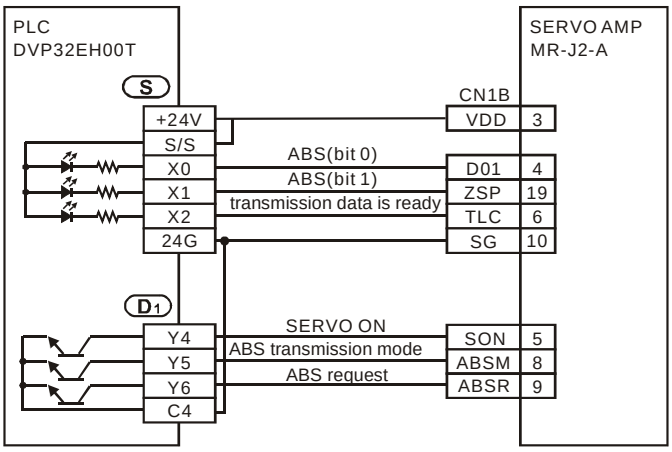
PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Input signal from Servo (occupies 3 consecutive devices) **D_1**: Control signal for controlling Servo (occupies 3 consecutive devices at most) **D_2**: Absolute position data (32-bit) read from Servo (occupies 4 consecutive devices at most)

Explanations: (For SX, EH3 V1.40 (and below), and SV2 V1.20 (and below))

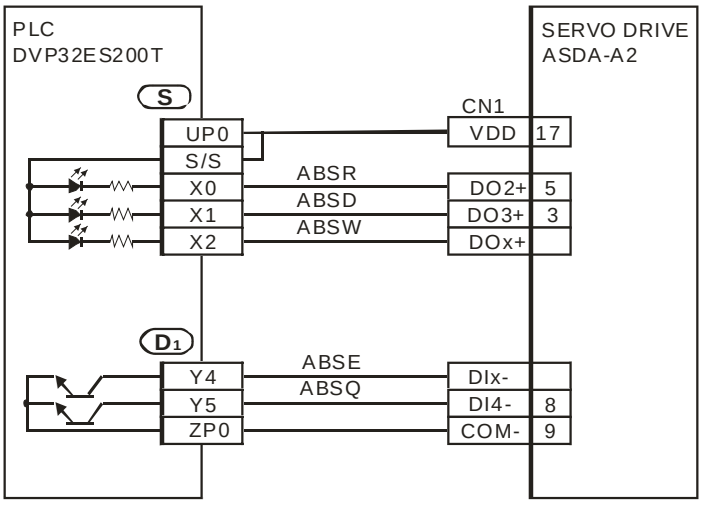
1. Operand S and D_1 of SX series MPU do not support E, F index register modification.
2. See the specifications of each model for their range of use.
3. This instruction can only be used once in the program.
4. Flag: see remarks for more details.
5. This instruction reads the absolute position (ABS) of MITSUBISHI MR-J2 servo drive (with absolute position check function).
6. **S** will occupy 3 consecutive devices, **S**, **S** +1, and **S** +2. **S** and **S** +1 are connected to the absolute position (bit 0, bit 1) on the servo for data transmitting. **S** +2 is connected to Servo for transmitting data ready flag. See the wiring example below for more details.
7. **D_1** will occupy 3 consecutive devices, **D_1**, **D_1** + 1, **D_1** + 2. **D_1** is connected to SERVO On (SON) of Servo. **D_1**+1 is connected to ABS transmisstion mode of Servo and **D_1**+2 is connected to ABS request signal. See the wiring example below for more details.



8. **D_2** will occupy 2 consecutive devices **D_2** and **D_2** + 1. **D_2** is the lower 16 bits and **D_2** + 1 is the higher 16 bits. The absolute position data should be written into the present value registers (D1337, D1336) of CH0 pulse (Y0, Y1) or the present value registers (D1339, D1338) of CH1 pulse (Y2, Y3) in EH series MPU; therefore, we suggest you designate the two corresponding registers. If you designate other devices as the registers, you still have to transmit the data to D1337 and D1336 of CH0 or D1339 and D1338 of CH1. In addition, the absolute position data should be written into the present value registers (D1348, D1349) of CH0 pulse (Y10) or the present value registers (D1350, D1351) of CH1 pulse (Y11) in SC series MPU; therefore, we suggest you designate the two corresponding registers. If you designate other devices as the registers, you still have to transmit the data to D1348 and D1349 of CH0 or D1350 and D1351 of CH1.
9. When DABSR instruction starts to read, after finishing reading the absolute position of SERVO, flag M1029 will be On. The user has to reset the flag.
10. When driving the DABSR command, please specify normally open contact. If the drive contact of DABSR command turns Off when DABSR command read starts, the execution of absolute current value read will be interrupted and result in incorrect data. Please be careful and notice that.

Explanations: (For SX, EH3 V1.40 (and below), and SV2 V1.20 (and below))

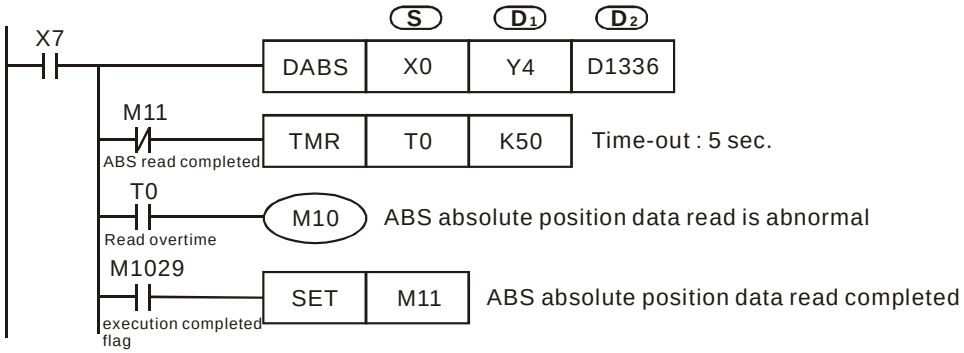
1. This instruction reads the absolute position (ABS) of MITSUBISHI MR-J2 servo drive (with absolute position check function), and the absolute position (ABS) of Delta ASDA-A2 servo drive (whose firmware version is 1.045 sub12 (and above).
2. The state of M1177 determines the servo drive which is used. If M1177 is Off, MITSUBISHI MR-J2 servo drive is used. Please refer to the points above for more information about setting MITSUBISHI MR-J2 servo drive. If M1177 is On, Delta ASDA-A2 servo drive is used. Please refer to the points below for more information about settiing Delta ASDA-A2 servo drive.
3. The input signal from a servo is stored in **S**. **S** occupies 3 consecutive devices. **S**, **S** +1, and **S** +2 are connected to ABSR, ABSD, ABSW on a servo. Please refer to the example below for more information about wiring.
4. **D_1** will occupy 2 consecutive devices, **D_1**, and **D_1** + 1. **D_1** is connected to ABSE on a servo. **D_1**+1 is connected to ABSQ on a servo. Please refer to the example below for more information about wiring.



5. D_2 will occupy 4 consecutive devices D_2, D_2 +1. D_2 +2, and D_2 +3. The absolute acoordinate system status (P0-50) is stored in D_2, the encoder absolute position (multiturn) (P0-51) is stored in D_2 +1. The lower 16 bits of the encoder absolute position (pulse number within singleturn or PUU) (P0-52) is stored in D_2 +2. The higher 16 bits of the encoder absolute position (pulse number within singleturn or PUU) (P0-52) is stored in D_2 +3.
6. After the the reading of the absolute positio of a servo through the instruciton DABSR is complete, M1580 will be On. If an error occurs during the execution of the instruciton, M1581 will be On.
7. When driving the DABSR instruction, please specify normally open contact. If the drive contact of DABSR command turns Off when DABSR command read starts, the execution of absolute current value read will be interrupted and result in incorrect data. Please be careful and notice that.
8. If the input signals are from the high-speed input points X0~X7, it takes 2 seconds for the instruction to be executed. if the input signals are form the input points following X20, it takes 3 seconds for the instruciton to be executed. The time it takes for the instruction to be executed is affected by the scan time.

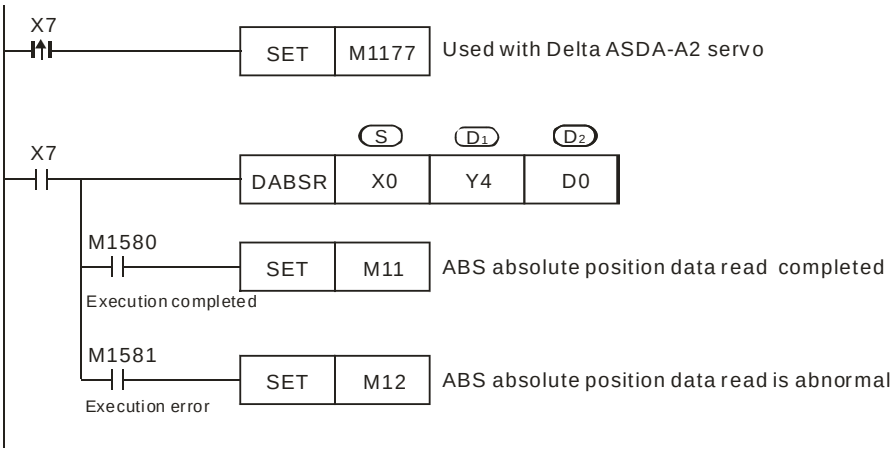
Program Example: (For SX, EH3 V1.40 (and below), SV2 V1.20 (and below))

1. When X7 = On, the 32-bit absolute position data read from Servo will be stored in the present value registers (D1337, D1336) of CH0 pulse in EH MPU. At the same time, the timer T10 is enabled and starts to count for 5 seconds. If the reading of the absolute position is not completed after 5 seconds, M10 will be On, indicating that the reading of absolute position encounters abnormality.
2. When enabling the connection to the system, please synchronize the power input of DVP-PLC EH3/SV2 and SERVO AMP or activate the power of SERVO AMP earlier than DVP-PLC.



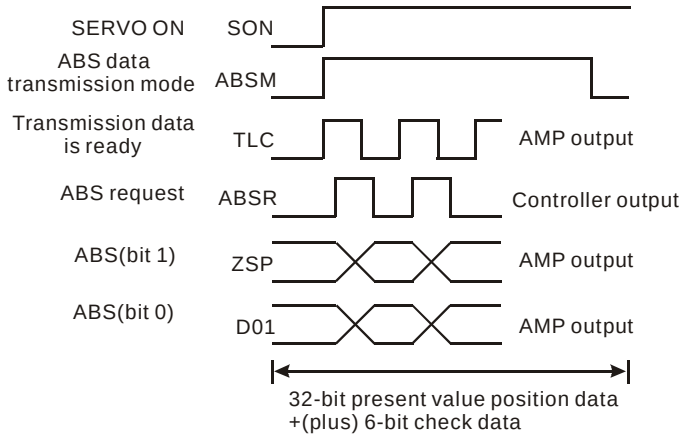
Program Example: (For SA/SX/SC, EH/SV, EH3 V1.40 (and below), SV2 V1.20 (and below))

1. When X7 = On, the absolute position data read from Delta ASDA-A2 servo will be stored in the registers D0~D3. The state of M1580 and the state of M1581 indicates whether the reading of the absolute position is successful.



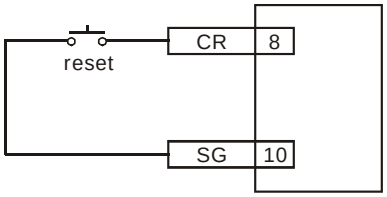
Remarks: (Used with Mitsubishi MR-J2 Servo drive)

1. If the instruction is interrupted when PLC is still reading the absolute position of SERVO, an ALARM message (ALE5) will occur in SERVO.
2. Timing chart of DABSR instruction reading absolute position:
   a) When DABSR instruction starts to execute, it will drive SERBVO On (SON) and ABS transmittion mode for output.
   b) By “transmission is ready” and “ABS request” signals, you can confirm the transmission and reciept of both sides as well as processing the transmission of the 32-bit present position data plus the 6-bit check data.
   c) The data are transmitted by ABS (bit0, bit1).



3. This instruction is applicable to the Servo motor equipped with absolute positioning function, e.g. Mitsubishi MR-J2-A Servo drive.
4. Select one of the following methods for the initial reading of present absolute position.
   a) Complete zero point return by using reset signal function to execute API 156 ZRN instruction.
   b) After using JOG or manual operation to adjust the zero point position, input a reset signal in SERVO AMP. See the figure of external switch below for whether to use DVP-PLC for output. For the wiring of DVP-PLC and Mitsubishi MR-H2-□A, see remarks of API 159 DRVA instruction.

Ex: Mitsubishi MR-J2-□A



5. Flags explanation:

M1010: (For EH3/SV2 series MPU) When M1010 is On, CH0 (Y0, Y1) and CH1 (Y2, Y3) will output pulses while END instruction is being executed. When the output starts, M1010 will automatically turn Off.

M1029: (For EH3/SV2 series MPU) When the first group CH0 (Y0, Y1) pulse output or the execution of other relevant instructions are completed, M1029 will turn On.

M1030: (For EH3/SV2 series MPU) When the second group CH1 (Y2, Y3) pulse output is completed, M1030 will turn on.

M1177: If M1177 is Off, MITSUBISHI MR-J2 servo drive is used with EH3/SV2. If M1177 is On, Delta ASDA-A2 servo drive is used with EH3/SV2.

M1258: (For EH3/SV2 series MPU) When M1258 is On, CH0 (Y0, Y1) will output reverse pulses.

M1259: (For EH3/SV2 series MPU) When M1259 is On, CH1 (Y2, Y3) will output reverse pulses.

M1305: (For EH3/SV2 series MPU) PLSV, DPLSV, DRVI, DDRVI, DRVA, DDRVA instructions for CH0 (Y1, Y2) reverse running.

M1306: (For EH3/SV2 series MPU) PLSV, DPLSV, DRVI, DDRVI, DRVA, DDRVA instructions for CH1 (Y2, Y3) reverse running.

M1334: (For EH series MPU) When M1334 = On, CH0 (Y0, Y1) pulse output will pause.
(For EH3/SV2 series MPU) When M1334 = On, CH0 (Y0, Y1) pulse output will stop.
(For SC series MPU) When M1334 = On, the DDRVI and DDRVA execution criteria will stop and CH0 (Y10) pulse output will stop immediately without deceleration.

M1335: (For EH3/SV2 series MPU) When M1335 = On, CH1 (Y2, Y3) pulse output will stop.

M1520: (For EH3/SV2 series MPU) When M1520 = On, CH2 (Y4, Y5) pulse output will stop.

M1521: (For EH3/SV2 series MPU) When M1521 = On, CH3 (Y6, Y7) pulse output will stop.

M1336: (For EH3/SV2 series MPU) CH0 (Y0, Y1) pulse output indication flag

M1337: (For EH3/SV2 series MPU) CH1 (Y2, Y3) pulse output indication flag

M1346: (For EH3/SV2 series MPU) ZRN instruction for “enabling CLEAR output signal” flag

M1580: If Delta ASDA-A2 servo drive is used, M1580 will be On after the execution of the instruction DABSR is complete.

M1581: If Delta ASDA-A2 servo drive is used, M1581 is On when DABSR is not executed successfully.

6. Special registers:

D1337, D1336: 1. (For EH3/SV2 series MPU) Registers for the first group (Y0, Y1) output pulse present value of position control instructions (API 156 ZRN, API 157 PLSV, API 158 DRVI, API 159 DRVA). The present value increases or decreases according to the corresponding rotation direction. D1337 is for high word; D1336 is for low word.

2. (For EH3/SV2 series MPU) Registers for storing the current number of output pulses of the first group (Y0, Y1) output of pulse output instructions (API 57 PLSY, API 59 PLSR). D1337 is for high word; D1336 is for low word.

D1338, D1339:
1. (For EH3/SV2 series MPU) Registers for the second group (Y2, Y3) output pulse present value of position control instructions (API 156 ZRN, API 157 PLSV, API 158 DRVI, API 159 DRVA). The present value increases or decreases according to the corresponding rotation direction. D1339 is for high word; D1338 is for low word.
2. (For EH3/SV2 series MPU) Registers for storing the current number of output pulses of the second group (Y2, Y3) output of pulse output instructions (API 57 PLSY, API 59 PLSR). D1339 is for high word; D1338 is for low word.

D1340 (D1352): For setting up the frequencies of the first acceleration segment and the last deceleration segment when the position control instructions (API 156 ZRN, API 158 DRVI, API 159 DRVA) are executing CH0 (CH1) outputs.

Range of setting:

For EH3/SV2 series MPU, the speed has to be higher than 10Hz. Frequency lower than 10Hz or higher than maximum output frequency will be output by 10Hz. The default setting in EH/EH2/SV series MPU is 200Hz. For SC series MPU, the speed has to be 100 ~ 100kHz. Frequency lower than 100Hz will be output by 100Hz and frequency higher than 100kHz will be output by 100kHz. The default setting in SC series MPU is 100Hz.

Note: During the control of the stepping motor, please consider the resonance and the limitation on the start frequency when you set up the speed.

D1341, D1342: (For EH3/SV2 series MPU) For setting up the maximum speed when the position control instructions (API 156 ZRN, API 158 DRVI, API 159 DRVA) are being executed. D1342 is for high word; D1341 is for low word.

Range of setting: 200kHz fixed.

D1343 (D1353): For setting up the time of the first acceleration segment and the last deceleration segment when the position control instructions (API 156 ZRN, API 158 DRVI, API 159 DRVA) are executing CH0 (CH1) outputs.

Range of setting:

For EH3/SV2 series MPU, the acceleration/deceleration time has to be 1 ~ 10,000ms. The time longer than 10,000ms will be output by the default 100ms. Note: During the control of the stepping motor, please consider the resonance and the limitation on the start frequency when you set up the speed.

API	Mnemonic	Operands	Function
156	D ZRN	(S1) (S2) (S3) (D)	Zero Return

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	ZRN: 9 steps DZRN: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
S_3	*	*	*	*												
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Zero return speed **S_2**: Creep speed **S_3**: Near p oint signal (DOG) **D**: Pulse output device (please use transistor output module)

Explanations:

1. When **S_1** and **S_2** are used in device F, only 16-bit instruction is applicable.
2. **S_1** and **S_2** of SC series MPU only support device K, H and D. **S_3** of SC series MPU only supports device X10 and X11.
3. Flag: see remarks of API 155 ABSR and API 158 DDRVI for more details.
4. **S_1** is the starting speed of zero return operation. For EH3/SV2 series MPU, the 16-bit instruction can designate the range of the speed, which is 10 ~ 32,767Hz and the range designated by the 32-bit instruction is 10 ~ 200,000Hz. If the designated speed is slower than 10Hz, the zero return will operate at 10Hz and when the designated speed is faster than 200kHz, the zero return will operate at 200kHz. For EC3-8K series MPU, the 32-bit instruction can designate the range of speed , which is 1 ~ 10,000Hz. If the designated speed is slower than 1Hz, the zero return will operate at 1Hz, and when the designated speed is faster than 100kHz, the zero return will operate at 10kHz.
5. **S_2** is the designated low speed after the near point signal (DOG) is On. EH3/SV2 series MPU can designate the range of **S_2**, which is 10 ~ 32,767Hz.
6. **S_3** is the designated near point signal (DOG) input (input from A contact). In EH3/SV2 series MPU, if devices other than the external output device (X10 ~ X17), e.g. X, Y, M, S are designated, they will be affected by the scan period, resulting in dispersion of the zero point. In addition, please note that the MPU cannot designate the same input points X10 ~ X17 as those designated by DCNT and PWD instructions.
7. EH3/SV2 series MPU has four groups of A/B phase pulse output, CH0 (Y0, Y1), CH1 (Y2, Y3), CH2 (Y4, Y5) and CH3 (Y6, Y7). 24SV2 series MPU does NOT support CH4 (Y10, Y11) and CH5 (Y12, Y13). See remarks for the setup methods.
8. Zero return output device in different models:

Model	EC3-8K MPU	EH3/SV2 MPU
Zero return output	Y0, Y2	Y0, Y2, Y4, Y6

9. EH3/SV2: The instruction DZRN can be used to detect the limit switch, nake the pulsed output stop at the positive position, search for the Z phase, and output a certain number of displacement. Therefore, when the instrcution is writtem, the input number of the DOG point should be consistent with the description in the table below. 24SV2 series MPU: DOG points of Y4 and Y6 are not available; for this function, you need to use input points instead. The functions of left limit and finding Z phase are not available.

Output point number (**D**)		Y0	Y2	Y4	Y6
Corresponding output point number		Y1	Y3	Y5	Y7
DOG point number (S_3)		X2	X6	X12	X16
Disabling the left limit		M1570=On	M1571=On	M1572=On	M1573=On
Left limit input point		X3	X7	X13	X17
The left limit switch is triggerred by a rising-edge signal or a falling-edge signal. (OFF: Rising-edge signal; ON: Falling-edge signal) (EH3 V1.40/SV2 V1.20 and above)		M1584	M1585	M1586	M1587
Stopping at the right side of DOG		M1574=On	M1575=On	M1576=On	M1577=On
Searching for the Z phase (M1578=Off)	Z phase number	X1	X5	X11	X15
	The number of times the Z phase is searched for is stored in D1312.	Positive value: Searching for the Z phase in the forward direction Negative value: Searching for the Z phase in the backward direction			
Number of displacement (M1578=On)	The number of displacement is stored in D1312.	Positive value: The pulse output is in the forward direction Negative value: The pulse output is in the backward direction			
Clearing the output (M1346=On)		Y10	Y11	Y12	Y13

10. EC3-8K:Use M1307 to set the left limit function of the channels CH0 (Y0, Y1) and CH1 (Y2, Y3) to ON or OFF before the instruciton execution. Use M1305 and M1306 to set the signal direction of Y1 and Y3 before the instruction execution.

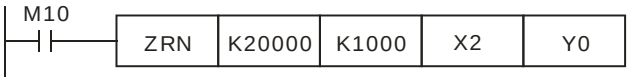
Channel / Input point	CH0(Y0,Y1)	CH1(Y2,Y3)
DOG point number	X4	X6
Enalbe the left limit; M1307=On	X5	X7
The left limit switch is triggerred by a rising-edge signal or a falling-edge signal. (OFF: Rising-edge signal; ON: Falling-edge signal)	M1584	M1585
Set the signal direction	M1305	M1306
Zero return point selection	M1106	M1107
M1346=On Clearing the output	Y4	Y5

D1312 != 0	M1308 = Off (Times to search for the Z phase)	
	X2	X3
D1312 != 0	M1308 = On (pulse number to output)	

11. When executing API 158 DRVI (releative positioning) or API 159 DRVA (absolute positioning), PLC will automatically store the increasing or decreasing forward/reverse pulses in the present D registers.In this way, you can keep track of the position of the machine at any time. However, errors may occur on the first time use or after a long period of operation time, you can enter the zero point position of the machine when executing zero return.

Program Example:

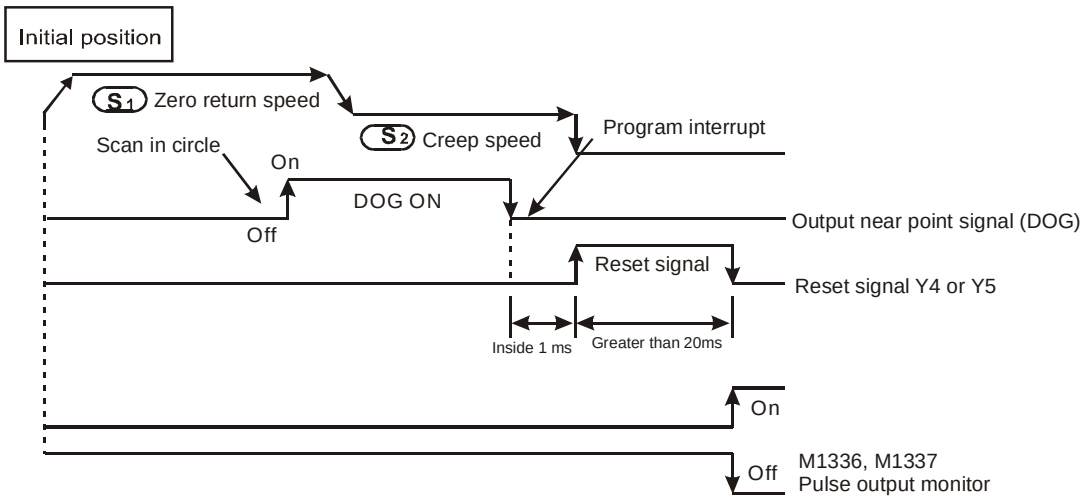
When M10= On, Y0 output pulses start to operate zero return at the frequency of 20kHz. When the zero return meets DOG X2 = On, Y0 output pulses will start to operate by creep speed 1kHz until X2 is Off.



Remarks:

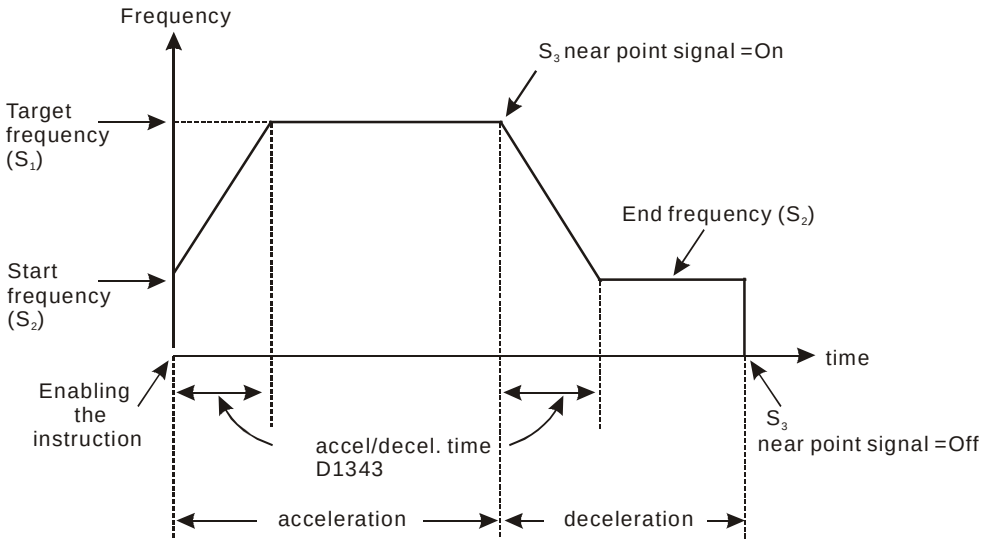
a) Timing chart of the reset signal output for EC3-8K/EH3/SV2 series MPU. When the reset signal flag M1346 = On, after zero return is completed, the PLC can send the reset signal to the servo drive and the signal will last for approximately 20ms. After 20ms, the reset signal will return to Off again.

b) Refer to the description above to see the output devices for reset signals.



Note: The designated devices, X, Y, M, and S, other than the external input devices X10 ~ X17 will be affected by the scan period, 2 times of the scan period at worst.

1. The zero return operation: (refer to PLSY or DRVI instruciton for relevant special D)
   a) When ZRN instruction is executed, set the frequency of the first acceleration segment as the start frequency. The acceleration time of special D is used for reference.S1 will start to move when the acceleration reaches the zero return speed.
   b) When the DOG signal goes from Off to On, the zero return speed will decelerate to **S_2** in the acceleration/deceleration time.
   c) When the DOG signal goes from On to Off, the pulse output will immediately stop, 0 will be written in the present value.
   d) When the DOG signal goes from On to Off and the reset signal flag M1346 = On, reset the current position of the special and output a reset signal simultaneously.
   e) When the pulse output is completed, the completion flag is ON and the in operation flag is OFF.
   f) When the instruciton is enabled, PLC searches for the DOG signal in the negative direction.
   g) When the instruciton is enabled in an EH3/SV2 series MPU, and the negative limit function is selected, the PLC searches for the DOG signal in the negative direction. If the PLC meets the negtavie limit during the process, it will search for the DOG signal in the positive direction. When the instruciton is enabled in an EH3/SV2 series MPU, but the negative limit function is not enabled, the PLC automatically refers to the present position of the axis, and searches for the DOG signal in the direction of 0. For example, if the poresent position of CH0 (D1336, D1137) is greater than or equal to 0, CH0 will search for the DOG signal in the negative direction. If the poresent position of CH0 (D1336, D1137) is less than 0, CH0 will search for the DOG singal in the positive direction.



   h) ZRN (DZRN) instruction is applicable to servo motor with absolute positioning function, e.g. Mitsubishi MR-J2-A servo drive. Even when the power is switched off, the current position can still be recorded. In addition, the current position of servo drive can be read by API 155 DABSR of EH3/SV2 series MPU; therefore only one zero return operation is required and no zero return has to be done after the power is switched off.
   i) Many ZRN instructions can be compiled in the program but only one instruction can be executed when the

PLC program is being executed. For example, provided there is already an instruction enabling Y0 output, other instructions enabling also Y0 output will not be executed. The principle of the instruction execution is “first come, first executed”.

j) After the instruction is executed, all parameters cannot be modified unless the execution of the instruction stops.

k) When the execution of the stops, all outputs will stop immediately no matter what type of the output it is.

API	Mnemonic		Operands	Function
157	D	PLSV	S D1 D2	Adjustable Speed Pulse Output

OP \ Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	PLSV: 7 steps DPLSV: 13 steps
D_1		*														
D_2		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Pulse output frequency **D_1**: Pulse output device (please use transistor output module) **D_2**: Output device for the signal of rotation direction

Explanations:

1. See remarks for the setting range of **S**, **D_1** and **D_2**.
2. Flag: see remarks of API 155 ABSR and API 158 DDRVI for more details.
3. **S** is the designated pulse output frequency. The 16-bit instruction can designate its range 0 ~ +32,767Hz, 0 ~ -32,768Hz. EH3/SV2: the ranges designated by 32-bit instruction are 0 ~ +200,000Hz and 0 ~ -200,000Hz, EC3-8K: : the ranges designated by 32-bit instruction are -200,000 ~ 200,000Hz and -20,000 ~ 20,000Hz. "+/-" signs indicate forward/backward directions. During the pulse output, the frequency can be changed, but not the frequencies of different directions.
4. **D_1** is the pulse output device. EC3-8K series MPU can designate Y0 and Y2 and EH3/SV2 series MPU can designate Y0, Y2, Y4 and Y6. 24SV2 series MPU can designate Y0 and Y2 and EH3/SV2 series MPU can designate Y0, Y2, Y4, Y6, Y10 and Y12.
5. The operation of **D_2** corresponds to the "+" or "-" of **S**. When **S** is "+", **D_2** will be On; when **S** is "-", **D_2** will be Off.
6. EH3/SV2 series MPU: M1592, M1593, M1594, M1595 are the signal reversed direction flag for CH0, CH1, CH2 and CH3. 24SV2 series MPU: M1596 and M1597 are the signal reversed direction flag for CH4 and CH5. When the signal reversed direction flag is ON, **S_1** is a negative value (-). When **D_2** is ON and **S_1** is a positoive value (+), **D_2** is OFF. If 24SV2 uses Y device for **D_1** and **D_2**, you can only use Y10/Y11 and Y12/Y13 to work with **D_1** and **D_2**.
7. PLSV instruction does not have settings for acceleration and deceleration. Please use API 67 RAMP for the acceleration and deceleration of pulse output frequency.
8. During the pulse output executed by PLSV instruction, the drive contact turning Off will result in the immediate stop of the output without going through a deceleration.
9. EH3/SV2 series MPU: When the absolute value of the input frequency during the execution of DPLSV is bigger than 200kHz, the output will operate at 200kHz. EC3-8K series MPU: When the absolute value of the input frequency during the execution of DPLSV is smaller than 10kHz, the output will operate at 10kHz.
10. For EH3/SV2 series MPU, D1222, D1223, D1383 and D1384 are the time differences sent between the direction setup signal and pulse output points of CH0, CH1, CH2 and CH3. Use this function when setting the output direction before pulse output.

11. For EH3/SV2 series MPU, M1305, M1306, M1532 and M1533 are the flags of the direction signals of CH0, CH1, CH2 and CH3. When S is “+”, the output will operate towards a forward direction and the flag will go Off. When S is “-“, the output will operate towards a backward direction and the flag will go On.

Program Example:

When M10 = On, Y0 will output pulses at 20kHz. Y5 = On indicates forward pulses.

API	Mnemonic		Operands	Function
158	D	DRVI	(S1) (S2) (D1) (D2)	Drive to Increment

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	DRVI: 9 steps DDRVI: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D_1		*														
D_2		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Number of output pulses (relative designation) **S_2**: Pulse output frequency **D_1**: Pulse output device (please use transistor output module) **D_2**: Output device for the signal of rotation direction

Explanations:

1. See remarks for the setting range of **S_1**, **S_2**, **D_1** and **D_2**.
2. **S_1** and **S_2** of SC series MPU only support device K, H and D.
3. Flag: see remarks for more details.
4. **S_1** is the number of output pulses (relative designation). For EH3/SV2 series MPU, the 16-bit instruction can designate the range -32,768 ~ +32,767. The range designated by 32-bit instruction is -2,147,483,648 ~ +2,147,483,647. If the value in **S_1** is 0, that means no output and no action.
5. **S_2** is the designated pulse output frequency. For EH3/SV2 series MPU, the 16-bit instruction can designate its range 10 ~ 32,767Hz. The range designated by 32-bit instruction is 10 ~ 200,000Hz. For EC3-8K series MPU, the 32-bit instruction can designate the range 0 ~ 20,000Hz.
6. 24SV2 series MPU: CH4 is pulse (Y10) + direction (Y11). CH5 is pulse (Y12) + direction (Y13).
7. Pulse output device **D_1** in different models

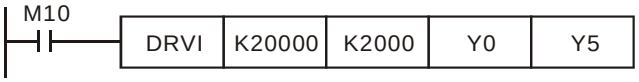
Model	EC3-8K MPU	EH3/28SV2 MPU	24SV2 MPU
Pulse output end	Y0, Y2	Y0, Y2, Y4, Y6	Y0, Y2, Y4, Y6, Y10, Y12

8. EH3/SV2 series MPU: The operation of **D_2** corresponds to the "+" or "-" of **S_1**. When **S_1** is "-", **D_2** will be Off; when **S_1** is "+", **D_2** will be On. **D_2** will not be Off immediately after the pulse output is over; it will be Off only when the drive contact of the instruction turns Off.
9. EC3-8K series MPU: The operation of **D_2** corresponds to the "+" or "-" of **S_1**. When **S_1** is "+", **D_2** will be On; when **S_1** is "+", **D_2** will be Off. **D_2** will not be Off immediately after the pulse output is over; it will be Off only when the drive contact of the instruction turns Off.
10. When DRVI instruction is executing pulse output, you cannot change the content of all operands. The changes will be valid next time when DRVI instruction is enabled.
11. EH3/SV2 series MPU: When the absolute value of the input frequency during the execution of DDRVI is bigger than 200kHz, the output will operate at 200kHz. EC3-8K series MPU: When the absolute value of the input frequency during the execution of DDRVI is smaller than 10kHz, the output will operate at 10kHz.
12. For EH3/SV2 series MPU, M1305, M1306, M1532, M1533 is the direction signal of CH0, CH1, CH2, and CH3 respectively. When **S_1** is a positive number, the output will be operated in a forward direction and M1305,

M1306, M1532, M1533 will be Off. When S_1 is a negative number, the output will be operated in a backward direction and M1305, M1306, M1532, M1533 will be On.

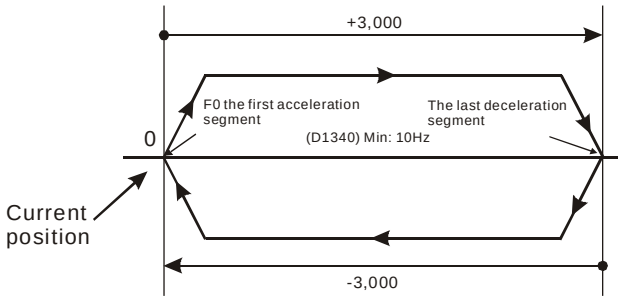
Program Example:

When M10= On, Y0 will output 20,000 pulses (relative designation) at 2kHz. Y5 = On indicates the pulses are executed in forward direction.

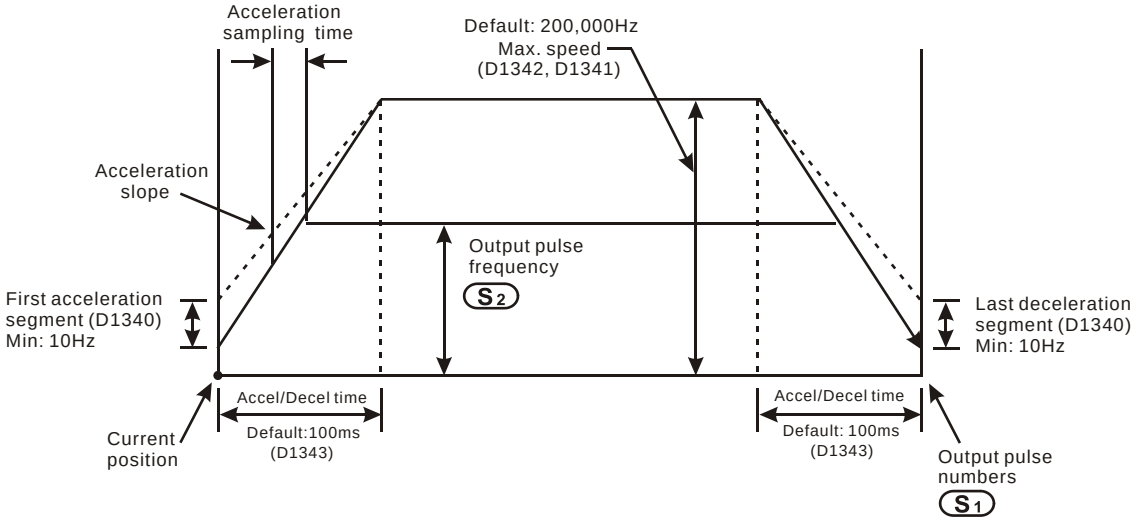


Remarks:

1. Explanations on EH3/SV2 series MPU:
   a) Relative position control: Designating the traveling distance starting from the current position by "+/ -" signs; also known as a relative driving method.



   b) Settings of relative positioning and the acceleration/deceleration speed:



   c) Many DRVI instructions can be compiled synchronously in the program, but only one instruction can be activated whenever the PLC executes the program. For example, if Y0 output has already been activated by an instruction, other instructions that are also used to activate Y0 output will not be excecuted. Therefore, the principle of the instruction activation sequence is "first activated, first executed".
   d) When Y0is activated by DDRVI instruction, the output function of Y0 will be disabled until DDRVI is OFF. The same rule applies to Y1.

e) Once the instruction is activated, all other parameters cannot be modified until the instruction is disabled.

f) When the instruction is disabled but the output has not yet completed:

M1334 = On indicates that Y0 will stop output immediately.

M1334 = Off indicates that Y0 will decelerate according to the deceleration time till it reaches end frequency and stop the pulse output.

M1335 corresponds to Y11 output and applies the same rule.

2. Flags for EC3-8K series MPU:

M1029:	On, after Y0 pulse output is completed.
M1102:	On, after Y2 pulse output is completed.
M1078:	Y0 pulse output pauses.
M1104:	Y2 pulse output pauses.
M1347:	Auto reset after Y0 output is completed.
M1524:	Auto reset after Y2 output is completed.
D1030:	Low word of the current number of output pulses from Y0 (retainable)
D1031:	High word of the current number of output pulses from Y0
D1336:	Low word of the current number of output pulses from Y2 (retainable)
D1337:	High word of the current number of output pulses from Y2

3. Flags for EH3/SV2 series MPU:

M1010:	When M1010 = On, CH0, CH1, CH2 and CH3 will output pulses when END instruction is being executed. M1010 will be Off automatically when the output starts.
M1029:	M1029 = On after CH0 pulse output is completed.
M1030:	M1030 = On after CH1 pulse output is completed.
M1036:	M1036 = On after CH2 pulse output is completed.
M1037:	M1037 = On after CH3 pulse output is completed.
M1078:	Stop pulse output on CH4. (for 24SV2)
M1104:	Stop pulse output on CH5. (for 24SV2)
M1119	The instruction DDRVI/DDRVA is enabled when M1119 is On.
M1257	The acceleration/deceleration slope of the high-speed pulse output is an S curve.
M1305:	Direction signal of CH0.
M1306:	Direction signal of CH1.
M1308	Off->On: The high-speed pulse output CH0 (Y0, Y1) pauses immediately. On->Off: Continuing to output the pulses which have not been output
M1309	Off->On: The high-speed pulse output CH1 (Y2, Y3) pauses immediately. On->Off: Continuing to output the pulses which have not been output
M1310	Off->On: The high-speed pulse output CH2 (Y4, Y5) pauses immediately. On->Off: Continuing to output the pulses which have not been output
M1311	Off->On: The high-speed pulse output CH3 (Y6, Y7) pauses immediately. On->Off: Continuing to output the pulses which have not been output

M1326: On after CH4 pulse output is completed. (for 24SV2)
M13327: On after CH5 pulse output is completed. (for 24SV2)
M1335: CH1 pulse output stops.
M1336: “CH0 sends out pulses” indication.
M1337: “CH1 sends out pulses” indication.
M1347: Reset flag for CH0 pulse output.
M1348: Reset flag for CH1 pulse output.
M1520: CH2 pulse output stops.
M1521: CH3 pulse output stops.
M1522: “CH2 sends out pulses” indication.
M1523: “CH3 sends out pulses” indication.
M1524: Reset flag for CH2 pulse output.
M1525: Reset flag for CH3 pulse output.
M1534: Designated deceleration time of CH0 (should be used with D1348).
M1535: Designated deceleration time of CH1 (should be used with D1349).
M1536: Designated deceleration time of CH2 (should be used with D1350).
M1537: Designated deceleration time of CH3 (should be used with D1351).
M1568: Designated deceleration time of CH4 (should be used with D1196). (for 24SV2)
M1569: Designated deceleration time of CH5 (should be used with D1197). (for 24SV2)
M1592: Forced to reverse signal direction flag of CH0 (for 24SV2)
M1593: Forced to reverse signal direction flag of CH1 (for 24SV2)
M1594: Forced to reverse signal direction flag of CH2 (for 24SV2)
M1595: Forced to reverse signal direction flag of CH3 (for 24SV2)
M1596: Forced to reverse signal direction flag of CH4 (for 24SV2)
M1597: Forced to reverse signal direction flag of CH5 (for 24SV2)
M1614: CH4 pulse output has been sent. (for 24SV2 series)
M1615: CH5 pulse output has been sent. (for 24SV2 series)
D1022: The first starting frequency and the later ending frequency of CH4
D1023: The first starting frequency and the later ending frequency of CH5
D1030: Low word of the current number of output pulses from CH4 (for 24SV2 series)
D1031: High word of the current number of output pulses from CH4 (for 24SV2 series)
D1032: Low word of the current number of output pulses from CH5 (for 24SV2 series)
D1033: High word of the current number of output pulses from CH5 (for 24SV2 series)
D1127: The number of pulses in the acceleration section in the position instruction (low word)
D1128: The number of pulses in the acceleration section in the position instruction (high word)
D1133: The number of pulses in the deceleration section in the position instruction (low word)
D1134: The number of pulses in the deceleration section in the position instruction (high word)
D1147: Setting for the acceleration / deceleration time for CH4 pulse output
D1149: Setting for the acceleration / deceleration time for CH5 pulse output

D1196: When M1568 is ON, you can set the deceleration time for CH4 pulse output.

D1197: When M1569 is ON, you can set the deceleration time for CH5 pulse output.

D1222: The time difference between the direction signal and pulse output sent by CH0.

D1223: The time difference between the direction signal and pulse output sent by CH1.

D1336: Low word of the current number of output pulses from CH0.

D1337: High word of the current number of output pulses from CH0.

D1338: Low word of the current number of output pulses from CH1.

D1339: High word of the current number of output pulses from CH1.

D1340: Settings of the first start frequency and the last end frequency of CH0.

D1343: Settings of acceleration/deceleration time for CH0 pulse output.

D1348: Deceleration time for CH0 pulse output when M1534 = On.

D1349: Deceleration time for CH1 pulse output when M1535 = On.

D1350: Deceleration time for CH2 pulse output when M1536 = On.

D1351: Deceleration time for CH3 pulse output when M1537 = On.

D1352: Settings of the first start frequency and the last end frequency of CH1.

D1353: Settings of acceleration/deceleration time for CH1 pulse output.

D1375: Low word of the current number of output pulses from CH2.

D1376: High word of the current number of output pulses from CH2.

D1377: Low word of the current number of output pulses from CH3.

D1378: High word of the current number of output pulses from CH3.

D1379: Settings of the first start frequency and the last end frequency of CH2.

D1380: Settings of the first start frequency and the last end frequency of CH3.

D1381: Settings of acceleration/deceleration time for CH2 pulse output.

D1382: Settings of acceleration/deceleration time for CH3 pulse output.

D1383: The time difference between the direction signal and pulse output sent by CH2.

D1384: The time difference between the direction signal and pulse output sent by CH3.

API	Mnemonic		Operands	Function
159	D	DRVA	(S1) (S2) (D1) (D2)	Drive to Absolute

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	DRVA: 9 steps DDRVA: 17 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
D_1		*														
D_2		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Number of output pulses (absolute designation) **S_2**: Pulse output frequency **D_1**: Pulse output device (please use transistor output module) **D_2**: Output device for the signal of rotation direction

Explanations:

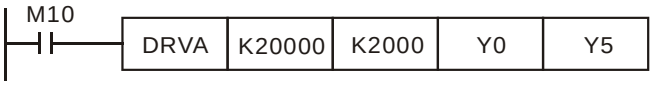
1. See remarks for the setting range of **S_1**, **S_2**, **D_1** and **D_2**.
2. EC3-8K only supports 32-bit instruciton.
3. Flag: see remarks of API 158 DRVI for more details.
4. **S_1** is the number of output pulses (absolute designation). For EH3/SV2 series MPU, the 16-bit instruction can designate the range -32,768 ~ +32,767. The range designated by 32-bit instruction is -2,147,483,648 ~ +2,147,483,647. If the absolute position and the current position in **S_1** are the same, which means the relative output pulse is 0. Then to execute this instruciton will NOT output any pulse but the special M flag will be ON, indicating the output is complete.
5. **S_2** is the designated pulse output frequency. For EH3/SV2 series MPU, the 16-bit instruction can designate its range 10 ~ 32,767Hz. The range designated by 32-bit instruction is 10 ~ 200,000Hz. For EC3-8K series MPU, the 32-bit instruction can designate the range 1 ~ 10,000Hz.
6. Pulse output device **D_1** in different models

Model	EH3/28SV2	24SV2	EC3-8K
Pulse output end	Y0, Y2, Y4, Y6	Y0, Y2, Y4, Y6, Y10, Y12	Y0, Y2

7. EH3/SV2 MPU: When **S_1** is larger than the current position, **D_2** will be ON and go in the forward direction; when **S_1** is smaller than the current relative position, **D_2** will be OFF and go in the backward directon. **D_2** will not be Off immediately after the pulse output is over; it will be Off only when the drive contact of the instruction turns Off.
8. EC3-8K MPU: When **S_1** is larger than the current position, **D_2** will be OFF and go in the forward direction; when **S_1** is smaller than the current relative position, **D_2** will be ON and go in the forward direction. **D_2** will not be Off immediately after the pulse output is over; it will be Off only when the drive contact of the instruction turns Off.

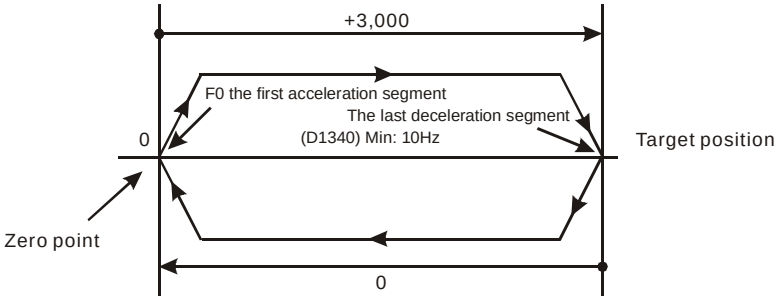
Program Example:

When M10= On, Y0 will output 20,000 pulses (absolute designation) at 2kHz. Y5 = On indicates the pulses are executed in forward direction.

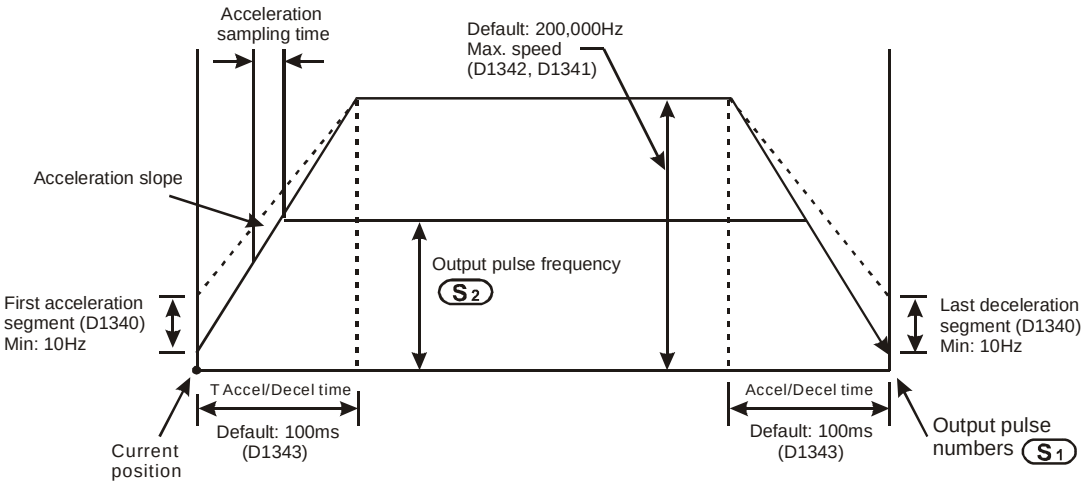


Remarks:

1. Explanations on EH3/SV2 series MPU:
   a) Absolute position control: Designating the traveling distance starting from the zero point (0); also known as a absolute driving method.

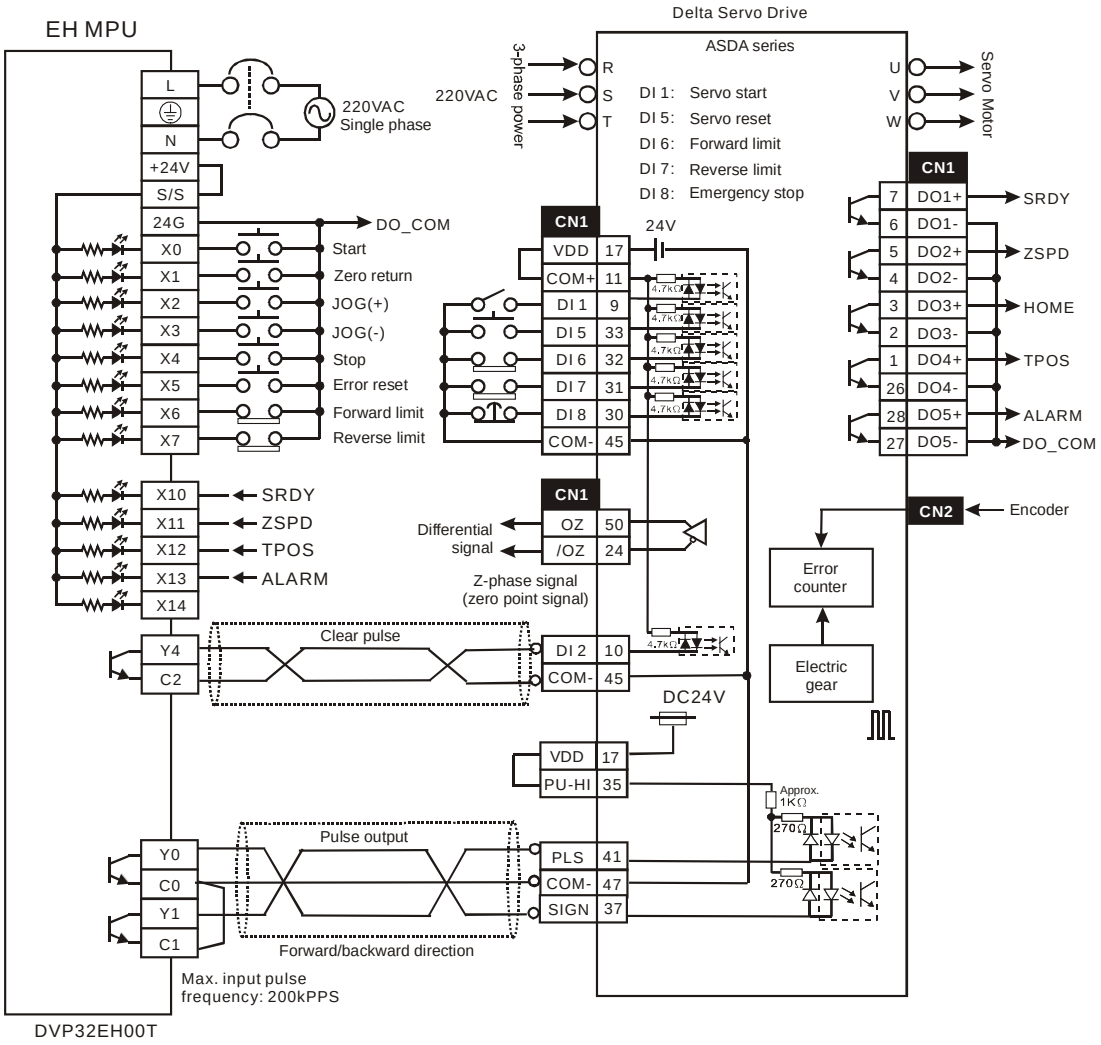


   b) Settings of absolute positioning and the acceleration/deceleration speed:



2. See remarks of DDRVI instruction for more details on the flags.

3. Wiring of DVP-EH3 series and Delta ASDA servo drive:



Note:

(a) The parameter setting of Delta ASDA servo drive:

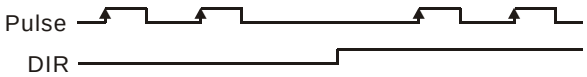
P1-01: position mode

P1-00: pulse input type as Pulse+DIR.

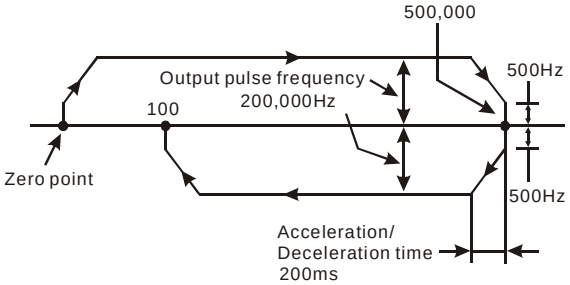
(b) The forward/reverse limit switch should be connected to SERVO AMP.

(c) The “clear pulse” signal will clear the current number of pulses left inside the servo.

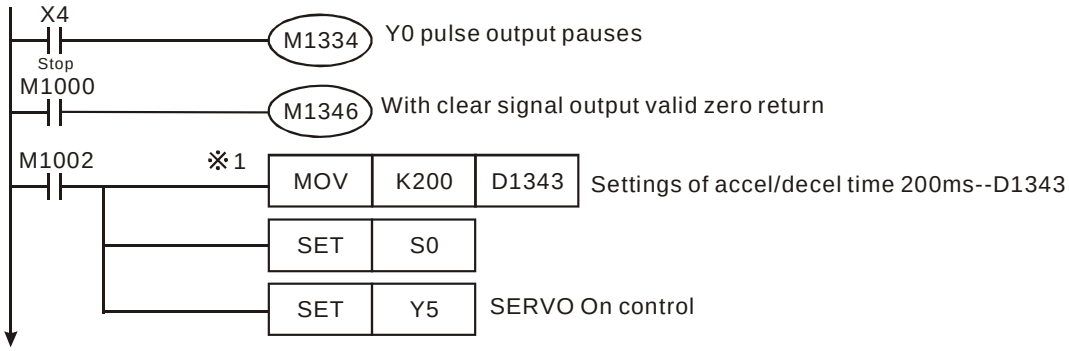
4. Cautions when designing a position control program:
   a) There is no limitation on the times of using the position control instructions, API 156 ZRN, API 157 PLSV, API 158 DRVI, and API 159 DRVA. However, the user still have to note that:
      i. Do not execute the position control instructions which use the same output CH0(Y0, Y1) or CH1(Y2, Y3) simultaneously in a scan cycle. Otherwise, they will be treated as repeated outputs and the upcoming instruciton cannot function normally.
      ii. It is recommended that you use the step ladder instruction (STL) to design the position control program (see the example below).
   b) How to use the position control instructions (API 156 ABSR, API 157 PLSV, API 158 DRVI, and API 159 DRVA) and pulse output instructions (API 57 PLSY, API 58 PWM and API 59 PLSR) at the same time.
      The position control instruction and pulse output instruction share the 32 bits of the present value register (D1337 high word; D1336 low word) of CH0 (Y0, Y1) or the present value register of CH1 (Y2, Y3), which will make the operation complicated. Therefore, it is recommanded that you replace the pulse output instruction with position control instruction.
   c) Explanations on the (Y0, Y1) pulses from CH0 and (Y2, Y3) pulses from CH1.
      Voltage range: DC5V ~ DC24V
      Current range: 10mA ~ 100mA
      Output pulse frequency: Y0, Y2 at 200kHz; Y1, Y3 at 10kHz.
5. Settings of pulse output signals in the operation of position control for EH3/SV2/EC3-8K series MPU:
   a) Pulse + DIR



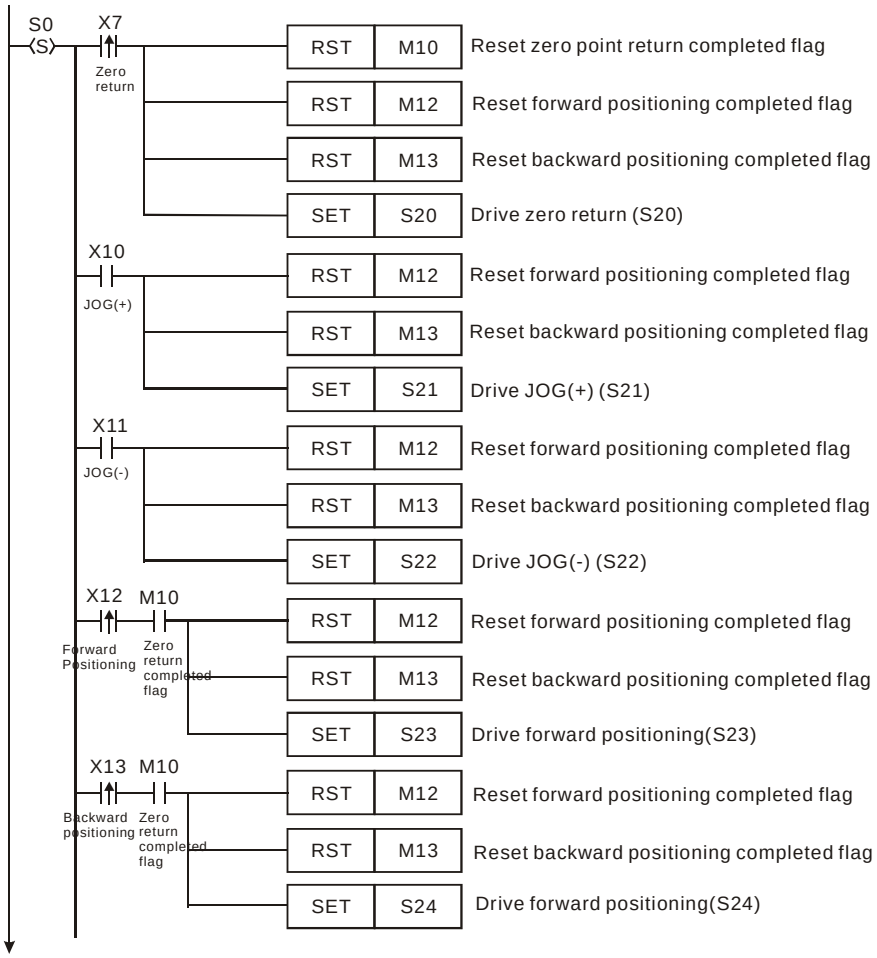
6. For EH3/SV2 series MPU, when Y0 output adopts many high-speed pulse output instructions (PLSY, PWM, PLSR) and position control instructions (ZRN, PLSV, DRVI, DRVA) in a program and these instructions are executed synchronously in the same scan period, PLC will execute the instruction with the fewest step numbers.
   Programming example for forward/reverse operation:
   For the wiring, see the wiring drawing of DVP-EH series and Mitsubishi MR-J2-□A servo drive
   One operation mode performs positioning by absolute position:



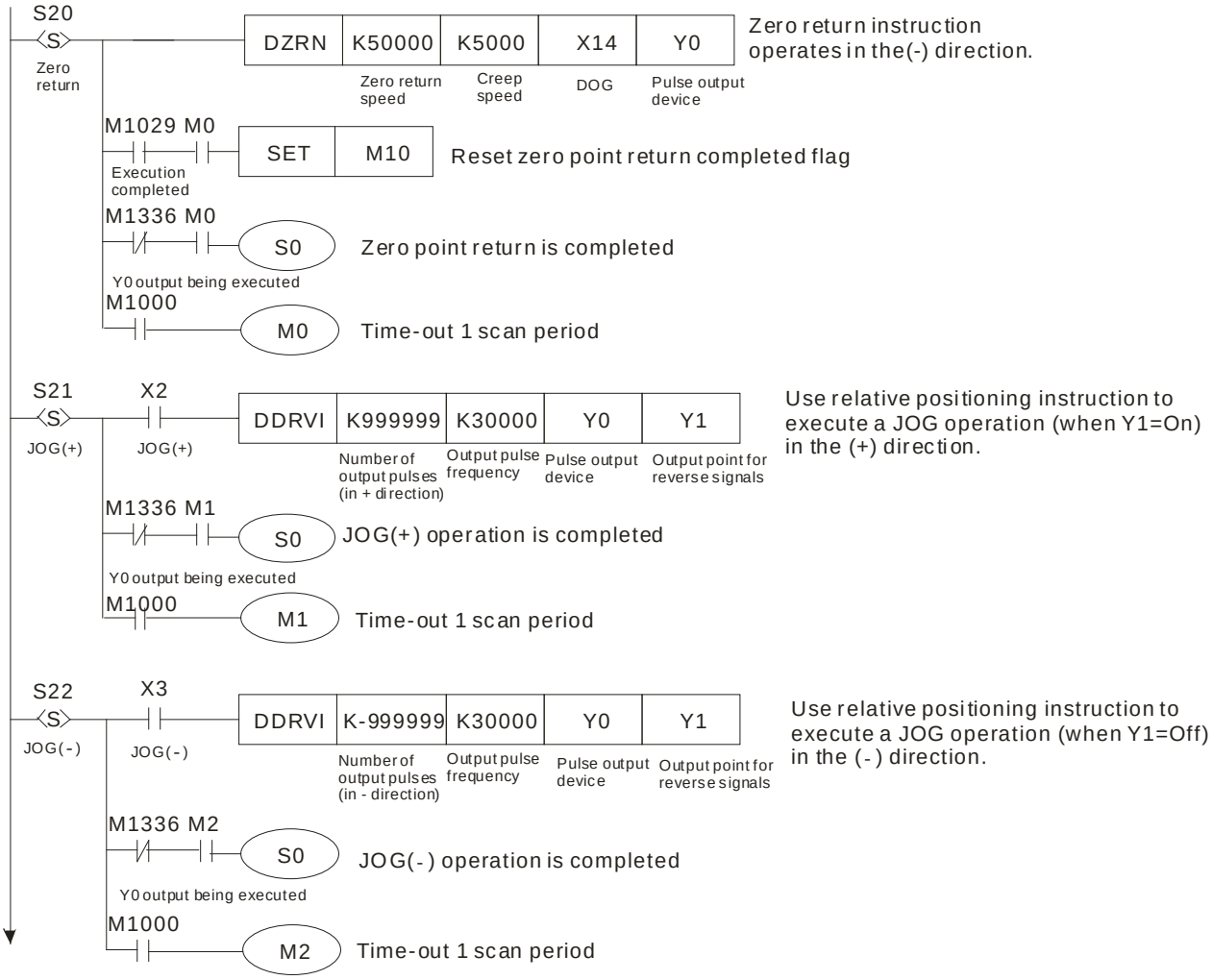
7. Programming example of using step ladder instruction (STL):

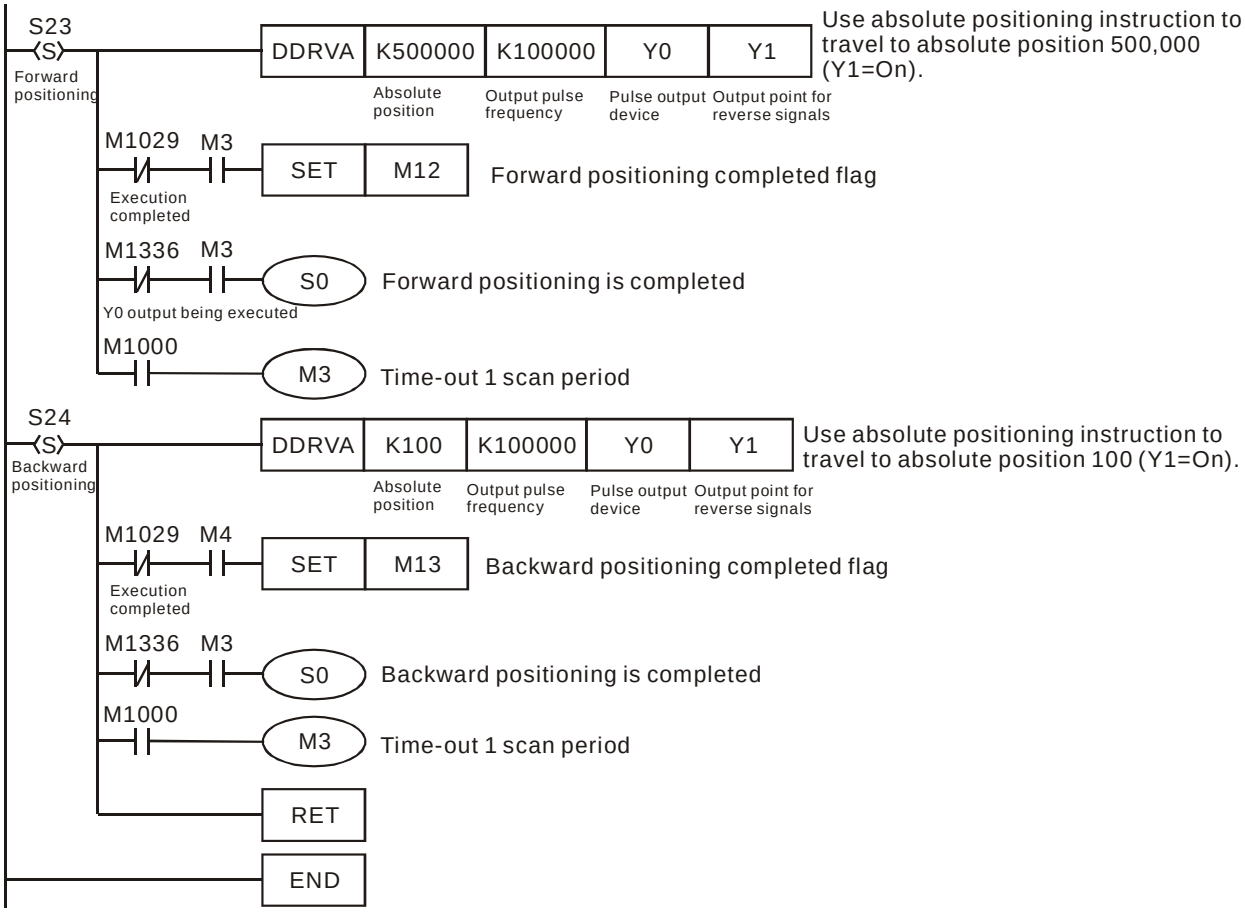


※1. If the accel./decel. time (D1343) of CH1 can be default setting, (100ms) this program step can be ignored.



※2. The max. traveling distance of a JOG operation equals to the max. number of output pulses (-2,147,483,648 ~ +2,147,483,647) of API 158 DDRVI instruction. Please re-execute JOG of the traveling distance exceeds the range.




S23
S
Forward positioning
DDRVA
K500000
K100000
Y0
Y1
Absolute position
Output pulse frequency
Pulse output device
Output point for reverse signals
Use absolute positioning instruction to travel to absolute position 500,000 (Y1=On).
M1029
M3
Execution completed
SET
M12
Forward positioning completed flag
M1336
M3
Y0 output being executed
S0
Forward positioning is completed
M1000
M3
Time-out 1 scan period
S24
S
Backward positioning
DDRVA
K100
K100000
Y0
Y1
Absolute position
Output pulse frequency
Pulse output device
Output point for reverse signals
Use absolute positioning instruction to travel to absolute position 100 (Y1=On).
M1029
M4
Execution completed
SET
M13
Backward positioning completed flag
M1336
M3
S0
Backward positioning is completed
M1000
M3
Time-out 1 scan period
RET
END

API	Mnemonic		Operands	Function
160	TCMP	P	(S1) (S2) (S3) (S) (D)	Time Compare

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	TCMP, TCMPP: 11 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	
S_3					*	*	*	*	*	*	*	*	*	*	*	
S											*	*	*			
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

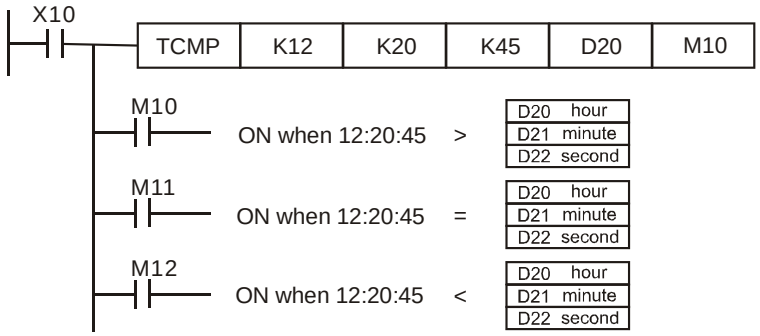
S_1: "Hour" for comparison **S_2**: "Minute" for comparison **S_3**: "Second" for comparison **S**: Current time of RTC **D**: Comparison result

Explanations:

1. Range of **S_1**: K0 ~ K23; range of **S_2** and **S_3**: K0 ~ K59
2. **S** will occupy 3 consecutive devices; **D** will occupy 3 consecutive points.
3. See the specifications of each model for their range of use.
4. **S_1**, **S_2** and **S_3** are compared with the present values of "hour", "minute" and "second" starting from **S**. The comparison result is stored in **D**.
5. **S** is the "hour" of the current time (K0 ~ K23) in RTC; **S** + 1 is the "minute" (K0 ~ K59) and **S** + 2 is the "second" (K0 ~ K59).
6. **S** is read by TRD instruction and the comparison is started by TCMP instruction. If **S** exceeds the range, the program will regard this as an operation error and the instruction will not be executed, M1067 and M1068 = On and D1067 will record the error code 0E1A (hex).

Program Example:

1. When X10= On, the instruction will compare the current time in RTC (D20 ~ D22) with the set value 12:20:45 and display the result in M10 ~ M12. When X10 goes from On to Off, the instruction will not be executed, but the On/Off stauts prior to M10 ~ M12 will remain.
2. Connect M10 ~ M12 in series or in parallel to obtain the result of ≧, ≦, and ≠.



API	Mnemonic		Operands	Function
161	TZCP	P	(S1) (S2) (S) (D)	Time Zone Compare

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			TZCP, TZCPP: 9 steps
S_2											*	*	*			
S											*	*	*			
D		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

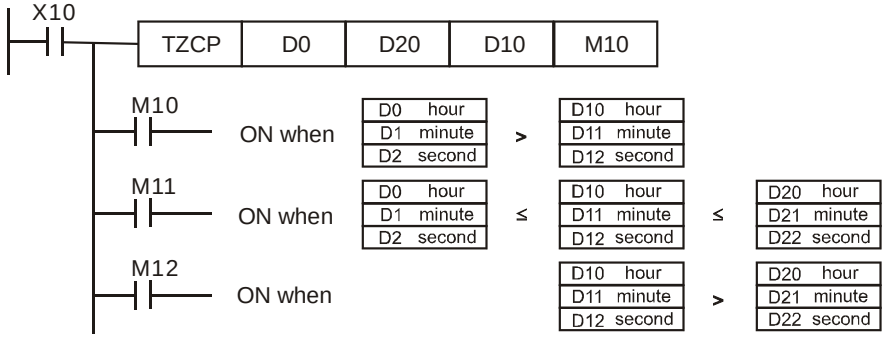
S_1: Lower bound of the time for comparison **S_2**: Upper bound of the time for comparison **S**: Current time of RTC **D**: Comparison result

Explanations:

1. **S_1**, **S_2**, and **S** will occupy 3 consecutive devices.
2. The content in **S_1** must be less than the content in **S_2**.
3. **D** will occupy 3 consecutive points.
4. See the specifications of each model for their range of use.
5. **S** is compared with **S_1** and **S_2**. The comparsion result is stored in **D**.
6. **S_1** , **S_1** +1, **S_1** +2: The “hour”, “minute” and “second” of the lower bound of the time for comparison.
7. **S_2**, **S_2** +1, **S_2** +2: The “hour”, “minute” and “second”ond” of the upper bound of the time for comparison.
8. **S** , **S** +1, **S** +2: The “hour”, “minute” and “second” of the current time of RTC.
9. D0 designated by **S** is read by TRD instruction and the comparison is started by TZCP instruction. If **S_1**, **S_2**, and **S** exceed their ranges, the program will regard this as an operation error and the instruction will not be executed, M1067 and M1068 = On and D1067 will record the error code 0E1A (hex).
10. When **S** < **S_1** and **S** < **S_2**, **D** will be On. When **S** > **S_1** and **S** > **S_2**, **D** + 2 will be On. In other occasions, **D** + 1 will be On.

Program Example:

When X10= On, TZCP instruction will be executed and one of M10 ~ M12 will be On. When X10 = Off, TZCP instruction will not be executed and the status of M10 ~ M12 prior to X10 = Off will remain unchanged.



API	Mnemonic		Operands	Function
162	TADD	P	S_1 S_2 D	Time Addition

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			TADD, TADDP: 7 steps
S_2											*	*	*			
D											*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Time summand **S_2**: Time addend **D**: Time sum

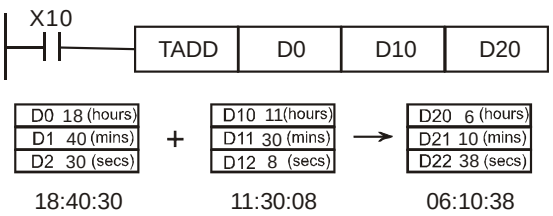
Explanations:

1. **S_1**, **S_2**, and **D** will occupy 3 consecutive devices.
2. See the specifications of each model for their range of use.
3. Flags: M1020 (zero flag); M1022 (carry flag)
4. **S_1** + **S_2** = **D**. The hour, minute, and second of the RTC designated in **S_1** plus the hour, minute, and second designated in **S_2**. The result is stored in the hour, minute, and second of the register designated in **D**.
5. If **S_1** and **S_2** exceed their ranges, the program will regard this as an operation error and the instruction will not be executed. M1067 and M1068 will be On and D1067 record the error code 0E1A (hex).
6. If the sum is larger than 24 hours, the carry flag M1022 will be On and the value in **D** will be the result of "sum minuses 24 hours".
7. If the sum equals 0 (00:00:00), the zero flag M1020 will be On.

Program Example:

1. When X10= On, TADD instruction will be executed and the hour, minute and second in RTC designated in D0 ~ D2 will plus the hour, minute and second in RTC designated in D10 ~ D12. The sum is stored in the hour, minute and second of the register designated in D20 ~ D22.

2. If the sum is larger than 24 hours, M1022 will be On.



API	Mnemonic	Operands	Function
163	TSUB P	(S1) (S2) (D)	Time Subtraction

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			TSUB, TSUBP: 7 steps
S_2											*	*	*			
D											*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

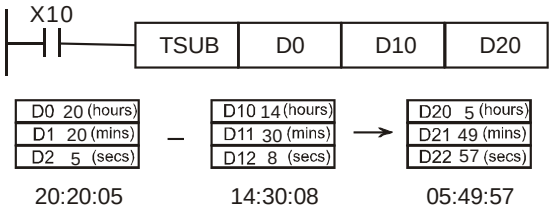
S_1: Time minuend **S_2**: Time subtrahend **D**: Time remainder

Explanations:

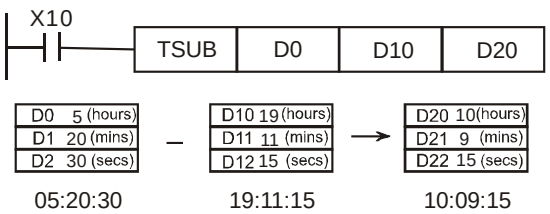
1. **S_1**, **S_2**, and **D** will occupy 3 consecutive devices.
2. See the specifications of each model for their range of use.
3. Flags: M1020 (zero flag); M1021 (borrow flag)
4. **S_1** − **S_2** = **D**. The hour, minute, and second of the RTC designated in **S_1** minus the hour, minute, and second designated in **S_2**. The result is stored in the hour, minute, and second of the register designated in **D**.
5. If **S_1** and **S_2** exceed their ranges, the program will regard this as an operation error and the instruction will not be executed. M1067 and M1068 will be On and D1067 record the error code 0E1A (hex).
6. If the remainder is a negative value, the borrow flag M1021 will be On. The value in **D** will be the result of "the negative value pluses 24 hours".
7. If the remainder equals 0 (00:00:00), the zero flag M1020 will be On.

Program Example:

1. When X10= On, TADD instruction will be executed and the hour, minute and second in RTC designated in D0 ~ D2 will minus the hour, minute and second in RTC designated in D10 ~ D12. The remainder is stored in the hour, minute and second of the register designated in D20 ~ D22.



2. If the subtraction result is a negative value, M1021 will be On.



API	Mnemonic			Operands	Function
166		TRD	P	D	Time Read

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D											*	*	*			TRD, TRDP: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

D: The device for storing the current time read in RTC

Explanations:

1. **D** will occupy 7 consecutive devices.
2. See the specifications of each model for their range of use.
3. Flags: M1016, M1017, M1076. See remarks for more details.
4. The built-in RTC in EH3/SV2/SX series MPU offers 7 data (year, week, month, day, hour, minute, second) stored in D1319 ~ D1313. TRD instruction is for program designers to read the current data in RTC and store the data to the 7 registers designated.
5. D1319 only reads the 2-digit year in A.D. If you wish D1319 to read the 4-digit year, see remarks for more information.

Program Example:

1. When X0 = On, the instruction will read the current time in RTC to the designated registers D0 ~ D6.
2. The content of D1318: 1 = Monday; 2 = Tuesday … 7 = Sunday.

Special D	Item	Content		General D	Item
D1319	Year (A.D.)	00~99	→	D0	Year (A.D.)
D1318	Day (Mon ~ Sun)	1~7	→	D1	Day (Mon ~ Sun)
D1317	Month	1~12	→	D2	Month
D1316	Day	1~31	→	D3	Date
D1315	Hour	0~23	→	D4	Hour
D1314	Minute	0~59	→	D5	Minute
D1313	Second	0~59	→	D6	Second

Remarks:

1. Flags and special registers for the built-in RTC in SX/EH3/SV2 series MPU.

Device	Name	Function
M1016	Displaying year in A.D. in RTC	When Off, D1319 will display 2-digit year in A.D. When On, D1319 will display "2-digit year in A.D + 2,000".
M1017	±30 seconds correction	Correction takes place when M1017 goes from Off to On (reset to 0 when in 0 ~ 29 second; minute pluses 1 and second resets to 0 in 30 ~ 59 second)
M1076	Malfunction of RTC	On when the set value exceeds the range. (only available when the power is being switched on).
D1313	Second	0 ~ 59
D1314	Minue	0 ~ 59
D1315	Hour	0 ~ 23
D1316	Day	1 ~ 31
D1317	Month	1 ~ 12
D1318	Week	1 ~ 7
D1319	Year	0 ~ 99 (2-digit year in A.D.)

2. How to correct RTC:

   There are 2 ways to correct the built-in RTC.

   a) By a specific instruction. (See API 167 TWR instruction)

   b) By peripheral devices, WPLSoft, the ladder diagram editing software.

3. How to display 4-digit year in A.D.:

   a) Normally, the year is only displayed in 2 digits (e.g. 2003 displayed as 03). If you wish the year to be displayed in 4 digits, please key in the following program at the start of the program.

display 4-digit year in A.D.

   b) The original 2-digit year will be switched to a 4-digit year, i.e. the 2-digit year will pluses 2,000.

   c) If you wish to write in new time in the 4-digit year display mode, you can only write in a 2-digit year (0 ~ 99, indicating year 2000 ~ 2099). For example, 00=year 2000, 50=year 2050 and 99=year 2099. However, 2000 ~ 2099 can be written in SX V3.0 and above.

API	Mnemonic	Operands	Function
167	TWR P	(S)	Time Write

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			TWR, TWRP: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Device for storing the new time to be written into RTC

Explanations:

1. **S** will occupy 7 consecutive devices.
2. See the specifications of each model for their range of use.
3. Flags: M1016, M1017, M1076. See remarks of API 166 TRD for more details.
4. To make adjustment on the RTC built in SX/EH3/SV2 series MPU, use this instruction to write the correct time into the RTC.
5. When this instruction is executed, the new set time will be written in the RTC built in PLC immediately. Therefore, please be noted that the new set time has to match the current time then when the instruction is executed.
6. If **S** exceeds its range, the program will regard it as an operation error and the instruction will not be executed. M1067 and M1068 will be On and D1067 will record the error code 0E1A (hex).
7. If you wish to write in new time in the 4-digit year display mode, you can only write in a 2-digit year (0 ~ 99, indicating year 2000 ~ 2099). For example, 00=year 2000, 50=year 2050 and 99=year 2099. However, 2000 ~ 2099 can be written in SX V3.0 and above

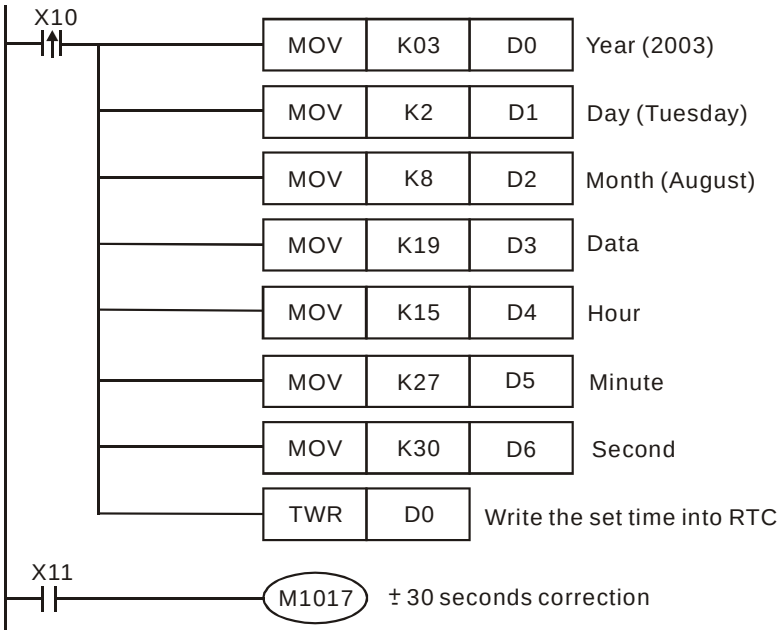
Program Example 1:

When X0= On, write the correct current time into the RTC.

	General D	Item	Content		Special D	Item	
New set time	D20	Year (A.D.)	00~99	→	D1319	Year (A.D.)	Real Time Clock
	D21	Day (Mon ~ Sun)	1~7	→	D1318	Day (Mon ~ Sun)	
	D22	Month	1~12	→	D1317	Month	
	D23	Date	1~31	→	D1316	Date	
	D24	Hour	0~23	→	D1315	Hour	
	D25	Minute	0~59	→	D1314	Minute	
	D26	Second	0~59	→	D1313	Second	

Program Example 2:

1. Set the current time in the RTC as 15:27:30, Tuesday, August 19, 2003.
2. D0 ~ D6 indicate the new set time in the RTC.
3. X10 = On for changing the current time in the RTC and make the changed value the new set value.
4. Whenever X11 = On, RTC will perform a ±30 second correction. The correction is performed according to the rules: When the second hand of RTC locates at 1 ~ 29, the second will be automatically reset to “0” and the minute hand will remain at its location. When the second hand locates at 30 ~ 59, the second will be automatically reset to “0” and the minute hand will increase by 1 minute.



API	Mnemonic			Operands	Function
168	D	MVM	P	(S1) (S2) (D)	Move the Designated Bit

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*	*	*	MVM, MVMP: 7 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	DMVM,DMVMP:
D							*	*	*	*	*	*	*	*	*	13 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source device 1 **S_2**: Bits to be masked (OFF) **D**: Source device 2 / Operation results

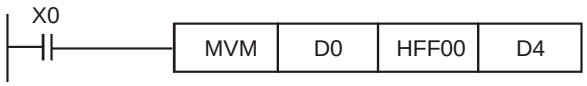
[**D** = (**S_1** & **S_2**) | (**D** & ~**S_2**)]

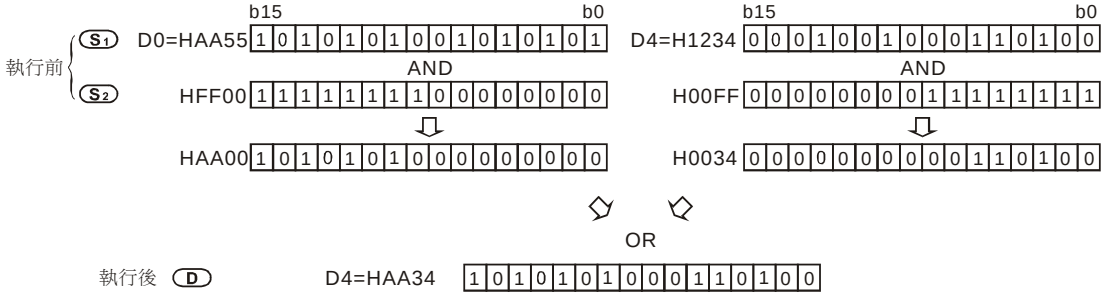
Explanations:

4. The instruction conducts logical AND operation between **S_1** and **S_2** first, logical AND operation between **D** and ~**S_2** secondly, and combines the 1st and 2nd results in **D** by logical OR operation.
5. Rule of Logical AND operation: 0 AND 1 = 0, 1 AND 0 = 0, 0 AND 0 = 0, 1 AND 1 = 1
6. Rule of Logical OR operation: 0 OR 1= 1, 1 OR 0 = 1, 0 OR 0 = 0, 1 OR 1 = 1.
7. Among the SX models, only SX V3.0 and above are supported by the 32-bit instruction.

Program Example 1:

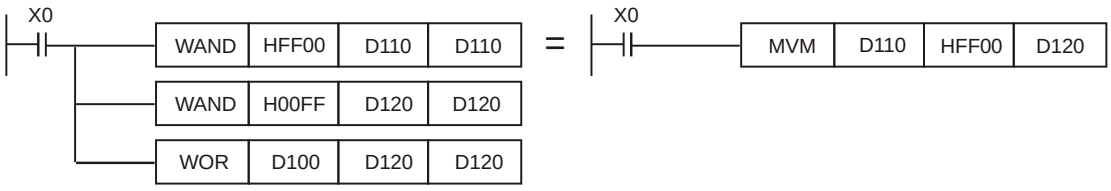
When X0 = ON, MVM instruction conducts logical AND operation between 16-bit register D0 and H'FF00 first, logical AND operation between D4 and H'00FF secondly, and combines the 1st and 2nd results in D4 by logical OR operation.





Program Example 2:

Simplify instructions:



API	Mnemonic	Operands	Function
169	D HOUR	S D_1 D_2	Hour Meter

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	HOUR: 7 steps DHOUR: 13 steps
D_1													*			
D_2		*	*	*												

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

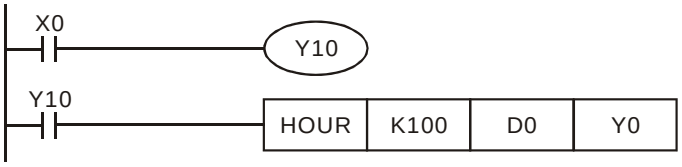
S: Period of time when D_2 is On (in hour) D_1: Current value being measured (in hour) D_2: Output device

Explanations:

1. If **S** is used in device F, only 16-bit instruction is applicable.
2. D_1 will occupy 2 consecutive points. D_1 + 1 uses 16-bit register in 16-bit or 32-bit instruction.
3. See the specifications of each model for their range of use.
4. Range of **S**: K1 ~ K32,767 (unit: hour); range of D_1: K0 ~ K32,767 (unit: hour). D_1 + 1 refers to the current time that is less than an hour (range: K0 ~K3,599; unit: second).
5. This instruction times the time and when the time reaches the set time (in hour), D_2 will be On. This function allows the user to time the operation of the machine or conduct maintenance works.
6. After D_2 is On, the timer will resume the timing.
7. In the 16-bit instruction, when the current time measured reaches the maximum 32,767 hours/3,599 seconds, the timing will stop. To restart the timing, D_1 and D_1 + 1 have to be reset to “0”.
8. In the 32-bit instruction, when the current time measured reaches the maximum 2,147,483,647 hours/3,599 seconds, the timing will stop. To restart the timing, D_1 ~ D_1 + 2 have to be reset to “0”.
9. There is no limitations on the times of using this instruction in the program; however, only 4 instructions can be executed at the same time.

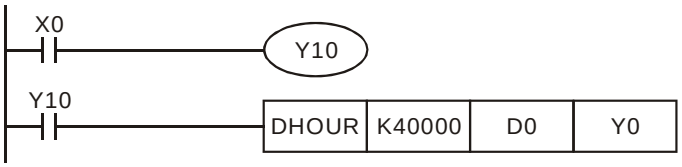
Program Example 1:

In 16-bit instruction, when X0 = On, Y10 will be On and the timing will start. When the timing reaches 100 hours, Y0 will be On and D0 will record the current time measured (in hour) and D1 will record the current time that is less than an hour (0 ~ 3,599; unit: second).



Program Example 2:

In 32-bit instruction, when X0 = On, Y10 will be On and the timing will start. When the timing reaches 40,000 hours, Y0 will be On. D1 and D0 will record the current time measured (in hour) and D2 will record the current time that is less than an hour (0 ~ 3,599; unit: second).



API	Mnemonic			Operands	Function
170	D	GRY	P	(S) (D)	BIN → Gray Code

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GRY, GRYP: 5 steps
S					*	*	*	*	*	*	*	*	*	*	*	DGRY, DGRYP: 9 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

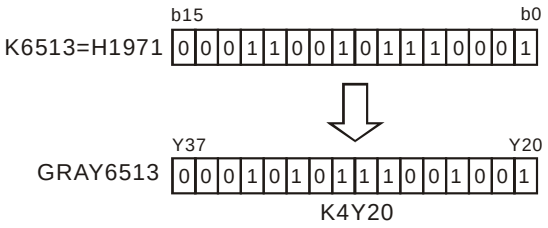
S: Source device for BIN value **D**: Device for storing Gray code

Explanations:

1. If **S** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. This instruction converts the BIN value in the device designated in **S** into Gray code and stores the value in **D**.
4. See the ranges of **S** as indicated below. If **S** exceeds the ranges, the program will regard it as an operation error and the instruction will not be executed. M1067 and M1068 will be On and D1067 will record the error code 0E1A (hex).
   In 16-bit instruction: 0 ~ 32,767
   In 32-bit instruction: 0 ~ 2,147,483,647

Program Example:

When X0 = On, the instruction will convert constant K6,513 into Gray code and store the result in K4Y20.



API	Mnemonic			Operands	Function
171	D	GBIN	P	(S) (D)	Gray Code → BIN

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	GBIN, GBINP: 5 steps DGBIN, DGBINP: 9 steps
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

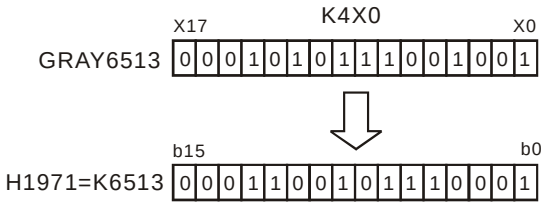
S: Source device for Gray code **D**: Device for storing BIN value

Explanations:

1. If **S** and **D** are used in device F, only 16-bit instruction is applicable.
2. See the specifications of each model for their range of use.
3. This instruction converts the Gray code in the device designated in **S** into BIN value and stores the value in **D**.
4. This instruction converts the content (in Gray code) in the absolute position encoder connected at the PLC input terminal into BIN value and store the result in the designated register.
5. See the ranges of **S** as indicated below. If **S** exceeds the ranges, the program will regard it as an operation error and the instruction will not be executed. M1067 and M1068 will be On and D1067 will record the error code 0E1A (hex).
   In 16-bit instruction: 0 ~ 32,767
   In 32-bit instruction: 0 ~ 2,147,483,647

Program Example:

When X20 = On, the Gray code in the absolute position encoder connected at X0 ~ X17 will be converted into BIN value and stored in D10.



API	Mnemonic			Operands	Function
172	D	ADDR	P	(S1) (S2) (D)	Floating Point Addition

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DADDR, DADDRP: 13 steps
S_2													*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

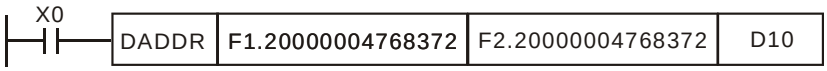
S_1: Floating point summand **S_2**: Floating point addend **D**: Sum

Explanations:

1. **S_1** and **S_2** can be floating point values (FX.XX).
2. See the specifications of each model for their range of use.
3. Flags: M1020 (zero flag), M1021 (borrow flag), M1022 (carry flag)
4. In DADDR instruction, floating point values (e.g. F1.2) can be entered directly into **S_1** and **S_2** or stored in register D for operation. When the instruction is being executed, operand **D** will store the operation result.
5. When **S_1** and **S_2** stores the floating point values in register D, their functions are the same as API 120 EADD.
6. **S_1** and **S_2** can designate the same register. In this case, if the "continuous execution" type instruction is in use and during the On period of the drive contact, the register will be added once in every scan by a "pulse execution" type instruction (DADDRP).
7. If the absolute value of the operation result is larger than the maximum floating point displayable, the carry flag M1022 will be On.
8. If the absolute value of the operation result is smaller than the minimum floating point displayable, the borrow flag M1021 will be On.
9. If the operation result is "0", the zero flag M1020 will be On.

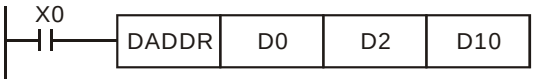
Program Example 1:

When X0 = On, the floating point F1.20000004768372 will plus F2.20000004768372 and the result F3.40000009536743 will be stored in the data registers (D10, D11).



Program Example 2:

When X0 = On, the floating point value (D1, D0) + floating point value (D3, D2) and the result will be stored in the registers designated in (D11, D10).



API	Mnemonic			Operands	Function
173	D	SUBR	P	S1 S2 D	Floating Point Subtraction

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DSUBR, DSUBRP: 13 steps
S_2													*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

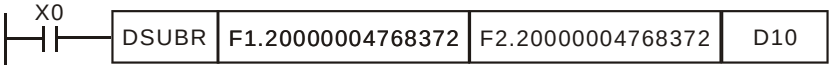
S_1: Floating point minuend **S_2**: Floating point subtrahend **D**: Remainder

Explanations:

1. **S_1** and **S_2** can be floating point values (FX.XX).
2. See the specifications of each model for their range of use.
3. Flags: M1020 (zero flag), M1021 (borrow flag), M1022 (carry flag)
4. In DSUBR instruction, floating point values (e.g. F1.2) can be entered directly into **S_1** and **S_2** or stored in register D for operation. When the instruction is being executed, operand **D** will store the operation result.
5. When **S_1** and **S_2** stores the floating point values in register D, their functions are the same as API 121 ESUB.
6. **S_1** and **S_2** can designate the same register. In this case, if the “continuous execution” type instruction is in use and during the On period of the drive contact, the register will be subtracted once in every scan by a “pulse execution” type instruction (DSUBRP).
7. If the absolute value of the operation result is larger than the maximum floating point displayable, the carry flag M1022 will be On.
8. If the absolute value of the operation result is smaller than the minimum floating point displayable, the borrow flag M1021 will be On.
9. If the operation result is “0”, the zero flag M1020 will be On.

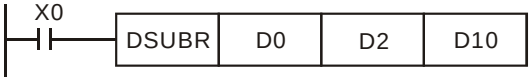
Program Example 1:

When X0 = On, the floating point F1.20000004768372 will minus F2.20000004768372 and the result F-1 will be stored in the data registers (D10, D11).



Program Example 2:

When X0 = On, the floating point value (D1, D0) – floating point value (D3, D2) and the result will be stored in the registers designated in (D11, D10).



API	Mnemonic			Operands	Function
174	D	MULR	P	(S1) (S2) (D)	Floating Point Multiplication

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DMULR, DMULRP: 13 steps
S_2													*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

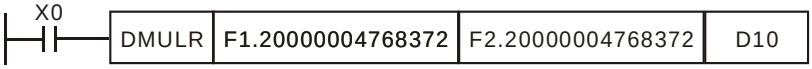
S_1: Floating point multiplicand **S_2**: Floating point multiplicator **D**: Product

Explanations:

1. **S_1** and **S_2** can be floating point values (FX.XX).
2. See the specifications of each model for their range of use.
3. Flags: M1020 (zero flag), M1021 (borrow flag), M1022 (carry flag)
4. In DMULR instruction, floating point values (e.g. F1.2) can be entered directly into **S_1** and **S_2** or stored in register D for operation. When the instruction is being executed, operand **D** will store the operation result.
5. When **S_1** and **S_2** stores the floating point values in register D, their functions are the same as API 122 EMUL.
6. **S_1** and **S_2** can designate the same register. In this case, if the "continuous execution" type instruction is in use and during the On period of the drive contact, the register will be multiplied once in every scan by a "pulse execution" type instruction (DMULRP).
10. If the absolute value of the operation result is larger than the maximum floating point displayable, the carry flag M1022 will be On.
11. If the absolute value of the operation result is smaller than the minimum floating point displayable, the borrow flag M1021 will be On.
12. If the operation result is "0", the zero flag M1020 will be On.

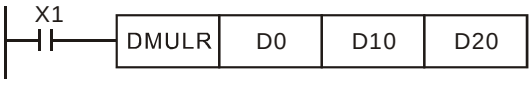
Program Example 1:

When X0 = On, the floating point F1.20000004768372 will multiply F2.20000004768372 and the result F2.64000010490417 will be stored in the data registers (D10, D11).



Program Example 2:

When X1 = On, the floating point value (D1, D0) × floating point value (D11, D10) and the result will be stored in the registers designated in (D21, D20).



API	Mnemonic			Operands	Function
175	D	DIVR	P	(S1) (S2) (D)	Floating Point Division

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DDIVR, DDIVRP: 13 steps
S_1													*			
S_2													*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

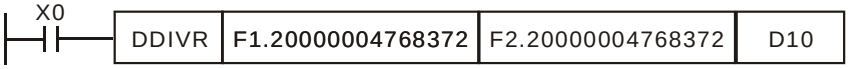
S_1: Floating point dividend **S_2**: Floating point divisor **D**: Quotient

Explanations:

1. **S_1** and **S_2** can be floating point values.
2. See the specifications of each model for their range of use.
3. Flags: M1020 (zero flag), M1021 (borrow flag), M1022 (carry flag)
4. In DDIVR instruction, floating point values (e.g. F1.2) can be entered directly into **S_1** and **S_2** or stored in register D for operation. When the instruction is being executed, operand **D** will store the operation result.
5. When **S_1** and **S_2** stores the floating point values in register D, their functions are the same as API 123 EDIV.
6. If **S_2** is "0", the program will regard it as an operation error and the instruction will not be executed. M1067 and M1068 will be On and D1067 will record the error code H'0E19.
7. If the absolute value of the operation result is larger than the maximum floating point displayable, the carry flag M1022 will be On.
8. If the absolute value of the operation result is smaller than the minimum floating point displayable, the borrow flag M1021 will be On.
9. If the operation result is "0", the zero flag M1020 will be On.

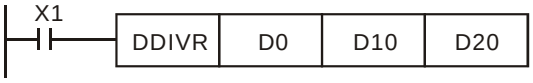
Program Example 1:

When X0 = On, the floating point F1.20000004768372 will be divided by F2.20000004768372 and the result F0.545454561710358 will be stored in the data registers (D10, D11).



Program Example 2:

When X1 = On, the floating point value (D1, D0) ÷ floating point value (D11, D10) and the quotient will be stored in the registers designated in (D21, D20).



API	Mnemonic		Operands	Function
176	MMOV	P	(S) (D)	Magnifying Transfer with Sign Extension

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			MMOV, MMOVP: 5 steps
D											*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

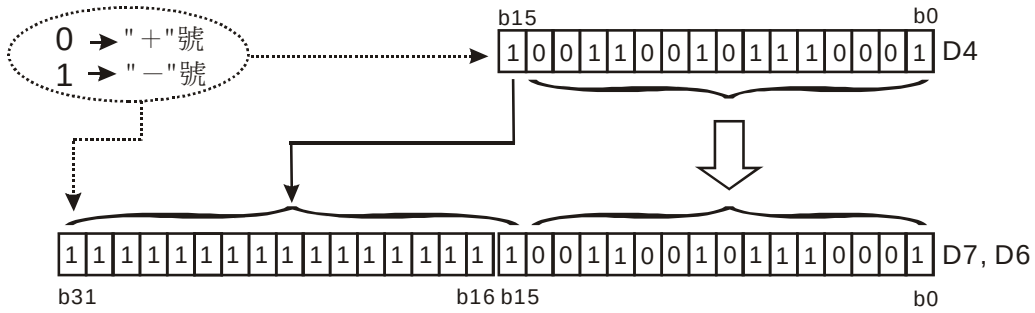
S: Data source (16-bit) **D**: Data destination (32-bit)

Explanations:

1. MMOV instruction sends the data in the 16-bit S device to the 32-bit D device. The designated sign bit will be copied and stored in the destination device.

Program Example 1:

When X23 = On, the data in D4 will be sent to D6 and D7.



In the example, b15 of D4 is sent to b15 ~ b31 of (D7, D6) as a negative value (same as it is in D4).

API	Mnemonic	Operands	Function
177	GPS	(S) (D)	GPS data receiving

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			GPS: 5 steps
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Sentence identifier for GPS data receiving **D**: Destination device for feedback data

Explanations:

1. GPS data receiving instruction is only applicable on COM1 (RS-232), with communication format: 9600,8,N,1, protocol: NMEA-0183, and communication frequency: 1Hz.
2. Operand **S** is sentence identifier for GPS data receiving. K0: $GPGGA, K1: $GPRMC.
3. Operand **D** stores the received data. Up to 17 consecutive words will be occupied and can not be used repeatedly. Please refer to the table below for the explanations of each **D** device.

- When **S** is set as K0, sentence identifier $GPGGA is specified. **D** devices refer to:

No.	Content	Range	Format	Note
D + 0	Hour	0 ~ 23	Word	
D + 1	Minute	0 ~ 59	Word	
D + 2	Second	0 ~ 59	Word	
D + 3~4	Latitude	0 ~ 90	Float	Unit: dd.mmmmmm
D + 5	North / South	0 or 1	Word	0(+)→North, 1(-)→South
D + 6~7	Longitude	0 ~ 180	Float	Unit: ddd.mmmmmm
D + 8	East / West	0 or 1	Word	0(+)→East, 1(-)→West
D + 9	GPS data valid / invalid	0, 1, 2	Word	0 = invalid
D + 10~11	Altitude	0 ~9999.9	Float	Unit: meter
D + 12~13	Latitude	-90 ~ 90	Float	Unit: ±dd.ddddd
D + 14~15	Longitude	-180 ~ 180	Float	Unit: ±ddd.ddddd

- When **S** is set as K1, sentence identifier $GPRMC is specified. **D** devices refer to:

No.	Content	Range	Format	Note
D + 0	Hour	0 ~ 23	Word	
D + 1	Minute	0 ~ 59	Word	
D + 2	Second	0 ~ 59	Word	
D + 3~4	Latitude	0 ~ 90	Float	Unit: dd.mmmmmm
D + 5	North / South	0 or 1	Word	0(+)→North, 1(-)→South
D + 6~7	Longitude	0 ~ 180	Float	Unit: ddd.mmmmmm

No.	Content	Range	Format	Note
D + 8	East / West	0 or 1	Word	0(+)→East, 1(-)→West
D + 9	GPS data valid / invalid	0, 1, 2	Word	0 = invalid
D + 10	Day	1 ~ 31	Word	
D + 11	Month	1 ~ 12	Word	
D + 12	Year	2000 ~	Word	
D + 13~14	Latitude	-90 ~ 90	Float	Unit: ±dd.ddddd
D + 15~16	Longitude	-180 ~ 180	Float	Unit: ±ddd.ddddd

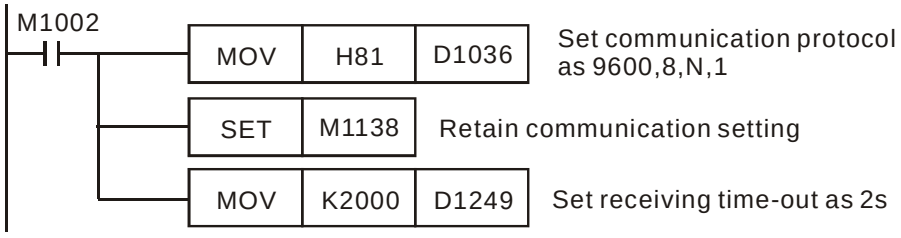
4. When applying GPS instruction, COM1 has to be applied in Master mode, i.e. M1312 has to be enabled to sending request. In addition, M1314 = ON indicates receiving completed. M1315 = ON indicates receiving error. (D1250 = K1, receiving time-out; D1250 = K2, checksum error)
5. Associated M flags and special D registers:

No.	Function
M1312	COM1 (RS-232) sending request
M1313	COM1 (RS-232) ready for data receiving
M1314	COM1 (RS-232) data receiving completed
M1315	COM1 (RS-232) data receiving error
M1138	Retaining communication setting of COM1
D1036	COM1 (RS-232) Communication protocol
D1249	COM1 (RS-232) data receiving time-out setting. (Suggested value: >1s)
D1250	COM1 (RS-232) communication error code

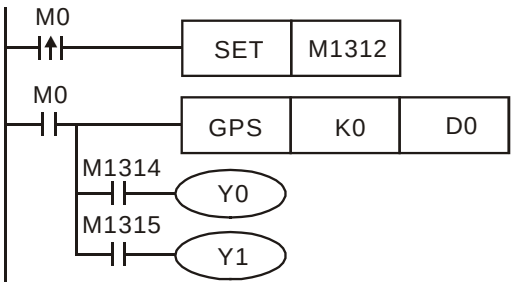
6. Before applying the received GPS data, please check the value in **D**+9. If **D**+9 = 0, the GPS data is invalid.
7. If data receiving error occurs, the previous data in **D** registers will not be cleared, i.e. the previous received data remains intact.

Program example: Sentence identifier: $GPGGA

1. Set COM1 communication protocol first



2. Then enable M0 to execute GPS instruction with sentence identifier $GPGGA

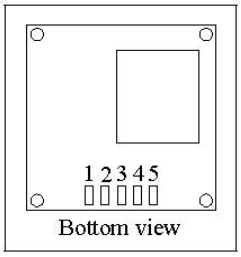


3. When receiving completed, M1314 = ON. When receiving failed, M1315 = ON. The received data will be stored in devices starting with D0.

No.	Content	No.	Content
D0	Hour	D8	East / West
D1	Minute	D9	GPS data valid / invalid
D2	Second	D10~D11	Altitude
D3~D4	Latitude	D12~D13	Latitude. Unit: ±dd.ddddd
D5	North / South	D14~D15	Longitude. Unit: ±ddd.ddddd
D6~D7	Longitude		

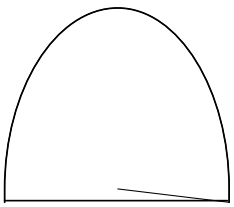
4. Pin number description on GPS module (LS20022)

Pin No. of GPS	1	2	3	4	5
Definition	VCC(+5V)	Rx	Tx	GND	GND



5. Pin number description on PLC COM1:

Pin No. of COM1	1	2	3	4	5	6	7	8
Definition	VCC(+5V)		--	Rx	Tx	--	--	GND



API	Mnemonic			Operands	Function
179	D	WSUM	P	(S) (D) (n)	Sum of multiple devices

OP \ Type	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			WSUM, WSUMP: 7 steps
n					*	*							*			DWSUM, DWSUMP: 13 steps
D											*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

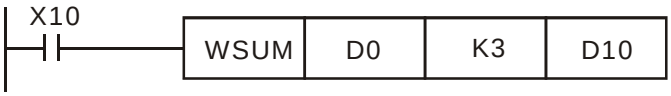
S: Source device **n**: Data length to be summed up **D**: Device for storing the result

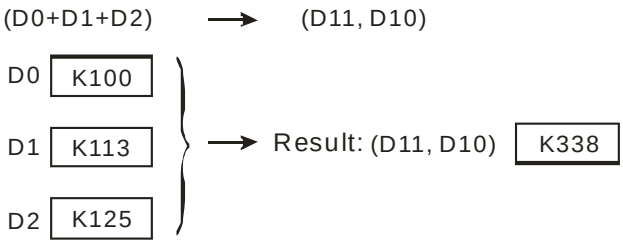
Explanations:

1. WSUM instruction sums up **n** devices starting from **S** and store the result in **D.**
2. The instruction supports SX series PLCs whose version is 3.0 (and above).
3. If the specified source devices **S** are out of valid range, only the devices in valid range will be processed.
4. Valid range for **n**: 1~64. If the specified **n** value is out of the available range (1~64), PLC will take the upper (64) or lower (1) bound value as the set value.
5. **D** used in the 16-bit/32-bit instruction is a 32-bit register.

Program example 1:

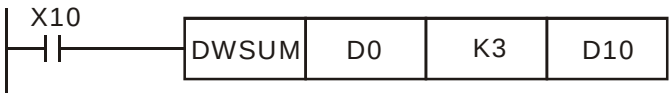
When X10 = ON, 3 consecutive devices (**n** = 3) from D0 will be summed up and the result will be stored in (D11, D10).

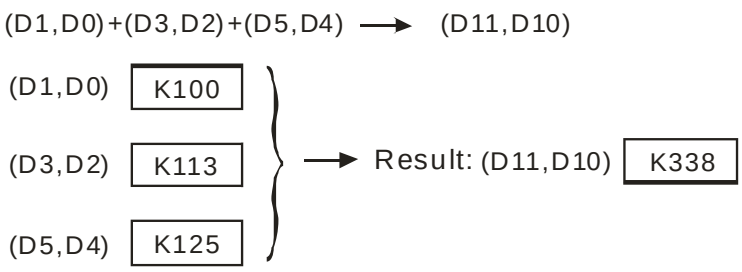




Program example 2:

When X10 = ON, 3 consecutive devices (**n** = 3) from (D1, D0) will be summed up and the result will be stored in (D11, D10).





API	Mnemonic		Operands	Function
180	MAND	P	(S1) (S2) (D) (n)	Matrix 'AND' Operation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MAND, MANDP: 9 steps
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

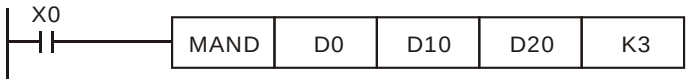
S_1: Matrix source device 1 **S_2**: Matrix source device 2 **D**: Operation result **n**: Array length

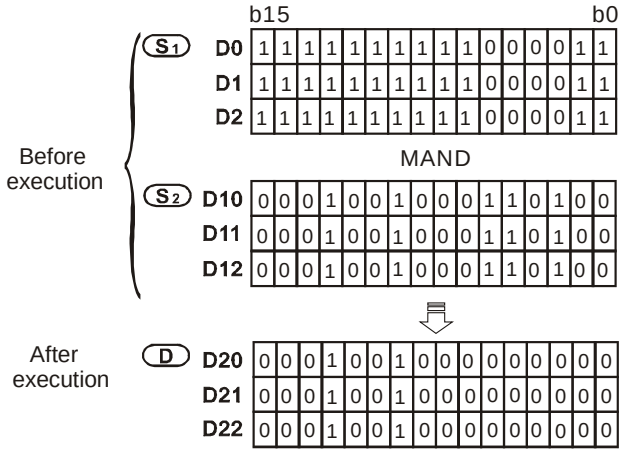
Explanations:

1. Range of **n**: K1 ~ K256
2. **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnYm KnM and KnS
3. SA/SX/SC can designate n = 4. EH3/SV2 can designate n ≦ 4.
4. See the specifications of each model for their range of use.
5. The two matrix sources **S_1** and **S_2** perform matrix 'AND' operation according to the array length **n**. The result is stored in **D**.
6. Operation rule of matix 'AND' : The result will be 1 if both two bits are 1; otherwise the result will be 0.

Program Example:

When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 and the 3 arrays of 16-bit registers D10 ~ D12 will perform a matrix 'AND' operation. The result will be stored in the 3 arrays of 16-bitd registers D20 ~ D22.





Remarks:

1. Explanations on the matrix instruction:

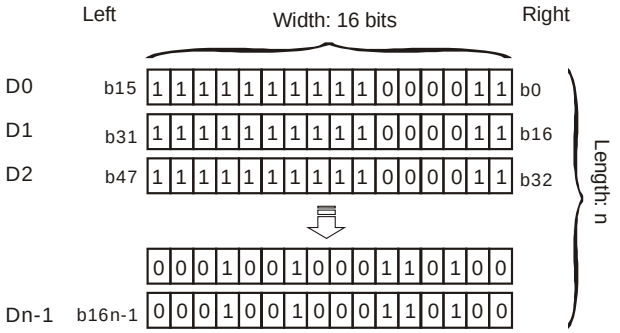
   a) A matix consists of more than 1 consecutive 16-bit registers. The number of registers in the matrix is the length of the array (n). A matrix contains 16 × n bits (points) and there is only 1 bit (point) offered for an operand at a time.

   b) The matrix instruction gathers a series of 16 × n bits (b_0 ~ b_{16n-1}) and designates a single point for operation. The point will not be seen as a value.

   c) The matrix instruction processes the moving, copying, comparing and searching of one-to-many or many-to-many matrix status, which is a very handy and important application instruction.

   d) The matrix operation will need a 16-bit register to designate a point among the 16n points in the matrix for the operation. The register is the Pointer (Pr) of the matrix, designated by the user in the instruction. The vaild range of Pr is 0 ~ 16n -1, corresponding to b0 ~ b16n-1 in the matrix.

   e) There are left displacement, right displacement and rotation in a matrix operation. The bit number decreases from left to right (see the figure below).



   f) The matrix width (C) is fixed at 16 bits.

   g) Pr: matrix pointer. E.g. if Pr is 15, the designated point will be b15.

   h) Array length (R) is n: n = 1 ~ 256.

   Example: The matrix is composed of D0, n = 3; D0 = HAAAA, D1 = H5555, D2 = HAAFF

	C_{15}	C_{14}	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0	
R_0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	**D0**
R_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	**D1**
R_2	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	**D2**

   Example: The matrix is composed of K2X0, n = 3; K2X0 = H37, K2X10 = H68, K2X20 = H45

	C_{15}	C_{14}	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0	
R_0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	X_0~X_7
R_1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	X_{10}~X_{17}
R_2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	X_{20}~X_{27}

   Fill “0” into the blank in R0(C_{15}-C_8), R1(C_{15}-C_8), and R2(C_{15}-C_8).

API	Mnemonic		Operands	Function
181	MOR	P	S_1 S_2 D n	Matrix 'OR' Operation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MOR, MORP: 9 steps
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

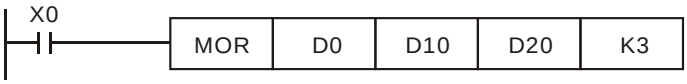
S_1: Matrix source device 1 **S_2**: Matrix source device 2. **D**: Operation result **n**: Array length

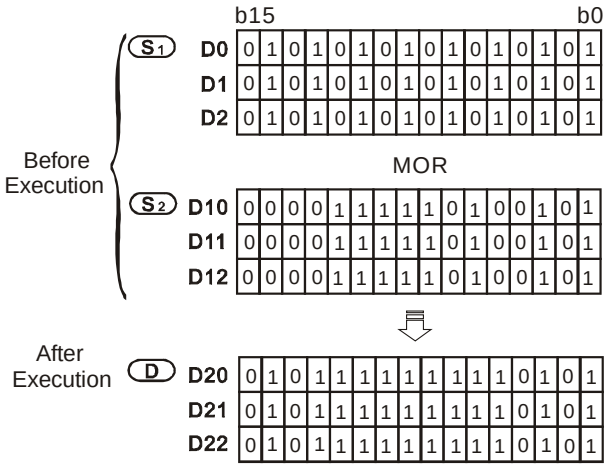
Explanations:

1. Range of **n**: K1 ~ K256
2. **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnYm KnM and KnS
3. SA/SX/SC can designate n = 4. EH3/SV2 can designate n ≦ 4
4. See the specifications of each model for their range of use.
5. The two matrix sources **S_1** and **S_2** perform matrix 'OR' operation according to the array length **n**. The result is stored in **D**.
6. Operation rule of matrix 'OR': The result will be 1 if either of the two bits is 1. The result is 0 only when both two bits are 0.

Program Example:

When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 and the 3 arrays of 16-bit registers D10 ~ D12 will perform a matrix 'OR' operation. The result will be stored in the 3 arrays of 16-bit registers D20 ~ D22.





API	Mnemonic		Operands	Function
182	MXOR	P	(S1) (S2) (D) (n)	Matrix 'XOR' Operation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MXOR, MXORP: 9 steps
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

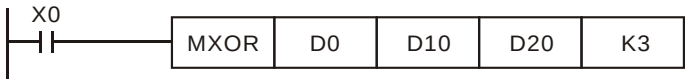
S_1: Matrix source device 1 **S_2**: Matrix source device 2 **D**: Operation result **n**: Array length

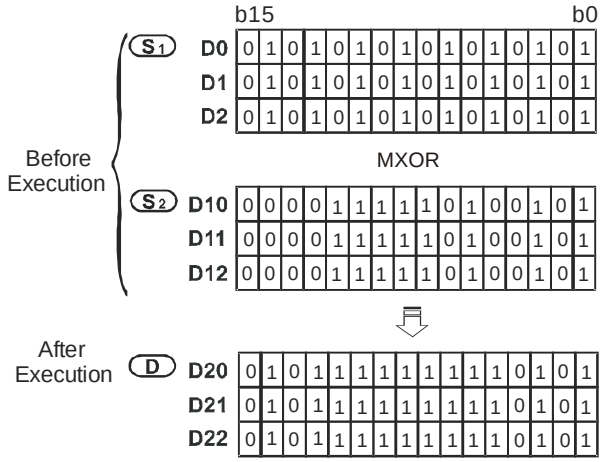
Explanations:

1. Range of **n**: K1 ~ K256
2. **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnYm KnM and KnS
3. See the specifications of each model for their range of use.
4. The two matrix sources **S_1** and **S_2** perform matrix 'XOR' operation according to the array length **n**. The result is stored in **D**.
5. Operation rule of matrix 'XOR': The result will be 1 if the two bits are different. The result will be 0 if the two bits are the same.

Program Example:

When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 and the 3 arrays of 16-bit registers D10 ~ D12 will perform a matrix 'XOR' operation. The result will be stored in the 3 arrays of 16-bit registers D20 ~ D22.





API	Mnemonic		Operands	Function
183	MXNR	P	(S1) (S2) (D) (n)	Matrix 'XNR' Operation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MXNR, MXNRP: 9 steps
S_1							*	*	*	*	*	*	*			
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

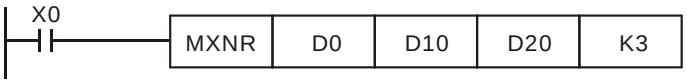
S_1: Matrix source device 1 **S_2**: Matrix source device 2 **D**: Operation result **n**: Array length

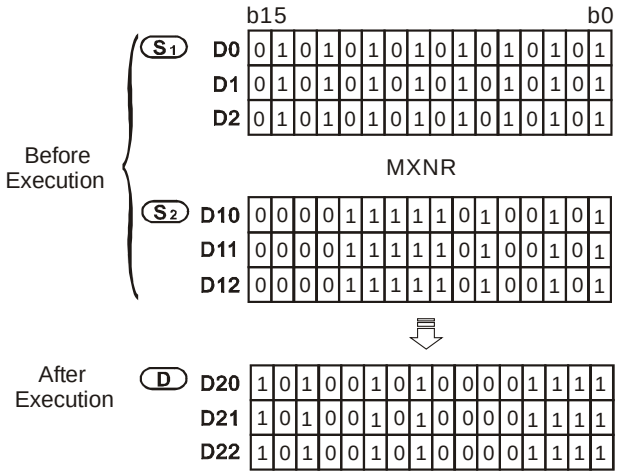
Explanations:

1. Range of **n**: K1 ~ K256
2. **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnYm KnM and KnS
3. See the specifications of each model for their range of use.
4. The two matrix sources **S_1** and **S_2** perform matrix 'XNR' operation according to the array length **n**. The result is stored in **D**.
5. Operation rule of matrix 'XNR': The result will be 1 if the two bits are the same. The result will be 0 if the two bits are different.

Program Example:

When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 and the 3 arrays of 16-bit registers D10 ~ D12 will perform a matrix 'XNR' operation. The result will be stored in the 3 arrays of 16-bit registers D20 ~ D22.





API	Mnemonic		Operands	Function
184	MINV	P	(S) (D) (n)	Matrix Inverse Operation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MINV, MINVP: 7 steps
D								*	*	*	*	*	*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

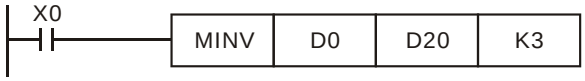
S: Matrix source device **D**: Operation result **n**: Array length

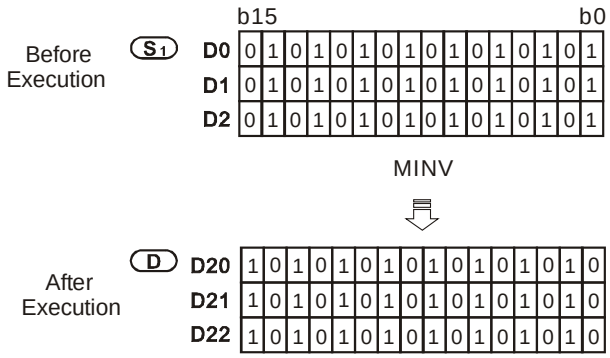
Explanations:

1. Range of **n**: K1 ~ K256
2. **S** designates KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS.
3. See the specifications of each model for their range of use.
4. **S** performs an inverse matrix operation according to the array length **n**. The result is stored in **D**.

Program Example:

When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 perform a matrix inverse operation. The result will be stored in the 3 arrays of 16-bit registers D20 ~ D22.





API	Mnemonic		Operands	Function
185	MCMP	P	(S1) (S2) (n) (D)	Matrix Compare

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MCMP, MCMPP: 9 steps
S_2							*	*	*	*	*	*	*			
n					*	*							*			
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Matrix source device 1 **S_2**: Matrix source device 2 **n**: Array length **D**: Pointer (Pr), for storing the value of target location

Explanations:

1. Range of **n**: K1 ~ K256
2. **S_1**, and **S_2** designate KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS.
3. See the specifications of each model for their range of use.
4. Flags: M1088 ~ M1092. See remarks for more details.
5. This instruction compares every bit in **S_1** with every bit in **S_2** starting from location **D** + 1 and finds out the location of different bits. The location will be stored in **D**.
6. The matrix comparison flag (M1088) decides to compare between equivalent values (M1088 = 1) or different values (M1088 = 0). When the comparison is completed, it will stop immediately and the matrix bit search flag will turn “On” (M1091 = 1). When the comparison progresses to the last bit, the matrix search end flag (M1089) will turn “On” and the No. where the comparison is completed is stored in **D**. The comparison will start from the 0^{th} bit in the next scan period when the matrix search start flag turns “On” (M1090 = 1). When **D** exceeds the range, the pointer error flag will turn “On” (M1092 = 1).
7. The matrix operation will need a 16-bit register to designate a point among the 16n points in the matrix for the operation. The register is the Pointer (Pr) of the matrix, designated by the user in the instruction. The vaild range of Pr is 0 ~ 16n -1, corresponding to b0 ~ b16n-1 in the matrix. Please avoid changing the Pr value during the operation in case the comparing and searching will not ne correct. If the Pr value exceeds its range, M1092 will be On and the instruction will not be executed.
8. When M1089 and M1091 take place at the same time, both flags will be “1” at the same time.

Program Example:

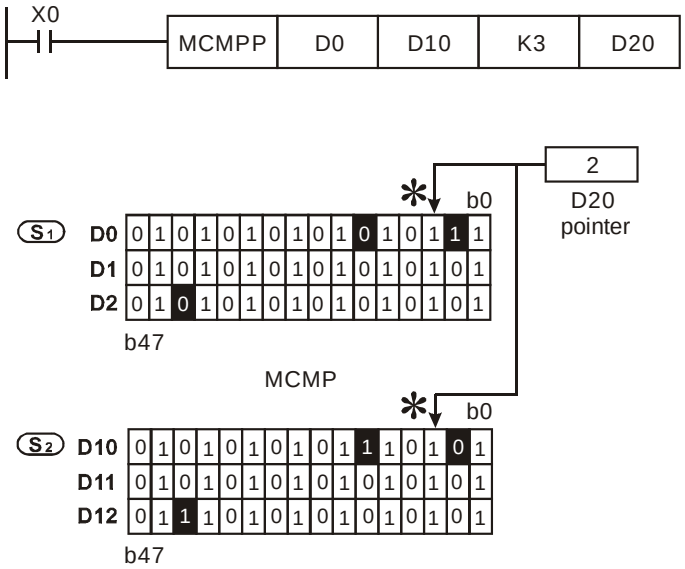
1. When X0 goes from Off to On, the matrix search start falg M1090 = 0. The searching will start from the bit marked with “*” (current Pr value +1) for bits of different status (M1088 = 0).
2. Set the Pr value D20 = 2. When X0 goes from Off to On for 4 times, we can obtain the 4 execution results ❶, ❷, ❸, ❹.

   ❶ D20 = 5, M1091 = 1, M1089 = 0.

   ❷ D20 = 45, M1091 = 1, M1089 = 0.

❸ D20 = 47, M1091 = 0, M1089 = 1.

❹ D20 = 1, M1091 = 1, M1089 = 0.



Remarks:

Flags explanations:

Flags	Function
M1088	Matrix comparison flag. Comparing between equivalent values (M1088 = 1) or different values (M1088 = 0).
M1089	Matrix search end flag. When the comparison reaches the last bit, M1089 = 1.
M1090	Matrix search start flag. Comparing from bit 0 (M1090 = 1).
M1091	Matrix bit search flag. When the comparison is completed, the comparison will stop immediately (M1091=1).
M1092	Matrix pointer error flag. When the pointer Pr exceeds its range, M1092 = 1.

API	Mnemonic		Operands	Function
186	MBRD	P	(S) (n) (D)	Read Matrix Bit

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBRD, MBRDP: 7 steps
n					*	*							*			
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Matrix source device **n**: Array length **D**: Pointer (Pr), for storing the value of target location

Explanations:

1. Range of **n**: K1 ~ K256
2. **S** designates KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS.
3. See the specifications of each model for their range of use.
4. Flags: M1089 ~ M1095. See remarks for more details.
5. When this instruction is executed, it first determines if the matrix pointer clear flag (M1094) is “On”. If it is “On”, pointer D is cleared as 0. The instruction then reads the On/Off status from the 0^{th} bit of **S** to the matrix rotation/displacement/output carry flag (M1095). Whenever finishing reading 1 bit, the instruction determines whether the matrix pointer increasing flag (M1093) is “On”. If it is “On”, the value of pointer D will plus 1. When the reading is processed to the last bit, the matrix search end flag (M1089) will turn “On” and pointer D record the No. of read bits.
6. The Pointer (Pr) of the matrix is designated by the user in the instruction. The vaild range of Pr is 0 ~ 16n -1, corresponding to b_0 ~ b_{16n-1} in the matrix. If the Pr value exceeds its range, M1092 will be On and the instruction will not be executed.

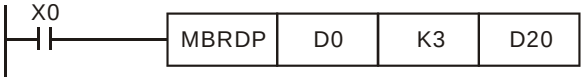
Program Example:

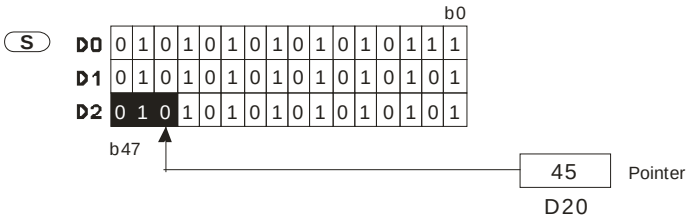
1. When X0 goes from Off to On, M1094 will be set to “0” and M1093 to “1”. Therefore, the Pr will plus 1 after every reading.
2. Set the Pr value D20 = 45. When X0 goes from Off to On for 3 times, we can obtain the 3 execution results ❶, ❷, ❸.

   ❶ D20 = 46, M1095 = 0, M1089 = 0.

   ❷ D20 = 47, M1095 = 1, M1089 = 0.

   ❸ D20 = 47, M1095 = 1, M1089 = 1.





Remarks:

Flag explanations:

Flags	Function
M1088	Matrix search end flag. When the comparison reaches the last bit, M1089 = 1.
M1092	Matrix pointer error flag. When the pointer Pr exceeds its range, M1092 = 1.
M1093	Matrix pointer increasing flag. Adding 1 to the current value of the Pr.
M1094	Matrix pointer clear flag. Clearing the current value of the Pr to 0.
M1095	Matrix rotation/displacement/output carry flag.

API	Mnemonic		Operands	Function
187	MBWR	P	(S) (n) (D)	Write Matrix Bit

OP \ Type	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBWR, MBWRP: 7 steps
n					*	*							*			
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

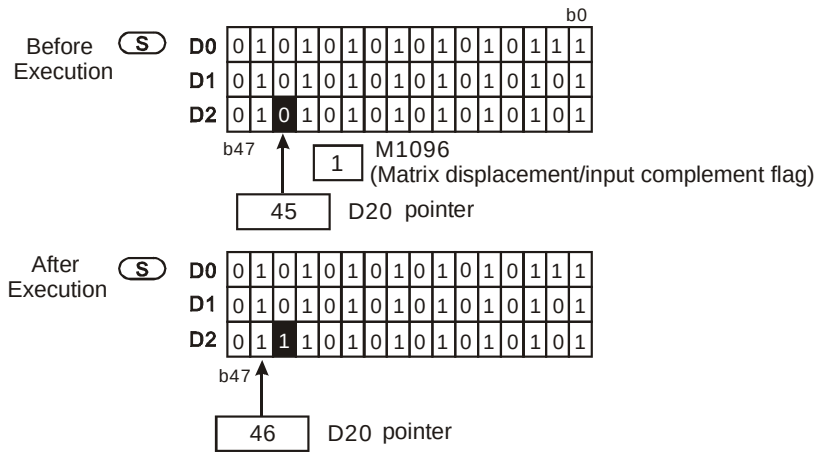
S: Matrix source device **n**: Array length **D**: Pointer (Pr), for storing the value of target location

Explanations:

1. Range of **n**: K1 ~ K256
2. **S** designates KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS.
3. See the specifications of each model for their range of use.
4. Flags: M1089 ~ M1096. See remarks for more details.
5. When this instruction is executed, if first determines if the matrix pointer clear flag (M1094) is “On”, If it is “On”, pointer D is cleared as 0. The instruction then writes the value in the matrix displacement/input complement flag (M1096) into the location starting from the 0^{th} bit of **S**. Whenever finishing writing 1 bit, the instruction determines whether the matrix pointer increasing flag (M1093) is “On”. If it is “On”, the value of pointer D will plus 1. When the writing is processed to the last bit, the matrix search end flag (M1089) will turn “On” and pointer D records the No. of written bits. If D exceeds its range, M1092 will be On.
6. The Pointer (Pr) of the matrix is designated by the user in the instruction. The vaild range of Pr is 0 ~ 16n -1, corresponding to b_0 ~ b_{16n-1} in the matrix. If the Pr value exceeds its range, M1092 will be On and the instruction will not be executed.

Program Example:

1. When X0 goes from Off to On, M1094 will be set to “0” and M1093 to “1”. Therefore, the Pr will plus 1 after every writing.
2. Set the Pr value D20 = 45 and M1096 = 1. When X0 goes from Off to On for 1 time, we can obtain the execution results: D20 = 46, M1096 = 1, M1089 = 0.



Remarks:

Flag explanations:

Flags	Function
M1088	Matrix search end flag. When the comparison reaches the last bit, M1089 = 1.
M1092	Matrix pointer error flag. When the pointer Pr exceeds its range, M1092 = 1.
M1093	Matrix pointer increasing flag. Adding 1 to the current value of the Pr.
M1094	Matrix pointer clear flag. Clearing the current value of the Pr to 0.
M1096	Matrix displacement/input complement flag.

API	Mnemonic		Operands	Function
188	MBS	P	(S) (D) (n)	Matrix Bit Displacement

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBS, MBSP: 7 steps
D								*	*	*	*	*	*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

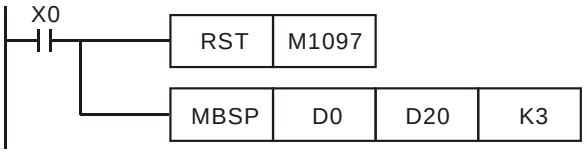
S: Matrix source device **D**: Operation result **n**: Array length

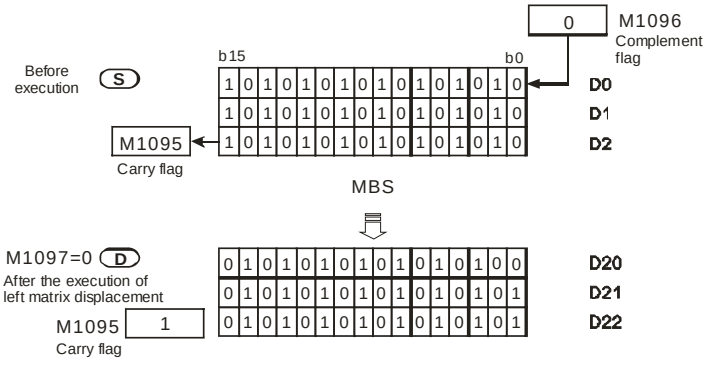
Explanations:

1. Range of **n**: K1 ~ K256
2. **S** designates KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS.
3. See the specifications of each model for their range of use.
4. Flags: M1095 ~ M1097. See remarks for more details.
5. This instruction performs left-right displacement on the matrix bits in **S** according to array length **n**. M1097 determines the left (M1097 = 0) or right (M1097 = 1) displacement of matrix bits. The empty bits derived from every displacement of 1 bit (when left displacement: b_0; when right displacement: b_{16n-1}) is filled by the status of the complement flag (M1096). The spare bits (when left displacement: b_{16n-1}; when right displacement: b_0) are sent to the carry flag (M1095). The result is stored in **D**.
6. The pulse execution instruction MBSP is generally adopted.

Program Example 1:

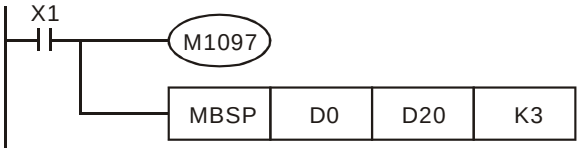
When X0 = On, M1097 = Off, indicating a left matrix displacement is performed. Set M1096 = 0 and the 16-bit registers D0 ~ D2 will perform a left matrix displacement and the result will be stored in the matrix of the 16-bit registers D20 ~ D22. The carry flag M1095 will be “1”.

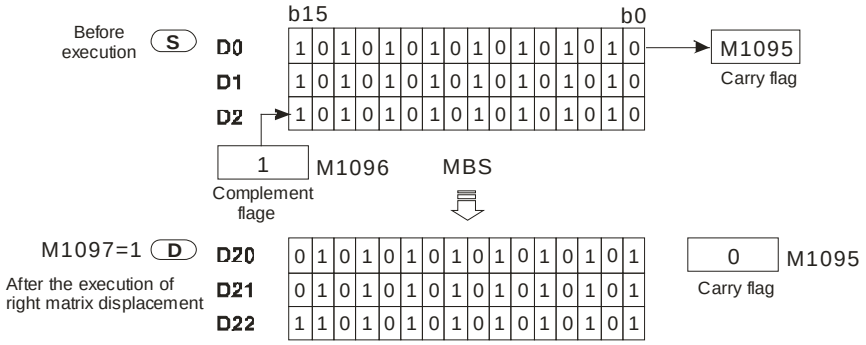




Program Example 2:

When X1 = On, M1097 = On, indicating a right matrix displacement is performed. Set M1096 = 1 and the 16-bit registers D0 ~ D2 will perform a right matrix displacement and the result will be stored in the matrix of the 16-bit registers D20 ~ D22. The carry flag M1095 will be “0”.





Explanations:

Flag explanations:

Flags	Function
M1095	Matrix rotation/displacement/output carry flag.
M1096	Matrix displacement/input complement flag.
M1097	Matrix rotation/displacement direction flag.

API	Mnemonic		Operands	Function
189	MBR	P	(S) (D) (n)	Matrix Bit Rotation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBR, MBRP: 7 steps
D								*	*	*	*	*	*			
n					*	*							*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

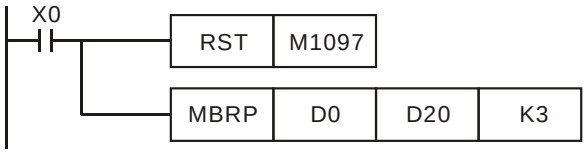
S: Matrix source device **D**: Operation result **n**: Array length

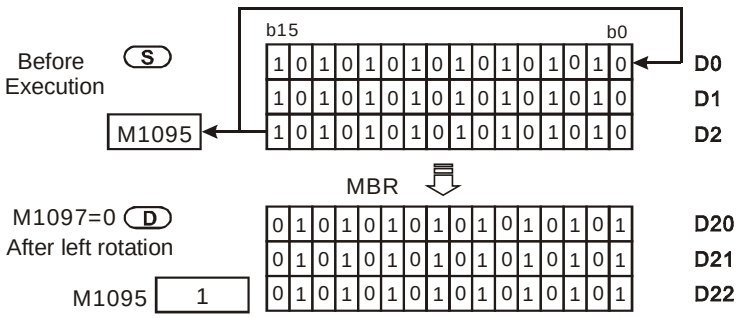
Explanations:

1. Range of **n**: K1 ~ K256
2. **S** designates KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS.
3. See the specifications of each model for their range of use.
4. Flags: M1095, M1097. See remarks for more details.
5. This instruction performs left-right rotation on the matrix bits in **S** according to array length **n**. M1097 determines the left (M1097 = 0) or right (M1097 = 1) rotation of matrix bits. The empty bits derived from every rotation of 1 bit (when left rotation: b_0; when right rotation: b_{16n-1}) is filled by rotation bits (when left rotation: b_{16n-1}; when right rotation: b_0). The result is stored in **D**. Rotation bits not only fill the empty bits but also send the status of bits to the carry flag M1095.
6. The pulse execution instruction MBRP is generally adopted.

Program Example 1:

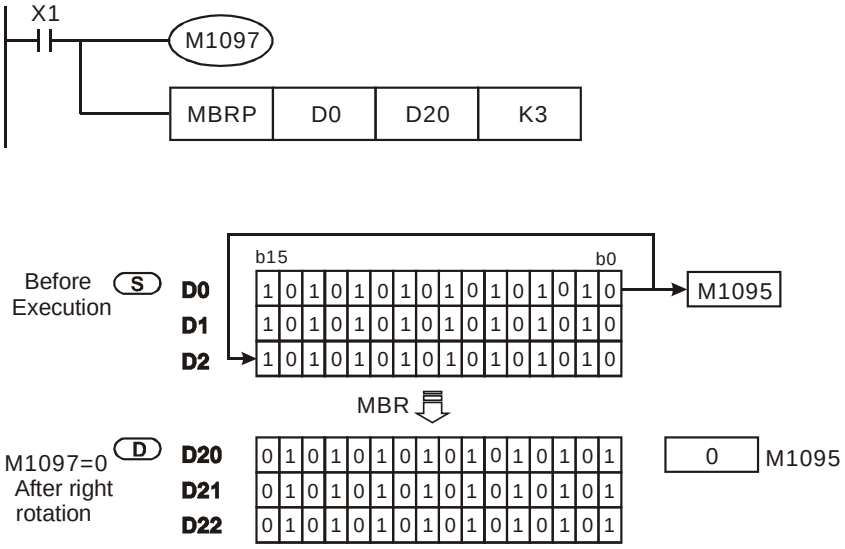
When X0 = On, M1097 = Off, indicating a left matrix rotation is performed. The 16-bit registers D0 ~ D2 will perform a left matrix rotation and the result will be stored in the matrix of the 16-bit registers D20 ~ D22. The carry flag M1095 will be “1”.





Program Example 2:

When X1 = On, M1097 = On, indicating a right matrix rotation is performed. The 16-bit registers D0 ~ D2 will perform a right matrix rotation and the result will be stored in the matrix of the 16-bit registers D20 ~ D22. The carry flag M1095 will be “0”.



Remarks:

Flag explanations:

Flags	Function
M1095	Matrix rotation/displacement/output carry flag.
M1097	Matrix rotation/displacement direction flag.

API	Mnemonic		Operands	Function
190	MBC	P	(S) (n) (D)	Matrix Bit Status Counting

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBC, MBCP: 7 steps
n					*	*							*			
D								*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

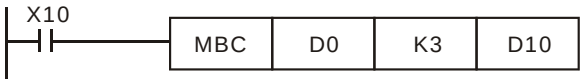
S: Matrix source device **n**: Array length **D**: Counting result

Explanations:

1. Range of **n**: K1 ~ K256
2. **S** designates KnX, KnY, KnM and KnS; **D** designates KnY, KnM and KnS.
3. See the specifications of each model for their range of use.
4. Flags: M1098, M1099. See remarks for more details.
5. This instruction counts the number of bits which are “1” or “0” in **S** by array length **n**. The result is stored in **D**.
6. The instruction counts the number of bits which are “1” when M1098 = 1 and counts the number of bits which are “0” when M1098 = 0. When the operation result is “0”, M1099 = 1.

Program Example:

When X10 = On, in the matrix of D0 ~ D2, when M1098 = 1, the instruction counts the total number of bits which are “1” and store the number in D10. When M1098 = 0, the instruction counts the total number of bits which are “0” and store the number in D10.



D0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1

D10 | 12 | M1098=0

D10 | 36 | M1098=1

Remarks:

Flag explanations:

Flags	Function
M1098	Counting the number of bits which are “1” or “0”
M1099	On when the counting result is “0”.

API	Mnemonic			Operands	Function
191	D	PPMR		(S1) (S2) (S) (D)	2-Axis Relative Point to Point Motion

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DPPMR: 17 steps
S_1					*	*							*			
S_2					*	*							*			
S					*	*							*			
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Number of output pulses of X axis **S_2**: Number of output pulses of Y axis **S**: Max. point to point output frequency **D**: Pulse output device

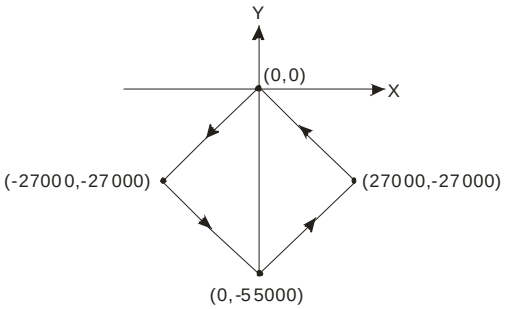
Explanations:

1. Flags: M1029, M1030, M1334, M1335. See remarks for more details.
2. This instruction only supports EH3/SV2 series MPU. In terms of pulse output methods, this instructin only supports “pulse + direction” mode.
3. **S_1** and **S_2** are the designated (relative designation) number of output pulses in X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of the number is -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in forward direction, the pulse present value registers CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word) and CH3 (D1378 high word, D1377 low word) will increase. When in backward direction, the present value will decrease.
4. **D** can designate Y0 and Y4.
   When Y0 is designated:
   Y0 refers to 1st group X-axis pulse output device.
   Y1 refers to 1st group X-axis direction signal.
   Y2 refers to 1st group Y-axis pulse output device.
   Y3 refers to 1st group Y-axis direction signal.
   Y4 refers to 2nd group X-axis pulse output device.
   Y5 refers to 2nd group X-axis direction signal.
   Y6 refers to 2nd group Y-axis pulse output device.
   Y7 refers to 2nd group Y-axis direction signal.
   When direction signal outputs, Off will not occur immediately after the pulse output is over. Direction signal will turn Off when the drive contact is Off. 24SV2 series MPU does NOT support Y10 and Y12.
5. D1340 (D1379) refers to the settings of the start/end frequencies of the 1st/2nd 2-axis motion. D1343 (D1381) refers to the time of the first acceleration segment and last deceleration segment of the 1st/2nd 2-axis motion. The time shall be longer than 10ms. If the time is shorter than 10ms or longer than 10,000ms, the output will be operated at 10ms. Default setting = 100ms.
6. If the maximum output frequency setting is less than 10Hz, the output will be operated at 10Hz. If the setting is more than 200kHz, the output will be operated at 200kHz.

7. When the 2-axis synchronous motion instruction is enabled, the start frequency and acceleration/deceleration time in Y axis will be same as the settings in X axis.
8. The number of output pulses for the 2-axis motion shall not be less than 59; otherwise the line drawn will not be straight enough.
9. There is no limitation on the number of times using the instruction. However, assume CH1 or CH2 output is in use, the 1st group X/Y axis will not be able to output. If CH3 or CH4 output is in use, the 2nd group X/Y axis will not be able to output.

Program Example:

1. Draw a rhombus as the figure below.

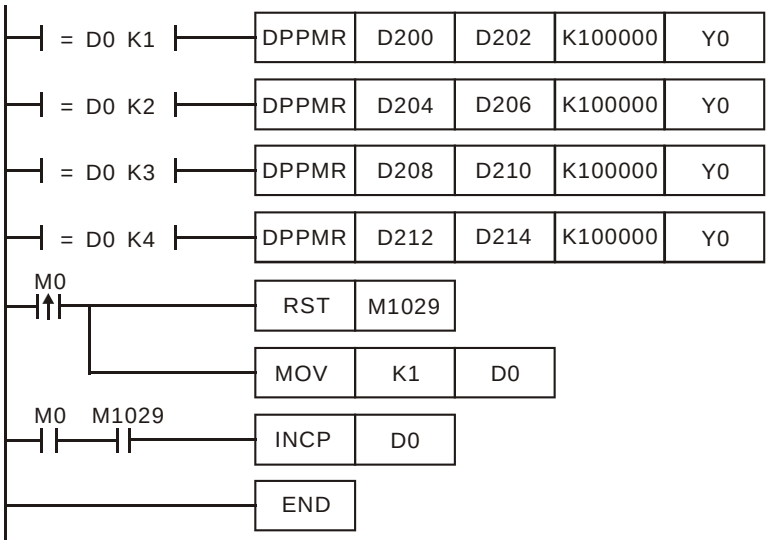


2. Steps:

a) Set the four coordinates (0, 0), (-27000, -27000), (0, -55000), (27000, -27000) (as the figure above). Calculate the relative coordinates of the four points and obtain (-27000, -27000), (27000, -28000), (27000, 27000), and (-27000, 27000). Place them in the 32-bit (D200, D202), (D204, D206), (D208, D210), (D212, D214).

b) Write program codes as follows.

c) PLC RUN. Set M0 as On and start the 2-axis line drawing.



3. Motion explanation:

When PLC RUN and M0 = On, PLC will start the first point-to-point motion by 100kHz. D0 will plus 1 whenever a point-to-point motion is completed and the second point-to-point motion will start to execute automatically. The same motion will keep executing until the fourth point-to-point motion is completed.

Remarks:

1. Flag explanations:

M1029:	On when the 1st group 2-axis pulse output is completed.
M1036:	On when the 2nd group 2-axis pulse output is completed.
M1334 & M1335:	When M1334 and M1335 are On, the first group of pulses outputs of the two axes stops immediately.
M1336:	1st group 2-axis pulse output indication flag
M1520 & M1521:	When M1520 and M1521 are On, the second group of pulse outputs of the two axes stops immediately.
M1522:	2nd group 2-axis pulse output indication flag

2. Special register explanations:

D1336, D1337:	Pulse present value register for Y0 output of the 1st group X-axis motion. The present value increases or decreases following the rotation direction. (D1337 high word; D1336 low word)
D1338, D1339:	Pulse present value register for Y2 output of the 1st group Y-axis motion. The present value increases or decreases following the rotation direction. (D1339 high word; D1338 low word)
D1340:	Frequency settings of the first acceleration and last deceleration segment for the Y0 output of the 1st group X-axis motion and Y2 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.
D1343:	Time settings of the first acceleration and last deceleration segment for the Y0 output of the 1st group X-axis motion and Y2 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.
D1375, D1376:	Pulse present value register for Y4 output of the 2nd group X-axis motion. The present value increases or decreases following the rotation direction. (D1337 high word; D1336 low word)
D1377, D1378:	Pulse present value register for Y6 output of the 2nd group Y-axis motion. The present value increases or decreases following the rotation direction. (D1339 high word; D1338 low word)
D1379:	Frequency settings of the first acceleration and last deceleration segment for the Y4 output of the 2nd group X-axis motion and Y6 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.
D1381:	Time settings of the first acceleration and last deceleration segment for the Y4 output of the 2nd group X-axis motion and Y6 of the Y-axis motion for API 191 DPPMR and API 192 DPPMA.

API	Mnemonic	Operands	Function
192	D PPMA	(S1) (S2) (S) (D)	2-Axis Absolute Point to Point Motion

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DPPMA: 17 steps
S_2					*	*							*			
S					*	*							*			
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Number of output pulses of X axis **S_2**: Number of output pulses of Y axis **S**: Max. point to point output frequency **D**: Pulse output device

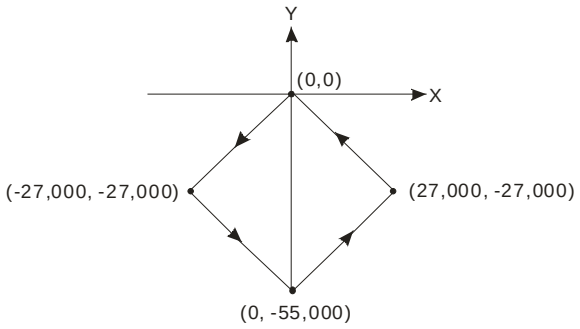
Explanations:

1. Flags: M1029, M1030, M1334, M1335. See remarks of API 191 DPPMR for more details.
2. This instruction only supports EH3/SV2 series MPU. In terms of pulse output methods, this instructin only supports “pulse + direction” mode.
3. **S_1** and **S_2** are the designated (absolute designation) number of output pulses in X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of the number is -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in forward direction, the pulse present value registers CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word) and CH3 (D1378 high word, D1377 low word) will increase. When in backward direction, the present value will decrease.
4. **D** can designate Y0 and Y4.
   When Y0 is designated:
   Y0 refers to 1st group X-axis pulse output device.
   Y1 refers to 1st group X-axis direction signal.
   Y2 refers to 1st group Y-axis pulse output device.
   Y3 refers to 1st group Y-axis direction signal.
   Y4 refers to 2nd group X-axis pulse output device.
   Y5 refers to 2nd group X-axis direction signal.
   Y6 refers to 2nd group Y-axis pulse output device.
   Y7 refers to 2nd group Y-axis direction signal.
   When direction signal outputs, Off will not occur immediately after the pulse output is over. Direction signal will turn Off when the drive contact is Off. 24SV2 series MPU does NOT support Y10 and Y12.
5. D1340 (D1379) refers to the settings of the start/end frequencies of the 1st/2nd 2-axis motion. D1343 (D1381) refers to the time of the first acceleration segment and last deceleration segment of the 1st/2nd 2-axis motion. The time shall be longer than 10ms. If the time is shorter than 10ms or longer than 10,000ms, the output will be operated at 10ms. Default setting = 100ms.
6. If the maximum output frequency setting is less than 10Hz, the output will be operated at 10Hz. If the setting is more than 200kHz, the output will be operated at 200kHz.

7. When the 2-axis synchronous motion instruction is enabled, the start frequency and acceleration/deceleration time in Y axis will be same as the settings in X axis.
8. The number of output pulses for the 2-axis motion shall not be the values within 1 ~ 59; otherwise the line drawn will not be straight enough.
9. There is no limitation on the number of times using the instruction. However, assume CH1 or CH2 output is in use, the 1st group X/Y axis will not be able to output. If CH3 or CH4 output is in use, the 2nd group X/Y axis will not be able to output.

Program Example:

1. Draw a rhombus as the figure below.

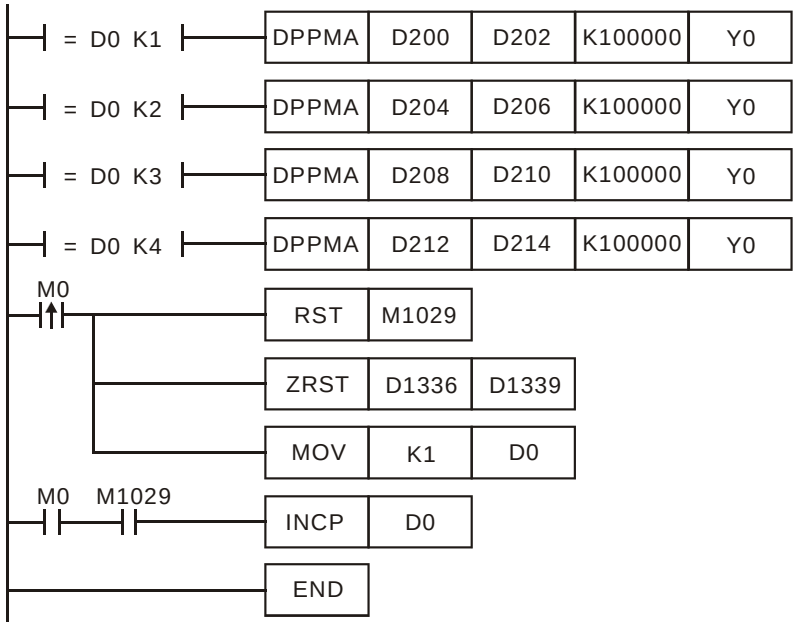


2. Steps:

a) Set the four coordinate (-27,000, -27,000), (0, -55,000), (27,000, -27,000), (0, 0) (as the figure above). Place them in the 32-bit (D200, D202), (D204, D206), (D208, D210), (D212, D214).

b) Write program codes as follows.

c) PLC RUN. Set M0 as On and start the 2-axis line drawing.



3. Motion explanation:

When PLC RUN and M0 = On, PLC will start the first point-to-point motion by 100kHz. D0 will plus 1 whenever a point-to-point motion is completed and the second point-to-point motion will start to execute automatically. The same motion will keep executing until the fourth point-to-point motion is completed.

API	Mnemonic	Operands	Function
193	D CIMR	(S1) (S2) (S) (D)	2-Axis Relative Position Arc Interpolation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DCIMR: 17 steps
S_1					*	*							*			
S_2					*	*							*			
S													*			
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Number of output pulses of X axis **S_2**: Number of output pulses of Y axis **S**: Parameter setting **D**: Pulse output device

Explanations:

1. Flags: M1029, M1030, M1334, M1335. See remarks of API 191 DPPMR for more details.
2. This instruction only supports EH3/SV2 series MPU. In terms of pulse output methods, this instructin only supports “pulse + direction” mode.
3. **S_1** and **S_2** are the designated (relative designation) number of output pulses in X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of the number is -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in forward direction, the pulse present value registers CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word) and CH3 (D1378 high word, D1377 low word) will increase. When in backward direction, the present value will decrease.
4. The lower 16 bits of **S** (settings of direction and resolution): K0 refers to clockwise 10-segment (average resolution) output; K2 refers to clockwise 20-segment (higher resolution) output and a 90° arc can be drawn (see figure 1 and 2). K1 refers to counterclockwise 10-segment (average resolution) output; K3 refers to counterclockwise 20-segment (higher resolution) output and a 90° arc can be drawn (see figure 3 and 4).
5. The higher 16 bits of **S** (settings of motion time): K1 refers to 0.1 second. The setting range for average resolution is K1 ~ K100 (0.1 sec. ~ 10 secs.), for higher resolution is K2 ~ K200 (0.2 sec. ~ 20 secs.) This instruction is restricted by the maximum pulse output frequency; therefore when the set time goes faster than the actual output time, the set time will be automatically modified.

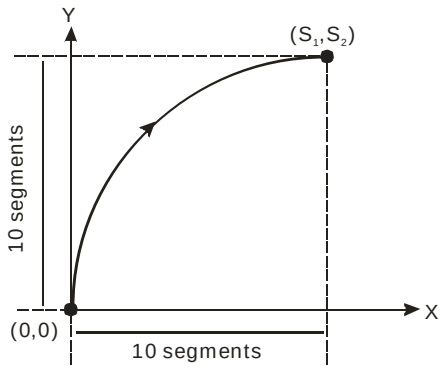


Figure 1

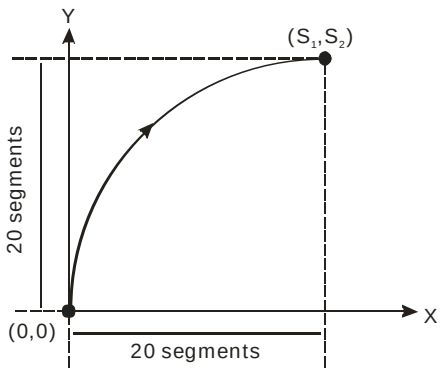


Figure 2

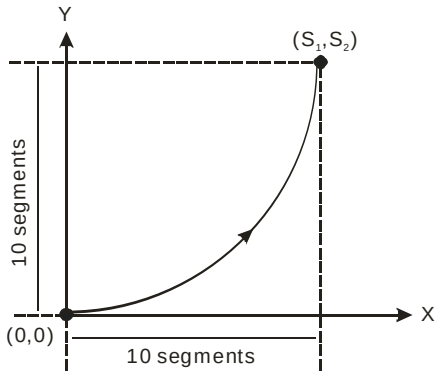


Figure 3

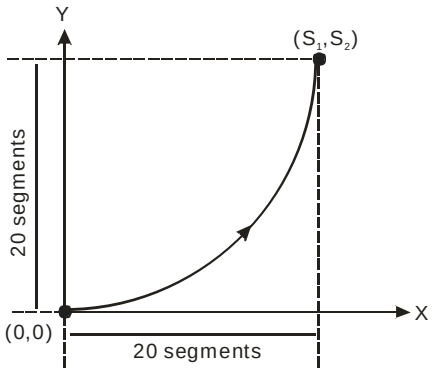


Figure 4

6. **D** can designate Y0 and Y4.
   When Y0 is designated:
   Y0 refers to 1st group X-axis pulse output device.
   Y1 refers to 1st group X-axis direction signal.
   Y2 refers to 1st group Y-axis pulse output device.
   Y3 refers to 1st group Y-axis direction signal.
   When Y4 is designated:
   Y4 refers to 2nd group X-axis pulse output device.
   Y5 refers to 2nd group X-axis direction signal.
   Y6 refers to 2nd group Y-axis pulse output device.
   Y7 refers to 2nd group Y-axis direction signal.
   When direction signal outputs, Off will not occur immediately after the pulse output is over. Direction signal will turn Off when the drive contact is Off. 24SV2 series MPU does NOT support Y10 and Y12.
7. Draw four 90° arcs.
8. When the direction signal is On, the direction is positive. When the direction signal is Off, the direction is negative. When **S** is set as K0, K2, the arcs will be clockwise (see figure 5). When **S** is set as K1, K3, the arcs will be counterclockwise (see figure 6).

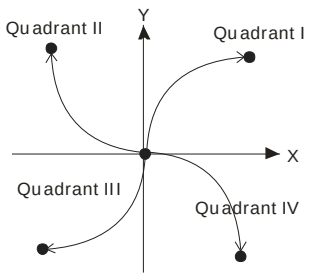


Figure 5

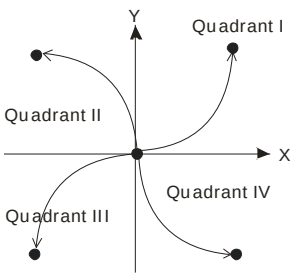


Figure 6

9. When the 2-axis motion is being executed in 10 segments (of average resolution), the operation time of the instruction when the instruction is first enabled is approximately 5ms. The number of output pulses cannot be less than 100 and more than 1,000,000; otherwise, the instruction cannot be enabled.
10. When the 2-axis motion is being executed in 20 segments (of high resolution), the operation time of the

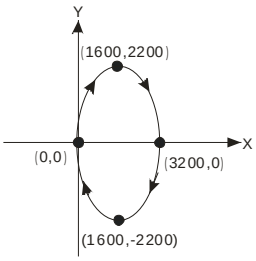
instruction when the instruction is first enabled is approximately 10ms. The number of output pulses cannot be less than 1,000 and more than 10,000,000; otherwise, the instruction cannot be enabled.

11. If you wish the number of pulses in 10-segment or 20-segment motion to be off the range, you may adjust the gear ratio of the servo for obtaining your desired number.
12. Every time when the instruction is executed, only one 90° arc can be drawn. It is not necessary that the arc has to be a precise arc, i.e. the numbers of output pulses in X and Y axes can be different.
13. There are no settings of start frequency and acceleration/deceleration time.
14. There is no limitation on the number of times using the instruction. However, assume CH1 or CH2 output is in use, the 1st group X/Y axis will not be able to output. If CH3 or CH4 output is in use, the 2nd group X/Y axis will not be able to output.
15. The settings of direction and resolution in the lower 16 bits of **S** can only be K0 ~ K3.
16. The settings of motion time in the high 16 bits of **S** can be slower than the the fastest suggested time but shall not be faster than the fastest suggested time.
17. The fastest suggested time for the arc interpolation:

Segments	Max. target position (pulse)	Fastest suggested set time (unit:100ms)
Average resolution	100 ~ 10,000	1
	10,001 ~ 19,999	2
	:	:
	Less than 1,000,000	Less than 100
Higher resolution	1,000 ~ 20,000	2
	20,000 ~ 29,999	3
	:	:
	Less than 10,000,000	Less than 200

Program Example 1:

1. Draw an ellipse as the figure below.



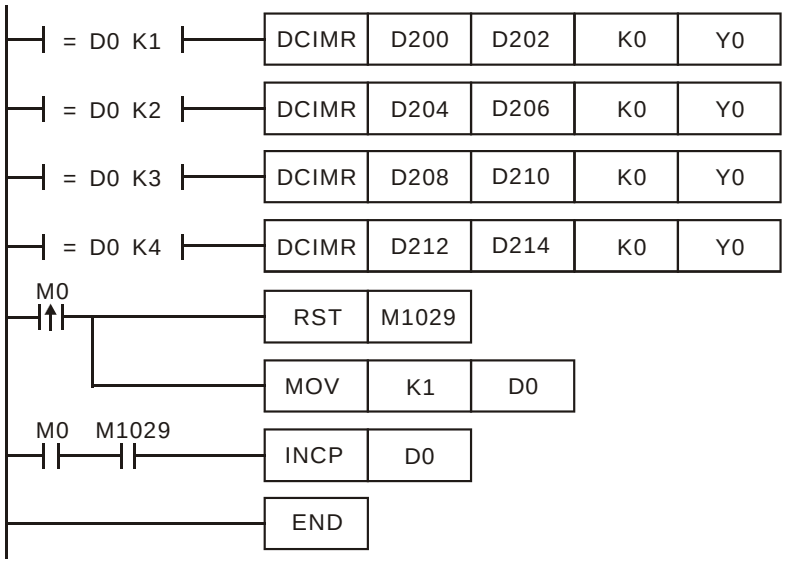
2. Steps:

a) Set the four coordinates (0,0), (1600, 2200), (3200, 0), (1600, -2200) (as the figure above). Calculate the relative coordinates of the four points and obtain (1600, 2200), (1600, -2200), (-1600, -2200), and (-1600, 2200). Place them in the 32-bit (D200, D202), (D204, D206), (D208, D210), (D212, D214).

b) Select “draw clockwise arc” and “average resolution” (**S** = K0).

c) Write program codes as follows.

d) PLC RUN. Set M0 as On and start the drawing of the ellipse.

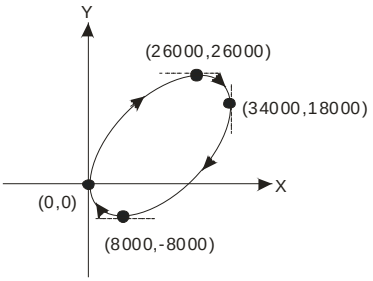


3. Motion explanation:

   When PLC RUN and M0 = On, PLC will start the drawing of the first segment of the arc. D0 will plus 1 whenever a segment of arc is completed and the second segment of the arc will start to execute automatically. The same motion will keep executing until the fourth segment of arc is completed.

Program Example 2:

1. Draw a tilted ellipse as the figure below.



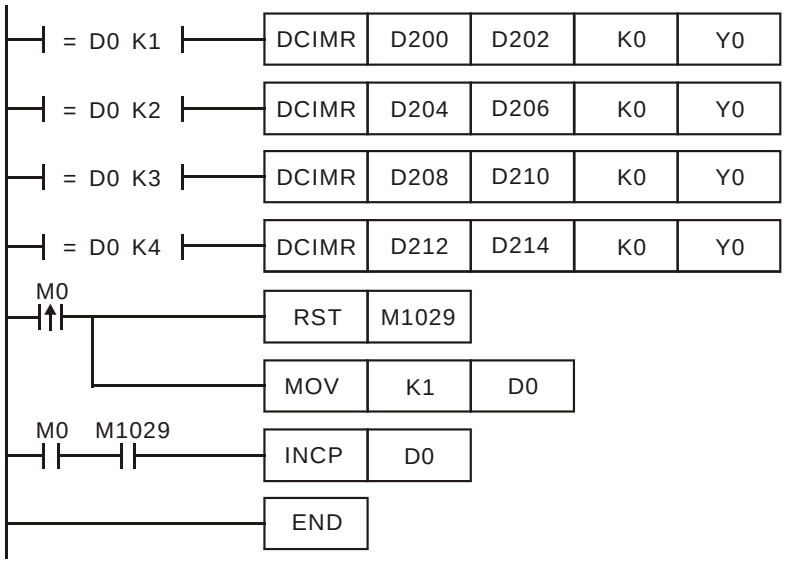
2. Steps:

a) Find the max. and min. coordinates on X and Y axes (0, 0), (26000, 26000), (34000, 18000), (8000, -8000) (as the figure above). Calculate the relative coordinates of the four points and obtain (26000, 26000)、(8000, -8000)、(-26000, -26000), (-8000, 8000). Place them respectively in the 32-bit (D200, D202), (D204, D206), (D208, D210) and (D212, D214).

b) Select “draw clockwise arc” and “average resolution” (**S** = K0).

c) Select DCIMR instruction for drawing arc and write program codes as follows.

d) PLC RUN. Set M0 as On and start the drawing of the ellipse.



3. Motion explanation:

   When PLC RUN and M0 = On, PLC will start the drawing of the first segment of the arc. D0 will plus 1 whenever a segment of arc is completed and the second segment of the arc will start to execute automatically. The same motion will keep executing until the fourth segment of arc is completed.

API	Mnemonic			Operands	Function
194	D	CIMA		S_1 S_2 S D	2-Axis Absolute Position Arc Interpolation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DCIMA: 17 steps
S_1					*	*							*			
S_2					*	*							*			
S													*			
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Number of output pulses of X axis **S_2**: Number of output pulses of Y axis **S**: Parameter setting **D**: Pulse output device

Explanations:

1. Flags: M1029, M1030, M1334, M1335. See remarks of API 191 DPPMR for more details.
2. This instruction only supports EH3/SV2 series MPU. In terms of pulse output methods, this instructin only supports “pulse + direction” mode.
3. **S_1** and **S_2** are the designated (absolute designation) number of output pulses in X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of the number is -2,147,483,648 ~ +2,147,483,647. When **S_1** and **S_2** are larger than pulse present value registers CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word), and CH3 (D1378 high word, D1377 low word), the output direction will be positive and direction signals Y1, Y3, Y5, Y7 will be On. When **S_1** and **S_2** are less than pulse present value registers, the output direction will be negative and direction signals Y1, Y3, Y5, Y7 will be Off.
4. The lower 16 bits of **S** (settings of direction and resolution): K0 refers to clockwise 10-segment (average resolution) output; K2 refers to clockwise 20-segment (higher resolution) output and a 90° arc can be drawn (see figure 1 and 2). K1 refers to counterclockwise 10-segment (average resolution) output; K3 refers to counterclockwise 20-segment (higher resolution) output and a 90° arc can be drawn (see figure 3 and 4).
5. The higher 16 bits of **S** (settings of motion time): K0 refers to 0.1 second. The setting range for average resolution is K1 ~ K100 (0.1 sec. ~ 10 secs.), for higher resolution is K2 ~ K200 (0.2 sec. ~ 20 secs.) This instruction is restricted by the maximum pulse output frequency; therefore when the set time goes faster than the actual output time, the set time will be automatically modified.

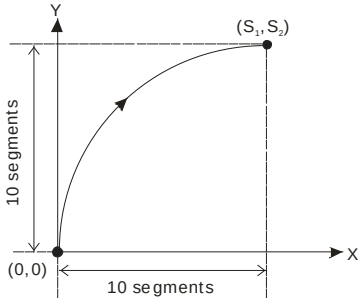


Figure 1

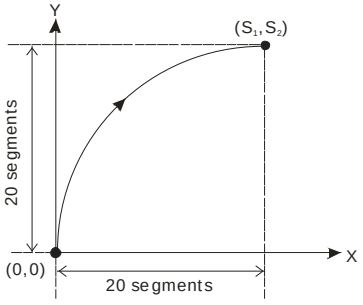


Figure 2

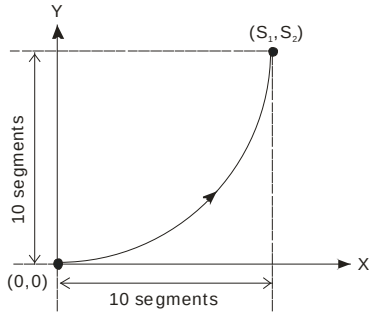


Figure 3

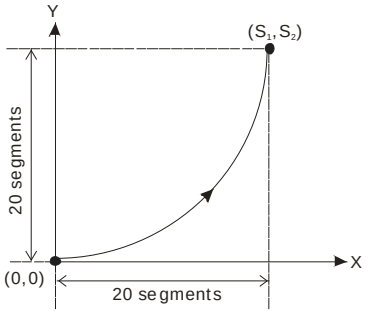


Figure 4

6. **D** can designate Y0 and Y4.
   When Y0 is designated:
   Y0 refers to 1st group X-axis pulse output device.
   Y1 refers to 1st group X-axis direction signal.
   Y2 refers to 1st group Y-axis pulse output device.
   Y3 refers to 1st group Y-axis direction signal.
   When Y4 is designated:
   Y4 refers to 2nd group X-axis pulse output device.
   Y5 refers to 2nd group X-axis direction signal.
   Y6 refers to 2nd group Y-axis pulse output device.
   Y7 refers to 2nd group Y-axis direction signal.
   When direction signal outputs, Off will not occur immediately after the pulse output is over. Direction signal will turn Off when the drive contact is Off. 24SV2 does NOT support Y10 and Y12.
7. Draw four 90° arcs.
8. When the direction signal is On, the direction is positive. When the direction signal is Off, the direction is negative. When **S** is set as K0, K2, the arcs will be clockwise (see figure 5). When **S** is set as K1, K3, the arcs will be counterclockwise (see figure 6).

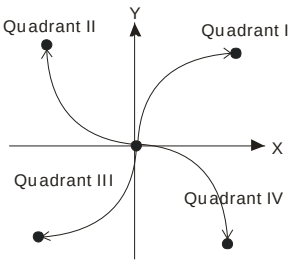


Figure 5

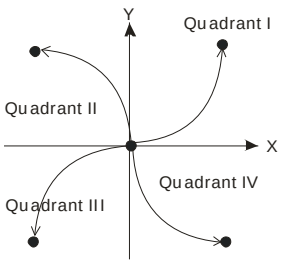


Figure 6

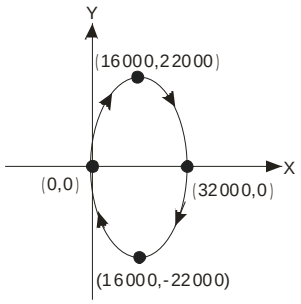
9. When the 2-axis motion is being executed in 10 segments (of average resolution), the operation time of the instruction when the instruction is first enabled is approximately 5ms. The number of output pulses cannot be less than 100 and more than 1,000,000; otherwise, the instruction cannot be enabled.
10. When the 2-axis motion is being executed in 20 segments (of high resolution), the operation time of the instruction when the instruction is first enabled is approximately 10ms. The number of output pulses cannot be less than 1,000 and more than 10,000,000; otherwise, the instruction cannot be enabled.

11. If you wish the number of pulses in 10-segment or 20-segment motion to be off the range, you may adjust the gear ratio of the servo for obtaining your desired number.
12. Every time when the instruction is executed, only one 90° arc can be drawn. It is not necessary that the arc has to be a precise arc, i.e. the numbers of output pulses in X and Y axes can be different.
13. There are no settings of start frequency and acceleration/deceleration time.
14. There is no limitation on the number of times using the instruction. However, assume CH1 or CH2 output is in use, the 1st group X/Y axis will not be able to output. If CH3 or CH4 output is in use, the 2nd group X/Y axis will not be able to output.
15. The settings of direction and resolution in the lower 16 bits of **S** can only be K0 ~ K3.
16. The settings of motion time in the high 16 bits of **S** can be slower than the the fastest suggested time but shall not be faster than the fastest suggested time.
17. The fastest suggested time for the arc interpolation:

Segments	Max. target position (pulse)	Fastest suggested set time (unit:100ms)
Average resolution	100 ~ 10,000	1
	10,001 ~ 19,999	2
	:	:
	Less than 1,000,000	Less than 100
Higher resolution	1,000 ~ 20,000	2
	20,000 ~ 29,999	3
	:	:
	Less than 10,000,000	Less than 200

Program Example 1:

1. Draw an ellipse as the figure below.



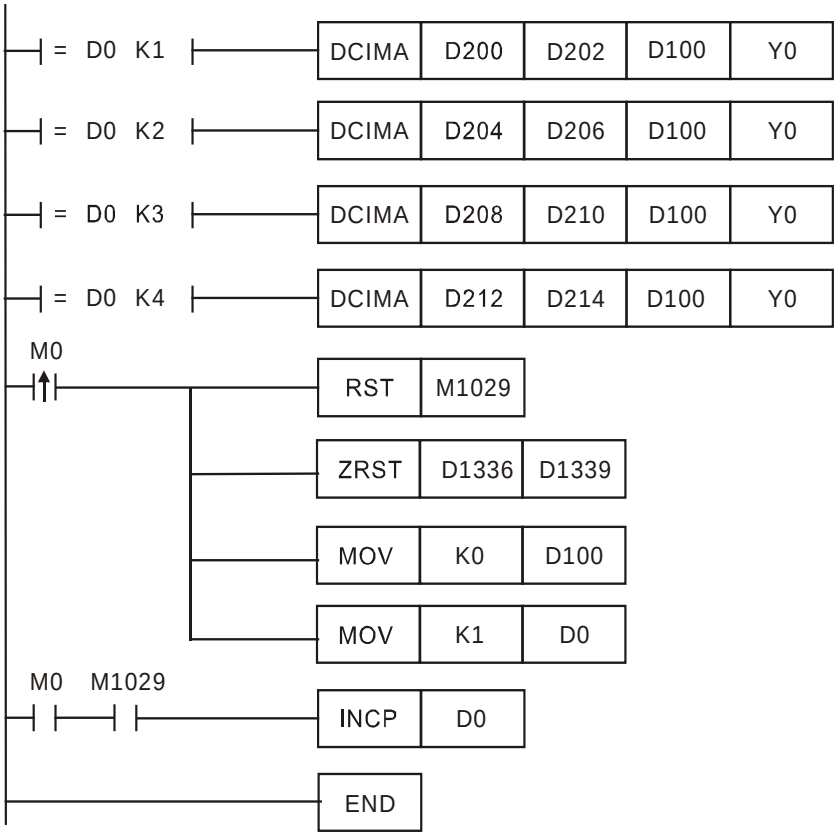
2. Steps:

a) Set the four coordinates (0, 0), (16000, 22000), (32000, 0), (16000, -22000) (as the figure above). Place them in the 32-bit (D200, D202), (D204, D206), (D208, D210), (D212, D214).

b) Select “draw clockwise arc” and “average resolution” (S =D100= K0).

c) Select DCIMA instruction for drawing arc and write program codes as follows.

d) PLC RUN. Set M0 as On and start the drawing of the ellipse.

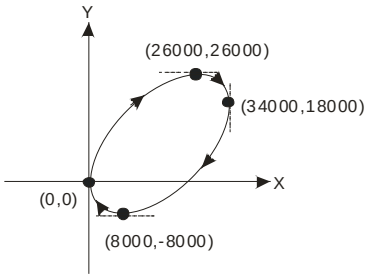


3. Motion explanation:

   When PLC RUN and M0 = On, PLC will start the drawing of the first segment of the arc. D0 will plus 1 whenever a segment of arc is completed and the second segment of the arc will start to execute automatically. The same motion will keep executing until the fourth segment of arc is completed.

Program Example 2:

1. Draw a tilted ellipse as the figure below.



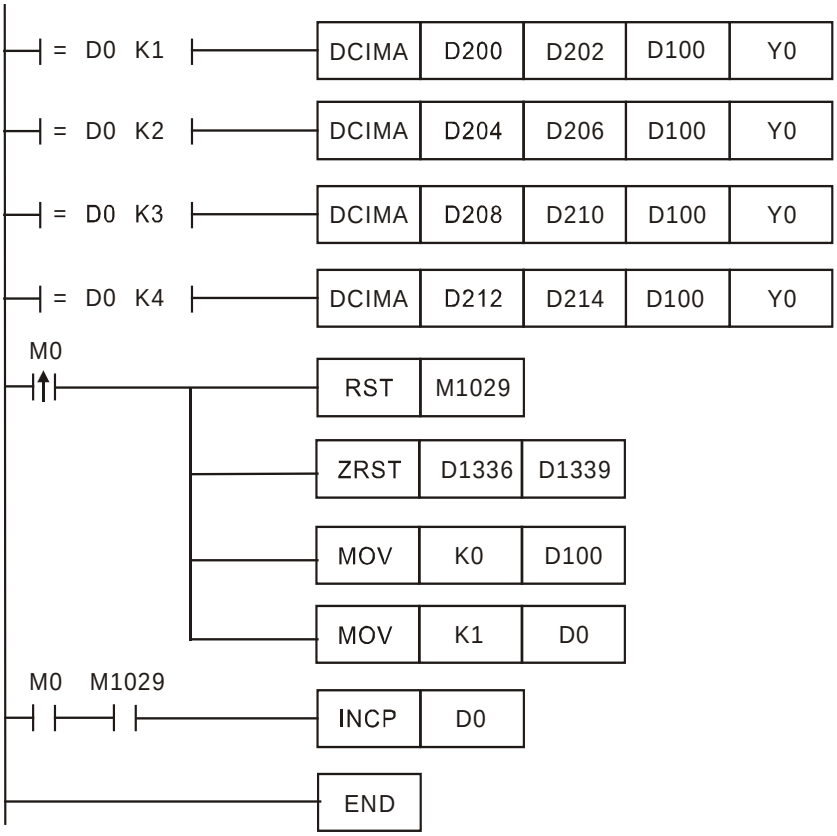
2. Steps:

a) Find the max. and min. coordinates on X and Y axes (0, 0), (26000, 26000), (34000, 18000), (8000, -8000) (as the figure above). Place them respectively in the 32-bit (D200, D202), (D204, D206), (D208, D210) and (D212, D214).

b) Select “draw clockwise arc” and “average resolution” (S =D100= K0).

c) Select DCIMA instruction for drawing arc and write program codes as follows.

d) PLC RUN. Set M0 as On and start the drawing of the ellipse.



3. Motion explanation:

   When PLC RUN and M0 = On, PLC will start the drawing of the first segment of the arc. D0 will plus 1 whenever a segment of arc is completed and the second segment of the arc will start to execute automatically. The same motion will keep executing until the fourth segment of arc is completed.

API	Mnemonic	Operands	Function
195	D PTPO	(S1) (S2) (D)	Single-Axis Pulse Output by Table

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DPTPO: 13 steps
S_2													*			
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source start device **S_2**: Number of segments **D**: Pulse output device

Explanations:

1. Flags: M1029, M1030, M1334, M1335. See remarks for more details.
2. This instruction only supports EH3/SV2 series MPU.
3. According to the value of **S_2** + 0, every segment consecutively occupy four register D. (**S_1** + 0) refers to output frequency. (**S_1** + 2) refers to the number of output pulses.
4. When the output frequency of **S_1** is less than 1, PLC will automatically modify it as 1. When the value is larger than 200,000kHz, PLC will automatically modify it as 200,000kHz.
5. **S_2** + 0: number of segments (range: 1 ~ 60). **S_2** + 1: number of segments being executed. Whenever the program scans to this instruction, the instruction will automatically update the segment No. that is currently being executed.

   D can only designate output devices Y0, Y2, Y4 and Y6 and can only perform pulse output control. For the pin for direction control, the user has to compile other programs to control. 24SV2 does NOT support Y10 and Y12.
6. This instruction does not offer acceleration and deceleration functions. Therefore, when the instruction is disabled, the output pulses will stop immediately.
7. In every program scan, each channel can only be executed by one instruction. However, there is no limitation on the number of times using this instruction.
8. When the instruction is being executed, the user is not allowed to update the frequency or number of the segments. Changes made will not be able to make changes in the actual output.

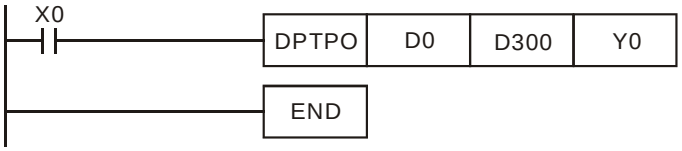
Program Example:

1. When X0 = On, the output will be operated according to the set frequency and number of pulses in every segment.

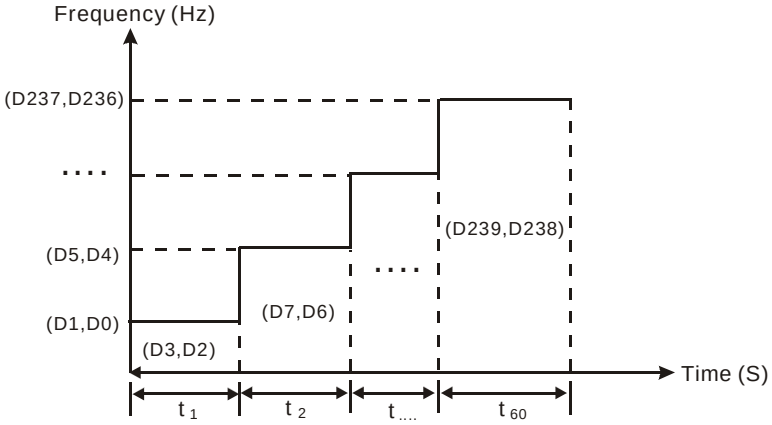
2. Format of the table:

S_2 = D300, number of segments (D300 = K60)	**S_1** = D0, frequency (**S_1** + 0)	**S_1** = D0, number of output pulses (**S_1** + 2)
K1 (1st segment)	D1, D0	D3, D2
K2 (2nd segment)	D5, D4	D7, D6
: :	: :	: :
K60 (60th segment)	D237, D236	D239, D238

3. Monitor the segment No. that is currently being executed in register D301.



4. The pulse output curve:



Remarks:

1. Flag explanations:

   M1029: On when CH0 (Y0) pulse output is completed.

   M1030: On when CH1 (Y2) pulse output is completed.

   M1036: On when CH2 (Y4) pulse output is completed.

   M1037: On when CH3 (Y6) pulse output is completed.

   M1334: When On, CH0 (Y0) pulse output will be forbidden.

   M1335: When On, CH1 (Y2) pulse output will be forbidden.

   M1520: When On, CH2 (Y4) pulse output will be forbidden.

   M1521: When On, CH3 (Y6) pulse output will be forbidden.

M1336: CH0 (Y0) pulse output indication flag

M1337: CH1 (Y2) pulse output indication flag

M1522: CH2 (Y4) pulse output indication flag

M1523: CH3 (Y6) pulse output indication flag

2. Special register explanations:

D1336, D1337: Pulse present value register of CH0 (Y0) (D1337 high word, D1336 low word)

D1338, D1339: Pulse present value register of CH1 (Y2) (D1339 high word, D1338 low word)

D1375, D1376: Pulse present value register of CH2 (Y4) (D1376 high word, D1375 low word)

D1377, D1378: Pulse present value register of CH3 (Y6) (D1378 high word, D1377 low word)

API	Mnemonic		Operands	Function
196	HST	P	(S)	High Speed Timer

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										HST, HSTP: 3 steps

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

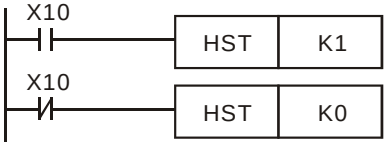
S: Condition to stop the startup of high speed timer

Explanations:

1. Range of **S**: S = K0 (H0), K1 (H1).
2. Flag: M1015
3. When **S** = 1, the high speed timer will be enabled and M1015 = On. The high speed timer starts to time and record the present value in D1015 (min. unit: 100us).
4. Timing range of D1015: K0 ~ K32,767. When the timing reaches K32,767, the next timing will restart from 0.
5. When **S** = 0, the high speed timer will be disabled and M1015 = Off. D1015 will stop the timing immediately.
6. When **S** is neither 1 nor 0, HST instruction will not be executed.

Program Example:

1. When X10 = On, M1015 will be On. The high speed timer will start to time and record the present value in D1015.
2. When X10 = Off, M1015 will be Off. The high speed timer will be shut down.



Remarks:

1. Flag explanations:
   M1015: high speed timer start-up flag
   D1015: high speed timer
2. EH3/SV2 series MPU do not use this instruction and use special M and special D directly for the timer.
   a) Special M and special D are only applicable when PLC RUN.
   b) When M1015 = On and PLC scans to END instruction, the high speed timer D1015 will be enabled. The minimum timing unit of D1015: 100us.
   c) Timing range of D1015: K0 ~ K32,767. When the timing reaches K32,767, the next timing will restart from K0.
   d) When M1015 = Off, D1015 will stop the timing when encountering END or HST instruction.

3. SX/EC3-8K series MPU do not use this instruction and use special M and special D directly for the timer.
   a) Special M and special D are applicable when PLC RUN or STOP.
   b) When M1015 = On, the high speed timer D1015 will be enabled. The minimum timing unit of D1015: 100us.
   c) Timing range of D1015: K0 ~ K32,767. When the timing reaches K32,767, the next timing will restart from K0.
   d) When M1015 = Off, D1015 will stop the timing immediately.

API	Mnemonic	Operands	Function
197	D CLLM	(S1) (S2) (S3) (D)	Close Loop Position Control

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DCLLM: 17 steps
S_1	*											*				
S_2					*	*							*			
S_3					*	*							*			
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Feedback source device **S_2**: Target number of feedbacks **S_3**: Target frequency of output **D**: Pulse output device

Explanations:

1. Flags: M1029, M1030, M1334, M1335. See remarks for more details.
2. This instruction only supports EH3/SV2 series MPU.
3. The corresponding interruption of **S_1**:

Source device	X0	X1	X2	X3	C241 ~ C254			
Corresponding outout	Y0	Y2	Y4	Y6	Y0	Y2	Y4	Y6
Interruption No.	I00□	I10□	I20□	I30□	I010	I020	I030	I040

□ = 1: rising-edige trigger; □ = 0: falling-edge trigger

   a) When **S_1** designates X as the input points and the pulse output reaches the set target number of feedbacks in **S_2**, the output will continue to operate by the frequency of the last segment until the interruption of X input points occurs.

   b) When **S_1** designates a high speed counter and the pulse output reaches the set target number of feedbacks in **S_2**, the output will continue to operate by the frequency of the last segment until the feedback pulses reaches the target number.

   c) **S_1** can be a high speed counter C or an external interruption X. If **S_1** is C, DCNT instruction should be first executed to enable the high-speed counting function and EI and I0x0 interruption service program to enable the high-speed interruption. If **S_1** is X, EI instruction and I0x0 interruption service program should be executed to enable the external interruption function.

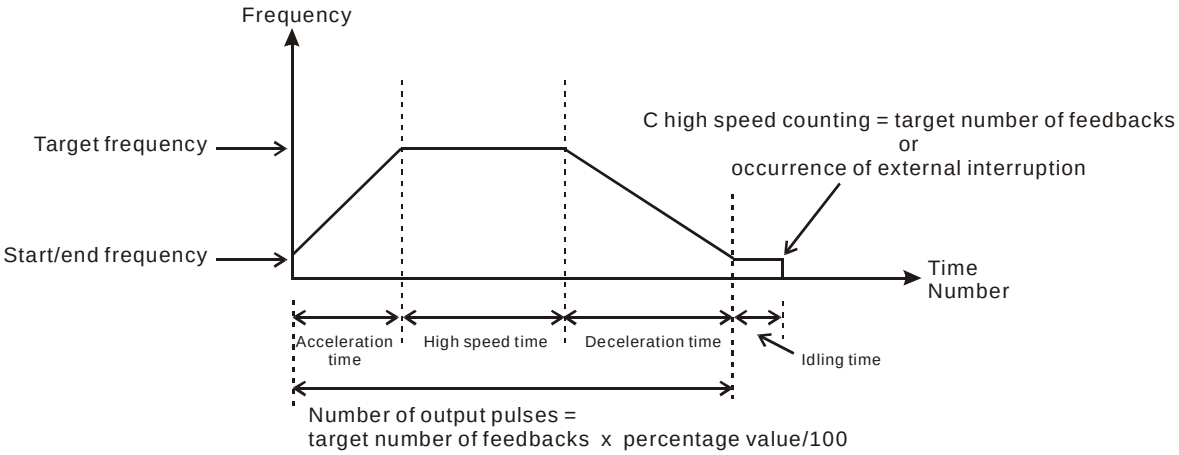
4. The range of **S_2**: -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in forward direction, the pulse present value registers CH0 (D1337 high word, D1336 low word), CH1 (D1339 high word, D1338 low word), CH2 (D1376 high word, D1375 low word) and CH3 (D1378 high word, D1377 low word) will increase. When in backward direction, the present value will decrease.
5. If **S_3** is lower than 10Hz, the output will operate at 10Hz; if **S_3** is higher than 200kHz, the output will operate at 200kHz.
6. **D** can only designate Y0, Y2, Y4 and Y6 and the direction signals repectively are Y1, Y3, Y5 and Y7. When there is a direction signal output, the direction signal will not be Off immediately after the pulse output is completed. The direction signal will be Off only when the drive contact is Off. 24SV2 does NOT support Y10 and

Y12.

7. D1340, D1352, D1379 and D1380 are the settings of start/end frequencies of CH0 ~ CH3. The minimun frequency is 10Hz and default is 200Hz.
8. D1343, D1353, D1381 and D1382 are the settings of the time of the first segment and the last deceleration segment of CH0 ~ CH3. The acceleration/deceleration time cannot be shorter than 10ms. The outptu will be operated in 10ms if the time set is shorter than 10ms or longer than 10,000ms. The dafault setting is 100ms.
9. D1131, D1132, D1478 and D1479 are the output/input ratio of the close loop control in CH0 ~ CH3. K1 refers to 1 output pulse out of the 100 target feedback input pulses; K200 refers to 200 output pulses out of the 100 target feedback input pulses. D1131, D1132, D1478 and D1479 are the numerators of the ratio (range: K1 ~ K10,000) and the denominator is fixed as K100 (the user does not have to enter a denominator).
10. M1305, M1306, M1532 and M1533 are the direction signal flags for CH0 ~ CH3. When $\mathbf{S_2}$ is a positive value, the output will be in forward direction and the flag will be Off. When $\mathbf{S_2}$ is a negative value, the output will be in backward direction and the flag will be On.

Close Loop Explanations:

1. Function: Immediately stop the high-speed pulse output according to the number of feedback pulses or external interruption signals.
2. The execution:

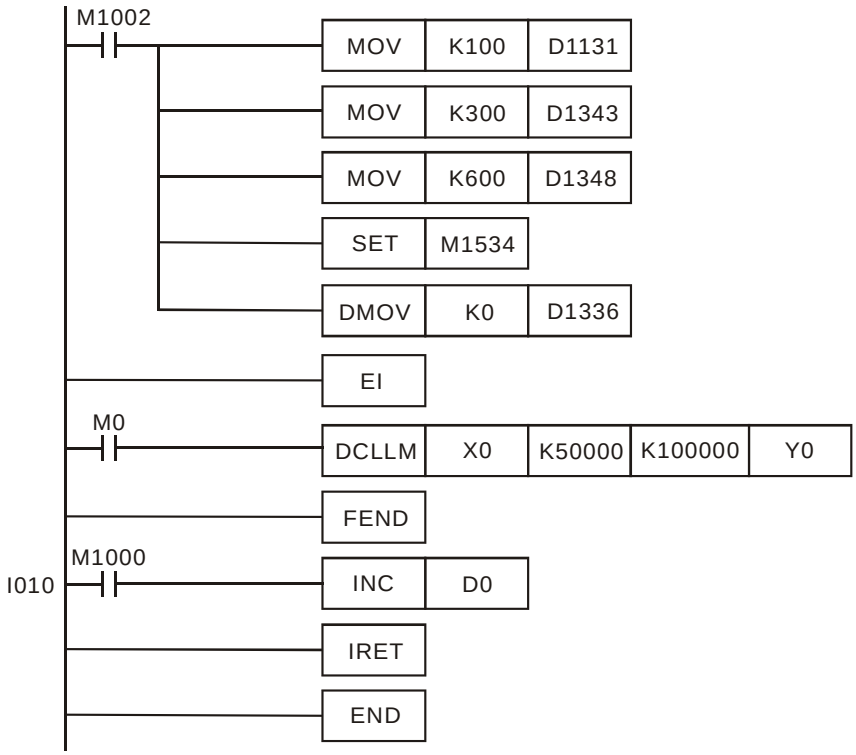


3. How to adjust the time for the completion of the positioning:
   a) The time for the completion of the positioning refers to the time for “acceleration + high speed + deceleration + idling” (see the figure above). For example, you can increase or decrease the entire number of output pulses by making adjustment on the percentage value and further increase or decrease the time required for the positioning.
   b) Among the four segments of time, only the idling time cannot be adjusted directly by the user. However, you can determine if the execution result is good or bad by the length of the idling time. In theory, a bit of idling left is the best result for a positioning.
   c) Owing to the close loop operation, the length of idling time will not be the same in every execution. Therefore, when the content in the special D for displaying the actial number of output pulses is smaller or larger than the calculated number of output pulses (taget number of feedbacks x percentage value/100),

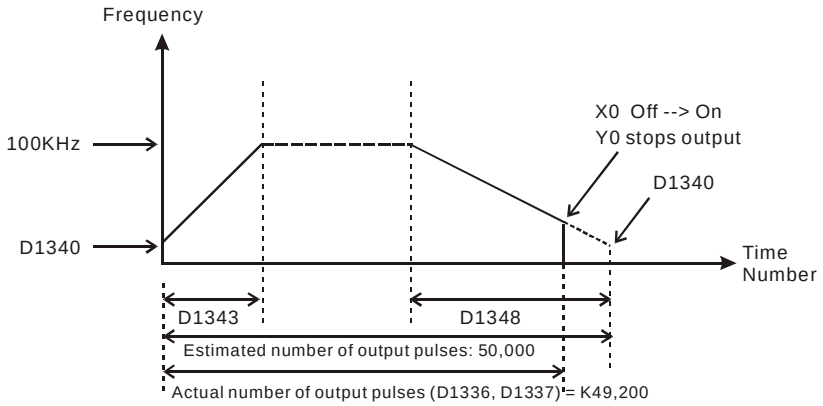
you can improve the situation by adjusting the percentage value, acceleration/decelartion time or target frequency.

Program Example:

1. Assume we adopt X0 as the external interruption, together with I001 (rising-edge trigger) interruption program; target number of feedbacks = 50,000; target frequency = 10kHz; Y0, Y1 (CH0) as output pulses; start/end frequency (D1340) = 200Hz; acceleration time (D1343) = 300ms; deceleration time (D1348) = 600ms; percentage value (D1131) = 100; current number of output pulses (D1336, D1337) = 0.
2. Write the program codes as follows:



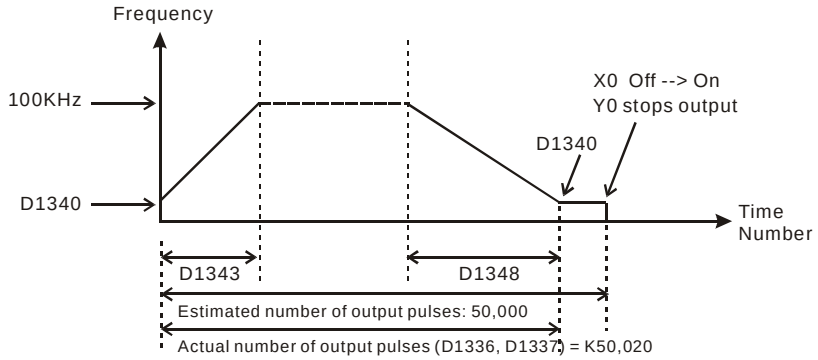
3. Assume the first execution result as:



4. Observe the result of the first execution:
   a) The actual output number 49,200 – estimated output number 50,000 = -800 (a negative value). A negative value indicates that the entire execution finishes earlier and has not completed yet.

b) Try to shorten the acceleration time (D1343) into 250ms and deceleration time (D1348) into 550ms.

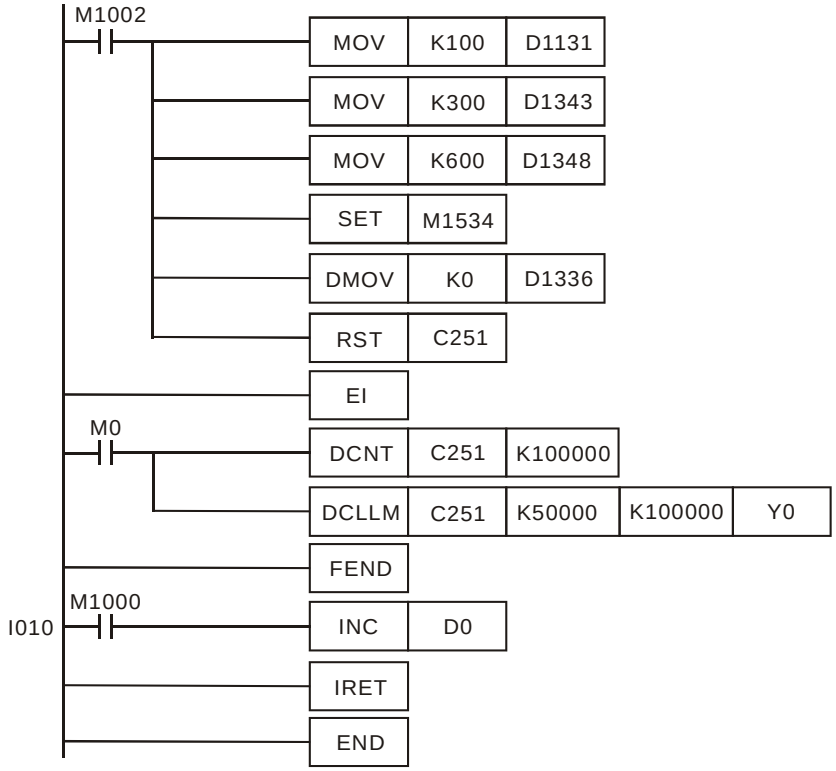
5. Obtain the result of the second execution:



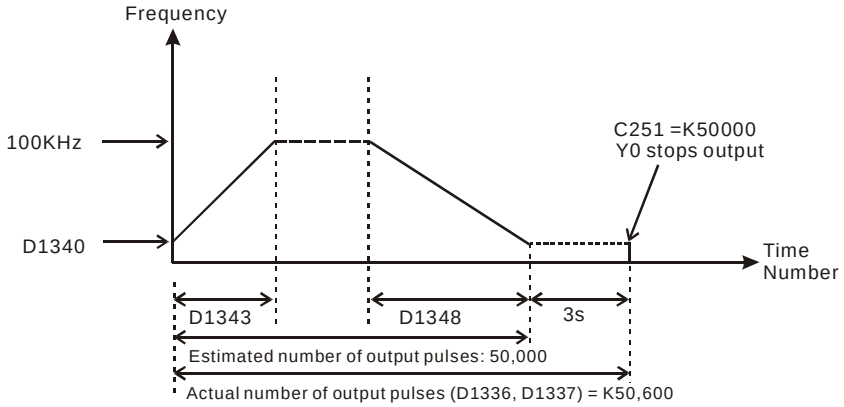
6. Observe the result of the second execution:
   a) The actual output number 50,020 – estimated output number 50,000 = 20
   b) 20 x (1/200Hz) = 100ms (idling time)
   c) 100ms is an appropriate value. Therefore, set the acceleration time as 250ms and deceleration time as 550ms to complete the design.

Program Example 2:

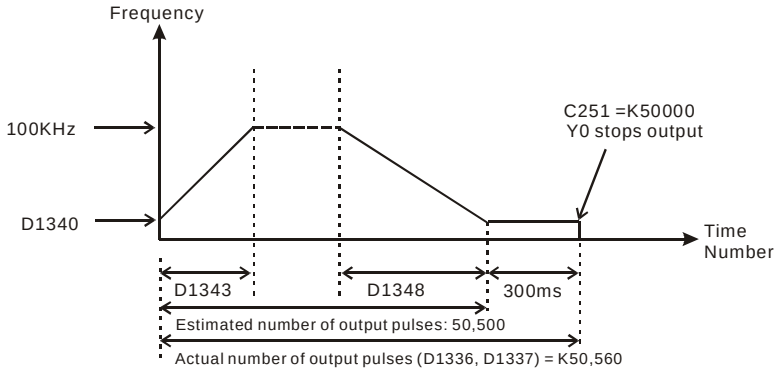
1. Assume the feedback of the encoder is an A/B phase input and we adopt C251 timing (we suggust you clear it to 0 before the execution); target number of feedbacks = 50,000; target output frequency = 100kHz; Y0, Y1 (CH0) as output pulses; start/end frequency (D1340) = 200Hz; acceleration time (D1343) = 300ms; deceleration time (D1348) = 600ms; precentage value (D1131) = 100; current number of output pulses (D1336, D1337) = 0.
2. Write the program codes as follows:



3. Assume the first execution result as:



4. Observe the result of the first execution:
   a) The actual output number 50,600 – estimated output number 50,000 = 600
   b) 600 x (1/200Hz) = 3s (idling time)
   c) 3 seconds are too long. Therefore, increase the percentage value (D1131) to K101.
5. Obatin the result of the second execution:



6. Observe the result of the second execution:
   a) The actual output number 50,560 – estimated output number 50,500 = 60
   b) 60 x (1/200Hz) = 300ms (idling time)
   c) 300ms is an appropriate value. Therefore, set the percentage value (D1131) as K101 to complete the design.

Remarks:

1. Flag explanations:

M1010:	When On, CH0, CH1, CH2 and CH3 will output pulses when encountering END instruction. Off when the output starts.
M1029:	On when CH0 pulse output is completed.
M1030:	On when CH1 pulse output is completed.
M1036:	On when CH2 pulse output is completed.
M1037:	On when CH3 pulse output is completed.
M1257:	The acceleration/deceleration slope of the high-speed pulse output is an S curve.

M1334: When On, CH0 pulse output will be forbidden.

M1335: When On, CH1 pulse output will be forbidden.

M1520: When On, CH2 pulse output will be forbidden.

M1521: When On, CH3 pulse output will be forbidden.

M1336: CH0 pulse output indication flag

M1337: CH1 pulse output indication flag

M1522: CH2 pulse output indication flag

M1523: CH3 pulse output indication flag

M1305: CH0 direction signal flag

M1306: CH1 direction signal flag

M1532: CH2 direction signal flag

M1533: CH3 direction signal flag

M1534: Deceleration time of CH0 setup flag (must used with D1348)

M1535: Deceleration time of CH1 setup flag (must used with D1349)

M1536: Deceleration time of CH2 setup flag (must used with D1350)

M1537: Deceleration time of CH3 setup flag (must used with D1351)

2. Special register explanations:

D1127: The number of pulses in the acceleration section in the position instruction (low word)

D1128: The number of pulses in the acceleration section in the position instruction (high word)

D1131: Close loop output/input ratio of CH0 (default: K100)

D1132: Close loop output/input ratio of CH1 (default: K100)

D1133: The number of pulses in the deceleration section in the position instruction (low word)

D1134: The number of pulses in the deceleration section in the position instruction (high word)

D1222: Time difference between the direction signal and pulse output of CH0

D1223: Time difference between the direction signal and pulse output of CH1

D1240: Low 16 bytes of the setting value for the end frequency of the high-speed output CH0 (available when the acceleration and deceleration are separate) (If D1240 < D1340, D1340 is adopted.)

D1241: High 16 bytes of the setting value for the end frequency of the high-speed output CH0 (available when the acceleration and deceleration are separate) (If D1240 < D1340, D1340 is adopted.)

D1244: Number of idle speed output from CH0 (> 0: Effective vale; <= 0: Continuous output)

D1245: Number of idle speed output from CH1 (> 0: Effective vale; <= 0: Continuous output)

D1246: Number of idle speed output from CH2 (> 0: Effective vale; <= 0: Continuous output)

D1247: Number of idle speed output from CH3 (> 0: Effective vale; <= 0: Continuous output)

D1383: Time difference between the direction signal and pulse output of CH2

D1384: Time difference between the direction signal and pulse output of CH3

D1336: Low word of the current number of output pulses of CH0

D1337: High word of the current number of output pulses of CH0

D1338: Low word of the current number of output pulses of CH1

D1339: High word of the current number of output pulses of CH1

D1375: Low word of the current number of output pulses of CH2

D1376: High word of the current number of output pulses of CH2

D1377: Low word of the current number of output pulses of CH3

D1378: High word of the current number of output pulses of CH3

D1340: Start/end frequency settings of CH0 (default: K200)

D1352: Start/end frequency settings of CH1 (default: K200)

D1379: Start/end frequency settings of CH2 (default: K200)

D1380: Start/end frequency settings of CH3 (default: K200)

D1348: Deceleration time of CH0 pulse output when M1534 = On (default: K100)

D1349: Deceleration time of CH1 pulse output when M1535 = On (default: K100)

D1350: Deceleration time of CH2 pulse output when M1536 = On (default: K100)

D1351: Deceleration time of CH3 pulse output when M1537 = On (default: K100)

D1343: Acceleration/deceleration time of CH0 pulse output (default: K100)

D1353: Acceleration/deceleration time of CH1 pulse output (default: K100)

D1381: Acceleration/deceleration time of CH2 pulse output (default: K100)

D1382: Acceleration/deceleration time of CH3 pulse output (default: K100)

D1478: Close loop output/input ratio of CH2 (default: K100)

D1479: Close loop output/input ratio of CH3 (default: K100)

API	Mnemonic	Operands	Function
198	D VSPO	(S1) (S2) (S3) (D)	Variable speed pulse output

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DVSPO: 17 steps
S_1													*			
S_2					*	*							*			
S_3					*	*							*			
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Target frequency of output **S_2**: Target number of pulses **S_3**: Gap time and gap frequency

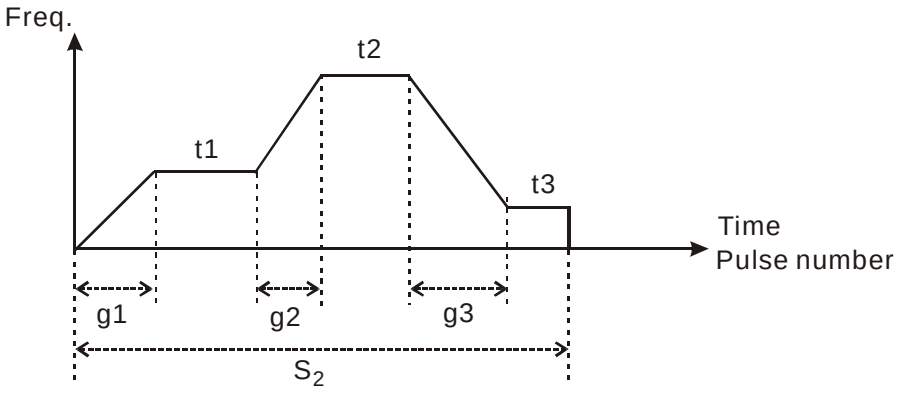
D: Pulse output device (EH2/SV2 supports Y0, Y2, Y4, and Y6.) (SX supports Y0.)

Explanations:

1. The instruciton only supports EH2 V2.0, SX V3.0. and above. It also supports EH3 and SV2.
2. Max frequency for **S_1**: 200kHz. (The maximum frequency that SX V3.0 and above support is 32767Hz.) Target frequency can be modified during the execution of instruction. When **S_1** is modified, VSPO will ramp up/down to the target frequency according to the ramp-up gap time and gap frequency set in **S_3**.
3. **S_2** target number of pulses is valid only when the instruction is executed first time. **S_2** can NOT be modified during the execution of instruction. **S_2** can be a negative value. When target number of pulses are specified with 0, PLC will perform continuous output and the special D shows the current value that is counting and going in the forward direction but that does NOT include any control over the output point direciton.
4. In an EH3/SV2 series PLC, **S_3** occupies 2 consecutive 16-bit devices. **S_3**+0 stores the gap frequency **S_3**+1 stores the gap time. Parameter setting can be modified during the execution of instruction. Set range for **S_3**+0: 1Hz ~ 32767Hz; set range for **S_3**+0: 1ms ~ 32767ms. If a setting value exceeds the available range, the PLC will take the maximum or the minimum value.
5. In an SX series PLC, the gap frequency in **S_3**+0 is in the range of 6Hz to 32767Hz, and the gap time in **S_3**+1 is in the range of 1ms to 80ms. If a setting value exceeds the available range, the PLC will take the maximum or the minimum value.
6. **D** pulse output device supports Y0, Y2, Y4 and Y6. Y1, Y3, Y5 and Y7 are corresponding output direction. The forward direction is On. 24SV2 does NOT support Y10 and Y12.
7. Parameters set in **S_3** can only be modified while modifying the value in **S_1**. When target frequency is set as 0, PLC will ramp down to stop according to parameters set in **S_3**. When the output is stopped, PLC will enable the flags indicating pause status (CH0: M1538; CH1: M1539; CH2: M1540; CH3: M1541). If target frequency other than 0 is specified again, pulse output will ramp up to target frequency and operates untill target number of pulses are completed.

Function Explanations:

Pulse output diagram:



1. Definitions:

   t1 → target frequency of 1st shift

   t2 → target frequency of 2nd shift

   t3 → target frequency of 3rd shift

   g1 → ramp-up time of 1st shift

   g2 → ramp-up time of 2nd shift

   g3 → ramp-down time of 3rd shift

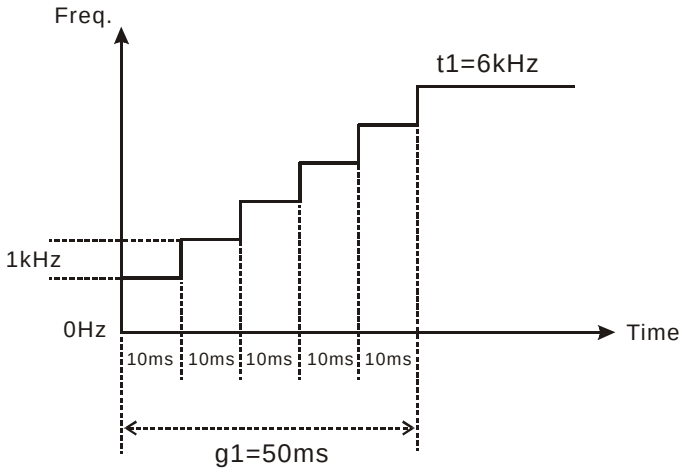
   S_2 → total output pulses

2. Explanations on each shift:

   ◆ 1st shift:

   Assume t1 = 6kHz, gap freqency = 1kHz, gap time = 10ms

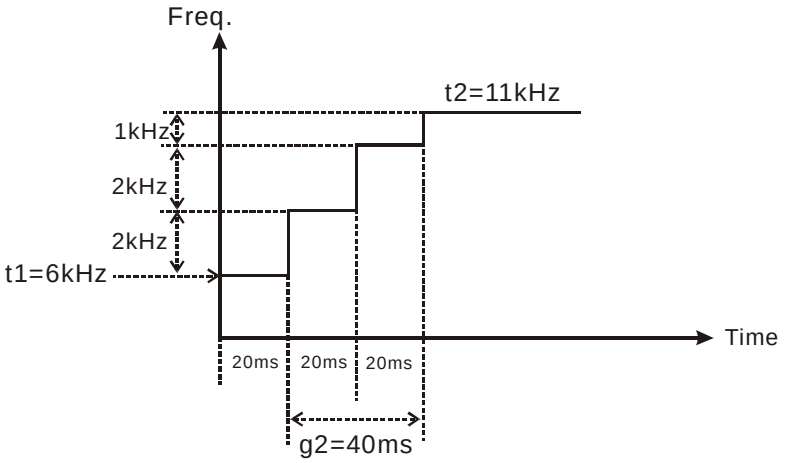
   Ramp-up steps of 1st shift:



◆ 2nd shift:

Assume t2 = 11kHz, internal frequency = 2kHz, gap time = 20ms

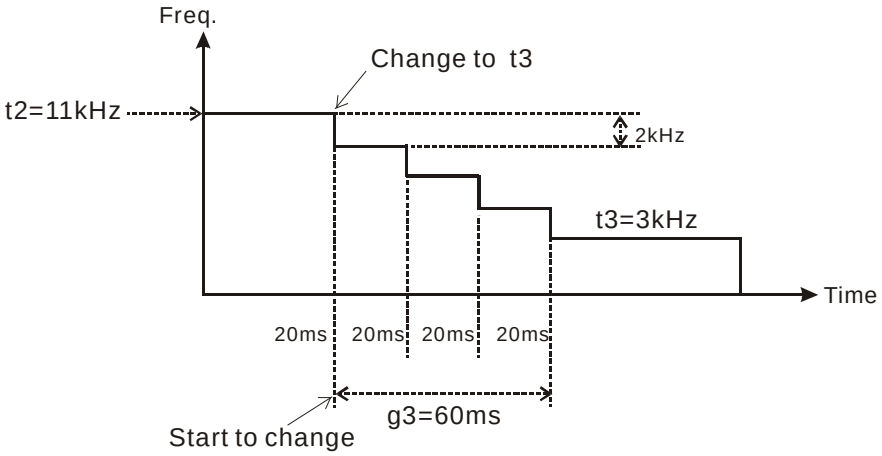
Ramp-up steps of 2nd shift:



◆ 3rd shift:

Assume t3 = 3kHz, gap frequency = 2kHz, gap time = 20ms

Ramp-down steps of 3rd shift:



◆ For program examples please refer to API 199

Points to note:

1. Associated flags:

M1029:	CH0 pulse output execution is completed
M1030:	CH1 pulse output execution is completed
M1036:	CH2 pulse output execution is completed
M1037:	CH3 pulse output execution is completed
M1538:	Indicating pause status of CH0
M1539:	Indicating pause status of CH1
M1540:	Indicating pause status of CH2
M1541:	Indicating pause status of CH0

M1542: CH0 executes the function that the constant speed output section reaches the target frequency.

M1544: CH1 executes the function that the constant speed output section reaches the target frequency.

M1546: CH2 executes the function that the constant speed output section reaches the target frequency.

M1548: CH3 executes the function that the constant speed output section reaches the target frequency.

M1543: CH0 executed the function that the constant speed output section reaches the target number.

M1545: CH1 executed the function that the constant speed output section reaches the target number.

M1547: CH2 executed the function that the constant speed output section reaches the target number.

M1549: CH3 executed the function that the constant speed output section reaches the target number.

M1528: Enabling the instruction DICF to execute the constant speed output section

M1529: Enabling the instruction DICF to execute the final output section

2. Special register explanations:

D1336: Low word of the present value of Y2 pulse output

D1337: High word of the present value of Y2 pulse output

D1338: Low word of the current number of output pulses from CH1

D1339: High word of the current number of output pulses from CH1

D1375: Low word of the current number of output pulses from CH2

D1376: High word of the current number of output pulses from CH2

D1377: Low word of the current number of output pulses from CH3

D1378: High word of the current number of output pulses from CH3

API	Mnemonic		Operands	Function
199	D	ICF	(S1) (S2) (D)	Immediately change frequency

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DICF: 13 steps
S_2					*	*							*			
D		*														

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

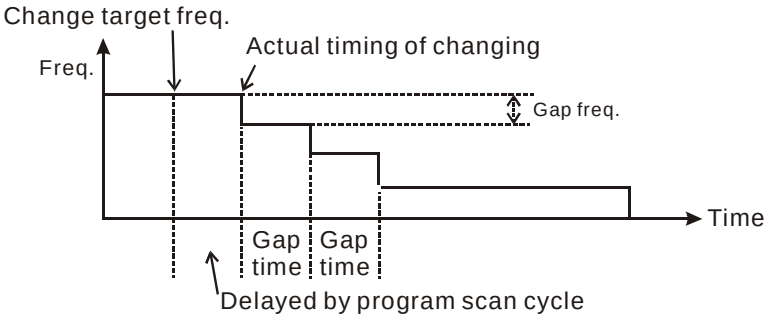
S_1: Target frequency to be changed **S_2**: Gap time and gap frequency **D**: Pulse output device (EH3/SV2 supports Y0, Y2, Y4, and Y6.) (SX supports Y0.)

Explanations:

1. The instruciton supports EH2 V2.0, SX V3.0, and above. It also supports EH3 and SV2.
2. Max frequency for **S_1**: 200kHz. (The maxumum freuency that SX V3.0 and above suppors is 32767Hz.) When ICF instruction executes, frequecy changing will start immediately with ramp-up/down process.
3. ICF instruction has to be executed after the execution of DVSPO or DPLSY instructions. When the instruction is used together with DVSPO, operands **S_1**, **S_2**, **D** of DICF has to be assigned the same device with **S_1**, **S_3**, **D** of DVSPO. When the instruction is used with DPLSY, operands **S_1** and **D** has to be assigned the same device with **S_1** and **D** of DPLSY**.**
4. If ICF instruction is used with DPLSY instruction, operand **S_2** is invalid.
5. When ICF instruction is used with DVSPO instruction, parameter setting of **S_2** functions the same as **S_3** in DVSPO instruction, specifying the gap time and gap frequency of ramp-up/down process.
6. The instruction is suggested to be applied in interrupt subroutines for obtaining the better response time and execution results
7. For associated flags and registers, please refer to **Points to note** of API 198 DVSPO instruction.

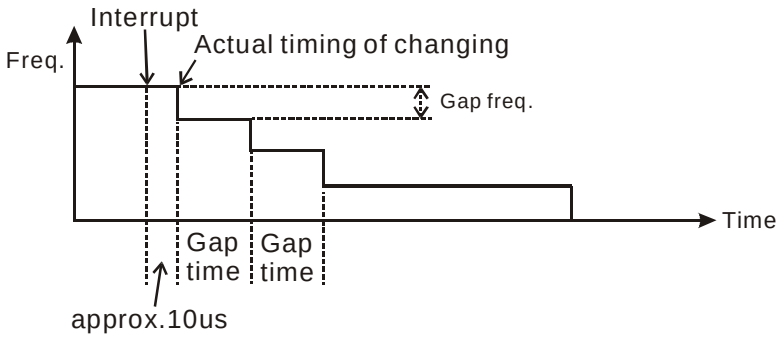
Function Explanations:

1. If users change the target frequency by using DVSPO instruction, the actual changing timing will be delayed due to the program scan time and the gap time as below.



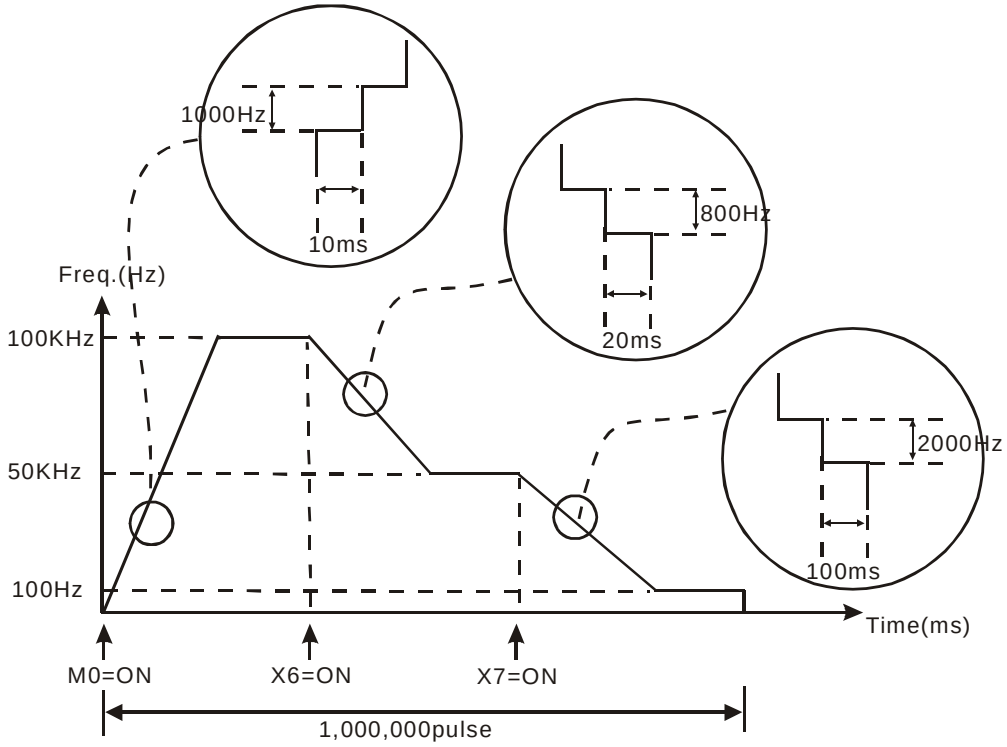
2. If users change the target frequency by applying DICF instruction in insterupt subroutines, the actual changing timing will be executed immediately with only an approx. 10us delay (execution time of DICF instruction).

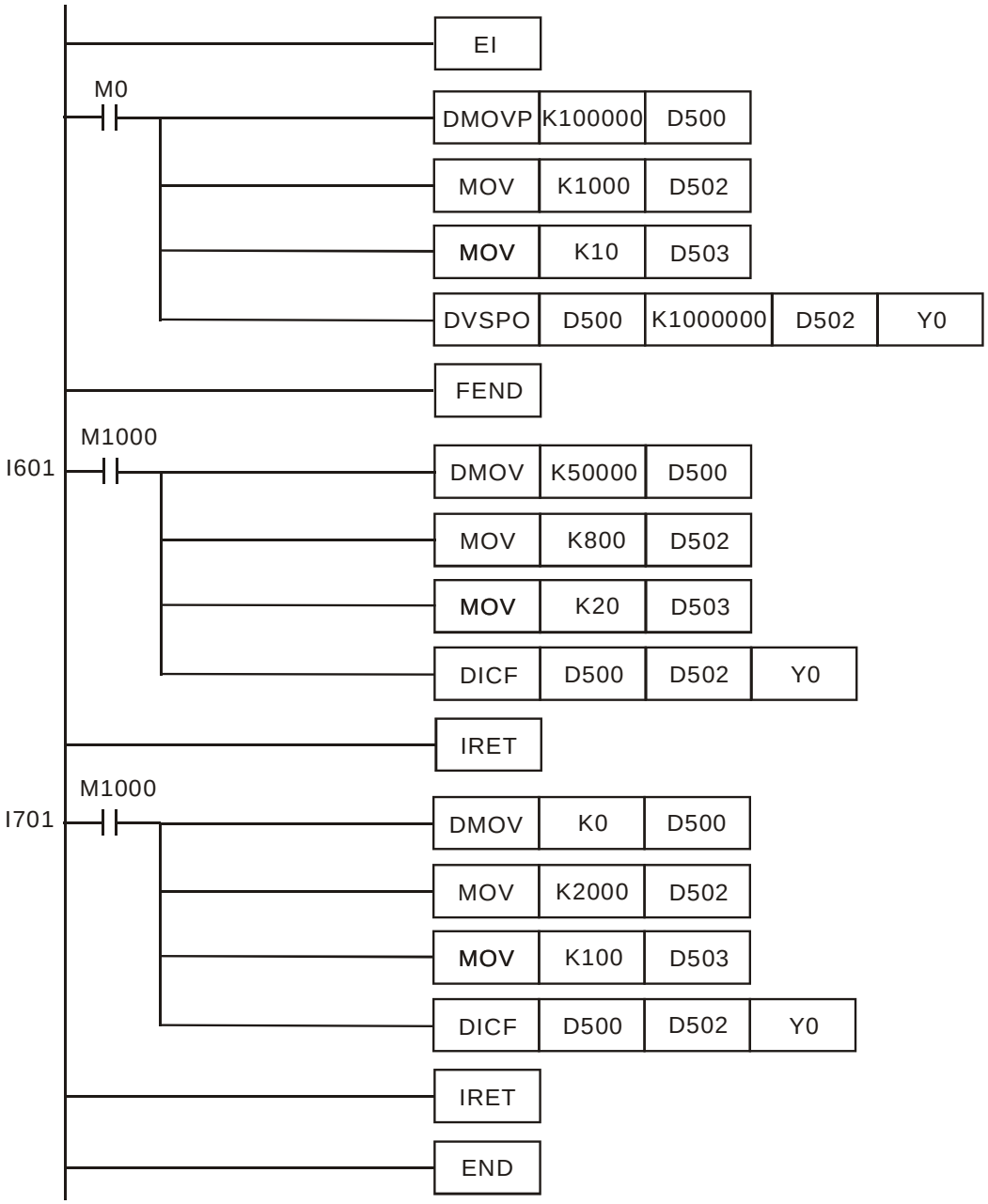
The timing diagram is as below:



Program Example:

1. When M0 = ON, pulse output ramps up to 100kHz. Total shifts: 100, Gap frequency: 1000Hz, Gap time: 10ms. Calculation of total shifts: (100,000 – 0) ÷ 1000 = 100.
2. When X6 external interrupt executes, target frequency is changed and ramp down to 50kHz immediately. Total shifts: 150, Gap frequency: 800Hz, Gap time: 20ms. Calculation of total shifts: (100,000 – 50,000) ÷ 800 = 125
3. When X7 external interrupt executes, target frequency is changed and ramp down to 100Hz immediately. Total shifts: 25, Gap frequency: 2000Hz, Gap time: 100ms. Calculation of total shifts: (50,000 – 100) ÷ 2000 = 25.
4. When pulse output reaches 100Hz, the frequency is kept constant and pulse output stops when 1,000,000 pulses is completed.




EI
M0
DMOVP K100000 D500
MOV K1000 D502
MOV K10 D503
DVSPO D500 K1000000 D502 Y0
FEND
M1000
I601
DMOV K50000 D500
MOV K800 D502
MOV K20 D503
DICF D500 D502 Y0
IRET
M1000
I701
DMOV K0 D500
MOV K2000 D502
MOV K100 D503
DICF D500 D502 Y0
IRET
END

DVP-PLC applicable to the application instruction. ES includes ES/EX/EC/EC3-8K (FW V8.60 or later) (EC3: FW V8.40 or previous version); SX (FW V3.00); EH3 includes EH3/SV2.

ES/EX/EC series MPU does not support pulse execution type instructions (P instruction).

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	16-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
Others	202	SCAL	–	✓	Proportional Value Calculation	✓	✓	✓	✓	9	–
	203	SCLP	–	✓	Parameter Proportional Value Calculation	✓	✓	✓	✓	9	13
	205	CMPT	DCMPT	✓	Compare table	–	–	–	✓	9	17
	207	CSFO	–	–	Catch speed and proportional output	–	–	–	✓	7	–
Communication	206	ASDRW	–	–	ASDA servo drive R/W	–	–	–	✓	7	–
	328	CANRS	–	–	User-defined CAN BUS communication sending and receiving	–	–	–	EH3-L/SV2	11	–
Contact Type Logic Operation	215	LD&	DLD&	–	S_1 & S_2	–	✓	✓	✓	5	9
	216	LD|	DLD|	–	S_1 | S_2	–	✓	✓	✓	5	9
	217	LD^	DLD^	–	S_1 ^ S_2	–	✓	✓	✓	5	9
	218	AND&	DAND&	–	S_1 & S_2	–	✓	✓	✓	5	9

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	16-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
	219	AND|	DAND|	–	S_1 | S_2	–	✓	✓	✓	5	9
	220	AND^	DAND^	–	S_1 ^ S_2	–	✓	✓	✓	5	9
	221	OR&	DOR&	–	S_1 & S_2	–	✓	✓	✓	5	9
	222	OR|	DOR|	–	S_1 | S_2	–	✓	✓	✓	5	9
	223	OR^	DOR^	–	S_1 ^ S_2	–	✓	✓	✓	5	9
Contact Type Comparison Instruction	224	LD=	DLD=	–	S_1 = S_2	✓	✓	✓	✓	5	9
	225	LD>	DLD>	–	S_1 > S_2	✓	✓	✓	✓	5	9
	226	LD<	DLD<	–	S_1 < S_2	✓	✓	✓	✓	5	9
	228	LD<>	DLD<>	–	S_1 ≠ S_2	✓	✓	✓	✓	5	9
	229	LD<=	DLD<=	–	S_1 ≦ S_2	✓	✓	✓	✓	5	9
	230	LD>=	DLD>=	–	S_1 ≧ S_2	✓	✓	✓	✓	5	9

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	16-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
	232	AND=	DAND=	–	$S_1 = S_2$	✓	✓	✓	✓	5	9
	233	AND>	DAND>	–	$S_1 > S_2$	✓	✓	✓	✓	5	9
	234	AND<	DAND<	–	$S_1 < S_2$	✓	✓	✓	✓	5	9
	236	AND<>	DAND<>	–	$S_1 \neq S_2$	✓	✓	✓	✓	5	9
	237	AND<=	DAND<=	–	$S_1 \leqq S_2$	✓	✓	✓	✓	5	9
	238	AND>=	DAND>=	–	$S_1 \geqq S_2$	✓	✓	✓	✓	5	9
	240	OR=	DOR=	–	$S_1 = S_2$	✓	✓	✓	✓	5	9
	241	OR>	DOR>	–	$S_1 > S_2$	✓	✓	✓	✓	5	9
	242	OR<	DOR<	–	$S_1 < S_2$	✓	✓	✓	✓	5	9
	244	OR<>	DOR<>	–	$S_1 \neq S_2$	✓	✓	✓	✓	5	9
	245	OR<=	DOR<=	–	$S_1 \leqq S_2$	✓	✓	✓	✓	5	9

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	16-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
	246	OR>=	DOR>=	–	$S_1 \geqq S_2$	✓	✓	✓	✓	5	9
Word Device Bit Instruction	266	BOUT	DBOUT	–	Output Specified Bit of a Word	–	✓	✓	✓	5	9
	267	BSET	DBSET	–	Set ON Specified Bit of a Word	–	✓	✓	✓	5	9
	268	BRST	DBRST	–	Reset Specified Bit of a Word	–	✓	✓	✓	5	9
	269	BLD	DBLD	–	Load NO Contact by Specified Bit	–	✓	✓	✓	5	9
	270	BLDI	DBLDI	–	Load NC Contact by Specified Bit	–	✓	✓	✓	5	9
	271	BAND	DBAND	–	Connect NO Contact in Series by Specified Bit	–	✓	✓	✓	5	9
	272	BANI	DBANI	–	Connect NC Contact in Series by Specified Bit	–	✓	✓	✓	5	9
	273	BOR	DBOR	–	Connect NO Contact in Parallel by Specified Bit	–	✓	✓	✓	5	9
	274	BORI	DBORI	–	Connect NC Contact in Parallel by Specified Bit	–	✓	✓	✓	5	9

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	16-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
Floating-point Contact Type Comparison Instruction	275	–	FLD=	–	$S_1 = S_2$	–	✓	✓	✓	–	9
	276	–	FLD>	–	$S_1 > S_2$	–	✓	✓	✓	–	9
	277	–	FLD<	–	$S_1 < S_2$	–	✓	✓	✓	–	9
	278	–	FLD<>	–	$S_1 \neq S_2$	–	✓	✓	✓	–	9
	279	–	FLD<=	–	$S_1 \leqq S_2$	–	✓	✓	✓	–	9
	280	–	FLD>=	–	$S_1 \geqq S_2$	–	✓	✓	✓	–	9
	281	–	FAND=	–	$S_1 = S_2$	–	✓	✓	✓	–	9
	282	–	FAND>	–	$S_1 > S_2$	–	✓	✓	✓	–	9
	283	–	FAND<	–	$S_1 < S_2$	–	✓	✓	✓	–	9
	284	–	FAND<>	–	$S_1 \neq S_2$	–	✓	✓	✓	–	9
	285	–	FAND<=	–	$S_1 \leqq S_2$	–	✓	✓	✓	–	9

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	16-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
	286	–	FAND>=	–	$S_1 \geqq S_2$	–	✓	✓	✓	–	9
	287	–	FOR=	–	$S_1 = S_2$	–	✓	✓	✓	–	9
	288	–	FOR>	–	$S_1 > S_2$	–	✓	✓	✓	–	9
	289	–	FOR<	–	$S_1 < S_2$	–	✓	✓	✓	–	9
	290	–	FOR<>	–	$S_1 \neq S_2$	–	✓	✓	✓	–	9
	291	–	FOR<=	–	$S_1 \leqq S_2$	–	✓	✓	✓	–	9
	292	–	FOR>=	–	$S_1 \geqq S_2$	–	✓	✓	✓	–	9
Floating-point Contact Type Comparison Instruction	296	LDZ>	DLDZ>	-	$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$	–	✓	✓	✓	5	9
	297	LDZ>=	DLDZ>=	-	$\lvert S_1 - S_2 \rvert \geqq \lvert S_3 \rvert$	–	✓	✓	✓	5	9
	298	LDZ<	DLDZ<	-	$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$	–	✓	✓	✓	5	9
	299	LDZ<=	DLDZ<=	-	$\lvert S_1 - S_2 \rvert \leqq \lvert S_3 \rvert$	–	✓	✓	✓	5	9

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	16-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
	300	LDZ=	DLDZ=	-	$\|S_1 - S_2\| = \|S_3\|$	–	✓	✓	✓	5	9
	301	LDZ<>	DLDZ<>	-	$\|S_1 - S_2\| \neq \|S_3\|$	–	✓	✓	✓	5	9
	302	ANDZ>	DANDZ>	-	$\|S_1 - S_2\| > \|S_3\|$	–	✓	✓	✓	5	9
	303	ANDZ>=	DANDZ>=	-	$\|S_1 - S_2\| \geqq \|S_3\|$	–	✓	✓	✓	5	9
	304	ANDZ<	DANDZ<	-	$\|S_1 - S_2\| < \|S_3\|$	–	✓	✓	✓	5	9
	305	ANDZ<=	DANDZ<=	-	$\|S_1 - S_2\| \leqq \|S_3\|$	–	✓	✓	✓	5	9
	306	ANDZ=	DANDZ=	-	$\|S_1 - S_2\| = \|S_3\|$	–	✓	✓	✓	5	9
	307	ANDZ<>	DANDZ<>	-	$\|S_1 - S_2\| \neq \|S_3\|$	–	✓	✓	✓	5	9
	308	ORZ>	DORZ>	-	$\|S_1 - S_2\| > \|S_3\|$	–	✓	✓	✓	5	9
	309	ORZ>=	DORZ>=	-	$\|S_1 - S_2\| \geqq \|S_3\|$	–	✓	✓	✓	5	9
	310	ORZ<	DORZ<	-	$\|S_1 - S_2\| < \|S_3\|$	–	✓	✓	✓	5	9

Category	API	Mnemonic		P instruction	Function	Applicable to				STEPS	
		16-bit	16-bit			ES	EC3-8K	SX	EH3	16-bit	32-bit
	311	ORZ<=	DORZ<=	-	$\lvert S_1 - S_2 \rvert \leqq \lvert S_3 \rvert$	–	✓	✓	✓	5	9
	312	ORZ=	DORZ=	-	$\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$	–	✓	✓	✓	5	9
	313	ORZ<>	DORZ<>	-	$\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$	–	✓	✓	✓	5	9

API	Mnemonic		Operands	Function
202	SCAL	P	S_1 S_2 S_3 D	Proportional Value Calculation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			SCAL, SCALP: 9 steps
S_2					*	*							*			
S_3					*	*							*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source value **S_2**: Slope **S_3**: Offset **D**: Destination device

Explanations:

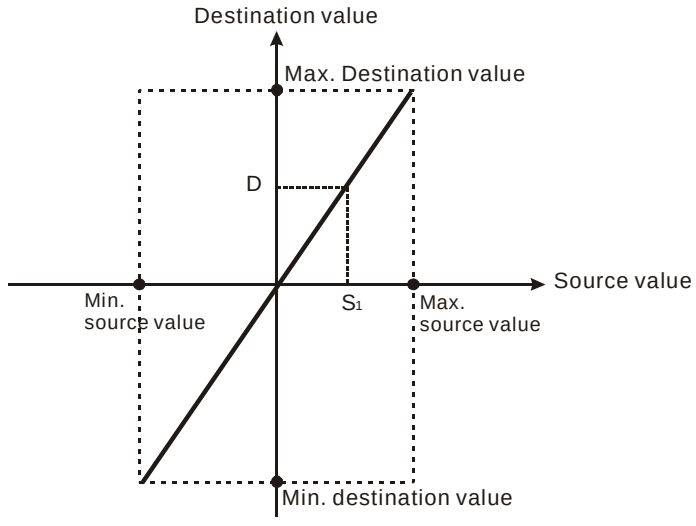
1. Range of **S_1**, **S_2**, **S_3**: -32,768 ~ 32,767
2. Unit of **S_2**: 0.001
3. See the specifications of each model for their range of use.
4. Operation equation in the instruction: **D** = (**S_1** × **S_2**) ÷ 1,000 + **S_3**.

   Users have to obtain **S_2** and **S_3** (decimals are rounded up into 16-bit integers) by using the slope and offset equations below.

   Slope equation: **S_2** = [(max. destination value – min. destination value) ÷ (max. source value – min. source value)] × 1,000

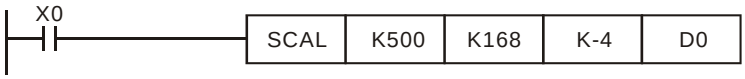
   Offset equation: **S_3** = min. destination value – min. source value × **S_2** ÷ 1,000

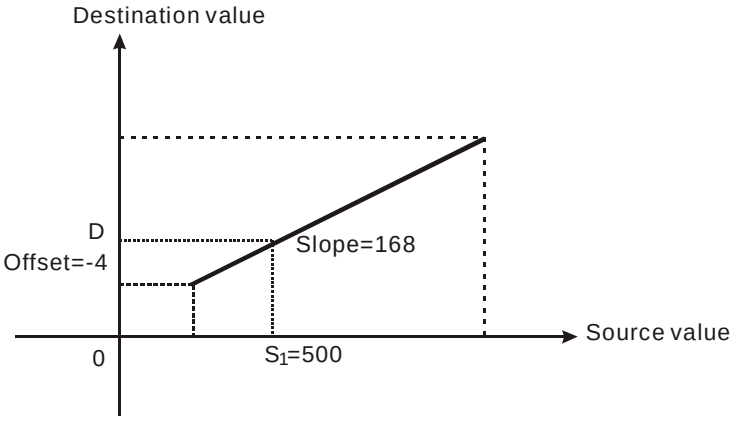
   The output curve is shown as the figure:



Program Example 1:

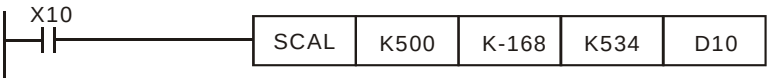
1. Assume **S_1** = 500, **S_2** = 168, **S_3** = -4. When X0 = On, SCAL instruction will be executed and obtain the proportional value at D0.
2. Equation: D0 = (500 × 168) ÷ 1,000 + (-4) = 80

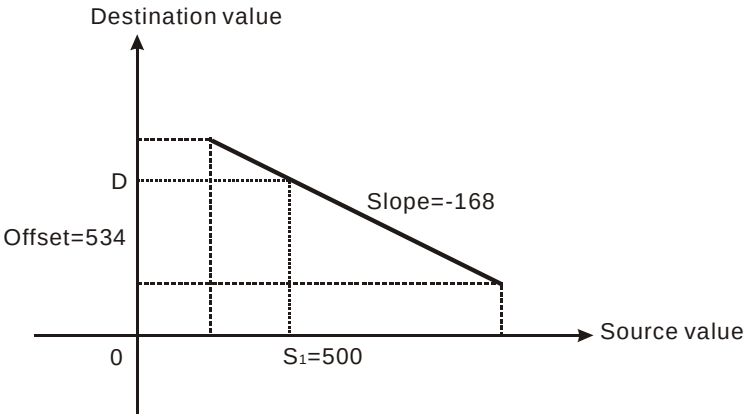




Program Example 2:

1. Assume **S_1** = 500, **S_2** = -168, **S_3** = 534. When X10 = On, SCAL instruction will be executed and obtain the proportional value at D10.
2. Equation: D0 = (500 × -168) ÷ 1,000 + 534 = 450





Remarks:

1. This instruction is applicable for known slope and offset. If slope and offset are unknown, use SCLP instruction for the calculation.
2. **S_2** has to be within the range -32,768 ~ 32,767. If **S_2** falls without the range, use SCLP instruction for the calculation.
3. When using the slope equation, please be aware that the max. source value must > min. source value, but it is not necessary that max. destination value > min. destination value.
4. If the value of **D** > 32,767, **D** = 32,767; if the value of **D** < -32,768, **D** = -32,768.

API	Mnemonic			Operands	Function
203	D	SCLP	P	(S1) (S2) (D)	Parameter Proportional Value Calculation

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			SCLP, SCLPP: 7 steps DSCLP, DSCLPP: 13 steps
S_2													*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source value **S_2**: Parameter **D**: Destination device

Explanations:

1. See the specifications of each model for the range of operands.
2. Flags: M1162 (decimal integer or binary floating point); M1162 = On -> Binary floating point
3. Settings of **S_2** for 16-bit instruction:

   S_2 occupies 4 consecutive devices in 16-bit instruction.

Device No.	Parameter	Range
S_2	Maximum source value	-32,768 ~ 32,767
S_2 + 1	Minimum source value	-32,768 ~ 32,767
S_2 + 2	Maximum destination value	-32,768 ~ 32,767
S_2 + 3	Minimum destination value	-32,768 ~ 32,767

4. Settings of **S_2** for 32-bit instruction:

   S_2 occupies 8 consecutive devices in 32-bit instruction.

Device No.	Parameter	Range	
		Integer	Floating point
S_2, **S_2** + 1	Maximum source value	-2,147,483,648 ~ 2,147,483,647	Range of 32-bit floating point
S_2 + 2, 3	Minimum source value		
S_2 + 4, 5	Maximum destination value		
S_2 + 6, 7	Minimum destination value		

5. Operation equation in the instruction: **D** = [(**S_1** – min. source value) × (max. destination value – min. destination value)] ÷ (max. source value – min. source value) + min. destination value
6. The operational relation between source value and destination value is as stated below:

   y = kx + b

   y= Destination value (**D**)

   k= Slope = (max. destination value – min. destination value) ÷ (max. source value – min. source value)

   x= Source value (**S_1**)

   b= Offset = Min. destination value – Min. source value × slope

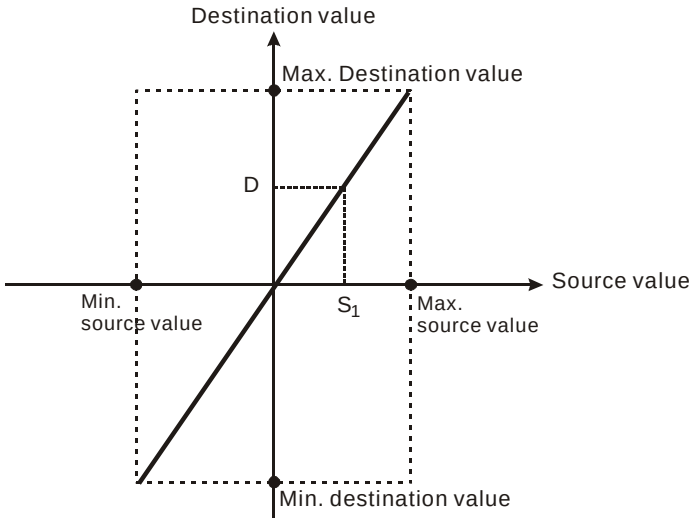
Bring all the parameters into equation y = kx + b and obtain the equation in the instruction:

y = kx + b = D = k $\mathbf{S_1}$ + b = slope ×$\mathbf{S_1}$ + offset = slope ×$\mathbf{S_1}$ + min. destination value – min. source value × slope = slope × ($\mathbf{S_1}$ – min. source value) + min. destination value = ($\mathbf{S_1}$ – min. source value) × (max. destination value – min. destination value) ÷ (max. source value – min. source value) + min. destination value

7. If **S1** > max. source value, **S1** = max. source value

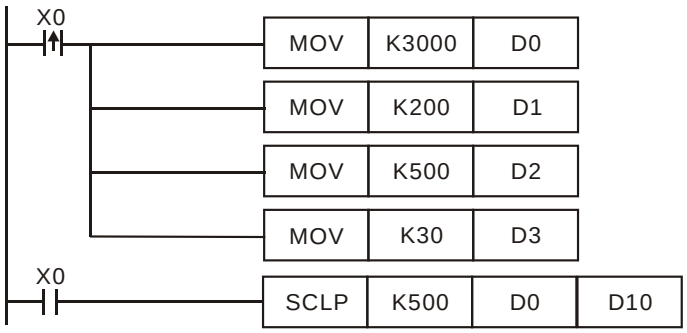
   If **S1** < min. source value, **S1** = min. source value

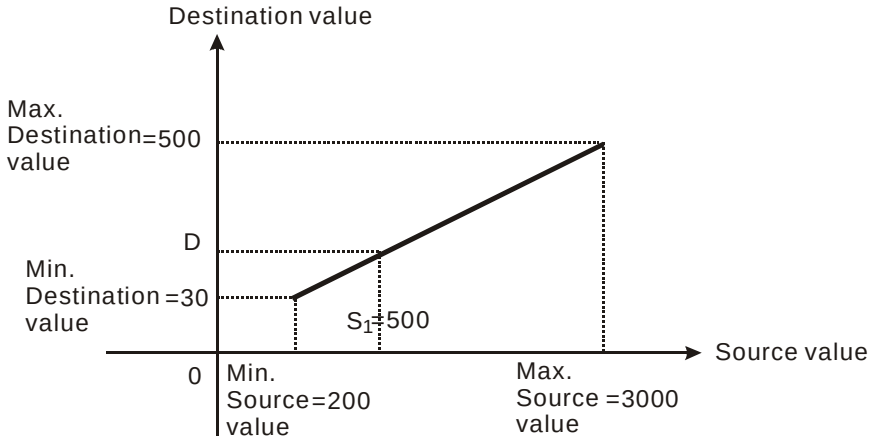
   When all the input values and parameters are set, the output curve is shown as the figure:



Program Example 1:

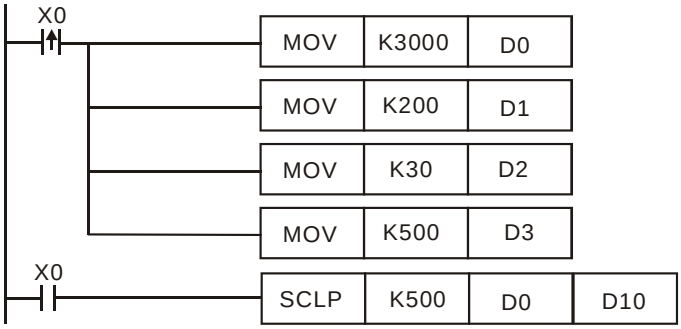
1. Assume $\mathbf{S_1}$ = 500, max. source value D0 = 3,000, min. source value D1 = 200, max. destination value D2 = 500, and min. destination value D3 = 30. When X0 = On, SCLP instruction will be executed and obtain the proportional value at D10.
2. Equation: D10 = [(500 – 200) × (500 – 30)] ÷ (3,000 – 200) + 30 = 80.35. Round off the result into an integer D10 = 80.

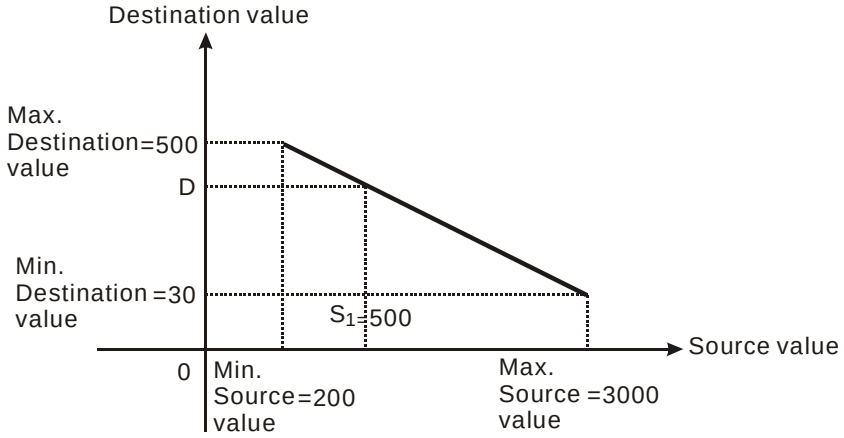




Program Example 2:

1. Assume **S_1** = 500, max. source value D0 = 3,000, min. source value D1 = 200, max. destination value D2 = 30, and min. destination value D3 = 500. When X0 = On, SCLP instruction will be executed and obtain the proportional value at D10.
2. Equation: D10 = [(500 – 200) × (30 – 500)] ÷ (3,000 – 200) + 500 = 449.64. Round off the result into an integer D10 = 450.

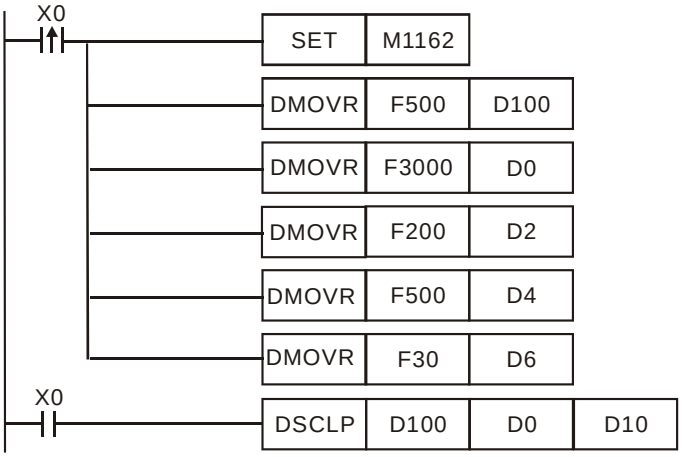


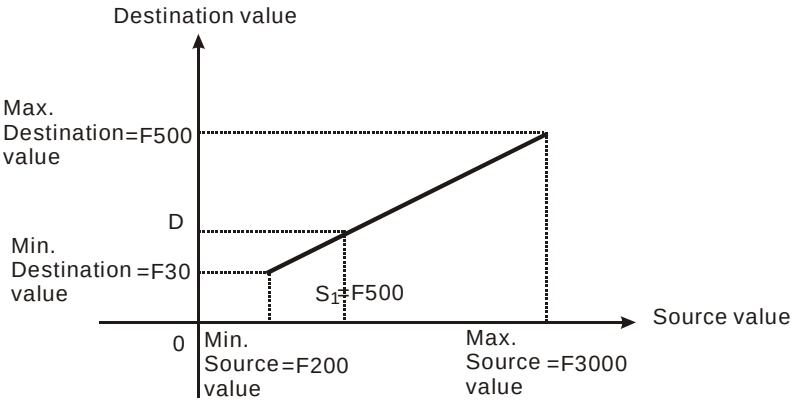


Program Example 3:

1. Assume the source of **S_1** D100 = F500, max. source value D0 = F3000, min. source value D2 = F200, max. destination value D4 = F500, and min. destination value D6 = F30. When X0 = On, set up M1162, adopt floating point operation and execute DSCLP instruction. The proportional value will be obtained at D10.
2. Equation: D10 = [(F500 – F200) × (F500 – F30)] ÷ (F3000 – F200) + F30 = F80.35. Round off the result into an

integer D10 = F80.





Remarks:

1. Range of S_1 for 16-bit instruction: max. source value ≥ S_1 ≥ min. source value; -32,768 ~ 32,767. If the value falls without the bounds, the bound value will be used for calculation.
2. Range of integer S_1 for 32-bit instruction: max. source value ≥ S_1 ≥ min. source value; -2,147,483,648 ~ 2,147,483,647. If the value falls without the bounds, the bound value will be used for calculation.
3. Range of floating point S_1 for 32-bit instruction: max. source value ≥ S_1 ≥ min. source value; following the range of 32-bit floating point. If the value falls without the bounds, the bound value will be used for calculation.
4. Please be aware that the max. source value must > min. source value, but it is not necessary that max. destination value > min. destination value.

API	Mnemonic		Operands	Function
205	CMPT	P	(S1) (S2) (n) (D)	Compare table

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CMPT: 9 steps
S_1											*	*	*			DCMPT: 17 steps
S_2											*	*	*			
n					*	*							*			DCMPTP: 17 steps
D								*	*	*	*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source device 1 **S_2**: Source device 2 **n**: Data length/function **D**: Destination device

Explanations:

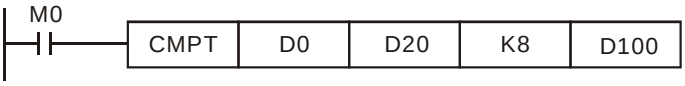
1. **S_1** and **S_2** can be T/C/D devices, for C devices only 16-bit devices are applicable (C0~C199).
2. The high 16-bit value in the operand **n** used in the 32-bit instruction is an invalid value.
3. The low 8-bit value in the operand **n** indicates the data length. The operand **n** used in the 16-bit instruction should be within the range between 1 and 16. The operand **n** used in the 32-bit instruction should be within the range between 1 and 32. PLC will take the upper/lower bound value if set value exceeds the available range.
4. The high 8-bit value in the operand **n** indicates the comparison condition.

Value	K0	K1	K2	K3	K4
Comparison condition	**S_1** = **S_2**	**S_1** < **S_2**	**S_1** <= **S_2**	**S_1** > **S_2**	**S_1** >= **S_2**

5. If n used in the 16-bit instruction is set to H0108, it means that 8 pieces of data are compared to 8 pieces of data, and the “larger than” comparison is performed. If n used in the 32-bit instruction is set to H00000320, it means that 32 pieces data are compared to 32 pieces of data, and the “less than” comparison is performed.
6. If the setting value for the comparison condition exceeds the range, or the firmware version does not support the comparison condition, the default “equal to” comparison is performed.
7. The comparison values used in the 16-bit instruction are signed values. The comparison values used in the 32-bit instruction are 32-bit values (M1162=Off), or floating-point values (M1162=On).
8. Data written in operand **D** will all be stored in 16-bit format or in 32-bit format. When data length is less than 16 or 32, the null bits are fixed as 0, e.g. if **n** = K8, bit 0~7 will be set according to compare results, and bit 8~15 will all be 0.
9. If the comparison result meets the condition, the corresponding bit is set to 1. Otherwise, it is set to 0.

Program example:

When M0 = ON, compare the 16-bit value in D0~D7 with D20~D27 and store the results in D100.



- Content in D0~D7:

No.	D0	D1	D2	D3	D4	D5	D6	D7
Value	K10	K20	K30	K40	K50	K60	K70	K80

- Content in D20~D27:

No.	D20	D21	D22	D23	D24	D25	D26	D27
Value	K12	K20	K33	K44	K50	K66	K70	K88

- After the comparison of CMPT instruction, the associated bit will be 1 if two devices have the same value, and other bits will all be 0. Therefore the results in D100 will be as below:

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8~15
D100	0	1	0	0	1	0	1	0	0…0
	H0052 (K82)								

API	Mnemonic	Operands	Function
206	ASDRW	(S1) (S2) (S)	ASDA servo drive R/W

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ASDRW: 7 steps
S_1					*	*							*			
S_2					*	*							*			
S													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Address of servo drive (K0~K254) **S_2**: Function code **S**: Register for read/written data

Explanations:

1. ASDRW communication instruction supports COM2 (RS-485) and COM3 (RS-485)
2. **S_1**: station number of servo drive. Range: K0~K254. K0 indicates broadcasting, i.e. PLC will not receive feedback data.
3. **S_2**: function code. Please refer to the table below.
4. **S**: Register for read/written data. Please refer to the table below for explanations.
5. Explanations of function code:

Exclusively for ASDA of A-type, AB type, A+ type, B type				
Code	Function	Parameter	Com. Addr.	Read/Write data (Settings)
K0(H0)	Status monitor	P0-04 ~ P0-08	0004H ~ 0008H	**S**+0 ~ **S**+4: Please refer to explanations in ASDA manuals.
K1(H1)	Block Data Read Register	P0-09 ~ P0-16	0009H ~ 0010H	**S**+0 ~ **S**+7: Please refer to explanations in ASDA manuals. B Type is not supported.
K2(H2)	Block Data Write Register	P0-09 ~ P0-16	0009H ~ 0010H	**S**+0 ~ **S**+7: Please refer to explanations in ASDA manuals. B Type is not supported.
K3(H3)	JOG Operation	P4-05	0405H	**S**: Range: 1~3000, 4999, 4998, 5000
K4(H4)	Servo ON/OFF	P2-30	021EH	**S**: K1 = ON, Others = OFF
K5(H5)	Speed Command (3 sets)	P1-09 ~ P1-11	0109H ~ 010BH	**S**+0 ~ **S**+2: Range: -5000~+5000
K6(H6)	Torque Command (3 sets)	P1-12 ~ P1-14	010CH ~ 010EH	**S**+0 ~ **S**+2: Range: -300~+300

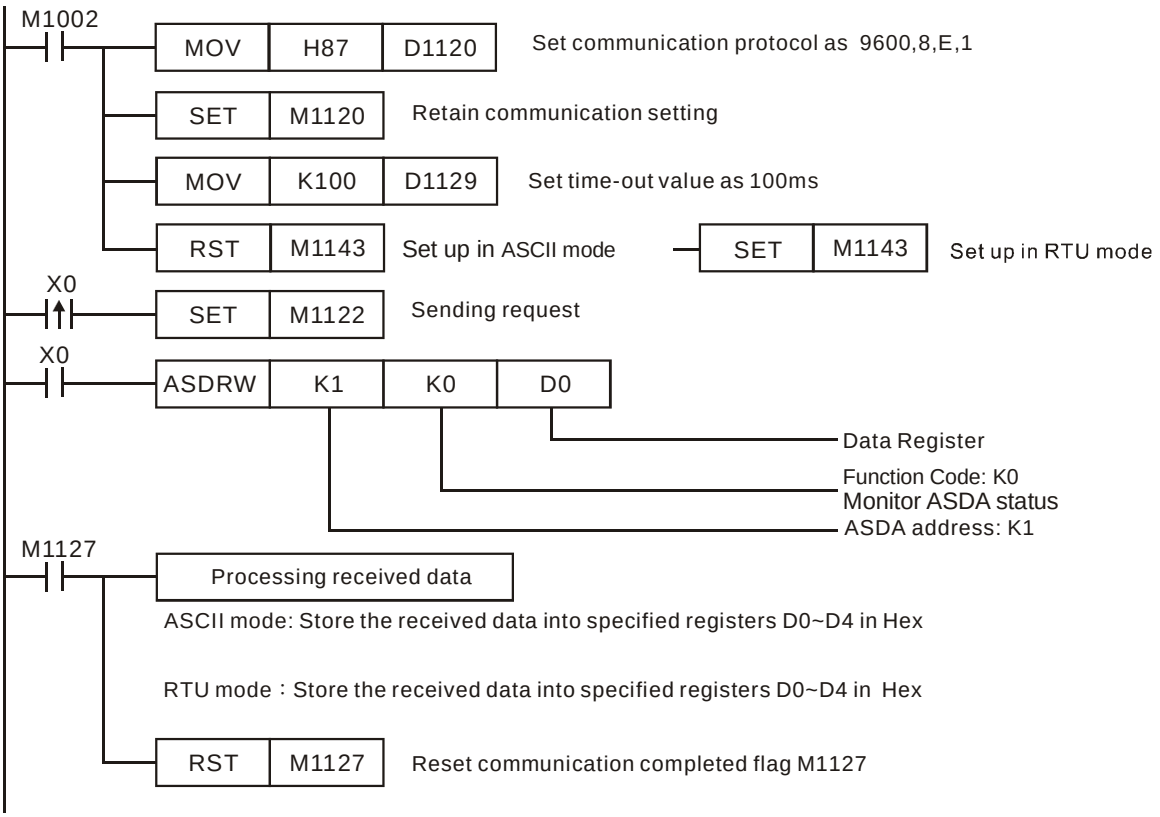
For A2-type only				
Code	Function	Parameter	Com. Addr.	Read/Write data (Settings)
K16(H10)	Status monitor (Read)	P0-09 ~ P0-13	0012H ~ 001BH	**S**+0 ~ **S**+9: Please refer to explanations in ASDA-A2 manual.
K17(H11)	Status monitor selection (Write)	P0-17 ~ P0-20	0022H ~ 0029H	**S**+0 ~ **S**+7: Please refer to explanations in ASDA-A2 manual.

For A2-type only				
Code	Function	Parameter	Com. Addr.	Read/Write data (Settings)
K18(H12)	Mapping parameter (Write)	P0-25 ~ P0-28	0032H ~ 0039H	**S**+0 ~ **S**+7: Please refer to explanations in ASDA-A2 manual.
K19(H13)	JOG Operation	P4-05	040AH	**S**: Range: 1~5000, 4999, 4998, 0
K20(H14)	Auxiliary Function (Servo ON/OFF)	P2-30	023CH	**S**: K1 = ON, Others = OFF
K21(H15)	Speed Command (3 sets)	P1-09 ~ P1-11	0112H ~ 0117H	**S**+0 ~ **S**+5: Range: -60000~+60000
K22(H16)	Torque Command (3 sets)	P1-12 ~ P1-14	0118H ~ 011DH	**S**+0 ~ **S**+5: Range: -300~+300
K23(H17)	Block Data Read / Write Register (for mapping parameter)	P0-35 ~ P0-38	0046H~ 004DH	**S**+0 ~ **S**+7: Please refer to explanations in ASDA-A2 manual.

6. For relative M flags and special D registers, please refer to explanations of API 80 RS instruction.

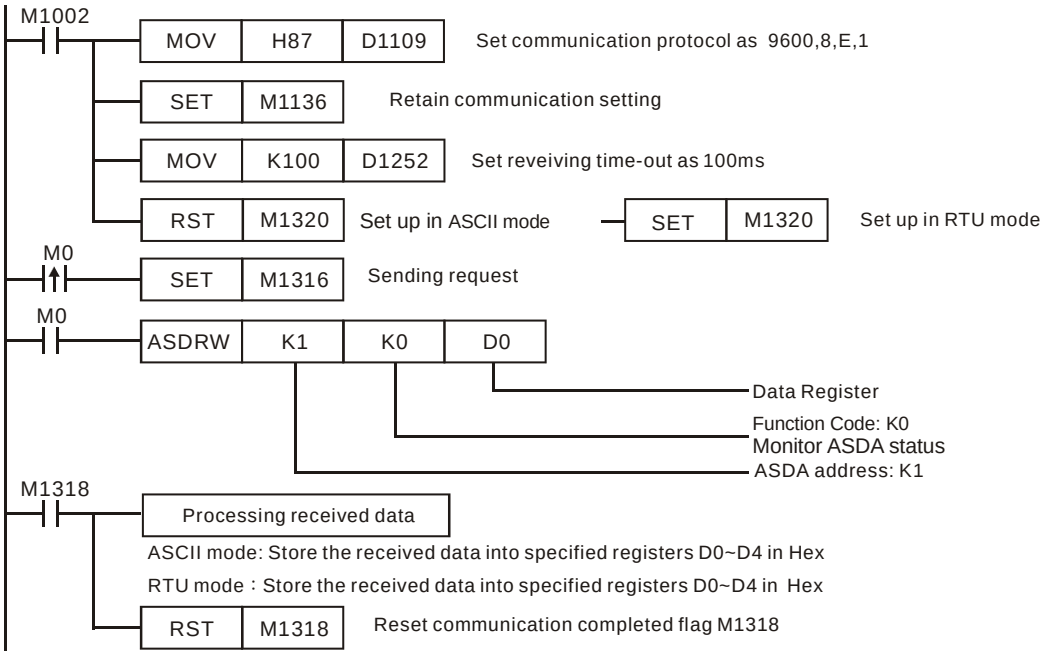
Program example 1: COM2 (RS-485)

1. When X0 = ON, PLC will send out communication commands by COM2 to read status of servo drive.
2. When PLC received the feedback data from ASDA, M1127 will be active and the read data will be stored in D0~D4.



Program example 2: COM3(RS-485)

1. When M0 = ON, PLC sends communication commands by COM3 to read servo drive status.
2. When PLC received the feedback data from ASDA, M1318 will be active and the read data will be stored in D0~D4.



Points to note: Relative flags and special D registers of COM2/COM3 :

	COM2	COM3	Function Description
Protocol setting	M1120	M1136	Retain communication setting
	M1143	M1320	ASCII/RTU mode selection
	D1120	D1109	Communication protocol
	D1121	D1255	PLC communication address
Sending request	M1122	M1316	Sending request
	D1129	D1252	Communication timeout setting (ms)
Receiving completed	M1127	M1318	Data receiving completed
Errors	-	M1319	Data receiving error
	-	D1253	Communication error code
	M1129	-	Communication timeout setting (ms)
	M1140	-	COM2 (RS-485) MODRD/MODWR/MODRW data receiving error
	M1141	-	MODRD/MODWR/MODRW parameter error (Exception Code exists in received data) Exception Code is stored in D1130
	D1130	-	COM2 (RS-485) Error code (exception code) returning from Modbus communication

API	Mnemonic	Operands	Function
207	CSFO	(S) (S1) (D)	Catch speed and proportional output

OP \ Type	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CSFO: 7 steps
S	*															
S_1													*			
D													*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Source device of signal input (Only X0 and X1 are available) **S_1**: Sample time setting and the input speed information **D**: Output proportion setting and output speed information

Explanations:

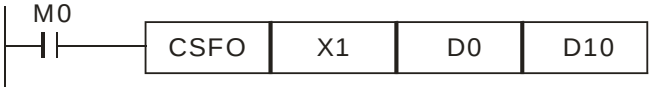
1. EH3/SV2 V1.88 or later: added M1609 flag to select the mode, a general mode (M1609=OFF) or a MPG mode (M1609=ON). For general mode, the pulse goes according to the actual pulse output behavior. The pulse output direction will only change, when the pulse outputs are all done. For MPG mode, the pulse goes according to the MPG behavior. If the MPG stops, the pulse stops. If the MPG changes direction, the pulse output direction changes accordingly. EH3/SV2 V1.86 supports general mode only.
2. EH3/SV2 V1.88 or later: added M1608 flag to added CH1 (Y2/Y3) output to follow the same source of pulse input (X0/X1). Use D1048 to set the proportional output value, ranging K K1(1%) ~ K10000(10000%).
3. When **S** specifies X0, PLC only uses X0 input point and its associated high speed pulse output: Y0, in this case Y1 is normal output point. When **S** specifies X1, PLC uses X0 (A phase) and X1 (B phase) input points and their associated output: Y0 (Pulse) / Y1 (Dir).
4. If **S** specifies X1 with 2-phase 2 inputs, the counting mode is fixed as quadruple frequency.
5. During pulse output process of Y0, special registers (D1337, D1336) storing the current number of output pulses will be updated when program scan proceeds to this instruction.
6. **S_1** occupies consecutive 4 16-bit registers. **S_1** +0 specifies the sampling times, i.e. when **S_1** +0 specifies K1, PLC catches the speed every time when 1 pulse is outputted. Valid range for **S_1** +0 in 1-phase 1-input mode: K1~K100, and 2-phase 2-input mode: K2~K100. If the specified value exceeds the valid range, PLC will take the lower/upper bound value as the set value. Sample time can be changed during PLC operation, however the modified value will take effect until program scan proceeds to this instruction. **S_1**+1 indicates the latest speed sampled by PLC (Read-only). Unit: 1Hz. Valid range: ±10kHz. **S_1**+2 and **S_1**+3 indicate the accumulated number of pulses in 32-bit data (Read-only).
7. **D** occupies 3 consecutive 16-bit registers. **D** +0 specifies the output proportion value. Valid range: K1 (1%) ~ K10000 (10000%). If the specified value exceeds the valid range, PLC will take the lower/upper bound value as the set value. Output proportion can be changed during PLC operation, however the modified value will take effect until program scan proceeds to this instruction. **D**+2 and **D**+1 indicates the output speed in 32-bit data. Unit: 1Hz. Valid range: ±200kHz. When selecting the MPG mode, it takes one more 16-bit register. **D**+3 indicates the pulse output channel, ranging from K0 to K3, indicating the output channels CH0~CH3. Wen

selecting the general mode, the pulse output channel is fixed to CH0. Note: if you need to change the mode from the MPG mode to the general mode or vise versa, you need to close the instruction and re-execute the instruction to ensure the channel switching can be normally done.

8. The pulse output channel selecting: when **S** input point uses X0 as the source, the corresponding pulse output points are Y0, Y2, Y4, Y6 and the general pulse output points are Y1, Y3, Y5, Y7. When **S** input point uses X1 as the source, the corresponding output points are Y0(Pulse) / Y1(Dir) or Y2(Pulse) / Y3(Dir) or Y4(Pulse) / Y5(Dir) or Y6(Pulse) / Y7(Dir) high speed output.
9. The execution of CSFO requires hardware high speed counter function as well as the high speed output function. Therefore, when program scan proceeds to CSFO instruction with high speed counter input points (X0, X1) enabled by DCNT instruction, or high speed pulse outputs (Y0, Y1) enabled by other high speed output instructions, CSFO instruction will not be activated.
10. When pulse outputs via Y0/Y2/Y4/Y6, their corresponding output pulse numbers stored in special D devices (D1337/D1336, D1339/D1338, D1376/D1375, D1378/D1377) are also updated when the instruction is scanned.
11. For single phase input, the max frequency is 10kHz; for 2-phase 2 inputs, the max frequency is 10kHz.
12. The speed sampled by the PLC will be multiplied with the output proportion **D**+0, then the PLC will generate the actual output speed. The PLC will take the integer of the calculated value, i.e. if the calculated result is smaller than 1Hz, the PLC will output with 0Hz. For example, input speed: 10Hz, output proportion: K5 (5%), then the calculation result will be 10 x 0.05 = 0.5Hz. Pulse output will be 0Hz; if output proportion is modified as K15 (15%), then the calculation result will be 10 x 0.15 = 1.5Hz. Pulse output will be 1Hz.

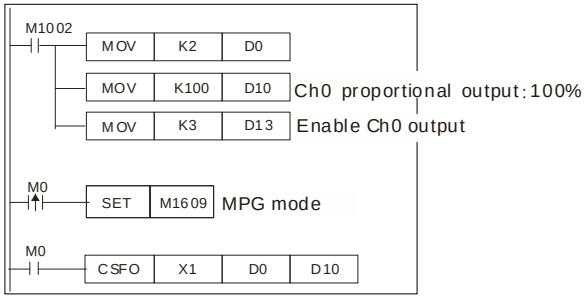
Program Example:

1. If D0 is set as K2, D10 is set as K100:
   When the sampled speed on (X0, X1) is +10Hz (D1 = K10), (Y0, Y1) will output pulses with +10Hz (D12, D11 = K10); When the sampled speed is -10Hz (D1 = K-10), (Y0, Y1) will output pulses with -10Hz (D12, D11 = K-10)
2. If D0 is set as K2, D10 is set as K1000:
   When the sampled speed on (X0, X1) is +10Hz (D1 = K10), (Y0, Y1) will output pulses with +100Hz (D12, D11 = K100); When the sampled speed is -100Hz (D1 = K-100), (Y0, Y1) will output pulses with -100Hz (D12, D11 = K-100).
3. If D0 is set as K10, D10 is set as K10:
   When the sampled speed on (X0, X1) is +10Hz (D1 = K10), (Y0, Y1) will output pulses with +1Hz (D12, D11 = K1); When the sampled speed is -10Hz (D1 = K-10), (Y0, Y1) will output pulses with -1Hz (D12, D11 = K-1)



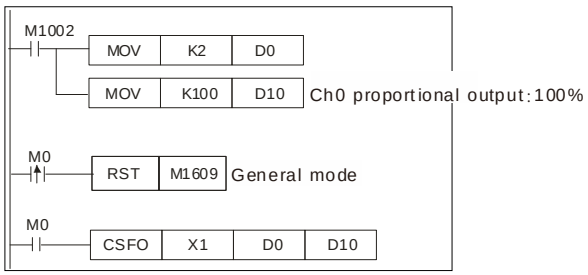
Program Example 1:

EH3/SV2 V.88 or later version: use SET to set the pulse mode to MPG mode (M1609=ON) and select CH3 to output pulse.



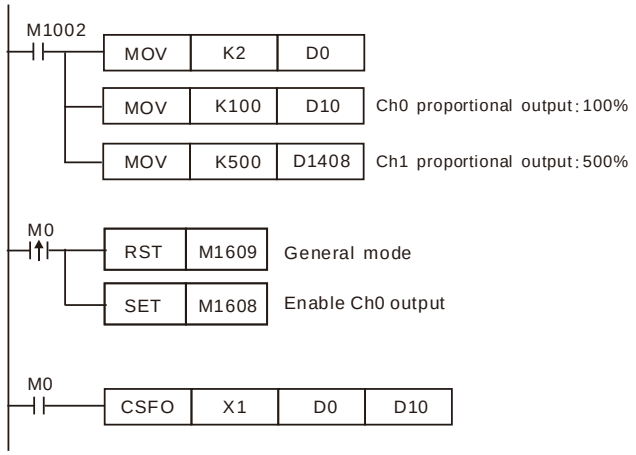
Program Example 2:

EH3/SV2 V.88 or later version: use RST to set the pulse mode to general mode (M1609=OFF).



Program Example 3:

EH3/SV2 V.88 or later version: use RST to set the pulse mode to general mode (M1609=OFF). For the proportional output, CH0: 100% and CH1: 500%.



API	Mnemonic	Operands	Function
215~217	D LD#	(S1) (S2)	Contact Logical Operation LD#

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	LD#: 5 steps DLD#: 9 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Data source device 1 **S_2**: Data source device 2

Explanations:

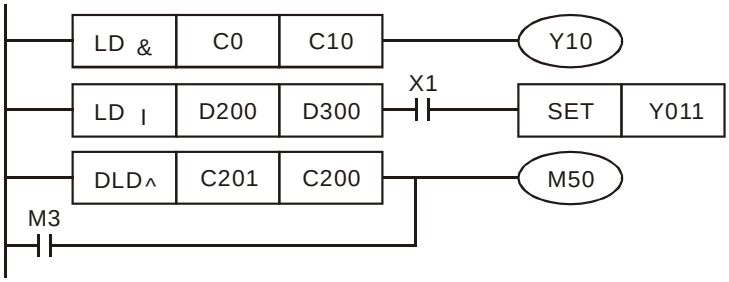
1. See the specifications of each model for the range of operands.
2. This instruction compares the content in **S_1** and **S_2**. If the result is not "0", the continuity of the instruction is enabled. If the result is "0", the continuity of the instruction is disabled.
3. LD# instruction is used for direct connection with BUS.

API No.	16 -bit instruction	32 -bit instruction	Continuity condition	No-continuity condition
215	LD&	**D**LD&	S_1 & $S_2 \neq 0$	S_1 & $S_2 = 0$
216	LD|	**D**LD|	S_1 | $S_2 \neq 0$	S_1 | $S_2 = 0$
217	LD^	**D**LD^	S_1 ^ $S_2 \neq 0$	S_1 ^ $S_2 = 0$

4. **&:** Logical "AND" operation
5. **|:** Logical "OR" operation
6. **^:** Logical "XOR" operation
7. When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DLD#). If 16-bit instructions (LD#) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

Program Example:

1. When the result of logical AND operation of C0 and C10 ≠ 0, Y10 = On.
2. When the result of logical OR operation of D200 and D300 ≠ 0 and X1 = On, Y11 = On will be retained.
3. When the result of logical XOR operation of C201 and C200 ≠ 0 or M3 = On, M50 = On.



API	Mnemonic	Operands	Function
218~220	D AND#	(S1) (S2)	Contact Logical Operation AND#

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	AND#: 5 steps DAND#: 9 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Data source device 1 **S_2**: Data source device 2

Explanations:

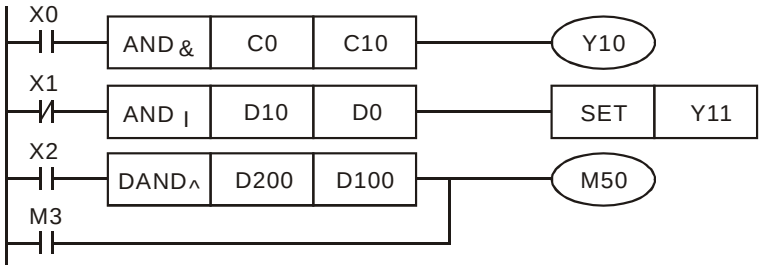
1. See the specifications of each model for the range of operands.
2. This instruction compares the content in **S_1** and **S_2**. If the result is not "0", the continuity of the instruction is enabled. If the result is "0", the continuity of the instruction is disabled.
3. AND# is an operation instruction used on series contacts.

API No.	16 -bit instruction	32 -bit instruction	Continuity condition	No-continuity condition
218	AND&	DAND&	S_1 & S_2 ≠ 0	S_1 & S_2 = 0
219	AND|	DAND|	S_1 | S_2 ≠ 0	S_1 | S_2 = 0
220	AND^	DAND^	S_1 ^ S_2 ≠ 0	S_1 ^ S_2 = 0

4. **&:** Logical "AND" operation
5. **|:** Logical "OR" operation
6. **^:** Logical "XOR" operation
7. When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DAND#). If 16-bit instructions (AND#) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

Program Example:

1. When X0 = On and the result of logical AND operation of C0 and C10 ≠ 0, Y10 = On.
2. When X1 = Off and the result of logical OR operation of D10 and D0 ≠ 0 and X1 = On, Y11 = On will be retained.
3. When X2 = On and the result of logical XOR operation of 32-bit register D200 (D201) and 32-bit register D100 (D101) ≠ 0 or M3 = On, M50 = On.



API	Mnemonic	Operands	Function
221~223	D OR#	S_1 S_2	Contact Logical operation OR#

Type / OP	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	OR#: 5 steps DOR#: 9 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Data source device 1 **S_2**: Data source device 2

Explanations:

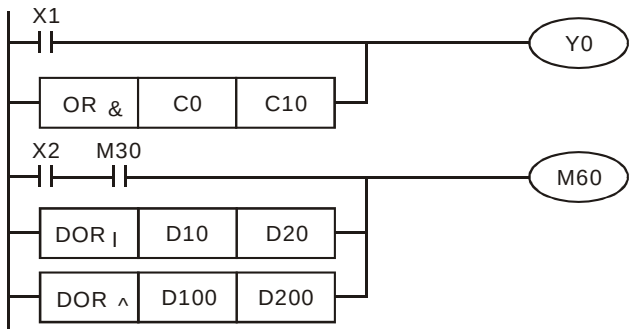
1. See the specifications of each model for the range of operands.
2. This instruction compares the content in **S_1** and **S_2**. If the result is not "0", the continuity of the instruction is enabled. If the result is "0", the continuity of the instruction is disabled.
3. OR# is an operation instruction used on parallel contacts.

API No.	16 -bit instruction	32 -bit instruction	Continuity condition	No-continuity condition
221	OR&	**D**OR&	S_1 & $S_2 \neq 0$	S_1 & $S_2 = 0$
222	OR|	**D**OR|	S_1 | $S_2 \neq 0$	S_1 | $S_2 = 0$
223	OR^	**D**OR^	S_1 ^ $S_2 \neq 0$	S_1 ^ $S_2 = 0$

4. **&:** Logical "AND" operation
5. **|:** Logical "OR" operation
6. **^:** Logical "XOR" operation
7. When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DOR#). If 16-bit instructions (OR#) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

Program Example:

1. When X1 = On and the result of logical AND operation of C0 and C10 ≠ 0, Y10 = On.
2. M60 will be On when X2 = On and M30 = On, or the result of logical OR operation of 32-bit register D10 (D11) and 32-bit register D20 (D21) ≠ 0, or the result of logical XOR operation of 32-bit register D200 (D201) and 32-bit counter C235 ≠ 0.



API	Mnemonic	Operands	Function
224~230	D LD※	(S1) (S2)	Load Compare

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	LD※: 5 steps
S_1					*	*	*	*	*	*	*	*	*	*	*	DLD※: 9 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Data source device 1 **S_2**: Data source device 2

Explanations:

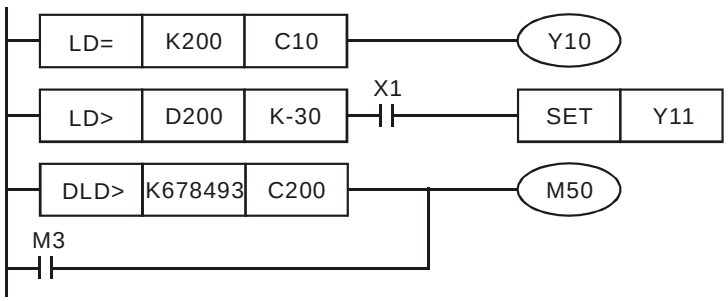
1. See the specifications of each model for the range of operands.
2. This instruction compares the content in **S_1** and **S_2**. Take API224 (LD=) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
3. LD※ instruction is used for direct connection with BUS.

API No.	16 -bit instruction	32 -bit instruction	Continuity condition	No-continuity condition
224	LD =	**D**LD =	**$S_1 = S_2$**	**$S_1 \neq S_2$**
225	LD >	**D**LD >	**$S_1 > S_2$**	**$S_1 \leq S_2$**
226	LD <	**D**LD <	**$S_1 < S_2$**	**$S_1 \geq S_2$**
228	LD < >	**D**LD < >	**$S_1 \neq S_2$**	**$S_1 = S_2$**
229	LD < =	**D**LD < =	**$S_1 \leq S_2$**	**$S_1 > S_2$**
230	LD > =	**D**LD > =	**$S_1 \geq S_2$**	**$S_1 < S_2$**

4. When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DLD※). If 16-bit instructions (LD※) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

Program Example:

1. When the content in C10 = K200, Y10 = On.
2. When the content in D200 > K-30 and X1 = On, Y11= On will be retained.
3. When the content in C200 < K678,493 or M3 = On, M50 = On.



API	Mnemonic	Operands	Function
232~238	D AND※	(S1) (S2)	AND Compare

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	AND※: 5 steps
S_1					*	*	*	*	*	*	*	*	*	*	*	DAND※: 9 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Data source device 1 **S_2**: Data source device 2

Explanations:

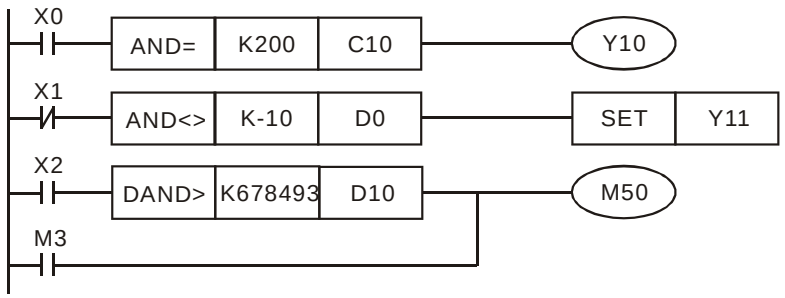
1. See the specifications of each model for the range of operands.
2. This instruction compares the content in **S_1** and **S_2**. Take API232 (AND=) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
3. AND※ is a comparison instruction is used on series contacts.

API No.	16 –bit instruction	32 –bit instruction	Continuity condition	No-continuity condition
232	AND =	**D**AND =	**$S_1 = S_2$**	**$S_1 \neq S_2$**
233	AND >	**D**AND >	**$S_1 > S_2$**	**$S_1 \leq S_2$**
234	AND <	**D**AND <	**$S_1 < S_2$**	**$S_1 \geq S_2$**
236	AND < >	**D**AND < >	**$S_1 \neq S_2$**	**$S_1 = S_2$**
237	AND < =	**D**AND < =	**$S_1 \leq S_2$**	**$S_1 > S_2$**
238	AND > =	**D**AND > =	**$S_1 \geq S_2$**	**$S_1 < S_2$**

4. When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DAND※). If 16-bit instructions (AND※) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

Program Example:

1. When X0 = On and the content in C10 = K200, Y10 = On.
2. When X1 = Off and the content in D0 ≠ K-10, Y11= On will be retained.
3. When X2 = On and the content in 32-bit register D0 (D11) < 678,493 or M3 = On, M50 = On.



API	Mnemonic	Operands	Function
240~246	D OR※	(S1) (S2)	OR Compare

Type / OP	Bit Devices				Word Devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	OR*: 5 steps DOR*: 9 steps
S_2					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Data source device 1 **S_2**: Data source device 2

Explanations:

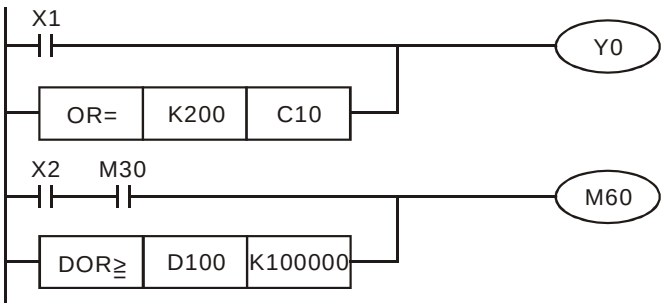
1. See the specifications of each model for the range of operands.
2. This instruction compares the content in **S_1** and **S_2**. Take API240 (OR=) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
3. OR※ is an comparison instruction used on parallel contacts.

API No.	16 -bit instruction	32 -bit instruction	Continuity condition	No-continuity condition
240	OR =	**D**OR =	**S_1 = S_2**	**S_1 ≠ S_2**
241	OR >	**D**OR >	**S_1 > S_2**	**S_1 ≤ S_2**
242	OR <	**D**OR <	**S_1 < S_2**	**S_1 ≥ S_2**
244	OR < >	**D**OR < >	**S_1 ≠ S_2**	**S_1 = S_2**
245	OR < =	**D**OR < =	**S_1 ≤ S_2**	**S_1 > S_2**
246	OR > =	**D**OR > =	**S_1 ≥ S_2**	**S_1 < S_2**

4. When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DOR※). If 16-bit instructions (OR※) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

Program Example:

1. When X1 = On and the present value of C10 = K200, Y0 = On.
2. M60 will be On when X2 = On, M30 = On and the content in 32-bit register D100 (D101) ≥ K100,000.



API	Mnemonic	Operands	Function
266	D BOUT	(D) (n)	Output Specified Bit of a Word

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*			BOUT: 5 steps DBOUT: 9 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

D: Destination output device **n**: Device specifying the output bit

Explanations:

1. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
2. BOUT instruction performs bit output on the output device according to the value specified by operand **n.**

Status of Coils and Associated Contacts:

Evaluation result	BOUT instruction		
	Coil	Associated Contacts	
		NO contact (normally open)	NC contact (normally closed)
FALSE	OFF	Current blocked	Current flows
TRUE	ON	Current flows	Current blocked

Program Example:

Instruction:			Operation:
LDI	X0		Load NC contact X0
AND	X1		Connect NO contact X1 in series.
BOUT	K4Y0	D0	When D0 = k1, executes output on Y1 When D0 = k2, executes output on Y2

API	Mnemonic		Operands	Function
267	D	BSET	(D) (n)	Set ON Specified Bit of a Word

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*			BSET: 5 steps DBSET: 9 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

D: Destination device to be Set ON **n**: Device specifying the bit to be Set ON

Explanations:

1. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
2. When BSET instruction executes, the output device specified by operand **n** will be On and latched. To reset the On state of the device, BRST instruction is required.

Program Example:

Instruction:			Operation:
LDI	X0		Load NC contact X0
AND	X1		Connect NO contact X1 in series.
BSET	K4Y0	D0	When D0 = k1, Y1 is ON and latched When D0 = k2, Y2 = ON and latched

API	Mnemonic		Operands	Function
268	D	BRST	(D) (n)	Reset Specified Bit of a Word

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*			BRST: 5 steps DBRST: 9 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

D: Destination device to be reset **n:** Device specifying the bit to be reset

Explanations:

1. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
2. When BRST instruction executes, the output device specified by operand **n** will be reset (OFF).

Program Example:

Instruction:			Operation:
LD	X0		Load NO contact X0
BRST	K4Y0	D0	When D0 = k1, Y1 is OFF When D0 = k2, Y2 = OFF

API	Mnemonic		Operands	Function
269	D	BLD	(S) (n)	Load NO Contact by Specified Bit

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BLD: 5 steps
S								*	*	*	*	*	*			DBLD: 9 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Reference source device **n**: Reference bit

Explanations:

1. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
2. BLD instruction is used to load NO contact whose contact state is defined by the reference bit **n** in reference device **D**, i.e. if the bit specified by **n** is ON, the NO contact will be ON, and vice versa.

Program Example:

Instruction:			Operation:
BLD	D0	K3	Load NO contact X0 with bit Status of bit3 in D0
OUT	Y0		Device coil Y0

API	Mnemonic		Operands	Function
270	D	BLDI	(S) (n)	Load NC Contact by Specified Bit

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BLDI: 5 steps
S								*	*	*	*	*	*			DBLDI: 9 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S: Reference source device **n**: Reference bit

Explanations:

1. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
2. BLD instruction is used to load NC contact whose contact state is defined by the reference bit **n** in reference device **D**, i.e. if the bit specified by **n** is ON, the NC contact will be ON, and vice versa.

Program Example:

Instruction:			Operation:
BLDI	D0	K1	Load NC contact with bit status of bit1 in D0
OUT	Y0		Drive coil Y0

API	Mnemonic		Operands	Function
271	D	BAND	(S) (n)	Connect NO Contact in Series by Specified Bit

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BAND: 5 steps DBAND: 9 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

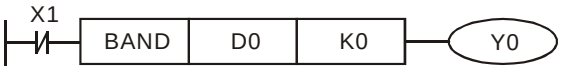
Operands:

S: Reference source device **n**: Reference bit

Explanations:

1. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
2. BAND instruction is used to connect NO contact in series. The current state of the contact which is connected in series is read, and then the logical AND operation is performed on the current state and the previous logical operation result. The final result is stored in the accumulative register.

Program Example:



Instruction:			Operation:
LDI	X1		Load NC contact X1
BAND	D0	K0	Connect NO contact in series, whose state is defined by bit0 of D0
OUT	Y0		Drive coil Y0

API	Mnemonic		Operands	Function
272	D	BANI	(S) (n)	Connect NC Contact in Series by Specified Bit

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BANI: 5 steps DBANI: 9 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

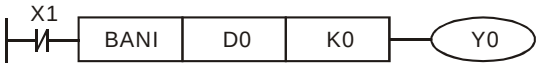
Operands:

S: Reference source device **n**: Reference bit

Explanations:

1. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
2. BANI instruction is used to connect NC contact in series. The current state of the contact which is connected in series is read, and then the logical AND operation is performed on the current state and the previous logical operation result. The final result is stored in the accumulative register.

Program Example:



Instruction:			Operation:
LDI	X1		Load NC contact X1
BANI	D0	K0	Connect NC contact in series, whose state is defined by bit0 of D0
OUT	Y0		Drive coil Y0

API	Mnemonic		Operands	Function
273	D	BOR	(S) (n)	Connect NO Contact in Parallel by Specified Bit

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BOR: 5 steps DBOR: 9 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

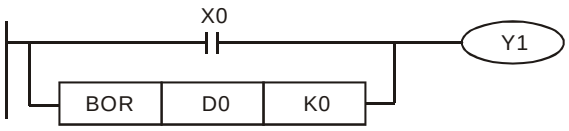
Operands:

S: Reference source device **n:** Reference bit

Explanations:

1. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
2. BOR instruction is used to connect NO contact in parallel. The current state of the contact which is connected in series is read, and then the logical OR operation is performed on the current state and the previous logical operation result. The final result is stored in the accumulative register.

Program Example:



Instruction:			Operation:
LD	X0		Load NO contact X0
BOR	D0	K0	Connect NO contact in parallel, whose state is defined by bit0 of D0
OUT	Y1		Drive coil Y1

API	Mnemonic		Operands	Function
274	D	BORI	(S) (n)	Connect NC Contact in Parallel by Specified Bit

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BORI: 5 steps DBORI: 9 steps
n					*	*	*	*	*	*	*	*	*	*	*	

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

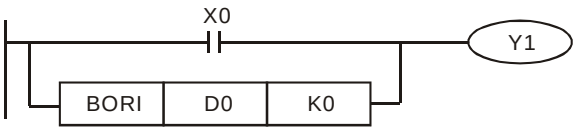
Operands:

S: Reference source device **n:** Reference bit

Explanations:

1. Available range for the value in operand **n**: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
2. BORI instruction is used to connect NC contact in parallel. The current state of the contact which is connected in series is read, and then the logical OR operation is performed on the current state and the previous logical operation result. The final result is stored in the accumulative register.

Program Example:



Instruction:			Operation:
LD	X0		Load NO contact X0
BORI	D0	K0	Connect NC contact in parallel, whose state is defined by bit0 of D0
OUT	Y1		Drive coil Y1

API	Mnemonic	Operands	Function
275~280	FLD※	(S1) (S2)	Floating Point Contact Type Comparison

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			FLD※: 9 steps
S_2											*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source device 1 **S_2**: Source device 2

Explanations:

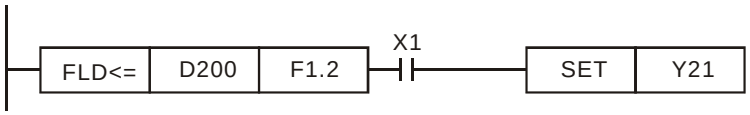
1. This instruction compares the content in **S_1** and **S_2**. Take API275 (FLD=) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
2. The user can specify the floating point value directly into operands **S_1** and **S_2** (e.g. F1.2) or store the floating point value in D registers for further operation.
3. FLD※ (※**:** =, >, <, <>, ≤, ≥) instruction is used for direct connection with left hand bus bar.

API No.	32 -bit instruction	Continuity condition	Discontinuity condition
275	FLD=	$S_1 = S_2$	$S_1 \neq S_2$
276	FLD>	$S_1 > S_2$	$S_1 \leqq S_2$
277	FLD<	$S_1 < S_2$	$S_1 \geqq S_2$
278	FLD<>	$S_1 \neq S_2$	$S_1 = S_2$
279	FLD<=	$S_1 \leqq S_2$	$S_1 > S_2$
280	FLD>=	$S_1 \geqq S_2$	$S_1 < S_2$

If the floating-point value in S1 or S2 is from external input, or from ISPSoft, chances are a minor value difference may occur and the result of the "FLD=" instruciton may not be dependable. Use instructions "FLD<" and" FAND>" or "FLD>" and "FAND<" instead when the floating-point value in S1 or S2 is from external input, or from ISPSoft.

Program Example:

When the content in D200 (D201) ≤ F1.2 and X1 is ON, Y21 = ON and latched.



API	Mnemonic	Operands	Function
281~286	FAND※	(S1) (S2)	Floating Point Serial Type Comparison

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			FAND※: 9 steps
S_2											*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source device 1 **S_2**: Source device 2

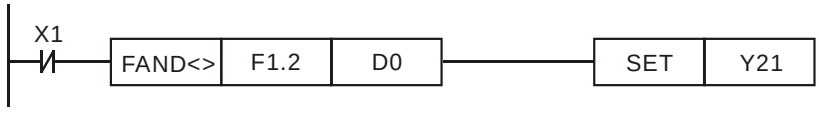
Explanations:

1. This instruction compares the content in **S_1** and **S_2**. Take API281 (FAND =) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
2. The user can specify the floating point value directly into operands **S_1** and **S_2** (e.g. F1.2) or store the floating point value in D registers for further operation.
3. FAND※ (※**:** =, >, <, <>, ≤, ≥) instruction is used for serial connection with contacts.

API No.	32-bit instruction	Continuity condition	Discontinuity condition
281	FAND＝	**S_1**＝**S_2**	**S_1**≠**S_2**
282	FAND＞	**S_1**＞**S_2**	**S_1**≦**S_2**
283	FAND＜	**S_1**＜**S_2**	**S_1**≧**S_2**
284	FAND＜＞	**S_1**≠**S_2**	**S_1**＝**S_2**
285	FAND＜＝	**S_1**≦**S_2**	**S_1**＞**S_2**
286	FAND＞＝	**S_1**≧**S_2**	**S_1**＜**S_2**

Program Example:

When X1 is OFF and the content in D0 (D1) does not equal to F1.2, Y21 = ON and latched.



API	Mnemonic	Operands	Function
287~292	FOR※	(S1) (S2)	Floating Point Parallel Type Comparison

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			FOR※: 9 steps
S_2											*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source device 1 **S_2**: Source device 2

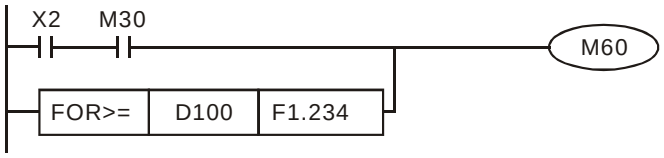
Explanations:

1. This instruction compares the content in **S_1** and **S_2**. Take API287 (FOR =) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled
2. The user can specify the floating point value directly into operands **S_1** and **S_2** (e.g. F1.2) or store the floating point value in D registers for further operation.
3. OR※ (※**:** =, >, <, <>, ≤, ≥) instruction is used for parallel connection with contacts.

API No.	32-bit instruction	Continuity condition	Discontinuity condition
287	FOR＝	**S_1**＝**S_2**	**S_1**≠**S_2**
288	FOR＞	**S_1**＞**S_2**	**S_1**≦**S_2**
289	FOR＜	**S_1**＜**S_2**	**S_1**≧**S_2**
290	FOR＜＞	**S_1**≠**S_2**	**S_1**＝**S_2**
291	FOR＜＝	**S_1**≦**S_2**	**S_1**＞**S_2**
292	FOR＞＝	**S_1**≧**S_2**	**S_1**＜**S_2**

Program Example:

When both X2 and M30 are OFF and the content in D100 (D101) ≥ F1.234, M60 = ON.



API	Mnemonic	Operands	Function
296~301	D LDZ※	(S1) (S2) (S3)	Comparing contact type absolute values LDZ※

Type / OP	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*			LDZ※: 7 steps DLDZ※: 13 steps
S_2					*	*	*	*	*	*	*	*	*			
S_3					*	*	*	*	*	*	*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source device 1 **S_2**: Source device 2 **S_3**: Source device 3

Explanations:

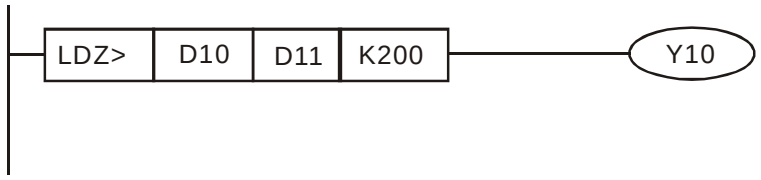
1. EH3 V1.40 and SV2 V1.20 (and above) are supported.
2. The absolute value of the difference between **S_1** and **S_2** is compared with the absolute value of **S_3**. Take LDZ> for example. If the comparison result is that the absolute value of the difference between **S_1** and **S_2** is greater than the absolute value of **S_3**, the condition of the instruction is met. If the comparison result is that the absolute value of the difference between **S_1** and **S_2** is less than or equal to the absolute value of **S_3**, the condition of the instruction is not met.
3. The instruction can be connected to a busbar.

API No.	16-bit instruction	32-bit instruction	Comparison result	
			On	Off
296	LDZ>	**D**LDZ>	**$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$**	**$\lvert S_1 - S_2 \rvert \leqq \lvert S_3 \rvert$**
297	LDZ>=	**D**LDZ>=	**$\lvert S_1 - S_2 \rvert \geqq \lvert S_3 \rvert$**	**$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$**
298	LDZ<	**D**LDZ<	**$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$**	**$\lvert S_1 - S_2 \rvert \geqq \lvert S_3 \rvert$**
299	LDZ<=	**D**LDZ<=	**$\lvert S_1 - S_2 \rvert \leqq \lvert S_3 \rvert$**	**$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$**
300	LDZ=	**D**LDZ=	**$\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$**	**$\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$**
301	LDZ<>	**D**LDZ<>	**$\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$**	**$\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$**

4. A 32-bit counter (C200~C255) must be used with the 32-bit instruction DLDZ※. If it is used with the 16-bit instruction LDZ※, a program error will occur, and the ERROR LED indicator on the PLC will blink.

Program Example:

If the absolute value of the difference between D10 and D11 is greater than K200, Y0 will be On. If the absolute value of the difference between D10 and D11 is less than or equal to K200, Y0 will be Off.



API	Mnemonic		Operands	Function
302~307	D	ANDZ※	(S1) (S2) (S3)	Comparing contact type absolute values ANDZ※

OP \ Type	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*			ANDZ※: 7 steps DANDZ※: 13 steps
S_2					*	*	*	*	*	*	*	*	*			
S_3					*	*	*	*	*	*	*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source device 1 **S_2**: Source device 2 **S_3**: Source device 3

Explanations:

1. EH3 V1.40 and SV2 V1.20 (and above) are supported.
2. The absolute value of the difference between **S_1** and **S_2** is compared with the absolute value of **S_3**. Take ANDZ> for example. If the comparison result is that the absolute value of the difference between **S_1** and **S_2** is greater than the absolute value of **S_3**, the condition of the instruction is met. If the comparison result is that the absolute value of the difference between **S_1** and **S_2** is less than or equal to the absolute value of **S_3**, the condition of the instruction is not met.
3. The instruction ANDZ※ is connected to a contact in series.

API No.	16-bit instruction	32-bit instruction	Comparison result	
			On	Off
302	ANDZ>	**D**ANDZ>	$\|S_1 - S_2\| > \|S_3\|$	$\|S_1 - S_2\| \leqq \|S_3\|$
303	ANDZ>=	**D**ANDZ>=	$\|S_1 - S_2\| \geqq \|S_3\|$	$\|S_1 - S_2\| < \|S_3\|$
304	ANDZ<	**D**ANDZ<	$\|S_1 - S_2\| < \|S_3\|$	$\|S_1 - S_2\| \geqq \|S_3\|$
305	ANDZ<=	**D**ANDZ<=	$\|S_1 - S_2\| \leqq \|S_3\|$	$\|S_1 - S_2\| > \|S_3\|$
306	ANDZ=	**D**ANDZ=	$\|S_1 - S_2\| = \|S_3\|$	$\|S_1 - S_2\| \neq \|S_3\|$
307	ANDZ<>	**D**ANDZ<>	$\|S_1 - S_2\| \neq \|S_3\|$	$\|S_1 - S_2\| = \|S_3\|$

4. A 32-bit counter (C200~C255) must be used with the 32-bit instruction DANDZ※. If it is used with the 16-bit instruction ANDZ※, a program error will occur, and the ERROR LED indicator on the PLC will blink.

Program Example:

If M0 is On, and the absolute value of the difference between D10 and D11 is greater than K200, Y0 will be On. If the absolute value of the difference between D10 and D11 is less than or equal to K200, Y0 will be Off.

API	Mnemonic	Operands	Function
308~313	D ORZ※	(S1) (S2) (S3)	Comparing contact type absolute values ORZ※

OP \ Type	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*			ORZ※: 7 steps DORZ※: 13 steps
S_2					*	*	*	*	*	*	*	*	*			
S_3					*	*	*	*	*	*	*	*	*			

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Operands:

S_1: Source device 1 **S_2**: Source device 2 **S_3**: Source device 3

Explanations:

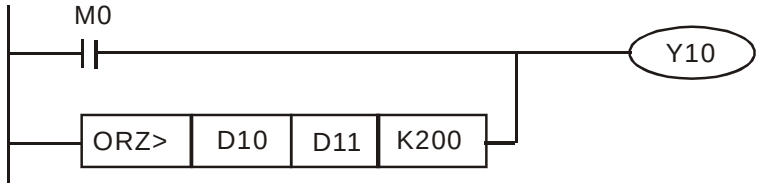
1. EH3 V1.40 and SV2 V1.20 (and above) are supported.
2. The absolute value of the difference between **S_1** and **S_2** is compared with the absolute value of **S_3**. Take ORZ> for example. If the comparison result is that the absolute value of the difference between **S_1** and **S_2** is greater than the absolute value of **S_3**, the condition of the instruction is met. If the comparison result is that the absolute value of the difference between **S_1** and **S_2** is less than or equal to the absolute value of **S_3**, the condition of the instruction is not met.
3. The instruction ORZ※ is connected to a contact in parallel.

API No.	16-bit instruction	32-bit instruction	Comparison result	
			On	Off
302	ORZ>	**D**ORZ>	$\|S_1 - S_2\| > \|S_3\|$	$\|S_1 - S_2\| \leqq \|S_3\|$
303	ORZ>=	**D**ORZ>=	$\|S_1 - S_2\| \geqq \|S_3\|$	$\|S_1 - S_2\| < \|S_3\|$
304	ORZ<	**D**ORZ<	$\|S_1 - S_2\| < \|S_3\|$	$\|S_1 - S_2\| \geqq \|S_3\|$
305	ORZ<=	**D**ORZ<=	$\|S_1 - S_2\| \leqq \|S_3\|$	$\|S_1 - S_2\| > \|S_3\|$
306	ORZ=	**D**ORZ=	$\|S_1 - S_2\| = \|S_3\|$	$\|S_1 - S_2\| \neq \|S_3\|$
307	ORZ<>	**D**ORZ<>	$\|S_1 - S_2\| \neq \|S_3\|$	$\|S_1 - S_2\| = \|S_3\|$

4. A 32-bit counter (C200~C255) must be used with the 32-bit instruction DORZ※. If it is used with the 16-bit instruction ORZ※, a program error will occur, and the ERROR LED indicator on the PLC will blink.

Program Example:

If M0 is On, or the absolute value of the difference between D10 and D11 is greater than K200, Y0 will be On.



API	Mnemonic	Operands	Function
328	CANRS	S_1 S_2 S_3 D_1 D_2	User-defined CAN BUS communication sending and receiving

OP \ Type	Bit Devices				Word devices											Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CANRS: 11 steps
S_1					*	*							*			
S_2													*			
S_3													*			
D_1													*			
D_2			*													

PULSE							16-bit							32-bit						
ES	EX	EC	EC3-8K	SX	EH3	SV2	ES	EX	EC	EC3-8K	SX	EH3-L	SV2	ES	EX	EC	EC3-8K	SX	EH3	SV2

Explanations:

1. User-defined CAN BUS communication sending, applicable for MPU with CAN BUS communication port, e.g. ES2-C or MPU connected with a left-side communication module, e.g. DVPCOPM-SL.
2. There is no limitation on the times of using this instruction in the program, but only one instruction will be executed at a time. If you start more than two instructions at the same time, PLC executes the first scanned instruction in the program.
3. This instruction is complied with communication protocols of CAN BUS 2.0A (ID 11-bits) (Arbitration) and 2.0B (ID 29-bits). The default is 2.0B (M1620=OFF). Set the M1620 to ON when the PLC is firstly supplied with power, if you need to change the communication protocol to 2.0A. This communication protocol can only be set when the PLC is from Stop to Run.
4. Use flag M1621 to set the mode of Master/Slave. M1621=OFF: Master mode and that means sending first and then receiving. M1621=ON: Slave mode and that means receiving first and then sending.
5. **S_1** sets the communication port number. When DVPCOPM-SL is installed on the left-side of the PLC as the first module, its number is K100; the second one is K101; the eighth one is K107 and so on. If the PLC CPU is ES2-C, its built-in communication port number is K0.
6. **S_2** is the ID of the transmitted message and data length. According to 2.0A or 2.0B protocol, the transmitted data automatically occupies D buffer registers.

   When 2.0A is selected, **S_2** is 11 bits of ID code with the following data transmission format.

S_2 No.	**S_2**	**S_2**+1
Description	Msg. ID	Data Length

   When 2.0B is selected, **S_2** (Lo-word) and **S_2**+1 (Hi-word) are both 29 bits of ID code.

S_2 No.	**S_2**	**S_2**+1	**S_2**+2
Description	Msg. ID（Lo-word）	Msg. ID（Hi-word）	Data Length

7. The length of the transmitted message should be in the range of K0~K8 with the unit of byte (8bits). If the setting value (<0 or >8) exceeds the range, the instruction will run at the minimum value 0 or the maximum 8. If the length of the transmitted message is 0, the communication mode will automatically change into the slave mode to receive messages without sending out any data. The mode can be used to monitor the communication packet.

8. **S_3** is the starting device where transmitted data are stored and only the following 8 bits of data are used.
   For example, 4 messages are transmitted with D10 as the starting device. See the data transmission sequence as below.

S_3 No.	D10	D11	D12	D13
Description	Data1	Data2	Data3	Data4

9. If **S_1** is the master mode in which the master will wait to receive data after sending data or the slave mode, the received data will be directly stored in the device specified by **D_1**. D100 is specified by D_1 Here See the stored content format.

   2.0A mode setting:

D_1 No.	D100	D101	D102 ~ D109 (Lower 8 bits)
Description	Msg. ID	Data Length	Data1 ~ Data8

   2.0B mode setting

D_1 No.	D100	D101	D102	D103 ~ D110 (Lower 8 bits)
Description	Msg. ID（Lo-word）	Msg. ID（Hi-word）	Data Length	Data1 ~ Data8

   Note: If the Msg. ID to be received need be specified at the stage of receiving data, set the value of **D_1** beforehand based on the 2.0A/2.0B mode. If the Msg. ID is not specified, please clear the value of **D_1** to 0 before receiving data.

10. If **S_1** is the master broadcast mode, the received data will be stored in the device specified by **D_1**. D100 is specified by **D_1** here. See the storage format as below.

    Selecting 2.0A mode: (Here is the introduction of receiving data from 2 slaves. For other data, please increase the Device number specified by **D_1**)

Response sequence	**Data from the first slave**			**Data from the second slave**		**Data from the third slave**
D_1 No.	D100	D101	D102 ~ D109 (Lower 8 bits)	D110	D111~D119	D120~129
Description	Msg. ID	Data Length	Data1 ~ Data8	Msg. ID	Length, Data	ID, Length, Data

Selecting 2.0B mode: (Here is the introduction of receiving data from 1 slaves. For other data, increase the number of **D_1**)

Response sequence	**Data from the first slave**				**Data from the second slave**
D_1 No.	D100	D101	D102	D104 ~ D111 (Lower 8 bits)	D112~D122
Description	Msg. ID （Lo-word）	Msg. ID （Hi-word）	Data Length	Data1 ~ Data8	ID, Length, Data

NOTE: if the Msg. ID of the next slave is 0, it indicates there is no data to be received.

11. When the instruction is set to the slave mode and set to receive after sending (M1621=ON, M1622=OFF), the Msg. ID of **D_1** is the receiving condition on ID. Therefore, if there is no requirements on the receivers, use the broadcast mode instead. When the receiver's ID is met with the set ID, the instruction sends data and after sending is complete, a completion flag will be set in **D_2**. If the sending data length is 0, the instruction does not send data and set a completion flag in **D_2**.
12. When the instruction is set to the slave mode and set to receive only (M1621=ON, M1622=ON), the receiving mode will be broadcast. This mode can be ended when timeout (D1177) occurs (M1623=ON) or when the value in D1175 is exceeding 100 packet limit (M1623=ON). If you still need to receive data when this mode is ended, you can stop executing this instruction for a scan cycle and start this mode again. Every time you reset this mode, the receiving log in D1175 will be cleared.
13. **D_2** is communication completion flag and only M device can be used. When the completion flag is ON, it indicates receiving is complete. The completion flag can be set to ON when the instruction is scanned and the communication is complete. From the status of the completion flag, you can tell if the communication is complete. The status of this flag will be clear each time this instruction is executed. You do not need to clear its status.
14. When the instruction is set to the master mode (M1621=OFF), it is recommended to use it to work with D1177 to set the communication timeout. If the communication packet has not been received fully within the specified period of time, the M1623 will be ON. The setting range for timeout is 0-3000 (default 200) and the unit is ms. If the receiving timeout time is set to 0, it indicates that the communication timeout is not limited and the status can be applied to the slave mode.

15. Descriptions on the Flags / Devices

Flags / Devices	Default	Descriptions
M1620	OFF	OFF→ CAN V2.0B protocol ON → CAN V2.0A protocol
M1621 / M1622	OFF/OFF	OFF/OFF → master mode: waiting to receive after sending; if you only need to send data, you can stop executing this instruction in the next scan. After sending is done, the slave response time should be longer than a scan cycle. OFF/ON → master mode: after sending in broadcast mode, receives data from multiple slaves until timeout occurs. ON/OFF → slave mode: sending data, after receiving is done. ON/ON → slave mode: only receiving in broadcast mode without responses
M1623	OFF	ON: communication error; PLC clears this flag when you start the instruction again.
D1175	0	The accumulated packet number (slave number) in the broadcast mode; this number will be accumulated during execution. You can use this number when the completion flag is ON. Up to 100 slaves can be counted, when exceeding 100, the program does not save and stops counting.
D1177	200	Timeout setting; the unit is ms. When the value is set to 0, it indicates this function is disabled until this instruction stops executing. When the mode is in master broadcast, the timeout value cannot be 0. If the timeout value is 0, the system automatically adjusts this value to 200. When timeout occurs, it indicates the broadcast communication is over.

16. The instruction supports the following series and firmware versions

Series	12SA2/ 20SX2	12SE	32ES2-C	SV2/EH3-L	COPM-SL
FW Version	V2.89	V1.87	V3.49	V2.05	V1.36

17. Here is the CAN BUS format and every bit of content for Msg. ID is explained as below.

As 2.0A protocol is selected and the value of **S_2** is H0123, the Msg. ID content is shown in the following table.

Bit No.	15 ~ 11	10 ~ 8	7 ~ 4	3 ~ 0
S_2 value (16bits)	-	1	2	3

As 2.0B protocol is selected, the value of **S_2** is set to H1234 (Lo-word) and **S_2**+1 is H0567 (Hi-word), the Msg. ID content is shown in the following table.

Bit No.	31 ~ 29	28	27 ~ 24	23 ~ 20	19 ~ 16	15 ~ 0
S_2 value (32bits)	-	0	5	6	7	1234

Example 1

System set: DVP12SA211T + DVPCOPM-SL

Mode: Master mode (receiving after sending)

MBB device Diagnostic description as below

Example of Diagnostic Session:

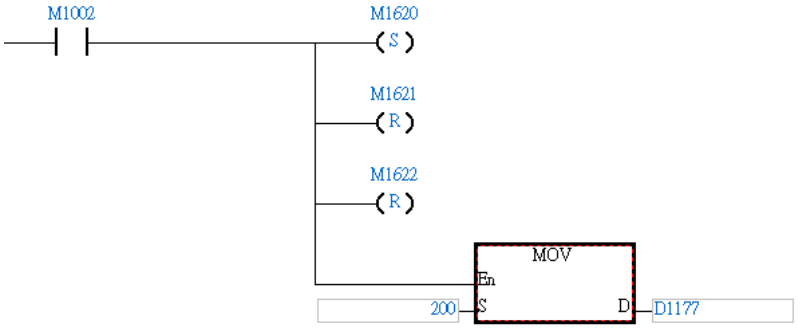
Following is an example of a diagnostic session to write the heartbeat address to 192 (CSM Address).

#	MsgId	Dir	B0	B1	B2	B3	B4	B5	B6	B7	Description
1	0x700	TX	05	2E	FD	01	00	C0	--	--	DID Write Request
2	0x709	RX	03	6E	FD	01	--	--	--	--	Positive Response

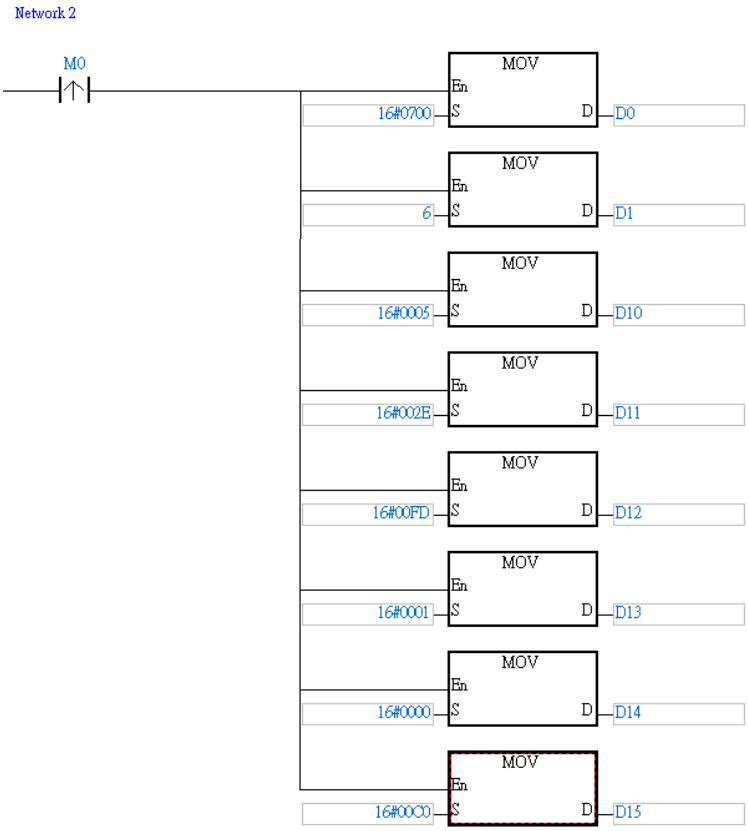
PLC program design:

Step 1) SET M1620 → 2.0A protocol

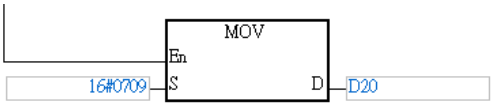
Step 2) RST M1621 & M1622 → Master mode; receiving after sending; set timeout to 200 ms



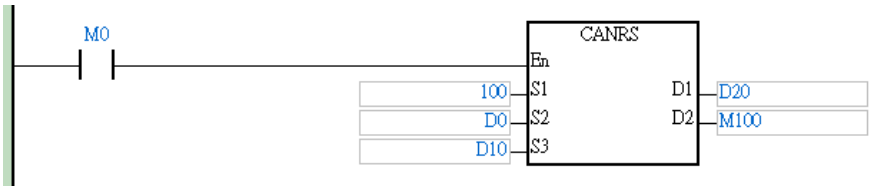
Step 3) LDP M0 → set up MsgId (0x700), data length and data



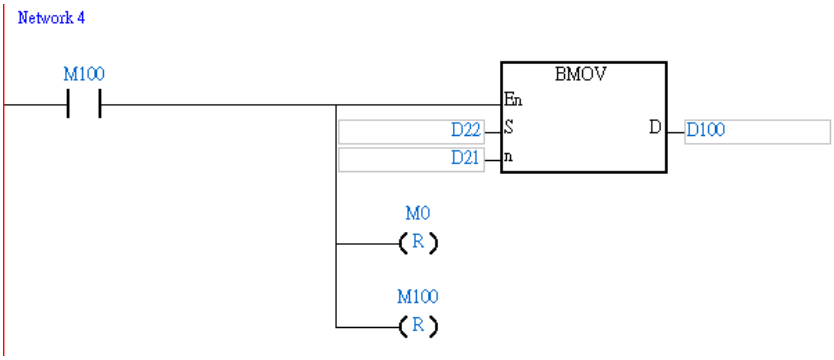
Step 4) LDP M0 → Msg. ID: 0x709



Step 5) LD M0 → use CANRS instruction to set the first left-side module COPM-SL to send data



Step 6) after receiving data is complete, M100 will be ON; stop executing CANRS instruction (RST M0).



Example 2

System set: DVP12SA211T + DVPCOPM-SL

Mode: Master mode (receiving data from all slaves after sending data in broadcast mode)

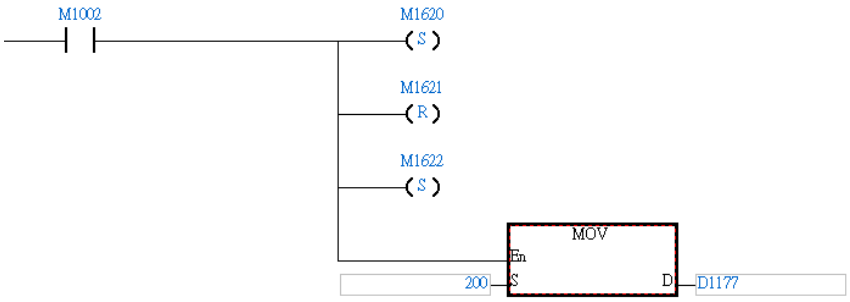
Communication packets:

Identifier	Type	Length	Data
50	Standard	4	11 00 00 00
201	Standard	8	12 FE 86 A4 89 08 87 78
200	Standard	8	12 80 8F 11 8F 18 8F 18
202	Standard	8	12 7D 8E BB 8E F0 8E D8
301	Standard	8	89 08 88 50 87 C8 86 A8
300	Standard	8	8F 1C 8F 14 8F 1C 8F 14
302	Standard	8	8E DC 8E BC 8E F4 8E E4
400	Standard	8	8F 14 8F 1C 8F 1C 8F 1C
401	Standard	8	86 A0 87 08 86 F8 87 48
402	Standard	8	8E DC 8E E4 8E EC 8E E4
500	Standard	8	00 00 00 00 00 00 4F 74
501	Standard	8	00 00 00 00 00 00 4B 98
502	Standard	8	00 00 00 00 00 00 4F 54

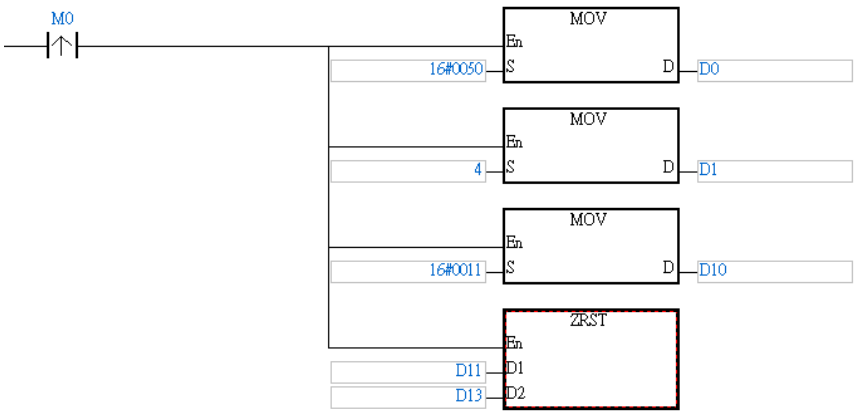
PLC program design:

Step 1) SET M1620 → 2.0A protocol

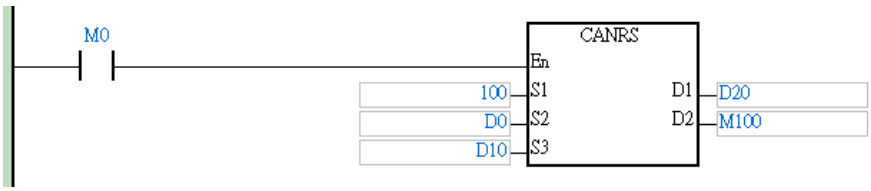
Step 2) RST M1621 and SET M1622 → Master mode; receiving packets from all slaves after sending in broadcast mode; set timeout to 200 ms; if no packets is received in a period of 200 ms, the communication is over.



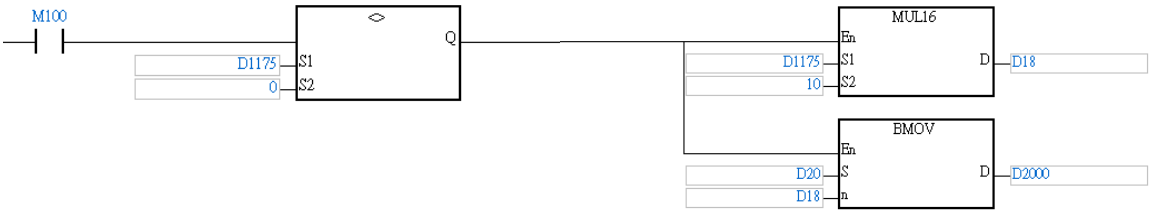
Step 3) LDP M0 → set up MsgId (0x050), data length and data



Step 4) LD M0 → use CANRS instruction to set the first left-side module COPM-SL to send data



Step 5) after receiving data is complete, M100 will be ON; check if the value in D1175 is NOT 0. When there is any value in D1175 other than zero, it indicates D20 has received responses from the slaves.



Example 3

System set: DVP12SA211T + DVPCOPM-SL

Mode: Slave mode (receiving first, if the set ID is met, it responds to master)

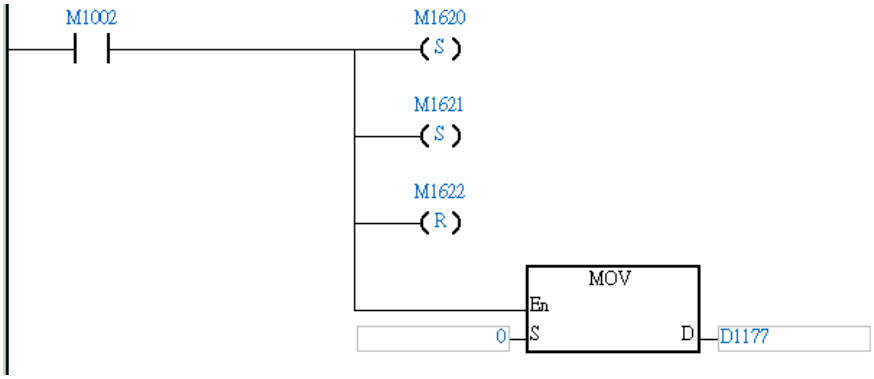
Slave ID is 0x0012 and when the packet contents are in hexadecimal format:

Identifier	Type	Length	Data	Description
012	standard	1	04	Master sending contents
012	standard	4	11 22 33 44	Slave's responses

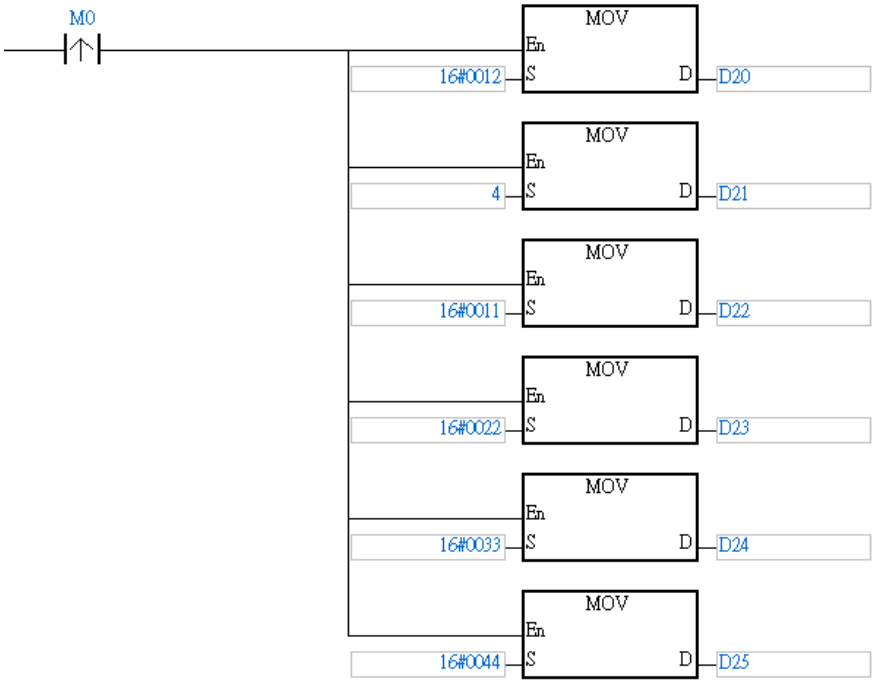
PLC program design:

Step 1) SET M1620 → 2.0A protocol

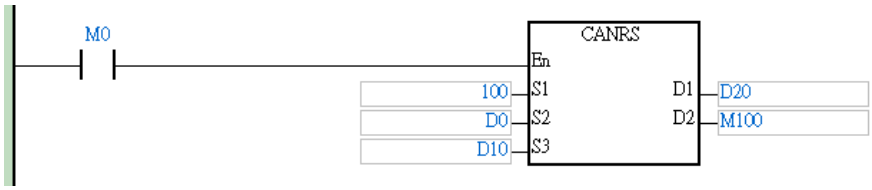
Step 2) RST M1621 and SET M1622 → Slave mode; receiving data from all slaves, if the set ID is met, it responds to master. In receiving mode, the timeout function is not available.



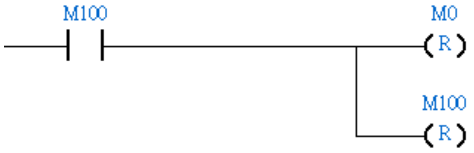
Step 3) LDP M0 → set up MsgId (0x012) and the responses



Step 4) LD M0 → use CANRS instruction to set the first left-side module COPM-SL to respond



Step 5) If M100 is ON, stop executing CANRS instruction.



Note: If Master is going to send data again, you can start executing another CANRS instruction when M100 is ON. Or enter a new ID in D20 and start executing CANRS instruction again.

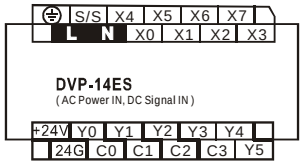
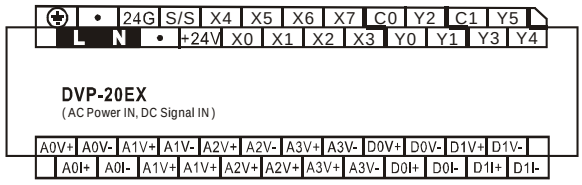
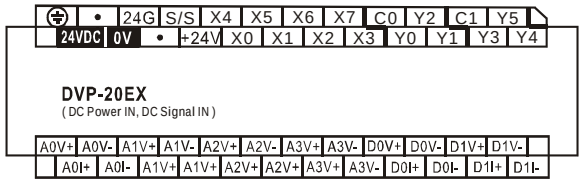
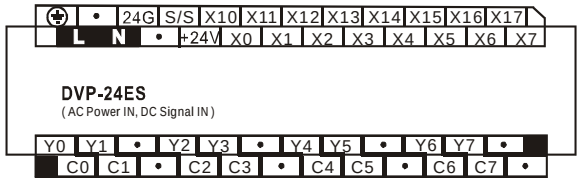
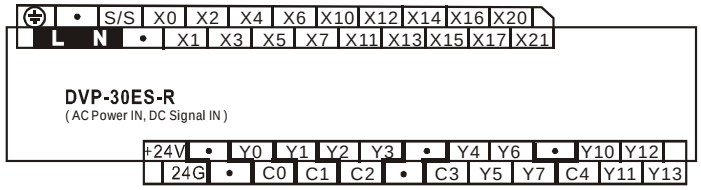
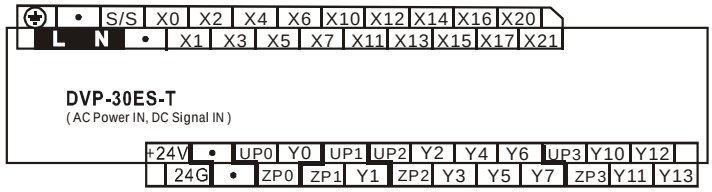
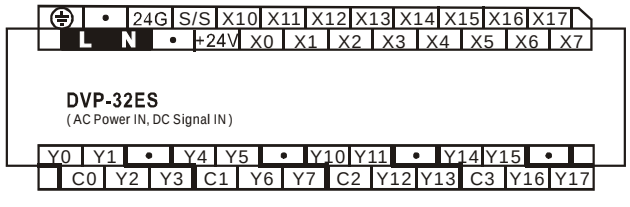
11.1 Appendix A: Table for Self-detecting Abnormality

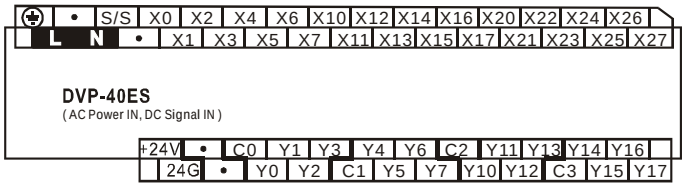
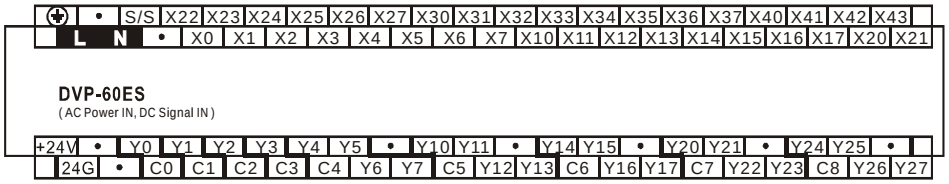
When you encounter abnormality using the product, you can analyze the problem first by doing the self detections below.

Abnormality	Possible cause	Suggested correction
Output point abnormality	Loosened terminal block	Check if the removable terminal block is loosened.
	The input counting specification may not match the pulse output frequency of the PLC model in use.	Check if the hardware is normal by low-frequency pulse counting.
Communication abnormality	The length of communication cable	Make sure the RS-232 cable is at least 3 meters long to ensure normal communication (specification unknown).
	Incorrect communication protocol or address setting	Broadcast from station 0 first by RS-232 to search for communication protocol and address and later confirm by RS-485 communication.
Extension module unable to work	Poor connection or MPU problem	Make sure the MPU is tightly connected to the extension module and compare to make sure whether the problem lies in the MPU or the extension module.
Counter (input point) abnormality	The applicable frequency exceeds the maximum bandwidth.	The frequency should be within the allowed PLC specifications.
ERROR LED flashes	Incorrect program syntax	Record the error code first, and write whether the ERROR LED should not flash anymore by syntactically correct program. Model with battery should be checked whether the time of RTC is correct (not being correct means the battery might once be unattached, causing reset of time).
L.V. LED On	Low input power supply	Make sure the power supply voltage is normal.
RUN LED Off after execution	There is no program inside the new PLC, resulting in misjudgment.	Write in the program first.

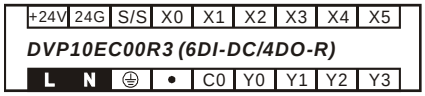
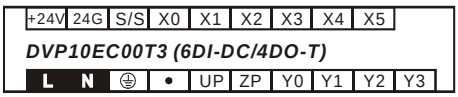
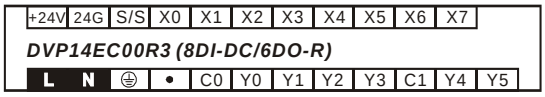
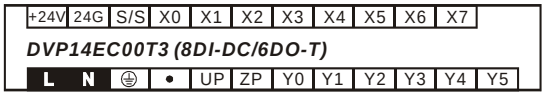
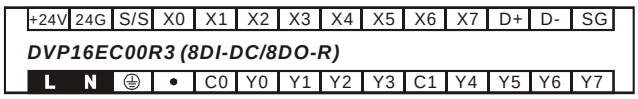
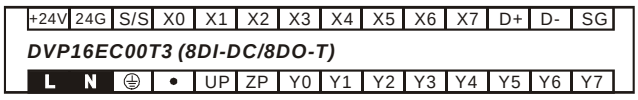
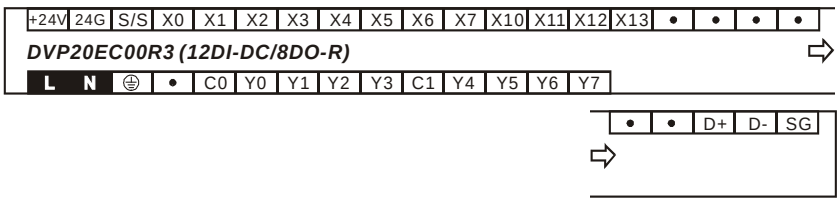
11.2 Appendix B: MPU Terminal Layout

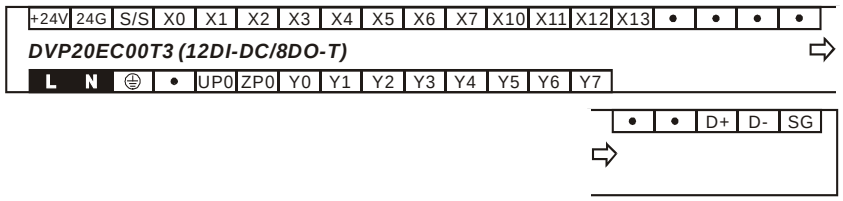
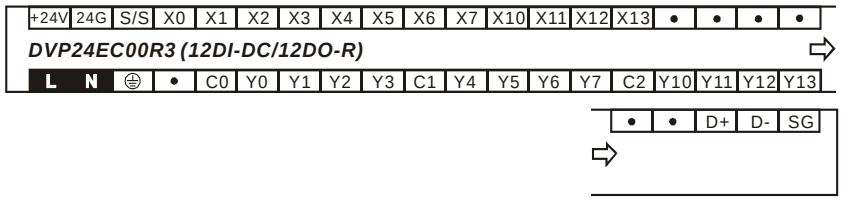
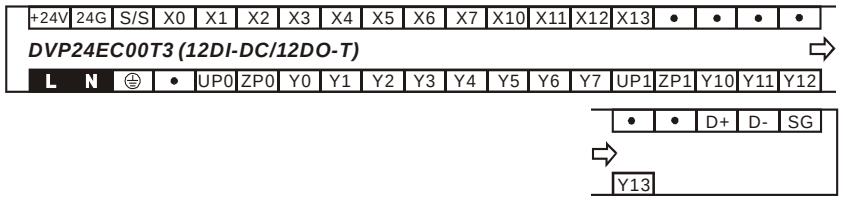
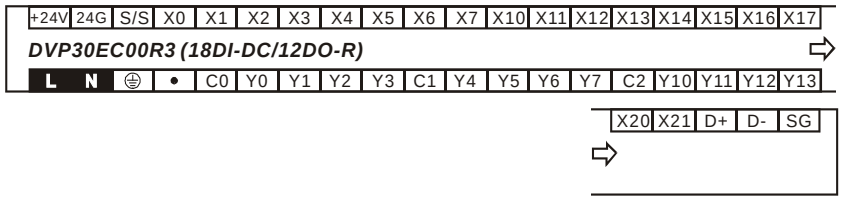
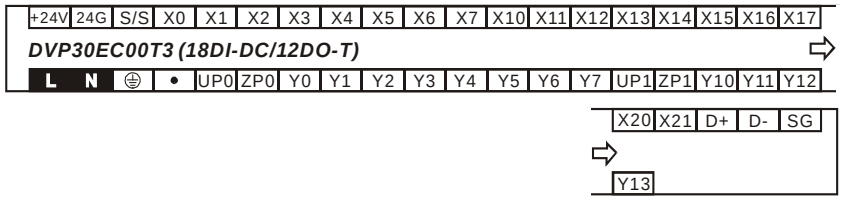
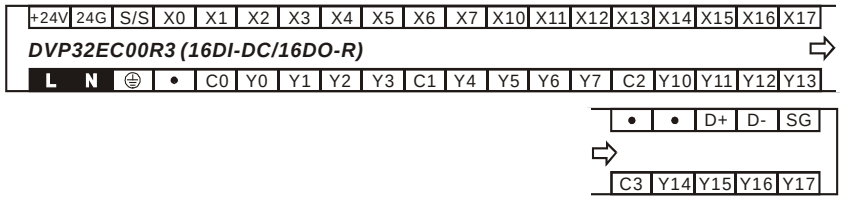
■ For ES series MPU

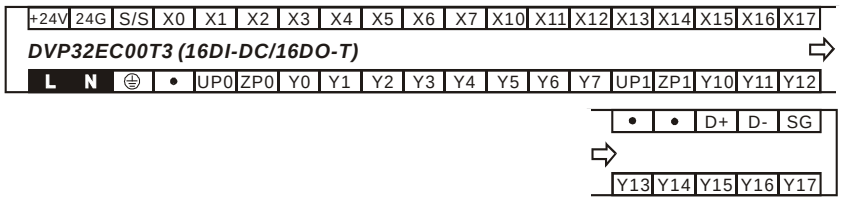
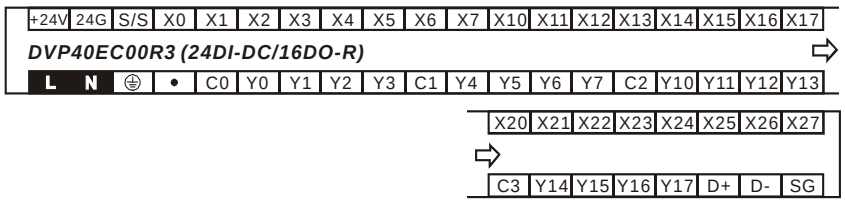
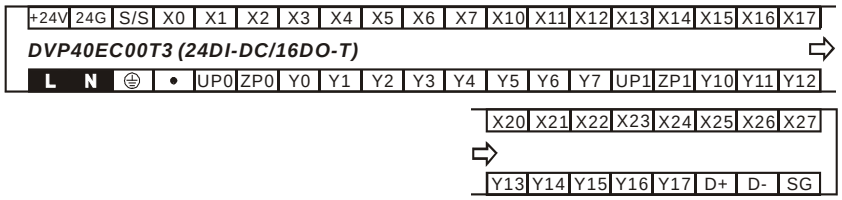
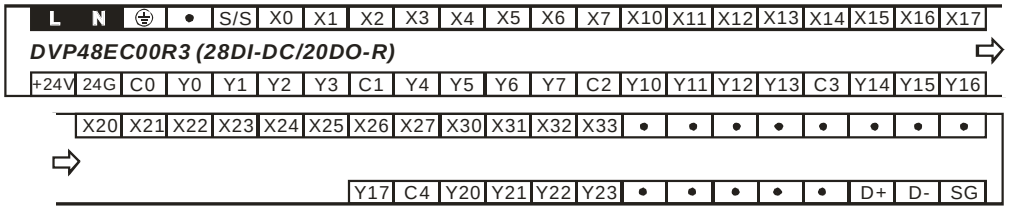
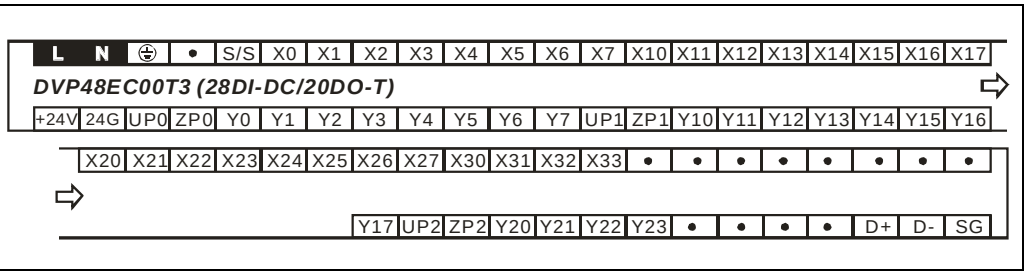
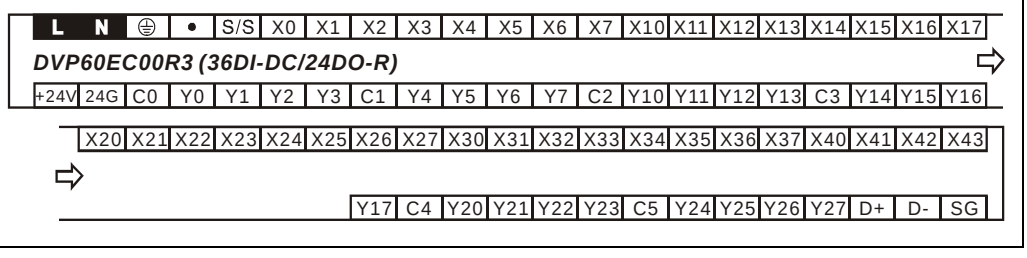
DVP14ES00R2/T2	
DVP20EX00R2/T2	
DVP20EX11R2/T2	
DVP24ES00R2/T2	
DVP30ES00R2	
DVP30ES00T2	
DVP32ES00R2/T2	

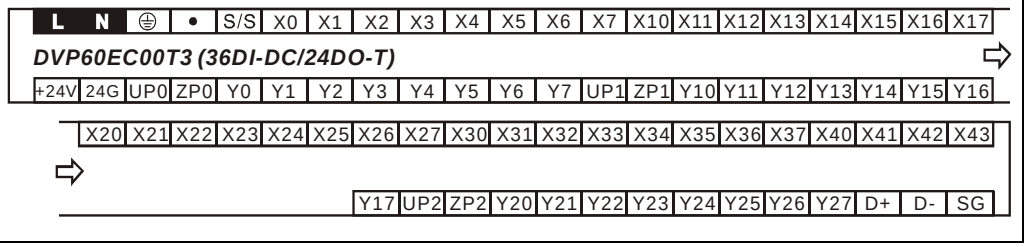
DVP40ES00R2/T2	
DVP60ES00R2/T2	

■ For EC/EC3-8K series MPU

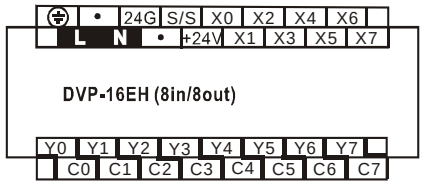
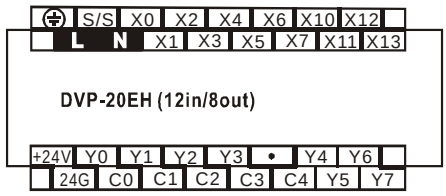
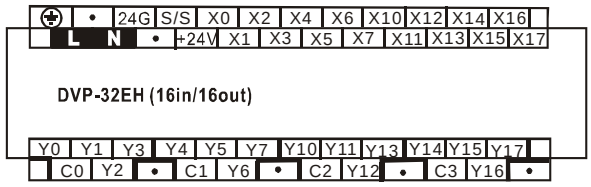
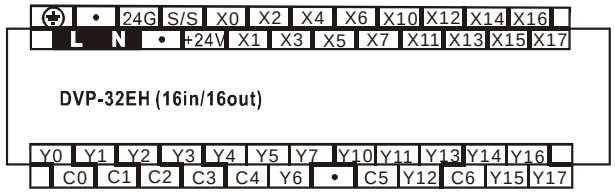
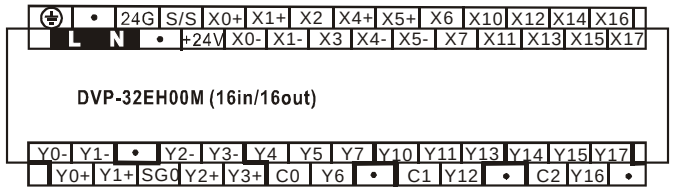
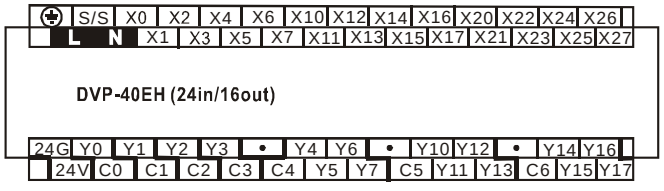
DVP10EC00R3	
DVP10EC00T3	
DVP14EC00R3	
DVP14EC00T3	
DVP16EC00R3	
DVP16EC00T3	
DVP20EC00R3	

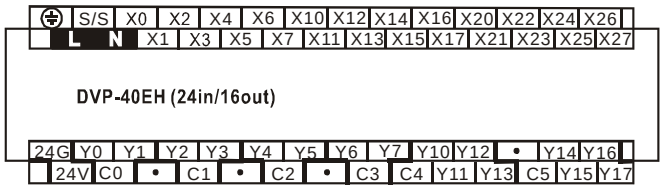
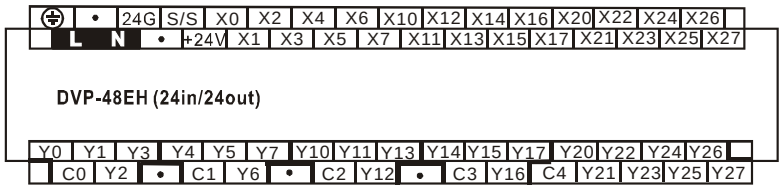
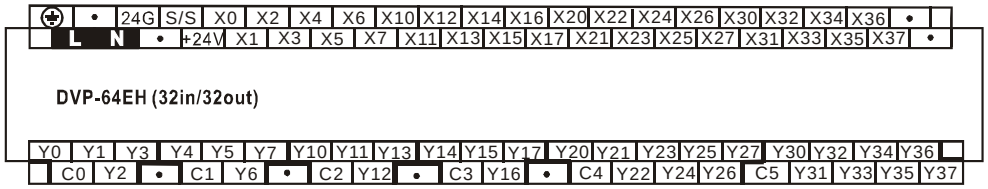
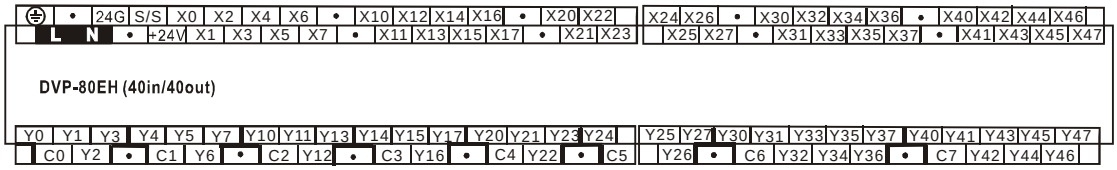
DVP20EC00T3	
DVP24EC00R3	
DVP24EC00T3	
DVP30EC00R3	
DVP30EC00T3	
DVP32EC00R3	

DVP32EC00T3	
DVP40EC00R3	
DVP40EC00T3	
DVP48EC00R3	
DVP48EC00T3	
DVP60EC00R3	

DVP60EC00T3	

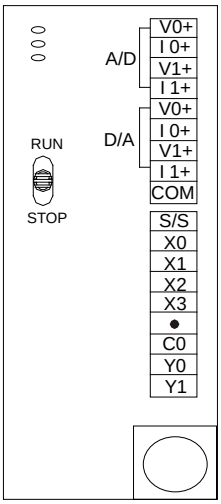
■ For EH2/EH3 series MPU

DVP16EH00R2/T2 DVP16EH00R3/T3	
DVP20EH00R2/T2 DVP20EH00R3/T3	
DVP32EH00R2 DVP32EH00R3	
DVP32EH00T2 DVP32EH00T3	
DVP32EH00M2 DVP32EH00M3	
DVP40EH00R2 DVP40EH00R3	

DVP40EH00T2 DVP40EH00T3	
DVP48EH00R2/T2 DVP48EH00R3/T3	
DVP64EH00R2/T2 DVP64EH00R3/T3	
DVP80EH 00R2/T2 DVP80EH 00R3/T3	

- For SX series MPU

DVP10SX

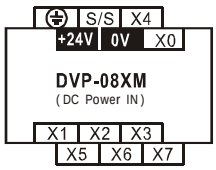
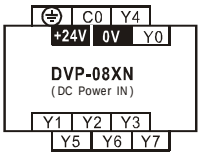
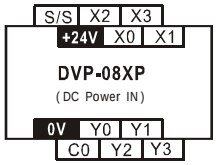
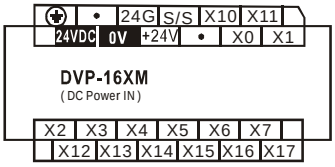
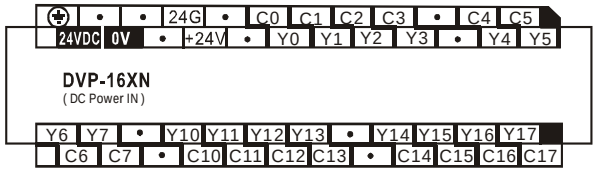
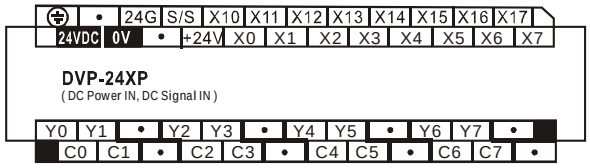
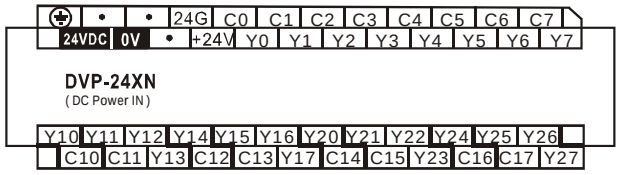
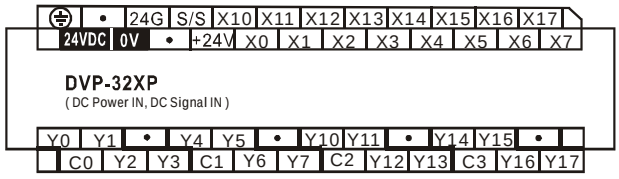


■ For SV/SV2 series MPU

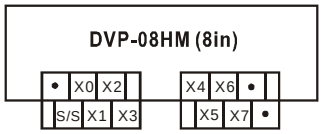
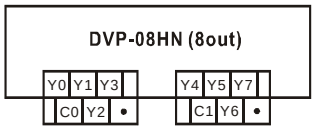
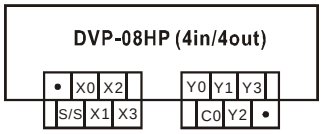
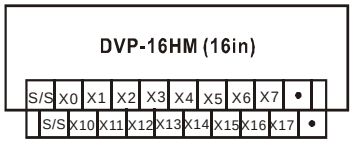
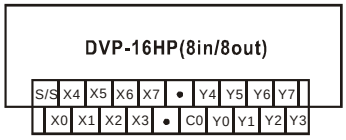
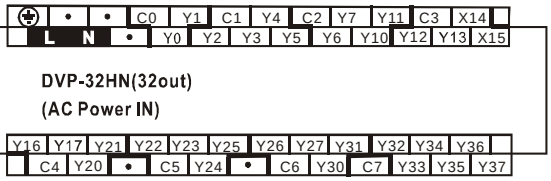
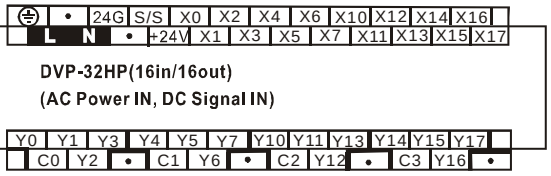
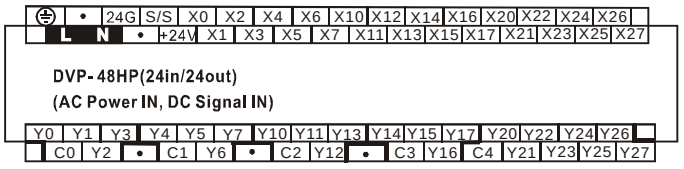
Models	Power	Input		Output		I/O									
		Points	Format	Points	Format	Relay		NPN				PNP			
								28SV		24SV2					
DVP28SV11R	24 VDC	16	Sink or Source	12	Relay	S/S	C0	S/S	C0	S/S	C0	S/S	UP0		
						X0	Y0	X0	Y0	X0	Y0	X0	ZP0		
DVP28SV11R2		16		12		X1	Y1	X1	Y1	X1	Y1	X1	Y0		
						X2	Y2	X2	C1	X2	C1	X2	Y1		
DVP28SV11T		16		12	NPN	X3	●	X3	Y2	X3	Y2	X3	Y2		
						X4	C1	X4	Y3	X4	Y3	X4	Y3		
DVP28SV11T2		16		12		X5	Y3	X5	C2	X5	C2	X5	Y4		
						X6	Y4	X6	Y4	X6	Y4	X6	Y5		
DVP24SV11T2		10		12		X7	Y5	X7	Y5	X7	Y5	X7	Y6		
						S/S	C2	S/S	C3	X10	C3	S/S	Y7		
DVP28SV11S2		16		12	PNP	X10	Y6	X10	Y6	X11	Y6	X10	●		
						X11	Y7	X11	Y7	●	Y7	X11	●		
						X12	Y10	X12	●	V0+	●	X12	UP1		
						X13	●	X13	C4	I0+	C4	X13	ZP1		
						X14	C3	X14	Y10	VI0-	Y10	X14	Y10		
						X15	Y11	X15	Y11	V1+	Y11	X15	Y11		
						X16	Y12	X16	Y12	I1+	Y12	X16	Y12		
						X17	Y13	X17	Y13	VI1-	Y13	X17	Y13		

11.3 Appendix C: Terminal Layout for Digital I/O Modules

■ For ES/EX series digital I/O modules

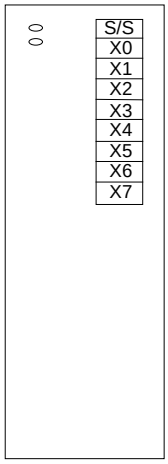
DVP08XM11N		DVP08XN11R/T	
DVP08XP11R/T		DVP16XM11N	
DVP16XN11R/T			
DVP24XP11R/T			
DVP24XN00R/T			
DVP32XP11R/T			

■ For EH3 series digital I/O modules

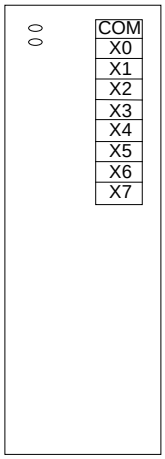
DVP08HM11N		DVP08HN11R/T	
DVP08HP11R/T		DVP16HM11N	
DVP16HP11R/T			
DVP32HN00R/T			
DVP32HP00R/T			
DVP48HP00R/T			

■ For Slim (SX/SV/SV2) series digital I/O modules

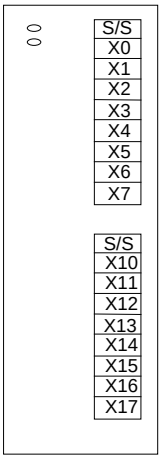
DVP08SM11N



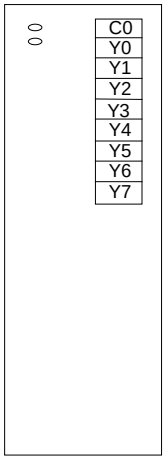
DVP08SM10N



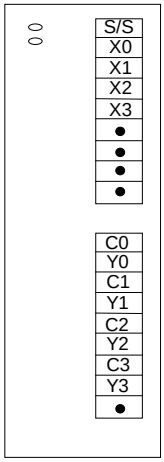
DVP16SM11N



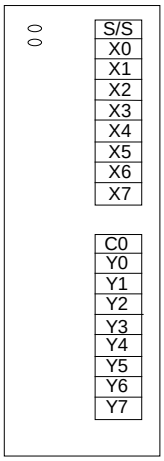
DVP08SN11R
DVP08SN11T



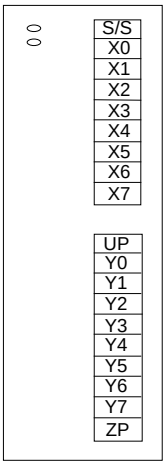
DVP08SP11R
DVP08SP11T



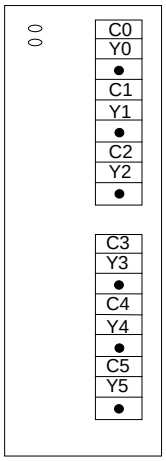
DVP16SP11R



DVP16SP11TS



DVP06SN11R



11.4 Appendix D: Difference between EH2 and EH3

- The capacity of the program is upgraded to 30k Steps, and the number of D devices is increased to 12000.
- The capacity of the memory card has doubled. To prevent the wrong insertion, the new memory card and the old one are not interchangeable
- Some function cards of DVP-EH2 can not be used in DVP-EH3. The function cards which can be used are all communication cards, 02AD card and 02DA card.
- When the position of the basic instruction is below 16k Step, the execution speed does not make any change. When its position is above 16k Step, the execution speed becomes 2.8μs.
- The execution speed of MOV instruction is improved to 4.8μs, and that of other application instructions is also improved by 4~5 times.
- The high-speed input/output functions of DVP-EH3
  - A. The number of external input interruptions is 16. (Please refer to section 2.1.)
  - B. There are 3 sets of masking functions of the interruption. (Please refer to section 2.11.)
  - C. Newly added CSFO instruction has the speed-tracing function and can be used with the manual pulse generator. (Please refer to API 207.)
  - D. Newly added DVSPO and DICF instructions have various speed-changing functions. (Please refer to API 198 and API 199.)
  - E. The zero return instruction (DZRN) has the function of detecting limit switches, stopping at the positive position, seeking Z phase, and outputting the displacement. (Please refer to API 156.)
  - F. The direction outputs of DZRN instruction are used with Y1, Y3, Y5, and Y7. (Please refer to API 156.)
  - G. That special M can set the start and reset functions of C235~C240 is cancelled. The bandwidth of C235~C240can be up to 10 KHz.
- **Other newly added functions of DVP-EH3**
  - A. COM1 card and COM3 card can be masters. COM3 is an independent communication port which does not occupy COM2. The flag of the transmitting function in the master is the same as that in DVP-ES2.
  - B. The GPS instruction is added. (Please refer to API 177)
  - C. We add the m servo convenience instruction ASDRW. (Please refer to API 206.)
  - D. The program can be automatically backed up, and will not disappear even if the battery has run down.
  - E. The second-backup function can store the second program and data.
  - F. We add the basic instructions which are used exclusively for words. For example, BLD, BOUT, and etc. (Please refer to API 266~274)
  - G. The comparison instruction of the floating point number (FLD>=….)is added. (Please refer to API 275~292)
  - H. M1356 can be used in PLC-LINK to designate the station numbers. Only when M1353 is on can D1900~D1931 be used.
  - I. ISPSoft and WPLSoft can set the read-only function of the communication and function of downloading the program.

11.5 Appendix E: Current Consumption of a Slim PLC/an Extension Module

Users can calculate the maximum current consumed by the combination of a slim PLC and modules by means of the data in the table below.

Current supply and current consumption of a PLC (+24VDC)

Item \ Model	14SS2 11R/T	12SS2 11S	12SA2 11R/T	12SE 11R/T	20SX2 11R/T/S	28SV 11R/T/S/R2/T2/S2
Internal maximum current consumed (mA)	R: 100 T: 50	S: 50	R: 100 T: 70	R: 110 T: 80	R: 220 T: 170 S: 170	R: 210 T: 170 S: 170
Maximum current consumed by the external DIO (A) (The current consumption of all inputs and outputs is calculated.) #1	R: 9.1 T: 3.1	S: 2.1	R: 5.1 T: 2.1	R: 5.1 T: 2.1	R: 9.1 T: 3.1 S: 1.9	R: 18.1 T: 3.8 S: 3.8

#1: The external maximum current consumed is estimated on the basis of a worst condition. It is suggested that users should calculate the maximum current consumed according to the actual arrangement.

Current supply and current consumption of a digital input/output module (+24VDC)

Item \ Model	08SM 11N	08SP 11R/T	08SN 11R/T	08ST 11N	16SM 11N	16SP 11R/T	16SP 11TS
Internal maximum current consumed by the IO-BUS (mA)	15	R: 35 T: 35	R: 55 T: 55	55	25	R: 65 T: 65	30
Maximum current consumed by the external DIO (A)	0.05	R: 5 T: 1.2	R: 5 T: 1.2	0	0.1	R: 5 T: 1.2	T: 2

Item \ Model	32SM11N	32SN11TN
Internal maximum current consumed by the IO-BUS (mA)	40	40
Maximum current consumed by the external DIO (A)	0.16	2

Current consumption of a special input/output module (+24VDC)

A special input/output module must be supplied with +24VDC power.

Item \ Model	04AD-S	06AA-S	04DA-S	06XA-S	04PT-S	04TC-S	01PU-S
Internal maximum current consumed by the IO-BUS (mA)	30	30	30	30	30	30	30
Maximum current consumed by the external AIO (mA)	83	83	167	83	83	83	105

Current consumption of a left-side high-speed special module (+24VDC)

Item \ Model	EN01-SL	COPM-SL	DNET-SL	04AD-SL	04DA-SL	02LC-SL	01LC-SL
Internal maximum current consumed by the IO-BUS (mA)	60	50	50	40	40	40	40
Maximum current consumed by the external AIO (mA)	0	0	0	15	80	125	125

Calculating the maximum current consumed by a system

Example: 28SV2 + 16SP + 04AD-S + 04TC-S + EN01-SL

The power module optionally purchased is DVPPS02. (It supplies 2A current.)

Model	Internal current consumption	External current consumption
DVP28SV11T2	170mA	3.8A
DVP16SP11R	65mA	5A
DVP04AD-S	30mA	83mA
DVP04TC-S	30mA	83mA
DVPEN01-SL	60mA	0

Maximum current consumed: Internal → 170 + 65 + 30 + 30 + 60 = 355 (mA) < 2A Pass

External → 3.8A + 5A + 83mA + 83mA = 9A > 2A Not pass

Conclusion: The 2A current supplied by DVPPS02 is sufficient for the PLC and the special modules. If the external I/O terminals are connected to loads, it is suggested that users should purchase an extra power module.

11.6 Appendix F: Current Consumption of an EH3 Series PLC/an Extension Module

Users can calculate the maximum current consumed by an EH2/EH3 system by means of the data in the table below.

Current supply and current consumption of a PLC (+24VDC)

Item \ Model	16EH00 R□/T□	20EH00 R□/T□	32EH00 R□/T□	40EH00 R□/T□	48EH00 R□/T□	64EH00 R□/T□	80EH00 R□/T□
Internal current supplied#1 (A)	0.5	0.5	0.5	0.5	0.5	1	1
External current supplied#2 (A)	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Internal maximum current consumed (mA)	R: 104 T: 140	R: 104 T: 140	R: 148 T: 180	R: 148 T: 180	R: 190 T: 220	R: 234 T: 260	R: 277 T: 300
Maximum current consumed by the external DIO (A)#3	R: 14 T: 2.6	R: 14 T: 2.6	R: 20 T: 4.7	R: 23 T: 5	R: 25 T: 9	R: 30 T: 13	R: 40 T: 14

#1: Internal current supplied → Internal maximum current consumed + Internal maximum current consumed by the IO-BUS

#2: External current supplied → Maximum current consumed by the external DIO + Maximum current consumed by the external digital input/output module + Maximum current consumed by the external special input/output module (AIO module)

#3: The external maximum current consumed is estimated on the basis of a worst condition. It is suggested that users should calculate the maximum current consumed according to the actual arrangement.

Current supply and current consumption of a digital input/output module (+24VDC)

Item \ Model	08HM 11N	08HP 11R/T	08HN 11R/T	16HM 11N	16HP 11R/T	32HP 11R/T
Internal maximum current consumed by the IO-BUS (mA)	15	35	55	25	65	100
Maximum current consumed by the external DIO (A)	0.05	R: 5 T: 1.2	R: 10 T: 2.4	0.1	R: 10 T: 2.4	R: 20 T:4.8

Item \ Model	32HN 00R	32HP 00R/T	48HP 00R/T
Current supplied (mA)	0	500	
Internal maximum current consumed by the IO-BUS (mA)	20	20	20
Maximum current consumed by the external DIO (A)	40	20	30

Current consumption of a special input/output module (+24VDC)

A special input/output module must be supplied with +24VDC power.

Item \ Model	04AD-H2	04DA-H2	06XA-H2	04PT-H2	04TC-H2	01HC-H2	01PU-H2
Internal maximum current consumed by the IO-BUS (mA)	30	30	30	30	30	30	30
Maximum current consumed by the external AIO (mA)	105	188	145	105	105	125	125

Calculating the maximum current consumed by a system

Example: 32EH00R3 + 16HM11N + 16HP11R + 04AD-H2 + 04DA-H2

Model	Internal current consumption	External current consumption
DVP32EH00R3	148mA	20A
DVP16HM11N	25mA	0.1A
DVP16HP11R	65mA	10A
DVP04AD-H2	30mA	105mA
DVP04DA-H2	30mA	188mA
Maximum current consumed by the system	298mA	30.3A

Maximum current consumed: Internal → 298mA < 500(mA) Pass

The internal current supplied by the PLC is sufficient for the modules.

External → 30.3A > 500(mA) Not pass

The PLC can only supply sufficient power to the three modules 16HM+04AD+04DA.

The other I/O terminals must be supplied with extra power.

11.7 Appendix G: Using Ethernet Communication

The specifications for a DVP series Ethernet port and the functions of a DVP series Ethernet port are listed below.

Specifications for an Ethernet interface:

Item	Specifications
Interface	RJ-45 with Auto MDI/MDIX
Number of ports	1 Port
Transmission method	IEEE802.3, IEEE802.3u
Transmission cable	Category 5e
Transmission rate	10/100 Mbps Auto-Defect
Protocol	ICMP, IP, TCP, UDP, DHCP, SMTP, NTP, MODBUS TCP

Ethernet functions:

Function	Built-in Ethernet port in a DVP-SE series PLC	DVPEN01-SL	DVP-FEN01 (Function card for a DVP-EH3 series PLC)
MODBUS/TCP	Master & Slave	Master & Slave	Master & Slave
Number of servers	16	16	4
Number of clients	8	16	4
Number of data exchanged	8	24	8
RTU mapping	-	4	-
E-mail	-	4	-
SNMP	-	2	-
IP filter	4	8	4

Station Addresses of Ethernet Modules and Control Registers

Station Addresses of Ethernet Modules

Model name	Built-in Ethernet port in a DVP-SE series PLC	DVPEN01-SL	DVP-FEN01 (Function card for a DVP-EH3 series PLC)
FROM/TO station address	K108	See example 1.	K108

Example 1. Suppose a DVP-SV series PLC is connected to three left-side communication modules.

PLC/Module name	DVPEN01-SL	DVPCOPM-SL	DVPEN01-SL	DVP28SV11R
FROM/TO station address	K102	K101	K100	--

DVP-SE Series PLC (Ethernet PLC)

In order to control and monitor Ethernet communication, users can read the data in the control registers listed below by means of the instruction FROM, and write data into the control registers listed below by means of the instruction TO. (Please refer to the explanation of API 78 and that of API 79 in chapter 3 for more information about FROM/TO.)

[Note] Please refer to DVPEN01-SL Manual for more information about control registers.

CR number		Attribute	Register name	Description
HW	LW			
#12	#0	-	Reserved	
	#13	R/W	Enabling the data exchange	Users can set CR#13 to “sending the data” or “not sending the data”.
	#14	R/W	Writing function of the RTU mapping	0: The PLC writes data continually. 1: The PLC writes data when the input changes.
	#15	R/W	Enabling flag for RTU mapping	1: Enable; 0: Disable. Default = 1
	#16	R/W	Connection status of RTU mapping slave	b0: Status of RTU slave 1 b1: Status of RTU slave 2 b2: Status of RTU slave 3 b3: Status of RTU slave 4
	#17	R/W	Execution cycle of the data exchange	Time unit: ms
	#18	-	Reserved	
	#19	R	States of the slaves involved in the data exchange	If the value of a bit is 1, an error occurs in the slave corresponding to the bit. b[0:7] indicate the states of the slaves 1~8 involved in the data exchange.
#86 ~ #20		-	Reserved	
	#87	R/W	IP address setting mode	0: Static IP 1: DHCP
#89	#88	R/W	IP address	When the IP address is 192.168.1.5, the data in CR#89 is 192.168, and the data in CR#88 is 1.5.
#91	#90	R/W	Mask address	When the mask address is 255.255.255.0 the data in CR#91 is 255.255, and the data in CR#90 is 255.0.
#93	#92	R/W	Gateway IP address	When the GIP address is 192.168.1.1, the data in CR#89 is 192.168, and the data in CR#88 is 1.1.
	#94	R/W	Enabling the IP address setting	0: The setting of the IP address is not executed. 1: The setting of the IP address is executed.
	#95	R	IP address setting status	0: The setting is unfinished. 1: The setting is being executed. 2: The setting is complete.
#113 ~ #96		-	Reserved	
	#114	R/W	MPDBUS TCP time-out	Setting up MODBUS TCP time-out (in ms) Default: 3000
	#115	R/W	MODBUS TCP trigger	Setting up whether to send out data in MODBUS TCP mode
	#116	R/W	MODBUS TCP status	Displaying current status of MODBUS TCP mode
#118	#117	R/W	MODBUS TCP destination IP	Setting up destination IP address for MODBUS TCP transaction
	#119	R/W	MODBUS TCP data length	Setting up the data length for MODBUS TCP transaction
#219~#120		R/W	MODBUS TCP data buffer	Data buffer of MODBUS TCP for storing sending/receiving data
#248~#220		-	Reserved	
	#249	R	Sub-version	
	#250	R	Update date	0xC820 (April 8, 2012)
	#251	R	Error code	Displaying the errors. See the error code table for more information.
#255~#252		-	Reserved	
Symbols “R” refers to “able to read data by FROM instrcution”; “W” refers to “able to write data by TO instrcution”.				

DVPEN01-SL (Ethernet Communication Module)

DVPEN01-SL Ethernet communication module				
CR number		Attribute	Register name	Description
HW	LW			
	#0	R	Model name	Set up by the system; read only. Model code of DVPEN01-SL = H'4050
	#1	R	Firmware version	Displaying the current firmware version in hex.
	#2	R	Communication mode	b0: MODBUS TCP mode b1: data exchange mode
	#3	W	E-Mail Event 1 trigger	Set up whether to send E-Mail 1
	#4	W	E-Mail Event 2 trigger	Set up whether to send E-Mail 2
	#5	W	E-Mail Event 3 trigger	Set up whether to send E-Mail 3
	#6	W	E-Mail Event 4 trigger	Set up whether to send E-Mail 4
	#7	R	Status of E-Mail 1, 2	b0 ~ b7: Current status of E-Mail 2 b8 ~ b15: Current status of E-Mail 1
	#8	R	Status of E-Mail 3, 4	b0 ~ b7: Current status of E-Mail 4 b8 ~ b15: Current status of E-Mail 3
	#9	R/W	E-Mail 1 additional message	Filled in by the user, and it will be sent by E-mail.
	#10	R/W	E-Mail 2 additional message	Filled in by the user, and it will be sent by E-mail.
	#11	R/W	E-Mail 3 additional message	Filled in by the user, and it will be sent by E-mail.
	#12	R/W	E-Mail 4 additional message	Filled in by the user, and it will be sent by E-mail.
	#13	R/W	Data exchange trigger	Set up whether to send out data in data exchange mode
	#14	R	Status of data exchange	Displaying current status of data exchange.
	#15	R/W	Enabling flag for RTU mapping	1: Enable; 0: Disable. Default = 0
	#16	R/W	Connection status of RTU mapping slave	b0: Status of RTU slave 1 b1: Status of RTU slave 2 b2: Status of RTU slave 3 b3: Status of RTU slave 4
#24 ~ #17		-	Reserved	
#26	#25	R/W	Destination IP	Destination IP address for data exchange
	#27	-	Reserved	
	#28	R/W	Destination Slave ID	Destination Slave ID for data exchange
#48 ~ #29		R/W	Data transmission buffer	Buffer for transmitted data in data exchange
#68 ~ #49		R	Data receiving buffer	Buffer for received data in data exchange
#69~#80		-	Reserved	
	#81	R/W	Read address for data exchange	Slave transmission buffer address for data exchange
	#82	R/W	Read length for data exchange	Number of registers for read data
	#83	R/W	Received address for data exchange	Buffer address for the receiving Master in data exchange
	#84	R/W	Written-in address for data exchange	Buffer address for the receiving Slave in data exchange
	#85	R/W	Written-in length for data exchange	Number of registers for data transmission
	#86	R/W	Transmission address for data exchange	Master transmission buffer address for data exchange
#110 ~ #87		-	Reserved	
	#111	R/W	8-bit processing mode	Setting up MODBUS TCP Master control as 8-bit mode
	#112	R/W	MODBUS TCP Keep-Alive Time-out	MODBUS TCP Keep-Alive Time-out (s)
	#113	-	Reserved	
	#114	R/W	MODBUS TCP time-out	Setting up MODBUS TCP time-out (in ms)
	#115	R/W	MODBUS TCP trigger	Setting up whether to send out data in MODBUS TCP mode
	#116	R/W	MODBUS TCP status	Displaying current status of MODBUS TCP mode

DVPEN01-SL Ethernet communication module				
CR number		Attribute	Register name	Description
HW	LW			
#118	#117	R/W	MODBUS TCP destination IP	Setting up destination IP address for MODBUS TCP transaction
	#119	R/W	MODBUS TCP data length	Setting up the data length for MODBUS TCP transaction
#219 ~ #120		R/W	MODBUS TCP data buffer	Data buffer of MODBUS TCP for storing sending/receiving data
#248 ~ #220		-	Reserved	
	#251	R	Error code	Displaying the errors. See the error code table for more information.
#255 ~ #252		-	Reserved	

DVP-FEN01 (DVP-EH3 Series Ethernet Communication Card)

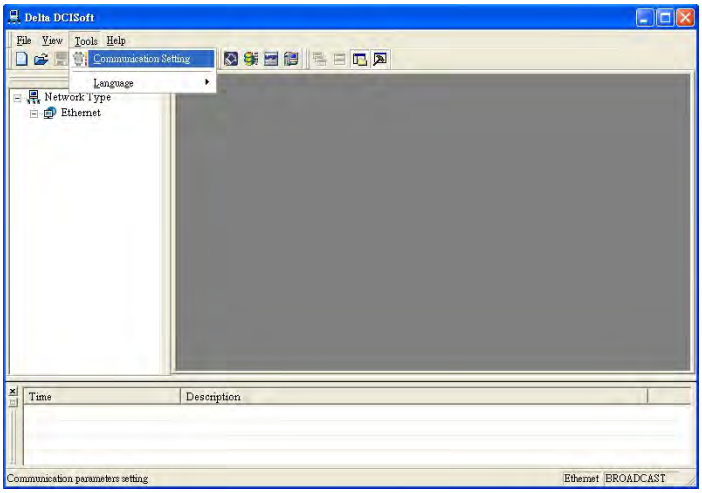
DVP-FEN01 Ethernet communication card				
CR number		Attribute	Register name	Description
HW	LW			
#0		R	Model name	Undefined
#1		R	Firmware version	It adopts the hexadecimal system, and the present firmware version is stored in it.
#2~#12		-	Reserved	
#13		R/W	Enabling the data exchange	Users can set CR#13 to “sending the data” or “not sending the data”.
#16~#14		-	Reserved	
#17		R/W	Execution cycle of the data exchange (ms)	
#18		-	Reserved	
#19		R	States of the slaves involved in the data exchange	b[0:7] indicate the states of the slaves 1~8 involved in the data exchange.
#20~#86		-	Reserved	
#87		R/W	IP address setting mode	0: Static IP 1: DHCP
#89	#88	R/W	IP address	When the IP address is 192.168.1.5, the data in CR#89 is 192.168, and the data in CR#88 is 1.5.
#91	#90	R/W	Mask address	When the mask address is 255.255.255.0 the data in CR#91 is 255.255, and the data in CR#90 is 255.0.
#93	#92	R/W	Gateway IP address	When the GIP address is 192.168.1.1, the data in CR#89 is 192.168, and the data in CR#88 is 1.1.
#94		R/W	Enabling the IP address setting	0: The setting of the IP address is not executed. 1: The setting of the IP address is executed.
#95		R	IP address setting status	0: The setting is unfinished. 1: The setting is being executed. 2: The setting is complete.
#96~#250		-	Reserved	
#251		R	Error status	bit 0: The network is unconnected. bit 3: CR#13 is set to “sending the data”, but the data exchange is not enabled. bit 8: DHCP does not acquire the correct network parameter.
#255~#252		-	Reserved	

Searching for an Ethernet PLC

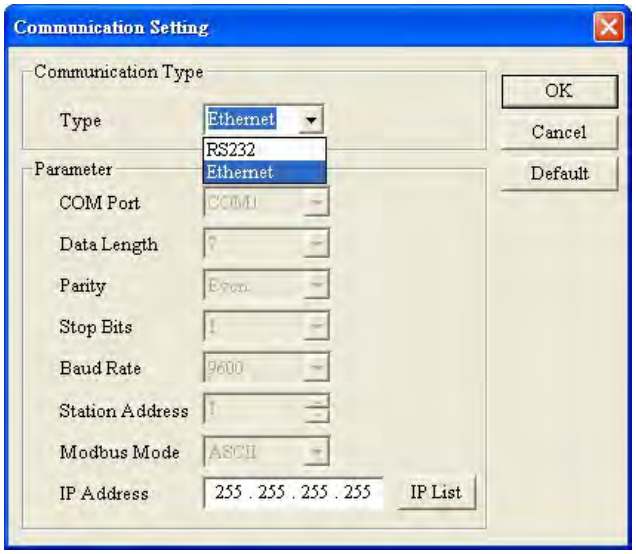
This section introduces how to search for and set an Ethernet PLC by DCISoft. Before you start a setup page, you have to select **Ethernet** in the **Communication Setting** window. Next, you can search by a broadcast, or an IP address. An Ethernet PLC is set up by UDP port 20006; therefore, you have to be aware of the relevant settings of the firewall.

Communication setting

(1) Start DCISoft in your PC, and click **Communication Setting** on the **Tools** menu.

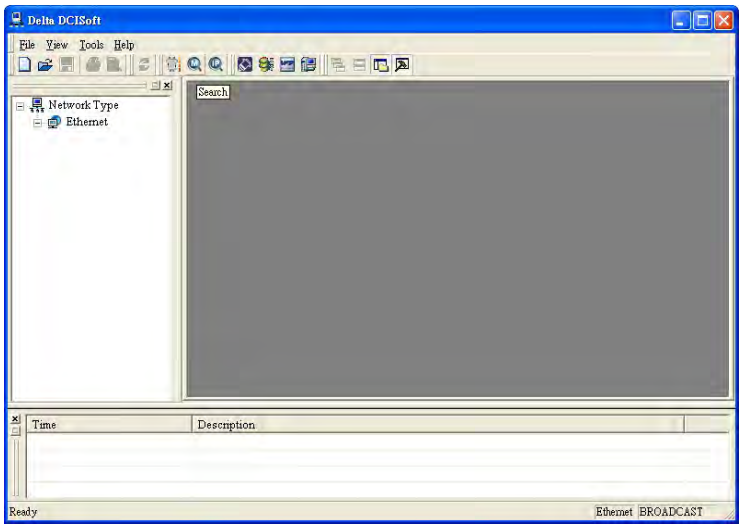


(2) Select **Ethernet** in the **Type** drop-down list box.

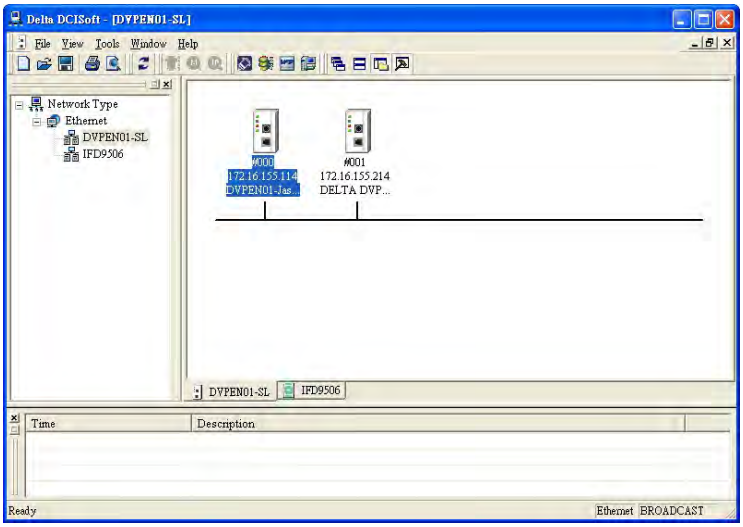


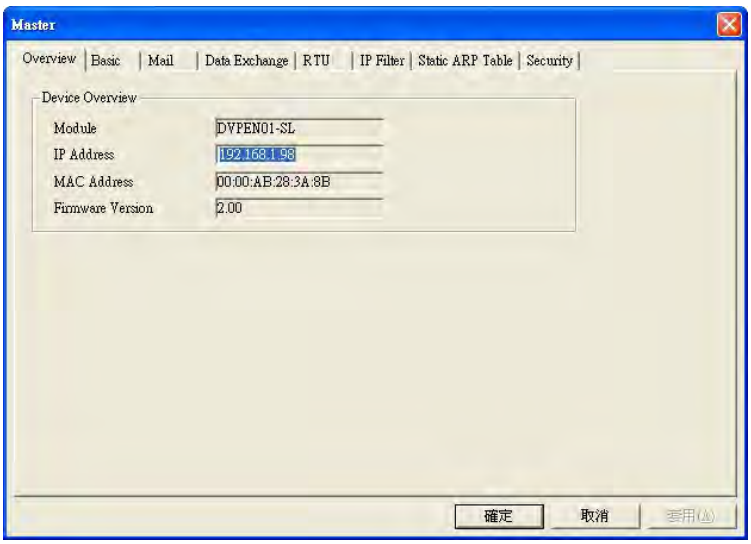
Broadcast Search

(1) Click **Search** on the toolbar in DCISoft to search for all Delta Ethernet products on the network. The window on the left hand side shows the models found, and the window on the right hand side displays the device list of all models.



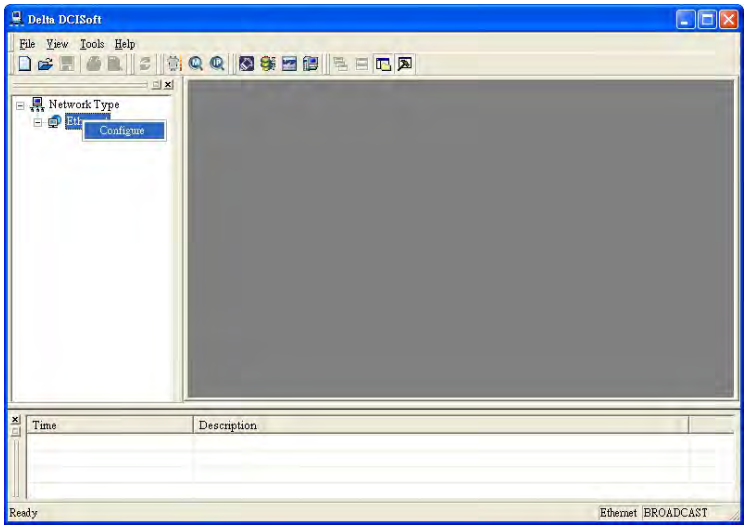
(2) Click a model on the left hand side, and you will see the device list of the model selected on the right hand side. Click the device to be set up to enter the setup page.



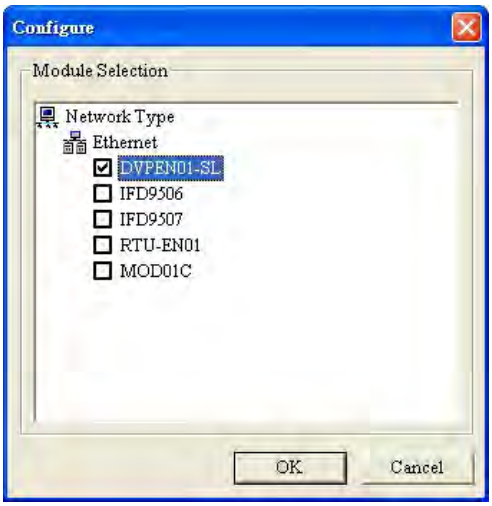


Searching for a Model Specified

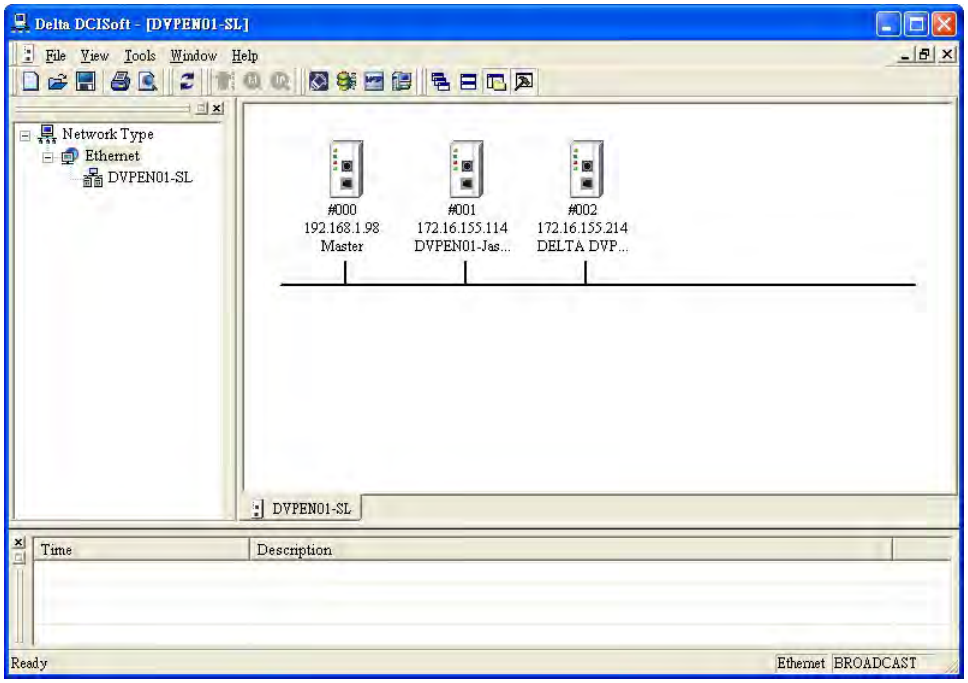
(1) Right-click **Ethernet** in the left hand side window, and click **Configure** to designate a model to be searched for.



(2) After users select a model which will be searched for, they can click **OK** to auto-search for the model on the network. In the window shown below, the **DVPEN01-SL** checkbox is selected.

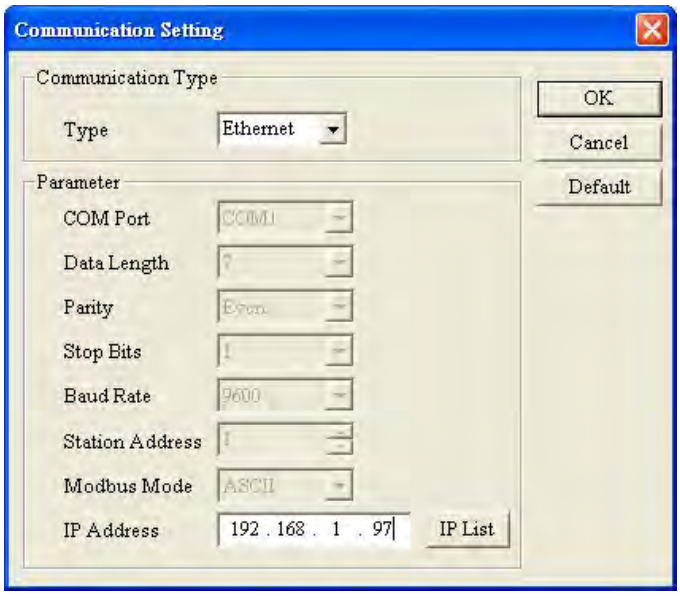


(3) A list of specified devices is in the window. If the users have selected several models, they can view these models.

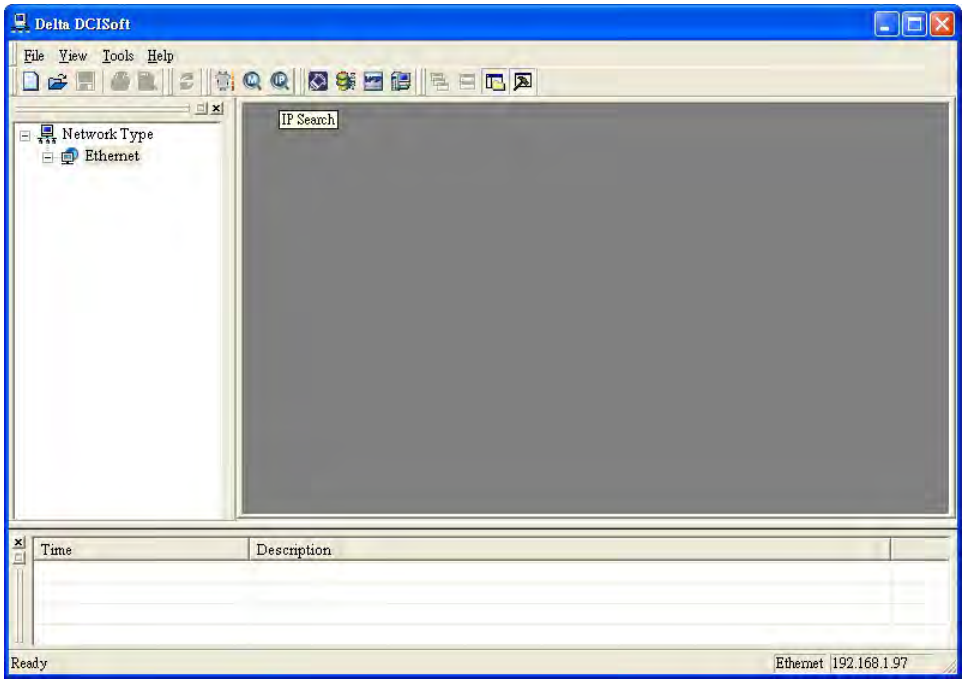


Searching by an IP Address

(1) Select **Ethernet** in the **Type** drop-down list box, type an IP address in the **IP Address** box, and click **OK**.



(2) Click **Search** on the toolbar to start searching for the designated IP address.



(3) The model found will be displayed in the right hand side window. Double-click the device to enter the setup page.

Data Exchange

A Delta Ethernet master can read/write data from/into a slave by means of instructions. It can also read/write data from/into a slave by means of tables. The number of data exchanges that models provide is different. Please refer to the information provided above for more information about the number of data exchanged.

(1) Enable:

Users can enable or disable a data exchange. After a data exchange is enabled, the data will be exchanged.

(2) Enable Condition:

You can select **Always Enable** or **Program Control**. If **Always Enable** is selected, DVPEN01-SL will execute data exchange continuously until the setting in DCISoft is changed. If **Program Control** is selected, DVPEN01-SL will execute data exchange according to the program setting. The internal registers in different models used to enable data exchanges are different. Please refer to section B.2 for more information.

(In DVPEN01-SL, the data exchange is executed if CR#13=2, and the data exchanged is not executed if CR#13=0.)

(3) Station Address-IP Address:

You have to type the IP address of a slave. If the IP address of a slave is 192.168.0.1, and the station number of the slave is 1, you can type 1 in the first **Station Address** cell, select the box in the first **Enable** cell, and type 192.168.0.1 in the first **IP Address** cell.

(4) Master Device, Slave Device, and Quantity:

Reading (←): Initial reception register in a master←Initial transmission register in a slave

Writing (→): Initial transmission register in a master→ Initial reception register in a slave

If a data exchange is enabled, the Ethernet PLC will write data, and then read data.

Quantity: A slave station can send 100 pieces of data at most and receive 100 pieces of data at most simultaneously.

※ If a device which is not a Delta PLC is connected, users can type a hexadecimal four-digit MODBUS absolute position in the **Slave Device** cell.

EtherNet/IP List

EtherNet/IP is a communication protocol defined by ODVA, and is different from the Ethernet mentioned in the previous sections. DVP-12SE series supports the EtherNet/IP slave communication protocol. The other DVP series PLCs can communicate with products related to EtherNet/IP through IFD9507 (an EtherNet/IP-MODBUS converter). The EtherNet/IP objects which are supported are described below.

EtherNet/IP Information Supported by DVP-12SE series PLCs

(1) Object list

Object Name	Class Code	#of Instance
Identity	0x01	7
Message Router	0x02	NA
Assembly	0x04	7
Connection Manager	0x06	NA
X input	0x64	256
Y output	0x65	256
T Timer	0x66	256
M Relay	0x67	4096
C Counter	0x68	256
D Register	0x69	12000
TCP/IP Interface	0xF5	6
Ethernet Link	0xF6	3

(2) Data types

8-bit	16-bit	32-bit	64-bit
USINT	WORD	UDINT	ULINT
SINT	UINT	DWORD	LINT
BYTE	INT	DINT	

(3) Error codes

Value	Name	Description
0	Success	Success
0x01	Connection Failure	The forwarding function can not be enabled.
0x04	Path Segment Error	The segment type is not supported. (ref. V1 C-1.4)
0x05	Path Destination Unknown	The instance is not supported.
0x08	Service Not Supported	The service (Get or Set) is not supported.
0x09	Invalid Attribute Value	The value written is incorrect.
0x0E	Attribute Not Settable	The setting of the attribute is not allowed.
0x13	Not Enough Data	The length of the data written is too short.
0x14	Attribute Not Supported	The attribute is not supported.
0x15	Too Much Data	The length of the data written is too long.
0x16	Object Not Exist	The object is not supported.
0x20	Invalid Parameter	The service parameter is not supported. (ref. V1 5-2.3.1)
0x26	Path Size Invalid	Incorrect item length

EtherNet/IP Objects Supported by DVP-12SE series PLCs

(1) Identity Object (0x01)

Instance: 0x01

Attribute	Name	Access	Data Type	Value
0x01	Vendor ID	Get	UINT	799 (Delta Electronics, inc.)
0x02	Device Type	Get	UINT	14 (Programmable Logic Controller)
0x03	Product Code	Get	UINT	0x0015
0x04	Revision	Get	STRUCT of:	1.32
	Major		USINT	0x01
	Minor		USINT	0x20
0x05	Status	Get	WORD	0 (Owned)
0x06	Serial Number	Get	UDINT	
0x07	Product Name	Get	SHORT_STRING	DVP12SE

(2) Message Router (0x02)

Instance: 0x01

Attribute	Name	Access	Data Type	Value
0x01	Not Support	NA	NA	NA

(3) Assembly (0x04)

Explicit message

Conformance Test is not supported.

Instance	Attribute	Name	Access	Data Type	Data
0x65	0x03	D Block 1	Set	10 words	D500~D509
0x66		D Block 2	Set	30 words	D510~D539
0x67		D Block 3	Set	60 words	D540~D599
0x68		D Block 4	Set	100 words	D600~D699
0x69		D Block 5	Set	100 words	D700~D799
0x6A		D Block 6	Set	100 words	D800~D899
0x6B		D Block 7	Set	100 words	D900~D999

(4) X input (0x64)

Instance	Attribute	Name	Access	Data Type
1	0x64	X0	Get	BYTE
2	0x64	X1	Get	BYTE
……				
256	0x64	X377	Get	BYTE

(5) Y output (0x65)

Instance	Attribute	Name	Access	Data Type
1	0x64	Y0	Set	BYTE (0x00 or 0x01)
2	0x64	Y1	Set	BYTE (0x00 or 0x01)
……				
256	0x64	Y377	Set	BYTE (0x00 or 0x01)

(6) T timer (0x66)

Instance	Attribute	Name	Access	Data Type
1	0x64	T0	Set	INT
2	0x64	T1	Set	INT
……				
256	0x64	T255	Set	INT

Instance	Attribute	Name	Access	Data Type
1	0x65	T0	Set	BYTE (0x00 or 0x01)
2	0x65	T1	Set	BYTE (0x00 or 0x01)
……				
256	0x65	T255	Set	BYTE (0x00 or 0x01)

(7) M Relay (0x67)

Instance	Attribute	Name	Access	Data Type
1	0x64	M0	Set	BYTE
2	0x64	M1	Set	BYTE
……				
4096	0x64	M4095	Set	BYTE

(8) C counter (0x68)

Instance	Attribute	Name	Access	Data Type
1	0x64	C0	Set	INT
2	0x64	C1	Set	INT
……				
200	0x64	C199	Set	INT

Instance	Attribute	Name	Access	Data Type
201	0x64	C200	Set	DINT
202	0x64	C201	Set	DINT
……				
256	0x64	C255	Set	DINT

Instance	Attribute	Name	Access	Data Type
1	0x65	C0	Set	BYTE (0x00 or 0x01)
2	0x65	C1	Set	BYTE (0x00 or 0x01)
……				
256	0x65	C255	Set	BYTE (0x00 or 0x01)

(9) D Register (0x69)

Instance	Attribute	Name	Access	Data Type
1	0x64	M0	Set	INT
2	0x64	M1	Set	INT
……				
12000	0x64	M11999	Set	INT

(10) TCP/IP Interface Object (0xF5)

Instance: 0x01

Attribute	Name	Access	Data Type	Value
0x01	Status	Get	DWORD	0x00000001UL
0x02	Configuration Capability	Get	DWORD	0x00000014UL (DHCP client, Configuration Settable)
0x03	Configuration Control	Get	DWORD	Static IP: 0U DHCP: 0x02U
0x04	Physical Link Object:	Get	STRUCT of:	
	Path Size		UINT	
	Path		Padded EPATH	
0x05	Interface Configuration:	Set	STRUCT of:	
	IP Address		UDINT	
	Network Mask		UDINT	
	Gateway Address		UDINT	
	Name Server		UDINT	
	Name Server 2		UDINT	
	Domain Name		STRING	
0x06	Host Name	Get	STRING	DVP12SE

(11) Ethernet Link Object (0xF6)

Instance: 0x01

Attribute	Name	Access	Data Type	Value
0x01	Interface Speed	Get	UDINT	10 or 100 Mbps
0x02	Interface Flag	Get	UDINT	Bit 0: Link Status Bit 1: Half/Full Duplex
0x03	MAC Address	Get	USINT[6]	

11.8 Appendix H: PLC Specifications

Item	Specifications
Operating temperature	0 to 55°C
Storage temperature	-25 to 70°C
Operating humidity	5–95%; No condensation
Storage humidity	5–95%; No condensation
Work environment	No corrosive gas exists.
Installation location	In a control box
Pollution degree	2
Ingress protection (IP ratings)	IP20
Surge voltage withstand level	1,500 VAC (Primary-PE), 500 VAC (Secondary-PE)
Insulation voltage	Above 5MΩ The voltage between all inputs/outputs and the ground is 500 VDC.
Noise Immunity	ESD: 8KV Air Discharge EFT: Power Line: 2KV, Digital I/O: 1KV, Analog & Communication I/O: 250V Damped-Oscillatory Wave: Power Line: 1KV, Digital I/O: 1KV, RS: 26MHz ~ 1GHz, 10V/m
Ground	The diameter of the ground should not be less than the diameters of the cables connected to the terminals L and N. It is required to use grounding if more than one PLC is being used at the same time.
Vibration / Shock resistance	International Standard IEC61131-2, IEC 68-2-6 (TEST Fc) / IEC61131-2 & IEC 68-2-27 (TEST Ea)
Ambient air temperature-barometric pressure-altitude	Operating: 1080 ~ 795hPa (-1000 ~ 2000 m) Storage:1080 ~ 660hPa (-1000 ~ 3500m)

11.9 Appendix G: Revision History

Item	Revisions	Chapter
#1	◆ Delete information about DVP-SS/SA/SC/EH/SV and add DVP-EC/24SV2 product information.	All chapters
#2	◆ Section 2.10: Update special M table: M1005-M1007, M1015-M1017, M1019, M1023, M1026-M1030, M1035-M1038, M1047-M1057, M1070-M1071, M1076-M1079, M1081-M1084, M1087-M1119, M1136-M1137, M1145-M1159, M1163-M1167, M1169-M1171, M1175-M1179, M1184-M1195, M1200-M1234, M1239-M1240, M1243, M1248, M1253, M1257-M1259, M1261, M1264-M1303, M1305-M1317, M1320, M1326-M1355, M1360-M1431, M1440-M1527, M1530-M1541, M1568-M1569, M1588-M1589, M1592-M1643 ◆ Section 2.10: Update special D table: D1002, D1003, D1007-D1009, D1015, D1021-D1023, D1030-D1035, D1037, D1040-D1049, D1056-D1057, D1061, D1063, D1064, D1100-D1111, D1115-D1118, D1127-D1128, D1131-D1136, D1145-D1175, D1177-D1247, D1313-D1353, D1355-D1991, D1994-D1999, D9800-D9979 ◆ Section 2.13: Add error LED descriptions and error codes, C40C, C421, C422, C423, C424, C431, C433, C437.	Chapter 2
#3	◆ Delete the instruction table in section 5.1 and move the table to the relevant chapters, including chapter 6, 7, 8, 9, and 10. ◆ Delete the instruction index in section 5.5 and move the table to the relevant chapters.	Chapter 5-10
#4	◆ Add a MPU general specification in section 11.8	Chapter 11

MEMO