Content summary

This manual has a comprehensive introduction to the basic functions of WECON PLC Editor and the actual use. This book is completely aimed at zero-based readers, is an essential reference book for entry-level readers to quickly and fully grasp WECON PLC and WECON PLC Editor.

This book starts from the basic product of WECON PLC and the basic concept and operation of WECON PLC Editor. It combines with a large number of cases and graphic analysis to comprehensively and deeply explain the use of WECON PLC Editor Software, as well as PLC program.

WECON technology Co., Ltd. All rights reserved.

Safety Precautions

Before installation, operation, maintenance or inspection of this product, thoroughly read through and understand this manual and all of the associated manuals. Also, take care to handle the module properly and safely.

1) Design precautions

Make sure to have the following safety circuits outside of the PLC to ensure safe system operation even during external power supply problems or PLC failure.

a) An emergency stop circuit, a protection circuit, an interlock circuit for opposite movements, and an interlock circuit (to prevent damage to the equipment at the upper and lower positioning limits).

b) Note that when the PLC CPU detects an error, such as a watchdog timer error, during self-diagnosis, all outputs are turned off. Also, when an error that cannot be detected by the PLC CPU occurs in an input/output control block, output control may be disabled. External circuits and mechanisms should be designed to ensure safe machinery operation in such a case.

c) Note that when an error occurs in a relay or transistor output device, the output could be held either on or off. For output signals that may lead to serious accidents, external circuits and mechanisms should be designed to ensure safe machinery operation in such a case.

2) Installation precautions

a) Use the product within the generic environment specifications described in this manual.

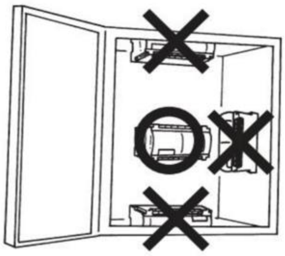
Never use the product in areas with excessive dust, oily smoke, conductive dusts, corrosive gas, flammable gas, vibration or impacts, or expose it to high temperature, condensation, or rain and wind. If the product is used in such conditions, electric shock, fire, malfunctions, deterioration or damage may occur.

b) When drilling screw holes or wiring, make sure that cutting and wiring debris do not enter the ventilation slits. Failure to do so may cause fire, equipment failures or malfunctions.

c) Connect the expansion module, expansion board and cable securely to their designate connectors. Loose connections may cause malfunctions.

- To prevent temperature rise, do not install at the bottom, top and vertical. Be sure to install the wall horizontally as shown on the right.

- Please leave more than 50mm space between PLC CPU and other equipment or structures. Try to avoid high-voltage lines, high-voltage equipment, and power equipment.

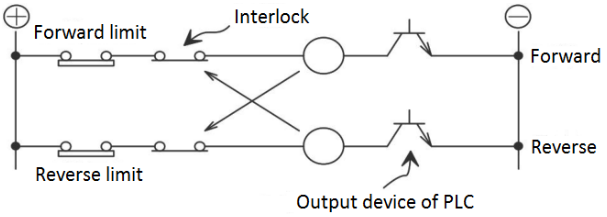


3) Wiring precautions

a) PLC signal input and output cannot be passed on the same cable;

b) Signal input cable and output cable cannot be in the same pipe with other power cable, cannot be bundled together;

c) If above precautions are followed, the input / output wiring will have almost no noise even with a length of 50 to 100 m. However, it is recommended that the wiring length should be within 20m;

Warning	 Forward limit Interlock Forward Reverse Reverse limit Output device of PLC It is very dangerous to switch the forward and reverse contactor at the same time. For such loads, the interlock must be set on the outside of the PLC besides the PLC internal program settings, as shown in the above picture.
Caution	Do not wire vacant terminals externally. Doing so may damage the product.

- Before installation, wiring and other operations, cut off all phases of the power supply externally. Failure to do so may cause electric shock;
- After installation, wiring and other work, the terminal cover must be installed in order to avoid electric shock, before the power operation;
- Connect the AC power supply wiring to the dedicated terminals described in this manual. If an AC power supply is connected to a DC input/output terminal or DC power supply terminal, the PLC will burn out;
- Do not supply power to the [24+] and [24V] terminals (24V DC service power supply) on the main unit or extension units. Doing so may cause damage to the product;
- Perform grounding to the grounding terminal on the main unit and

extension units. Do not use common grounding with heavy electrical systems;

- When there is less than 10ms instantaneous power failure, PLC will continue to work;
- When the power is cut off or the voltage is low for a long time, the PLC will stop working and the output will turn off. However, once the power is restored, the operation will restart automatically;

4) Startup and maintenance precautions

Warning
● Do not touch any terminal while the PLC's power is on. Doing so may cause electric shock or malfunctions; ● Before cleaning or retightening terminals, cut off all phases of the power supply externally. Failure to do so may cause electric shock; ● Before modifying or disrupting the program in operation or running the PLC, carefully read through this manual and the associated manuals and ensure the safety of the operation. An operation error may damage the machinery or cause accidents.

Caution
● Do not disassemble or modify the PLC. Doing so may cause fire, equipment failures, or malfunctions. ✧ *For repair, contact WECON technology Co.,* ● Turn off the power to the PLC before connecting or disconnecting any extension cable. Failure to do so may cause equipment failures or malfunctions.

5) Maintenance and repair

- Periodic inspection: PLC is equipped with shorter life expectancy consumables;
- For relay output, if it has high frequency of abnormal work or it drives large capacity load, please pay attention to its impact on PLC service life.
- Check with other equipment, please note the following points
  a) Is there any abnormal temperature rise due to other heating bodies or direct sunlight?
  b) Is dust or conductive dust invading the machine?
  c) Is there any wiring and terminal loosening and other anomalies?

Catalog

WECON PLC Editor Software Overview

PLC is a digital computer used for automation of typically industrial electromechanical processes; PLCs are used in many machines, in many industries.

It reads external input signals such as: the state of buttons, sensors , switches and pulse waves , and then uses a microprocessor to perform logic, sequence, timing, counting and arithmetic operations, resulting in the corresponding output signal based on the input signal status or internally stored value and pre-written program.

WECON PLC editor uses ladder and instructions list as programming language.

1) Ladder

Ladder logic is widely used to program PLCs, where sequential control of a process or manufacturing operation is required. Ladder logic is useful for simple but critical control systems or for reworking old hardwired relay circuits. As programmable logic controllers became more sophisticated it has also been used in very complex automation systems. It is a graphic language evolution came in relay ladder original relay control system based on the devices used in the design, such as buttons X, intermediate relay M, time relay T, counter C, and so on similar properties contact time of electrical device. The ladder as figure 0-1 shows.

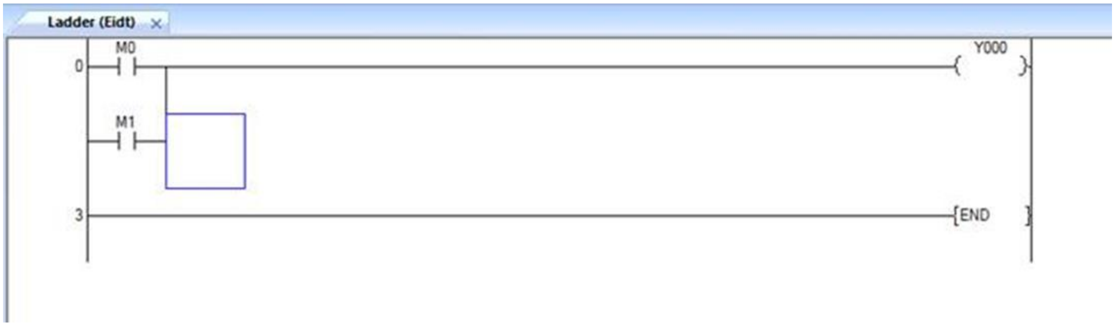
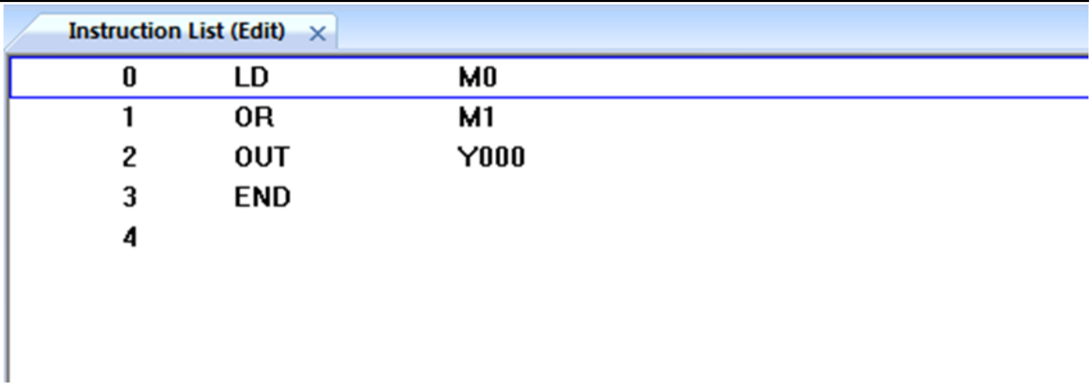


Figure 0-1

2) Instructions list

Instruction List (IL) is designed for programmable logic controllers (PLCs). It is a low level language and resembles assembly. All the instructions and operands are inputted for PLC programming. The IL as figure 0-2 shows.



Instruction List (Edit)

```
0    LD     M0
1    OR     M1
2    OUT    Y000
3    END
4
```

Figure 0-2

3) Program switch

According to their own programming practice, users can switch ladder and instruction list in order to improve programming efficiency. There is switch function as figure 0-3 shows.

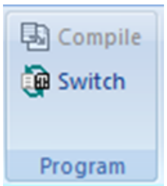


Figure 0-3

1. WECON PLC introduction

1.1 PLC host composition

According to the different hardware structure, PLC can be divided into host unit, expansion module and BD board.

WECON PLC has the built-in battery, CPU and I/O points. The PLC CPU can connect with I/O expansion modules to expend the I/O point number or connect with some special function expansion modules.

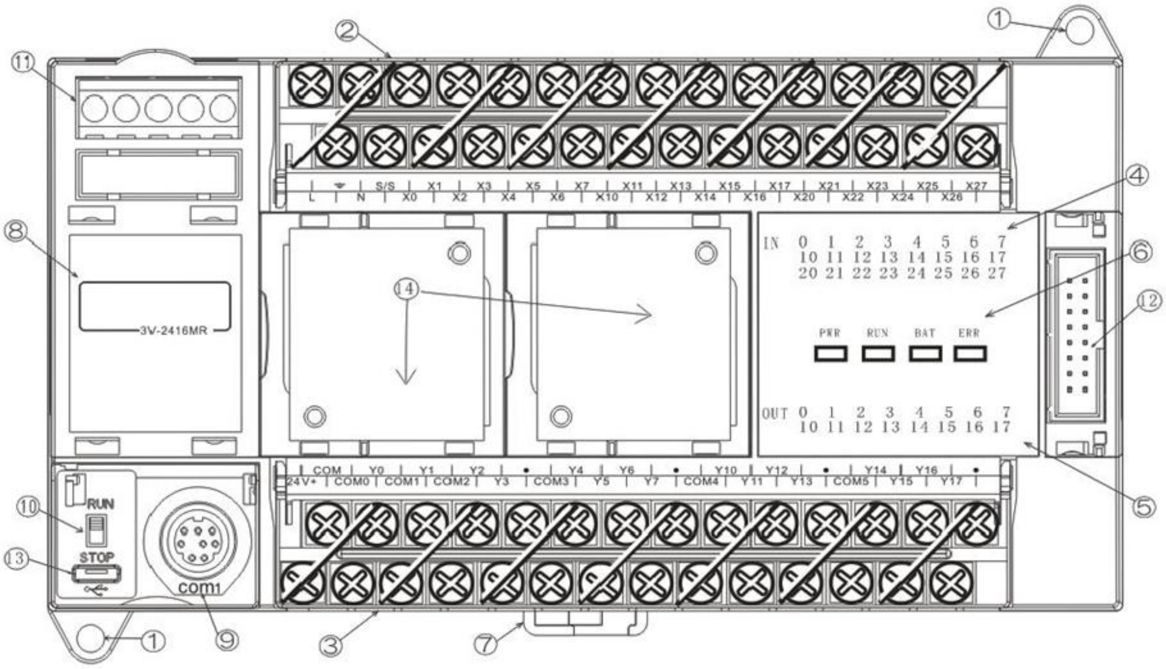
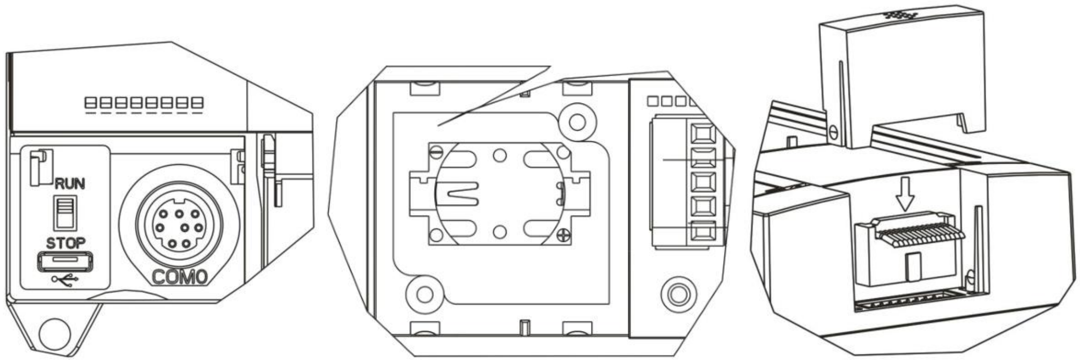


Figure 1-1

① Mounting hole(⌀:4.5)
② Power supply and input signal terminal
③ 24v output power supply and output terminal
④ Input indicator
⑤ Output indicator
⑥ PLC state indicator:

a) PWR: power light
b) RUN: running light
c) BAT: low power light
d) ERR: blinks when the program is wrong;
e) :always on when a CPU error occurs

⑦ DIN rail mounting clip
⑧ Cover plate
⑨ Com1 port
⑩ RUN/STOP switch
⑪ COM1/COM2 port and rs485 port
⑫ Expansion module interface
⑬ Micro-USB port
⑭ BD board slot



Device interface (enlarged)   Battery slot (enlarged)   Expansion interface (enlarged)

Figure 1-2

1.2 Expansion module

WECON PLC has many kinds of expansion modules, including I/O module, analog input and output module, high speed output module and Ethernet communication protocol.

① Expansion cable
② Com light: always on when digital transmission is normal

③ 24v: always on when connect with external 24v power supply

④ Module power status light: Always on when normal

⑤ The name of the expansion module

⑥ Analog signal output terminal

⑦ Expansion interface

⑧ DIN rail mounting clip

⑨ DIN rail hooks

⑩ Mounting hole(⌀:4.5)

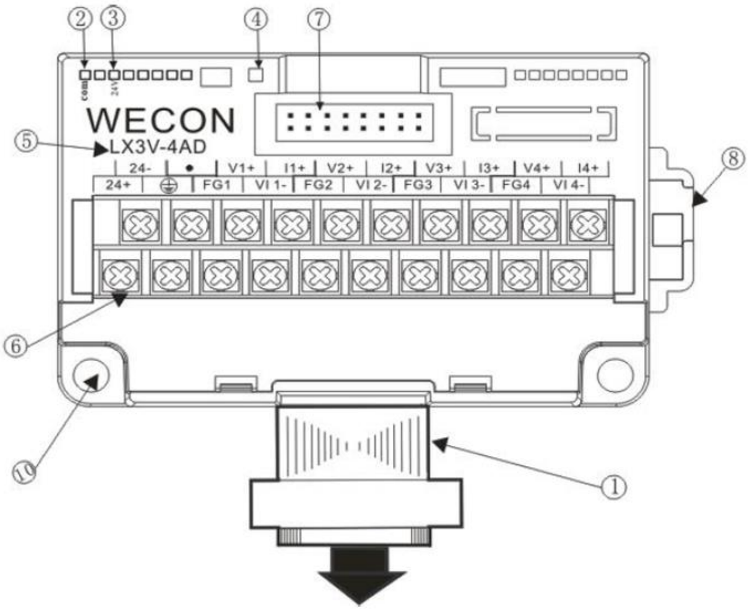


Figure 1-3

The PLC CPU and the expansion module are the same size in height and depth, but different width. So they can connect with each other in a neat form, also the configuration is flexible.

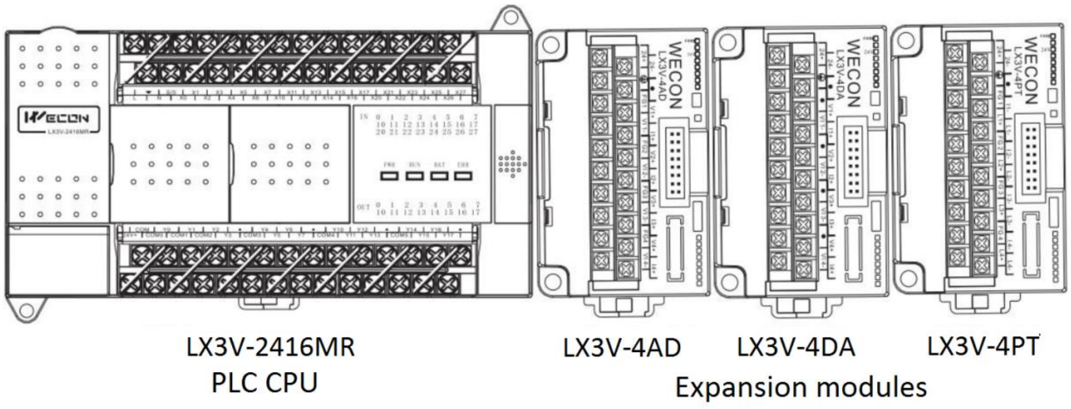


Figure 1-4

1.3 BD board

Compared with expansion module, BD board is smaller, more flexible (up to 2 BD boards in one PLC) and cheaper, but also with multi-function. Because the BD board is installed in the host unit, so it will not occupy extra space.

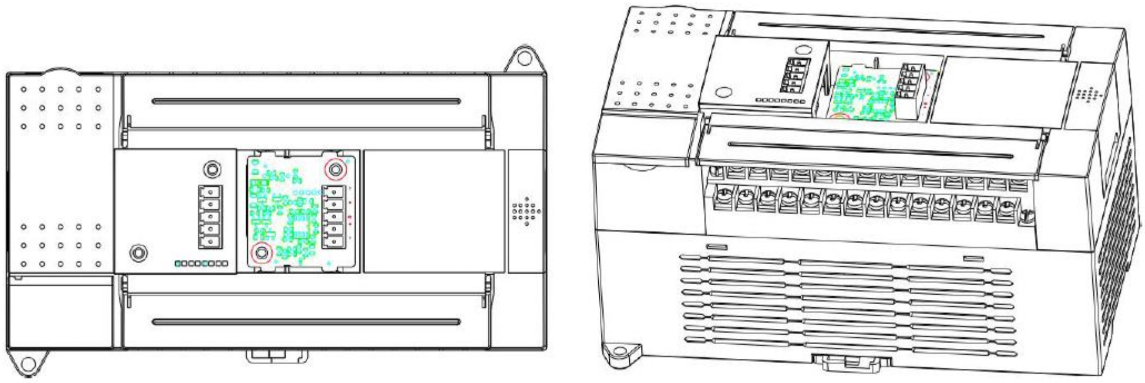


Figure 1-5

1.4 Dimension

1) PLC CPU

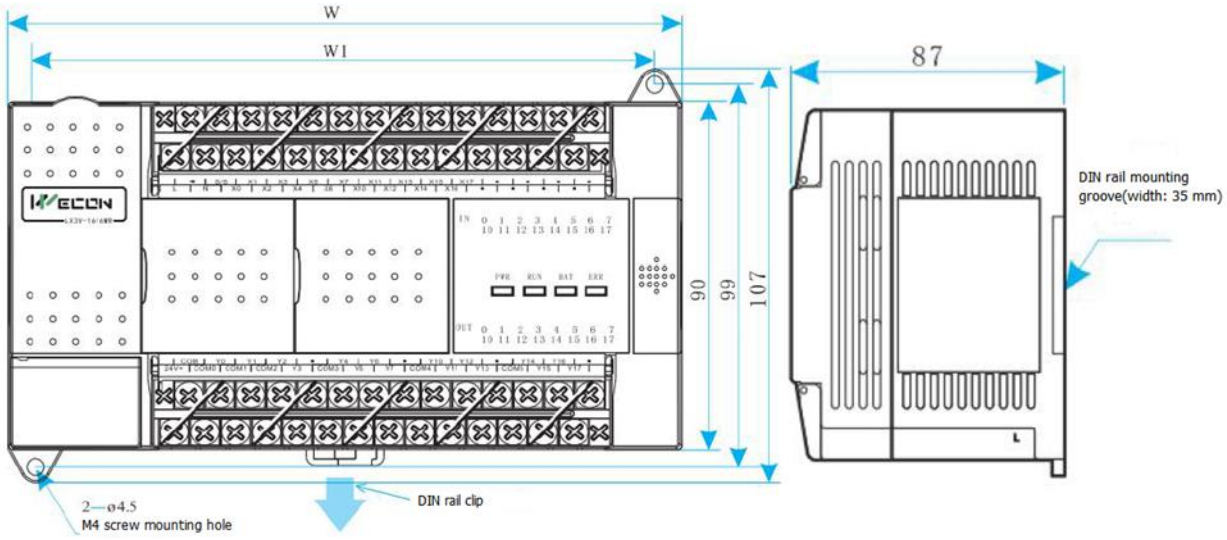


Figure 1-6

Mount the plc in the DIN (35mm) rail directly. Gently pull out the DIN rail mounting clip from below, then we can remove the main unit from DIN rail. The mounting holes can be used to directly mount the programmable controller with M4 screws. Please refer to the following table for the pitch and location of mounting holes.

Table 1-1

MODEL	W(mm)	W1(mm)	MODEL	W(mm)	W1(mm)
LX3V-0806MX	75	61	LX3V-1208MX	75	61
LX3V-1212MX	136	123	LX3V-1412MX	136	123
LX3V-1616MX	175	161	LX3V-2416MX	175	161
LX3V-2424MX	221	207	LX3V-3624MX	221	207

2) Expansion module

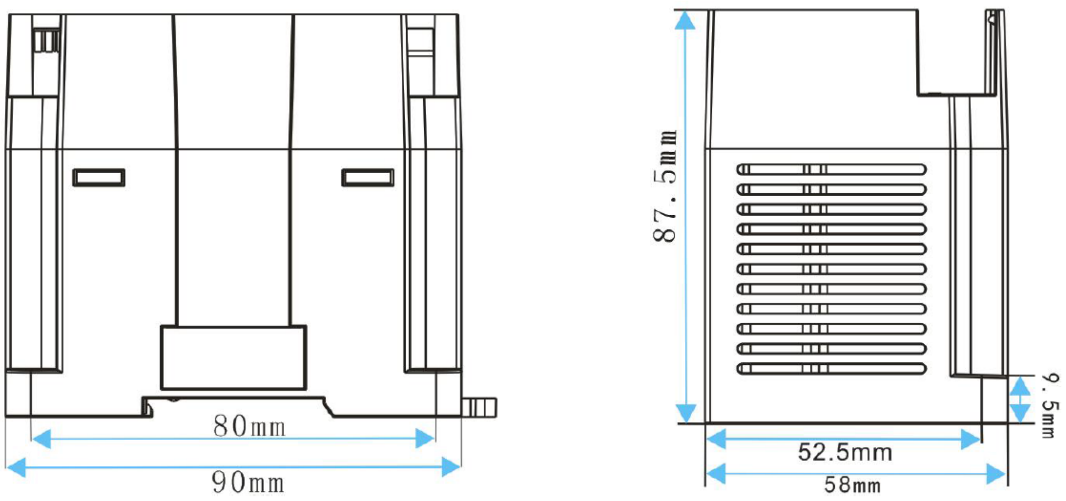


Figure 1-7

3) BD board

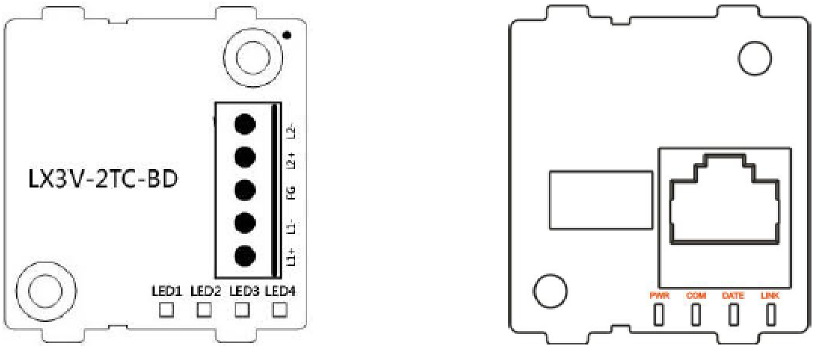


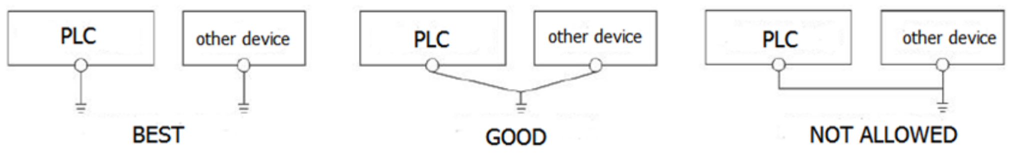
Figure 1-8

2. Product specifications

2.1 General specifications

<table>
<tr><td><b>Temperature</b></td><td colspan="5">Working temperature: 0~55 °C storage temperature: -20~70 °C</td></tr>
<tr><td><b>Humidity</b></td><td colspan="5">Working humidity 5 ~ 85% RH (no condensation)</td></tr>
<tr><td rowspan="6"><b>Vibration Resistance</b></td><td colspan="5">Meet JISC0040 standard</td></tr>
<tr><td></td><td>Frequency</td><td>Acceleration</td><td>Amplitude</td><td rowspan="5">10 times for each direction(X, Y, Z), each direction for 80 minutes</td></tr>
<tr><td rowspan="2">Mounted in DIN rail</td><td>10~57 Hz</td><td>---------------</td><td>0.035mm</td></tr>
<tr><td>57~150 Hz</td><td>4.9 m/s²</td><td>---------------</td></tr>
<tr><td rowspan="2">Mounted directly</td><td>10~57 Hz</td><td>----------------</td><td>0.075mm</td></tr>
<tr><td>57~150 Hz</td><td>9.8 m/s²</td><td>---------------</td></tr>
<tr><td><b>Shock Resistance</b></td><td colspan="5">JISC0040 standard (147 m/s², duration:11ms, sine half-wave pulse in three directions (X, Y, Z), three times for each direction</td></tr>
<tr><td><b>Noise Resistance</b></td><td colspan="5">Noise voltage 1000Vp-p, Noise amplitude 1ns/us, frequency 30 ~ 100Hz</td></tr>
<tr><td><b>Voltage Resistance</b></td><td colspan="3">AC1500V(1 min)</td><td colspan="2" rowspan="2">The power supply terminal and the grounding terminal conform to the JEM-1021 standard</td></tr>
<tr><td><b>Insulation impedance</b></td><td colspan="3">above 5MΩ(measured by DC500V insulation tester)</td></tr>
<tr><td><b>Earthing</b></td><td colspan="5">Third kinds of earthing (not to be combined with high powered system) ※1</td></tr>
<tr><td><b>Environment</b></td><td colspan="5">No corrosive, flammable gas, no conductive dust</td></tr>
</table>

※1



2.2 Electricity specification

1) AC power type

Model	LX3V-0806/1208/1212/1410/1412MX-A	LX3V-1616/2416/2424/3624MX-A

Rated voltage	AC100~240V	
Voltage range	AC85~264	
Rated frequency	50/60 Hz	
Power outage time	Continue to work within 10ms power outage time	
Power fuse	250V 1A	250V 3.15A
Impulse current	Less than 20A 5ms/AC 100V	
Power	20W	50W
Sensor power supply	DC 24V 70mA	

※1 The input current part (7mA / 1 point, 5mA / 1 point) is also included.

2) DC power type

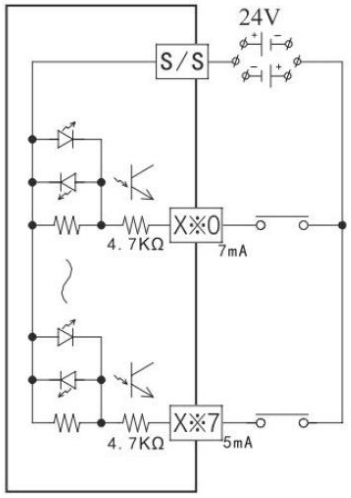
Model	**LX3V/LX3VP/LX3VE**
Rated voltage	DC24V
Voltage range	DC24V±10%
Power outage time	Continue to work within 10ms power outage time
Power fuse	250V 3.15A
Impulse current	Less than 15A 1ms/AC 100V
Power	Less than 30W(not include the power of the expansion module)

2.3 Input specification

The specifications of the basic units of the LX series programmable controllers are shown in the table2-1 below:

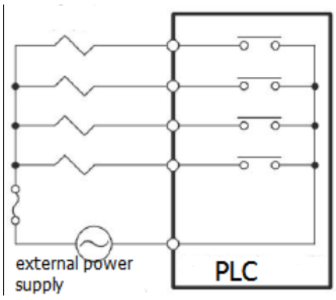
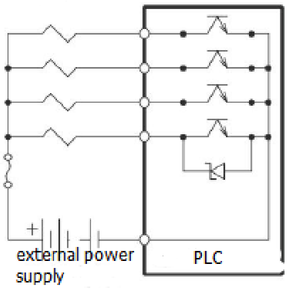
Table 2-1

Item	AC power supply, DC output
Model	LX series basic unit
Input signal voltage	DC 24V±10%
Input signal current	7mA/DC 24V (behind X002, 3.5 mA/DC24V)
Input off current	Less than 1.5 mA

Input responding time	About 10ms
	Can be changed to 0 ~ 15ms by built-in digital filter D8020
Input signal type	Contact input or NPN, PNP open electrode transistor input
Insulated return	Optical coupler insulation
Input statues	When input is ON, LED is ON
Input loop structure	 The diagram is NPN connection method, if S/S is connected to negative pole, X is connected to positive pole, that is PNP connection method.

※1: After X002 is 4.7KΩ.

2.4 Output specification

Output type	Relay output	Transistor output
Model	All LX series	
Output loop		
External power supply	Less than AC 250/DC 30V	DC 5~30V
Insulation	Mechanical insulation	Optical coupler insulation
Action	Relay coil drived, LED on	Optical coupler drived, LED on
Max **Resistive**	2A/point, 8A/4 points	0.5A/point,

load			0.8A/4points,0.3A/point(Y0,Y1)
	Inductive	80VA	12W/DC24V,7.2W/DC24V(Y0,Y1)
	General	100W	0.9W/DC24V,0.9W/DC24V(Y0,Y1)
Leak current		-----------	0.1 Ma/DC30V
Minimum load		-----------	DC5V 2mA(reference)
Response time		About 10ms	Less than 0.2 ms, 5us(Y0,Y1)
Output signal type		-----------	NPN type

[Output loop]

Please connect the DC inductive load and the freewheeling diode in parallel. Otherwise, the contact life will be significantly reduced. Freewheeling diode reverse withstand voltage is 5 to 10 times the load voltage, the forward current value is higher than the load current.

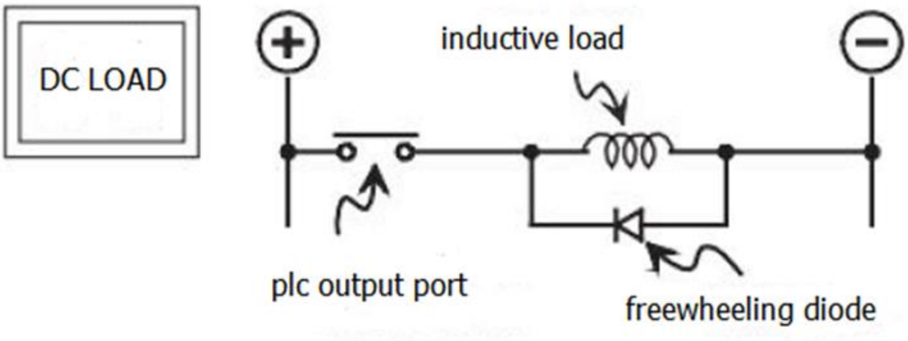


Figure 2-1

If the load is AC inductive, connect the load and surge absorber in parallel can reduce the noise.

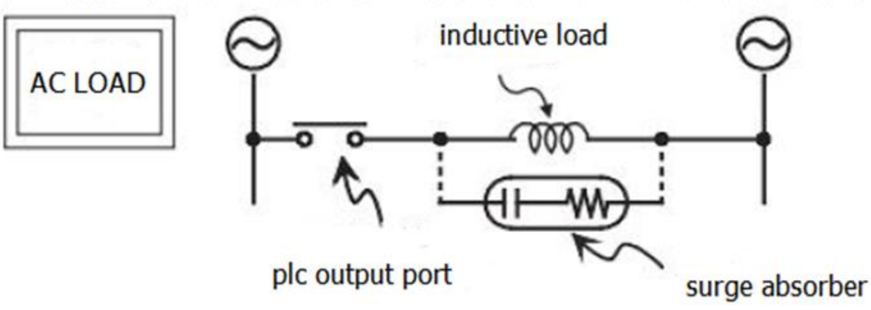


Figure 2-2

It is best to use the plc output contacts on the same phase as Figure 2-3 shown below.

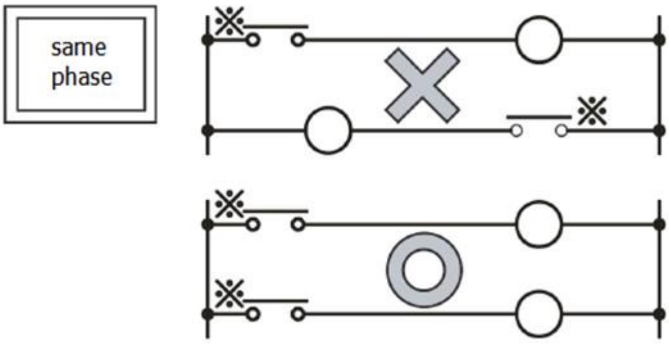


Figure 2-3

It is very dangerous to connect the FWD and REV contact at the same time. In addition to adding the interlocking control of the program in the plc, there also should be interlock outside the PLC.

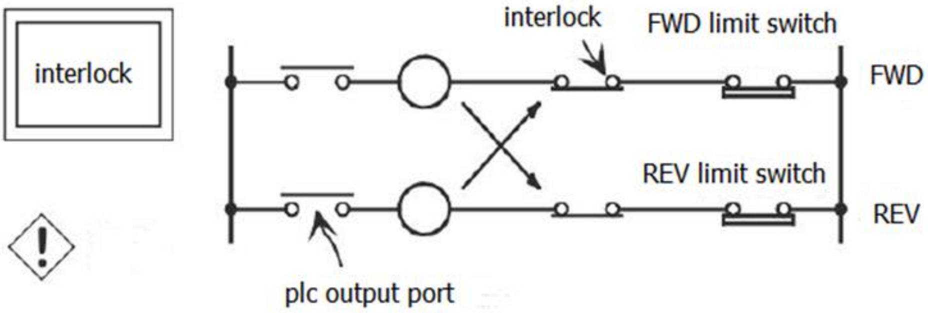
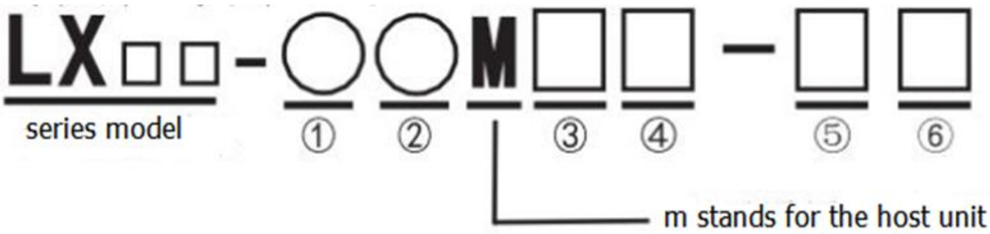


Figure 2-4

2.5 Product naming rule

The model of the plc can be found on the label on the side of the product.

[The basic unit's model name constitutes]



The meaning of ①~⑥ is as below.

① Input points

② Output points

③ Output type, R means relay, T means transistor. Relay can support DC and AC. Transistor can only support DC.

④ High speed output port, 4H means the plc can support 4 channels of high speed pulse output, the same meaning for 2H. If there are no 2H and 4H, the default is 2H.

⑤ Power supply type: A means AC220V, D means DC24V.

⑥ Instruction set: 1 means LX1S, 2 means LX2N, the default instruction set is LX2N.

3. Device description

The following table lists all the devices that WECON LX3V series PLC supports.

Table 3-1

No.	Device	Descriptions
1	X - Input	Representation of physical inputs to PLC;
2	Y - Output	Representation of physical outputs from PLC;

3	M - Intermediate	Common intermediate register;
		System special register;

4	S - State	PLC internal states flag for step control;
5	T - Timer	16-bit timer (1, 10 and 100ms)

6	C - Counter	16-bit and 32-bit up/down counter;
		High speed counter;

7	D – Data register	Data register ;
		String register;
		Indirect addressing address;

8	P, I - Pointer	Jump pointer;
		Sub-program pointer;
		Interrupt pointer (high speed,);

9	K, H - Constant	Binary, decimal, hexadecimal, floating point, etc.

Table 3-2

Device	LX3V(1S firmware)	LX3V (2N firmware)	LX3VP	LX3VE	Expansion module
X - input	X0~X13 (Max. 12)	X0~X43 (Max. 36)	X0~X43 (Max. 36)	X0~X43 (Max. 36)	X0~X77 (Max.128)
Y - output	Y0~Y7 (Max. 8)	Y0~Y27 (Max. 24)	Y0~Y27 (Max. 24)	Y0~Y27 (Max. 24)	Y0~Y77 (Max.128)

3.1 Input relay X

The input relay X represents the physical inputs to PLC. It can detect the external signal states. 0 is for open circuit, 1 is for closed circuit.

The states of input relays can’t be modified by program instruction, the node signal (normally open, normally closed) can be unlimited use in the program.

If connected IO expansion module, the port starts from the main module, according to the order of the numbers. But DI is named in groups of eight. For example main module is X0~X7, X10~X14. The X0 in DI expansion module corresponds to X20, not X15.

Devices numbered in: Octal, i.e. X0 to X7, X10 to X17

[Available devices]

Table 3-3

Model	Input	Output	Model	Input	Output
LX3V-0806MR/MT-A1(D1)	X0~X7	Y0~Y5	LX3VP-1208MR/MT-A(D)	X0~X7	Y0~Y5
LX3V-1208MR/MT-A1(D1)	X0~X13	Y0~Y7	LX3VP-1212MR/MT-A(D)	X0~X13	Y0~Y13
LX3V-0806MR/MT-A2(D2)	X0~X7	Y0~Y5	LX3VP-1412MR/MT-A(D)	X0~X15	Y0~Y13
LX3V-1208MR/MT-A2(D2)	X0~X13	Y0~Y7	LX3VP-1616MR/MT-A(D)	X0~X17	Y0~Y17
LX3V-1212MR/MT-A(D)	X0~X13	Y0~Y13	LX3VP-2416MR/MT-A(D)	X0~X27	Y0~Y17
LX3V-1410MR/MT-A(D)	X0~X15	Y0~Y11	LX3VP-2424MR/MT-A(D)	X0~X27	Y0~Y27
LX3V-1412MR/MT-A(D)	X0~X15	Y0~Y13	LX3VP-3624MR/MT-A(D)	X0~X43	Y0~Y27
LX3V-1616MR/MT-A(D)	X0~X17	Y0~Y17	LX3VE-1412MR/MT-A(D)	X0~X15	Y0~Y13
LX3V-2416MR/MT-A(D)	X0~X27	Y0~Y17	LX3VE-1616MR/MT-A(D)	X0~X17	Y0~Y17
LX3V-2424MR/MT-A(D)	X0~X27	Y0~Y27	LX3VE-2416MR/MT-A(D)	X0~X27	Y0~Y17
LX3V-3624MR/MT-A(D)	X0~X43	Y0~Y27	LX3VE-2424MR/MT-A(D)	X0~X27	Y0~Y27
			LX3VE-3624MR/MT-A(D)	X0~X43	Y0~Y27

3.2 Output replay Y

The output relay Y represents physical outputs from PLC. 0 is for open circuit, 1 is for closed circuit.

Depending on the output element can be divided into relay type, transistor type etc.

If connected IO expansion module, the port starts from the main module, according to the order of the numbers. But DO is named in groups of eight. For example main module is Y0~Y7, Y10~Y14. The Y0 in DO expansion module corresponds to Y20, not Y15.

Devices numbered in: Octal, i.e. Y0 to Y7, Y10 to Y17.

3.3 Auxiliary relays M

Auxiliary Relay M device is used as an intermediate variable during the execution of a program, as auxiliary relays in the practical power control system which is used to transfer the state messages. It can use the word variable formed by M variables. M variables is not directly linked with any external ports, but it can contact with the outside world by the manners of copying X to M or M to Y through the program coding. A variable M can be used repeatedly.

Devices numbered in: Decimal, i.e. M0 to M9, M10 to M19. The variables that are more than M8000 are the system-specific variables, which are used to interact with the PLC user program with the system states; part of the M variables have the feature of power-saving.

1) General Stable State Auxiliary Relays

The general stable state Auxiliary relays in LX3V series PLC are M0 ~ M499, there are total of 500 points. The type of auxiliary relay is related to its part number and PLC serial.

Table 3-4

PLC	General	Latched	Latched-specific	System-specific
LX3V (1S firmware)	384 ※3 (M0 – M383)	-	128 ※3 (M383 – M511)	256 (M8000-M8255)
LX3V (2N firmware)	500 ※1 (M0 – M499)	524 ※ 2 (M500 – M1023)	2048 ※3 (M1024 – M3071)	256 (M8000-M8255)
LX3VP	500 ※1 (M0 – M499)	524 ※ 2 (M500 – M1023)	2048 ※3 (M1024 – M3071)	256 (M8000-M8255)
LX3VE	500 ※1 (M0 – M499)	524 ※ 2 (M500 – M1023)	2048 ※3 (M1024 – M3071)	256 (M8000-M8255)

Users can set non-latched and latched area for Auxiliary relays in PLC by parameter setting

※1, Non-latched area, it can be changed to latched area by parameter setting.
※2, Latched area, it can be changed to non-latched area by parameter setting.
※3, The non-latched or latched feature can’t be changed.

2) Latched auxiliary relays

There are a number of latched relays whose state is retained. If a power failure should occur all output and general purpose relays are switched off. When operation is resumed the previous state of these relays is restored.

As below pictures show, in (a), relay M500 is activated when X0 is turned ON. If X0 is turned OFF after the activation of M500, the ON state of M500 is self-retained. (b) shows Circuit Waveform diagram of (a). For using this function, (c) could makes M500 “Turn ON” all the time.

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

Figure 3-1

3) System-specific auxiliary relays

A PLC has a number of special auxiliary relays. These relays all have specific functions such as provide clock pulse and sign, set PLC operation mode, or use for step control, prohibit interrupt, set counter is add count or subtract count, etc. And they are classified into the following two types.

- Using contacts of special auxiliary relays, coils are driven automatically by the PLC. Only the contacts of these coils may be used by a user defined program.
  Examples: M8000: RUN monitor (ON during run);
  M8002: Initial pulse (Turned ON momentarily when PLC starts);
  M8012: 100 msec clock pulse;

- Driving coils of special auxiliary relays, a PLC executes a predetermined specific operation when these coils are driven by the user.
  Examples: M8033: All output statuses are retained when PLC operation is stopped;
  M8034: All outputs are disabled;
  M8039: The PLC operates under constant scan mode;

3.4 State relays S

State relays S is used to design and handle step procedures, controls transfer of step by STL step instructions to simplify programming design. S also can be used as M, if there is no STL instruction. Part of the S has the feature of power-saving.

Devices numbered in: Decimal, i.e. S0 to S9, S10 to S19.

Table 3-5

PLC	General			Latched			Alarm
	-	Initialized	-	-	Initialized	-	
LX3V (1S firmware)	-	-	-	128 ※3 (S0 – S127)	10 (S0 – S9)	10 (S10 – S19)	
LX3V (2N firmware)	500 ※1 (S0 – S499)	10 (S0 – S9)	10 (S10 – S19)	400 ※2 (S500 – S899)	-	-	100 ※2 (S900 – S999)
LX3VP	500 ※1 (S0 – S499)	10 (S0 – S9)	10 (S10 – S19)	400 ※2 (S500 – S899)	-	-	100 ※2 (S900 – S999)
LX3VE	500 ※1 (S0 – S499)	10 (S0 – S9)	10 (S10 – S19)	400 ※2 (S500 – S899)	-	-	100 ※2 (S900 – S999)

※1, Non-latched area, it can be changed to latched area by parameter setting.
※2, Latched area, it can be changed to non-latched area by parameter setting.
※3, The non-latched or latched feature can’t be changed.

1) General State Relays

As above picture shows, when X0=ON, then S0 set ON, and Y0 is activated. When X1=ON, then S11 set ON, and Y1 is activated. When X2=ON, S12 set ON, then Y2 is activated, as Figure 3-2 shows.

2) Latched State Relays

There are a number of latched relays whose state is retained. If a power failure should occur all output and general purpose relays are switched off. When operation is resumed the previous state of these relays is restored.

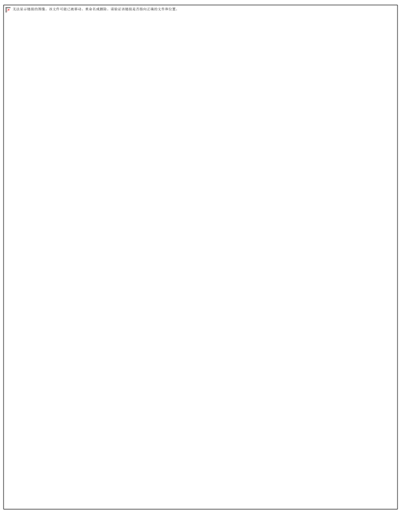
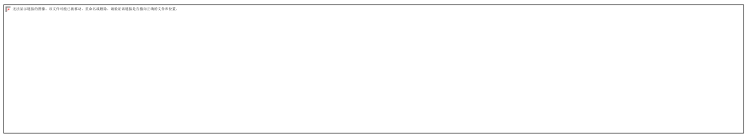

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

Figure 3-2

3) Annunciator Flags

Some state flags can be used as outputs for external diagnosis (called annunciation) when certain applied instructions are used.


无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

If X1 and X2 set ON at the same time and keep more than 1 seconds, S900 is activated, if X1 or X2 is turned OFF after the activation of S900, the ON state of S900 is self-retained. If X1 and X2 set ON at the same time less than 1 seconds, S900 is not activated.

3.5 Timer

The timer is used to perform the timing function. Each timer contains coils, contacts, and counting time value register. A driven coil sets internal PLC contacts. Various timer resolutions are possible, from 1 to 100ms. If the coil power shuts off (insufficient power), the contacts will restore to their initial states and the value will automatically be cleared. Some timers have the feature of accumulation and power-saving.

Devices numbered in: Decimal, i.e. T0 to T9, T10 to T19.

Table 3-6

PLC	100ms 0.1– 3276.7s	100ms 0.1 – 3276.7s 0.01–327.67s	10ms 0.01-327.67s	Retentive 1ms 0.001-32.767s	Retentive 100ms 0.1–3276.7s
LX3V (1S Firmware)	32 (T0 – T31)	31 (T32 – T62)	31 (T32 – T62)	1 (T63)	
LX3V (2N Firmware)	200 (T0 – T199) Sub-program 8 (T192–T199)	-	46 (T200 – T245)	Interrupted 4 (T246 – T249)	6 (T250 – T255)
LX3VP	200 (T0 – T199) Sub-program 8 (T192–T199)	-	46 (T200 – T245)	Interrupted 4 (T246 – T249)	6 (T250 – T255)
LX3VE	200 (T0 – T199) Sub-program 8 (T192–T199)	-	46 (T200 – T245)	Interrupted 4 (T246 – T249)	6 (T250 – T255)

1) General timer (T0~T245)

The timer output contact is activated when the count data reaches the value set by the constant K.

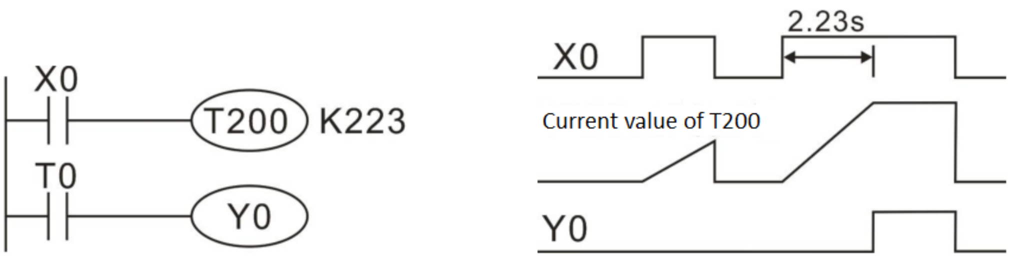


Figure 3-3

As above picture shows, when X0 is on, T200 counts from zero and accumulates 10ms clock pulses. When the current value is equal to the set value 223, timer output contact is activated; the output contact of the T200 is actuated after its coil is driven by 2.23s.

2) Retentive Timers (T246~T255)

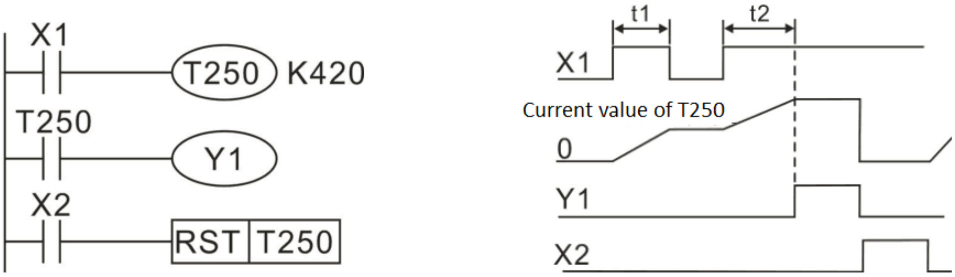
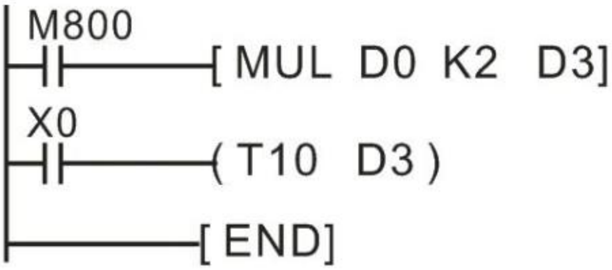


Figure 3-4

As above picture shows, T250 has the ability to retain the currently reached present value even after X1 has been removed. If T1+T2=42s, T250 (open contact) set on. When X2 set ON, timer T250 will be reset.

3) Set value

The set value of the timer can be determined by constant (K, H) in the program memory and can also be specified indirectly with the contents of the data register (D).



As above program shows, D3 is set value for T10, D3=D0*2.

3.6 Counter

Counter performs counting function, it contains coil, contact and count value register. The current value of the counter increases each time coil C0 is turned ON. The output contact is activated when count value reach to preset value.

Counters which are latched are able to retain their status information, even after the PLC has been powered down. This means on re-powering up, the latched counters can immediately resume from where they were at the time of the original PLC power down.

Devices numbered in: Decimal, i.e. C0 to C9, C10 to C19

Table 3-7

PLC	16bit UP Counters 0 – 32,767		32bit Bi-directional Counters -2,147,483,648 - +2,147483647	
	General	Latched	General	Latched
LX1S	16 (C0 – C15) ※3	16 (C16 – C31) ※3	-	-
LX2N	100 (C0-C99) ※1	100(C100 – C199) ※2	20 (C200 – C219) ※1	15 (C220 – C234) ※2
LX3V	100 (C0-C99) ※1	100(C100 – C199) ※2	20 (C200 – C219) ※1	15 (C220 – C234) ※2

※1, Non-latched area, it can be changed to latched area by parameter setting.
※2, Latched area, it can be changed to non-latched area by parameter setting.
※3, The non-latched or latched feature can't be changed.

1) 16bit up counter

16bit counters: 1 to +32,767, as below picture shows, the current value of the counter increases each time coil C0 is turned ON by X2. The output contact is activated when the coil is turned ON for the tenth time.

After this, the counter data remains unchanged when X2 is turned ON. The counter current value is reset to '0' (zero) when the RST instruction is executed by turning ON X1 in the example. The output contact Y0 is also reset at the same time.

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

Figure 3-5

2) 32bit bi-directional counter

32bit bi-directional counters: -2,147,483,648 to +2,147,483,647. C200- 219 are general, C220- 234 are latched.

The counting direction is designated with special auxiliary relays M8200 to M8234. When the special auxiliary relay is ON, it is decremented; otherwise, it is counting up.

3.7 High speed counter

Although counters C235 to C255 (21 points) are all high speed counters, they share the same range of high speed inputs. Therefore, if an input is already being used by a high speed counter, it cannot be used for any other high speed counters or for any other purpose, i.e as an interrupt input.

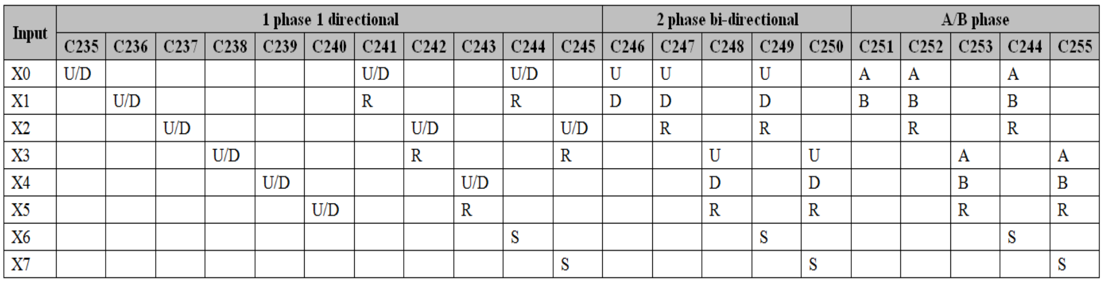
The selection of high speed counters is not free, they are directly dependent on the type of counter required and which inputs are available.

1) Available counter types

a) 1 phase with user start/reset: C235 to C240
b) 1 phase with assigned start/reset: C241 to C245
c) 2 phase bi-directional: C246 to C250
d) A/B phase type: C251 to C255

Different types of counters can be used at the same time but their inputs must not coin-cider. Inputs X0 to X7 cannot be used for more than one counter.

Table 3-8



Input	1 phase 1 directional											2 phase bi-directional					A/B phase				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C244	C255
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			U/D		R		R			R		R	
X3				U/D				R			R			U		U			A		A
X4					U/D				U/D					D		D			B		B
X5						U/D			R					R		R			R		R
X6										S					S					S	
X7											S					S					S

U: up counter input

D: down counter input

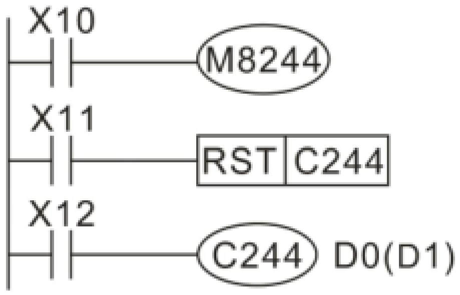
R: reset counter (input)

S: start counter (input)

A: A phase counter input

B: B phase counter input

2) 1 phase



As above program shows, C244 is 1 phase high speed counter with start, stop and reset functions. From the table, X1~X6 are for start and reset. C244 start counting when X12 and X6 are turned ON, the counter input terminal is X0, set value for C244 is determined by D0 (D1), so C244 can be reset by X0 or X11.

3) 2 phase

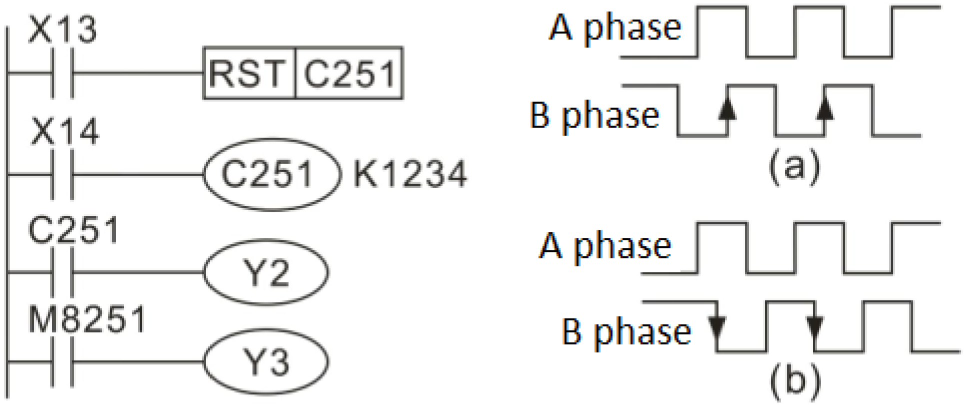


Figure 3-6

C251~C255 are 2 phase (AB phase) high speed counter. As above (b) picture shows, C251 counts according from X0 (A phase) and X1 (B phase), when X14 is turned ON. C251 is reset when X13 is turned ON.

While A phase is turned ON, if B changes state from OFF to ON, C251 executes up count operation. While A phase is turn ON, if B changes state from ON to OFF, C251 executes down count operation. According to this principle, C251 executes up count operation while machine forward, and C251 executes down count operation while machine reverse. The M8251 monitors the C251's up / down counting status, OFF is for up counting, ON is for down counting.

4) Output Y: high speed pulse output transistor

- It supports up to 4 channels, and each channel maximum output frequency is 200K;
- The output frequency can be used for controlling inverter, stepper and servo motors and so on;

5) Input X: one phase

- If first two(X0, X1) terminals were for hardware counter, they can support maximum 200 KHz input signal at the same time; If first two terminals were for software counter could support maximum 100 KHz input signal at same time; they default for hardware counter, but users can change them for software counter by HSCS, HSCR, HSZ instructions;
- X2, X3, X4 and X5 are for software counter, they can support maximum 10 KHz input signal at the same time;

6) Input X: A/B phase

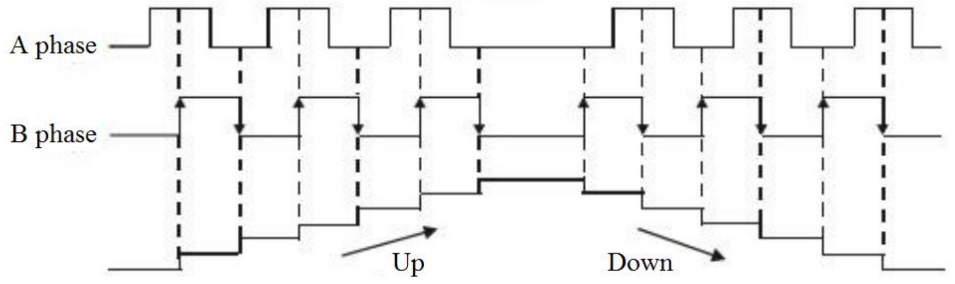
- If first two(X0, X1) terminals were for hardware counter, they can support maximum 100 KHz input signal at the same time;If first two terminals were for software counter could support maximum 50 KHz input signal at same time; they default for hardware counter, but users can change them for software counter by HSCS, HSCR, HSZ instructions;
- X2, X3, X4 and X5 are for software counter, they can support maximum 5 KHz input signal at the same time;

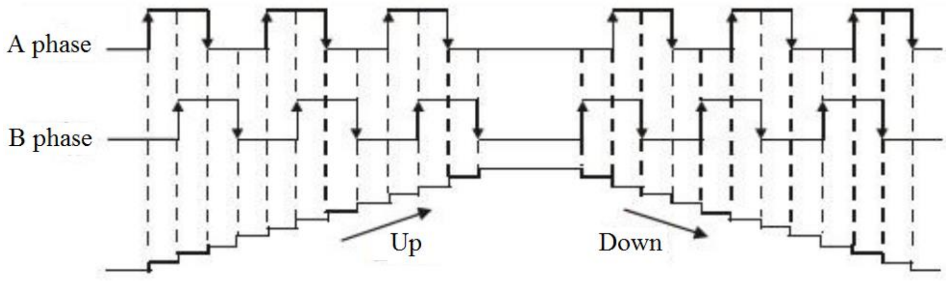
There are two frequency modes for 2 phase 2 input, one is 2 times, and the other is 4 times, as following table shows, users select mode in D8200.

Note:

HSCS, HSCR and HSCZ can’t be used with Frequency multiplication

Table 3-9

Value in D8200	Count icon
K2 (two times)	

K4 or others (four times) (default)	

3.8 Data register D

Data registers, as the name suggests, store data. The stored data can be interpreted as a numerical value or as a series of bits, being either ON or OFF. A single data register contains 16bits or one word. However, two consecutive data registers can be used to form a 32bit device more commonly known as a double word. If the contents of the data register are being considered numerically then the Most Significant Bit (MSB) is used to indicate if the data has a positive or negative bias. As bit devices can only be ON or OFF, 1 or 0 the MSB convention used is, 0 is equal to a positive number and 1 is equal to a negative number.

In WECON LX Series PLC, most data in the instructions are signed numbers. The bit 15 in 16-bit address is sign bit (0 means positive, 1 means negative). The high bit 15 in 32-bit address is sign bit, the data range is -32,768 - +32,767.

Devices numbered in: Decimal, i.e. D0 to D9, D10 to D19

Table 3-10

PLC	General	Latched	Latched- specific		System-specific	Special
			-	Files		
LX3V (1S firmware)	128 ※3 (D0-D127)	-	128 ※3 (D128-D255)	D1000-D2499 can be used for files by parameter setting	256 (D8000-D8255)	16 (V0-V7) (Z0-Z7)
LX3V (2N firmware)	200※1 (D0-D199)	312※2 (D200-D511)	7488 ※3 (D512-D7999)	D1000-D7999 can be used for files by parameter setting	256 (D8000-D8255)	16 ※3 (V0-V7) (Z0-Z7)
LX3VP	200※1 (D0-D199	312※2 (D200-D51	7488 ※3 (D512-D79	D1000-D7999 can be used for files	256 (D8000-D825	16 ※3 (V0-V7)

	)	1)	99)	by parameter setting	5)	(Z0-Z7)
LX3VE	200※1 (D0-D199)	312※2 (D200-D511)	7488 ※3 (D512-D7999)	D1000-D7999 can be used for files by parameter setting	256 (D8000-D8255)	16 ※3 (V0-V7) (Z0-Z7)
LX3VM	200※1 (D0-D199)	312※2 (D200-D511)	7488 ※3 (D512-D7999)	D1000-D7999 can be used for files by parameter setting	256 (D8000-D8255)	16 ※3 (V0-V7) (Z0-Z7)

※1, Non-latched area, it can be changed to latched area by parameter setting.

※2, Latched area, it can be changed to non-latched area by parameter setting.

※3, The non-latched or latched feature cannot be changed.

1) General

A single data register contains 16bits or one word. However, two consecutive data registers can be used to form a 32bit device more commonly known as a double word. Data remains the same until the next time it is rewritten. When switch the PLC state (RUN to STOP or STOP to RUN), the data will be erased. If the special auxiliary relay M8033 is ON, the data in general data register will be retained while switch PLC state.

2) Latched

The data in register will be retained while switch PLC state. The latched register range can be modified by parameters.

3) System-special

System-special data register D8000 ~ D8255 are used for controlling and monitoring a variety of work methods and components in PLC, such as battery voltage, scan time, and is the state of action and so on. The default value will be written into those registers while PLC power on.

4) Index registers V, Z

The index registers are same as common data registers, is 16-bit registers for data reading and writing. There are totally 64 registers, V0-V31, Z0-Z31.

The index registers can be used in combination with other registers or values by application instructions. But they cannot be used in combination with the basic instructions and step ladder diagram instruction.

5) File registers D

The file registers start from D1000 to D7999. File registers can be secured in the program memory in units of 500 points. File registers are actually setup in the parameter area of the PLC. For every block of 500 file registers allocated and equivalent block of 500 program steps are lost.

3.9 Pointers registers P, I

Pointers register P is used for entry address of jump program, and identification of sub-program starting address.

Pointer register I is used for identification of interrupted program starting address.

Devices numbered in: Decimal, i.e. P0 to P9, P10 to P19, I0 to I9, I10 to I19.

Table 3-11

PLC	Sub-program		Insert	Insert counter	Counter interrupt
	-	Jump to end			
LX3V (1S)	63 (P0-P62)	1 (P63)	6 I00_(X000), I10_(X001), I20_(X002), I30_(X003), I40_(X004), I50_(X005)	-	-
LX3V (2N)	127 (P0-P62) (P64-P127)	1 (P63)	6 I00_(X000), I10_(X001), I20_(X002), I30_(X003), I40_(X004), I50_(X005)	3 (I6_, I7_, I8_)	6 (I010, I020, I030, I040, I050, I060)
LX3VP	127 (P0-P62) (P64-P127)	1 (P63)	6 I00_(X000), I10_(X001), I20_(X002), I30_(X003), I40_(X004), I50_(X005)	3 (I6_, I7_, I8_)	6 (I010, I020, I030, I040, I050, I060)

LX3VE	127 (P0-P62) (P64-P127)	1 (P63)	6 I00_(X000), I10_(X001), I20_(X002), I30_(X003), I40_(X004), I50_(X005)	3 (I6_, I7_, I8_)	6 (I010, I020, I030, I040, I050, I060)
LX3VM	127 (P0-P62) (P64-P127)	1 (P63)	6 I00_(X000), I10_(X001), I20_(X002), I30_(X003), I40_(X004), I50_(X005)	3 (I6_, I7_, I8_)	6 (I010, I020, I030, I040, I050, I060)

Note:

The input X for interrupt register can’t be used for [high speed counter] and [SPD] instruction as the same time.

1) Sub-program pointer

As below demos show, the left one is for conditional jump with [CJ] instruction, the right one is for Sub-program call with [CALL] instruction.

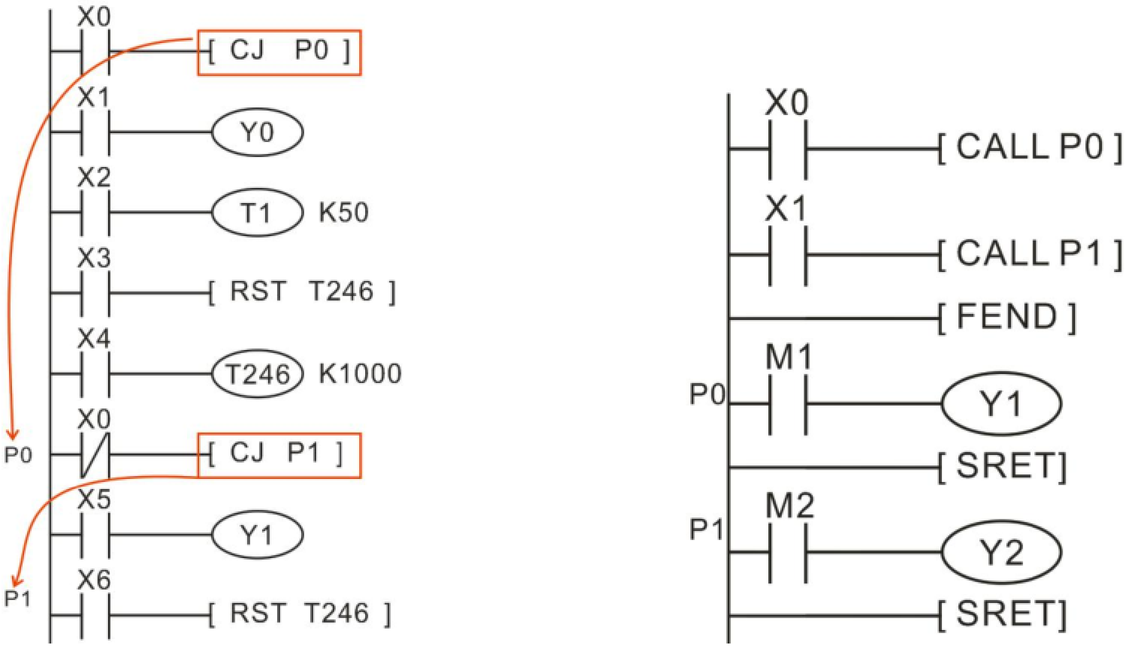


Figure 3-7

2) Interrupt pointer

An interrupt pointer and various usage of three, dedicated interrupt applied instructions;

- IRET: interrupt return
- EI: enable interrupt
- DI: disable interrupt

3) Usage of interrupt

- Input Interrupt: Receive signals from a particular input without being affected by the scan cycle of PLC;
- Timer Interrupt: The interrupt is repeatedly triggered at intervals of the specified time (10ms~99ms);
- Counter Interrupt: The interrupt is triggered according to the comparison result of the built-in high-speed counter of PLC;

3.10 Constant K, H, E

LX Series PLC could support five kinds of contacts for programming, the detailed as the following table shows.

Table 3-12

Format	Description
Decimal	The set value of timer and counter (K is a constant); The number of Auxiliary Relay(M), Timer(T), Counter(C), Status(S) and so on (the number of registers); The value and instruction action in the operand, which are applied (K is a constant);
Hexadecimal	As with the decimal, it is applied in the operand and the specific actions in the application instruction.
Binary	Using decimal number or hexadecimal number to design the value of the timer, counter or data register. However, in the internal PLC, these data is dealt with binary numbers. Moreover, when monitoring external devices, these registers will be converted to a decimal number automatically (16 hex can be converted as well).
Octal	It is used for distribute the register number of input relay and output relay. Use the binary values of [0-7, 10-17 ... 70-77, 100-107]. [8, 9] do not exist in the octal.
BCD	Binary-coded decimal (BCD) is a class of binary encodings of decimal

	numbers where each decimal digit is represented by a fixed number of bits, usually four or eight. Special bit patterns are sometimes used forseven segment display controlling.
BIN float	BIN float is used for calculation in PLC internal.
Decimal float	It is only used for monitoring and improving readability.

1) Constant K

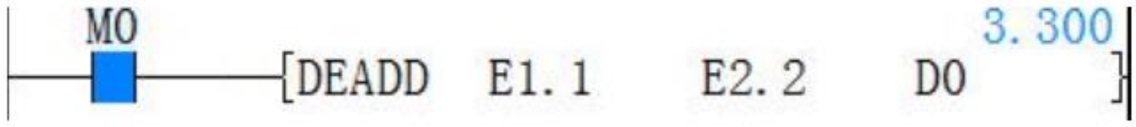
[K] is decimal integer symbol, mainly used for setting the value of the timer or counter or application instruction operand values. The value range in 16-bit is -32,768 – 32,767, the value range in 32-bit is -2, 147,483, 648 – 2, 147, 483, 647.

2) Constant H

[H] is hexadecimal numbers symbol, mainly used for setting the value of application instruction operand value. The value range in 16-bit instruction is 0000-FFFF, the value range in 32-bit instruction is 0000, 0000– FFFF, FFFF.

3) Constant E

[E] is single-precision floating symbol, mainly used for setting the value of application instruction operand value. It is only available in DECMP、DEZCP、DSINH、DCOSH、DTANH、DEBCD、DEBIN、DEADD、DESUB、DEMUL、DEDIV、DEXP、DLOGE、DLOG10、DESQR、DINT、DSIN、DCOS、DTAN、DASIN、DACOS、 DATAN、DRAD、DDEG instructions in LX3VP and LX3VE series. The value range is ±1.175495 E-38～±3.402823 E+38.



```
M0                                    3.300
──┤■├──[DEADD   E1.1    E2.2    D0        ]
```

3.11 system-special address list

Table 3-13

M	Description	LX1S	LX2N or later	D	Description	LX1S	LX2N or later
System operation							
M8000	RUN monitor, NO contact	O	O	D8000	Watchdog timer	O	O
M8001	RUN monitor, NC contact	O	O	D8001	PLC type and version LX3V/3V-A2:250**	O	O

					LX3V-A1: 220** LX3VP: 251** LX2V: 240** ** is viewed by D8101		
M8002	Initial pulse NO contact	O	O	D8002	Memory capacity 0002: 2K steps 0004: 4K steps 0008: 8K step	O	O
M8003	Initial pulse NC contact	O	O	D8003	Memory type default value is 0x10.	O	O
M8004	ON when one or more error flags from the range M8060to M8067 [except M8062]are ON	O	O	D8004	Error BCD code of M8060~M8067, the default value is 0.	O	O
M8005	Battery voltage Low	-	O	D8005	Battery voltage	-	O
M8006	Battery error latch	-	O	D8006	The level at which a battery voltage low is detected	-	O
M8007	Power loss has occurred more than 5ms, M8007&M8008 are ON	-	O	D8007	The number of time a momentary power failure has occurred since power ON.	-	O
M8008	Power loss has occurred	-	O	D8008	The time period before shutdown when a power failure occurs (default 10ms)	-	O
M8009	Power failure of 24V DC service supply	-	O	D8009	The device number of module, which affected by 24VDC power failure	-	O
Clock Devices							
M8010	Reserved	O	O	D8010	Current operation cycle / scan time in units of 0.1 msec	O	O
M8011	Oscillates in 10	O	O	D8011	Minimum cycle/	O	O

	msec cycles				scan time in units of 0.1 msec		
M8012	Oscillates in 100 msec cycles	O	O	D8012	Maximum cycle/ scan time inunits of 0.1 msec	O	O
M8013	Oscillates in 1 sec cycles	O	O	D8013	Seconds data for use with an RTC (0-59)	O	O
M8014	Oscillates in 1 min cycles	O	O	D8014	Minute data for use with anRTC (0-59)	O	O
M8015	When ON - clock stops, ON→OFF restarts clock	O	O	D8015	Hour data for use with an RTC (0-23)	O	O
M8016	When ON D8013 to 19 are frozen for display but clock continues	O	O	D8016	Day data for use with an RTC (1-31)	O	O
M8017	When pulsed ON set RTC to nearest minute	O	O	D8017	Month data for use with an RTC (1-12)	O	O
M8018	When ON Real Time Clockis installed	O	O	D8018	Year data for use with an RTC (2000-2099)	O	O
M8019	Clock data has been set outof range	O	O	D8019	Weekday data for use with an RTC (0-6)	O	O
Operation Flags							
M8020	Set when the result of anADDor SUBis "0"	O	O	D8020	Input filter setting for devicesX000 to X007 default is 10msec, (0-60)	O	O
M8021	Set when the result of a SUBis less than the min. negative number	O	O	D8021	Reserved		
M8022	Set when 'carry' occurs during an ADD orwhen an overflow occurs asa result of a data shift	O	O	D8022	Reserved		

	operation						
M8023	Reserved	O	O	D8023	Reserved		
M8024	Direction of BMOV	-	O	D8024	Reserved		
M8025	HSC mode	-	O	D8025	Reserved		
M8026	RAMP mode	-	O	D8026	Reserved		
M8027	PR 16 element data string	-	O	D8027	Reserved		
M8028	Switch100ms/10 ms timer	O	-	D8028	Current value of the Z index register	O	O
M8029	Instruction execution complete such as PLSR	O	O	D8029	Current value of the V index register	O	O
PLC Operation Mode							
M8030	Battery voltage is low but BATT.V LED not lit	-	O	D8030	Reserved		
M8031	Clear all unsaved memory	O	O	D8031	Reserved		
M8032	Clear all the saved memory	O	O	D8032	Reserved		
M8033	The device statuses and settings are retained when thePLC changes from RUN toSTOP and back into RUN	O	O	D8033	Reserved		
M8034	All of the physical switchgear for activating outputs is disabled. However, the program still operates normally.	O	O	D8034	Reserved		
M8035	Forced operation 1	O	O	D8035	Reserved		
M8036	Forced operation 2	O	O	D8036	Reserved		

M8037	Forced stop	O	O	D8037	Reserved		
M8038	Communication parameter setting flag	O	O	D8038	Reserved		
M8039	Constant scan	O	O	D8039	Constant scan time, default 0, in units of MS	O	O
Step Ladder (STL) Flags							
M8040	When ON STL state transfer is disabled	O	O	D8040	Up to 8 active STL states, from the range S0 to S899, are stored in D8040 to D8047 in ascending numerical order (Updated at END)	O	O
M8041	When ON STL transfer from initial state is enabled during automatic operation	O	O	D8041		O	O
M8042	A pulse output is given in response to a start input	O	O	D8042		O	O
M8043	On during the last state of ZERO RETURN mode	O	O	D8043		O	O
M8044	ON when the machine zero is detected	O	O	D8044		O	O
M8045	Disables the all output reset function when the operation mode is changed	O	O	D8045		O	O
M8046	ON when STL monitoring has been enable (M8047)	O	O	D8046		O	O
M8047	When ON D8040 to D8047 are enabled for active STL step monitoring	O	O	D8047		O	O
M8048	ON when annunciator	-	O	D8048	Reserved		

	monitoring has been enabled (M8049) and there is an active annunciator flag						
M8049	When ON D8049 is enabled for actove annunciator state monitoring.	-	O	D8049	Stores the lowest currently active annunciator from the range S900 to S999 (Updated at END)	-	O
Interrupt Control Flags							
M8050	I00□ disabled	O	O	D8050	Reserved		
M8051	I10□ disabled	O	O	D8051	Reserved		
M8052	I20□ disabled	O	O	D8052	Reserved		
M8053	I30□ disabled	O	O	D8053	Reserved		
M8054	I40□ disabled	O	O	D8054	Reserved		
M8055	I50□ disabled	O	O	D8055	Reserved		
M8056	I6□□ disabled	-	O	D8056	Reserved		
M8057	I7□□ disabled	-	O	D8057	Reserved		
M8058	I8□□ disabled	-	O	D8058	Reserved		
M8059	Counters disabled	-	O	D8059	Reserved		
Error Detection							
M8060	I/O configuration error	-	O	D8060	The first I/O number of the unit or block causing the error	-	O
M8061	PLC hardware error	O	O	D8061	Error code for hardware error	O	O
M8062	PLC communication error	-	O	D8062	Error code for PLC Communications error	-	O
M8063	Parallel link error	O	O	D8063	Error code for parallel link error	O	O
M8064	Parameter error	O	O	D8064	Error code identifying parameter error	O	O
M8065	Syntax error	O	O	D8065	Error code identifying syntax error	O	O
M8066	Loop error	O	O	D8066	Error code identifying loop error	O	O

M8067	Operation error	O	O	D8067	Error code identifying operation error.	O	O
M8068	Operation error latch	O	O	D8068	Operation error step number latched	O	O
M8069	Reserved			D8069	Step numbers for found errors corresponding to flags M8065 to M8067	O	O
High-speed ring counter							
M8099	High-speed ring counter operation	O	O	D8099	High-speed ring counter, range: 0 to 32,767 in units of 0.1 ms	O	O
Other functions							
M8100	SPD (X000) pulse/ minute	O	O	D8100	Reserved	O	O
M8101	SPD (X001) pulse/ minute	O	O	D8101	Firmware sub-version LX3V/3VP: 160** LX2V: 240** The ** and D8001** combines a complete firmware version number	O	O
M8102	SPD (X002) pulse/ minute	O	O	D8102	User program capacity	O	O
M8103	SPD (X003) pulse/ minute	O	O	D8103	Reserved	O	O
M8104	SPD (X004) pulse/ minute	O	O	D8104	The AC/DE time for DRVI, DRVA, [100 ms default value] it effected by M8135 (Y0), it must be the same as D8165.	O	O
M8105	SPD (X005) pulse/ minute	O	O	D8105	The AC/DE time for DRVI, DRVA, [100 ms default value] it effected by M8135 (Y1), it must be the same as D8166.	O	O

M8106	Reserved			D8106	The AC/DE time for DRVI, DRVA, [100 ms default value] it effected by M8135 (Y2), it must be the same as D8167.	O	O
M8107	Reserved			D8107	The AC/DE time for DRVI, DRVA, [100 ms default value] it effected by M8135 (Y3), it must be the same as D8168.	O	O
M8108	Reserved			D8108	Reserved		
M8109	Output refresh error	O	O	D8109	Output refresh error device number;	O	O
COM1 communication settings							
M8110	Reserved			D8110	Com1 port setting (only available in 22319, 24320, 25007 or later)	O	O
M8111	Reserved			D8111	Reserved		
M8112	Reserved			D8112	Reserved		
M8113	Reserved			D8113	Reserved		
M8114	Reserved			D8114	Reserved		
M8115	Reserved			D8115	Reserved		
M8116	Reserved			D8116	Reserved		
M8117	Reserved			D8117	Reserved		
M8118	Reserved			D8118	Reserved		
M8119	Reserved			D8119	Reserved		
COM2 communication settings							
M8120	Reserved			D8120	Com2 port setting, the default value is 0	O	O
M8121	Sending and waiting (RS instruction)	O	O	D8121	Station number settings, the default value is 1	O	O
M8122	Sending flag (RS instruction) Instruction execution status (MODBUS)	O	O	D8122	Amount of remaining data to be transmitted (Only for RS instruction) unit:0.1ms	O	O
M8123	Receiving complete flag (RS)	O	O	D8123	Amount of data already received	O	O

	Communication error flag (MODBUS)				(Only to RS instruction)			
M8124	Receiving (only to RS instruction)	O	O	D8124	Start character STX (Only to RS instruction)		O	O
M8125	Reserved			D8125	End character ETX (Only to RS instruction)		O	O
M8126	Reserved			D8126	Communication protocol setting, the default value is 0		O	O
M8127	Reserved			D8127	Starting address for PC protocol		O	O
M8128	Reserved			D8128	Data length for PC protocol		O	O
M8129	Timeout judgement	O	O	D8129	Timeout judgement, default value is 10 (100ms)		O	O
High speed & Position								
M8130	Selects comparison tables to be used with the HSZ instruction	O	O	D8130	Contains the number of the current record being processed in the HSZ comparison table		O	O
M8131		O	O	D8131	HSZ&PLSY speed mode		O	O
M8132	HSZ&PLSY speed mode	O	O	D8132	HSZ&PLAY speed mode frequency	Low	O	O
M8133		O	O	D8133		-		
M8134	Reserved			D8134	HSZ&PLAY speed mode pulses	Low	O	O
M8135	Reserved			D8135		High		
M8136	Reserved			D8136	total output pulse of Y000&Y001	Low	O	O
M8137	Reserved			D8137		High		
M8138	Reserved			D8138	Reserved			
M8139	Reserved			D8139	Reserved			
M8140	The CLR signal output function	O	O	D8140	Accumulated value of	Low	O	O

	of ZRN is valid				PLSY & PLSR output pulse in Y000			
M8141	Accumulator register of output pulse can latched when turn ON (D8136, D8137, D8140~D8143, D8150~D8153)	O	O	D8141		High		
M8142	Reserved			D8142	Accumulated value of PLSY & PLSR output pulse in Y001	Low	O	O
M8143	Reserved			D8143		High		
M8144	Reserved			D8144	Reserved			
M8145	Stop pulse output in Y000	O	O	D8145	Bias speed of DRVI & DRVA		O	O
M8146	Stop pulse output in Y001	O	O	D8146	Highest speed of DRVI & DRVA (default is 100,000)	Low	O	O
M8147	Monitor pulse output in Y000	O	O	D8147		High		
M8148	Monitor pulse output in Y001	O	O	D8148	ACC/DEC time of DRVI & DRVA (default is 100)		O	O
M8149	Monitor pulse output in Y002	O	O	D8149	Reserved			
M8150	Monitor pulse output in Y003	O	O	D8150	Accumulated value of PLSY & PLSR output pulse in Y002	Low	O	O
M8151	Reserved			D8151		High		
M8152	Stop pulse output in Y002	O	O	D8152	Accumulated value of PLSY & PLSR output pulse in Y003	Low	O	O
M8153	Stop pulse output in Y003	O	O	D8153		High		
M8154	Reserved			D8154	Reserved			
M8155	Reserved			D8155	Reserved			
Extend function								
M8156	Reserved			D8156	Define clear signal in Y0 (ZRN) (default is		O	O

					5=Y5)		
M8157	Reserved			D8157	Define clear signal in Y1 (ZRN) (default is 6=Y6)	O	O
M8158	Reserved			D8158	Define clear signal in Y2 (ZRN) (default is 7=Y7)	O	O
M8159	Reserved			D8159	Define clear signal in Y3 (ZRN) (default is 8=Y10)	O	O
M8160	SWAP function is XCH	-	O	D8160	Define clear signal in Y4 (ZRN) (default is 9=Y11)	O	O
M8161	Bit processing mode of ASC/RS/ASCII/HEX /CCD	O	O	D8161	Reserved		
M8162	High-speed connection in parallel mode	O	O	D8162	Reserved		
M8163	Reserved			D8163	Reserved		
M8164	Variable transmission points mode (FROM/TO)	-	O	D8164	Special transmission points mode (FROM/TO)	O	O
M8165	Reserved			D8165	When enable acceleration and deceleration time, ensure the values is the same as D8104's	O	O
M8166	Reserved			D8166	When enable acceleration and deceleration time, ensure the values is the same as D8105's	O	O
M8167	HEX processing function of SMOV	-	O	D8167	When enable acceleration and deceleration time, ensure the values is the same as D8106's	O	O
M8168	HEX processing function of HEY	-	O	D8168	When enable acceleration and	O	O

					deceleration time, ensure the values is the same as D8107's		
M8169	Reserved			D8169	Reserved		
Pulse catch				**Communication**			
M8170	X000 pulse catch	O	O	D8170	Reserved		
M8171	X001 pulse catch	O	O	D8171	Reserved		
M8172	X002 pulse catch	O	O	D8172	Reserved		
M8173	X003 pulse catch	O	O	D8173	Station number setting state	O	O
M8174	X004 pulse catch	O	O	D8174	Communication sub-station setting state	O	O
M8175	X005 pulse catch	O	O	D8175	Refresh range setting state	O	O
M8176	Reserved			D8176	Station number setting	O	O
M8177	Reserved			D8177	Communication sub-station setting	O	O
M8178	Reserved			D8178	Refresh range setting	O	O
M8179	Reserved			D8179	Retries setting	O	O
M8180	Reserved			D8180	Timeout setting	O	O
Communication				**Indexed addressing**			
M8181	Reserved			D8181	Reserved		
M8182	Reserved			D8182	No.2 bit device/ Content of Z1 device	O	O
M8183	Master transfers data error	O	O	D8183	No.3 bit device/ Content of V1 device	O	O
M8184	Slave 1 transfers data error	O	O	D8184	No.4 bit device/ Content of Z2 device	O	O
M8185	Slave 2 transfers data error	O	O	D8185	No.5 bit device/ Content of V2 device	O	O
M8186	Slave 3 transfers data error	O	O	D8186	No.6 bit device/ Content of Z3 device	O	O
M8187	Slave 4 transfers data error	O	O	D8187	No.7 bit device/ Content of V3 device	O	O
M8188	Slave 5 transfers data error	O	O	D8188	No.8 bit device/ Content of Z4 device	O	O
M8189	Slave 6 transfers data error	O	O	D8189	No.9 bit device/ Content of V4 device	O	O
M8190	Slave 7 transfers	O	O	D8190	No.10 bit device/	O	O

	data error				Content of Z5 device		
M8191	Data transferring	O	O	D8191	No.11 bit device/ Content of V5 device	O	O
M8192	Reserved			D8192	No.12 bit device/ Content of Z6 device	O	O
M8193	Reserved			D8193	No.13 bit device/ Content of V6 device	O	O
M8194	Reserved			D8194	No.14 bit device/ Content of Z7 device	O	O
M8195	Reserved			D8195	No.15 bit device/ Content of V7 device	O	O
M8196	Reserved			D8196	Reserved		
M8197	Reserved			D8197	Reserved		
M8198	Reserved			D8198	Reserved		
M8199	Reserved			D8199	Reserved		
Counters states				**Communication**			
M8200	C200 Control	O	O	D8200	Frequency multiplication of C251 device D8200=0: 1 frequency multiplication D8200=1: 2 frequency multiplication D8200=2: 4 frequency multiplication Note: HSCS, HSCR and HSCZ instructions can be used with frequency multiplication simultaneously. And this function is available in V311 or later version	O	O
M8201	C201 Control	O	O	D8201	Reserved		
M8202	C202 Control	O	O	D8202	Reserved		
M8203	C203 Control	O	O	D8203	Reserved		
M8204	C204 Control	O	O	D8204	Reserved		
M8205	C205 Control	O	O	D8205	Reserved		

M8206	C206 Control	O	O	D8206	Reserved		
M8207	C207 Control	O	O	D8207	Reserved		
M8208	C208 Control	O	O	D8208	Reserved		
M8209	C209 Control	O	O	D8209	Reserved		
M8210	C210 Control	O	O	D8210	Reserved		
M8211	C211 Control	O	O	D8211	Reserved		
M8212	C212 Control	O	O	D8212	Reserved		
M8213	C213 Control	O	O	D8213	Reserved		
M8214	C214 Control	O	O	D8214	Reserved		
M8215	C215 Control	O	O	D8215	Reserved		
M8216	C216 Control	O	O	D8216	Reserved		
M8217	C217 Control	O	O	D8217	Reserved		
M8218	C218 Control	O	O	D8218	Reserved		
M8219	C219 Control	O	O	D8219	Reserved		
M8220	C220 Control	O	O	D8220	D8220=1 to enable the new filtering methods (four points constitute a set of filter). When use new filtering methods, the filter time which set by D8020 is not valid. And before using this filtering methods, users need to set the filtering time for each X terminals (D8221~D8228), Filter time unit is ms. Note: This filter method only works on CPU IO, the IO in extension module is not invalid.	O	O
M8221	C221 Control	O	O	D8221	Low bits are for setting filter time of X0~X3; High bits are for setting filter time of	O	O

					X4~X7 Unit is ms		
M8222	C222 Control	O	O	D8222	Low bits are for setting filter time of X10~X13; High bits are for setting filter time of X14~X17 Unit is ms	O	O
M8223	C223 Control	O	O	D8223	Low bits are for setting filter time of X20~X23; High bits are for setting filter time of X24~X27 Unit is ms	O	O
M8224	C224 Control	O	O	D8224	Low bits are for setting filter time of X30~X33; High bits are for setting filter time of X34~X37 Unit is ms	O	O
M8225	C225 Control	O	O	D8225	Low bits are for setting filter time of X40~X43; High bits are for setting filter time of X44~X47 Unit is ms	O	O
M8226	C226 Control	O	O	D8226	Low bits are for setting filter time of X50~X53; High bits are for setting filter time of X54~X57 Unit is ms	O	O
M8227	C227 Control	O	O	D8227	Low bits are for setting filter time of X60~X63; High bits are for setting filter time of	O	O

						X64~X67 Unit is ms		
M8228	C228 Control		O	O	D8228	Low bits are for setting filter time of X70~X73; High bits are for setting filter time of X74~X77 Unit is ms	O	O
M8229	C229 Control		O	O	D8229	Reserved		
M8230	C230 Control		O	O	D8230	Reserved		
M8231	C231 Control		O	O	D8231	Reserved		
M8232	C232 Control		O	O	D8232	Reserved		
M8233	C233 Control		O	O	D8233	Reserved		
M8234	C234 Control		O	O	D8234	Reserved		
M8235	One phase one directional	C235 Control	O	O	D8235	Reserved		
M8236		C236 Control	O	O	D8236	Reserved		
M8237		C237 Control	O	O	D8237	Reserved		
M8238		C238 Control	O	O	D8238	Reserved		
M8239		C239 Control	O	O	D8239	Reserved		
M8240		C240 Control	O	O	D8240	Reserved		
M8241		C241 Control	O	O	D8241	Reserved		
M8242		C242 Control	O	O	D8242	Reserved		
M8243		C243 Control	O	O	D8243	Reserved		
M8244		C244 Control	O	O	D8244	Reserved		
M8245		C245 Control	O	O	D8245	Reserved		
M8246	2 phase bi-directional	C246 Control	O	O	D8246	Reserved		
M8247		C247 Control	O	O	D8247	Reserved		
M8248		C248	O	O	D8248	Reserved		

		Control						
M8249		C249 Control	O	O	D8249	Reserved		
M8250		C250 Control	O	O	D8250	Reserved		
M8251	A/B phase	C251 Control	O	O	D8251	Reserved		
M8252		C252 Control	O	O	D8252	Reserved		
M8253		C253 Control	O	O	D8253	Reserved		
M8254		C254 Control	O	O	D8254	Reserved		
M8255		C255 Control	O	O	D8255	Reserved		

4. Instruction lists

4.1 Basic program instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Basic instructions	LD	Operation contact type NO (normally open)	○	○	○	○	56
	LDI	Operation contact type NC (normally closed)	○	○	○	○	56
	OUT	Final logical operation type coil drive	○	○	○	○	56
	AND	Serial connection of NO(normally open)	○	○	○	○	58
	ANI	Serial logical, operation contact type NC(normally closed) contacts	○	○	○	○	58
	OR	Parallel, connection of NO (normally open) contacts	○	○	○	○	59
	ORI	Parallel, connection of NC (normally closed) contacts	○	○	○	○	59
	LDP	Initial logical, operation -Rising edge pulse	○	○	○	○	56
	LDF	Initial logical, operation falling/trailing edge pulse	○	○	○	○	56
	ANDP	Serial connection of Rising edge pulse	○	○	○	○	58
	ANDF	Serial connection of falling/ trailing edge pulse	○	○	○	○	58
	ORP	Parallel, connection of NO Rising edge pulse	○	○	○	○	59

	ORF	Parallel connection of Falling/trailing edge pulse	○	○	○	○	59
	ORB	Serial connection of multiple parallel circuits	○	○	○	○	60
	ANB	Serial connection of multiple parallel circuits	○	○	○	○	60
	MPS	Stores the current result of the internal PLC operations	○	○	○	○	63
	MRD	Reads the current result of the internal PLC operations	○	○	○	○	63
	MPP	Pops (recalls and removes) the currently stored result	○	○	○	○	63
	MC	Denotes the start of a master control block	○	○	○	○	62
	MCR	Denotes the end of a master control block	○	○	○	○	62
	INV	Invert the current result of the internal PLC operations	○	○	○	○	61
	PLS	Rising edge pulse	○	○	○	○	65
	PLF	Falling / trailing edge pulse	○	○	○	○	65
	SET	Sets a bit device permanently ON	○	○	○	○	66
	RST	Resets a bit device permanently OFF	○	○	○	○	66

4.2 Step ladder instructions list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Step ladder instructions	STL	STL programming start instruction	○	○	○	○	316

	RET	STL programming end instruction	○	○	○	○	316

4.3 Program Flow instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Program flow instructions	CJ	Conditional jump	○	○	○	○	67
	CALL	Call Subroutine	○	○	○	○	69
	EI	Enable Interrupt	○	○	○	○	70
	DI	Disable Interrupt	○	○	○	○	70
	WDT	Watchdog Timer	○	○	○	○	73
	FOR	Start of a For/Next Loop	○	○	○	○	74
	NEXT	End a For/Next Loop	○	○	○	○	74

4.4 Move and Compare instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Move and Compare instructions	CMP	Compare	○	○	○	○	76
	ZCP	Zone compare	○	○	○	○	77
	MOV	Move	○	○	○	○	78
	SMOV	Shift move	-	○	○	○	79
	CML	Compliment	-	○	○	○	81
	BMOV	Block move	○	○	○	○	83
	FMOV	Fill move	-	○	○	○	84
	XCH	Exchange	-	○	○	○	85
	BCD	Binary coded decimal	○	○	○	○	86
	BIN	Binary	○	○	○	○	87

4.5 Arithmetic and Logical Operations instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Arithmetic	ADD	Addition	○	○	○	○	109

and Logical Operations instructions	SUB	Subtraction	○	○	○	○	111
	MUL	Multiplication	○	○	○	○	112
	DIV	Division	○	○	○	○	114
	INC	Increment	○	○	○	○	116
	DEC	Decrement	○	○	○	○	117
	WAND	Word AND	○	○	○	○	118
	WOR	Word OR	○	○	○	○	119
	WXOR	Word exclusive OR	○	○	○	○	120
	NEG	Negation	-	○	○	○	121

4.6 Rotation and Shift instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Rotation and Shift instructions	ROR	Rotation right	○	○	○	○	144
	ROL	Rotation left	○	○	○	○	145
	RCR	16-bit rotation right with carry	-	○	○	○	146
	RCL	16-bit rotation left with carry	-	○	○	○	147
	SFTR	(bit) shift right	○	○	○	○	148
	SFTL	(bit) shift left	○	○	○	○	149
	WSFR	word shift right	-	○	○	○	150
	WSFL	word shift left	-	○	○	○	151
	SFWR	shift register write	○	○	○	○	152
	SFRD	shift register read	○	○	○	○	153

4.7 Data operation instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Data operation instructions	ZRST	Zone reset	○	○	○	○	88
	DECO	Decode	○	○	○	○	89
	ENCO	Encode	○	○	○	○	90
	SUM	The sum of active bits	-	○	○	○	92
	BON	Check specified bit	-	○	○	○	93

		status					
	MEAN	Mean	-	○	○	○	94
	ANS	(timed) annunciator set	-	○	○	○	95
	ANR	Annunciator reset	-	○	○	○	96
	SQR	Square root	-	○	○	○	97
	FLT	Float	-	○	○	○	98
	SWAP	High and low bit conversion	-	○	○	○	99

4.8 High-speed Processing Instruction

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
High-speed Processing Instruction	REF	Refresh	○	○	○	○	123
	REFF	Refresh and filter adjust	-	○	○	○	125
	MTR	Input matrix	○	○	○	○	126
	DHSCR	High speed counter set	○	○	○	○	128
	DHSCS	High speed counter reset	○	○	○	○	129
	DHSZ	High speed counter zone compare	-	○	○	○	131
	SPD	Speed detect	○	○	○	○	133
	PLSY	16-bit pulse Y output	○	○	○	○	134
	PWM	Pulse width modulation	○	○	○	○	136
	PLSR	Ramp pulse output	○	○	○	○	138

4.9 ECAM instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
ECAM instruction	DECAM	ECAM configuration	-	-	-	○	178
	DEGEAR	Electronic gear	-	-	-	○	185

		configuration					
	ECAMTBX	Create E-CAM datasheet	-	-	-	○	189

4.10 External I/O Devices instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
External I/O device instruction	TKY	Ten key input	-	○	○	○	154
	HKY	Hexadecimal input	-	○	○	○	156
	DSW	Digital switch (thumbwheel input)	○	○	○	○	158
	SEGD	Seven segment decoder	-	○	○	○	160
	SEGL	Seven segment with latch	○	○	○	○	162
	ARWS	Arrow switch	-	○	○	○	165
	ASC	ASCII code	-	○	○	○	167
	PR	“print” to a display	-	○	○	○	169
	FROM	Read from a special function block	○	○	○	○	171
	TO	Write to a special function block	○	○	○	○	173
	GRY	Converts binary integer to GRAY code	-	○	○	○	175
	GBIN	Converts GRAY CODE to binary	-	○	○	○	177

4.11 External Devices instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
External devices instruction	RS	RS communications	○	○	○	○	227
	RS2	RS2 communications	○	○	○	○	231
	CPAVL	Serial port settings	○	○	○	○	250
	CPAVL	Ethernet port settings	○	○	○	○	246
	PRUN	Octal bit transmission	○	○	○	○	253
	ASCI	hexadecimal to ASCII	○	○	○	○	255

	HEX	ASCII to hexadecimal	○	○	○	○	257
	CCD	check code	○	○	○	○	259
	PID	PID control loop	○	○	○	○	261

4.12 Floating Point instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Floating point instruction	DECMP	Binary floating point data compare	-	○	○	○	265
	DEZCP	Binary floating point zone compare	-	○	○	○	266
	DEBCD	Binary to BCD floating point data conversion	-	○	○	○	267
	DEBIN	BCD to Binary floating point data conversion	-	○	○	○	268
	DEADD	Binary floating point addition	-	○	○	○	269
	DESUB	Binary floating point subtraction	-	○	○	○	270
	DEMUL	Binary floating point multiplication	-	○	○	○	272
	DEDIV	Binary floating point division	-	○	○	○	274
	DESQR	Binary floating point square root	-	○	○	○	276
	INT	16-bit binary floating point to integer	-	○	○	○	277
	DSIN	Sin operation	-	○	○	○	278
	DCOS	Cosine operation	-	○	○	○	279
	DTAN	Tangent operation	-	○	○	○	280
	DASIN	Calculate radian value, according to the corresponding value of SIN	-	-	○	○	281
	DSINH	Binary floating point operation of Hyperbolic	-	-	○	○	284

		Sine function SINH					
	DACOS	Calculate radian value, according to the corresponding value of COS	-	-	○	○	282
	DCOSH	Binary floating point operation of Hyperbolic Cosine function COSH	-	-	○	○	285
	DATAN	Calculate radian value, according to the corresponding value of TAN	-	-	○	○	286
	DTANH	Binary floating point operation of Hyperbolic Tangent function TANH	-	-	○	○	286
	DEXP	Perform exponent operation of binary floating-point number to base e (2.71828)	-	-	○	○	289
	DLOG10	Perform common logarithm operation of binary floating-point number to base 10	-	-	○	○	290
	DLOGE	Perform natural logarithm operation of binary floating-point number to base e (2.71828)	-	-	○	○	291

4.13 Positioning Instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Positioning instruction	DABS	Absolute current value read	○	○	○	○	215
	ZRN	Setting of zero return speed	○	○	○	○	217
	PLSV	Variable speed pulse	○	○	○	○	219

		output					
	DRVI	Relative position control	○	○	○	○	222
	DRVA	Absolute position control	○	○	○	○	224

4.14 Real Time Clock Control

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Real time clock control	TCMP	time compare	○	○	○	○	100
	TZCP	time zone compare	○	○	○	○	102
	TADD	time addition	○	○	○	○	103
	TSUB	time subtraction	○	○	○	○	104
	TRD	read RTC data	○	○	○	○	105
	TWR	set RTC data	○	○	○	○	106
	HOUR	16-bit chronograph	○	○	○	○	108

4.15 Handy Instructions list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Handy instruction	IST	initial state	○	○	○	○	190
	SER	Search	-	○	○	○	198
	ABSD	Absolute drum	○	○	○	○	200
	INCD	Incremental drum	○	○	○	○	202
	TTMR	Teaching timer	-	○	○	○	204
	STMR	Special timer - definable	-	○	○	○	206
	ALT	Alternate state	○	○	○	○	208
	RAMP	Ramp - variable value	○	○	○	○	209
	ROTC	Rotary table control	-	○	○	○	211
	SORT	Sort data	-	○	○	○	213

4.16 Circular interpolation instruction list

Instruction type	Instruction	Decryption	LX3 V	LX3 VP	LX3 VE	LX3 VM	Page
Circular interpolation instruction	G90G01	Absolute position line interpolation	-	-	-	○	292
	G91G01	Relative position line interpolation	-	-	-	○	295
	G90G02	Absolute position of the clockwise circular interpolation	-	-	-	○	298
	G91G02	Relative position clockwise circular interpolation	-	-	-	○	301
	G90G03	Absolute position anticlockwise circular interpolation	-	-	-	○	304
	G91G03	Relative position anticlockwise circular interpolation	-	-	-	○	307

4.17 Inline Comparisons Instruction list

Instruction type	Instruction	Description	LX3V (1S)	LX3V (2N)	LX3VP	LX3VE	Page
Inline comparisons instruction	LD=	Comparison of 16-bit data (==)	○	○	○	○	310
	LD>	Comparison of 16-bit data (>)	○	○	○	○	310
	LD<	Comparison of 16-bit data (<)	○	○	○	○	310
	LD<>	Comparison of 16-bit data (<>)	○	○	○	○	310
	LD<=	Comparison of 16-bit data (<=)	○	○	○	○	310
	LD>=	Comparison of 16-bit data (>=)	○	○	○	○	310

	AND=	Comparison of 16-bit data (==)	○	○	○	○	312
	AND>	Comparison of 16-bit data (>)	○	○	○	○	312
	AND<	Comparison of 16-bit data (<)	○	○	○	○	312
	AND<>	Comparison of 16-bit data (<>)	○	○	○	○	312
	AND<=	Comparison of 16-bit data (<=)	○	○	○	○	312
	AND>=	Comparison of 16-bit data (>=)	○	○	○	○	312
	OR=	Comparison of 16-bit data (==)	○	○	○	○	314
	ORD=	Comparison of 32-bit data (==)	○	○	○	○	314
	OR>	Comparison of 16-bit data (>)	○	○	○	○	314
	OR<	Comparison of 16-bit data (<)	○	○	○	○	314
	OR<>	Comparison of 16-bit data (<>)	○	○	○	○	314
	OR<=	Comparison of 16-bit data (<=)	○	○	○	○	314
	OR>=	Comparison of 16-bit data (>=)	○	○	○	○	314

5. Instruction description

5.1 Basic instructions

LD, LDI, LDP, OUT Instructions

1) Instruction description

LD, LDI takes 1 process step. LDP, LDF take 2 process steps. The operands of these 4 instructions can be X, Y, S, M, T, C.

The operand of OUT can be Y, S, T, M or C. Soft component Y and the general M takes 1 process step. S and special auxiliary relay M take 2 process steps. Timer T takes 3 process steps. Counter takes 3-5 process steps.

Connect the LDP and LDF instructions directly to the left hand bus bar. Or use LDP and LDF instructions to define a new block of program when using the ORB and ANB instructions (see later sections).

LDP is active for one scanning cycle after the associated device switches from OFF to ON. LDF is active for one scanning cycle after the associated device switches from ON to OFF.

The number of repetitions of LD, LDI, LDP and LDF instructions is below 8. That is, the maximum number of times used in series or parallel connection with the following ANB and ORB instructions is 8.

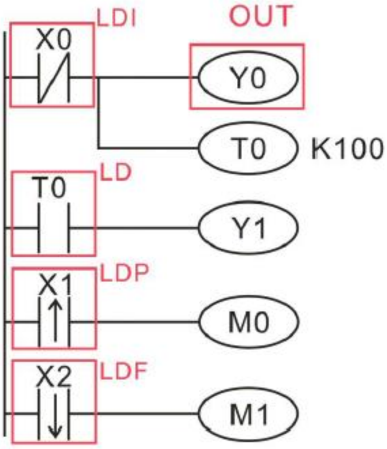
The steps of Y and normal M are 1; it is 2 for S and special M and 3 for T (timer), 3-5 for C (counter).

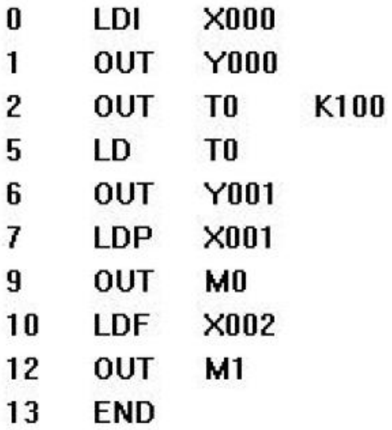
It is not possible to use the OUT instruction to drive ‘X’ type input devices. It is possible to connect multiple OUT instructions in parallel.

When using the OUT instruction to drive counter, when the front coil turns from ON to OFF, or from OFF to ON, the counter will add one.

2) Example

Ladder and Instruction List:





```
0    LDI    X000
1    OUT    Y000
2    OUT    T0     K100
5    LD     T0
6    OUT    Y001
7    LDP    X001
9    OUT    M0
10   LDF    X002
12   OUT    M1
13   END
```

Use LD, LDI, LDP, and LDF to connect with bus. Use OUT instruction to drive output coil. When using OUT instruction to drive timer or counter, it is no need to set the time value and count value. It can be a constant K, or indirectly set by the register.

AND, ADNI, ANDP, ANDF Instructions

1) Instruction description

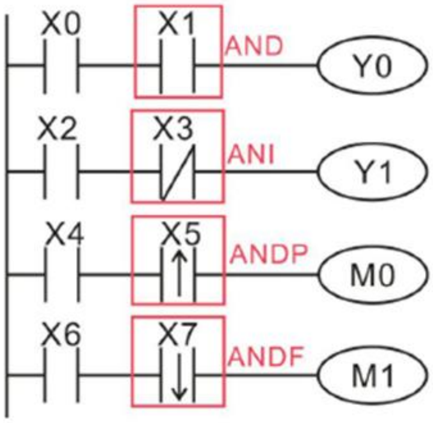
The steps of AND and ANI is 1, the steps of ANDP and ANDF is 2. The operands of these 4 instructions can be X, Y, S, M, T, C.

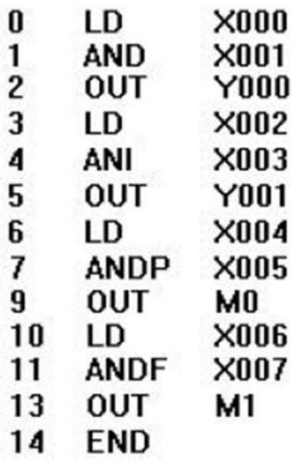
Use the AND and ANI instructions for serial connection of contacts. As many contacts as required can be connected in series. Use the ANDP and ANDF instructions for the serial connection of pulse contacts.

ANP is active for one scanning cycle after the associated device switches from OFF to ON. ANF is active for one scanning cycle after the associated device switches from ON to OFF.

2) Program example:

Ladder and instruction list:





```
0    LD     X000
1    AND    X001
2    OUT    Y000
3    LD     X002
4    ANI    X003
5    OUT    Y001
6    LD     X004
7    ANDP   X005
9    OUT    M0
10   LD     X006
11   ANDF   X007
13   OUT    M1
14   END
```

In this example, X0, X3, Y1 are connected with preceding contacts as the cascade contacts.

OR, ORI, ORP, ORF Instructions

1) Instruction description

The steps of OR and ORI is 1, the steps of ORP and ORF is 2. The operands of these 4 instructions can be X, Y, S, M, T, C.

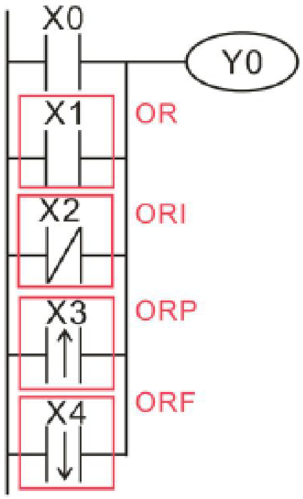
The instructions OR, ORI, ORP and ORF can only contact one circuit. For two or more series circuits, need to use ORB instruction when they connect in parallel.

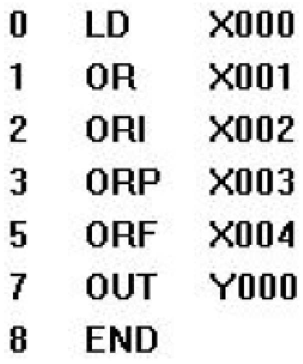
ORP is active for one program scan after the associated device switches from OFF to ON. ORF is active for one program scan after the associated device switches from ON to OFF.

There are no limitations to the number of parallel circuits when using an OR, ORI, ORP, ORF instructions and LD, LDI, LDP, LDF instruction.

2) Program example

Ladder and instruction list:





```
0   LD    X000
1   OR    X001
2   ORI   X002
3   ORP   X003
5   ORF   X004
7   OUT   Y000
8   END
```

ANB and ORB Instructions

1) Instruction description

The ANB instruction has no operand, and the number of steps is 1. The ORB instruction operand can be X, Y, S, M, T, C, the step number is 1.

Use the ANB instruction to connect multi-contact circuits (usually parallel circuit blocks) to the preceding circuit in series. Parallel circuit blocks are those in which more than one contact connects in parallel or the ORB instruction is used.

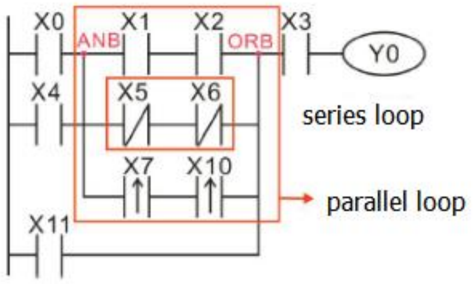
To declare the starting point of the circuit block, use a LD or LDI instruction. After Completing the parallel circuit block, connect it to the preceding block in series using the ANB instruction.

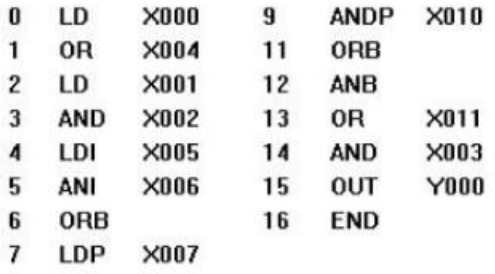
ANB and ORB instruction is an independent instruction.

When using ANB instruction in a batch, use no more than 8 LD and LDI instructions in the definition of the program blocks (to be connected in parallel).

2) Program example

Ladder and instruction list:





```
0   LD    X000     9    ANDP  X010
1   OR    X004     11   ORB
2   LD    X001     12   ANB
3   AND   X002     13   OR    X011
4   LDI   X005     14   AND   X003
5   ANI   X006     15   OUT   Y000
6   ORB            16   END
7   LDP   X007
```

ORB instruction is used in the end of each branch, not in the end of all branches, as it shown above.

ANB instruction is only used to connect parallel circuit blocks as the picture shown above.

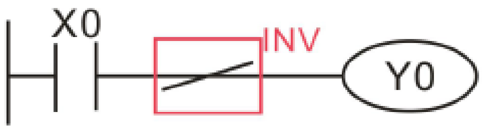
INV Instruction

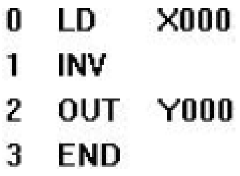
1) Instruction description

INV is the instruction which invert the result that before the INV instruction. And it has no operands. The instruction takes up 1 process step.

2) Program example

Ladder and instruction list:





```
0  LD    X000
1  INV
2  OUT   Y000
3  END
```

MC, MCR Instruction

1) Instruction description

The process step of MC instruction is 3 and the operands can be Y, M (except for special M). The process step of MCR instruction is 2 and the operands can be Y, M (except for special M).

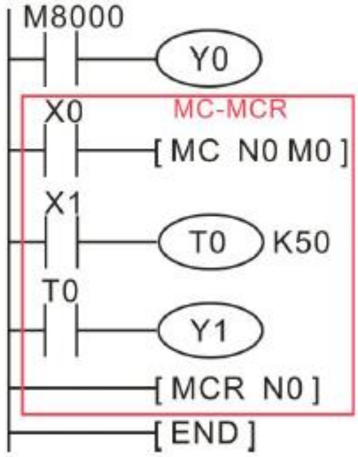
After the execution of an MC instruction, the bus line (LD, LDI point) shifts to a point after the MC instruction. An MCR instruction returns this to the original bus line.

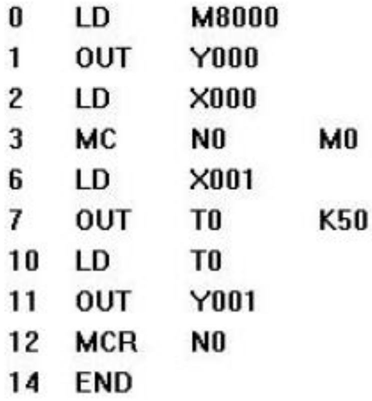
When using MC instruction, the number K of the nesting class increases in order, that is, only the K0 can nest K1. Instead, when using MCR instruction, it must return bus bar from large to small. Maximum nesting level is 7 (K6).

The MC instruction can be used as many times as necessary by changing the device number Y and M. Using the same device number twice is processed as a double coil

2) Program example

Ladder and instruction:





```
0    LD     M8000
1    OUT    Y000
2    LD     X000
3    MC     N0      M0
6    LD     X001
7    OUT    T0      K50
10   LD     T0
11   OUT    Y001
12   MCR    N0
14   END
```

When input X0=ON, all instructions between the MC and the MCR instruction execute.

When input X0=OFF, none of the instruction between the MC and MCR instruction execute; this resets all devices except for retentive timers, counters and devices driven by SET/RST instructions.

MPS, MRD and MPP Instruction

1) Instruction description

Instruction MPS, MRD and MPP have no operand, the steps of all of these three instructions is 1.

Use these instructions to connect output coils to the left hand side of a contact. Without these instructions connections can only be made to the right hand side of the last contact.

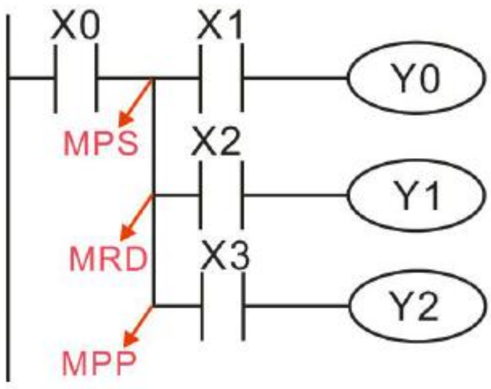
MPS stores the connection point of the ladder circuit so that further coil branches can recall the value later. MRD recalls or reads the previously stored connection point data and forces the next contact to connect to it.

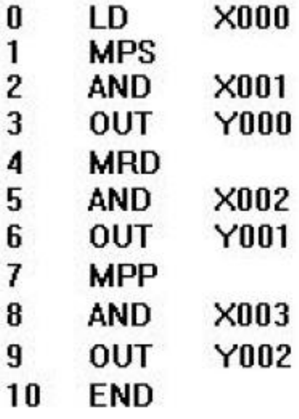
MPP pops (recalls and removes) the stored connection point. First, it connects the next contact, and then it removes the point from the temporary storage area. For every MPS instruction there must be a corresponding MPP instruction. The last contact or coil circuit must connect to an MPP instruction.

At any programming step, the number of active MPS-MPP pairs must be no greater than 11.

2) Program example

Ladder 1 and instruction list 1:

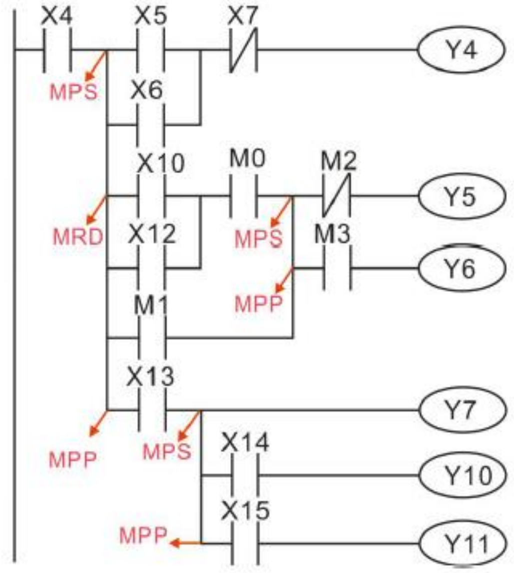




```
0    LD     X000
1    MPS
2    AND    X001
3    OUT    Y000
4    MRD
5    AND    X002
6    OUT    Y001
7    MPP
8    AND    X003
9    OUT    Y002
10   END
```

Example 1 uses a one-stage stack (one MPS, one MRD and one MPP).

Ladder 2 and instruction list 2:



```
0    LD    X004       15   OUT   Y005
1    MPS              16   MPP
2    LD    X005       17   AND   M3
3    OR    X006       18   OUT   Y006
4    ANB              19   MPP
5    ANI   X007       20   AND   X013
6    OUT   Y004       21   OUT   Y007
7    MRD              22   MPS
8    LD    X010       23   AND   X014
9    ORI   X012       24   OUT   Y010
10   AND   M0         25   MPP
11   OR    M1         26   AND   X015
12   ANB              27   OUT   Y011
13   MPS              28   END
14   ANI   M2
```

Example 2 uses a two-stage stack and it is mixed with OR, ORB, and ANB instructions.

PLS, PLF Instructions

1) Instruction description

The steps of PLS and PLF are 1, and operands can be Y and M (except for special M).

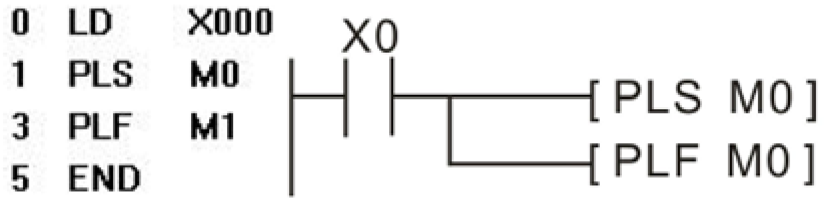
When a PLS instruction is executed, object devices Y and M operate for one operation cycle after the drive input signal has turned ON.

When a PLF instruction is executed, object devices Y and M operate for one operation cycle after the drive input signal has turned OFF.

If an M coil which is latched was used, then it only operates for one operation cycle for the first time.

2) Program example

Ladder and instruction list:



```
0  LD    X000
1  PLS   M0
3  PLF   M1
5  END
```

SET, RST Instructions

1) Instruction description

The operands of SET instruction are Y, M, S; RST operands are X, Y, S, M, T, C, D, V, Z.

The steps of Y and the general M is 1, the steps of S and special auxiliary relay M, timer T, counter C is 2, the steps of data register D and variable address register V and Z is 3.

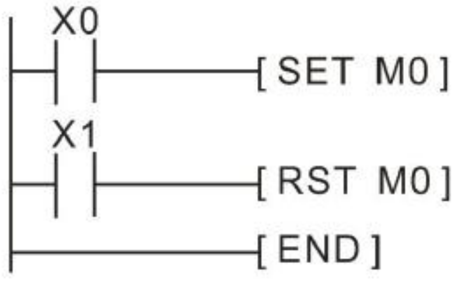
SET and RST instructions can be used for the same device as many times as necessary. However, the last instruction activated determines the current status.

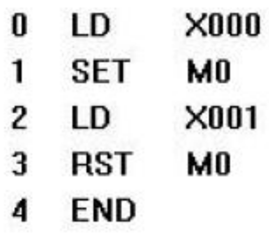
SET command set the soft component when the coil is connected, unless reset the soft component with RST instruction, it will remain 1. Similarly, the RST instructions reset the soft component, and it will remain 0, unless using the SET command to set.

It is also possible to use the RST instruction to reset the contents of data devices such as data registers, index registers, timer and counter. The effect is similar to moving ‘K0’ into the data device.

2) Program example

Ladder instruction list:





```
0   LD    X000
1   SET   M0
2   LD    X001
3   RST   M0
4   END
```

5.2 Applied instructions

5.2.1 Program flow

CJ instruction

1) Instruction description

This instruction disables the sequence control program from CJ, CJP instruction to point (p). It can help to decrease circle time (scan period) and implement the program applying double coil.

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
CJ	Conditional jump	16	No	CJ P* ※	3
CJP		16	Yes		3

※: LX3V (1S): P0-P63, LX3V (2N): P0-P127, LX3VP or later: P0-P127

When the CJ instruction is active it forces the program to jump to an identified program marker. While the jump takes place the intervening program steps are skipped.

When the CJ instruction is not active, program is executed sequentially.

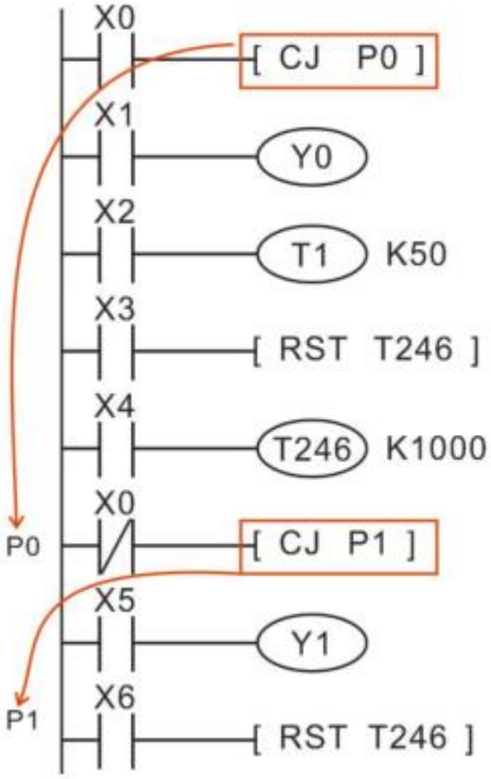
Timers and Canters will execute below operation when is active but skipped by CJ instruction.

Devices	Skipped	No skipped
T192~T199	Operating normally	Operating normally
Other timers	Stop timing	
C235~C255	Operating normally	
Other counters	Stop counting	

2) Program example

In the below example, program will execute as below.

- About X1, it drives Y1. Assuming X1 is ON and the CJ instruction is activated the load X1, out Y1 is skipped. Now even if X1 is turned OFF Y1 will remain ON while the CJ instruction forces the program to skip to the pointer P0. Once the CJ instruction is deactivated X1 will drive Y1 in the normal manner. This situation applies to all types of outputs, e.g. SET, RST, OUT, Y, M & S devices etc.



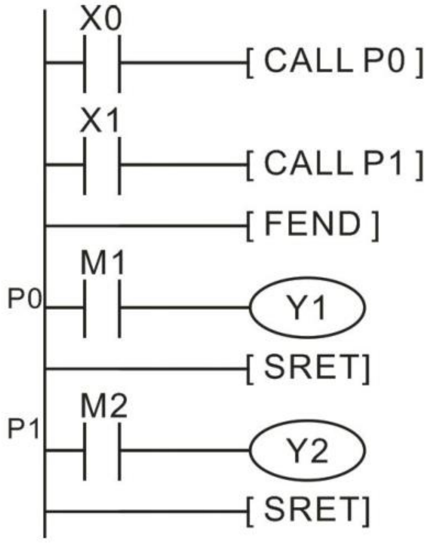
- Timers and counters will freeze their current values if they are skipped by a CJ instruction. The contents of T1 and T246 would not change/increase until the CJ instruction is no longer driven, i.e. the current timer value would freeze.
- If the reset instruction of the timer and counter is out of the jump, the timer coil and jump counter coil reset is effective

CALL instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
CALL	Subroutine call	16	No	CALL subroutine name	3
CALLP		16	Yes		3
SERT	Subroutine return	None	No	SERT	1
FEND	End main routine	None	No	FEND	1

2) Program example



In above program, when X0 is triggered, the CALL instruction is active it forces the program to run the subroutine associated with the called pointer (area identified as subroutine P0). Also when X1 is triggered, the CALL instruction will force the problem to run subroutine P1.

Subroutines can be nested for 4 levels including the initial CALL instruction.

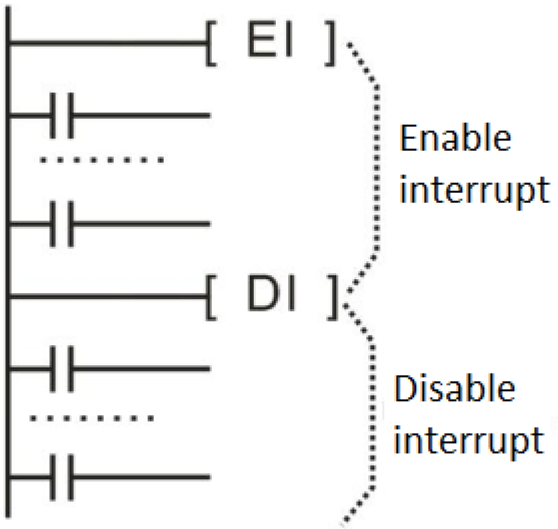
- A CALL instruction must be used in conjunction with FEND and SRET instructions. The program jumps to the subroutine pointer (located after an FEND instruction) and processes the contents until an SRET instruction is encountered.
- Error will occur if FEND instruction between CALL and IRET instructions, or between FOR and Next instructions.
- If more than FEND instructions please place subroutine between the last FEND and END instructions.

EI, DI instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
EI	Enable interrupt	16	No	--	1
DI	Disable interrupt	16	No		1
IRET	Return interrupt	16	No		1

Interrupt disabled is the default state, if there is no interrupt to insert the prohibited range in the program, the user cannot use the DI instruction.



2) Types and settings of interrupts

- External signal input interrupt: they can be defined to trigger interrupts by rising or falling edges. For an X signal that doesn’t need an immediate response, pulse capture function can also be used;
- Timer interrupts: they occur every fix period of 1ms~99ms;
- High speed interrupts: they are used with DHSCS instruction, interrupt occurs when the present value of a high speed counter reaches the setting value;

3) External signal input interrupt pointer and setting

Input No.	Pointer No.		Interrupt disable instruction
	Rising edge interrupt	Falling edge interrupt	

X000	I001	I000	M8050
X001	I101	I100	M8051
X002	I201	I200	M8052
X003	I301	I300	M8053
X004	I401	I400	M8054
X005	I501	I500	M8055

4) Timing interrupt pointer and setting

Input No.	Interrupt period (ms)	Interrupt disable instruction
I6□□	Input 1~99 to □□ in the instructions, for example,I605 which executes one timing interrupt every 5 ms	M8056
I7□□		M8057
I8□□		M8058

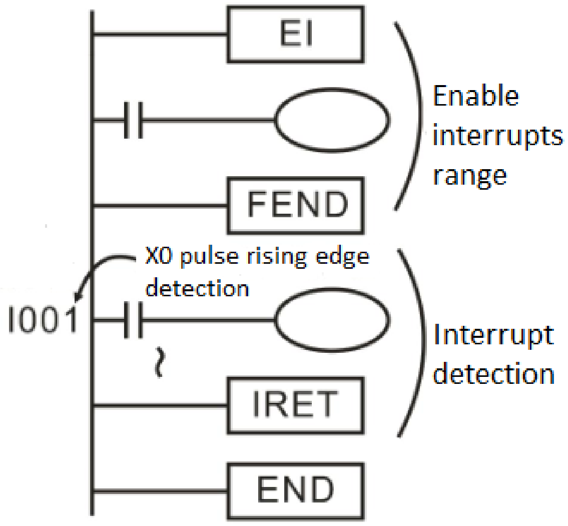
5) High Speed Count Interrupt Pointer and Settings

Input No.	Interrupt disable instruction
I010	M8059
I020	
I030	
I040	
I050	
I060	

Interrupt subroutine use a different number, that is, select a different port and interrupt trigger edge;

In the external input interrupt, the rising and falling interrupt numbers cannot be applied to the same X input. For an X input port, only one trigger edge can be used, the trigger edge is set by the pointer number;

- External signal input interrupt: Disable interrupts function of the corresponding X port, when M8050~M8055 set ON;
- Timer interrupts: Disable interrupts function of the corresponding timer, when M8056~M8058 set ON;
- High speed interrupts: Disable all high speed interrupts, when M8059 set ON;



6) Programming and execution characteristics of interrupts

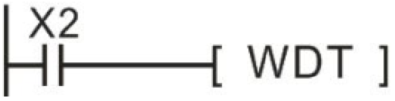
- Interrupts between the DI-EI instructions (interrupt disable interval), can also be memorized and executed after the EI instruction;
- The pointer number cannot be repeated;
- When multiple interrupts occur, execute in sequence, when multiple interrupts occur at the same time, execute in the priority level. Priority from high to low: high-speed counter interrupt, external input interrupt, time interrupt;
- Other interrupts are disabled during the execution of the interrupt detection. If program EI, DI in subroutine, it allows 2 levels;
- When the input relay is controlled during the interrupt processing, the input refresh instruction (REFF) can be used for reading the latest input state for high speed control;
- Please use T192~T199 for interrupt detection, other timers can work normally for interrupt detection. Also please pay more attention to 1ms latched timer;
- If specify the input interrupt pointer is I□0□, the input filter characteristics will be disabled for input relay, therefore, it is not necessary to use the REFF instruction and the special data register D8020 (input filter adjustment). In addition, the input filter of the input relay which is not used as an input interrupt pointer can be maintained for 10 ms (initial value).

WDT instruction

1) Instruction Description

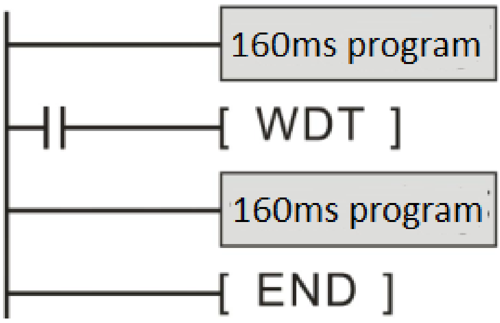
Name	Function	Bits	Pulse type	Instruction format	Step
WDT	Refresh the watch dog timer during a program scan	16	No	--	1
WDTP		16	yes		1

The WDT instruction refreshes the PLC's watchdog timer. The watchdog timer checks that the program scan time does not exceed an arbitrary time limit. It is assumed that if this time limit is exceeded there is an error at some point. The PLC will then stop operation to prevent any further errors from occurring. By causing the watchdog timer to refresh (driving the WDT instruction) the usable scan (program operation) time is effectively increased.



If the operation of user's program is too complex (for example, too many Cycles of calculation), an error may occur when the implementation of programming running out. If necessary, the program can use WDT instruction (for example, between the FOR ~ NEXT instruction can insert the instructions); If the program's scan time is longer than the value of D8000 (default is 200ms), users can insert the WDT instructions, then program will be divided into many pieces; every piece's scan time is less than 200ms or changes the setting value of D8000.

2) Program example



This program scan time is 320ms. We can divide program into two parts with the WDT instruction, so that each part of the program scan time is less than 200ms (D8000 default value).

FOR 、NEXT instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
FOR	Start position for the loop	16	No	FOR S_1	1
NEXT	End position for the loop	--	--	--	1

FOR instruction identifies the start position and the number of repeats for the loop, it must be an associated NEXT instruction. S_1 is for repeats time.

NEXT instruction identifies the end position for the loop.

FOR-NEXT instructions can be nested for 4 levels. This means that 4 FOR-NEXT loops can be sequentially programmed within each other. When using FOR-NEXT loops care should be not taken exceeding the PLC’s watchdog timer setting. The use of the WDT instruction and/or increasing the watchdog timer value is recommended.

Error occurs in below situations:

- NEXT instruction is before FOR instruction;
- No NEXT instruction for FOR instruction;
- The number of FOR instruction and NEXT instructions are inconsistent;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√

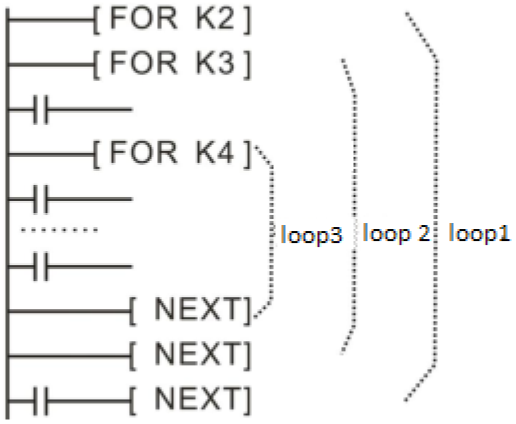
2) Program example

Example 1

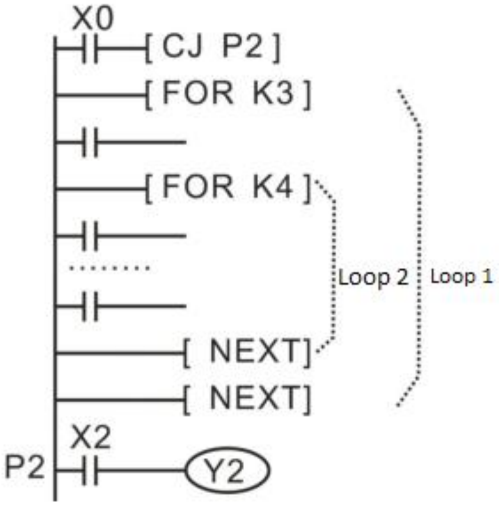
In below example, loop 1 executes 2 times, then back to main program, but loop2 will execute 3 times in every loop1, and loop3 will execute 4 times in every loop2.

The following number of operations would take place in ONE program scan:

- Number of loop C operations = 2 times
- Number of loop B operations = 6 times (C × B, 3× 2)
- Number of loop A operations = 24 times (C × B × A, 4 × 3 × 2)

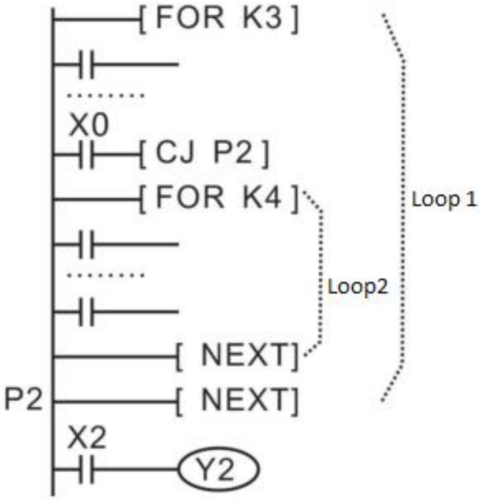


Example 2



In left example, CJ instruction can skip FOR, NEXT instructions. When the CJ instruction is active it forces the program to jump to P2, otherwise, FOR, NEXT instruction will be executed.

Example 3



In above example, when CJ instruction is active it forces the program to jump to P2 in Loop1, Loop2 is skipped. Otherwise Loop2 will be executed.

5.2.2 Move and compare

CMP instruction

1) Instruction description

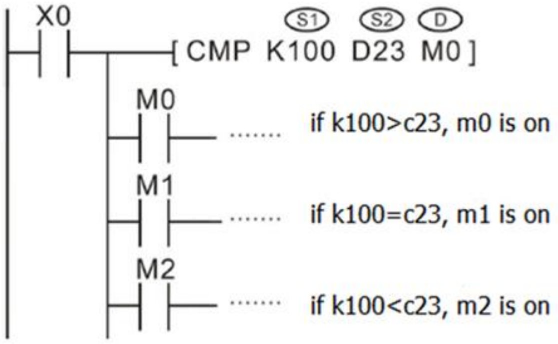
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
CMP	Comparison instruction	16	No	CMP S_1 S_2 D	7
CMPP		16	Yes		7
DCMP		32	No		13
DCMPP		32	Yes		13

This instruction compares two operational variables and outputs the comparison result to a specified bit variable. The operands are all algebra compared according to signed data.

D will occupy 3 continue bit variables address.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√	√	√	√	√	√	√	√	√	√	√
S_2					√	√	√	√	√	√	√	√	√	√	√	√
D		√	√	√												

2) Program example



When X0=ON, M0 or M1 or M2 will be ON.

When X0=OFF, CMP will not be executed, M0, M1 and M2 keep the initial state. If user wants to clear the result of comparison, RST or ZRST can be used.

By series or parallel M0, M1 and M2 to achieve the results of ≤ or ≥ or ≠.

ZCP instruction

1) Instruction description

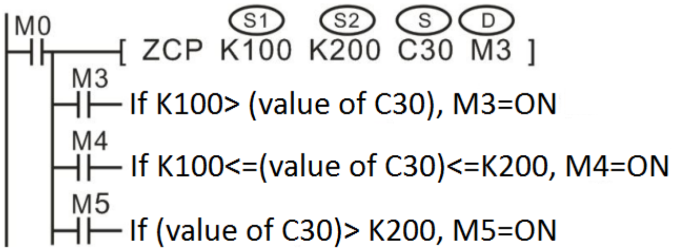
Name	Function	Bits	Pulse type	Instruction format	Step
ZCP	Regional comparison	16	No	ZCP S_1 S_2 S D	7
ZCPP		16	Yes		7
DZCP		32	No		13
DZCP		32	Yes		13

The operation is the same as the CMP instruction, except a single data value (S) is compared against a data range (S_1~S_2).

- S is less than S_1 and S_2 - bit device D is ON
- S is equal to or between S_1 and S_2 - bit device D +1 is ON
- S is greater than both S_1 and S_2 - bit device D +2 is ON
- S_1: Lower limit of comparison area
- S_2: Upper limit of comparison area
- S: Comparison variable
- D: Storage cell of comparison result; it will occupy three continuous bit variables.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
S					√	√		√	√	√	√	√	√	√	√	√
D		√	√	√												

2) Program example


M0
S1
S2
S
D
[ZCP K100 K200 C30 M3]
M3
If K100> (value of C30), M3=ON
M4
If K100<=(value of C30)<=K200, M4=ON
M5
If (value of C30)> K200, M5=ON

When X0=ON, M3 or M4 or M5 will be ON.

When X0=OFF, ZCP will not be executed, M3, m4 and m5 keep the initial state. If user wants to clear the result of comparison, RST or ZRST can be used.

MOV instruction

1) Instruction description

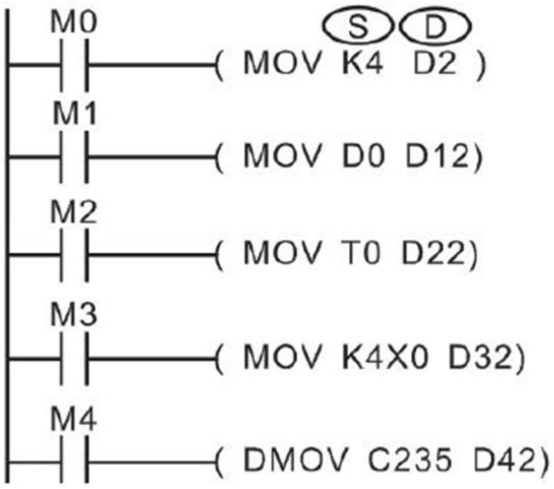
Name	Function	Bits	Pulse type	Instruction format	Step
MOV	Moves data from one storage area to a new storage area	16	No	MOV S D	7
MOVP		16	Yes		7
DMOV		32	No		13
DMOVP		32	Yes		13

The content of the source device (S) is copied to the destination (D) device when the control input is active. If the MOV instruction is not driven, no operation takes place.

For 32bit instructions (DMOV), two devices will be copied to the destination device, for example DMOV D1 D5, the result is D1→D5, D2→D6.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

2) Program example



When M0 is on, D2=K4, when M0 becomes off, D2 keeps the initial value. Only when user copy another value to D2 or power off the plc or set plc off and on again, the value of D2 will change.

SMOV instruction

1) Instruction description

Name	Function	Bit(bits)	Pulse type	Instruction format	Step
SMOV	Shift Move	16	No	SMOV S M_1 M_2 D n	11
SMOVP		16	Yes		11

This instruction copies a specified number of digits from a 4 digit decimal source (S) and places them at a specified location within a destination (D) number (also a 4 digit decimal). The existing data in the destination is overwritten.

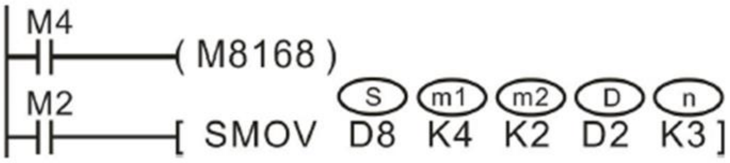
- M_1 - The source position of the 1st digit to be moved
- M_2 - The number of source digits to be moved
- n- The destination position for the first digit

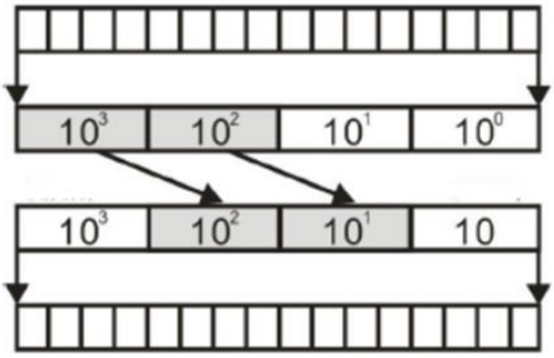
Allows BCD numbers to be manipulated in exactly the same way as the 'normal' SMOV manipulates decimal numbers, i.e. This instruction copies a specified number of digits from a 4 digit BCD source (S) and places them at a specified location within a destination (D) number (also a 4 digit BCD number).

To select the BCD mode the SMOV instruction is coupled with special M coil M8168 which is driven ON. Please remember that this is a 'mode' setting operation and will be active, i.e. all SMOV instructions will operate in BCD format until the mode is reset, i.e. M8168 is forced OFF.

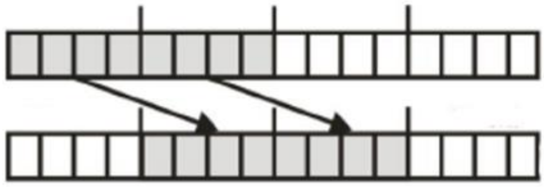
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√	√	√
M_1					√	√										
M_2					√	√										
D									√	√	√	√	√	√	√	√
n					√	√										

2) Program example





when m4=0, the process is as below:
D8(bin format 16bit)
↓ change automatically
D8(BCD format 4bit)
↓ bit move
D2(BCD format 4bit)
↓ change automatically
D2(BIN format 16bit)



when m4=1, the process is:
D8(hex format 4bit)
↓ bit mov
D2(hex format 4bit0

Suppose D8=K1234, D2=K5678, then when m8168 is off (bcd mode), set m2, then the value of D2 becomes K5128.

When m8168 is on (bin mode) and D8=H04D2=K1234, D2=H162E=K5678, set m2, then D2=H104E=K4174

CML instruction

1) Instruction description

Name	Function	Bits	Pulse type	Instruction format	Step
CML	Copies and inverts the source bit pattern to a specified destination	16	No	CML S D	5
CMLP		16	Yes		5
DCML		32	No		13
DCMLP		32	Yes		13

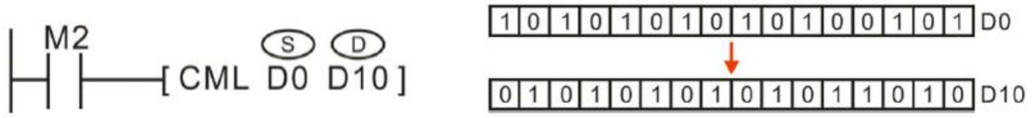
A copy of each data bit within the source device (S) is inverted and then moved to a designated destination (D).

This means each occurrence of a '1' in the source data will become a '0' in the destination data while each source digit which is '0' will become a '1'. If the destination area is smaller than the source data then only the directly mapping bit devices will be processed.

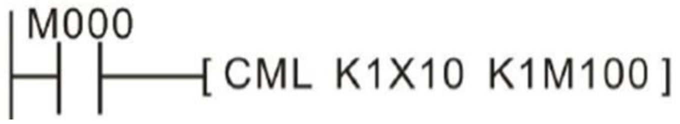
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

2) Program example

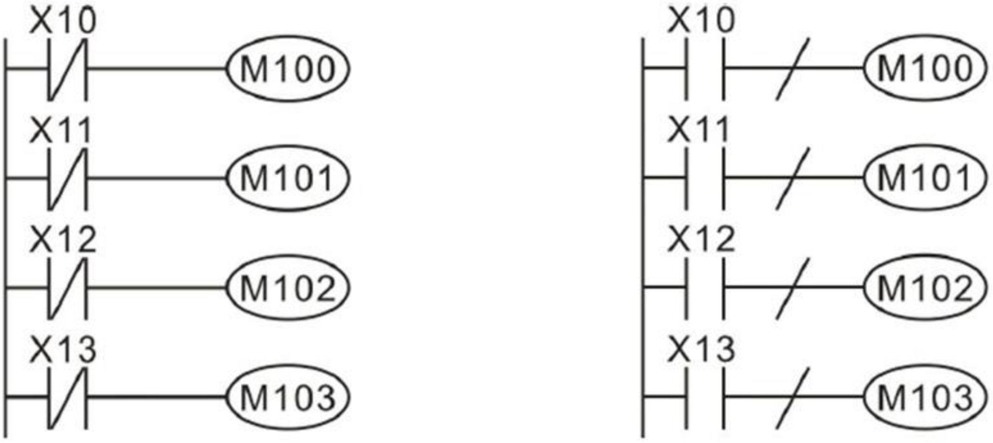
Example 1:


M2
S
D
[CML D0 D10]
1 0 1 0 1 0 1 0 1 0 1 0 0 1 0 1 D0
0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 D10

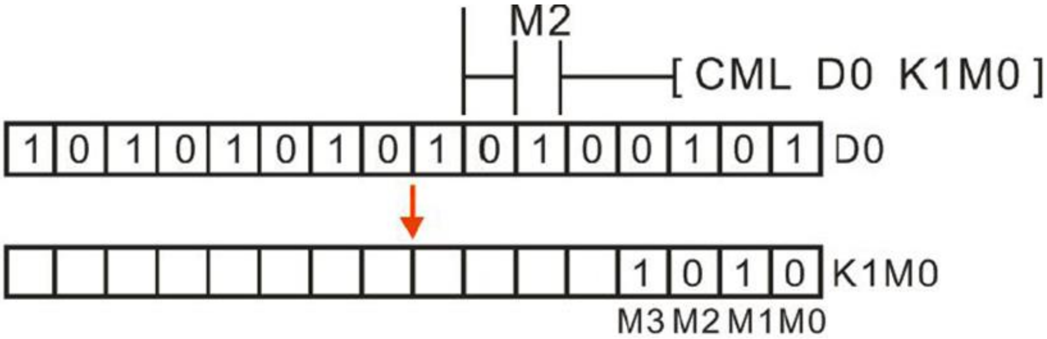
Example 2:



This program is equal to the below ladder diagrams.



Example 3:



1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1	D0
												1	0	1	0	K1M0
												M3	M2	M1	M0	

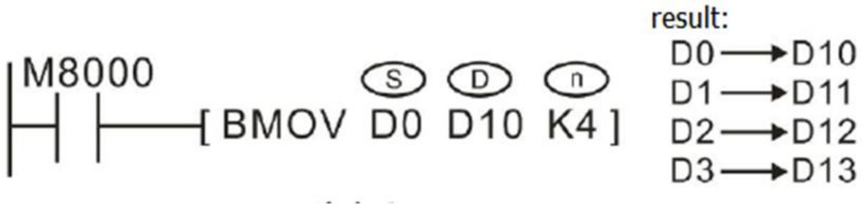
BMOV instruction

1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
BMOV	Copies a specified block of multiple data elements to a new destination	16	No	BMOV S D n	7
BMOVP		16	Yes		7

A quantity of consecutively occurring data elements can be copied to a new destination. The source data is identified as a device head address (S) and a quantity of consecutive data elements (n). This is moved to the destination device (D) for the same number of elements (n).

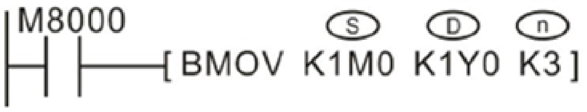
When the special variable is M8024=ON, the transmission direction is opposite, i.e. S becomes the destination address, D becomes the source address.



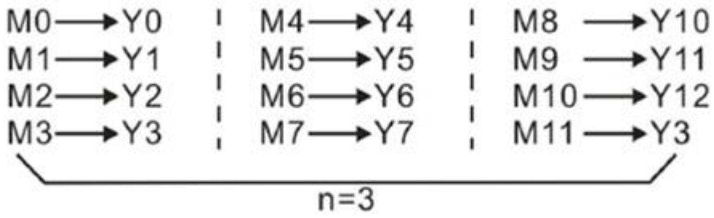
When the operand is bit device, the digit number of S and D need to be the same.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√		
D									√	√	√	√	√	√		
n	Constant n=1 to 512															

2) Program example



Result



FMOV instruction

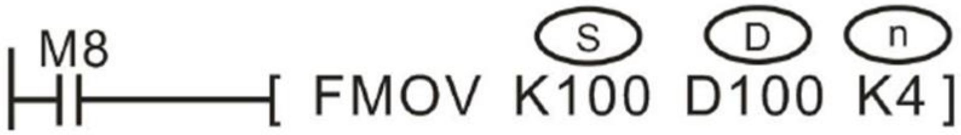
1) Instruction description

Name	Function	Bits	Pulse type	Instruction format	Step
FMOV	Copies a single data device to a range of destination devices	16	No	MOV S D n	7
FMOVP		16	Yes		7
DFMOV		32	No		13
DFMOVP		32	Yes		13

The data stored in the source device (S) is copied to every device within the destination range. The range is specified by a device head address (D) and a quantity of consecutive elements (n). If the specified number of destination devices (n) exceeds the available space at the destination location, then only the available destination devices will be written to.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√		
n	Constant, n=1 to512															

2) Program example



When M8 is on, k100→D100, k100→D101, k100→D102, k100→D103.

XCH instruction

1) Instruction description

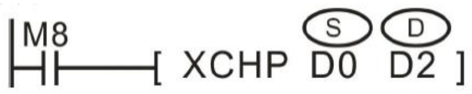
Name	Function	Bits	Pulse type	Instruction format	Step
XCH	Data in the designated devices is exchanged	16	No	XCH S D	5
XCHP		16	Yes		5
DXCH		32	No		9
DXCHP		32	Yes		9

The contents of the two destination devices S and D are swapped, i.e. the complete word devices are exchanged.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S									√	√	√	√	√	√		
D									√	√	√	√	√	√		

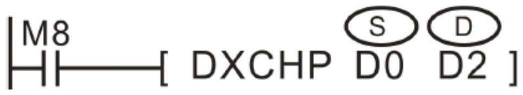
2) Program example

Example 1:



Before		After
D0=180	->	D0=200
D2=200	->	D2=180

Example 2:



Before		After
D0=180	->	D0=200
D1=150	->	D1=100
D2=200	->	D2=180
D3=100	->	D3=150

This function is equivalent to SWAP the bytes within each word of the designated devices D1 are exchanged when ‘byte mode flag’ M8160 is ON. Please note that the mode will remain active until it is reset, i.e. M8160 is forced OFF.

BCD instruction

1) Instruction description

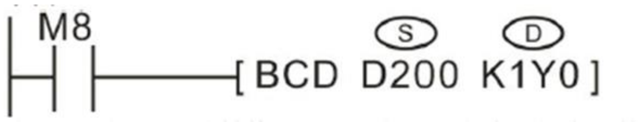
Name	Function	Bits	Pulse type	Instruction format	Step
BCD	Converts binary numbers to BCD equivalents / Converts floating point data to scientific format	16	No	BCD S D	5
BCDP		16	Yes		5
DBCD		32	No		9
DBCDP		32	Yes		9

The binary source data (S) is converted into an equivalent BCD number and stored at the destination device (D).

If the converted BCD number exceeds the operational ranges of 0 to 9,999 (16 bit operation) and 0 to 99,999,999 (32 bit operation) an error will occur. M8067 and m8068 will be ON, and D8067 will record the error code.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√	√	√
D								√	√	√	√	√	√	√	√	√

2) Programming example



The BIN value in D200 is converted to BCD value and the units' digit is saved in K1Y0 (Y0 to Y3).

- If D200=H000E (hex) =K14 (decimal), then Y0~Y3=0100(BIN).
- If D200=H0028 (hex) =K40 (decimal), then Y0~Y3=0000(BIN).

BIN instruction

1) Instruction description

Name	Function	Bits	Pulse type	Instruction format	Step
BIN	Converts BCD umbers to their binary equivalent /Converts scientific format data to floating point format	16	No	BIN S D	5
BINP		16	Yes		5
DBIN		32	No		9
DBINP		32	Yes		9

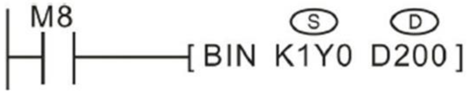
The BCD source data (S) is converted into an equivalent binary number and stored at the destination device (D). If the source data is not provided in a BCD format an error will occur. This instruction can be used to read in data directly from thumbwheel switches.

The value of S (BCD) ranges from 0 to 9999(16 bit) and 0 to 99999999(32 bit)

When the value of D is not BCD, there will be an error, and M8067 and M8068 will be ON.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√	√	√
D								√	√	√	√	√	√	√	√	√

2) Program example



When M8 is ON, K1Y0 (BCD value) will be converted into BIN and stored in the D200.

5.2.3 Data operation

ZRST instruction

1) Instruction description

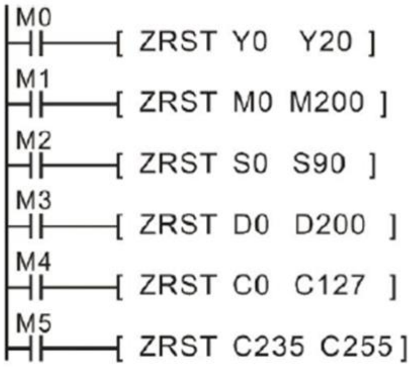
Name	Function	Bits	Pulse type	Instruction format	Step
ZRST	Reset a range of like devices in one operation	16	No	ZRST D_1 D_2	5
ZRSTP		16	Yes		5

The range of devices, including those specified as the two destinations are reset, D_1 and D_2 can be word or bit(Y, M or S). D_1 and D_2 must be the same kind device.

The number of D_1 should be smaller than D_2. If D_1 is 32bit counter, then D_2 must be 32bit counter too.

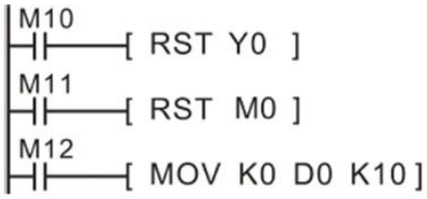
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D_1		√	√	√								√	√	√	√	
D_2		√	√	√								√	√	√	√	

2) Program example



```
M0
─┤├──[ ZRST Y0   Y20 ]
M1
─┤├──[ ZRST M0  M200 ]
M2
─┤├──[ ZRST S0   S90 ]
M3
─┤├──[ ZRST D0  D200 ]
M4
─┤├──[ ZRST C0  C127 ]
M5
─┤├──[ ZRST C235 C255]
```

Bit device(Y, M, S) and word device(T, C, D) can be set by RST; KnY, KnM and KnS and T, C , D can also be clear by FMOV, e.g



```
M10
─┤├──[ RST Y0 ]
M11
─┤├──[ RST M0 ]
M12
─┤├──[ MOV K0 D0 K10 ]
```

DECO instruction

1) Instruction description

Name	Function	Bits	Pulse type	Instruction format	Step
DECO	Source data value Q identifies the Qth bit of the destination device which will be turned ON	16	No	DECO S D n	7
DECOP		16	Yes		7

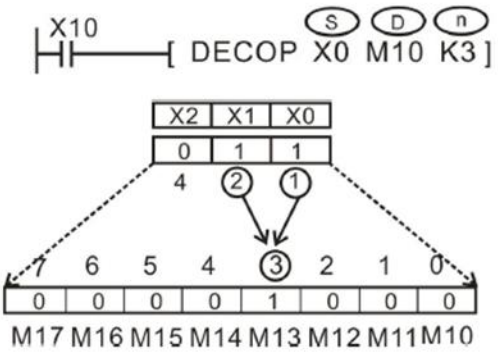
The lower n bits (n ≤ 4) of the source address are decoded to the destination address. When n ≤ 3, the high bit of the destination address will be 0.

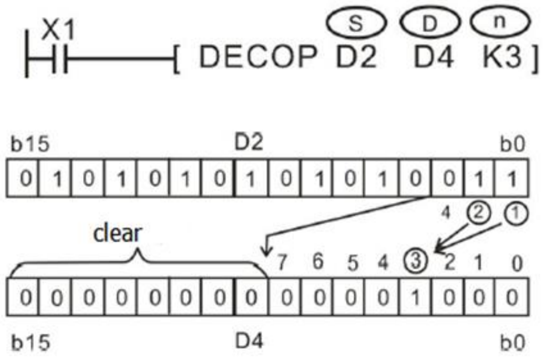
- If n=0, the instruction is not executed, if n is not equal to 0~8, then an error will occur.
- When n=8 and the D1, D2 are bit devices, means the points are 256.
- When the drive input is OFF, the instruction is not executed and the decoded output of the action is not changed.

Generally, DECOP is used in the real application.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	√	√	√	√	√							√	√	√	√	√
D		√	√	√								√	√	√		
n	Constant, n=1~8, if n=0, the instruction is not executed, if n is not equal to 0~8, then an error will occur.															

2) Program example





ENCO instruction

1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
ENCO	Encode	16	No	ENCO S D n	7
ENCOP		16	Yes		7

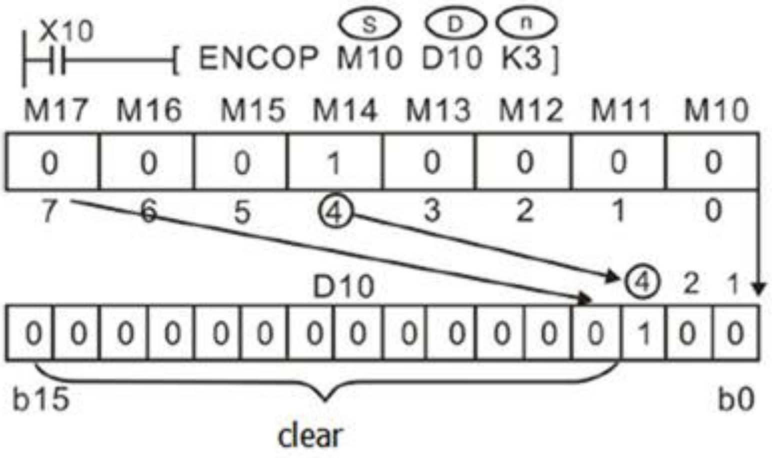
The highest active bit within the readable range has its location noted as a numbered offset from the source head address (S). This is stored in the destination register (D).

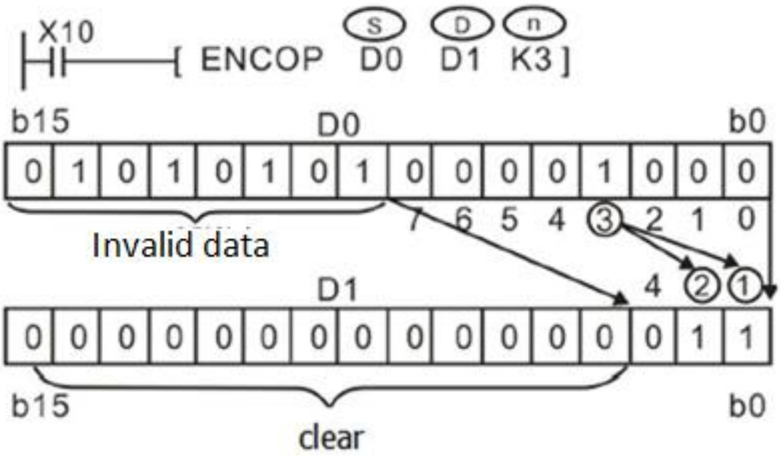
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	√	√	√	√	√	√						√	√	√	√	√
D		√	√	√								√	√	√		
n	Constant, n=1~8, if n=0, the instruction is not executed															

2) Points to note

- The readable range is defined by the largest number storable in a binary format within the number of destination storage bits specified by n, i.e. if n was equal to 4 bits a maximum number within the range 0 to 15 can be written to the destination device. Hence, if bit devices were being used as the source data, 16 bit devices would be used, i.e. the head bit device and 15 further, consecutive devices.
- If the stored destination number is 0 (zero) then the source head address bit is ON, i.e. The active bit has a 0 (zero) offset from the head address. However, if NO bits are ON within the source area, 0 (zero) is written to the destination device and an error is generated.
- When the source device is a data or word device n must be taken from the range 1 to 4 as there are only 16 source bits available within a single data word.

3) Program example


X10
ENCOP M10 D10 K3
S
D
n
M17 M16 M15 M14 M13 M12 M11 M10
0 0 0 1 0 0 0 0
7 6 5 4 3 2 1 0
4 2 1
D10
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
b15
b0
clear


X10
ENCOP D0 D1 K3
S
D
n
b15
D0
b0
0 1 0 1 0 1 0 1 0 0 0 0 1 0 0 0
7 6 5 4 3 2 1 0
Invalid data
D1
4 2 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
b15
b0
clear

SUM instruction

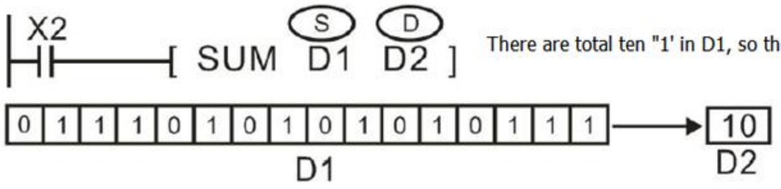
1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
SUM	The number (quantity) of active bits in the source data is stored in the destination device	16	No	SUM S D	5
SUMP		16	Yes		5
DSUM		32	No		9
DSUMP		32	Yes		9

The numbers of active (ON) bits within the source device (S), i.e. bits which have a value of “1" are counted. The count is stored in the destination register (D). If a double word format is used, both the source and destination devices use 32 bit, double registers. The destination device will always have its upper 16 bits set to 0 (zero) as the counted value can never be more than 32.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

2) Program example



There are total ten "1' in D1, so the value of D2 is 10.

BON instruction

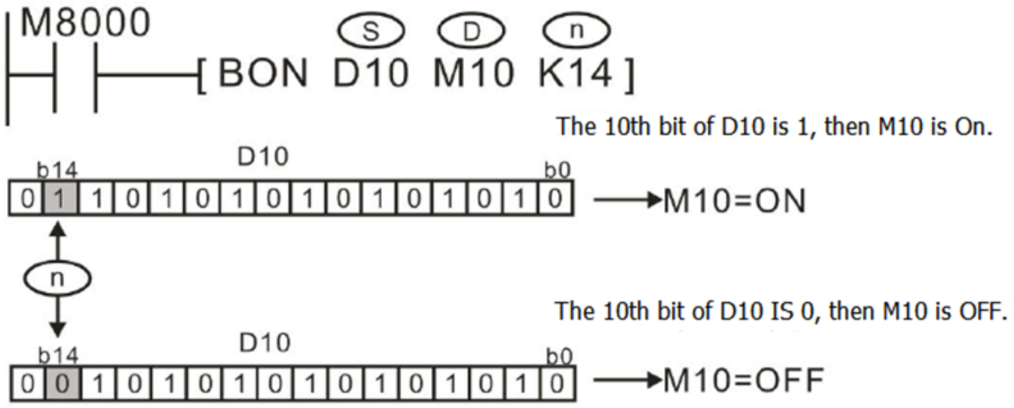
1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
BON	The status of the specified bit in the source device is indicated at the destination	16	No	BON S D n	7
BONP		16	Yes		7
DBON		32	No		13
DBONP		32	Yes		13

Determine the nth bit state of S and save the value to D.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D		√	√	√												
n	N=0~15 (16 bit); n=0~31(32bit)															

2) Program example



The 10th bit of D10 is 1, then M10 is On.

The 10th bit of D10 IS 0, then M10 is OFF.

When M10 turns from On to OFF, M10 keeps the initial value.

MEAN instruction

1) Instruction description

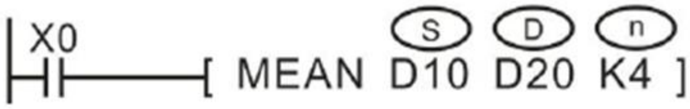
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
MEAN	Calculates the mean of a range of devices	16	No	MEAN S D n	7
MEANP		16	Yes		7
DMEAN		32	No		13
DMEANP		32	Yes		13

The range of source data is defined by operands S and n. S is the head address of the source data and n specifies the number of consecutive source devices used.

The value of all the devices within the source range is summed and then divided by the number of devices summed, i.e. n. This generates an integer mean value which is stored in the destination device (D). The remainder of the calculated mean is ignored.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√		
D									√	√	√	√	√	√	√	√
n	Constant, n=1~64															

2) Program example



(D10+D11+D12+D13)/4=D20

For example, D10=K5，D11=K5，D12=K15，D13=D52 , then D20=K19，the remainder 1 is ignored.

ANS instruction

1) Instruction description

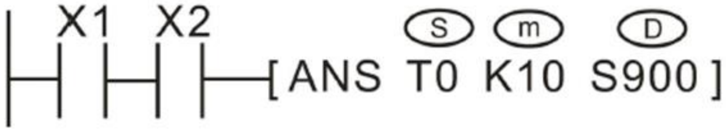
Name	Function	Bits	Pulse type	Instruction format	Step
ANS	(Timed) Annunciator Set	16	No	ANS S m D	7

This instruction, when energized, starts a timer (S) for n, 100 ms. when the timer completes its cycle the assigned annunciator (D) is set ON.

If the instruction is switched OFF during or after completion of the timing cycle the timer is automatically reset. However, the current status of the annunciator coil remains unchanged.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S												√				
D				√												
m	Constant, n=1~32767 (unit: 1000ms)															

2) Program example



If X1 and X2 are set for more than 1 second, S900 is set ON. After that, S900 will keep ON, even if X1 or X2 is reset (but T10 will be reset). If X1 and X2 are connected for less than 1 second, X1 or X2 becomes OFF. Then the timer will reset.

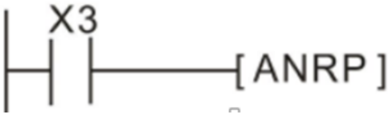
If M8049 (signal alarm is available) is set, the lowest number of S900~S999 that is set ON will be saved at D8049 (The lowest S number with the ON state). when any signal in S900~S999 is ON then M8048 is ON.

ANR instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ANR	The lowest active annunciator is reset on every operation of this instruction	16	No	ANR	1
ANRP		32	Yes		1

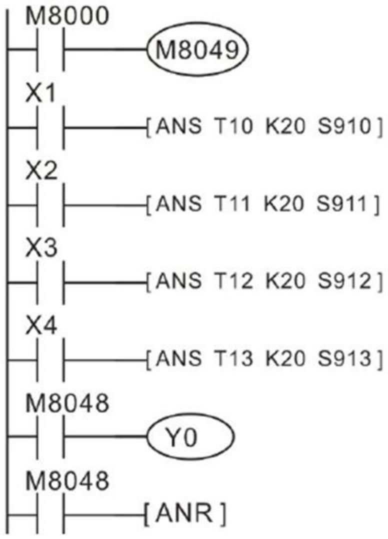
ANR is used for reset the alarm signal, e.g.



If X3 is ON, then the alarm signal from S900 to S999 will be reset. If there are more than one alarm signal, then the alarm signal with the smallest number will be reset.

If X3 is ON again, then the next alarm signal from S900 to S999 will be reset. Generally, we will use ANRP instruction.

2) Program example



When M8049 is ON, when any one of s900~s999 is ON, then M8048 is ON, and Y0 output the alarm signal.

If S910, S911 and S912 all are ON, then when X5 turns from OFF to ON, S910 will be reset, when X5 turns from OFF to ON for the next time, S911 will be reset and the like.

SQR instruction

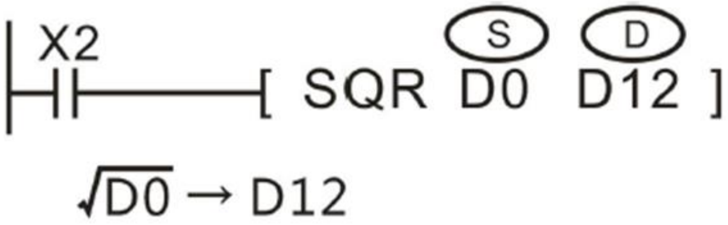
1) Instruction description

Name	Function	Bits	Pulse type	Instruction format	Step
SQR	Performs a mathematical square root	16	No	SQR S D	5
SQRP		16	Yes		5
DSQR		32	No		9
DSQRP		32	Yes		9

This instruction performs a square root operation on source data (S) and stores the result at destination device (D). The operation is conducted entirely in whole integers rendering the square root answer rounded to the lowest whole number. For example, if (S) = 154, then (D) is calculated as being 12. M8020 is set ON when the square root operation result is equal to zero. Answers with rounded values will activate M8021.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



If D0=K100, then when X2 is ON, D12=K10;

If D0=K110, then when X2 is ON, D12=K10, decimal is ignored.

FLT instruction

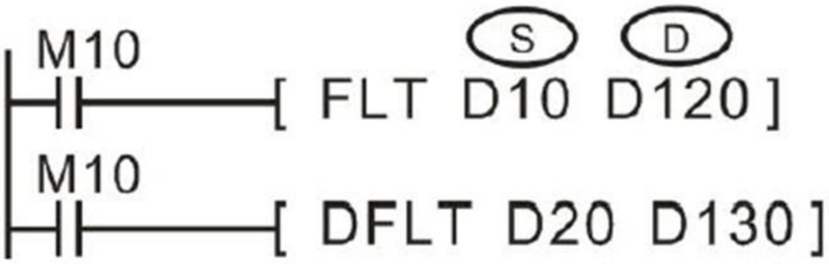
1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
FLT	Used to convert data to floating point format	16	No	FLT S D	5
FLTP		16	Yes		5
DFLT		32	No		9
DFLTP		32	Yes		9

The instruction coverts the decimal data S to floating digits, and saves the result in D and D+1. Please note that two consecutive devices (D and D+1) will be used to store the converted float number. This is true regardless of the size of the source data (S), i.e. whether (S) is a single device (16 bits) or a double device (32 bits) has no effect on the number of destination devices (D) used to store the floating point number. (The instruction INT: Convert floating point value to decimal value)

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



When M8=ON, D10 (16bit BIN) will be converted to binary floating number and saved in D120 and D121.

When M10=ON, D20 (32bit BIN) will be converted to binary floating number and saved in D130 and D131.

SWAP instruction

1) Instruction description

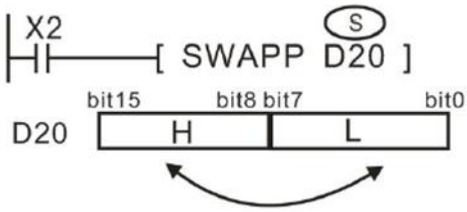
Name	Function	Bit	Pulse type	Instruction format	Step
SWAP	The high and low byte of the designated devices are exchanged	16	No	SWAP S	3
SWAPP		16	Yes		3
DSWAP		32	No		5
DSWAPP		32	Yes		5

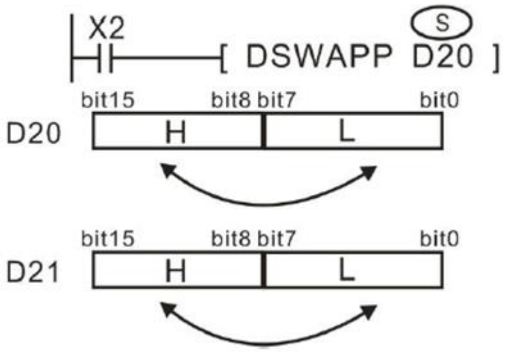
In single word (16 bit) operation the upper and lower byte of the source device are exchanged.

In double word (32 bit) operation the upper and lower byte of each or the two 16 bit devices are exchanged.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√	√	√

2) Program example





In the left demo, the upper and lower byte of D20 is exchanged.
In the right demo, the upper and lower byte of D20 and D21 are exchanged.

5.2.4 Real time clock

TCMP instruction

1) Instruction description

Name	Function	Bits	Pulse type	Instruction format	Step
TCMP	Compares two times - results of <, = and > are given	16	No	TCMP S_1 S_2 S_3 S D	9
TCMPP		16	Yes		9

S_1, S_2 and S_3 represent hours, minutes and seconds respectively. This time is compared to the time value in the 3 data devices specified by the head address S. The result is indicated in the 3 bit devices specified by the head address D.

The bit devices in D indicate the following:

- D+0 is set ON, when the time in S is less than the time in S_1, S_2 and S_3.
- D +1 is set ON, when the time in S is equal to the time in S_1, S_2 and S_3.
- D +2 is set ON, when the time in S is greater than the time in S_1, S_2 and S_3.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
S_3					√	√		√	√	√	√	√	√	√	√	√
S												√	√	√		
D		√	√	√												

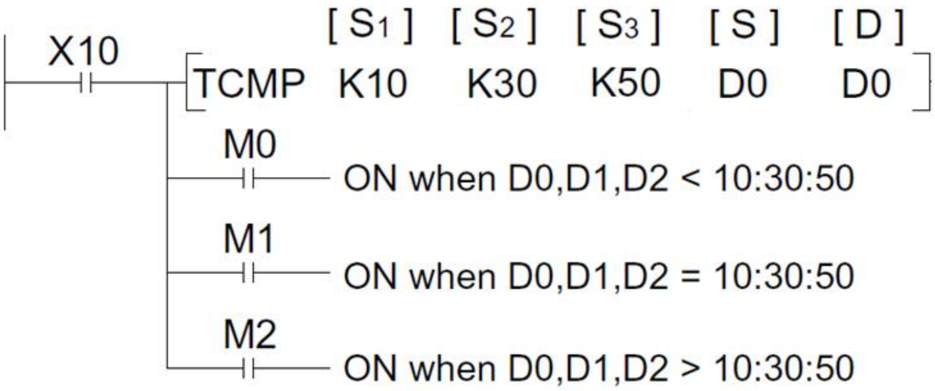
2) Program example

When X10 is ON, M0 or M1 or M2 will be ON.

When X10 turns off, TCMP is not executed; M0, M1 and M2 keep the initial value.

User could use RST or ZRST to reset M0~M2.

User could parallel or cascade M0~M2 to achieve >=, <= or ≠.


X10
[S1]
[S2]
[S3]
[S]
[D]
TCMP
K10
K30
K50
D0
D0
M0
ON when D0,D1,D2 < 10:30:50
M1
ON when D0,D1,D2 = 10:30:50
M2
ON when D0,D1,D2 > 10:30:50

TZCP instruction

1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
TZCP	Compares a time to a specified time range	16	No	TZCP S_1 S_2 S D	9
TZCPP		16	Yes		9

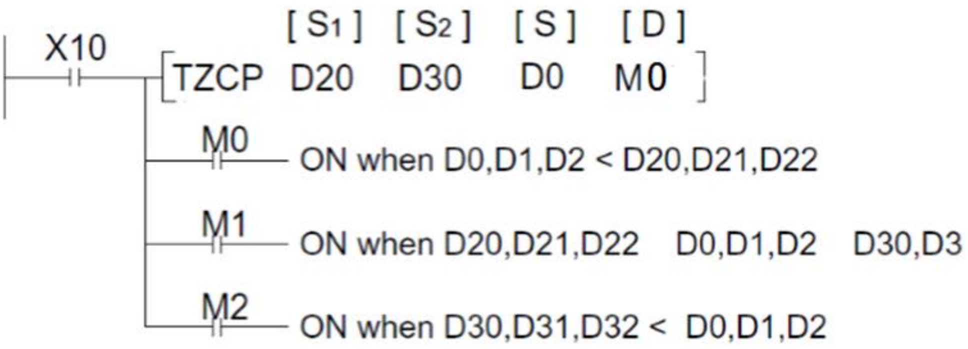
S_1, S_2 and S represent time values. Each specifying the head address of 3 data devices. S is compared to the time period defined by S_1 and S_2. The result is indicated in the 3 bit devices specified by the head address D.

The bit devices in D indicate the following:

- D +0 is set ON, when the time in S is less than the times in S_1 and S_2.
- D +1 is set ON, when the time in S is between the times in S_1 and S_2.
- D +2 is set ON, when the time in S is greater than the times in S_1 and S_2.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1												√	√	√		
S_2												√	√	√		
S												√	√	√		
D		√	√	√												

2) Program example


X10
[S1] [S2] [S] [D]
TZCP D20 D30 D0 M0
M0
ON when D0,D1,D2 < D20,D21,D22
M1
ON when D20,D21,D22 D0,D1,D2 D30,D3
M2
ON when D30,D31,D32 < D0,D1,D2

When X10=ON, m0 or m1 or m2 will be ON.

When M12 turns from ON to OFF, TZCP is not executed. M0~M2 keep the initial value. User could use RST or ZRST to reset M0~M2.

TADD instruction

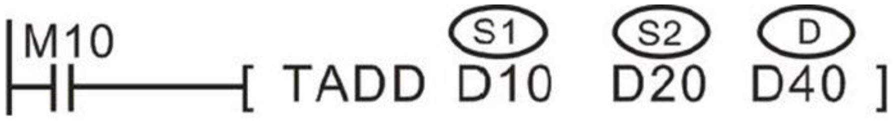
1) Instruction description

Name	Function	Bits	Pulse type	Instruction format	Step
TADD	Adds two time values together to give a new time	16	No	TADD S_1 S_2 D	7
TADDP		16	Yes		7

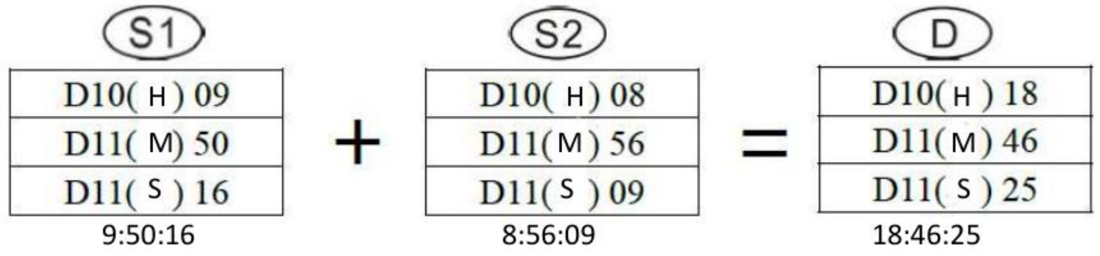
Each of S_1, S_2 and D specify the head address of 3 data devices to be used a time value. The time value in S_1 is added to the time value in S_2, the result is stored to D as a new time value. D occupies 3 continuous addresses (i.e. hour, minute and second). If the time is more than 24 hours, the carry flag M8022 is set ON.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1												√	√	√		
S_2												√	√	√		
D												√	√	√		

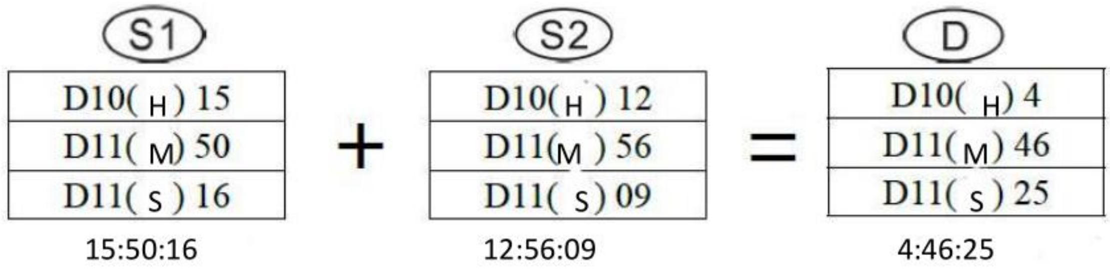
2) Program example



Process:



If the time is more than 24 hours, the carry flag M8022 is set ON.



TSUB instruction

1) Instruction description

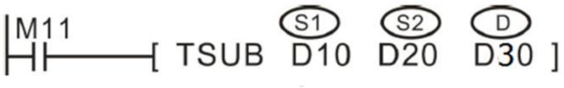
Name	Function	Bits	Pulse type	Instruction format	Step
TSUB	Subtracts one time value from another to give a new time	16	No	TSUB S_1 S_2 D	7
TSUBP		16	Yes		7

Each of S_1, S_2 and D specify the head address of 3 data devices (hour, minute, second) to be used a time value. The time value in S_1 is subtracted from the time value in S_2, the result is stored to D as a new time value.

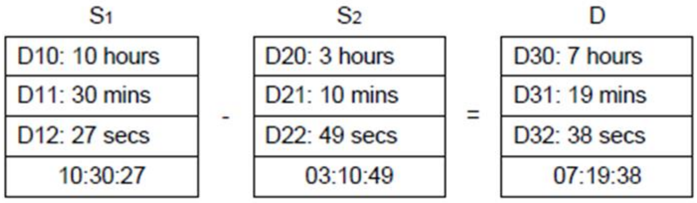
If the result is minus number, M8021 will be set ON; if the result is 00:00:00, M8020 will be set ON.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1												√	√	√		
S_2												√	√	√		
D												√	√	√		

2) Program example

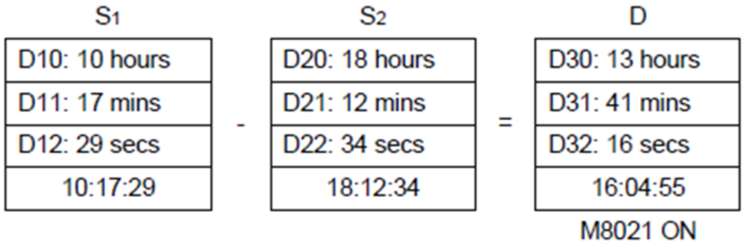


Process:



S1		S2		D
D10: 10 hours		D20: 3 hours		D30: 7 hours
D11: 30 mins	-	D21: 10 mins	=	D31: 19 mins
D12: 27 secs		D22: 49 secs		D32: 38 secs
10:30:27		03:10:49		07:19:38

The result is smaller than zero.



S1		S2		D
D10: 10 hours		D20: 18 hours		D30: 13 hours
D11: 17 mins	-	D21: 12 mins	=	D31: 41 mins
D12: 29 secs		D22: 34 secs		D32: 16 secs
10:17:29		18:12:34		16:04:55
				M8021 ON

TRD instruction

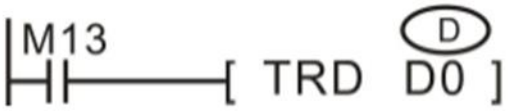
1) Instruction description

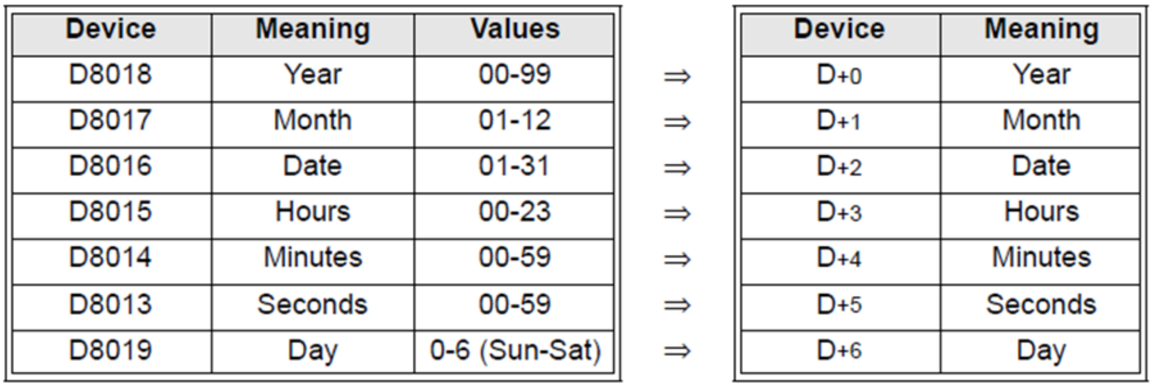
Name	Function	Bits	Pulse type	Instruction format	Step
TRD	Reads the current real time clock value to registers	16	No	TRD D	3
TRDP		16	Yes		3

The current time and date of the real time clock are read and stored in the 7 data devices specified by the head address D.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D												√	√	√		

2) Program example





Device	Meaning	Values		Device	Meaning
D8018	Year	00-99	⇒	D+0	Year
D8017	Month	01-12	⇒	D+1	Month
D8016	Date	01-31	⇒	D+2	Date
D8015	Hours	00-23	⇒	D+3	Hours
D8014	Minutes	00-59	⇒	D+4	Minutes
D8013	Seconds	00-59	⇒	D+5	Seconds
D8019	Day	0-6 (Sun-Sat)	⇒	D+6	Day

Generally, it is suggested to read the time from D8013~D8019 to other D device, rather than use D8013~D8019 directly.

TWR instruction

1) Instruction description

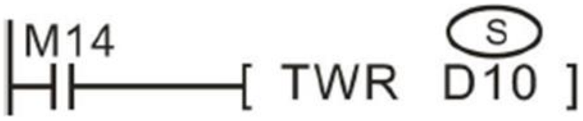
Name	Function	Bits	Pulse type	Instruction format	Step
TWR	Sets the real time clock to the value stored in registers	16	No	TWR S	3
TWRP		16	Yes		3

The 7 data devices specified with the head address S are used to set a new current value of the real time clock.

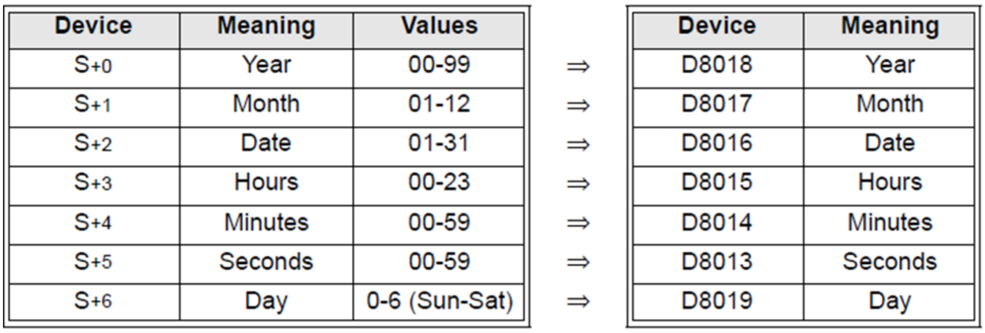
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S												√	√	√		

2) Program example

Program 1:



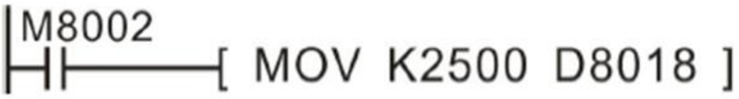
The seven devices

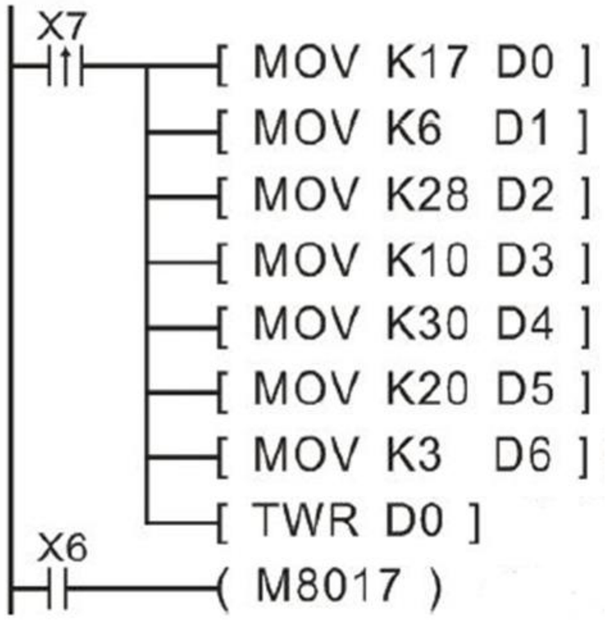


Device	Meaning	Values		Device	Meaning
S+0	Year	00-99	⇒	D8018	Year
S+1	Month	01-12	⇒	D8017	Month
S+2	Date	01-31	⇒	D8016	Date
S+3	Hours	00-23	⇒	D8015	Hours
S+4	Minutes	00-59	⇒	D8014	Minutes
S+5	Seconds	00-59	⇒	D8013	Seconds
S+6	Day	0-6 (Sun-Sat)	⇒	D8019	Day

In the usual case it shows only 2 digits for years (for example: in 2009 only show 09), If user hopes that "year" shows four digits format, the following program is needed:

Program 2:





```
X7
-|↑|---[ MOV K17 D0 ]
      -[ MOV K6  D1 ]
      -[ MOV K28 D2 ]
      -[ MOV K10 D3 ]
      -[ MOV K30 D4 ]
      -[ MOV K20 D5 ]
      -[ MOV K3  D6 ]
      -[ TWR D0 ]
X6
-| |---( M8017 )
```

D0~D6 correspond to year~second, when X7 is ON, it will write the time to real to real time clock.

Generally, it is suggested to use TWR instruction to change the time not the MOV instruction.

HOUR instruction

1) Instruction description

Name	Function	Bits	Pulse type	Instruction format	Step
HOUR	Hour meter	16	No	HOUR S D_1 D_2	7
DHOUR		16	Yes		13

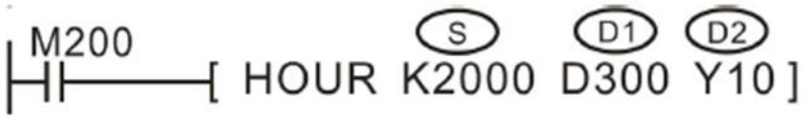
When use HOUR, D_1's range is k0~k32767, the unit is hour. [D_1]+1 is the current value, the time is specified in seconds. D_1 occupies 2 addresses.

When use DHOUR, the range(D_1 and [D_1]+1) is K0~k2147483647, the unit is hour. [D_1]+2 is the current value, it ranges from k0 to k3599, the unit is second. D_1 occupies 3 addresses.

- D_1 cannot be a minus value. In order to continuously use the current value data, even after a power OFF and ON, specify a data register which is backed up against power interruption.
- S: The value of setting time;
- D1: Current value in hours;
- D2: Alarm output destination, turns on when D1 exceeds S;

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D_1														√		
D_2		√	√	√												

2) Program example



When M200 is ON, D300 will record the duration, if the duration is less than 1 hour, it will be recorded in d301. When the value of D300 exceeds 2000, Y10 is ON. When Y10 is on, the value of D300 will continuously increase until d300 is 32767(hour) and d301 is 3599(second). If user wants to record the time from the beginning, user should reset d300 and d301 at first.

5.2.5 Arithmetic and logical operations

ADD instruction

1) Instruction description

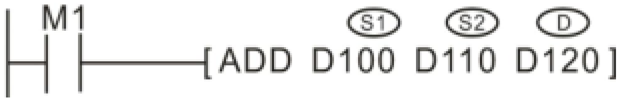
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ADD	BIN addition	16	No	ADD S_1 S_2 D	7
ADDP		16	Yes		7
DADD		32	No		13
DADDP		32	Yes		13

The data contained within the source devices (S_1, S_2) is combined and the total is stored at the specified destination device (D).

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

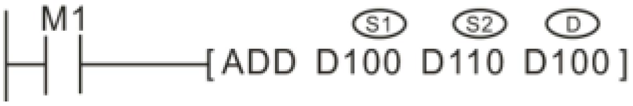
2) Program example

Example 1



When M1 is triggered, D100 combines D110 and the total is stored in D120. If D100=K8, D110=K-12, then D120=K8+K-12=K-4.

Example 2



When M1 is triggered, D100 combines D110 and the total is stored in D100, this operation will be repeated until M1 is released. This is a cumulative operation.

3) Note for use

- If the result of a calculation is “0" then a special auxiliary flag M8020 is set ON.
- If the result of an operation exceeds 32,767 (16 bit limit) or 2,147,483,647 (32 bit limit) a special auxiliary flag M8021 is set ON
- If the result of an operation exceeds -32,768 or -2,147,483,648 a special auxiliary flag M8022 is set ON.
- When using 32bit calculation, the instruction variable address is a low 16bit address, and the adjoining address is a high 16bit address. It should be prevented from repeating or overwriting in the program.

SUB instruction

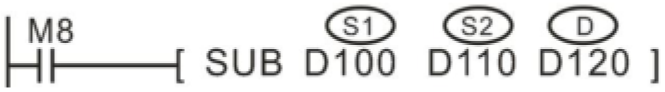
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
SUB	BIN subtraction	16	No	ADD S_1 S_2 D	7
SUBP		16	Yes		7
DSUB		32	No		13
DSUBP		32	Yes		13

The data contained within the source devices, S_2 is subtracted from the contents of S_1. The result or remainder is stored at destination device (D). The source devices are processed by the signed number, the most significant bit is the sign bit, 0 means positive and 1 means negative.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

2) Program example



When M8 is triggered, D100 subtracts D110 and the result is stored in D120. If D100=K10, D110=K8, then D120=K10-K8=K2.

3) Note for use

- If the result of a calculation is “0" then a special auxiliary flag M8020 is set ON.
- If the result of an operation exceeds 32,767 (16 bit limit) or 2,147,483,647 (32 bit limit) a special auxiliary flag M8021 is set ON
- If the result of an operation exceeds -32,768 or -2,147,483,648 a special auxiliary flag M8022 is set ON.
- When using 32bit calculation, the instruction variable address is a low 16bit address, and the adjoining address is a high 16bit address. It should be prevented from repeating or overwriting in the program.

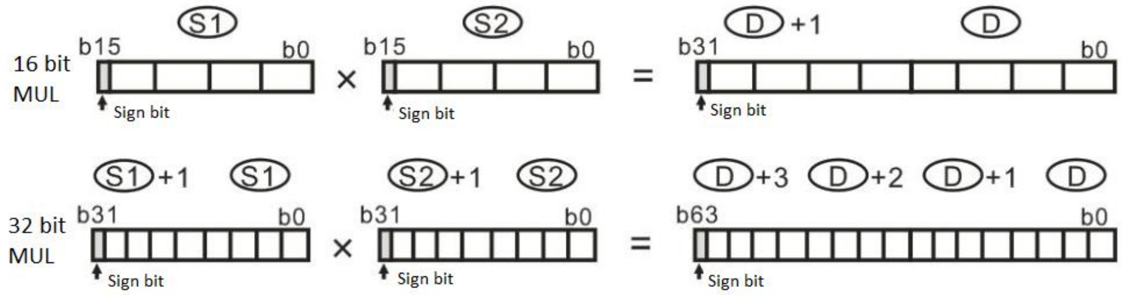
MUL instruction

1) Instruction description

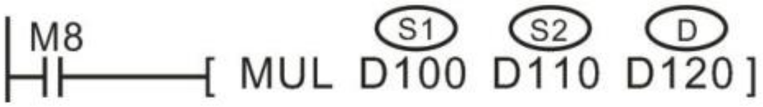
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
MUL	BIN multiplication	16	No	MUL S_1 S_2 D	7
MULP		16	Yes		7
DMUL		32	No		13
DMULP		32	Yes		13

The contents of the two source devices (S_1, S_2) are multiplied together and the result is stored at the destination device (D). The source devices are processed by the signed number, the most significant bit is the sign bit, 0 means positive and 1 means negative.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√		



2) Program example



When M8 is triggered, contents of D100 and D110 are multiplied together and the result is stored at D120. If D100=K100, D110=K25, D120=K100*K25=K2500.

3) Note for use

- V, Z devices are only available in 16bit operation;
- When using 32bit calculation, the instruction variable address is a low 16bit address, and the adjoining address is a high 16bit address. It should be prevented from repeating or overwriting in the program;
- Even when a word device is used, the operation result of the 64-bit data cannot be monitored;
- The results of the calculation can only be 32bit, for more than 32bit range of the calculation, it is best to use floating-point instructions EMUL to calculate;

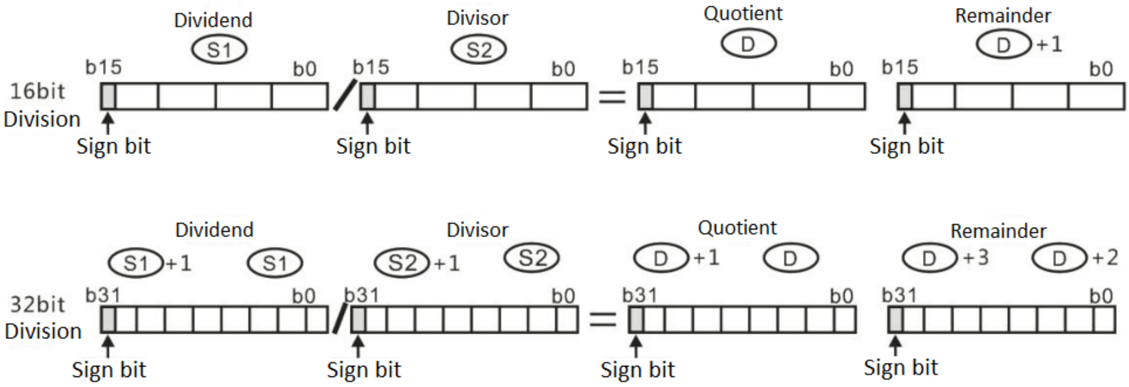
DIV instruction

1) Instruction description

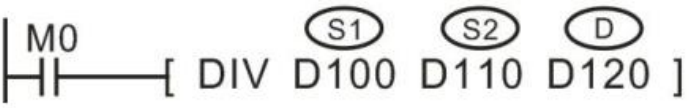
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DIV	BIN Division	16	No	DIV S_1 S_2 D	7
DIVP		16	Yes		7
DDIV		32	No		13
DDIVP		32	Yes		13

The primary source (S_1) is divided by the secondary source (S_2). The result is stored in the destination (D). Note the normal rules of algebra apply. The source devices are processed by the signed number, the most significant bit is the sign bit, 0 means positive and 1 means negative.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√		



2) Program example



When M8 is triggered, D100 is divided by D110. The result is stored in D120. If D100=200, D110=4, D120=50.

3) Note for use

- V, Z devices are only available in 16bit operation;
- When operating the DIV instruction in 32 bit mode, two 32 bit data sources are divided into each other. They produce two 32 bit results. The device identified as the destination address is the lower of the two devices used to store the quotient (D, D+1) and the following two devices are used to store the remainder (D+2, D+3);
- An error occurs when S_2 is zero;
- If (KnX/KnY/KnM/KnS) is specified as D, there is no remainder;
- If the divisor is negative, the remainder also is negative;

INC instruction

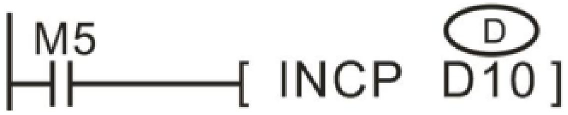
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
INC	BIN increase	16	No	INC D	3
INCP		16	Yes		3
DINC		32	No		5
DINCP		32	Yes		5

The designated device is incremented by 1 on every execution of the instruction.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D									√	√	√	√	√	√	√	√

2) Program example



M5 rising edge triggers D10 plus one

3) Note for use

- In 16 bit operation, when +32,767 is reached, the next increment will write a value of -32,768 to the destination device;
- In 32 bit operation, when +2,147,483,647 is reached the next increment will write a value of -2,147,483,648 to the destination device;
- This instruction does not refresh the 0 flag, carry, and borrower flag;

DEC instruction

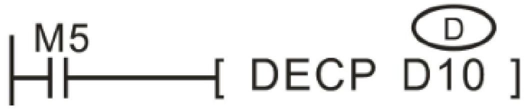
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DEC	BIN decrease	16	No	INC D	3
DECP		16	Yes		3
DDEC		32	No		5
DDECP		32	Yes		5

The designated device is decremented by 1 on every execution of the instruction.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D									√	√	√	√	√	√	√	√

2) Program example



M5 rising edge triggers D10 decrease one

3) Note for use

- In 16 bit operation, when -32,768 is reached the next decrement will write a value of +32,767 to the destination device.
- In 32 bit operation, when -2,147,483,648 is reached the next decrement will write a value of +2,147,483,647 to the destination device. This instruction does not refresh the 0 flag, carry, and borrower flag;
- This instruction does not refresh the 0 flag, carry, and borrower flag;

WAND instruction

1) Instruction description

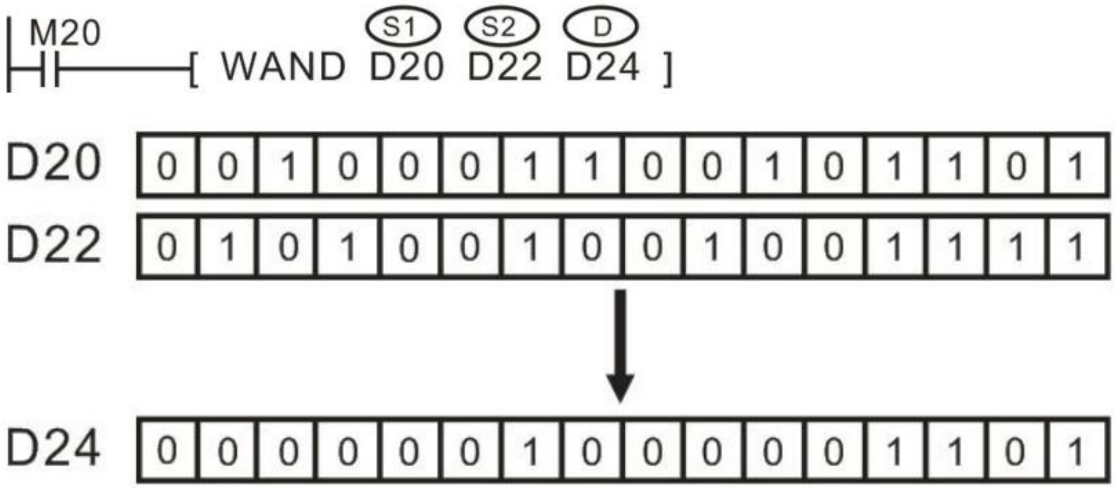
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
WAND	Logical AND	16	No	WAND S_1 S_2 D	7
WANDP		16	Yes		7
DAND		32	No		13
DANDP		32	Yes		13

The bit patterns of the two source devices are analyzed (the contents of S_2 is compared against the contents of S_1). The result of the logical AND analysis is stored in the destination device (D).

1∧1=1 1∧0=0 0∧1=0 0∧0=0

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

2) Program example



D20	0	0	1	0	0	0	1	1	0	0	1	0	1	1	0	1
D22	0	1	0	1	0	0	1	0	0	1	0	0	1	1	1	1
D24	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1

WOR instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
WOR	Logical OR	16	No	WOR S1 S2 D	7
WORP		16	Yes		7
DOR		32	No		13
DORP		32	Yes		13

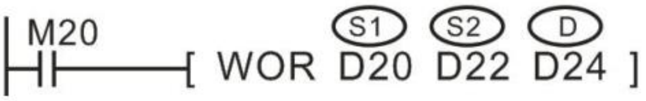
The bit patterns of the two source devices are analyzed (the contents of S_2 is compared against the contents of S_1). The result of the logical OR analysis is stored in the destination device (D). The following rules are used to determine the result of a logical OR operation. This takes place for every bit contained within the source devices:

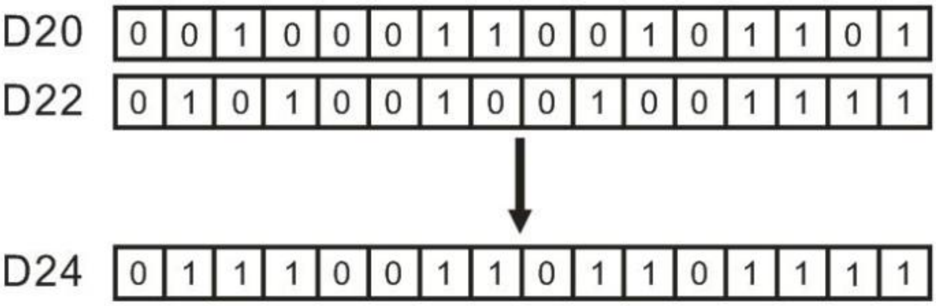
General rule: (S1) Bit n WOR (S2) Bit n = (D) Bit n

- 1 WOR 1 = 1 0 WOR 1 = 1
- 1 WOR 0 = 1 0 WOR 0 = 0

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

2) Program example





D20	0	0	1	0	0	0	1	1	0	0	1	0	1	1	0	1
D22	0	1	0	1	0	0	1	0	0	1	0	0	1	1	1	1
D24	0	1	1	1	0	0	1	1	0	1	1	0	1	1	1	1

WXOR instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
WXOR	Logical XOR	16	No	WXOR S_1 S_2 D	7
WXORP		16	Yes		7
DXOR		32	No		13
DXORP		32	Yes		13

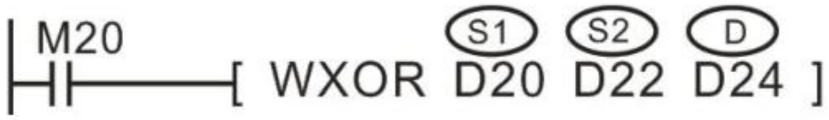
The bit patterns of the two source devices are analyzed (the contents of S_2 is compared against the contents of S_1). The result of the logical XOR analysis is stored in the destination device (D).The following rules are used to determine the result of a logical XOR operation.

General rule: (S_1) Bit n WXOR (S_2) Bit n = (D) Bit n

- 1 WXOR 1 = 0 0 WXOR 1 = 1
- 1 WXOR 0 = 1 0 WXOR 0 = 0

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

2) Program example



D20	0	0	1	0	0	0	1	1	0	0	1	0	1	1	0	1
D22	0	1	0	1	0	0	1	0	0	1	0	0	1	1	1	1
D24	0	1	1	1	0	0	0	1	0	1	1	0	0	0	1	0

NEG instruction

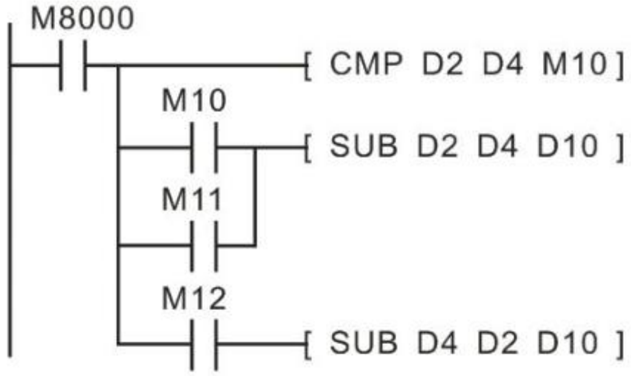
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
NEG	Complement	16	No	NEG D	7
NEGP		16	Yes		7
DNEG		32	No		13
DNEGP		32	Yes		13

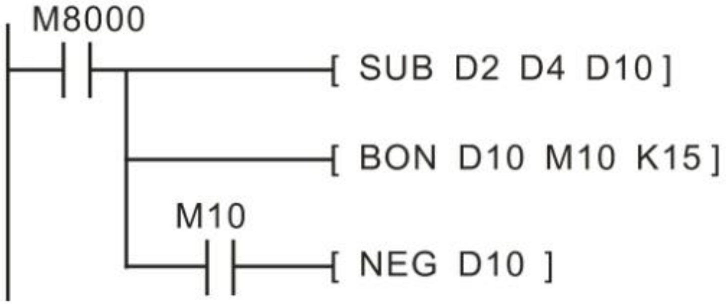
The bit pattern of the selected device is inverted. This means any occurrence of a ‘1’ becomes a ‘0’and any occurrence of a ‘0’ will be written as a ‘1’.When this is complete, a further binary 1 is added to the bit pattern. The result is the total logical sign change of the selected devices contents, e.g. a positive number will become a negative number or a negative number will become a positive.

2) Program example

The absolute value of subtraction



In above program, if D2>D4, M10 will be triggered, if D2=D4, M11 will be triggered, if D2<D4, M12 will be triggered. This program ensures that D10 is positive, also below picture can meet this requirements.



When the D10 of bit15 is "1" (denoted D10 is negative), when M10 triggered, NEG instruction gets absolute value of D10.

3) Note for use

- The positive and negative numbers are represented by the bit contents of the register's most significant bit (leftmost), “0” is for positive and “1” is for negative.
- When the most significant bit is 1, the NEG instruction can be used for obtaining the absolute value.

5.2.6 High speed process

REF instruction

1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
REF	Forces an immediate update of inputs or outputs as specified	16	No	REF D n	5
REFP		16	Yes		5

Refresh n devices immediately stating from D.

- D must be the device like X0, X10, Y0 or Y10… i.e the unit's digit need to be zero.
- The value of n must be the multiple of 8(n=8~256)

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D	√	√														
n																

Standard PLC operation processes output and input status between the END instruction of one program scan and step 0 of the following program scan. If an immediate update of the I/O device status is required the REF instruction is used.
REF can be used between the instruction FOR~ NEXT or CJ .

REF can be used in the interrupt subprogram to refresh the input information and the output result.

The delay of the input port state depends on the filter time of the input device. X0 to X7 have the digital filter function, the filter time is between 0 and 60 ms, the other IO ports are hardware filter that the filter time is 10 ms. The specific parameter you need to refer to the PLC manual.

The delay of the output port state change depends on the response time of the output element, such as relay. The output contact will not act until the response time of the relay or transistor is over.

The response latency of the relay output type plc is about 10 ms (max :20ms),the high speed output port of the transistor plc is about 10 us, for the common output port of the transistor plc is about 0.5 ms. The specific parameter you need to refer to the PLC manual.

2) Program example

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

During the operation, once X20 is ON, the state of the input port X0 to X17 will be read immediately, the input signal will be refreshed and there is no input delay.

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

During the operation, once X20 is ON, the state of the output port Y0 to Y17 will be read immediately, the output signal will be refreshed immediately rather than until the END instruction.

REFF instruction

1) Instruction description

Name	Function	Bit(bits)	Pulse type	Instruction format	Step
REFF	Inputs are refreshed, and their input filters are reset to the newly designated value	16	No	REFF n	7
REFFP		16	No		7

n is the filter time for X0 ~ X7 input port.

X0~X7 use digital filters, the default filter time is set by the D8020. D8020 can be changed to 0 ~ 60ms by REFF instruction. The remaining X ports only have hardware RC filter that the filter time is about 10ms and can't be changed.

When using the interrupts or high speed counting, the filter time of the related port reduce to minimum automatically. The unrelated ports stay as it was.

User can also use MOV instruction to change the value of D8020.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
n	Constant, n=0~60, the unit is ms															

2) Program example

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

When X10 is triggered, the filter time of X0~X7 is 5ms, when X10 is OFF, The filter time of X0~ X7 is 15 ms.

MTR instruction

1) Instruction description

Name	Function	Bit(bits)	Pulse type	Instruction format	Step
MTR	Multiplexes a bank of inputs into a number of sets of devices. Can only be used once.	16	No	MTR S D_1 D_2 n	9

This instruction is only for transistor plc. This instruction allows a selection of 8 consecutive input devices (head address S) to be used multiple (n) times, i.e. each physical input has more than one, separate and quite different (D_1) signal being processed. The result is stored in a matrix-table (head address D_2). “n” is the number of scanning column in matrix.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	√															
D1		√														
D2		√	√	√												
n	Constant, n=2~8															

2) Program example

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

The wiring:

When output Y30 is ON only those inputs in the first bank are read. These results are then stored; in this example, auxiliary coils M10 to M17. The second step involves Y30 going OFF and Y31coming ON. This time only inputs in the second bank are read. These results are stored in devices M20 to M27. The last step of this example has Y31 going OFF and Y32 coming ON. This then allows all of the inputs in the second bank to be read and stored in devices M20 to M27. The processing of this instruction example would take 20 × 2 = 40 msec.

A scanning input with a maximum of 64 points can be achieved using 8-point X output and 8-point transistor Y output. But it is not suitable for high speed input operations because it needs a time of 20 ms,8 line= 160 ms to read each input. Therefore, the ports after X20 are typically used as the scanning inputs. This instruction is allowed to be used only once in the program.

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

DHSCR instruction

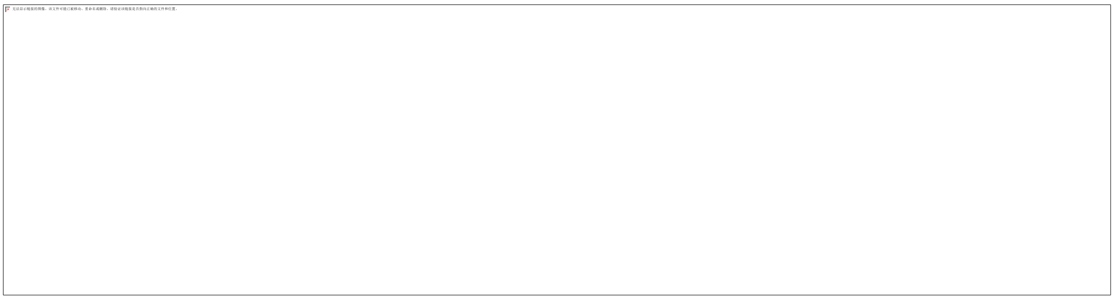
1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
DHSCR	Resets the selected output when the specified high speed counter equals the test value	32	No	DHSCR S_1 S_2 D	13

The HSCR compares the current value of the selected high speed counter (S_2) against a selected value (S_1). When the counters current value changes to a value equal to S1, the device specified as the destination (D) is reset.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2													√			
D		√	√	√												

2) Program example



无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

In the example above, Y10 would be reset only when C255's value stepped from 199 to 200 or from 201 to 200. If the current value of C255 was forced to equal 200 by test techniques, output Y10 would NOT reset.

The operation principle of the HSCR command is similar to that of the HSCS instruction, except that the HSCR output action is just opposite to the HSCS instruction, i.e., when the counter value is equal, the specified output will be reset.

DHSCS instruction

1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
DHSCS	Sets the selected output when the specified high speed counter value equals the test value	32	No	DHSCS S_1 S_2 D	13

The HSCS set, compares the current value of the selected high speed counter (S_2) against a selected value (S_1). When the counters current value changes to a value equal to S1 the device specified as the destination (D) is set ON.

It is recommended that the drive input used for the high speed counter functions; HSCS, HSCR, HSCZ is the special auxiliary RUN contact M8000.

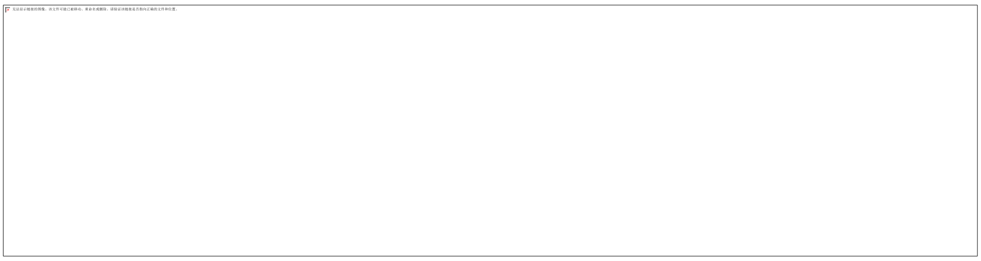
If more than one high speed counters function is used for a single counter the selected flag devices (D) should be kept within 1 group of 8 devices, i.e. Y0-7, M10-17.

All high speed counter functions use an interrupt process; hence, all destination devices (D) are updated immediately.

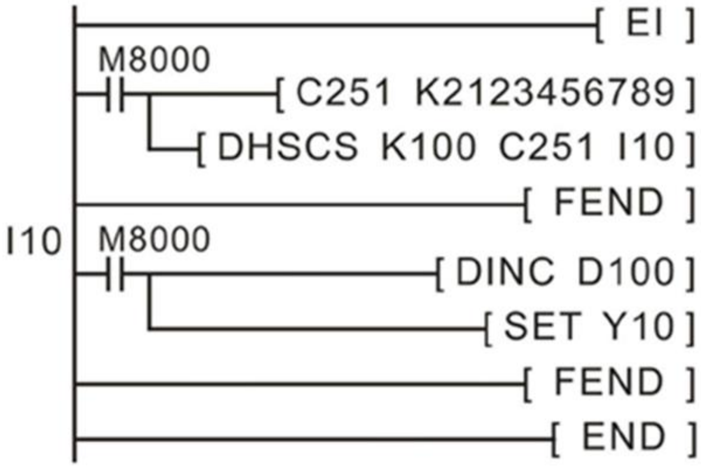
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2													√			
D		√	√	√												

2) Program example

Example 1:


无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

Example 2:



LX3V can use interrupt pointers I010 through I060 (6 points) as destination devices (D). This enables interrupt routines to be triggered directly when the value of the specified high speed counter reaches the value in the HSCS instruction. When (D) is between I010~I060, the subprogram for interrupting 0~5 in the high-speed counter needs to be initiated.

DHSZ instruction

1) Instruction description

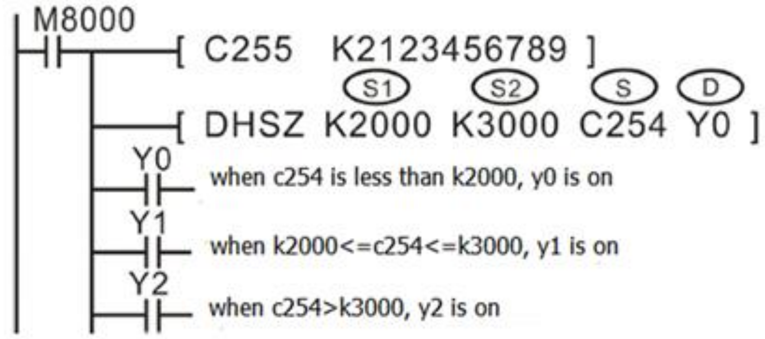
Name	Function	Bit	Pulse type	Instruction format	Step
DHSZ	The current value of a high speed counter is checked against a specified range	32	No	DHSZ S_1 S_2 S D	17

This instruction works in exactly the same way as the standard ZCP. The only difference is that the device being compared is a high speed counter (specified as S). Also, all of the outputs (D) are updated immediately due to the interrupt operation of the DHSZ. It should be remembered that when a device is specified in operand D it is in fact a head address for 3 consecutive devices. Each one is used to represent the status of the current comparison.

- S_1 is the lower limit; S_1 must be equal to or less than S2.
- S_2 is the upper limit.
- S must be C235~C255, because C235~C255 are 32bit counter, so user must use DHSZ not HSZ.
- D is for storing comparison result. When it is Y0~Y17 or M or S, there is no latency. For other output port, the output will not be executed until program END.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
S													√			
D		√	√	√												

2) Program example


M8000
C255 K2123456789
S1
S2
S
D
DHSZ K2000 K3000 C254 Y0
Y0
when c254 is less than k2000, y0 is on
Y1
when k2000<=c254<=k3000, y1 is on
Y2
when c254>k3000, y2 is on

SPD instruction

1) Instruction description

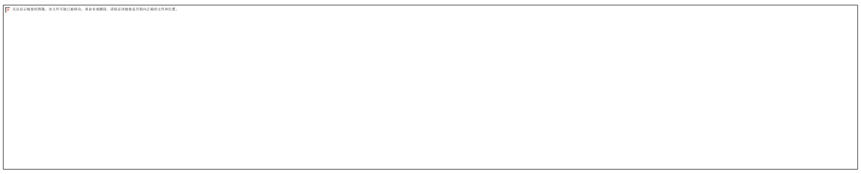
Name	Function	Bit	Pulse type	Instruction format	Step
SPD	Detects the number of 'encoder' pulses in a given time frame. Results can be used to calculate speed	16	No	SPD S_1 S_2 D	7

The number of pulses received at S_1 are counted and stored in D+1; this is the current count value. The counting takes place over a set time frame specified by S_2 in msec. The time remaining on the current 'timed count', is displayed in device D+2.

The number of counted pulses (of S_1) from the last timed count is stored in D.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1	√															
S_2					√	√		√	√	√	√	√	√	√	√	√
D												√	√	√	√	√

2) Program example



无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

X0 is the pulse input port.

D0 defines the set time frame.

Current count value, device D10

Accumulated/ last count value, device D11

Current time remaining in msec, device D12

PLSY instruction

1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
PLSY	Outputs a specified number of pulses at a set frequency	16	No	PLSY S_1 S_2 D	7
DPLSY		32	Yes		13

A specified quantity (S2) of pulses is output through device D at a specified frequency S1. This instruction is used in situations where the quantity of outputs is of primary concern.

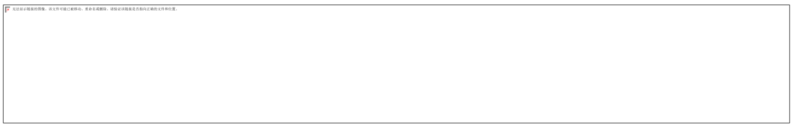
For PLSY, S1's range is 1~32767 Hz, for DPLSY, S1's range is 1~200000 Hz.

For PLSY, S2's range is 1~32767, for DPLSY, S1's range is 1~2147483647. If S2 is 0, it means there is no limitation for the output pulse quantity.

For LX3V/3VP/3VE, D could be Y0~Y3. For LX3V (1s) type, D could only be Y0 or Y1.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1								√	√	√	√	√	√	√	√	√
S_2								√	√	√	√	√	√	√	√	√
D		√														

2) Program example



无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

In the example, when X0 is OFF, the output becomes 0 too, when X0 becomes ON again it will react initially.

A single pulse is described as having a 50% duty cycle. This means it is ON for 50% of the pulse and OFF for the 50% of the pulse. The actual is controlled by interrupt handling, i.e. the output cycle is not affected by the scan time of the program.

The pulse completion flag (M8029) is set when the PLSY instruction is done.

3) The related variable in the PLSY:

- D8141 (high byte), D8140 (low byte):Y000 the count of output pulse, when the direction is reverse, Y000 decrease. (32 bits)
- D8143 (high byte), D8142 (low byte):Y001 the count of output pulse, when the direction is reverse, Y000 decrease. (32 bits)
- D8151 (high byte), D8150 (low byte):Y002 the count of output pulse, when the direction is reverse, Y000 decrease.(32 bits)
- D8153 (high byte), D8152 (low byte):Y003 the count of output pulse, when the direction is reverse, Y000 decrease.(32 bits)
- M8145: Y000 stop output pulse (immediately)
- M8146: Y001 stop output pulse (immediately)
- M8152: Y002 stop output pulse (immediately)
- M8153: Y003 stop output pulse (immediately)
- M8147: Y000 monitor in the output pulse(BUSY/READY)
- M8148: Y001 monitor in the output pulse(BUSY/READY)
- M8149: Y002 monitor in the output pulse(BUSY/READY)
- M8150: Y003 monitor in the output pulse(BUSY/READY)

PWM instruction

1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
PWM	Generates a pulse train with defined pulse characteristics	16	No	PWM S_1 S_2 D	7

Only transistor type PLC can support PWM instruction. S_1 defines the width of the pulse, S_2 defines the pulse period, and D defines the output port. For LX3V (1S firmware), the output port could be Y0 or Y1, for LX3V (2N firmware), the output port could be Y0~Y3.

The output port can’t be the same with PLSY or PLSR instruction.

S_1 <= S_2, the setting range of S_1 is 0~32767 ms. S_2 ranges from 1 to 32767ms.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D		√														

2) Program example

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

3) Thousand-ratio pattern

The thousand-ratio pattern: the thousand-ratio pattern is to divide the periodic parameters evenly equal to 1000. The user sets correspond to the control bit ON for thousand-ratio pattern as follows:

Outputs	Y0	Y1	Y2	Y3
Control bits	M8134	M8135	M8136	M8137

[Example]

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

Cycle set to 100ms, duty ratio if set to 500, then output to high level is 50ms, low level is 50ms; duty ratio if set to 100, then output to high level is 10ms, low level is 90ms; duty ratio if set to 900, then output high level is 90ms, low level is 10ms;

Calculation formula: t (ms) =T0 (ms) *K/1000

High level time (ms) = cycle time (ms) * duty ratio/1000

Low level time (ms) = cycle time (ms) – high level time (ms)

PLSR instruction

1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
PLSY	Outputs a specified number of pulses at a set frequency	16	No	PLSY S_1 S_2 S_3 D	7
DPLSY		32	Yes		17

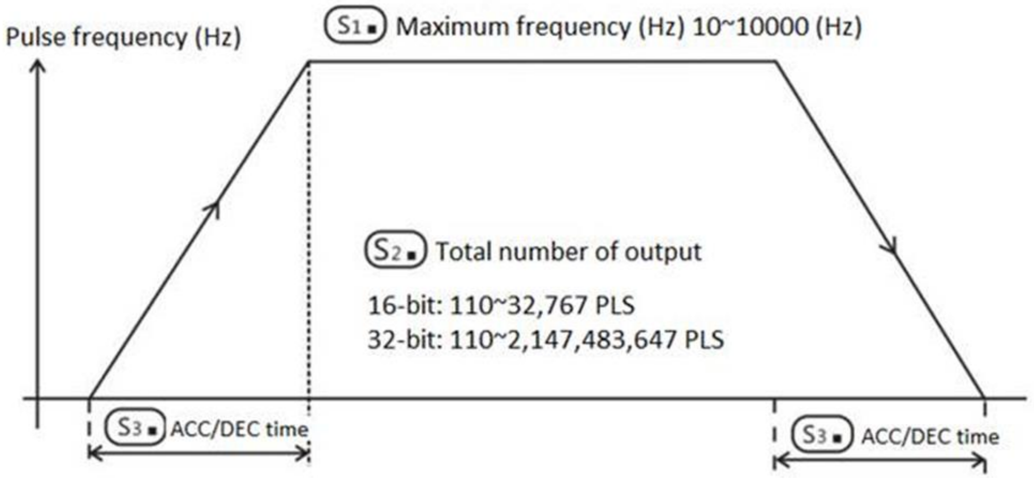
Because of the nature of the high speed output, transistor output units should be used with this instruction. Relay outputs will suffer from a greatly reduced life and will cause false outputs to occur due to the mechanical 'bounce' of the contacts.

- S_1: The maximum frequency, the range is 10~100,000Hz
- S_2: A specified quantity of output pulses, 16 bit operation: 110 to 32,767 pulses, 32 bit operation: 110 to 2,147,483,647 pulses. If it was less than 110, PLC can't output pulse;
- S_3: The acceleration time, the range is 10~32,000 (ms).
- D: output port, for LX3V/3VP/3VE, D could be Y0~Y3, for LX3V (1s) type, D could only be Y0 or Y1.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
S_3					√	√		√	√	√	√	√	√	√	√	√
D		√														

2) Program example

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。


Pulse frequency (Hz)
S1 Maximum frequency (Hz) 10~10000 (Hz)
S2 Total number of output
16-bit: 110~32,767 PLS
32-bit: 110~2,147,483,647 PLS
S3 ACC/DEC time
S3 ACC/DEC time

The special registers corresponding to each output port are listed as follow:

Register		Definition	Remarks
D8140	Low byte	Number of total pulses output to Y0 port set in the PLSY or PLSR instruction	Applicable instructions: use DMOV K0 D81xx to perform clear operation
D8140	High byte		
D8142	Low byte	Number of total pulses output to Y1 port set in the PLSY or PLSR instruction	
D8143	High byte		
D8150	Low byte	Number of total pulses output to Y2 port set in the PLSY or PLSR instruction	
D8151	High byte		
D8152	Low byte	Number of total pulses output to Y3 port set in the PLSY or PLSR instruction	
D8153	High byte		
D8136	Low byte	Accumulative value of the number of the pulses already output to Y0 and Y1	
D8137	High byte		

The output frequency range of this instruction is 10 ~ 100, 000Hz. When it is out of range, it will be automatically converted into the range and then executed. However, the actual output frequency depends on the following formula.

Output frequency=

The frequency of the initial and final stages of acceleration should not be lower than the result of the above formula.

Example: Maximum speed is 50,000, acceleration /deceleration time is 100ms.

= 500 (Hz)

When maximum frequency S1 is specified to 50000Hz, the actual output frequency at the early stage of acceleration and at the late stage of deceleration is 500Hz.

无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

3) Note for use

- The instruction is executed in an interruption way, therefore it will not be influenced by the scanning cycle;
- When the instruction power flow is OFF, the deceleration stop is active; when the power flow is changed from OFF→ON, the pulse output process starts over again.
- Special auxiliary coil M8029 is turned ON when the specified number of pulses has been completed. The pulse count and completion flag (M8029) are reset when the PLSY instruction is de-energized. If “0" (zero) is specified the PLSY instruction will continue generating pulses for as long as the instruction is energized.
- The process can't be repeated with the output port number of the PWM instruction.

PTO instruction

1) Instruction description

Name	Function	Bit	Pulse type	Instruction format	Step
PTO	Pulse envelope output instruction	16	No	PTO S_1 S_2	5
DPTO		32	No		9

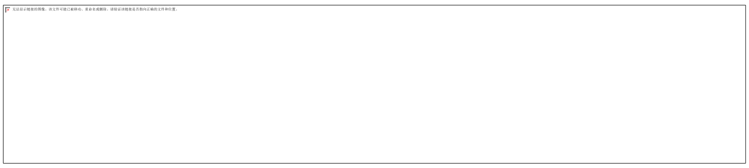
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2		√														

Take operator S1 as the starting address, then the data table is as below:

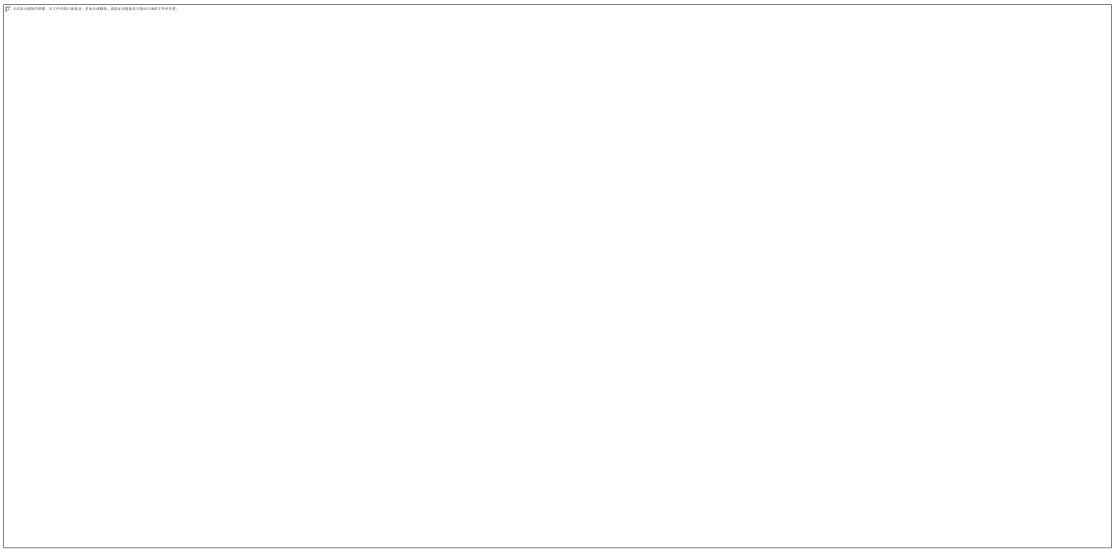
ADDRESS OFFSET	SECTION	DESCRIPTION
0		Number of segments: 1 to 255 (0 means no output)
1		Record the number currently being
2		The number of executions of the Envelope table (-1: doesn't execute 0: always execute) (Restart to take effect)
.....		Reserved
10	#1	Initial frequency (range of frequencies) (0~200,000)
11		Frequency increment (signed: -20,000~20,000)
12		Pulse number(1-4,294,967,295)
13	#2	Initial frequency(range of frequencies) (0~200,000)
14		Frequency increment (signed: -20,000~20,000)
15		Pulse number (1-4,294,967,295)
(continuous)	#3	(continuous)

When using the 32-bit instruction DPTO, the address offset is 2.

2) Program example


无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

Use the PTO to control a stepper motor to achieve a simple acceleration, constant speed and deceleration or a complex process consisting of up to 255 pulses, and every waveform is acceleration, constant speed or deceleration operation. Starting and final pulse frequency is 2KHZ, the maximum pulse frequency is 10KHZ, and it requires 4000 pulses to achieve the desired number of revolutions of the motor.


无法显示链接的图像。该文件可能已被移动、重命名或删除。请验证该链接是否指向正确的文件和位置。

The example above required to produce a output signal contained three sections:

- Acceleration (section 1);
- Constant speed (section 2);
- Deceleration (section 3);

Frequency increment of each section:

- Sec 1(acceleration) frequency increment=40
- Sec 2(constant speed) frequency increment=0
- Sec 3(deceleration) frequency increment= -20

The corresponding envelope table is as below:

Segment	Register address	Value	Description
Parameter setting	D0	3	Total segments
	D1	0	Record the

			number currently being executed
	D2	0	Number of executions of envelope table
#1	D10	2khz	Initial frequency
	D11	40	Frequency increment
	D12	200	Pulse number
#2	D13	10khz	Initial frequency
	D14	0	Frequency increment
	D15	3400	Pulse number
#3	D16	10khz	Initial frequency
	D17	-20	Frequency increment
	D18	400	Pulse number

3) Note for use:

a) Take the frequency as the standard, run the command during the operation.

b) The range of frequency:0 to 100 kHz

c) If the envelope table is beyond the effective range of the device, no pulse will be sent out.

d) Frequency increment formula:

e) Frequency increment= (final frequency - initial frequency)/ the number of pulse

f) The frequency interval of pulse (including inter-segment and segment) cannot exceed 2000Hz, otherwise it will go wrong (the wrong number is 6780) and the instruction will not be executed.

g) If the frequency interval of pulse (including inter-segment and segment) exceeds 2000Hz, then PTO will not be executed:

- Cyclic transmission mode: the last pulse of the last segment and the first pulse of the first segment are regarded as the neighboring pulse.
- Single transmission mode: the last pulse of the last segment and the first pulse of the first segment are not regarded as the neighboring pulse.

5.2.7 Rotation and shift

ROL instruction

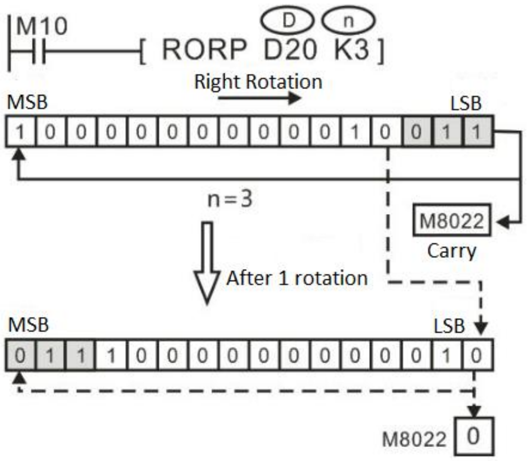
1) Instruction description

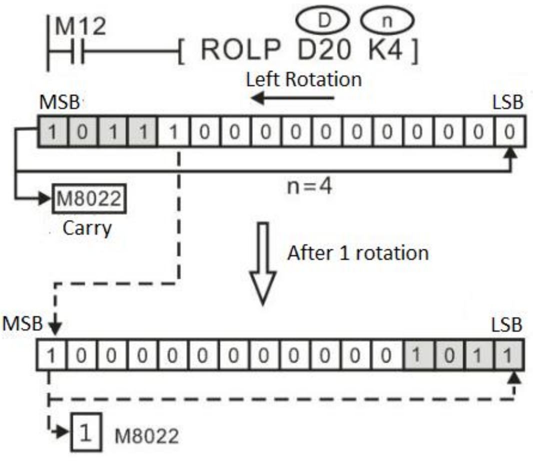
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ROL	Make 16-bit or 32-bit data shift left	16	No	ROL D n	5
ROLP		16	Yes		5
DROL		32	No		7
DROLP		32	Yes		7

The bit pattern of D is rotated n bits to the left on every execution. This instruction is generally used in pulse execution instruction. When the instruction is 32 bit, it occupies the subsequent neighboring address. When the device in D is KnY, KnM or KnS, only K4 (16-bit) and K8 (32-bit) is effective. The status of the last bit rotated is copied to carry flag M8022.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								√	√	√	√	√	√	√	√	√
n	Constant, n=1~16(16bit);n=1~32(32 bit)															

2) Program example





ROR instruction

1) Instruction description

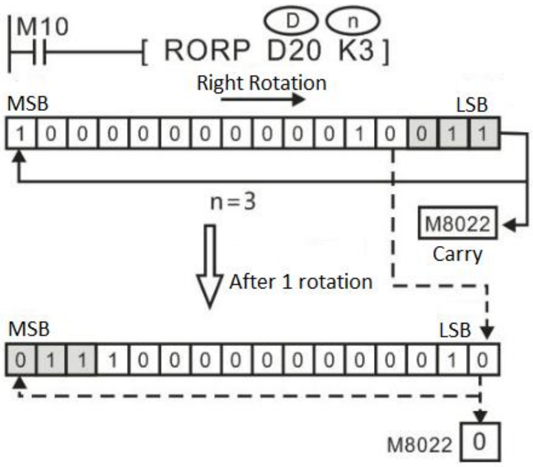
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ROR	Make 16-bit or 32-bit data shift right	16	No	ROR D n	5
RORP		16	Yes		5
DROR		32	No		7
DRORP		32	Yes		7

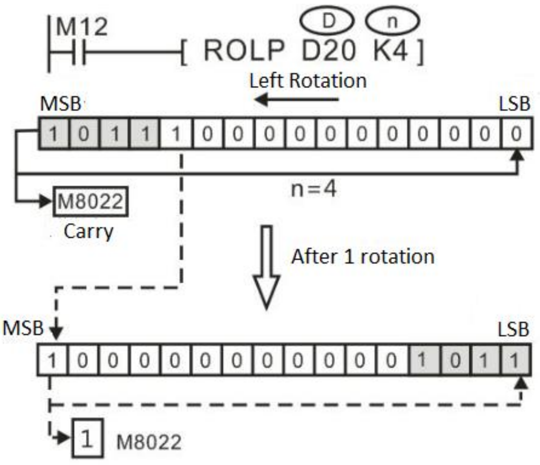
The bit pattern of D is rotated n bits to the right on every execution. This instruction is generally used in pulse execution instruction. When the instruction is 32 bit, it occupies the subsequent neighboring address.

When the device in D is KnY, KnM or KnS, only K4 (16-bit) and K8 (32-bit) is effective. The status of the last bit rotated is copied to carry flag M8022.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								√	√	√	√	√	√	√	√	√
n	Constant, n=1~16(16bit);n=1~32(32 bit)															

2) Program example





RCL instruction

1) Instruction description

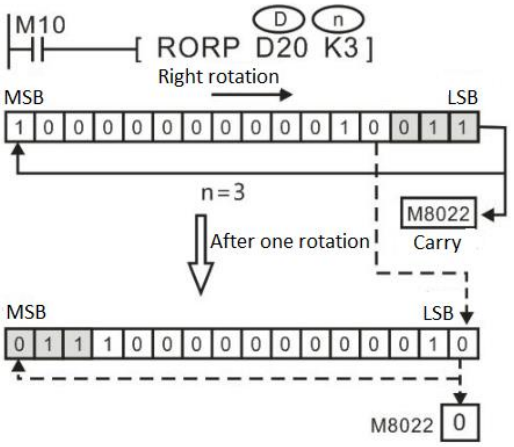
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
RCL	Make 16-bit or 32-bit data shift left with carry	16	No	RCL D n	5
RCLP		16	Yes		5
DRCL		32	No		9
DRCLP		32	Yes		9

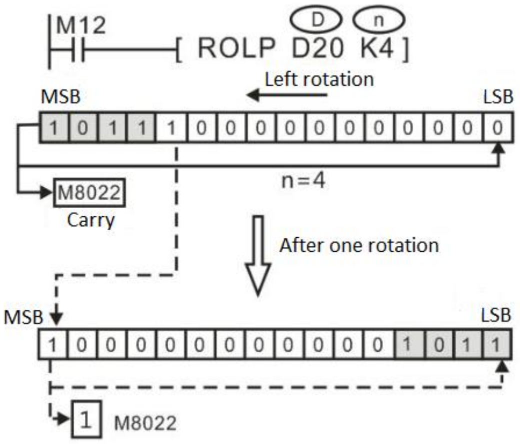
The contents of the D are rotated left n bit with the carry flag M8022. This instruction is generally used as pulse execution instruction, i.e. use the RCLP or DRCLP. When the instruction is 32bit, it takes 2 sequential addresses.

When D is KnY or KnM or KnS, only K4 (16 bit) and K8 (32 bit) are effective.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								√	√	√	√	√	√	√	√	√
n	Constant, n=1~16(16bit);n=1~32(32 bit)															

2) Program example





RCR instruction

1) Instruction description

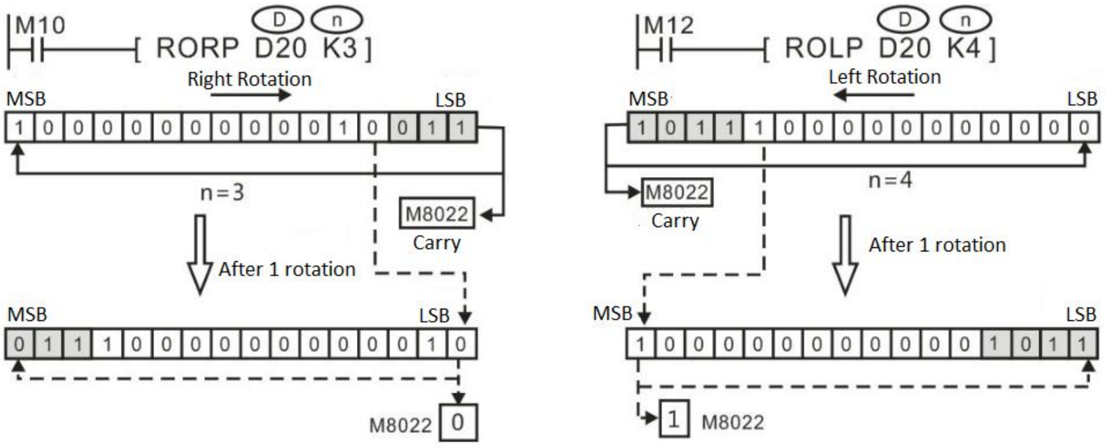
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
RCR	Make 16-bit or 32-bit data shift right with carry	16	No	RCR D n	5
RCRP		16	Yes		5
DRCR		32	No		9
DRCRP		32	Yes		9

The contents of the D are rotated right n bit with the carry flag M8022. This instruction is generally used as pulse execution instruction, i.e. use the RCLP or DRCRP. When the instruction is 32bit, it takes 2 sequential addresses.

When D is KnY or KnM or KnS, only K4 (16 bit) and K8 (32 bit) are effective.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								√	√	√	√	√	√	√	√	√
n	Constant, n=1~16(16bit);n=1~32(32 bit)															

2) Program example



SFTL instruction

1) Instruction description

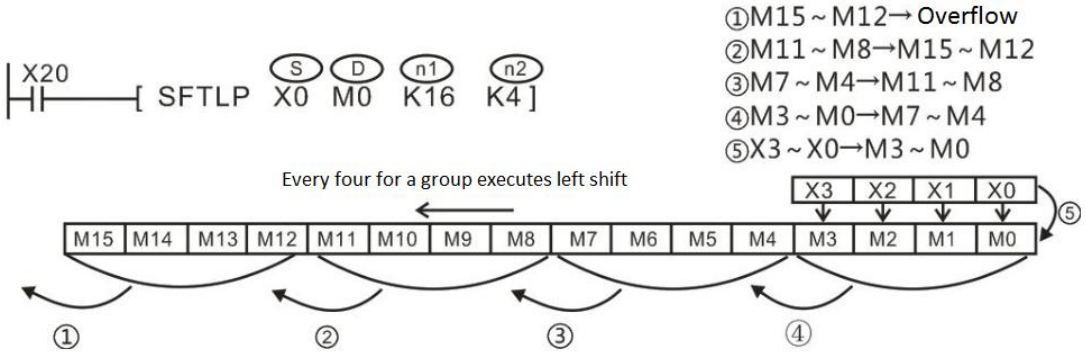
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
SFTL	Bit left shift	16	No	SFTL S D n1 n2	7
SFTLP		16	Yes		7

The instruction copies n2 source devices beginning form S to a bit stack of length n1 beginning from D. For every new addition of n2 bits， the existing data within the bit stack is shifted n1 bits to the left. Any bit data moving to a position exceeding the n1 limit is diverted to an overflow area.

This instruction is generally used as pulse instruction, i.e. SFTLP.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S		√	√	√												
D	√	√	√	√												
n1	Constant, n1≤ 1024															
n2	Constant, n2≤ n1															

2) Program example



SFTR instruction

1) Instruction description

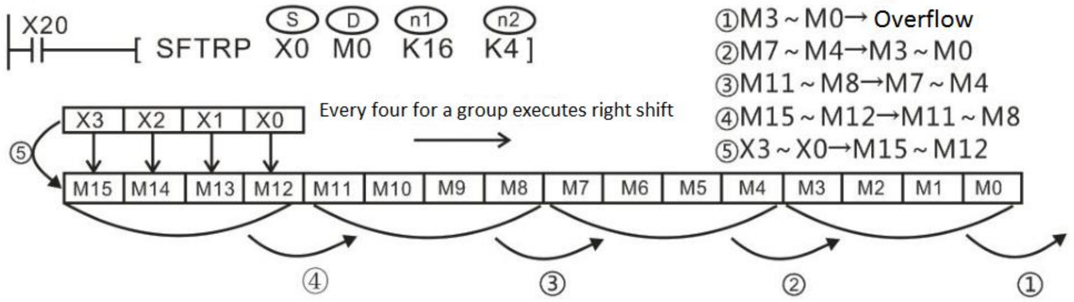
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
SFTR	Bit right shift	16	No	SFTR S D n1 n2	7
SFTRP		16	Yes		7

The instruction copies n2 source devices beginning form S to a bit stack of length n1 beginning from D. For every new addition of n2 bits, the existing data within the bit stack is shifted n1 bits to the right. Any bit data moving to a position exceeding the n1 limit is diverted to an overflow area.

This instruction is generally used as pulse instruction, i.e. SFTRP.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S		√	√	√												
D	√	√	√	√												
n1	Constant, n1≤ 1024															
n2	Constant, n2≤ n1															

2) Program example



WSFL instruction

1) Instruction description

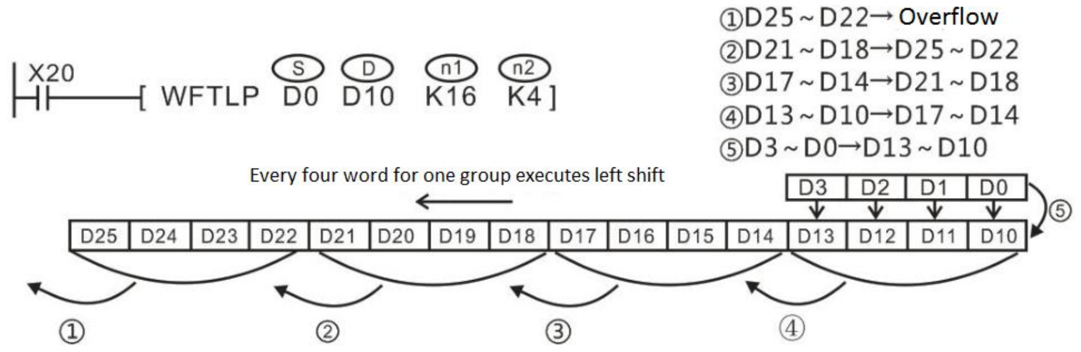
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
WSFL	Word left shift	16	No	WSFL S D n1 n2	9
WSFLP		16	Yes		9

The instruction copies n2 source devices to a word stack of length n1. For each addition of n2 words, the existing data within the word stack is shifted n2 words to the left. Any word data moving to a position exceeding the n1 limit is diverted to an overflow area.

The word shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√		
D									√	√	√	√	√	√		
n1	Constant, n1 ≤ 2048															
n2	Constant, n2 ≤ n1															

2) Program example



WSFR instruction

1) Instruction description

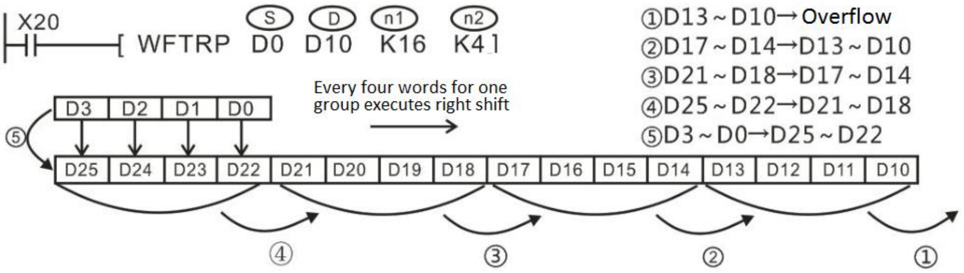
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
WSFR	Word right shift	16	No	WSFR S D n1 n2	9
WSFRP		16	Yes		9

The instruction copies n2 source devices to a word stack of length n1. For each addition of n2 words, the existing data within the word stack is shifted n2 words to the right. Any word data moving to a position exceeding the n1 limit is diverted to an overflow area.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√		
D									√	√	√	√	√	√		
n1	Constant, n1 ≤ 2048															
n2	Constant, n2 ≤ n1															

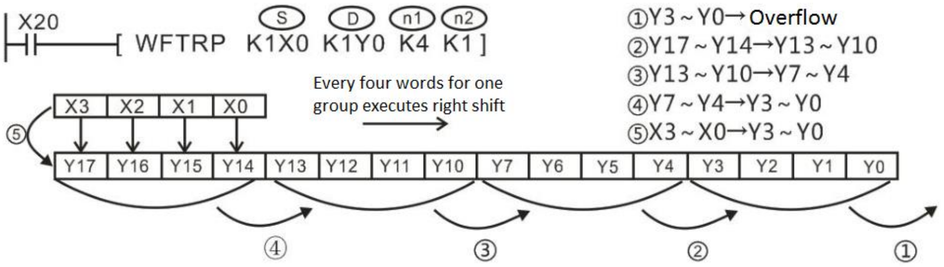
2) Program example

Example 1



Example 2

When using a Kn type device, users need to specify the same number of bits.



SFRD instruction

1) Instruction description

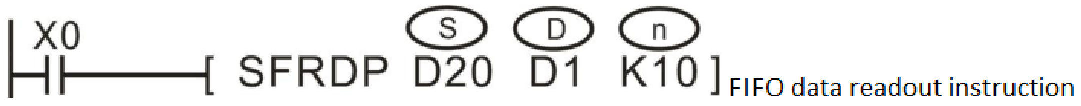
Name	Function	Bits	Pulse type	Instruction format	Step
SFRD	Shift read (the reading instruction for controlling FIFO data)	16	No	SFRD S D n	7
SFRDP		16	Yes		7

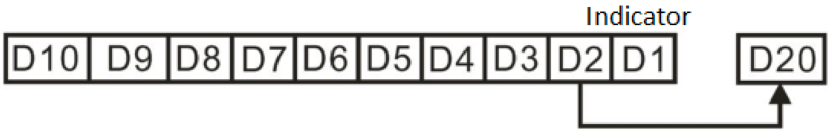
The source device(S) identifies the head address of the FIFO stack. This instruction reads the first piece of data from the FIFO stack (register S+1), moves all of the data within the stack 'up' one position to fill the read area and decrements the contents of FIFO head address(S) by 1. The read data is written to the destination device (D).When the contents of source device (S) are equal to '0'(zero), i.e. the FIFO stack is empty, the flag M8020 is turned ON.

This instruction is generally used as pulse instruction, i.e. SFRDP.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√		
D									√	√	√	√	√	√		
n	Constant, n=1~256(16bit);n=1~128(32 bit)															

2) Program example





While X0 is turned from OFF to ON, this instruction executes operations according to the following orders (D10 content remains unchanged).

a) The content in D2 is transferred to D20;

b) D10~D3 move a bit to right;

c) The Indicator (D1) minus 1;

SFWR instruction

1) Instruction description

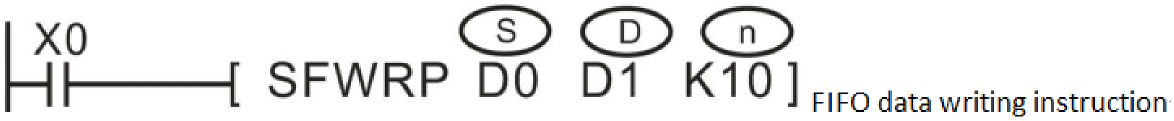
Name	Function	Bits	Pulse type	Instruction format	Step
SFWR	Shift write (the writing instruction for controlling FIFO data)	16	No	SFWR S D n	7
SFWRP		16	Yes		7

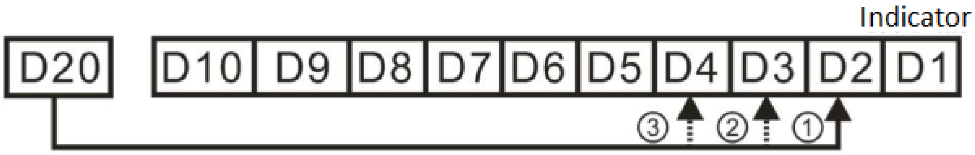
The contents of source device (S) are written to the FIFO stack. The position of insertion into stack is automatically calculated by the PLC. The destination device (D) is the head address of the FIFO stack. The contents of D identify where the next record will be stored (as an offset from D+1).

This instruction is generally used as pulse instruction, i.e. SFWRP.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√		
D									√	√	√	√	√	√		
n	Constant, 2≤n≤2048															

2) Program example





When X0 is triggered, the contents of D0 are stored in D2, and the contents of D1 become 1. While X0 is turned from OFF to ON, the contents of D0 are stored in D3, and the contents of D1 become 2, and so on. If the contents of Dl exceed n-1, the instruction is not processed and the carry flag M8022 is set to 1

5.2.8 External IO Devices

TKY instruction

1) Instruction description

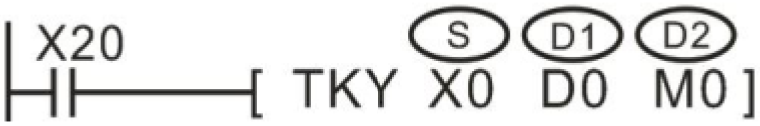
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
TKY	Ten key input	16	No	TKY S D_1 D_2	7
DTKY		32	No		13

This instruction can read from 10 consecutive devices(S+0 to S+9) and will store an entered numeric string in device D_1.

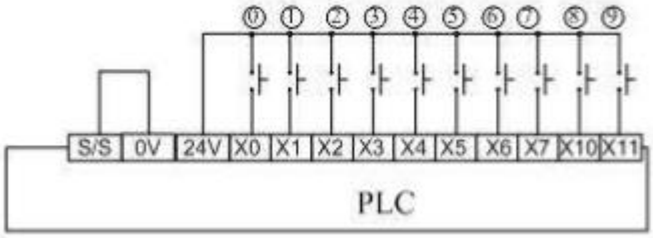
- S is the starting input port of pressing key, occupying the following ten bit units (such as X port);
- D_1 is the storage unit for inputted value;
- D_2 is the temp starting unit for state of current pressing key group, occupying the following eleven bit units;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	√	√	√	√												
D_1									√	√	√	√	√	√	√	√
D_2		√	√	√												

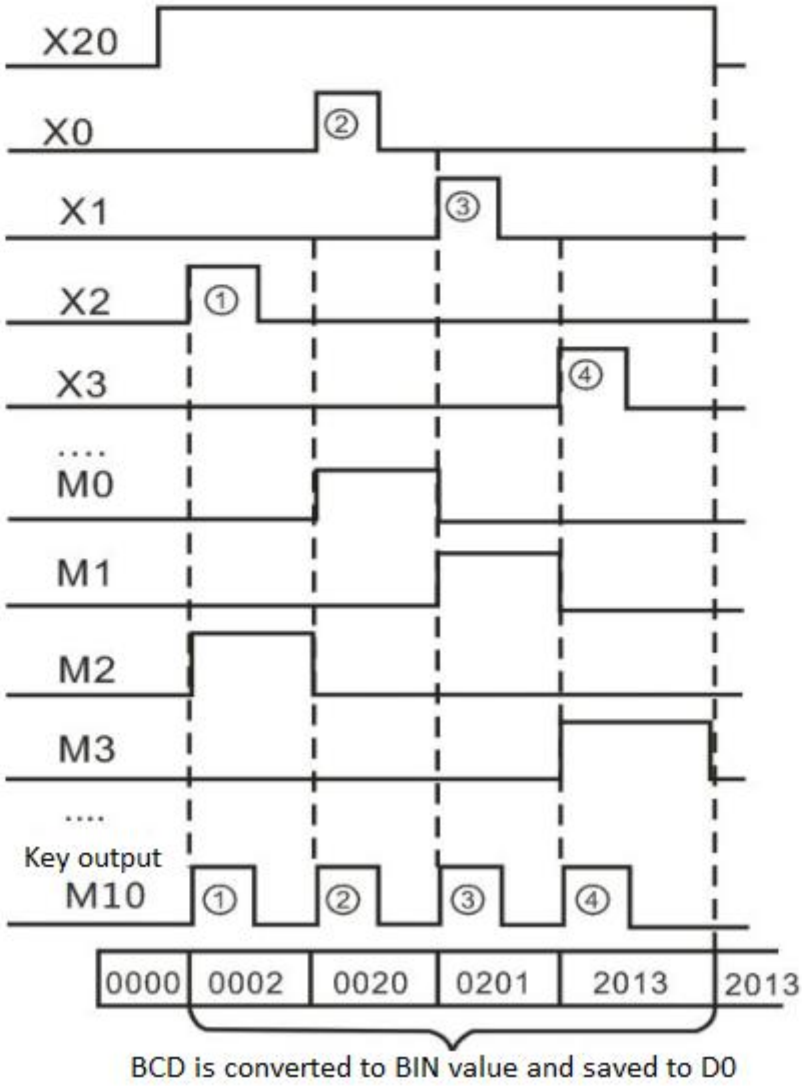
2) Program example



The corresponding hardware wiring is shown in below figure.



If user want to input "2013", just pressing key 2, 0, 1, 3 in order. The operation of PLC internal variable is shown as below figure. If using 32bit instruction (DTKY), and occupies 32bit variable. For the above case, they are D1, D0, which is higher word and lower word respectively.



Set by parameters in an instruction, X0~X11 respectively correspond to numeric keys 0~9; M0~M9 correspond to the status of keys, key output unit will be reset whenever a key is pressed.

Key values are converted to BIN and saved to the designated D_1 unit D; D_0 will never change even when the power flow turns OFF.

When several keys are pressed simultaneously, the key which is firstly detected is valid; if the number entered is more than 4 digits, the first entered number will overflow and only a 4-digit number is left.

HKY instruction

1) Instruction description

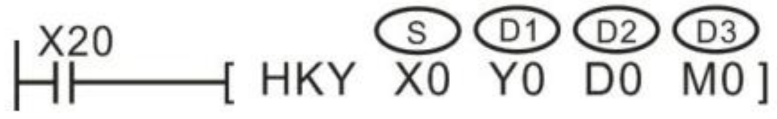
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
HKY	Hex key input	16	No	HKY S D_1 D_2 D_3	9
DHKY		32	Yes		17

This instruction creates a multiplex of 4 outputs (D_1) and 4 inputs (S) to read in 16 different devices. Which are the decimal 0~9 keys and the functional keys of A~F. When the keys are pressed (ON), decimal numbers of 4 bits between 0~9999 or functional keys between A~F can be entered, depending on the sequence of the key press actions. If 32bit instructions are used, decimal numbers of 8 bits between 0~99,999,999 or functional keys between A~F can be entered.

- S is the input port X of the keys, 4 X ports will be used;
- D_1 is the starting port button of scanning output Y port, and it uses the four Y ports.
- D_2 is the storage unit for the entered values from the keys, with a range of 0~9999. If 32bit instructions are used, decimal numbers of 8 bits between 0~99,999,999 can be entered.
- D_3 is the address which displays the entering status of the keys, which occupies a variable unit of 8 continuous bits;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	√															
D_1		√														
D_2											√	√	√	√	√	√
D_3		√	√	√												

2) Program example



- MT transistor type controller should be used due to large delay in relay output;

- When driver power flow X20 turns OFF, D0 remains the same and M0~M7 become OFF;
- It takes 8 scanning cycles to perform key scanning. After that, M8029 will be set for 1 scanning cycle;

DSW instruction

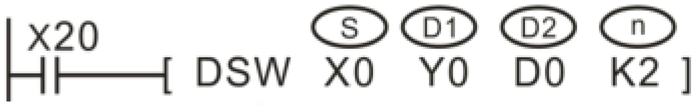
1) Instruction description

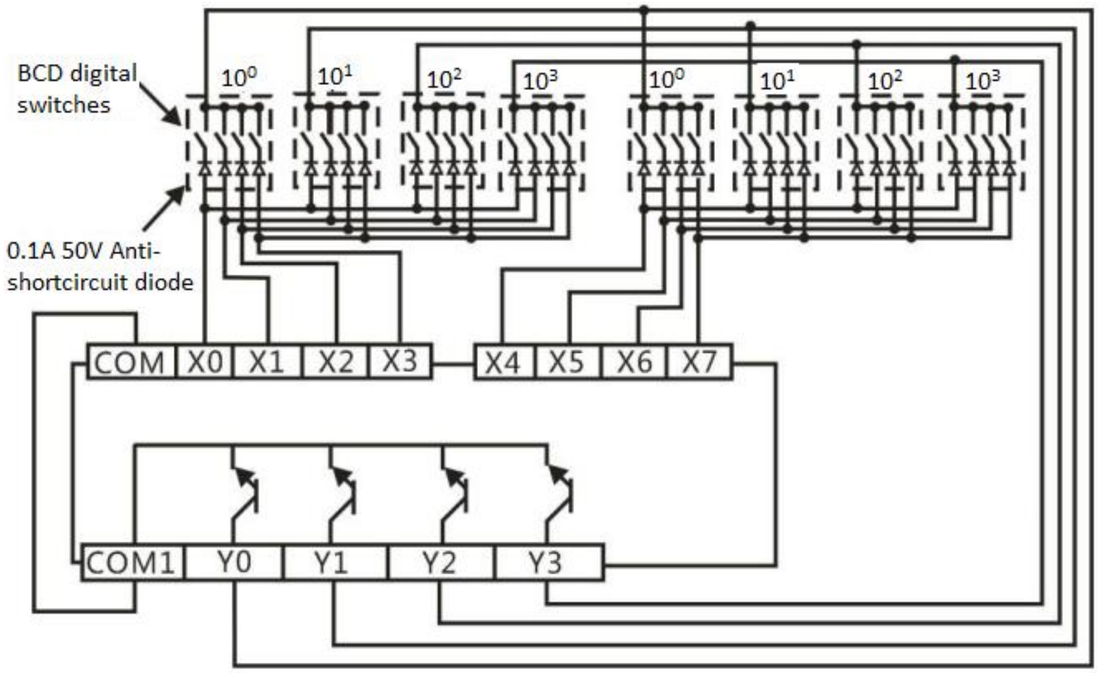
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DSW	Digital Switch	16	No	DWS S D_1 D_2 n	9

This instruction multiplexes 4 outputs (D_1) through 1 or 2(n) sets of switches. Each set of switches consists of 4 thumb wheels providing a single digit input.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	√															
D_1		√														
D_2												√	√	√	√	√
n	Constant, n=1~2															

2) Program example



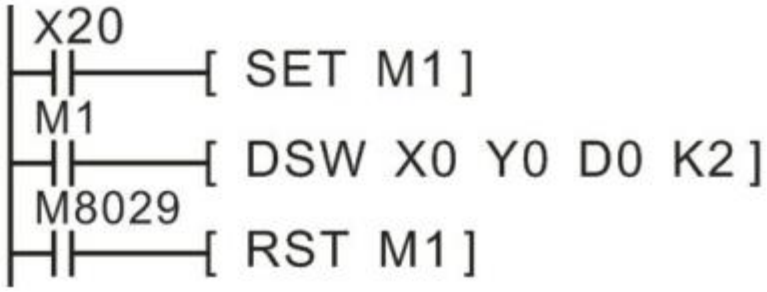


Perform the operation to scan and read the digit switches setting if X20=ON

- The setting values for the first set of digit switches are converted to BIN and saved to D0;
- The setting values for the second set of digit switches are converted to BIN and saved to D1;
- M8029 will be set for scanning cycle after one-time reading is completed;

3) Note for use

- It is recommended that transistor output units are used with this instruction.
- A digital switch set to read operation requires multiple scan cycle to complete, if the use of keystrokes to start the read operation, it is recommended to use the following program to ensure the integrity of the readable cycle:



SEGD instruction

1) Instruction description

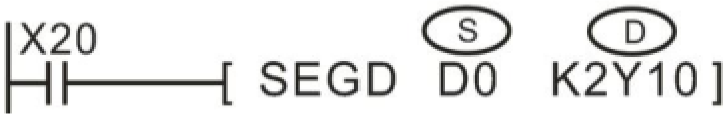
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
SEGD	Seven segment decoder	16	No	SEGD S D	5
SEGDP		16	Yes		5

A single hexadecimal digit occupying the lower 4 bits of source device S is decoded into a data format used to drive a seven segment display. A representation of the hex digit is then displayed. The decoded data is stored in the lower 8 bits of destination device D. The bit devices indicate:

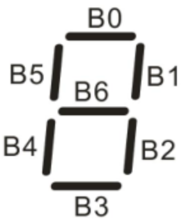
- S: The source data remaining to be decoded (b0 to b3)
- D: The variable used to store the decoded data

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

2) Program example



When X20 is triggered, a single hexadecimal digit occupying the lower 4 bits of D0 is decoded into a data format, and then output to Y10~Y17. The decoded format as below

Data		Segments	Decoded table value								Decoded Character
HEX	BIN		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000	 Each bit corresponds to a segment 1=ON 0=OFF	0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	0	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	b
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	d
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

SEGL instruction

1) Instruction description

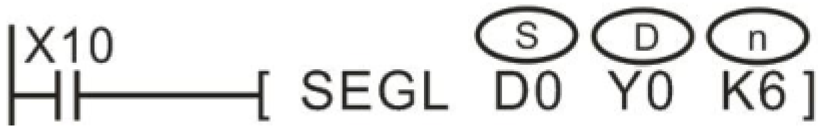
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
SEGL	Seven segment with latch	16	No	SEGL S D n	7

SEGL uses 8 or 12 Y port to drive 4 bits or 8 bits seven-segment digital tube. Tube is display by scan PLC programming manual 4.

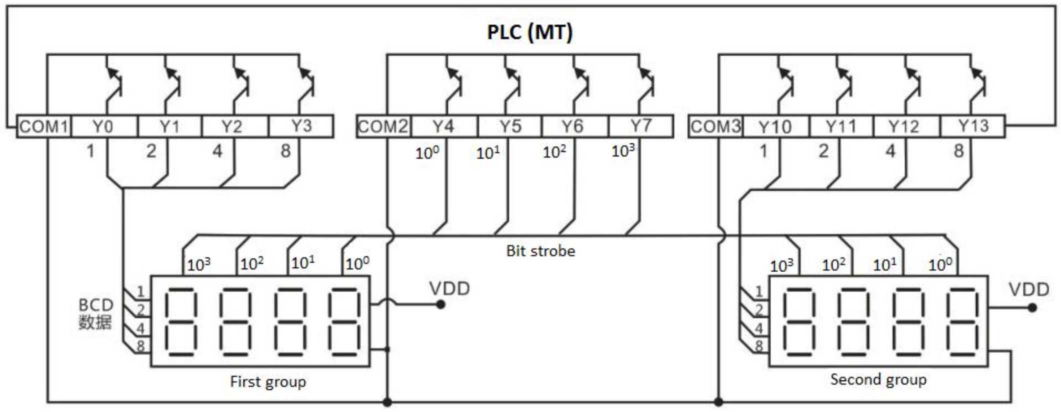
- S: The data to be displayed, it will not be displayed until the value is converted to BCD;
- D: The beginning number of the Y port that used to drive digital tube;
- n: The number of display groups, signal positive and negative logic, and other related set values;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D		√														
n	Constant, n=0~7															

2) Program example



Corresponding hardware connection is as follows. The contents of D0 are displayed in the first group of digital tube, the contents of D1 are displayed in the second group of digital tube and the procedure operation will run error when D0 or D1's numerical reading exceeds 9999:



The digital tubes in the wiring diagram come with the data show's latch, decoding and driving of 7 segment digital tube, negative logic type (the input data is considered as 1, or strobe when input port is low) 7-segment digital display tubes. In the display processing, PLC's Y4 ~ Y7 port will automatically scan cycle and only one port is ON and as a bit strobe. In this moment, the data of Y0~Y3 port is the BCD code data sent to the corresponding bits and when bit strobe signal change from the ON → OFF, it will be latched to the latch of digital tube. The digital tubes will display the number after internal decoding and driving. PLC systems will deal with Y4 ~ Y7 cycle in turn and by the same process until all the 4 bits has been processed. Similarly, Y10 ~ Y13 is the second group data output port of 4-bit digital tubes and share Y4 ~ Y7 bit strobe line, so the process is in the same and both groups' display is processed at the same time. For the example, the first group will display 2468 and the second group willdisplay9753whenD0=K2468, D1=K9753.

The SEGL instruction takes 12 program scans to complete one output cycle regardless of the number of display sets used. On completion, the execution complete flag M8029 is set.

If there is one group has 4 digits, n=0~3.If there are two groups have 4 digits, n=4~7.

Displayed number	First group				Second group			
Polarity of Y output	PNP		NPN		PNP		NPN	
Strobe and data polarity	Same	Opposite	Same	Opposite	Same	Opposite	Same	Opposite
Value of n	0	1	2	3	4	5	6	7

The SEGL instruction may be used TWICE on LX3V series PLC.

3) Note for use

- It is recommended that transistor output units are used with this instruction.
- This instruction is executed concurrently with the scan period (operation cycle) of the programmable controller. In order to perform a series of display, the scan cycle of PLC needs more than 10ms; when less than 10ms, using a constant scan mode, please make sure the scan cycle is more than 10ms to run regularly;
- The voltage of the transistor output of the programmable controller is about 1.5V;

ARWS instruction

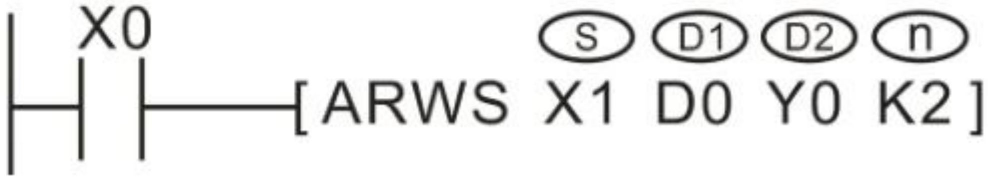
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ARWS	Arrow Switch	16	No	ARWS S D_1 D_2 n	7

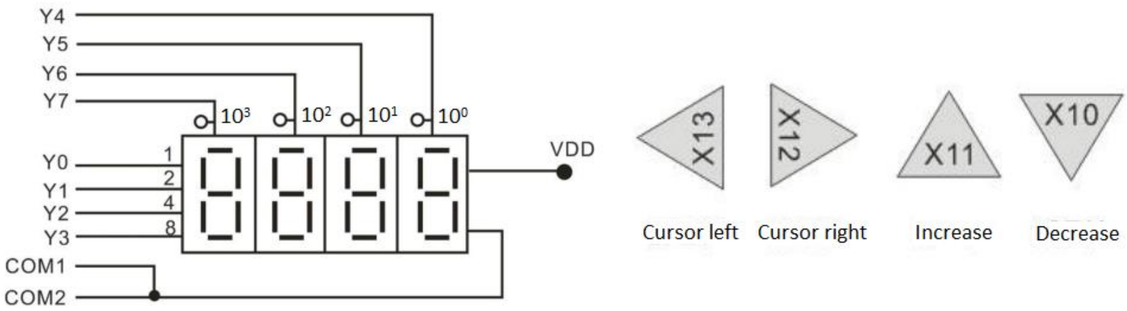
This instruction displays the contents of a single data device D_1 on a set of 4 digits, seven segment displays. The data within D_1 is actually in a standard decimal format but is automatically converted to BCD for display on the seven segment units. Each digit of the displayed number can be selected and edited. The editing procedure directly changes the value of the device specified as D_1.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	√	√	√	√												
D_1												√	√	√	√	√
D_2		√														
n	Constant, n=0~3															

2) Program example



The corresponding hardware wiring is shown in the following figure, in which PLC is the transistor output type.



Operating procedures

- The tube shows a figure value of D0. Press X10~X13 to modify the value, which should be within the 0~9999 range;

- When the X20 is ON, the cursor digit is shown as kilobits. Each time the cursor right (X12) is pressed, the specified bit switches in the order of "thousand→hundred→ten→thousand"; When pressing the cursor left (X13), the switch order reverses; and the digit cursor is indicated by the LED which is connected with the gating pulse signal (Y004 ~Y007.
- The cursor digit number switches in the order of 0 → 1→ 2→……8→9→0→1 when the increment key (X11) is pressed, when pressing the decrement key (X10), the number switches in the order of 0→9→8→7→…… 1→0→9,and the modified value becomes operative at once.

3) Note for use

If the scan time of user program, please run in constant scan time or at a fixed time interval within a timed interrupt.

ASC instruction

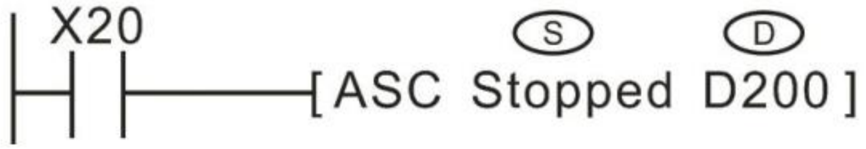
1) Instruction description

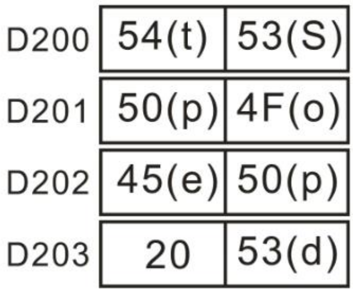
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ASC	ASCII Code	16	No	ASC S D	11

- S: The source data string , it consists of up to 8 characters taken from the printable ASCII character set;
- D: The start address to saved code. It occupies four (M8161=0) or eight(M8161=1) variable addresses;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	Only one, 8 character string may be entered at any one time.															
D												√	√	√		

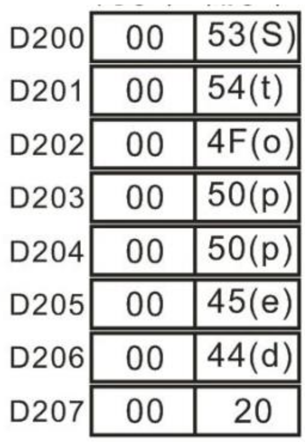
2) Program example





D200	54(t)	53(S)
D201	50(p)	4F(o)
D202	45(e)	50(p)
D203	20	53(d)

X20 is triggered, the operation for D200~D203 as below left shows.



D200	00	53(S)
D201	00	54(t)
D202	00	4F(o)
D203	00	50(p)
D204	00	50(p)
D205	00	45(e)
D206	00	44(d)
D207	00	20

If M8161=ON, then every ASCII code will occupy 16bit, the result as left shows.

3) ASCII code table

Decimal	ASCII (Hex)	Decimal	ASCII (Hex)
0	30	5	35
1	31	6	36
2	32	7	37
3	33	8	38
4	34	9	39
English letter	**ASCII (Hex)**	**English letter**	**ASCII (Hex)**
A	41	N	4E
B	42	O	4F
C	43	P	50
D	44	Q	51
E	45	R	52
F	46	S	53
G	47	T	54
H	48	U	55
I	49	V	56
J	4A	W	57
K	4B	X	58
L	4C	Y	59
M	4D	Z	5A
Code	**ASCII (Hex)**	**Code**	**ASCII (Hex)**
STX	02	ETX	03

PR instruction

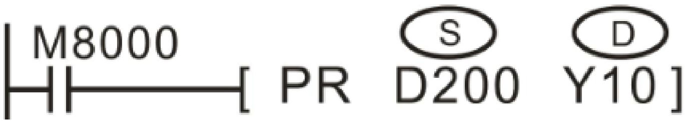
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
PR	ASCII Code print	16	No	PR S D	5

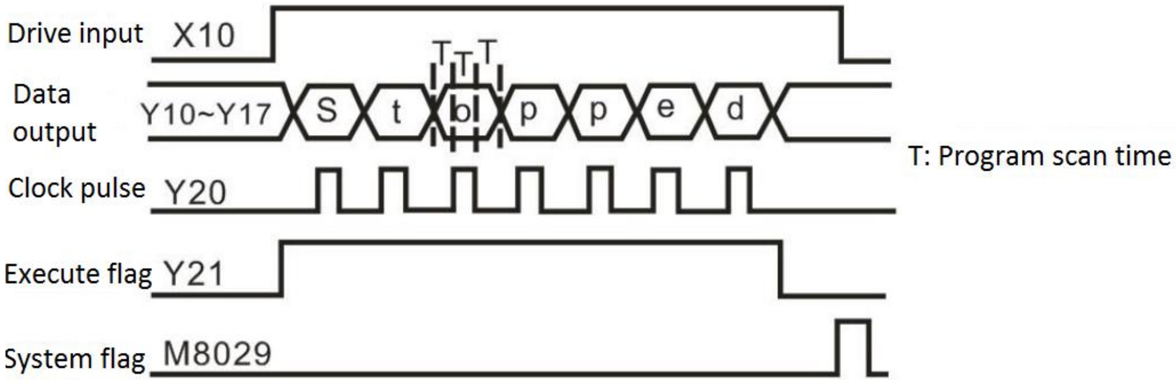
Source data (stored as ASCII values) is read byte by byte from the source data devices. Each byte is mapped directly to the first 8 consecutive destination devices D +0 to D +7). The final two destination bits provide a strobe signal (D +10, numbered in octal) and an execution/busy flag (D +11, in octal)

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S												√	√	√		
D		√														

2) Program example



If the ASCII code in D200~D203 is "Stopped", the corresponding output port signal and its timing are shown below.



3) Note for use

- This instruction should only be used on transistor output PLC’s;

- The PR instruction will not automatically repeat its operation unless the drive input has been turned OFF and ON again;
- 16 byte operation requires the special auxiliary flag M8027 to be driven ON, unless 8 byte operation will be executed;
- Once the PR instruction is activated it will operate continuously until all 16 bytes of data have been sent or the value 00H (null) has been sent. M8029 the execution complete flag is set.
- If the scan time of user program, please run in constant scan time or at a fixed time interval within a timed interrupt.

FROM instruction

1) Instruction description

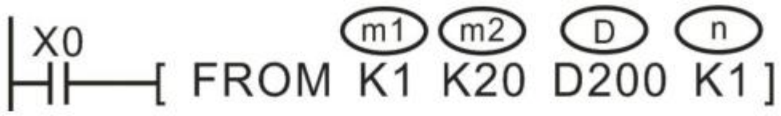
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
FROM	Read data from BFM	16	No	FROM m1 m2 D n	9
FROMP		16	Yes		9
DFROM		32	No		17
DFROMP		32	Yes		17

The FROM instruction reads data from BFM of the special function block.

- m1: The special function block with the logical block position;
- m2: The BFM memory address;
- D: The start address for stored data;
- n: Data length;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D									√	√	√	√	√	√	√	√
m1, m2= 0~32767; n=1~32767; D= K1~K4 (16bit) or K1~K8 (32bit); m1, m2, n can't support D device;																

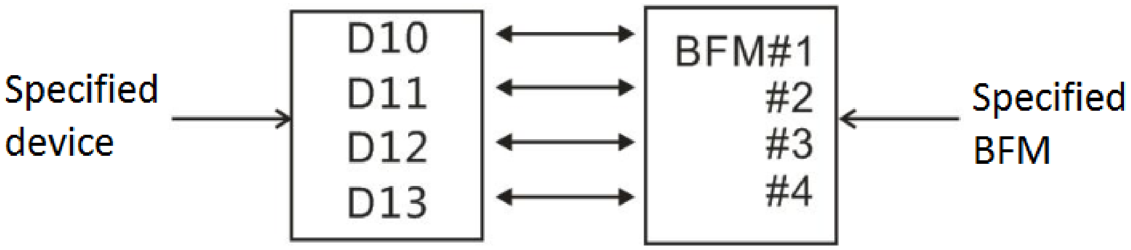
2) Program example



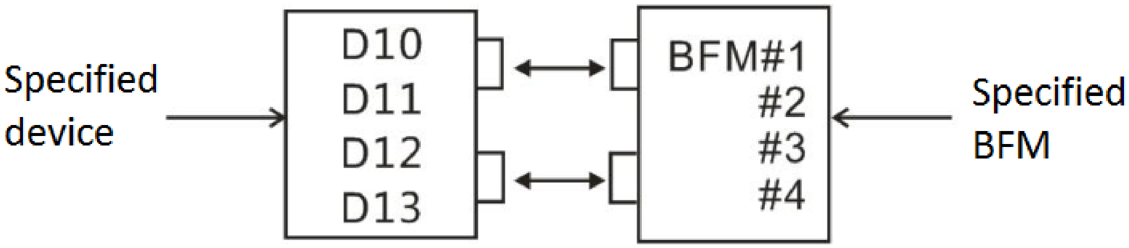
When X0 is triggered, PLC reads data from BFM20 of #1 special function block, and stores data in D200, the data length is 1;

When using instructions in 32-bit, addresses designated by D are the low 16-bit addresses; addresses designated by D+1 are the high 16-bit addresses;

n means data length, in 16 bit mode, n=2 means 2 words, but in 32bit mode, n=1 means 2 words.



16-bit mode n=4



32-bit mode n=2

TO instruction

1) Instruction description

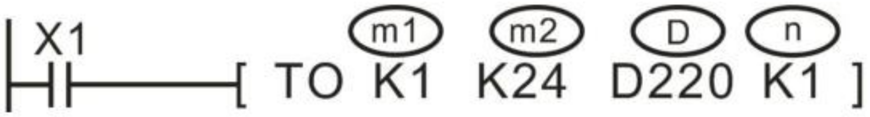
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
TO	Write date to BFM	16	No	TO m1 m2 D n	9
TOP		16	Yes		9
DTO		32	No		17
DTOP		32	Yes		17

The TO instruction writes data to BFM of the special function block.

- m1: The special function block with the logical block position;
- m2: The BFM memory address;
- D: The start address for stored data;
- n: Data length;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D									√	√	√	√	√	√	√	√
m1, m2= 0~32767; n=1~32767; D= K1~K4 (16bit) or K1~K8 (32bit); m1, m2, n can't support D device;																

2) Program example



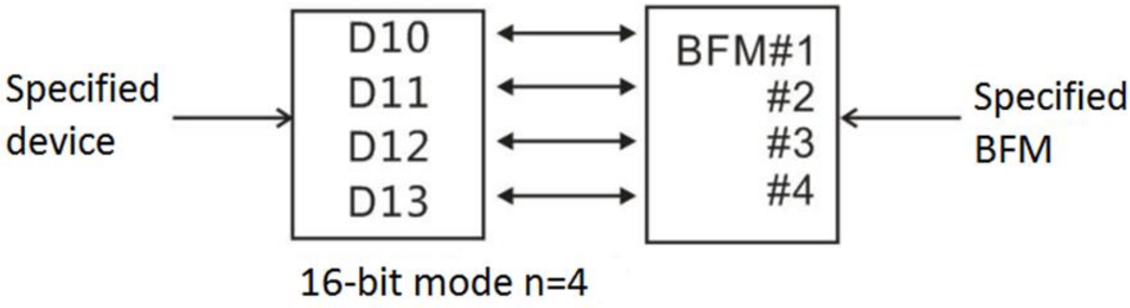
When X1 is triggered, PLC writes data from D220 to BFM24 of #1 special function block, and stores, the data length is 1;

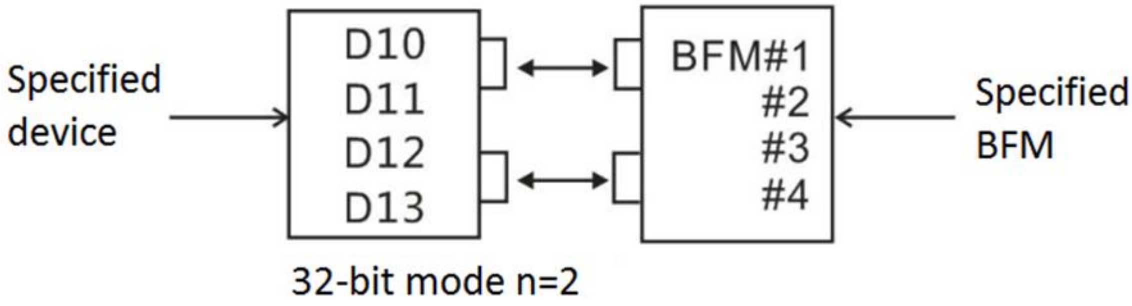
When using instructions in 32-bit, addresses designated by D are the low 16-bit addresses; addresses designated by D+1 are the high 16-bit addresses;

n means data length, in 16 bit mode, n=2 means 2 words, but in 32bit mode, n=1 means 2 words.

3) Points to note about FROM/TO instruction

Accessing the expansion module with the FROM/TO instruction is a time-consuming operation, so the scan cycle will be extended if there were many FROM/TO instructions. In order to prevent running timeout, users can add WDT instruction before FROM/TO, or stagger the execution time of the FROM/TO instruction, or using pulse type instruction.





GRY instruction

1) Instruction description

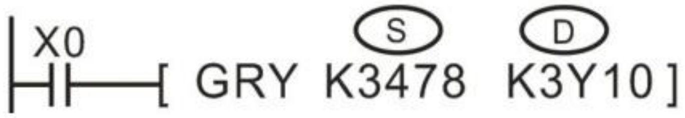
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
GRY	Calculates the gray code value of an integer	16	No	GRY S D	5
GRYP		16	Yes		5
DGRY		32	No		9
DGRYP		32	Yes		9

The binary integer value in S is converted to the GRAY CODE equivalent and stored at D.

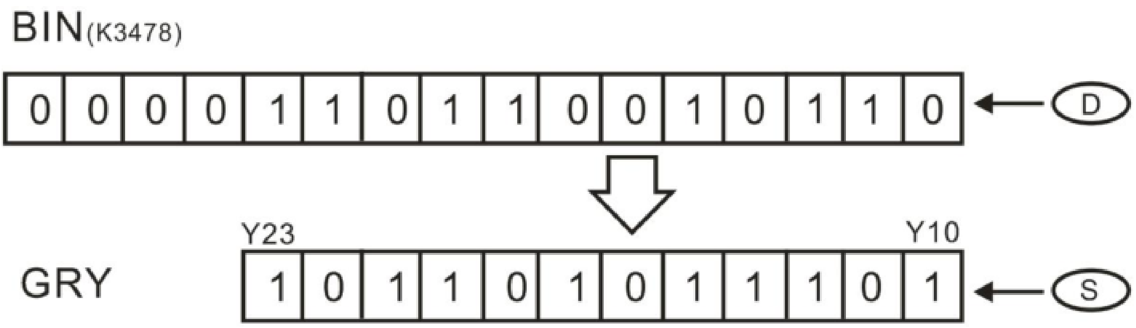
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

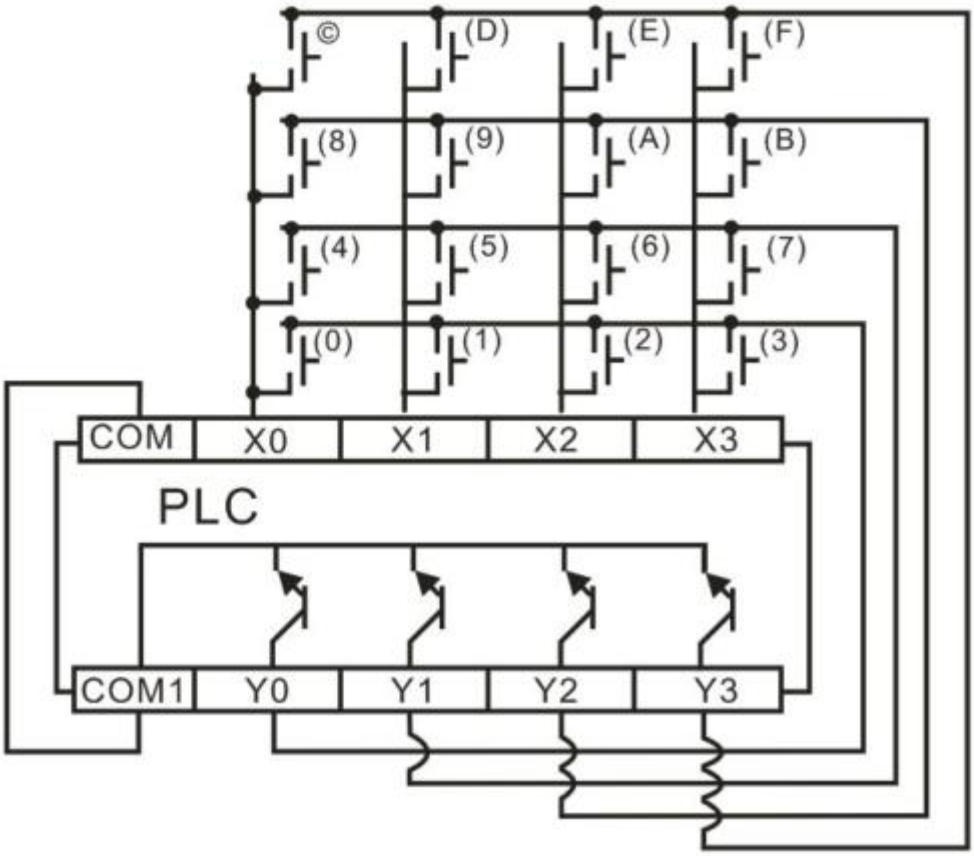
BIN → GRY Mathematical algorithm: from the right one, in turn, each bit do the XOR operation with the left bit (XOR), as the corresponding GRY bit of the value, the left one unchanged (equivalent to the left is 0);

2) Program example



[Result]:




©
(D)
(E)
(F)
(8)
(9)
(A)
(B)
(4)
(5)
(6)
(7)
(0)
(1)
(2)
(3)
COM
X0
X1
X2
X3
PLC
COM1
Y0
Y1
Y2
Y3

GBIN instruction

1) Instruction description

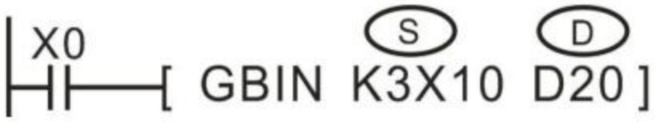
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
GBIN	Calculates the gray code value of an integer	16	No	GBIN S D	5
GBINP		16	Yes		5
DGBIN		32	No		9
DGBINP		32	Yes		9

The GRAY CODE value in S is converted to the normal binary equivalent and stored at D.

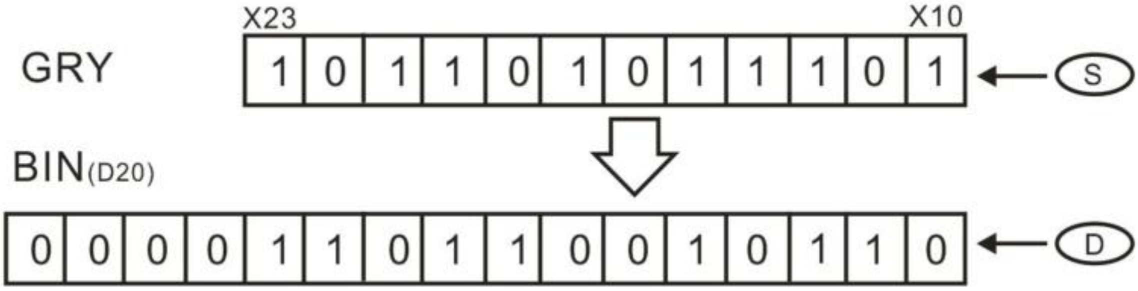
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√	√	√

GRY → BIN Mathematical algorithm: from the left of the second place, each bit with the left side of a decoded value of XOR, as the bit after decoding the value (the left one is still the same).

2) Program example



[Result]



GRY

1	0	1	1	0	1	0	1	1	1	0	1

BIN(D20)

0	0	0	0	1	1	0	1	1	0	0	1	0	1	1	0

5.2.9 ECAM instructions

DECAM instruction

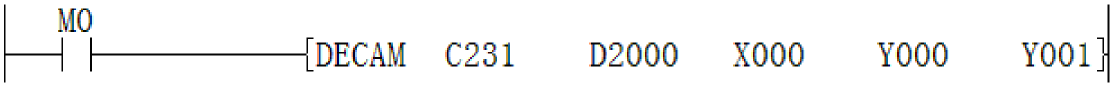
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DECAM	ECAM configuration	32	No	DECAMP S_1 S_2 S_3 D_1 D_2	21

- S_1: Master axis input, please use C register, K register;
- S_2: Parameters' address of E-cam, please use D register;
- S_3: External start signal, please use X register, M register;
- D_1: Salve axis output pulse, please use Y0~Y4;
- D_2: Slave axis output direction, please use Y register;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√								√			
S_2														√		
S_3	√		√													
D_1		√														
D_2		√														

2) Program example



Each unit of S_2 parameter value function and the setting way is shown below:

Offset	Function	Explanation	Initial	Range
0	Version number of Chart	Display the version of E-cam	5200	5200
1	Flag register	**Bit0-Flag bit:** completion of initialization. After start signal is		

		enabled, the bit will on ON after E-cam finishes the calculation of E-cam, this signal should be reset manually.		
		Bit1-Flag bit, completion of cyclic E-cam. Bit1 will be ON when cyclic electronic cam is completed. To restart the cyclic electronic cam, this signal should be reset manually.		
		Bit2-Flag bit, Pulse outputs was delayed **Bit3-Flag bit,** something is Error, E-cam stop running. **Bit4-Flag bit,** Parameter error, E-cam stop running **Bit5-Flag bit,** datasheet error, E-cam stop running **Bit6-Flag bit,** Cyclic cam **Bit7-Flag bit,** Noncyclical cam **Bit9-Flag bit,** Completion of current cycle. **Bit10-Flag bit,** In synch area **Bit11-Flag bit,** Time shaft	0	-
2	Error register	Parameter error: display error code Datasheet error: display the section number of datasheet	0	-

3	Function register (it need to set before using E-CAM)	Bit0 - delay start Bit1 – starts at specified position Bit2 - master axis zoom Bit3 - vice-axis zoom Bit5 - start signal is from outside Bit6 - Start from the current position	0	-
4	Function register (it could be changed during E-cam execution)	Bit1 - Enable sync signal Bit2 - Stop E-cam after current cycle finished	0	-
5	Start register	0- Stop E-cam immediately 1- Enable cyclic E-cam 2- Enable noncyclical E-cam Others: Stop E-cam immediately	0	-
6	Maximum frequency of E-cam (Low bit)	Maximum frequency of E-cam If frequency>200K or frequency<0, then the frequency value is 200K.	200000	0~200000
7	Electronic cam max frequency(High bit)			
8	Terminal number of synch-Y axis	Output terminal ID. Set the ID of the output terminal Y, range 0 ~ 255, when sync function starts, Yid would output sync signal.	0	0 ~ 65535
9	Low threshold value of the synchronized position of E-CAM (Low word)	Setting the Low/High threshold of the synchronized position of E-CAM When lower threshold≦ master axis position≦ higher threshold and the synch signal enable and	0	32-bit integer
10	High threshold value of the synchronized position of E-CAM (High word)			

11	Low threshold value of the synchronized position of E-CAM (Low word)	the synch-signal of Y-port set to ON.	0	32-bit integer
12	High threshold value of the synchronized position of E-CAM (High word)			
13	Number of remaining pulses sent by the master axis of E-cam	Reserved	Reserved	Reserved
14	Repeated times of noncyclical E-cam	**Reserved for cyclic E-cam.** For noncyclical E-cam: The repeated time for E-cam. When the value=0X0000, E-cam runs only once, When value=0X0001, E-cam run two Cycles. Other value is by the same way. If value=HFFFF， the noncyclical E-cam will turn into cyclic mode.	0	0 ~ 65535
15	Setting delay-pulse of E-cam (Low word)	It is only for noncyclical E-cam: (Delay pulse output could be started by set S3+bit0 to ON) When noncyclical E-cam runs, a start signal of E-cam is accepted, if E-cam does not run immediately, it needs some pulses delayed to run E-cam,, the data of this register is the number of delayed pulses. When PLC accept a start signal, master axis will run	0	32-bit unsigned integer
16	Setting delay-pulse of E-CAM (High word)			

		for the specified pulses, and then, E-cam start to run.		
17	Master axis' start position (Low word)	It only for noncyclical E-cam: If you want start the specified position start function, please use Register 3, Bit1 of this datasheet.	0	32-bit unsigned integer
18	Master axis' start position (High word)			
19	Current position of slave axis (input axis) (Converted) (Low word)	**No.1**: current position of slave axis (after conversion). **No.2**: current position of Slave axis (after zooming) during E-cam running.	0	32-bit integer
20	Current position of Slave axis (input axis) (Converted) (High word)			
21	Current position of Slave axis (input axis) (before conversion) (Low word)	**No.1**: current position of slave axis (before conversion). **No.2**: current position of Slave axis (before zooming) during E-cam running.	0	32-bit integer
22	Current position of Slave axis (input axis) (before conversion) (High word)			
23	Ratio denominator of Slave axis	Zoom magnification of Slave axis	1	1~65535
24	Ratio Numerator of Slave axis		1	1~65535
25	Current position of master axis (input axis) (Converted) (Low word)	**No.1**: current position of master axis after conversion. **No.2**: current position of Master axis (after zooming) during E-cam running	0	32-bit integer
26	Current position of master axis (input axis) (Converted) (High word)			

27	Current position of master axis (input axis) (before conversion) (Low word)	**No.1**: current position of master axis (before conversion). **No.2**: current position of master axis (before zooming) during E-cam running.	0	32-bit integer
28	Current position of master axis (input axis) (before conversion) (High word)			
29	Ratio denominator of Master axis	Zoom magnification of Master axis	1	1~65535
30	Ratio Numerator of Master axis		1	1~65535
31	Reserved	Reserved	-	-
32	Reserved	Reserved	-	-
33	Reserved	Reserved	-	-
34	Reserved	Reserved	-	-
35	Reserved	Reserved	-	-
36	Reserved	Reserved	-	-
37	Reserved	Reserved	-	-
38	Number of datasheet sections	Data sections of E-cam datasheet	0~512	
39	Starting offset of datasheet	Offset address of E-cam datasheet: default is 40	40	40
40	Section 0 of master axis(Low word)	The master axis' position of section 0	0	32-bit unsigned integer
41	Section 0 of master axis(High word)			
42	Section 0 of slave axis(Low word)	The slave axis' position of section 0	0	32-bit integer
43	Section 0 of slave axis(High word)			
44	Section 1 of master axis(Low word)	The master axis' position of section 1	0	32-bit unsigned integer
45	Section 1 of master axis(High word)			
46	Section 1 of slave axis(Low word)	The slave axis' position of section 1	0	32-bit integer
47	Section 1 of salve			

	axis(High word)			
40 + N * 2	Section N of master axis(Low word)	The master axis' position of section N	0	32-bit unsigned integer
40 + N * 2 + 1	Section N of master axis(High word)			
40 + N * 2 + 2	Section N of slave axis(Low word)	The slave axis' position of section N	0	32-bit integer
40 + N * 2 + 3	Section N of salve axis(High word)			

3) Error

- 6781: Parameter error;
- 6782: The form is beyond the range;
- 6783: The number of cam is beyond;
- Electronic cam would not work when error happens;

4) Sign

- D8141 (high byte), D8140 (low byte): The number of output pulse in Y000. It would be reduced during reversal. (32-bit)
- D8143 (high byte), D8142 (low byte): The number of output pulse in Y001. It would be reduced during reversal. (32-bit)
- D8151 (high byte), D8150 (low byte): The number of output pulse in Y002. It would be reduced during reversal. (32-bit)
- D8153 (high byte), D8152 (low byte): The number of output pulse in Y003. It would be reduced during reversal. (32-bit)
- M8145: Stop output pulse in Y000 (stop immediately)
- M8146: Stop output pulse in Y001 (stop immediately)
- M8152: Stop output pulse in Y002 (stop immediately)
- M8153: Stop output pulse in Y003 (stop immediately)
- M8147: Monitoring the output pulse in Y000 (BUSY/READY)
- M8148: Monitoring the output pulse in Y001 (BUSY/READY)
- M8149: Monitoring the output pulse in Y002 (BUSY/READY)
- M8150: Monitoring the output pulse in Y003 (BUSY/READY)

DEGEAR instruction

1) Instruction description:

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DEGEAR	Electronic gear configuration	32	No	DEGEAR S_1 S_2 S_3 D_1 D_2	21

- S1: [C and D register available] high speed pulse input.When the EGEAR value (Master shaft) read from high speed counter, the value can be changed when PLC is running. But the value could be changed if read from data resister (D) or regular counter (C);
- S2: [D register available] data saving;
- S3: [D register and constant K,H available] response time. Range: 0 ~500ms; For example: when the value is 0 or 1, it means 1ms;
- D1: Pulse output terminal: Y0 ~ Y3;
- D2: Pulse output direction: Any Y registers but different with pulse output terminal D1 above;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1													√	√		
S_2														√		
S_3					√	√								√		
D_1		√														
D_2		√														

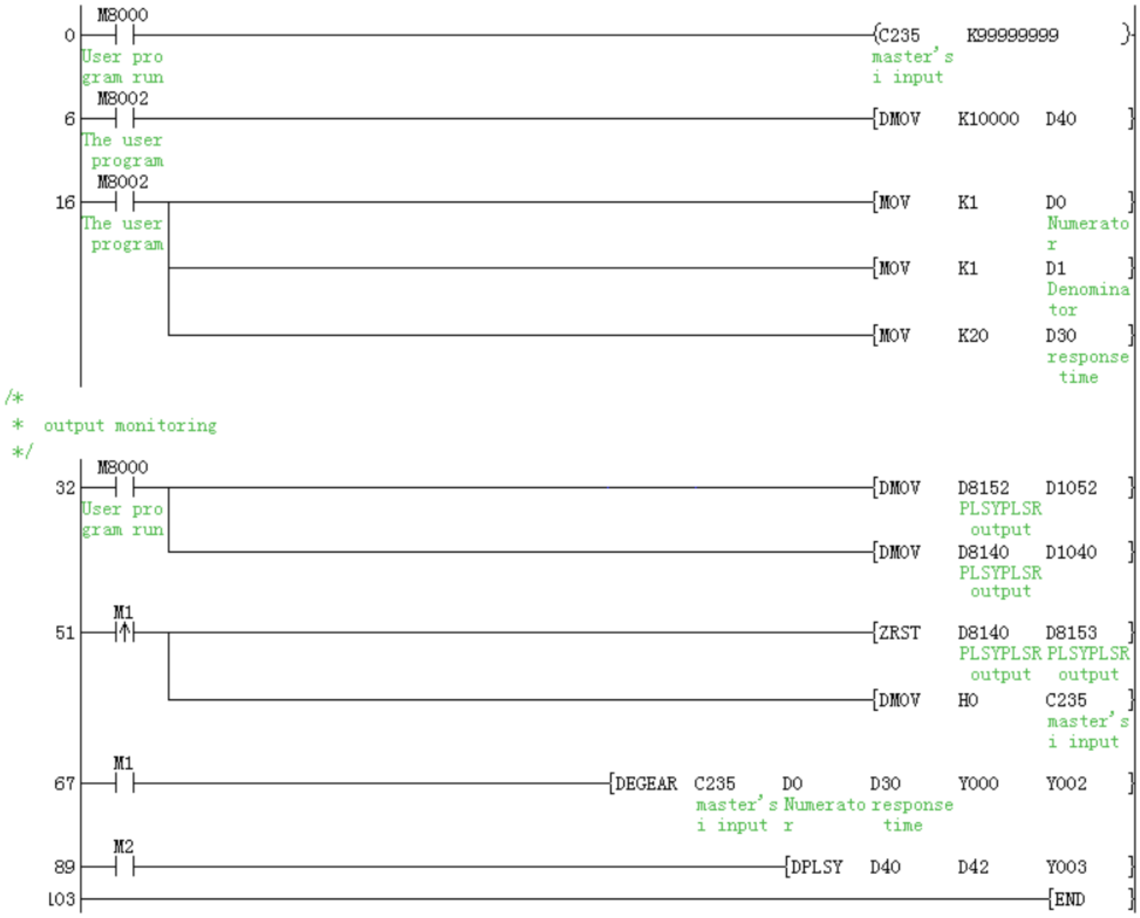
When EGEAR instruction is executing, PLC would calculate the average frequency according to input pulse amount per respond time. And output pulse based on EGEAR ratio, Slave shaft frequency could not exceed the highest frequency.

Parameter instruction				
Address offset	content	instruction	range	read/write
0	EGEAR's ratio (numerator)	Output pulse amount = input pulse amount	0~32767	read/write
1	EGEAR's ratio			

	(denominator)	(respond time) * numerator / denominator	0~32767	
2	Highest output pulse frequency(high byte)	the highest output frequency of slave shaft	0~200000	read/write
3	highest output pulse frequency(low byte)	the highest output frequency of slave shaft		read/write
4	average frequency of master shaft(high byte)	Average frequency of master shaft	-	read
5	average frequency of master shaft (low byte)	Average frequency of master shaft		read
6	Counted input pulse amount of master shaft (High byte)	Counted input pulse amount of master shaft	-	read
7	Counted input pulse amount of master shaft (Low byte)			
8	flag	Reserved	Reserved	Reserved
9	interval time	Actual value	-	read
10	EGEAR's ratio (numerator)	Actual value	-	read
11	EGEAR's ratio (denominator)	Actual value	-	read
12	highest output pulse frequency(high byte)	Actual value	0~200000	read/write
13	highest output pulse frequency (low byte)			read/write

2) Program example

The demo below shows the follow-up control between Y0 and Y3 (1:1 ratio).



```
0    M8000 User program run        [C235 K99999999]            C235 master's i input
6    M8002 The user program        [DMOV K10000 D40]
16   M8002 The user program        [MOV K1 D0]                 D0 Numerator
                                   [MOV K1 D1]                 D1 Denominator
                                   [MOV K20 D30]               D30 response time
/*
 *  output monitoring
 */
32   M8000 User program run        [DMOV D8152 D1052]          D8152 PLSYPLSR output
                                   [DMOV D8140 D1040]          D8140 PLSYPLSR output
51   M1 (rising edge)              [ZRST D8140 D8153]          D8140 PLSYPLSR output, D8153 PLSYPLSR output
                                   [DMOV H0 C235]              C235 master's i input
67   M1                            [DEGEAR C235 D0 D30 Y000 Y002]   C235 master's i input, D0 Numerator, D30 response time
89   M2                            [DPLSY D40 D42 Y003]
103                                [END]
```

Wiring: connect Y3 with X0

Control instruction: Step1: set M1 ON. Step2: set M2 ON. Then Y0 and Y3 will output pulse synchronously. (Y0 pulse amount: Y1 pulse amount =1:1)

3) Note

When the EGEAR value (Master shaft) read from high speed counter, the value can be changed when PLC is running. But the value could be changed if read from data resister (D) or regular counter (C).

4) Sign

- D8141 (high byte), D8140 (low byte): The number of output pulse in Y000. It would be reduced during reversal. (32-bit)
- D8143 (high byte), D8142 (low byte): The number of output pulse in Y001. It would be reduced during reversal. (32-bit)
- D8151 (high byte), D8150 (low byte): The number of output pulse in Y002. It would be reduced during reversal. (32-bit)
- D8153 (high byte), D8152 (low byte): The number of output pulse in Y003. It

would be reduced during reversal. (32-bit)

- M8145: Stop output pulse in Y000 (stop immediately)
- M8146: Stop output pulse in Y001 (stop immediately)
- M8152: Stop output pulse in Y002 (stop immediately)
- M8153: Stop output pulse in Y003 (stop immediately)
- M8147: Monitoring the output pulse in Y000 (BUSY/READY)
- M8148: Monitoring the output pulse in Y001 (BUSY/READY)
- M8149: Monitoring the output pulse in Y002 (BUSY/READY)
- M8150: Monitoring the output pulse in Y003 (BUSY/READY)

ECAMTBX instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DCAMTBX	Create E-CAM datasheet	32	No	DCAMTBX S_1 S_2 D_1 D_2	9

- S_1: D device could be used as Parameters’ address.

  Note: it used for creating E-cam chart, please refer to the [Appendix] - [Parameters List] for detailed.

- S_2: The type of E-cam Chart, D register or H, K could be used;

  Note:

  K0~K1: Create S type of acceleration and deceleration chart

  K100: Create rotary saw chart

  K101: Create fly saw chart;

- D_1: First address of E-cam parameters

  Note: Data for Chart stored in D_1+40, sections for Chart stored in D_1+38;

- D_2: The result of chart

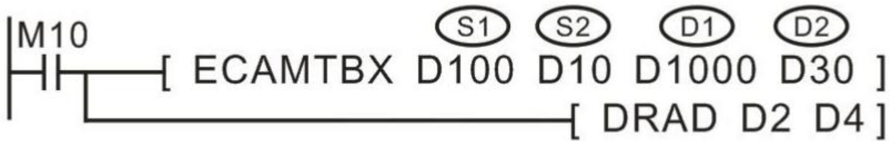
  Note:

  D_2<0: Error in chart generating

  D_2>0: Chart created successfully, D_2: Totally number of current chart sections;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2					√	√								√		
D_1														√		
D_2														√		

2) Program example



5.2.10 Handy instructions

IST instruction

1) Instruction description

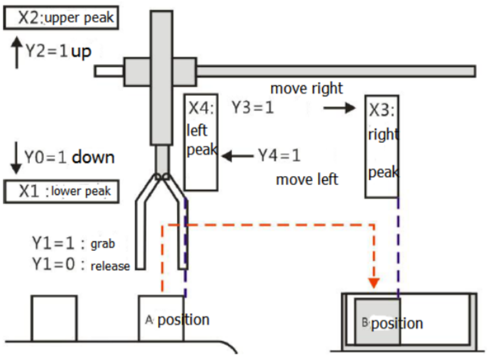
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
IST	Status initialization	16	No	IST S D_1 D_2	7

This instruction can be used to initialize the control status of a typical multi-action looping execution mechanism and to specify parameters for the operation mode such as the input signal, action status, etc.

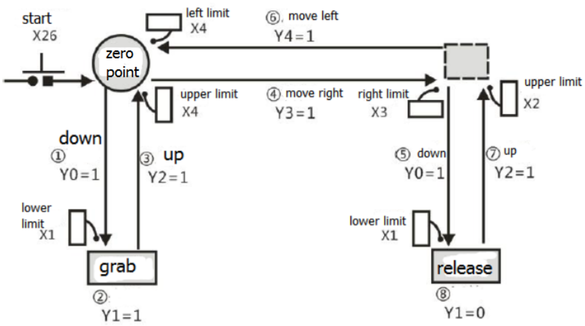
- S is the is the component address of the starting bit variable of the input of the specified operation mode. It occupies 8 continuous address units from S to S+7. The special function definition for each variable is described below:
- D_1 is the minimum serial number using the S status in the specified automatic operation mode.
- D_2 is the maximum serial number using the S status in the specified automatic operation mode. D_1 to D_2 are the status serial numbers of the looping action of the control system, which determine the status numbers.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	√	√	√													
D_1				√												
D_2				√												
Notice: 1) The instruction is allowed to be used only once in the user program. 2) For D1 and D2, only S20~S899 is available, and D1<D2. 3) The special M variable of the system will also be used when using this instruction.																

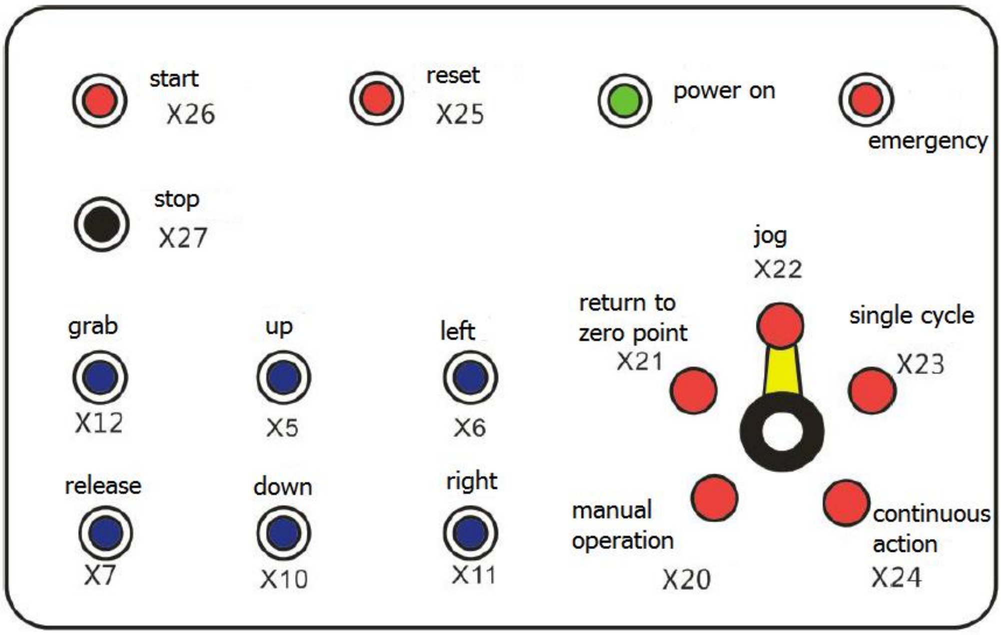
For example, in the illustrated system below, the execution mechanism acts sequentially in such a way: the grabbing device drops to the position of work piece A from the base point to grab the work piece, and then it lifts the work piece to the specified height and translates to the desired position and drops. After arriving at the required position, it releases the work piece and back tracks to start the next looping action. It is possible to use the IST instruction to specify the control signal input, the control of the status transferring, etc. of the operational mechanism to achieve automatic control. In addition, it supports manual commissioning of single-step actions and zero point reset, etc.



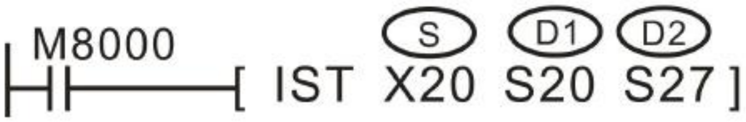
signal port distribution and action sequence:



Instruction keys and status changing switches are required to control the operational mechanism using manual commissioning, single actions, and looping actions, etc. The following is a schematic diagram of the operation panel, including the key ports and their function assignments:



For applications like the above diagram, each complete cycle can be divided into 8 steps (i.e. 8 statuses). The following instruction clauses can be used to initialize the status of the control system:



S specifies X20 as the starting input of the operation mode. Therefore, the input ports X21 to X27 of the subsequent addresses will also be used. The functional action features will be defined respectively as: (it is similar for variables X, M, or Y)

- X20: This is the manual operation mode to switch on/off the various control output signals using a single button.
- X21: This is the base point reset mode to reset the device to the base point by pressing the base point reset button.
- X22: This is the single-step operation mode to step forward a process each time the starting button is pressed.
- X23: This is the one-cycle looping mode. When the start button is pressed, it will run the one-cycle looping automatically and stop at the base point. The operation can be stopped by pressing the stop button. Then, if the start button is pressed, the operation will continue and stop at the base point automatically.
- X24: This is the continuous operation mode to run continuously by pressing the start button. When the stop button is pressed, it will move to the base point and stop.
- X25: To start the base point rest command signal.
- X26: To start the automatic command signal.
- X27: To stop the automatic command signal.

Note: In these port signals, the operation mode is determined by X20 to X24, for which the statuses can’t be ON at the same time. Therefore, it is suggested to use rotary switches for the selection and switching of the signals.

D1 and D2 are used to specify the minimum and maximum serial number S20 to S27 of the service statuses (8 for total) in the automatic operation mode. The following special variables for the definition and use requirements of the IST instruction should be noted:

When driving the IST instruction, the control of the following components will be automatically switched and can be referenced by user programs. In order to make the status switching and control of the IST instruction cooperate, the operation of certain special variables is required in the user programs. See the description in the table below:

Default variables in IST instruction		Variables driven in user program	
M8040	1= disable transfer of all states	M8043	1=original return completed. In the original return mode, after a

			machine returns to original, the special M variable will be set by the user program.
M8041	1= transfer start	M8044	1= original condition detect the original condition of a machine and drive the special assistant relay, it is set in all modes.
M8042	1= Start pulse	M8045	1= all output reset disabled. If a machine is switched among manual, return and automatic modes when it is not at original, all output and action states will be reset. But only action status can be reset if M8045 has been driven.
S0	Manual operation initial state	M8047	1= STL monitoring valid. After M8047 has been driven, the saved in the special assistant relay D8040~D8047 in ascendant order, thus monitoring action state numbers of 8 points. In addition, if any of these states is enabled, the special assistant relay M8046 will act.
S1	Original return initial state		
S2	Automatic operation initial state		

Under the "automatic operation" mode, free conversion is possible between: single step<-->one-cycle looping<-->continuous operation.

When performing conversion between "manual operation"<-->"base point reset"<-->"automatic operation" while the machine is running, the switched mode is effective after all the outputs are reset. (Reset is not applicable for M8045 drive.)

S10 to S19 can be used for the base point reset when using the IST instruction. Therefore, don't use these statuses as common statuses. In addition; S0 to S9 are used for the initial status process, S0 to S2, as mentioned in the above manual operations, are used for the base point reset and automatic operation, and S3 to S9 can be used freely.

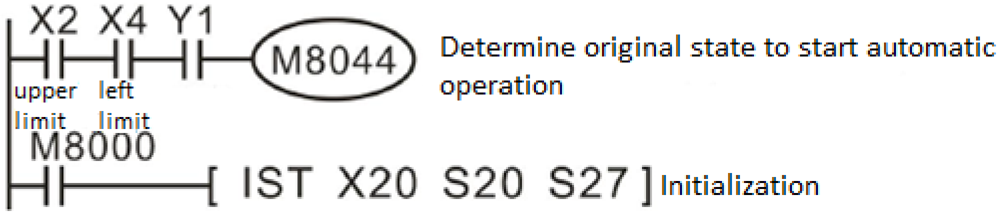
When programming, the IST instruction must be programmed with a higher priority than the various STL circuit, such as status S0 to S2, etc.

Rotary switches must be used to avoid the situation that X20 to X24 are ON at the same time.

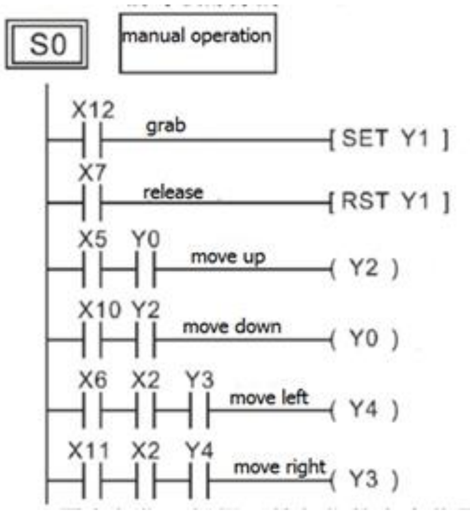
When switching between each (X20), base point reset (X21), auto (X22, X23, X24) before the base point completion signal (M8043) is activated, all the outputs are switched OFF. And the automatic operation can’t drive again until the base point reset is finished.

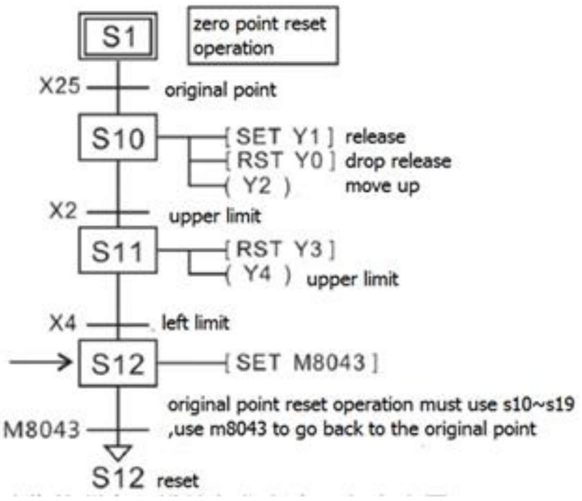
After initialization of the control instruction using the IST instruction, the action of each status of the execution mechanism and the conditions for status transferring need to be programmed, as detailed below:

- System initialization: defines the conditions for base point reset and defines the input ports of the operation mode signals used in the IST instruction and the status variables of the looping actions. The program clauses used are illustrated in the following diagram.

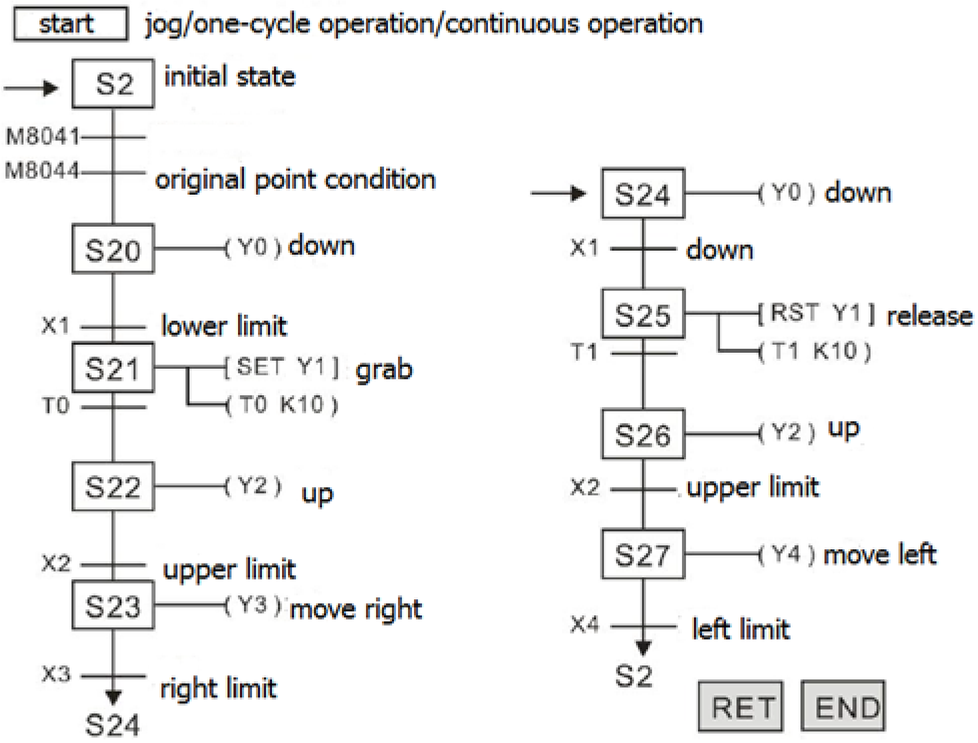


- Manual operation: driven to execute by the command signals defined on the operation plate. See the program clauses of status S0 in the following diagram. This part of the program can be skipped if there is no manual mode:





- Base point reset: design reset program based on the command signal at the starting of the reset and the sequence of the reset actions, as shown in the upper right:
- Automatic operation: write program based on the required action conditions and sequence and the control signal output, as shown in the diagram below:

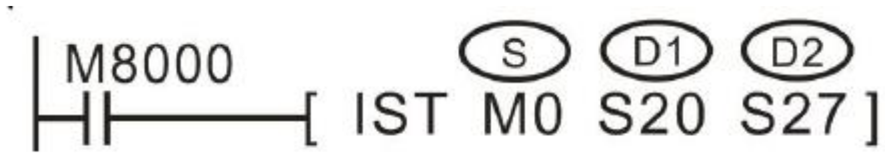


Up to this point, the control system is allowed to complete the looping action according to the above mentioned action requirements. The above programming description uses step instructions for the convenience of reading, while the user is free to program using the equivalent ladder diagrams.

When different status numbers occur to the "automatic operation" mode in a control system, the above example can be referenced to program in modifying the setting items of D1 and D2 corresponding works need to be done in the "automatic operation" mode.

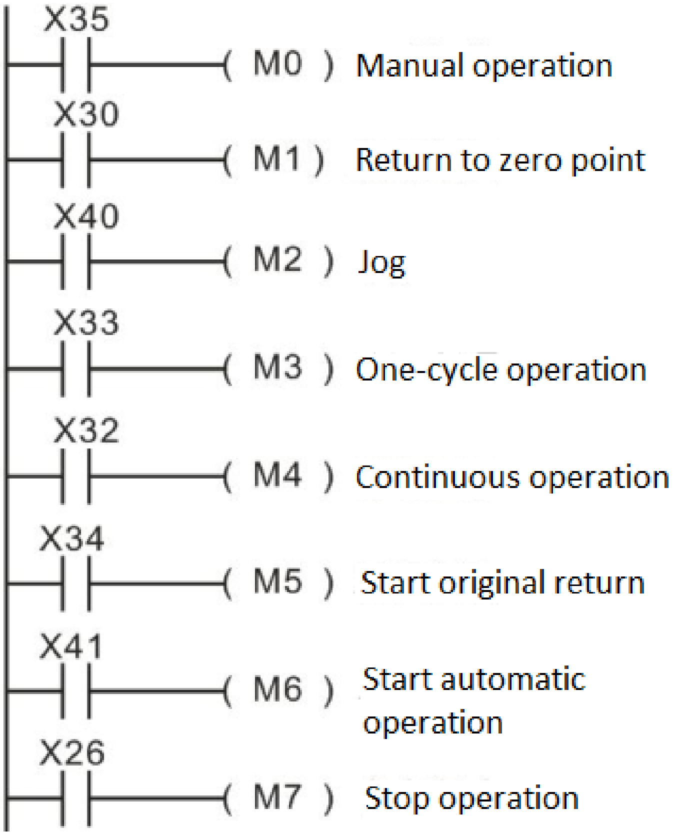
Handling methods for non-continuous X input:

If an X input port with non-continuous addresses needs to be used as the provided input of the operation mode, the M variable can be used for a "transitional" transmission. That is, the non-continuous X input status will be copied to an M variable with continuous addresses one by one using the simple OUT instruction rather than the instructions below:

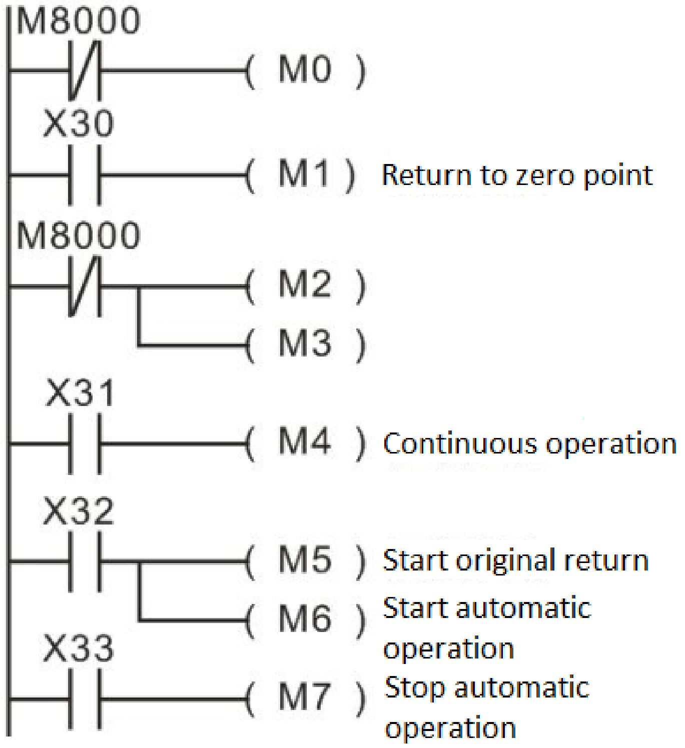


Specific to the continuous M0 to M7 variable area in the IST, the programming instructions can be used to shield the non-existent control mode. For example, the corresponding relationship between X as the mode input end and the M variable in the following diagram. For un-required modes, you simply input the M variable and fix it to zero:

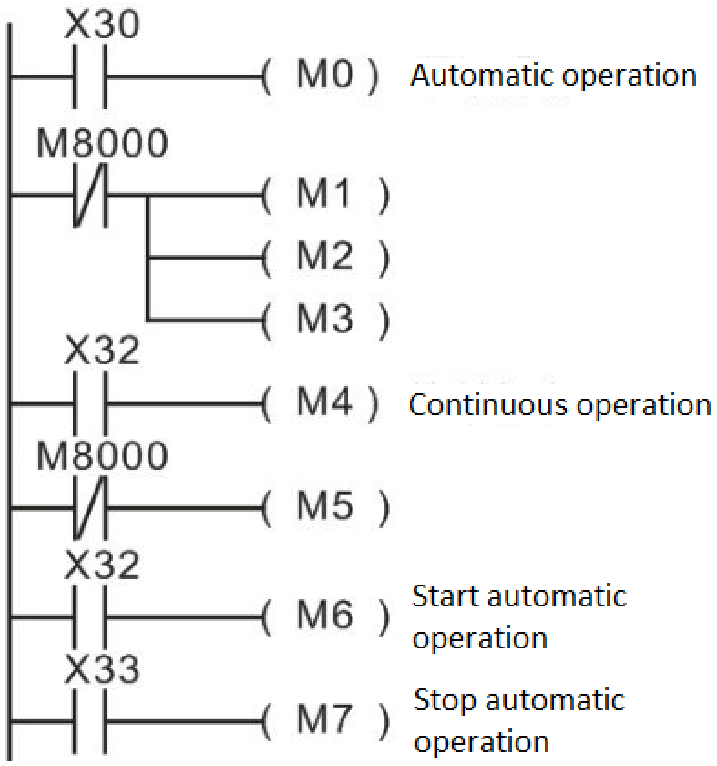
- When X input port is not continuous, then use continuous M register.



- Without manual mode



- Only with manual mode and continuous mode



SER instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
SER	Data search	16	No	SER S_1 S_2 D n	7
SERP		16	Yes		7
DSER		32	No		17
DSERP		32	Yes		17

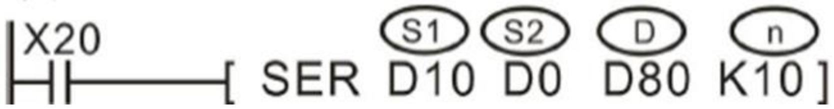
The instruction is to search the units with same data, or maximum value and minimum value.

- S_1 is the starting address of the data array;
- S_2 is the data, which is to be searched;
- D is the starting address of storage range for search result;
- n is the length of data range, which is to be searched. For 16 bit instruction, n=1~256, for 32 bit instruction, n=1~128.

When using 32 bit instruction, S_1, S_2, D and n are regarded as 32 bit.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1								√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D								√	√	√	√	√	√	√	√	√
n					√	√								√		

2) Program example



S_1	Retrieved data	S_2	Number	Condition
D10	D10=K100	Compare with (D10)=K100	0	Equal
D11	D11=K123		1	
D12	D12=K100		2	Equal
D13	D13=K98		3	
D14	D14=K111		4	

D15	D15=K66		5	minimum
D16	D16=K100		6	equal
D17	D17=K100		7	equal
D18	D18=K210		8	maximum
D19	D19=K88		9	

Search result

D	PARAMETER	DEFINATION
D80	4	No. of equal parameters
D81	0	serial number of the first equal parameter
D82	7	serial number of the last equal parameter
D83	5	Serial number of the minimum parameter
D84	8	Serial number of the maximum parameter

When X20 is ON, the operation is implemented;

The comparison method is signed algebra comparison, for example -8<2;

When there are several minimum or maximum, all the components with the largest serials number are displayed respectively;

The storage units for search results occupy five continue units started with D. If there is no same data, D80~D82 in above example are all 0.

ABSD instruction

3) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ABSD	E-Cam control (absolute mode)	16	No	ABSD S_1 S_2 D n	9
DABSD		32	No		17

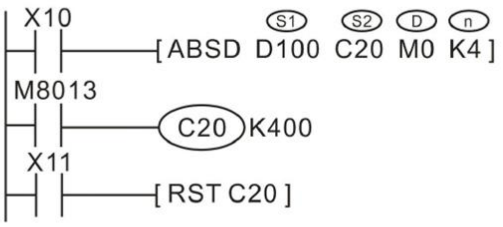
This instruction generates a variety of output patterns (there are n number of addressed outputs) in response to the current value of a selected counter (S_2).

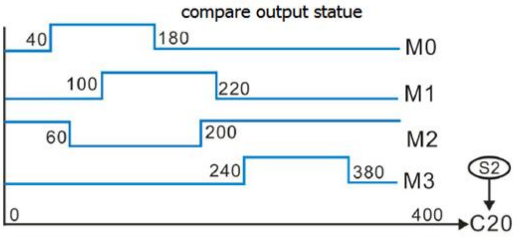
- S_1: The starting component address of the comparison table.
- S_2: The counter component serial number. When using 32 bit instruction, it could be used as a 32 bit counter.
- D: The starting address of the comparison result, occupying n several continuous bit variable units.
- n: The number of multi-segment comparison data.
- When using 32-bit instruction, S_1, S_2 and D are all pointing to 32-bit variable, and n is also calculated according to 32-bit variable width.

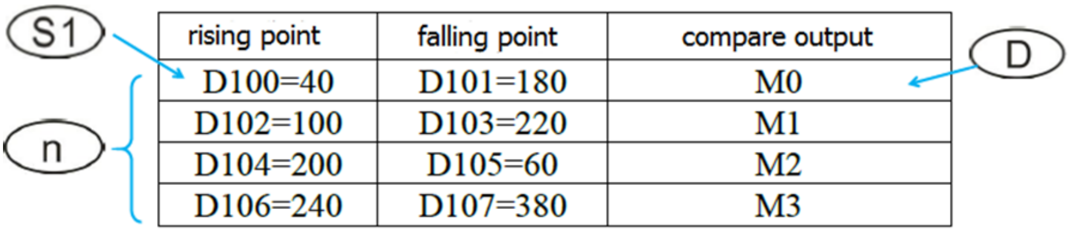
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1								√	√	√	√	√	√	√		
S_2													√			
D				√												
n	Constant, n=1~64															

When S_1 operands are KnX, KnY, KnM,KnS , if it is 16-bit instruction, K4 must be specified; if it is 32bit instruction, K8 must be specified and the component number of X,Y,M,S must be a multiple of 8. S_1 operand can only specify C0 to C199 with 16-bit instruction, and specify C200 to C254 with 32-bit instruction.

4) Program example







rising point	falling point	compare output
D100=40	D101=180	M0
D102=100	D103=220	M1
D104=200	D105=60	M2
D106=240	D107=380	M3

Before ABSD instruction is implemented, all the variables in the table should be assigned a value by MOV instruction.

Even there are high-speed devices in the DABSD instruction, the comparison result D is also affected by user program scan time delay. For the application with time response requirement, the HSZ high-speed comparison instruction is recommended.

ABSD can be only used once in the program.

INCD instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
INCD	E-Cam control increment mode	16	No	SORT S_1 S_2 D n	9

The instruction to complete the operation is multi-section comparison, it is used for E-cam control, comparison tables, counters, etc. is set by incremental mode. The instruction is executed in the main program and the result of the comparison is affected by the lag of the scan time.

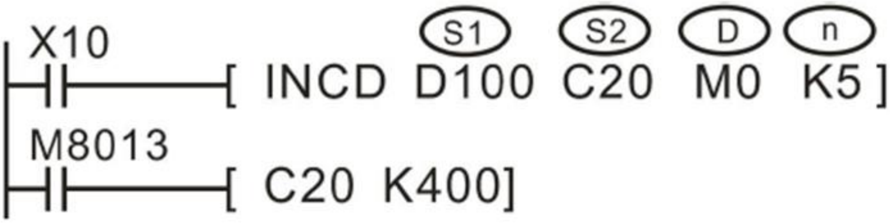
- S_1: The comparison table.
- S_2: The timer. The neighboring S2+1 unit is used to reset the time on the counter after the calculation and comparison process. (32bit counters are applicable to 32bit instructions)
- D: The comparison results record, which is a bit variable unit occupying n continuous addresses.
- n: The number of multi-segment comparison sets.

When the set comparison of N is done, the "instruction done" flag "M8029" will automatically switch on.

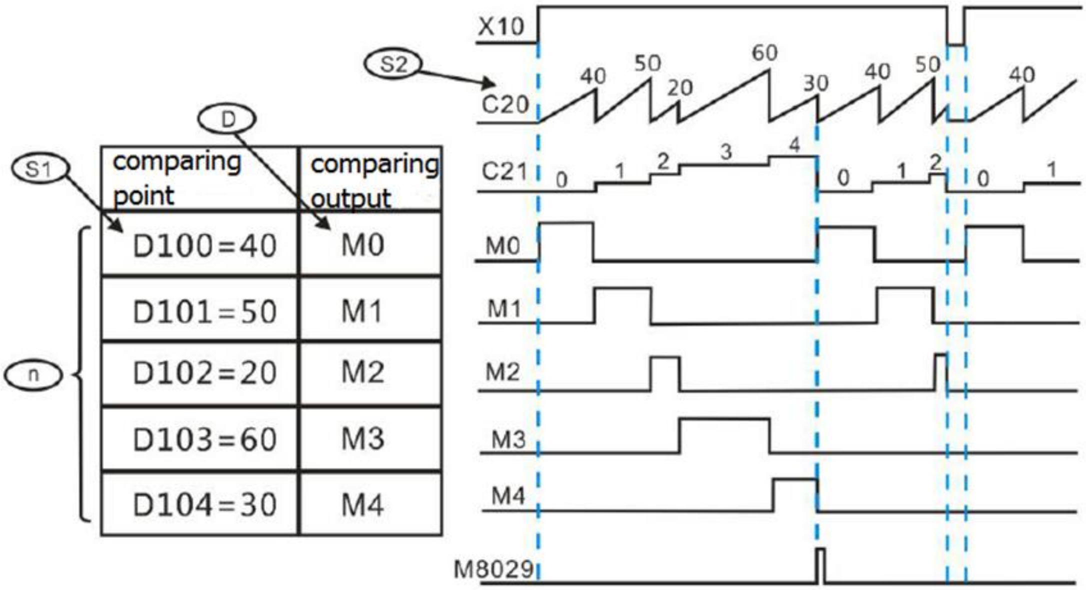
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1								√	√	√	√	√	√	√		
S_2													√			
D																
n		√	√	√												

For 16bit –S_1 operation numbers KnX, KnY, KnM and KnS, "K4" must be specified.
For 32bit -"K8" must be specified and the number of components X, Y, M and S must be multiples of 8.
S_1 operation numbers are limited to C0~C199 for 16bit instruction.
S_1 operation numbers are limited to C200~C254 for 32bit instruction

2) Program example



If the relevant variables have been set as follows, when X10=ON, the implementation result is shown as the following figure.



comparing point	comparing output
D100=40	M0
D101=50	M1
D102=20	M2
D103=60	M3
D104=30	M4

All the variables of the relevant tables should be assigned a value by MOV instruction before implementing the INCD instruction.

The comparison output is also affected by the delay of the program scanning time. Therefore, the HSZ high speed comparison instruction can be used.

The INCD instruction can only be used once in the program.

TTMR instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
TTMR	Monitors the duration of a signal and places the timed data into a data register	16	No	TTMR D n	5

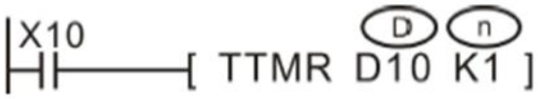
The duration of time that the TTMR instruction is energized, is measured and stored in device D+1 (as a count of 100ms periods). The data value of D+1 (in secs), multiplied by the factor selected by the operand n, is moved in to register D. The contents of D could be used as the source data for an indirect timer setting or even as raw data for manipulation. When the TTMR instruction is de-energized D+1 is automatically reset (D is unchanged).

- When n=K0, the actual multiple is 1;
- When n=K1, the actual multiple is 10;
- When n=K2, the actual multiple is 100;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D														√		
n					√	√										

2) Program example

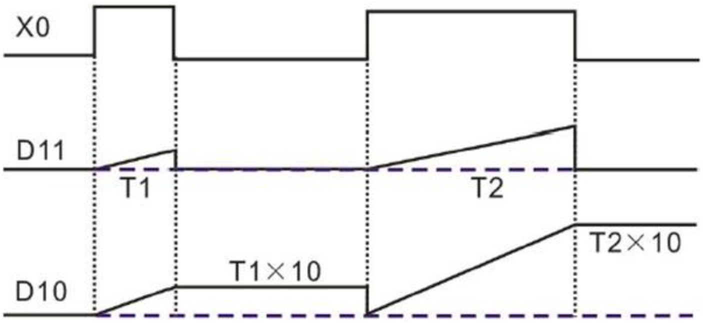
Example 1:



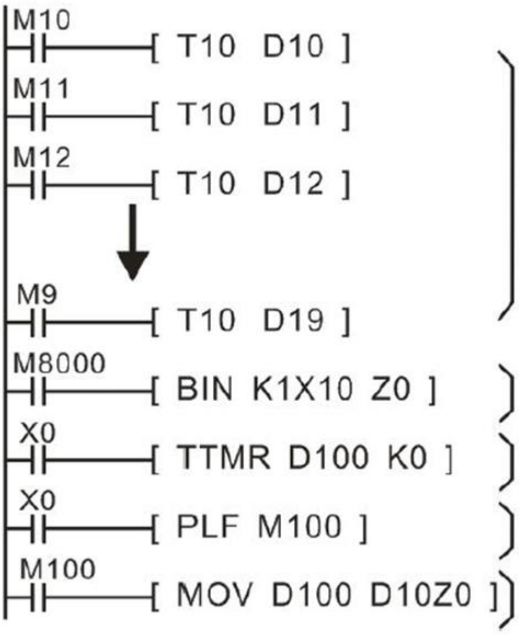
- When X10 is closed, D10=D11;
- When X10 is opened, D100 remains the same and D11 becomes 0.

If holding time of pressing key X10 is T seconds, the relationships between D10, D11, and n are listed as below:

n	D10	D11(unit: 100ms)
K0(unit: 1 s)	1*T	D11=D10*10
K1(unit: 100ms)	10*T	D11=D10
K2(unit: 10 ms)	100*T	D11=D10/10



Example 2:



- Use TMR instruction to write ten sets of setting time to D10~D19 in advance. This set of timers are 100ms timer, so the 1/10 of the teach data are actual action time(sec).
- Connect 1 digit DIP switch to X10~X13 and use one BIN instruction to convert the setting value of the DIP switch to BIN and save it to Z0.
- On time for X0(sec.) is saved in D100.
- M100 is the one-time scanning cycle pulse produced by the release of the demo timer button X0.
- Use setting no. of DIP switch as an indirectly specified pointer and send the content of D100 to D10Z0 (D10~D19).

STMR instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
STMR	Special timer	16	No	STMR S m D	7

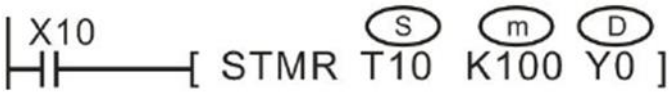
The function of this instruction is to generate 4 kinds of special instruction of delay action according to instruction power flow.

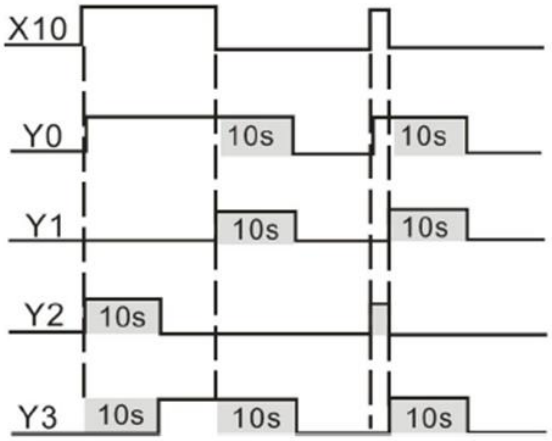
- S: The timer number. T0~T19 can be used for triggering delay action
- m: The delay setting in 100 ms ranging from K1 to K32767;
- D: The starting number for delay action outputting components and occupies 4 consecutive units.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S												√				
m	Constant, 1~32767															
D		√	√	√												

2) Program example

Example 1:





When X10 turns from OFF to ON, Y0 will turn OFF after a delay of 10 seconds.

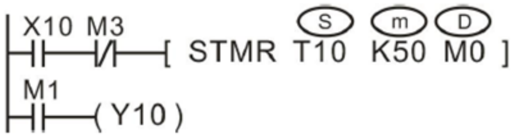
When X10 turns from ON to OFF, Y1=ON after a delay of 10 s.

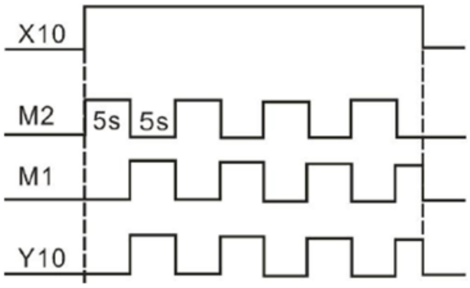
When X10 turns from OFF to ON, Y2=ON after a delay of 10 s.

When X10 turns from OFF to ON, Y3=ON after a delay of 10 s

Example 2:

It is easy to generate a oscillator output. (The function can also be implemented by using a ALT instruction), which is shown as below:





ALT instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ALT	Alternate State	16	No	ALT D	3
ALTP		16	Yes		3

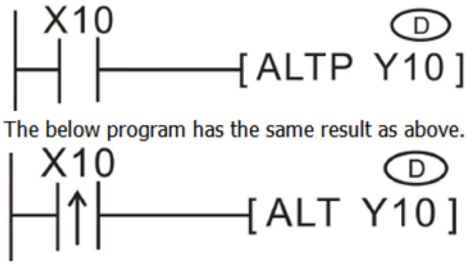
The status of the destination device (D) is alternated on every operation of the ALT instruction.

This instruction reverses D component state when the energy flow is effective.

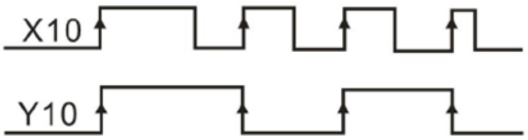
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
D		√	√	√												

2) Program example

Example 1:

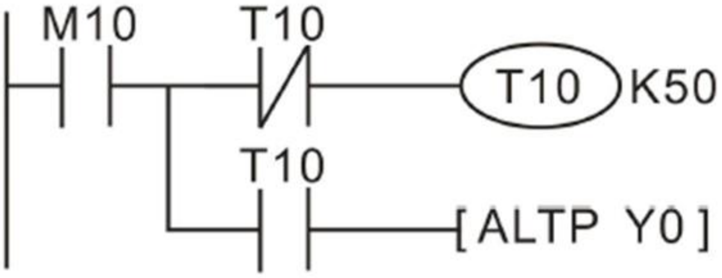


The below program has the same result as above.



Example 2:

With the use of timer, it is easy to generate an oscillator output. The function can also be implemented by using a special timer STMR instruction), which is shown in the following figure.



RAMP instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
RAMP	Ramps a device from one value to another in the specified number of steps	16	No	RAMP S_1 S_2 D n	9

The RAMP instruction varies a current value (D) between the data limits set by the user (S_1 and S_2). The 'journey' between these extreme limits takes n program scans. The current scan number is stored in device D+1. Once the current value of D equals the set value of S_2 the execution complete flag M8029 is set ON.

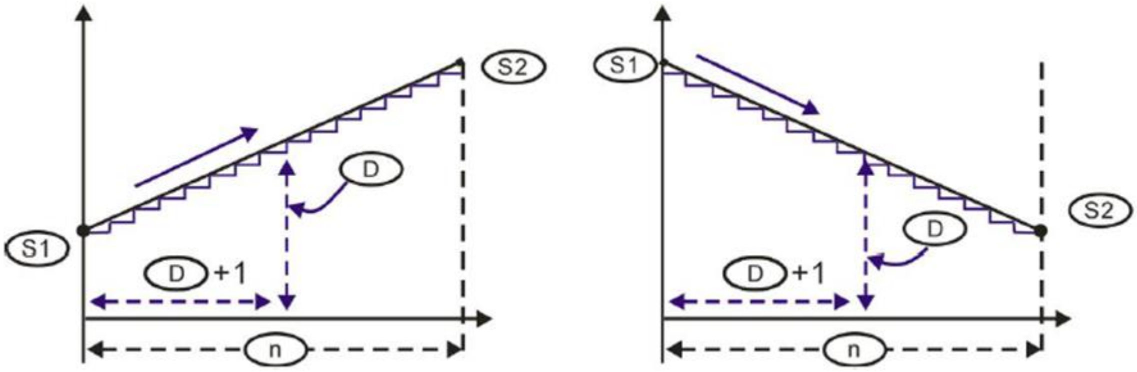
The RAMP instruction can vary both increasing and decreasing differences between S_1 and S_2.

- S_1: The starting value unit of slope signal
- S_2: The end-point value unit of slope signal
- D: The memory point for procedure value of linear interpolation signal, yet the timer which is used to count the times of interpolation is stored in unit D+1
- n: The times of program scanning execution for process of Interpolation. Because the output of interpolation is carried on during main loop, it's necessary to set the program execution to fixed scanning mode. (The demonstration is on M8039, D8039)

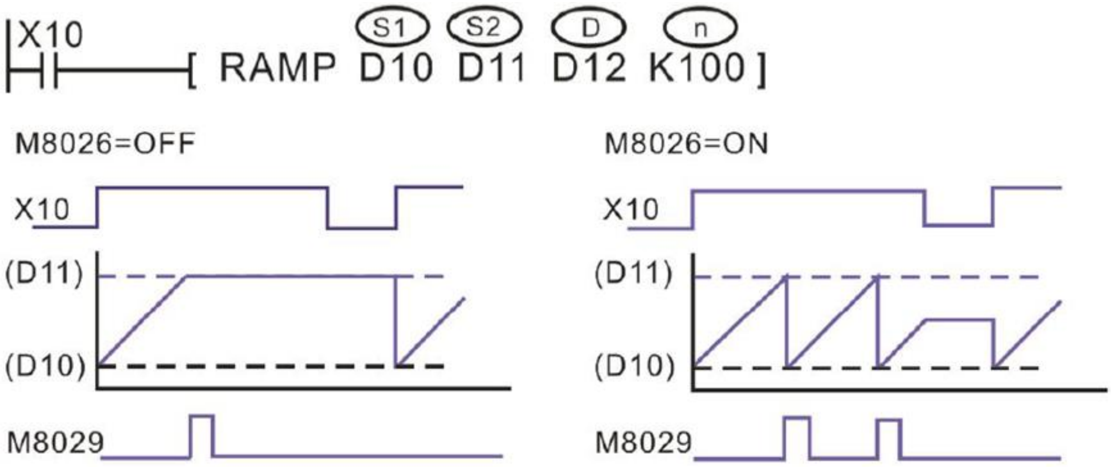
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2														√		
D														√		
n	Constant, 1~32767															

2) Program example

The interpolation calculation is based on integer number and has discarded the decimal calculation. Command function is showed in the following chart:



There are 2 modes for RAMP command execution which is defined by M8026. After every interpolation, M8029 set on for a scanning cycle .The execution features is showed in the follow example:



ROTC instruction

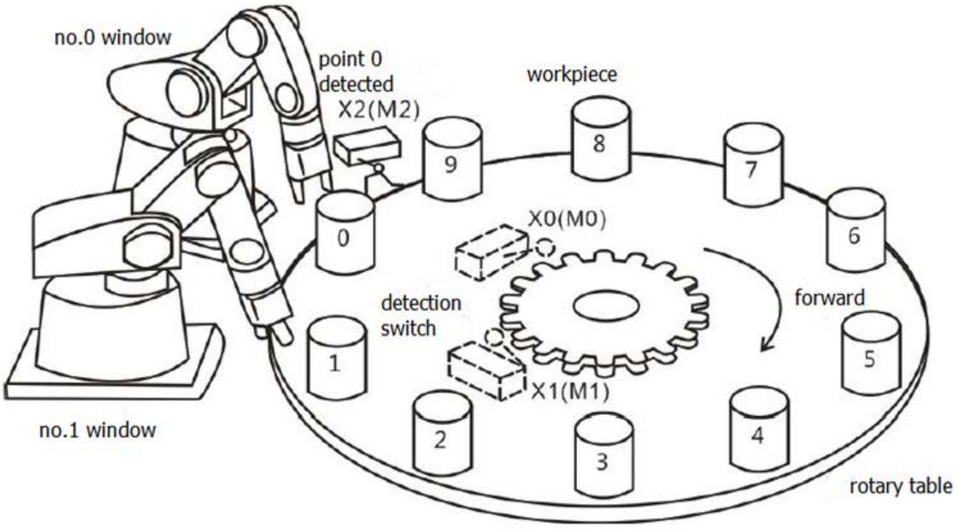
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ROTC	Controls a rotary tables movement is response to a requested destination/ position	16	No	ROTC S D m1 m2	9

The ROTC instruction is used to aid the tracking and positional movement of the rotary table as it moves to a specified destination.

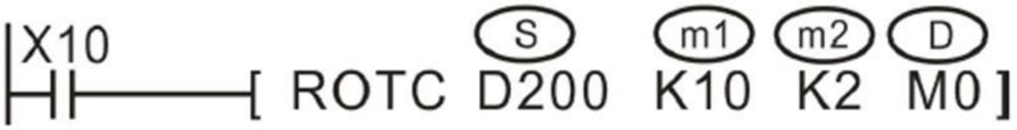
- S: The initial cell of count variable.
- m1: Numbers of station on rotary workbench, which must be M1 ≥ M2;
- m2: Numbers of low-speed rotary workbench, which must be M1 ≥ M2;
- D: The initial cell to storage position detection signal of rotary workbench, which occupies the next 8 bit variable units.

As the picture below, X0, X1 connect with the A and B phase output of AB Quadrature Encoder respectively, and we can get the Quadrature signals by mechanical switch. X2 will be used as the detection input of No.0 station ("ON" when turning to No.0 station), the rotational speed, direction, and workstation can be detected by these three signals.



Operands	**Bit device**				**Word device**											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S				√										√		
D		√	√	√												
m₁	Constant, 2~32767, m1>=m2															
m₂	Constant, 0~32767, m1>=m2															

2) Program example



SORT instruction

1) Instruction description

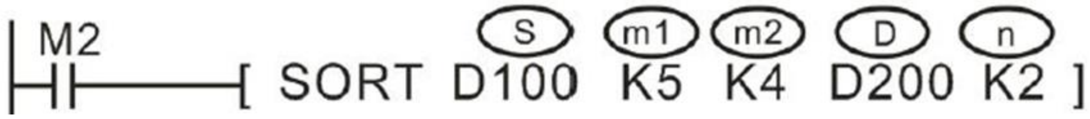
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
SORT	Data in a defined table can be sorted on selected fields while retaining record integrity	16	No	SORT S m1 m2 D n	17

This instruction constructs a data table of m1 records with m2 fields having a start or head address of S. Then the data in field is sorted in to numerical order while retaining each individual records integrity. The resulting (new) data table is stored from destination device D.

- S: The starting unit of the first variable in first line (or called first record);
- m1: The line number of the array, or called record number;
- m2: The row number, or called item number in each record;
- D: The starting unit for saving result, occupying following variable unit number is same as that of array before sorting;
- n: The array row number, according which the sort operation is implemented. n is within the range of 1 ~ m2.

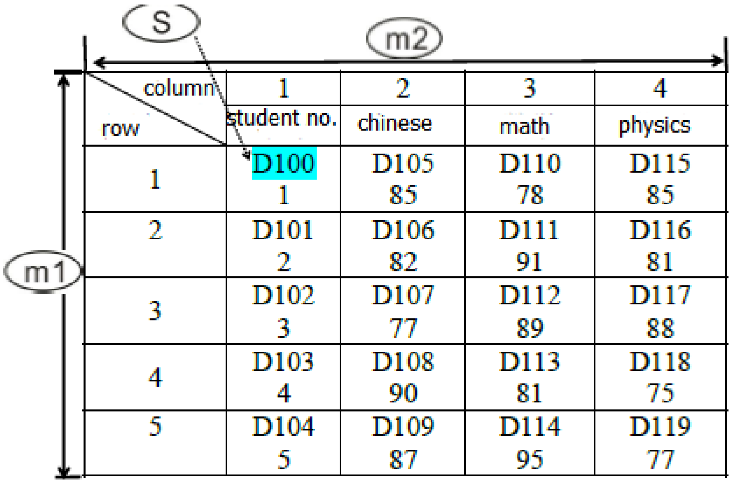
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S				√										√		
m_1	Constant, 1~32															
m_2	Constant, 1~6															
D														√		
n					√	√								√		

2) Program example



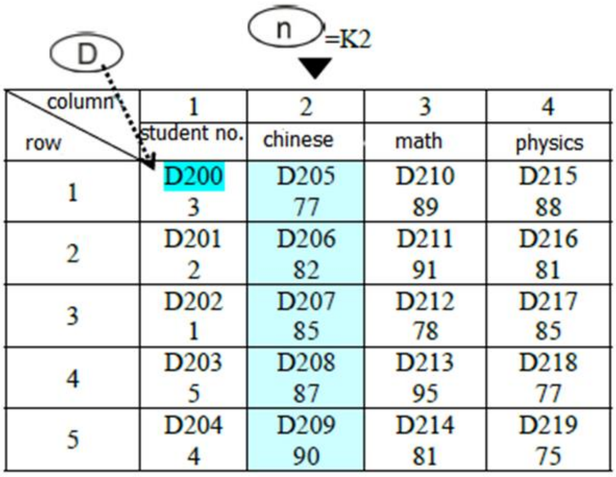
When X10=ON, sort operation is implemented, and after the implementation, M8029 is set on for one scanning cycle; If it needs re-sorting, X10 should be reset and turn on again.

The equivalent form of the above instruction and its data:



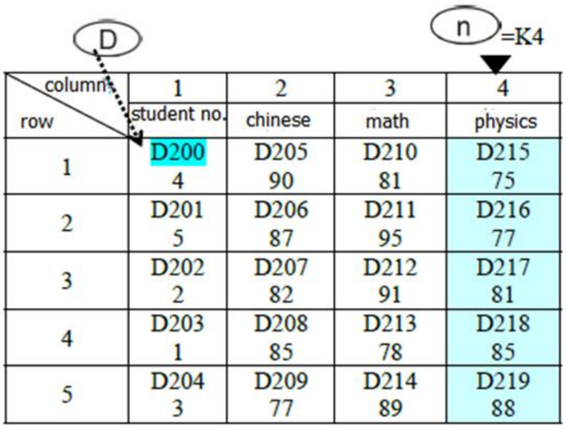
column / row	1	2	3	4
	student no.	chinese	math	physics
1	D100 1	D105 85	D110 78	D115 85
2	D101 2	D106 82	D111 91	D116 81
3	D102 3	D107 77	D112 89	D117 88
4	D103 4	D108 90	D113 81	D118 75
5	D104 5	D109 87	D114 95	D119 77

The result of sorting when N=k2 is as below:



column / row	1	2	3	4
	student no.	chinese	math	physics
1	D200 3	D205 77	D210 89	D215 88
2	D201 2	D206 82	D211 91	D216 81
3	D202 1	D207 85	D212 78	D217 85
4	D203 5	D208 87	D213 95	D218 77
5	D204 4	D209 90	D214 81	D219 75

The result of sorting when N=k4 is as below:



column / row	1	2	3	4
	student no.	chinese	math	physics
1	D200 4	D205 90	D210 81	D215 75
2	D201 5	D206 87	D211 95	D216 77
3	D202 2	D207 82	D212 91	D217 81
4	D203 1	D208 85	D213 78	D218 85
5	D204 3	D209 77	D214 89	D219 88

5.2.11 Positioning control

DABS instruction

1) Instruction description

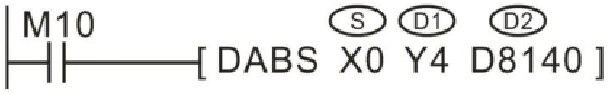
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DABS	Absolute current value read	32	No	DABS S D_1 D_2	13

This instruction reads the absolute position data when a servo motor with absolute positioning function is connected.

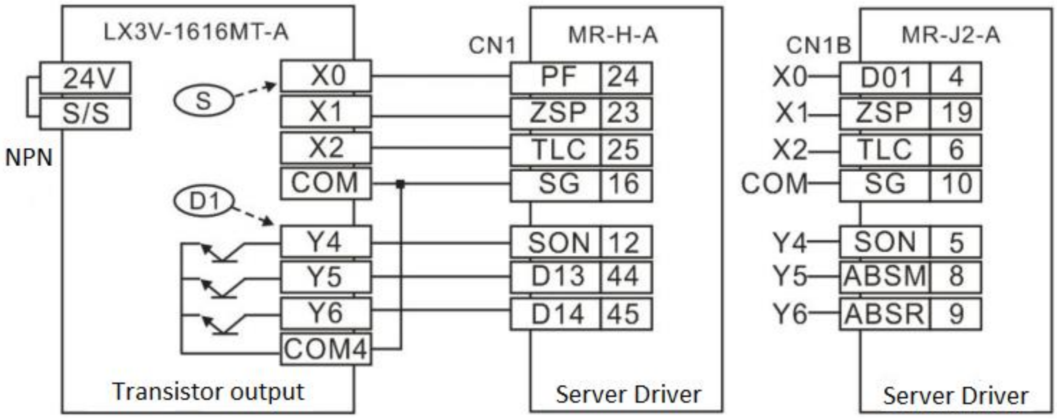
- S: The first of three inputs used for communication flags;
- D_1: The first of three communication outputs;
- D_2: The data destination registers;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	√	√	√	√												
D_1		√	√	√												
D_2									√	√	√	√	√	√	√	√

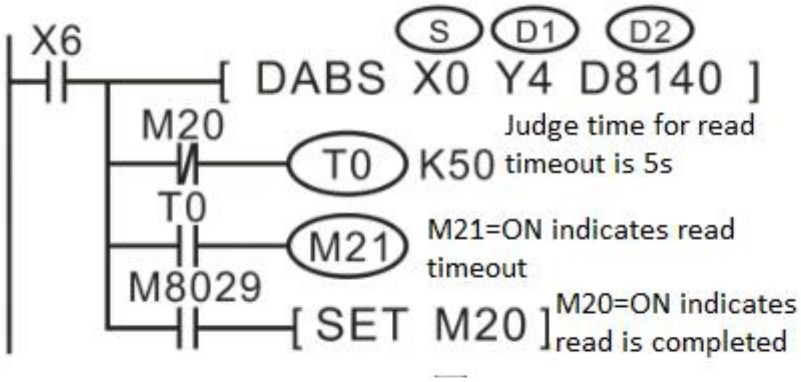
2) Program example



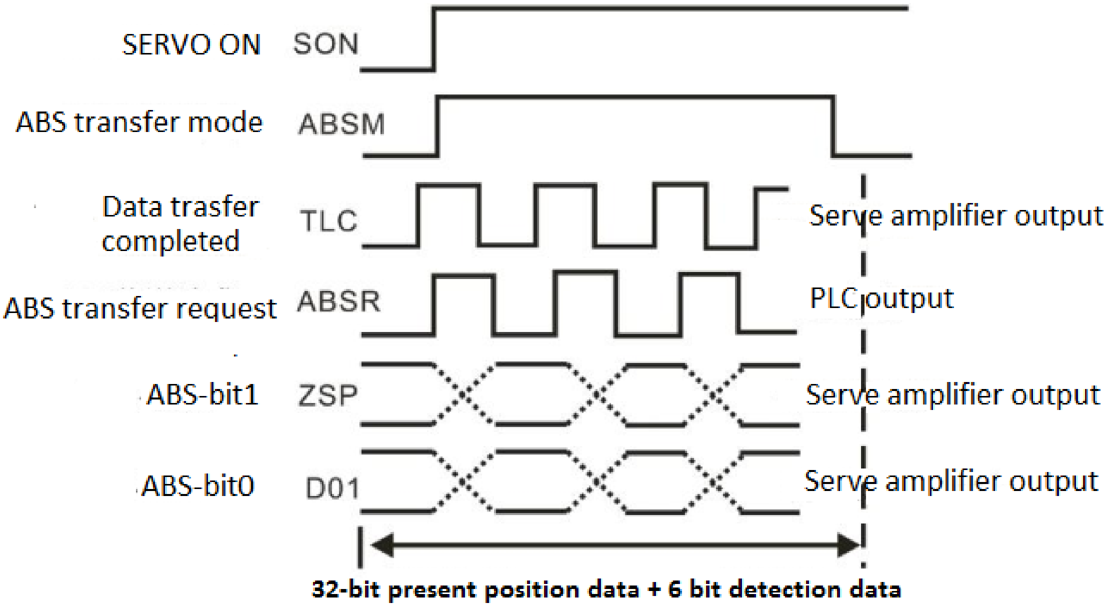
Corresponding wiring as shown below, it shows servos drive with absolute position detection of the encoder servo motor.



- When M10 is set to ON, it begins to read. When DABS instruction is completed, the M8029 flag is set to ON;
- When the instruction is running in process and the driver flag is set to OFF, the read operation will be interrupted;
- The programming example for reading ABS data is as follows: when the X6 terminal is closed, it begins to read. If it is not completed in 5s, the timeout flag M21 will be set. The demo is as following:


X6
S
D1
D2
DABS X0 Y4 D8140
M20
T0 K50
Judge time for read timeout is 5s
T0
M21
M21=ON indicates read timeout
M8029
SET M20
M20=ON indicates read is completed

The signal time sequence of the ABS read operation is shown in the following figure. When implementing an instruction, the PLC will automatically implement the access operation with servo driver.



ZRN instruction

1) Instruction description

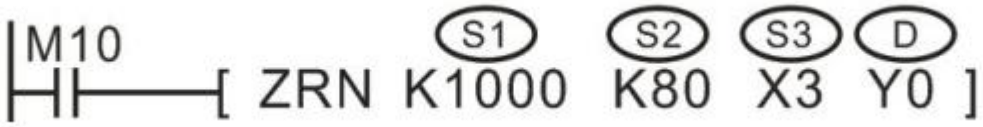
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ZRN	Return to home point	16	No	ZRN S_1 S_2 S_3 D	9
DZRN		32	No		13

When executing incremental or absolute positioning, the PLC stores the current position values which increase or decrease during operation. Using these values, the PLC always knows the machine position. When the power to the PLC is turned off, this data would be lost. To cope with this the machine should return to the zero point when the power is turned ON, or during initial set up, to teach the zero position.

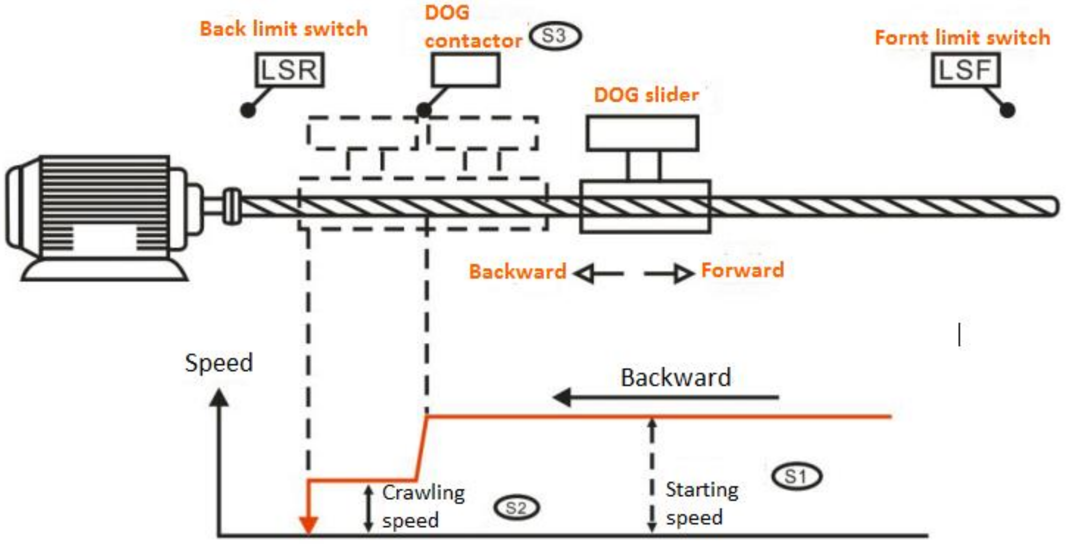
- S_1: The Zero Return Speed, the range is 10~32,767Hz (16bit), 10~100,000Hz (32bit);
- S_2: The Creep Speed, the range is 10~32,767Hz;
- S_3: The Near Point Signal;
- D: The Pulse Output Designation, Y0~Y3 are for LX3V (2N firmware), LX3VP, LX3VE, Y0~Y1 are for LX3V (1S firmware);

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
S_3	√	√	√	√												
D		√														

2) Program example



This instruction means that, after M10 turns ON, PLC sends out pulses at speed of 1000Hz fromY0, which makes stepper motor run back toward original point. While when X3 (DOG) turns ON, the output pulse frequency turns into 80Hz creep speed, until X3(DOG) turns OFF again, and Y0 stops to output pulse, and write 0 to the current value register (Y000: [D8141, D8140], Y001: [D8143, D8142]). In addition, when M8140 (clear signal output function) is ON, a clear signal is output at the same time. Subsequently, when the execution completion flag (M8029) is turned ON, the monitoring of the pulse output (Y000: [M8147], Y00l: [M8148]) becomes OFF.



During this instruction is executed, systemic variables concerned are:

- D8141 (high byte), D8140 (low byte):Y000 outputs value of current register (using 32 bit)
- D8143 (high byte), D8142 (low byte):Y001 outputs value of current register (using 32 bit)
- M8145: Y000 represents the pulse output stopped (instantly)
- M8146: Y001 represents the pulse output stopped (instantly)
- M8147: Y000 represents monitoring during the pulse output process (BUSY/READY)
- M8148:Y001 represents monitoring during the pulse output process (BUSY/READY)

Since servo driver has the function of power-fail-safeguard towards location information, this command does not need to execute after power-on every time. Meanwhile, for servo driver can only move one way, movement of backing to original point must be done before DOG.

DRVI instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DRVI	Increment positioning	16	No	DRVI S_1 S_2 D_1 D_2	9
DDRVI		32	No		17

This instruction is for single speed positioning in the form of incremental movements.

- S_1: The number of pulses, the range is 16-bit -32,768 to 32,767 pulses or 32-bit -2,147,483,648 to 2,147,483,647 pulses;
  If D_1=Y0, [D8141 (high byte), D8140 (low byte)] (32-bit) are increment position;
  If D_1=Y1, [D8143 (high byte), D8142 (low byte)] (32-bit) are increment position;
  If D_1=Y2, [D8151 (high byte), D8150 (low byte)] (32-bit) are increment position;
  If D_1=Y3, [D8153 (high byte), D8152 (low byte)] (32-bit) are increment position;
- S_2: The output frequency, he range is 16-bit 10 to 32,767Hz or 32-bit 10 to 200 kHz;
- D_1: The Pulse Output Designations, only Y000 or Y001or Y002 or Y003 in LX3V (2N firmware), LX3VP, LX3VE and LX3VM can be used for the pulse output. Only Y000 or Y001 in LX3V (1S firmware) can be used for the pulse output.
- D_2: The rotation direction signal. If D_2= OFF, rotation = negative, if D_2= ON, rotation = positive.

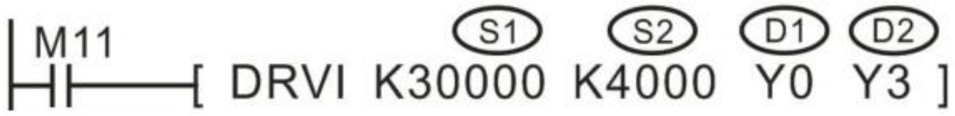
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D_1		√														
D_2		√	√	√												

If the contents of an operand are changed while the instruction is executed, it is not reflected on the operation. The new contents become effective when the instruction is next driven.

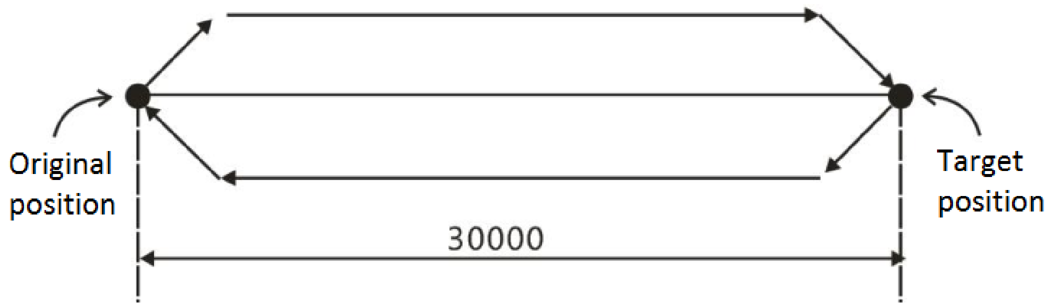
If the instruction drive contact turns off while the instruction is being executed, the machine decelerates and stops. At this time the execution complete flag M8029 does not turn ON.

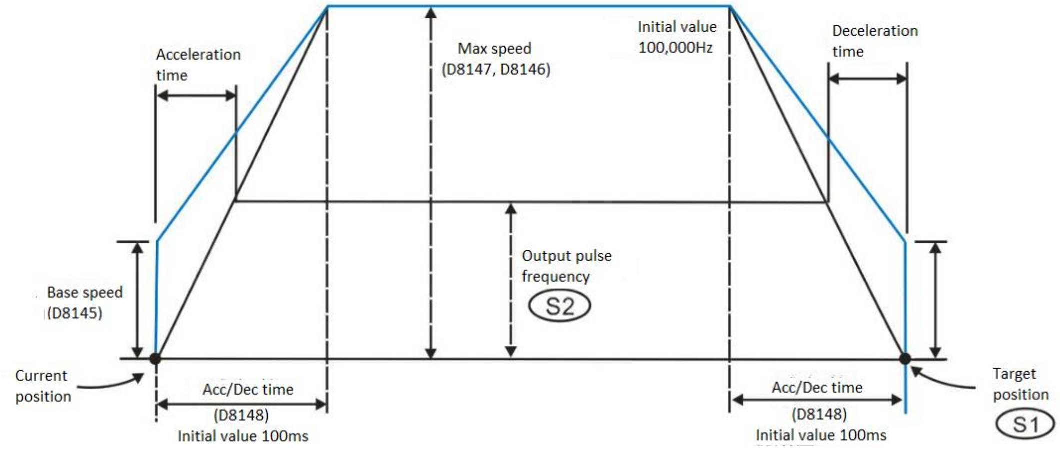
Once the instruction drive contact is off, re-drive of the instruction is not possible while the pulse output flag (Y000: [M8147], Y001: [M8148]) is ON.

2) Program example



With 30000 pulses exported from the Y0 port at the frequency of 4 kHz, the external server allows the machine to operate in directions that are determined by Y3.





During the pulse output, the frequency will increase or decrease according to the predetermined value.

The minimum value of output pulse frequency which can be actually used is determined by the following equation.

$$\sqrt{(\text{Maxspeed}[\text{D8147}, \text{D8146}]/(2 * (\text{Acc or Dec time }[\text{D8148}]\text{ms}/1000))}$$

Even if the assigned value is lower than the above calculated result, the frequency to be exported will still be the calculated value. The frequencies in the initial stage of acceleration and in the final section of deceleration must not be lower than the above calculated result.

During the instruction execution, the involved system variables are as follows:

- [D8145]: Base speed when executing DRVI and DRVA instructions. During the operation of stepping motor, the stepping motor's resonance region and automatic start frequency must be considered when setting up the speed. Setting Range: below 1/10 of the maximum speed (D8147, D8146). When the setting surpasses the indicated range, the operating speed will automatically decelerate to the 1/10 of the highest speed.
- [D8147 (high byte), D8146 (low byte)]: Maximum speed when executing DRVI and DRVA instructions. The assigned output pulse frequency must be lower than the maximum speed. Setting range: 10 ~100,000(Hz)
- [D8148]: acceleration and deceleration time when executing FNC158 (DRVI) and FNC159 (DRVA) instructions. Acceleration/Deceleration time means the time required in order to reach the maximum speed (D8147, D8146). The output pulse frequency is lower than the maximum speed (D8147, D8146), the actual acceleration/deceleration time will reduce. Setting range: 50 ~ 5,000 (ms)
- [M8145]: Y000 pulse output stopping (immediate stopping)
- [M8146]: Y001 pulse output stopping (immediate stopping)
- [M8152]: Y002 pulse output stopping (immediate stopping)
- [M8153]: Y003 pulse output stopping (immediate stopping)
- [M8147]: Y000 pulse output monitoring (BUSY/READY)
- [M8148]: Y001 pulse output monitoring (BUSY/READY)
- [M8149]: Y002 pulse output monitoring (BUSY/READY)
- [M8150]: Y003 pulse output monitoring (BUSY/READY)

3) Note for use

- Position instruction (ZRN/PLSV/DRVI/DRVA) can be reused in the program, but do not output to the same port;
- If the drive power flow for an instruction turns OFF and ON again, if can only be driven after one operation cycle when status bit (Y000: [M8147], Y001: [M8148], Y0002: [M8149], Y003: [M8150]) turns OFF.
- When positioning instruction is driven again, there should be at least one cycle of OFF time. If the re-drive is implemented in the time less than above condition, there will be calculation error when firstly implementing calculation instruction.

PLSV instruction

1) Instruction description

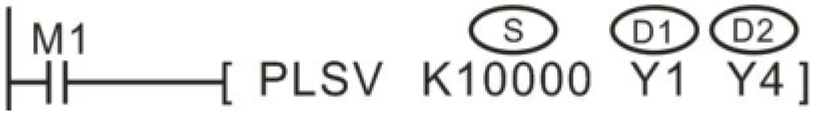
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
PLSV	Variable speed pulse output	16	No	PLSV S D_1 D_2	9
DPLSV		32	No		13

This is a variable speed output pulse instruction, with a rotation direction output. Only the PLC with the transistor output can execute the instruction.

- S: The pulse frequency. In 16-bit mode, the range are 1~32,767Hz and -1~-32,768Hz. In 32-bit mode, the range are 1~ 200,000Hz and -1~-200,000Hz;
- D_1: The pulse output designation, Y0~Y3 are specified by LX3V (2N firmware), LX3VP, LX3VE and LX3VM, Y0~Y1 are specified by LX3V (1S firmware);
- D_2: The rotation direction, the ON state means forward, the OFF state means reverse;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D_1		√														
D_2		√	√	√												

2) Program example



When M1 is triggered, Y1 is specified for output pulse, the frequency of pulse is 10KHz, Y4 is specified for direction control. If Y4=ON means forward.

During this instruction is executed, systemic variables concerned are:

- D8141 (high byte), D8140 (low byte):Y000 outputs value of current register (using 32 bit)
- D8143 (high byte), D8142 (low byte):Y001 outputs value of current register (using 32 bit)
- M8145: Y000 represents the pulse output stopped (instantly)
- M8146: Y001 represents the pulse output stopped (instantly)

- M8147: Y000 represents monitoring during the pulse output process (BUSY/READY)
- M8148:Y001 represents monitoring during the pulse output process (BUSY/READY)

DRVA instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DRVA	Absolute positioning	16	No	DRVA S_1 S_2 D_1 D_2	9
DDRVA		32	No		17

This instruction is for single speed positioning using a zero home point and absolute measurements.

- S_1: The number of pulses, the range is 16-bit -32,768 to 32,767 pulses or 32-bit -2,147,483,648 to 2,147,483,647 pulses;
  If D_1=Y0, [D8141 (high byte), D8140 (low byte)] (32-bit) are absolute position;
  If D_1=Y1, [D8143 (high byte), D8142 (low byte)] (32-bit) are absolute position;
  If D_1=Y2, [D8151 (high byte), D8150 (low byte)] (32-bit) are absolute position;
  If D_1=Y3, [D8153 (high byte), D8152 (low byte)] (32-bit) are absolute position;
- S_2: The output frequency, he range is 16-bit 10 to 32,767Hz or 32-bit 10 to 200 kHz;
- D_1: The Pulse Output Designations, only Y000 or Y001or Y002 or Y003 in LX3V (2N firmware), LX3VP, LX3VE and LX3VM can be used for the pulse output. Only Y000 or Y001 in LX3V (1S firmware) can be used for the pulse output.
- D_2: The rotation direction signal. If D_2= OFF, rotation = negative, if D_2= ON, rotation = positive.

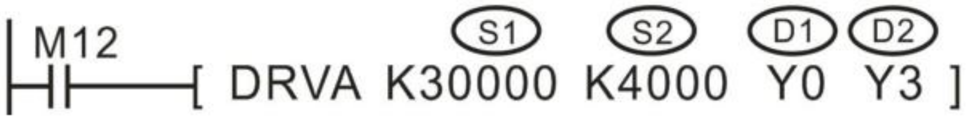
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√
D_1		√														
D_2		√	√	√												

If the contents of an operand are changed while the instruction is executed, it is not reflected on the operation. The new contents become effective when the instruction is next driven.

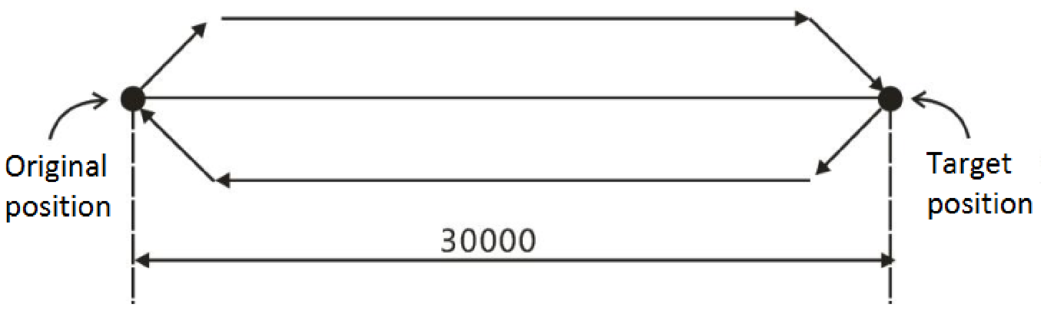
If the instruction drive contact turns off while the instruction is being executed, the machine decelerates and stops. At this time the execution complete flag M8029 does not turn ON.

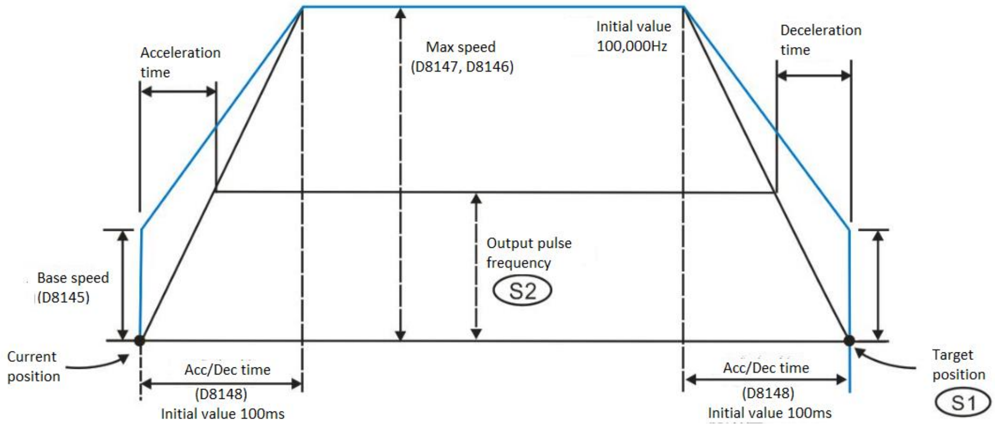
Once the instruction drive contact is off, re-drive of the instruction is not possible while the pulse output flag (Y000: [M8147], Y001: [M8148]) is ON.

2) Program example



The instruction controls the target to run from the specified origin position to the target position;





During the pulse output, the frequency will increase or decrease according to the predetermined value.

The minimum value of output pulse frequency which can be actually used is determined by the following equation.

$$\sqrt{(\mathrm{Maxspeed}[\mathrm{D8147}, \mathrm{D8146}]/(2 * (\mathrm{Acc\ or\ Dec\ time\ }[\mathrm{D8148}]\mathrm{ms}/1000))}$$

Even if the assigned value is lower than the above calculated result, the frequency to be exported will still be the calculated value. The frequencies in the initial stage of acceleration and in the final section of deceleration must not be lower than the above calculated result.

During the instruction execution, the involved system variables are as follows:

- [D8145]: Base speed when executing DRVI and DRVA instructions. During the operation of stepping motor, the stepping motor's resonance region and automatic start frequency must be considered when setting up the speed. Setting Range: below 1/10 of the maximum speed (D8147, D8146). When the setting surpasses the indicated range, the operating speed will automatically decelerate to the 1/10 of the highest speed.
- [D8147 (high byte), D8146 (low byte)]: Maximum speed when executing DRVI and DRVA instructions. The assigned output pulse frequency must be lower than the maximum speed. Setting range: 10 ~100,000(Hz)
- [D8148]: acceleration and deceleration time when executing FNC158 (DRVI) and FNC159 (DRVA) instructions. Acceleration/Deceleration time means the time required in order to reach the maximum speed (D8147, D8146). The output pulse frequency is lower than the maximum speed (D8147, D8146), the actual acceleration/deceleration time will reduce. Setting range: 50 ~ 5,000 (ms)
- [M8145]: Y000 pulse output stopping (immediate stopping)
- [M8146]: Y001 pulse output stopping (immediate stopping)
- [M8152]: Y002 pulse output stopping (immediate stopping)
- [M8153]: Y003 pulse output stopping (immediate stopping)
- [M8147]: Y000 pulse output monitoring (BUSY/READY)
- [M8148]: Y001 pulse output monitoring (BUSY/READY)
- [M8149]: Y002 pulse output monitoring (BUSY/READY)
- [M8150]: Y003 pulse output monitoring (BUSY/READY)

3) Note for use:

- Position instruction (ZRN/PLSV/DRVI/DRVA) can be reused in the program, but do not output to the same port;
- If the drive power flow for an instruction turns OFF and ON again, if can only be driven after one operation cycle when status bit (Y000: [M8147], Y001: [M8148], Y0002: [M8149], Y003: [M8150]) turns OFF.
- When positioning instruction is driven again, there should be at least one cycle of OFF time. If the re-drive is implemented in the time less than above condition, there will be calculation error when firstly implementing calculation instruction.

5.2.12 External Device SER instruction

RS instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
RS	Serial data transfer	16	No	RS S m D n	9

RS is a transceiver instruction that automatically sends the data stored in the specific register to the serial port sequentially and stores the data received by serial port in the specific area. It is equivalent to directly access the communication buffer.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
m					√	√								√		
D														√		
n					√	√								√		

COM2 (communication port): it uses RS485, it supports program protocol (PLC protocol), MODBUS protocol (MODBUS-RTU slave, MODBUS-RTU master, MODBUS-ASCII slave, MODBUS-ASCII master), N:N network protocol (so far, only available in LX3VP series PLC). The communication protocol is set by D8126, the communication parameters are set by D8120, the detailed as below.

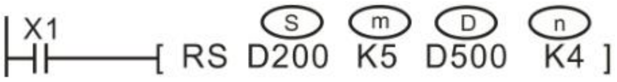
Communication setting for COM2		
Protocol	The value of D8126	Communication parameters
HMI monitor protocol (PLC protocol)	0x01	Set by D8120
MODBUS-RTU slave	0x02	Set by D8120
MODBUS-ASCII slave	0x03	Set by D8120
User-defined protocol	0x10	Set by D8120
MODBUS-RTU master	0x20	Set by D8120
MODBUS-ASCII master	0x30	Set by D8120

Item	Parameter	Bit value of D8120							
		b7	b6	b5	b4	b3	b2	b1	b0
Baud rate (Bps)	115200	1	1	0	0	-	-	-	-
	57600	1	0	1	1	-	-	-	-
	38400	1	0	1	0	-	-	-	-
	19200	1	0	0	1	-	-	-	-
	9600	1	0	0	0	-	-	-	-
	4800	0	1	1	1	-	-	-	-
Stop bit	1 bit	-	-	-	-	0	-	-	-
	2 bit	-	-	-	-	1	-	-	-
Parity	None	-	-	-	-	-	0	0	-
	Odd	-	-	-	-	-	0	1	-
	Even	-	-	-	-	-	1	1	-
Data bit	7 bit	-	-	-	-	-	-	-	0
	8 bit	-	-	-	-	-	-	-	1
Example: the communication format is 9600.1.8.None, b7b6b5b4=1000, b3=0, b2b1=00, b 0=1. D8120=81H ((10000001)2=81H, 81H means hexadecimal number)									

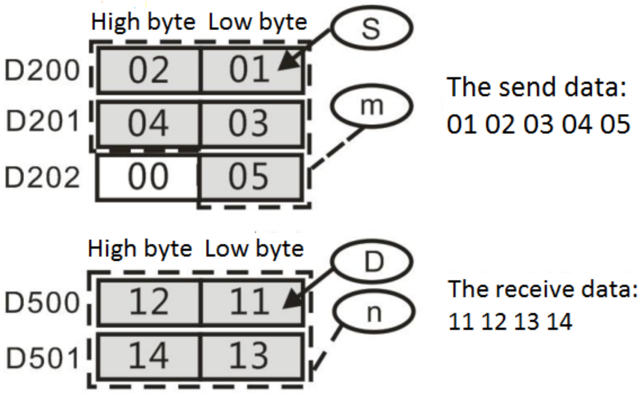
2) RS (user-defined Protocol) Instruction Description

- S: the head address of the register where the to be sent data stored in
- m: the length of the to be sent data (byte), 0 to 256.
- D: the head address of the register where the receive data stored in
- n: the length of the receive data(byte),0 to 256

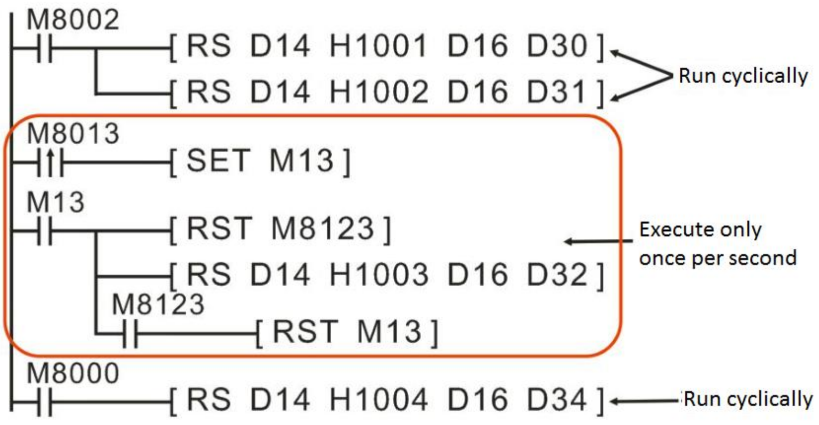
Example



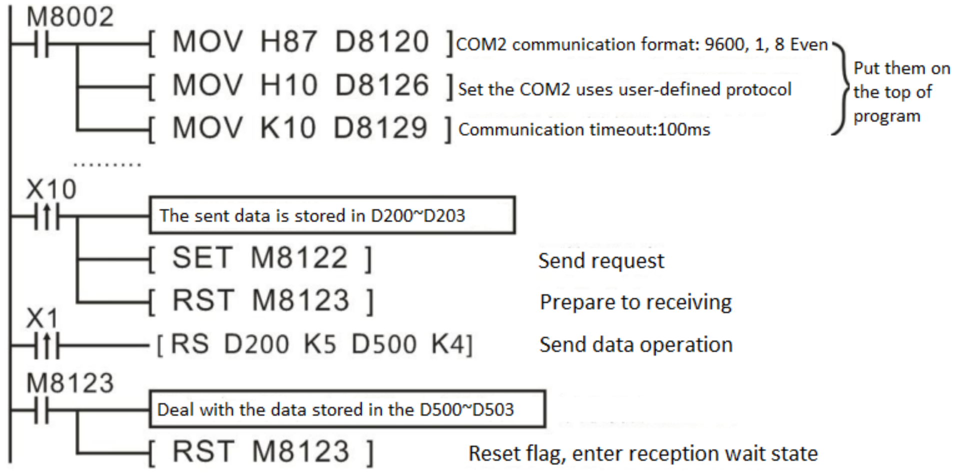
When X1 is ON, the receive data and the sand data is shown as below.



The RS (MODBUS mode) instruction automatically sets the M8123 once every time a transmit data is received and the acknowledge operation is received. Using this flag, it is possible to determine whether the RS instruction has been executed.



In actual programming, you need to do some preparation for serial communication and configuration, such as baud rate, check bit, timeout judgment condition, protocol etc. The same example, a relatively complete set of RS communication procedures are as follows.

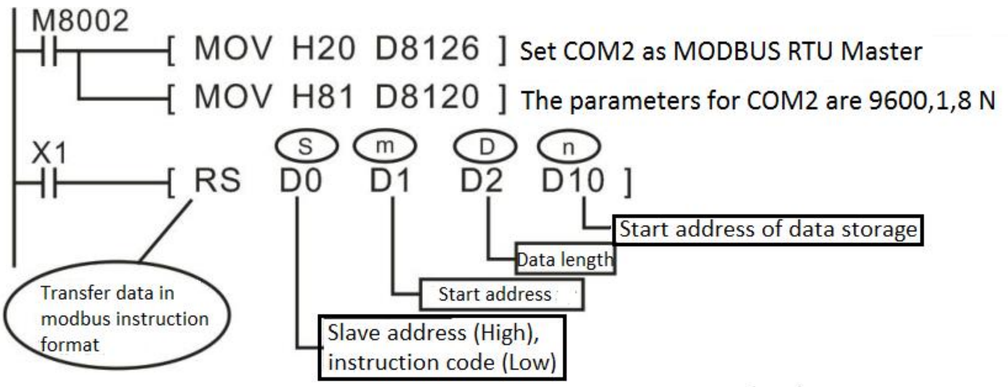

M8002
[MOV H87 D8120]
COM2 communication format: 9600, 1, 8 Even
[MOV H10 D8126]
Set the COM2 uses user-defined protocol
[MOV K10 D8129]
Communication timeout:100ms
Put them on the top of program
....
X10
The sent data is stored in D200~D203
[SET M8122]
Send request
[RST M8123]
Prepare to receiving
X1
[RS D200 K5 D500 K4]
Send data operation
M8123
Deal with the data stored in the D500~D503
[RST M8123]
Reset flag, enter reception wait state

3) RS (MODBUS Protocol) Instruction Description

The definitions of each operand in the RS (MODBUS mode) instruction are different from those of a standard RS instruction (user-defined protocol).

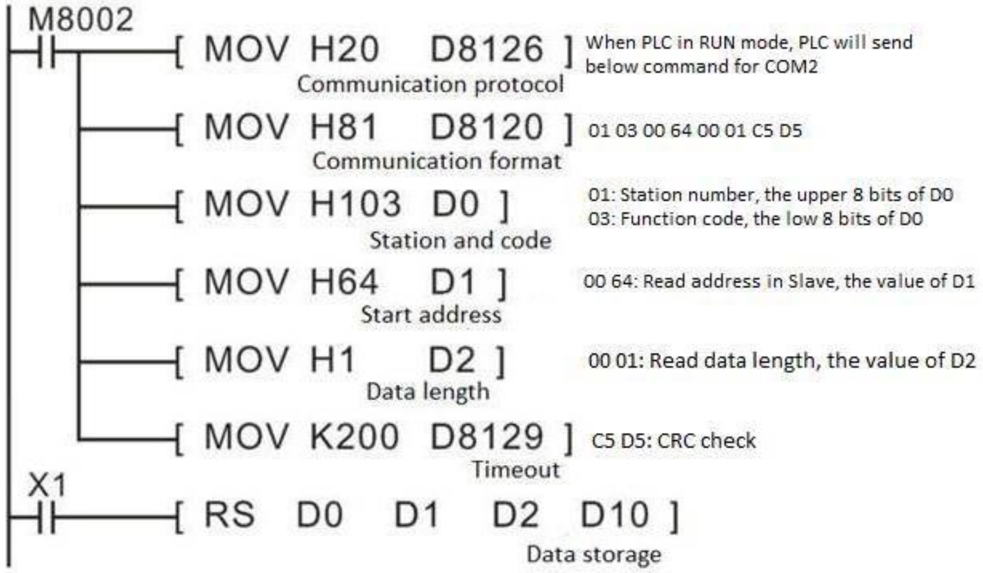
- S: Slave address (high byte), communication command (low byte, defined by MODBUS protocol);
- m: Start address of accessing slave;
- D: Data length, unit: word;

- n: Start address of data storage, the take up length of the subsequent address defined by D;


M8002
MOV H20 D8126
Set COM2 as MODBUS RTU Master
MOV H81 D8120
The parameters for COM2 are 9600,1,8 N
X1
RS D0 D1 D2 D10
S
m
D
n
Start address of data storage
Data length
Start address
Slave address (High),
instruction code (Low)
Transfer data in modbus instruction format

4) Program example

The PLC is set to MODBUS-RTU master mode, it reads data from address 100 of Slave 1, and read the data stored in D10.



RS2 instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
RS2	Serial data transfer 2	16	No	RS S m D n n1	9

This instruction is mainly used for serial data transfer instruction in BD board module. RS2 is a transceiver instruction that automatically sends the data stored in the specific register to the serial port sequentially and stores the data received by serial port in the specific area. It is equivalent to directly access the communication buffer.

The RS2 instruction is used to configure the communication protocol according to the CPAVL instruction. For details, refer to the LX3V-2RS485-BD User's Manual, LX3V-ETH-BD User's Manual or the CPAVL Instruction Manual.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
m					√	√								√		
D														√		
n					√	√								√		
n1					√	√								√		

2) In LX3V-2RS485-BD module

- User-defined protocol

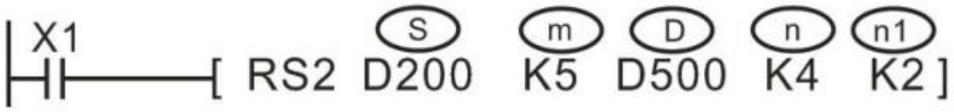
S: Starting address of transmitted data.

m: Length of transmitted data, the range is 0~256

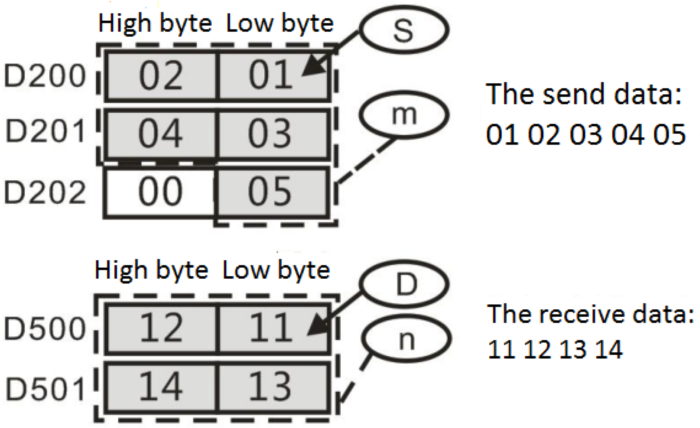
D: Starting address for storage data.

n: Length of received data, the range is 0~256

n1: Serial port Number, 2 means using COM2, 3 means using COM3, 4 means using COM4, 5 means using COM5, 6 means using COM6; Program can write multiple RS2 instructions, but only one RS2 instruction can be triggered at the same time.



In this example, n1 is set K2, so the RS2 instruction is used in COM2. When X1 is triggered program will transfer data as below shows.



- MODBUS protocol

The definitions of each operand in the RS2 (MODBUS mode) instruction are different from those of a standard RS instruction (user-defined protocol).

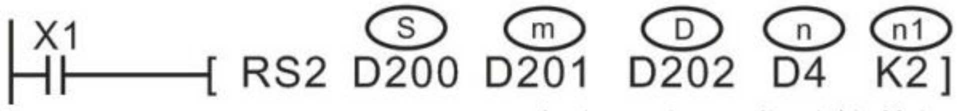
S: Slave station address (high byte), communicational command (low byte, define by MODBUS protocol);

M: Register start address of call on slave station;

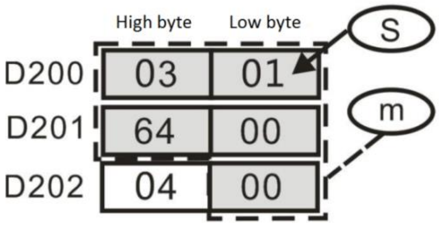
D: Data length will be read or write, units is word;

N: Memory units original address for read or write data, engross continuous address units, length decided by D;

n1: Serial port Number, 2 means using COM2, 3 means using COM3, 4 means using COM4, 5 means using COM5, 6 means using COM6; Program can write multiple RS2 instructions, but only one RS2 instruction can be triggered at the same time.



In this example, n1 is set K2, so the RS2 instruction is used in COM2. When X1 is triggered program will transfer data as below shows.



According to example the sending data is 01 03 00 64 00 04 + CRC.

Description: Read data of four addresses from addresses 100 to 103 in the slave whose station number is 1 and store the read data in four addresses from D202 to D205.

3) In LX3V-ETH-BD module

- MODBUS TCP protocol

S: The address of slave (high byte) and communication command (low byte, defined by MODBUS protocol);

m: The starting address number of the slave

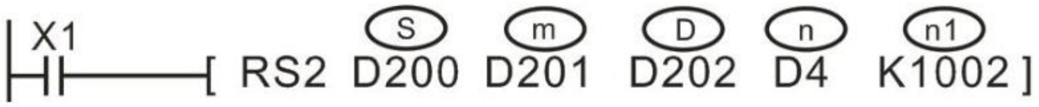
D: The length of the data (read or writes), the unit is word. (The specific setting is shown in the following table)

Function code	Length	Length (HEX)
Write coils	1968	0x7B0
Read coils	2000	0x7D0
Write registers	123	0x7B
Read registers	125	0x7D

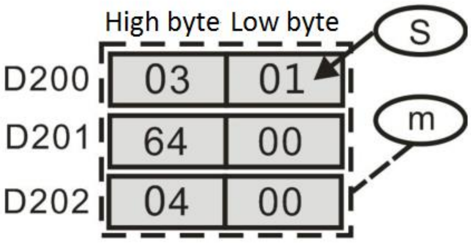
n: The starting address of the storage unit for reading or writing data, occupying the subsequent address unit, and the length is determined by the D

n1: The connection number corresponding to the Ethernet port connection number (specific settings is shown as the following table)

Ethernet port 1		Connection number	Ethernet port 2		Connection number
RS2 instruction	Connection 1	1000	RS2 instruction	Connection 1	1100
	Connection 2	1001		Connection 2	1101
	Connection 3	1002		Connection 3	1102
	Connection 4	1003		Connection 4	1103
	Connection 5	1004		Connection 5	1104
	Connection 6	1005		Connection 6	1105
	Connection 7	1006		Connection 7	1106
	Connection 8	1007		Connection 8	1107



In this example n1 is set as K1002, then RS2 is configured for Ethernet port 1, connection 3. When x1 is ON, the data is shown as below.



	High byte	Low byte
D200	03	01
D201	64	00
D202	04	00

The sending data is 01 02 03 00 64 00 04 + CRC check

Description: Read the data of the slave ranges from 100 to 103, and transfer the data to D202, D203, D204, and D205.

RSLIST instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
RSLIST	Formulated communication instructions	16	No	RSLIST S_1 S_2 m1	9

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2														√		
m1			√													

Only LX3VP series and above plc (advanced series) support RSLIST instruction, and major version number and this version number of LX3VP series plc must be "25103" and "16001" and above edition, while major version number and this version number of LX3VPE series plc must be "25201" and "16001" and above edition.

This instruction is the one tabulating RS instruction, which replaces communication protocol by changing D8126 (communication protocol). Most of its functions are the same as the ones of RS instruction, but with simplified cumbersome engineering writing, in which, parameters of each transmission command are directly set by table.

S_1 is the starting device address controlled by table, with value range of D0~D7999; The addresses of starting device between S_1~S_4+4+n*6 cannot be occupied by other instructions. (As far as possible, store data in power-down save (latched) (D200-D7999) to avoid data loss).

Communication table of RSLIST instruction (MODBUS protocol)		
Address	Name	Function description
S_1+0	Header	Header = 50h, correct MODBUS and MODBUSASC protocol communication form. (No modification)
S_1+1	Communication command	Communication command number: A

	number		transmission needs to use six devices to describe, that is, six devices describe a data transmission. (No modification)
S_1+2	Check bit of header and communication command number		Check header and communication command number (No modification)
S_1+3	Error replication number (retry times) (for all commands)		Error replication number (retry times) (0-255)
S_1+4	Slave station number of current communication	Function code	Station number: 0 ~ 255 (0 means that the master station broadcast to all slave station, while slave station does not respond to the received information.) Function code: please check the table below.
S_1+5	Start address of Slave data		Word is valid. Define start address of Slave.
S_1+6	Data length		Word is valid, range: 1~126 (word data), 1 ~ 2039 (bit data).
S_1+7	Master stores data starting device		Word is valid. Define Master to receive the start address of data.
S_1+8	Sending control bit		Sending control bit: no running temporarily as long as it's 0. (when it is not 0, no execution)
S_1+9	Single error replication number(retry times)		Single error replication number (retry times)(0-255)
S_1+10	Slave station number	Function code	Description of second data transmission
S_1+11	Start address of Slave data		
S_1+12	Data length		
S_1+13	Master stores data starting device		
S_1+14	Sending control bit		
S_1+15	Single error replication number (Retry times)		
…	…		…
S_1+4+n*6	Save		N is the total number of data transmission commands

Communication table of RSLIST instruction (User-defined protocol)		
Address	**Name**	**Function description**
S_1+0	Header	Header = 51h, correct MODBUS and MODBUSASC protocol communication form. (No modification)
S_1+1	Communication command number	Communication command number: A transmission needs to use six devices to describe, that is, six devices describe a data transmission. (No modification)
S_1+2	Check bit of header and communication command number	Check header and communication command number (No modification)
S_1+3	Error replication number (retry times) (for all commands)	Error replication number (retry times) (0-255)
S_1+4	Master sends data starting device	Receive the start address of data.
S_1+5	Data length of Master	Word is valid. PLC determines data length according to cache block. (LX3VP:0~528)
S_1+6	Master receives data starting device	Word is valid. Define Master to send the start address of data.
S_1+7	Data length of slave station	Word is valid. PLC determines data length according to cache block. (LX3VP:0~528)
S_1+8	Sending control bit	Sending control bit: no running temporarily as long as it's 0. (when it is not 0, no execution)
S_1+9	Single error replication number (Retry times)	Single error replication number(Retry times) (0-255)
S_1+10	Data length of Master	Description of second data transmission.
S_1+11	Data length of Slave	
S_1+12	Master sends data starting device	
S_1+13	Master receives data starting device	
S_1+14	Sending control bit	
S_1+15	Single error replication number (Retry time)	

…	…	…
S_1+4+n*6	Save	N is the total number of data transmission commands

S_2 is the starting device address of table cache, with value range of D0~D7999; the addresses of cache starting device between S_2~S_2+12 cannot be occupied by other instructions.

Note: any random data in the above operation-forbidden area will lead to communication anomalies.

Address	High byte	Low byte	Operated
S_2+0	Operation serial number: indicate which command is operating at present.		No
S_2+1	Result code: = 0, normal; = Other value, abnormal		No
S_2+2	Slave station device number	Function code	No
	User-defined Protocol: master station sends starting device		No
S_2+3	Start address of Master device (User-defined protocol: Master receives the start address of data)		No
S_2+4	Received or sent data size (User-defined protocol: data size received by master		No
S_2+5	Error replication number(Retry times) of current command		No
S_2+6	Error times record		No
S_2+7	Bit0 = 1, Port has been occupied, this command waits for data transmission rights; Bit4, communication transmission output indication is represented by "M1"; Bit5, communication error output indication is represented by "M1 + 1"; Bit6, communication completion output indication is represented by "M1 + 2".		No
S_2+8	Select which command to implement		Yes
S_2+9	Select which command to open and close: which command		Yes
S_2+10	Select which command to open and close: 0: none, 1: close, 2: open;		Yes

S_2+11	(System occupancy)	No

Name	Numerical value	Function description
Function code	01H	Read the state of consecutive multiple single-points from Slave
	03H	Read data of consecutive multiple registers from Slave
	05H	Write the state of individual single-point into Slave
	06H	Write data of single register into Slave
	0FH	Write the state of consecutive multiple single-points into Slave
	10H	Write data of consecutive multiple registers into Slave
Result code (error code)	0x00	successful communication transaction
	0x01	frame error
	0x02	illegal communication table (header error)
	0x04	Data length error (the position read or written by command is beyond range of device size)
	0x05	the set read and write length range is beyond device range (starting device plus the length is beyond the range of D0-D7999)
	0x06	Function code error (incorrect function code or not supporting this function code).
	0x07	Slave station number error
	0x08	Slave-address error
	0x09	No response in Slave (abnormal time-out)
	0x0A	Abnormal communications (receive erroneous data or slave station responses to error message).
	0x0B	selected commands exceed maximum number of commands
	0x0F	Skip this command (the sending control bit of this command is not 0)

m1 is the start address of communication flag, with value range of M0 ~ M3068; (m1~m1+2) cannot be used by other instructions.

Address	Function

m1+0	Transmission flag
m1+1	Error flag
m1+2	Completion flag

Other related settings are listed below:

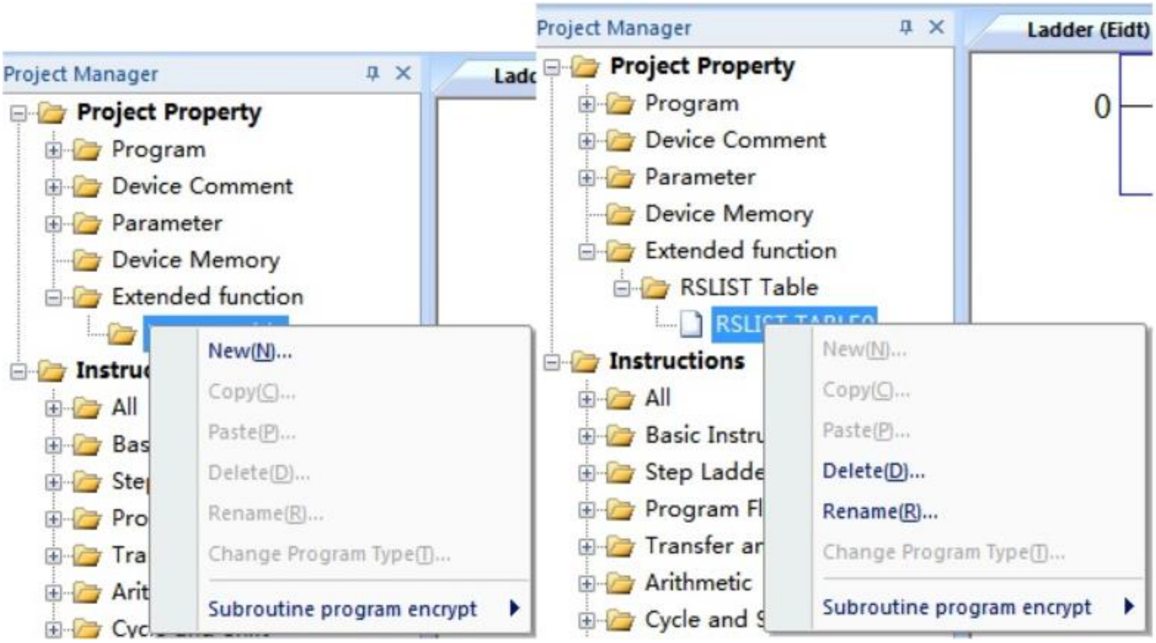
Item	Parameter	Bit value of D8120							
		b7	b6	b5	b4	b3	b2	b1	b0
Baud rate (Bps)	115200	1	1	0	0	-	-	-	-
	57600	1	0	1	1	-	-	-	-
	38400	1	0	1	0	-	-	-	-
	19200	1	0	0	1	-	-	-	-
	9600	1	0	0	0	-	-	-	-
	4800	0	1	1	1	-	-	-	-
Stop bit	1 bit	-	-	-	-	0	-	-	-
	2 bit	-	-	-	-	1	-	-	-
Parity	None	-	-	-	-	-	0	0	-
	Odd	-	-	-	-	-	0	1	-
	Even	-	-	-	-	-	1	1	-
Data bit	7 bit	-	-	-	-	-	-	-	0
	8 bit	-	-	-	-	-	-	-	1
Example: the communication format is 9600.1.8.None, b7b6b5b4=1000, b3=0, b2b1=00, b 0=1. D8120=81H ((10000001)2=81H, 81H means hexadecimal number)									

Address	Description
D8120	Com2 port communication format, interface configuration settings (see the above table for details).
D8122	User-defined protocol: send remaining data (for RS instruction only); MODBUS protocol: command sending interval, 0 = 5ms. Unit 0.1ms
D8124	starting character STX (only for RS user-defined protocol)
D8125	terminating character ETX (only for RS user-defined protocol)
D8126	Communication protocol settings, interface configuration settings
D8129	MODBUS: determine the time of communication timeout, interface configuration settings, the default is 10 (10ms)

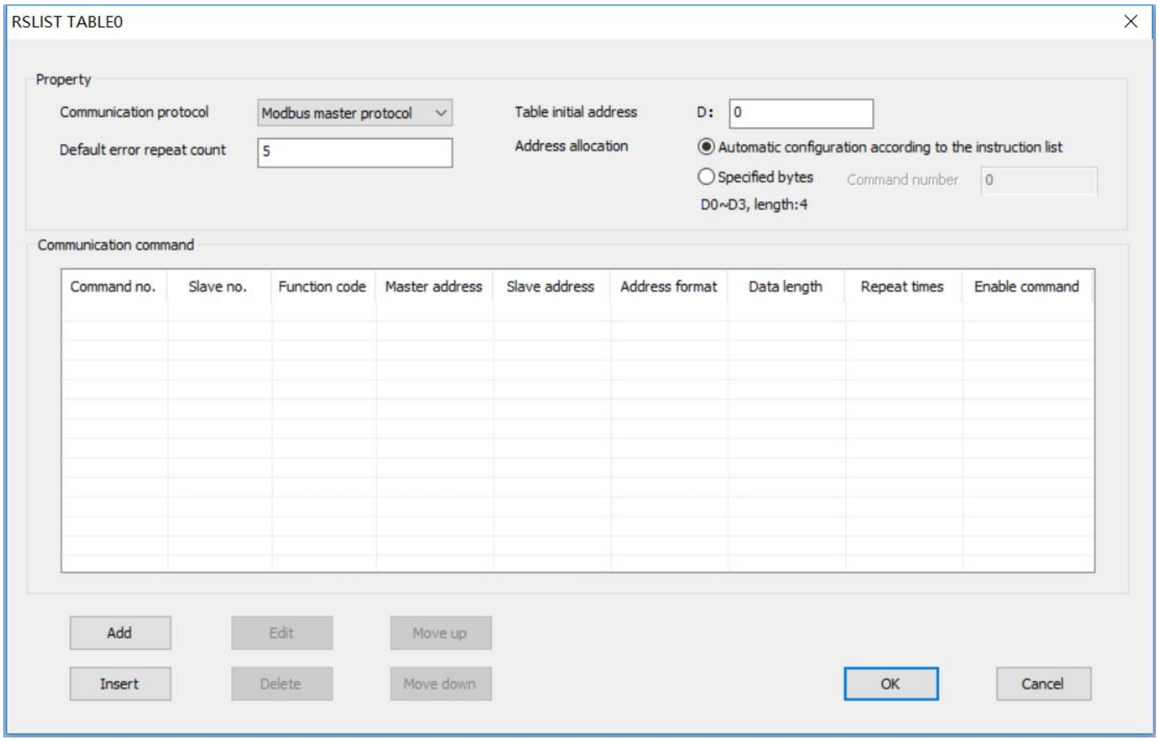
	RS user-defined protocol: inter-character timeout, interface configuration settings, the default is 10 (10ms)
D8172	First character timeout, interface configuration settings, the default is 10 (10ms) First character timeout is not calculated when M8172 is 0. (only for RS user-defined protocol)

2) Create new RSLIST table

Right click (Project Manager -> Project Property -> Extended Function -> RSLIST Table) to create or edit table, as shown below:



- **Table setting interface as below**
- **Communication protocol:** should be consistent with the configuration of communication protocol control address (D8126, etc.) (Modbus Master / RS user-defined protocol).
- **Default error repeat count:** use the replication number (retry times)of a single error when it is not 0; use the error replication number (retry times) set in header when it is 0; default 3 times when both are 0. The set replication number includes number of times for the first run, that is, repeat errors only for two times and then turn to the next when the set replication number is 2.
- **Table initial address:** should be consistent with in the corresponding RSLIST instruction.
- **Address allocation:** Table space can be automatically configured or selected as fixed length (4 + n * 6), where n is the number of communication commands.



RSLIST TABLE0

Property

Communication protocol: Modbus master protocol

Default error repeat count: 5

Table initial address D: 0

Address allocation: ● Automatic configuration according to the instruction list ○ Specified bytes Command number 0

D0~D3, length:4

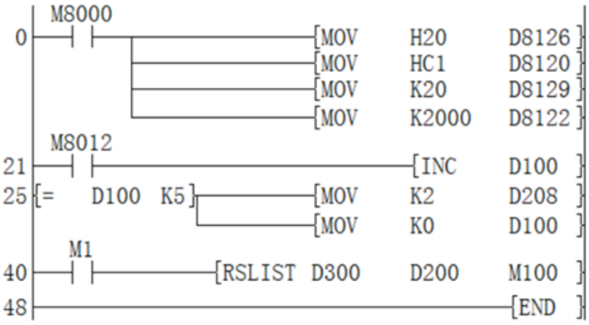
Communication command

Command no.	Slave no.	Function code	Master address	Slave address	Address format	Data length	Repeat times	Enable command

Note: Number of table commands should be less than 255. And do address planning to avoid data confusion caused by repeatedly occupied addresses.

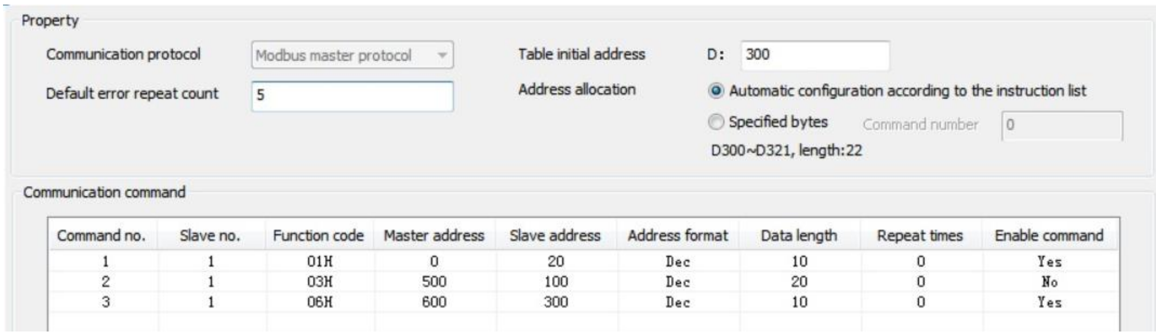
3) Modbus Protocol Configuration

- **Ladder configuration**



```
 0  M8000  MOV  H20    D8126   Set "communincation protocol" as "modbus rtu master"
           MOV  HC1    D8120   Set "communication parameter" as "115200, 1, 8, none"
           MOV  K20    D8129   Set command sending interval
           MOV  K2000  D8122   Set communication timeout
21  M8012  INC         D100    Set data transmission interval
25  [= D100 K5]  MOV  K2  D208  D208 controls command address for RSLIST table instruction (The
                 MOV  K0  D100  detailed please refer to S2 address assignment list)
40  M1     RSLIST D300  D200  M100  Start modbus protocol table in RSLIST. Start address of protocol
48                              END   control is D300,  cache control is D200, state indication is M100
```

- **Table configuration**



Property

Communication protocol: Modbus master protocol

Default error repeat count: 5

Table initial address D: 300

Address allocation: ● Automatic configuration according to the instruction list ○ Specified bytes Command number 0

D300~D321, length:22

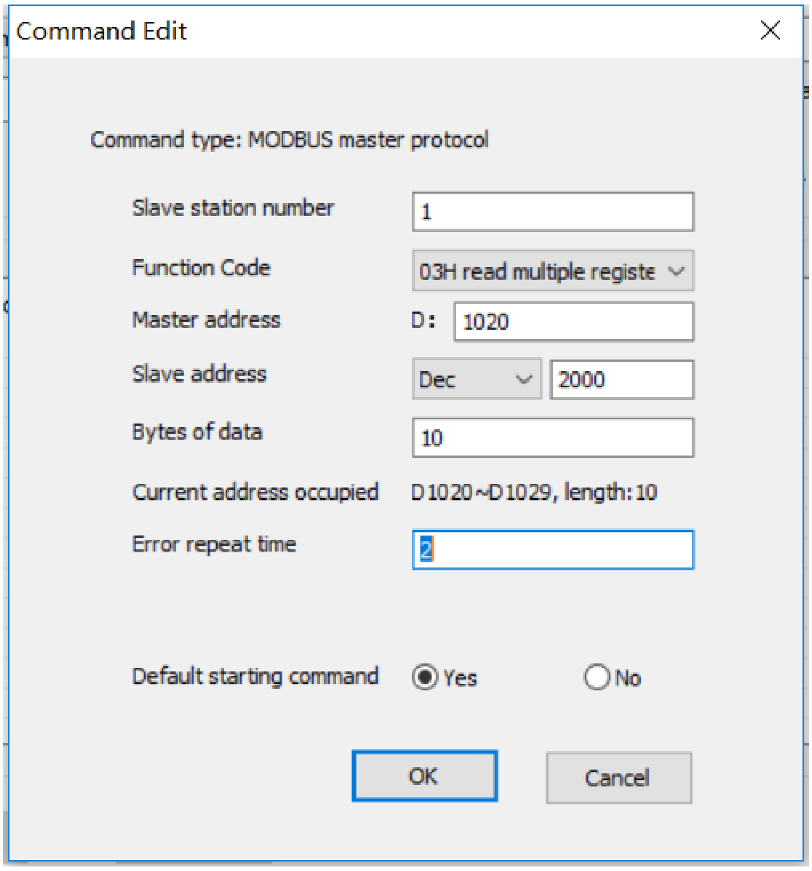
Communication command

Command no.	Slave no.	Function code	Master address	Slave address	Address format	Data length	Repeat times	Enable command
1	1	01H	0	20	Dec	10	0	Yes
2	1	03H	500	100	Dec	20	0	No
3	1	06H	600	300	Dec	10	0	Yes

Table start address is D300, which corresponds to the RSLIST instruction in ladder.

When M1 = ON, RSLIST instruction starts execution. When "YES" in "whether to enable command" is selected as instruction in the table ", number 1 and 3 instructions in execution table are executed cyclically; when "No" is selected, execution is controlled by "S_2+8"=D208. As shown in the above ladder, D208 = 2 is triggered every 500ms, that is, number 2 instruction in the table is executed every 500ms. (Read data of 20 addresses starting from station 100 and store the read data in D500-D519 separately).

- **Communication command configuration**



Command type: MODBUS master protocol

Slave station number: 1

Function Code: 03H read multiple registe

Master address: D: 1020

Slave address: Dec 2000

Bytes of data: 10

Current address occupied: D1020~D1029, length:10

Error repeat time: 2

Default starting command: ● Yes ○ No

Slave station number: (limited between 0-255). Account 0 is used as broadcast and will not receive. See "Function Code List".

The data start address (D device) range in master station is (D0-D7999). Store relevant data in power-down save (D200-D7999) to avoid data loss. Store data start address in slave station.

Byte of data: (bit length in bits, word length in words), range 1 to 126 (word data), 1 to 2039 (bit data).

Error repeat time: 0 indicates that this error replication number (retry times) is the same as the one in table edit.

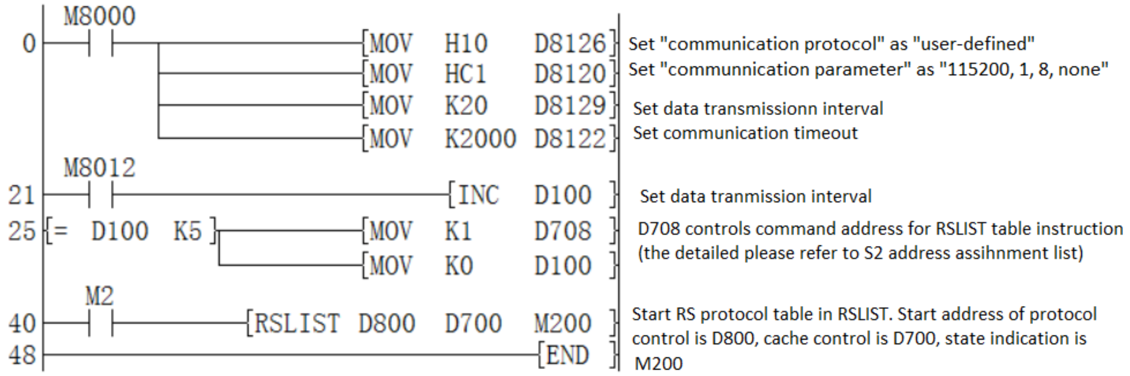
Default starting command:

Yes: execute cyclically when RSLIST is started;

No: execute only when +8 is in action.

4) RS User-defined Protocol

- **Ladder configuration**



Set "communication protocol" as "user-defined"

Set "communnication parameter" as "115200, 1, 8, none"

Set data transmissionn interval

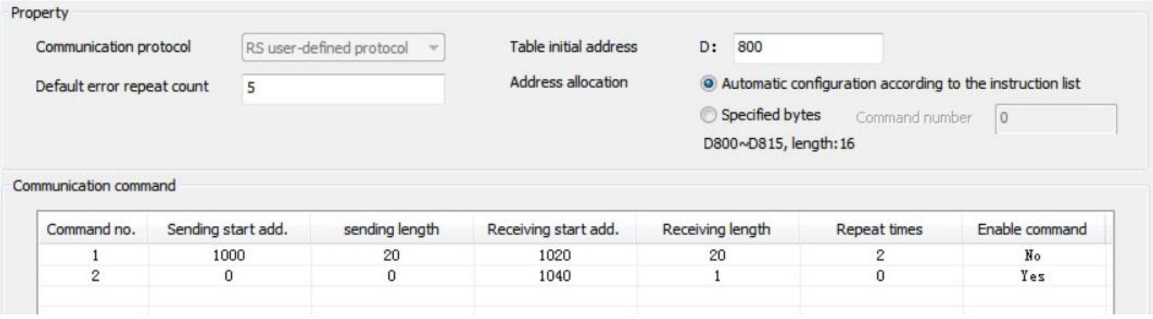
Set communication timeout

Set data tranmission interval

D708 controls command address for RSLIST table instruction (the detailed please refer to S2 address assihnment list)

Start RS protocol table in RSLIST. Start address of protocol control is D800, cache control is D700, state indication is M200

- **Table configuration**



Property

Communication protocol: RS user-defined protocol

Default error repeat count: 5

Table initial address D: 800

Address allocation: ● Automatic configuration according to the instruction list ○ Specified bytes Command number 0

D800~D815, length:16

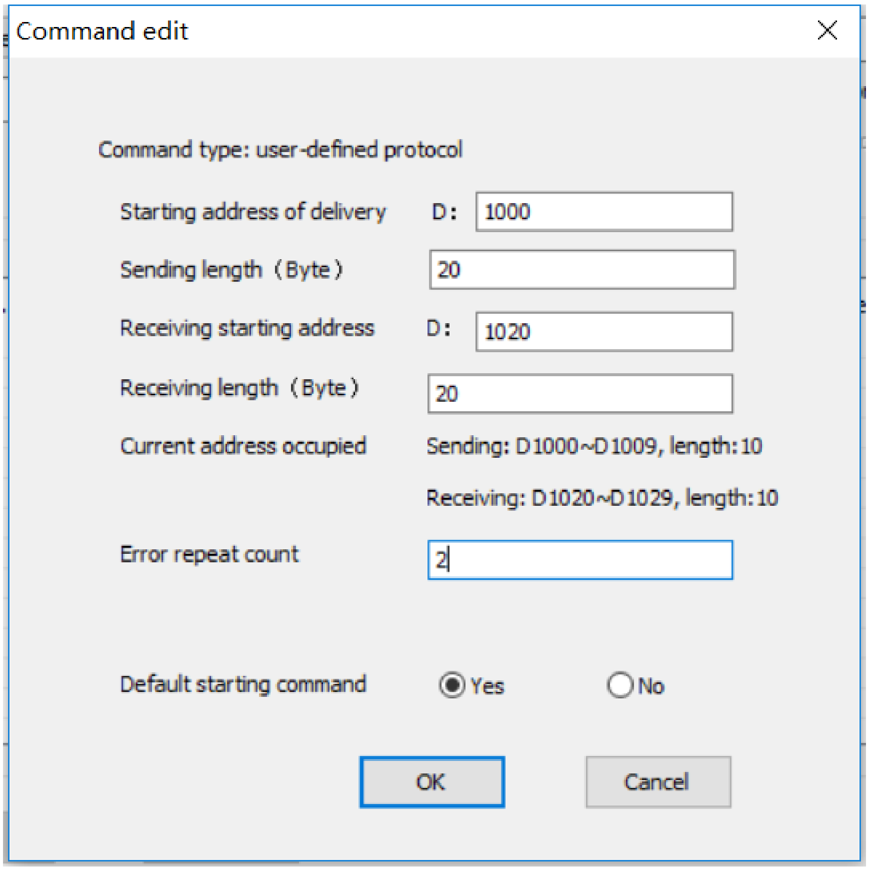
Communication command

Command no.	Sending start add.	sending length	Receiving start add.	Receiving length	Repeat times	Enable command
1	1000	20	1020	20	2	No
2	0	0	1040	1	0	Yes

The table start address is D800, which corresponds to the RSLIST instruction in ladder.

When M2=ON, RSLIST instruction starts execution. When "YES" in "whether to enable command" is selected as instruction in the table ", number 2 instructions in execution table are executed cyclically; when "No" is selected, execution is controlled by"S_2+8"=D208. As shown in the above ladder, D708 = 1 is triggered every 500ms, that is, number 1 instruction in the table is executed every 500ms. (Send data of the 20 addresses starting from D1000 to device and store the returned data in the 20 addressed starting from D1020.)

- **Communication command configuration**



Command type: user-defined protocol

Starting address of delivery D: 1000

Sending length（Byte） 20

Receiving starting address D: 1020

Receiving length（Byte） 20

Current address occupied Sending: D1000~D1009, length:10

Receiving: D1020~D1029, length:10

Error repeat count 2

Default starting command ● Yes ○ No

Starting address of delivery (D device) range is (D0-D7999). Store relevant data in power-down save (D200-D7999) to avoid data loss.

Sending length: the length here is in BYTE (range LX3VP: 0-528). No sending when the sent data length is 0.The receiving start address (D device) range is (D0-D7999). Store relevant data in power-down save (D200-D7999) to avoid data loss.

Receiving length: the length here is in BYTE (range LX3VP: 0-528). No receiving when the received data length is 0.

Error repeat count: 0 indicates that this error replication number (retry times)is the same as the one in table edit.

Default starting command:
Yes: execute cyclically when RSLIST is started;
No: execute only when +8 is in action;

CPAVL instruction (Ethernet port)

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
CPAVL	Communication port setting	16	No	CPAVL S D M	11

- S: The starting address of "D" device;
- D: The starting address of "M" device;
- M: Serial port Number, 0 means using COM0, 1 means using COM1, 2 means using COM2, 3 means using COM3, 4 means using COM4, 5 means using COM5, 6 means using COM6; Program can write multiple RS2 instructions, but only one RS2 instruction can be triggered at the same time.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D			√													
m					√	√										

Connection number description		
CPAVL instruction	Port	Connection number
	Ethernet port 1	1000
	Ethernet port 2	1100

Note:

Only need one CPAVL instruction to configure multiple connections. The RS instruction needs to be used for the corresponding connection.

2) Address definition

- M device (Bit)

Bit address	Description	Connection	Bit address	Description	Connection
D+0	Reserved	Connection 1 Configuration	D+10	Reserved	Connection2 Configuration
D+1	Instruction execution		D+11	Instruction execution	

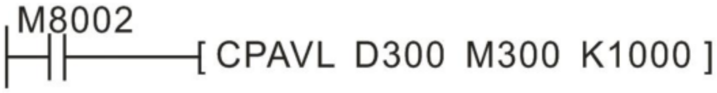
D+2	Instruction execution state		D+12	Instruction execution state	
D+3	Communication error flag		D+13	Communication error flag	
D+4	Reserved		D+14	Reserved	
D+5	Reserved		D+15	Reserved	
D+6	Reserved		D+16	Reserved	
D+7	Reserved		D+17	Reserved	
D+8	Reserved		D+18	Reserved	
D+9	Timeout flag		D+19	Timeout flag	
D+20	…	…	D+30	…	…

- D device (Word)

Word Address	Description D	Detailed description	Other instructions	Read and write features
0		Version number	BD Board parameter settings	R
1	BD Board IP Address	IP First 1 Section		R/W
2		IP First 2 Section		R/W
3		IP First 3 Section		R/W
4		IP First 4 Section		R/W
5	Port	0 default K502		R/W
6	Gateway	Gateway first 1 Section		R/W
7		Gateway first 2 Section		R/W
8		Gateway first 3 Section		R/W
9		Gateway first 4 Section		R/W
10	Subnet mask	Subnet mask first 1 Section		R/W
11		Subnet mask first 2 Section		R/W
12		Subnet mask first 3 Section		R/W
13		Subnet mask first 4 Section		R/W
14	MAC	MAC First 1 Section		R
15		MAC First 2 Section		R
16		MAC First 3 Section		R
17		MAC First 4 Section		R

18		MAC First 5 Section		R
19		MAC First6 Section		R
20	(Reserved)			R/W
21	(Reserved)			R/W
22	Number of connections	How many connections are required to set the number of connections		R/W
23	Protocol	Protocol	Connection 1 Configuration	R/W
24	Slaves IP	IP First 1 Section		R/W
25		IP First 2 Section		R/W
26		IP First 3 Section		R/W
27		IP First 4 Section		R/W
28	Port	0 default K502		R/W
29	(Reserved)			R/W
30	Instruction sending interval	Set the instruction sending interval. Unit: 0.1ms		R/W
31	(Reserved)			R/W
32	(Reserved)			R/W
33	(Reserved)			R/W
34	Timeout			R/W
35	Protocol	Communication protocol	Connection 2 Configuration	R/W
36	Slaves IP	IP First 1 Section		R/W
37		IP First 2 Section		R/W
38		IP First 3 Section		R/W
39		IP First 4 Section		R/W
40	Port	0 default K502		R/W
41	(Reserved)			R/W
42	(Reserved)			R/W
43	(Reserved)			R/W
44	(Reserved)			R/W
45	(Reserved)			R/W
46	Timeout			R/W
47	…	…	…	R/W

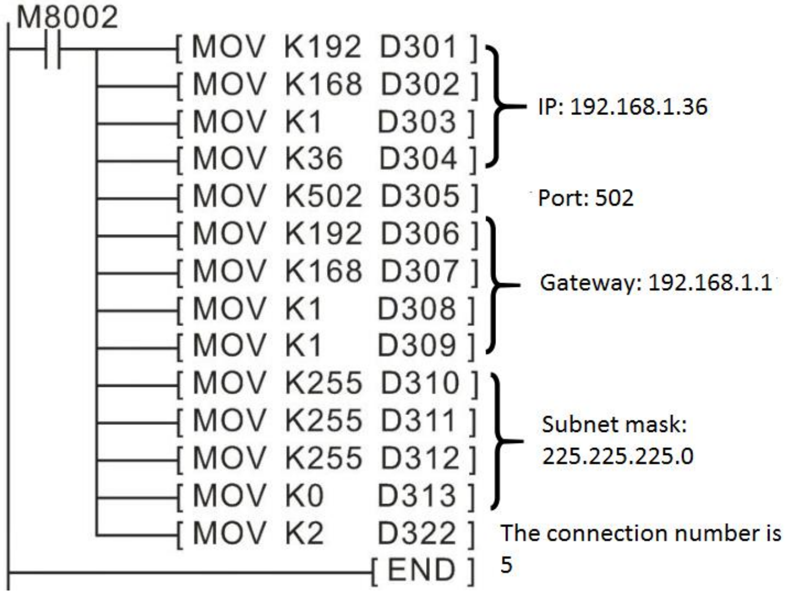
3) Program example



```
M8002
--| |--[ CPAVL D300 M300 K1000 ]
```

Use the Ethernet port 1, and the parameter table start from D300 and M300.

- The Ethernet parameter setting of LX3V-ETH-BD



```
M8002
--| |--[ MOV K192 D301 ]
       [ MOV K168 D302 ]
       [ MOV K1   D303 ]
       [ MOV K36  D304 ]
       [ MOV K502 D305 ]
       [ MOV K192 D306 ]
       [ MOV K168 D307 ]
       [ MOV K1   D308 ]
       [ MOV K1   D309 ]
       [ MOV K255 D310 ]
       [ MOV K255 D311 ]
       [ MOV K255 D312 ]
       [ MOV K0   D313 ]
       [ MOV K2   D322 ]
------------------[ END ]
```

IP: 192.168.1.36

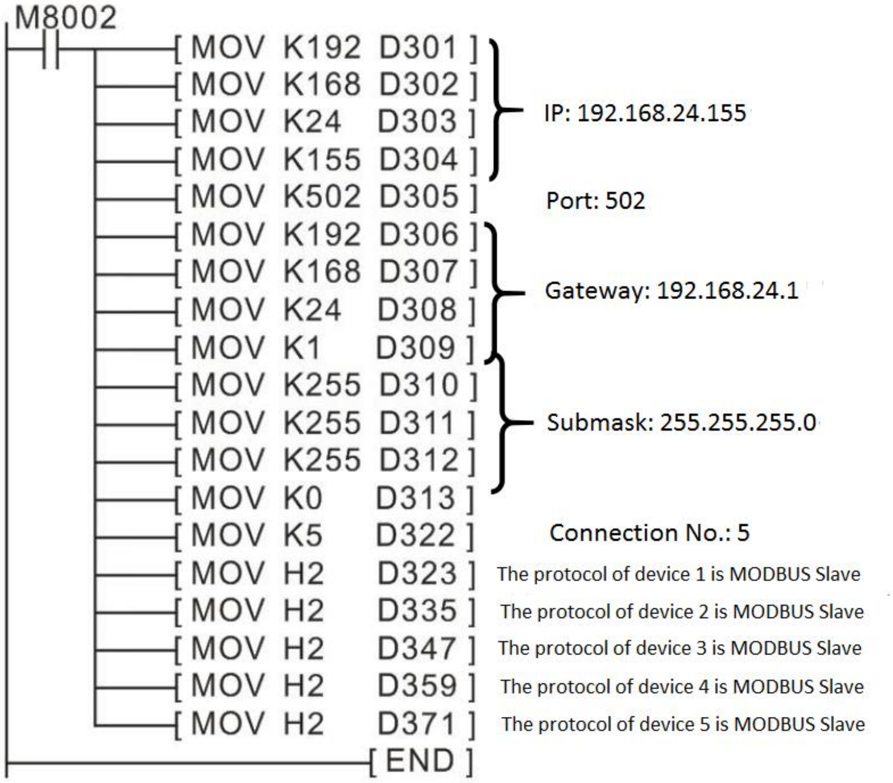
Port: 502

Gateway: 192.168.1.1

Subnet mask: 225.225.225.0

The connection number is 5

- MODBUS protocol setting



```
M8002
--| |--[ MOV K192 D301 ]
       [ MOV K168 D302 ]
       [ MOV K24  D303 ]
       [ MOV K155 D304 ]
       [ MOV K502 D305 ]
       [ MOV K192 D306 ]
       [ MOV K168 D307 ]
       [ MOV K24  D308 ]
       [ MOV K1   D309 ]
       [ MOV K255 D310 ]
       [ MOV K255 D311 ]
       [ MOV K255 D312 ]
       [ MOV K0   D313 ]
       [ MOV K5   D322 ]
       [ MOV H2   D323 ]
       [ MOV H2   D335 ]
       [ MOV H2   D347 ]
       [ MOV H2   D359 ]
       [ MOV H2   D371 ]
------------------[ END ]
```

IP: 192.168.24.155

Port: 502

Gateway: 192.168.24.1

Submask: 255.255.255.0

Connection No.: 5

The protocol of device 1 is MODBUS Slave

The protocol of device 2 is MODBUS Slave

The protocol of device 3 is MODBUS Slave

The protocol of device 4 is MODBUS Slave

The protocol of device 5 is MODBUS Slave

CPAVL instruction (Serial port)

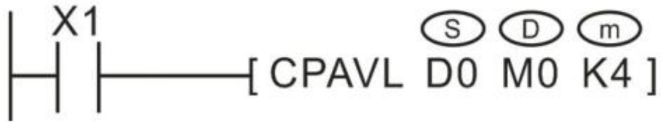
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
CPAVL	Communication port setting	16	No	CPAVL S D M	11

- S: The starting address of "D" device;
- D: The starting address of "M" device;
- M: Serial port Number, 0 means using COM0, 1 means using COM1, 2 means using COM2, 3 means using COM3, 4 means using COM4, 5 means using COM5, 6 means using COM6; Program can write multiple RS2 instructions, but only one RS2 instruction can be triggered at the same time.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D			√													
m					√	√										

2) Address definition



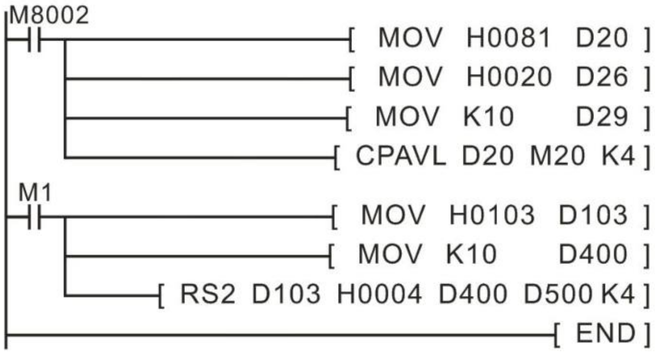
Setting the parameters of COM4 are in 20 consecutive addresses beginning of D0 and M0.

Bit address	Content	Word address	Content
D+0	Retention	S+0	Communication format, defined is 0
D+1	Sending(RS2)	S+1	Station number, defined is 0
D+2	Sending flag (RS2) Instruction state (MODBUS)	S+2	Remaining amount of data transmission(RS2)

			Interval of sending(MODBUS)
D+3	Receiving flag(RS2) Communication error flag (MODBUS)	S+3	The number of receiving data (RS2)
D+4	Receiving (RS2)	S+4	Starting code STX(RS2)
D+5	Retention	S+5	Ending code ETX(RS2)
D+6	Retention	S+6	Communication protocol
D+7	Retention	S+7	Retention
D+8	Retention	S+8	Retention
D+9	Timeout flag	S+9	Timeout, defined is 10 (10ms)
D+10~ D+19	Retention	S+10~ S+19	Retention

3) Program example

- MODBUS RTU Master



```
M8002
  ├─[ MOV   H0081  D20 ]
  ├─[ MOV   H0020  D26 ]
  ├─[ MOV   K10    D29 ]
  └─[ CPAVL D20 M20 K4 ]
M1
  ├─[ MOV   H0103  D103 ]
  ├─[ MOV   K10    D400 ]
  └─[ RS2 D103 H0004 D400 D500 K4 ]
─────────────────────────[ END ]
```

9600, 1, 8 none

Protocol is Modbus RTU master

Timeout

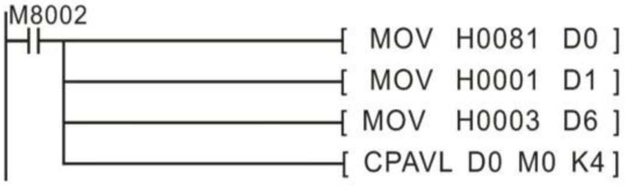
Set COM4 port, Starting address is D20& M20

Station number is 1, function code is 3

Number of data

Users can set MODBUS RTU master communication by RS2 instruction, as above example shows. RS2 is a communication instruction, which can send data in the specified register area to the serial port and store receive data to specified register. Equivalent to the user program directly access the communication cache, with the help of the user program processing of the communication cache, to achieve the communication. RS instruction only is available in COM2 port, but RS2 instruction can be available in COM3/ COM4/ COM5/ COM6 ports.

- MODBUS RTU Slave



9600, 1, 8 none

Station number is 1

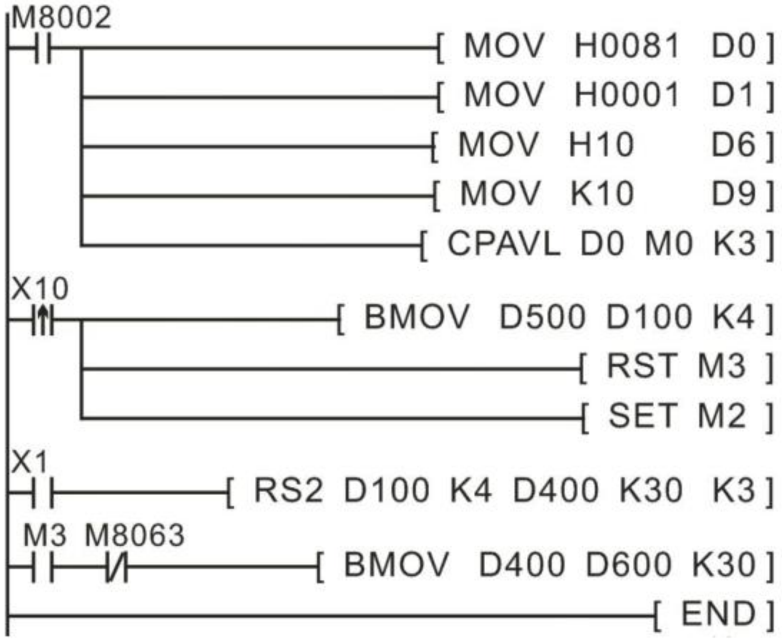
The protocol is Modbus ASCII slave

Setting the parameters from COM4 port, the starting address is D0& M0

When plc switches from stop to run state, PLC performs MODBUS RTU Slave communication; the function code and address mapping are consistent with COM2.

PLC internal address	MODBUS address	Number	Description
D0~D8255	0 (0)	8256	
T0~T255	0x F000 (61440)	256	
C0~C199	0x F400 (62464)	200	
C200~C255	0x F700 (63232)	56	32-bit register

- User-defined



9600, 1, 8 none

16bit mode

Set User- defined protocol

Timeout setting

Set COM3,start with D0&M0

Preparing sending data

Flag of reset

Reset sending request

Sending and receiving data

Dealing with receiving data

PRUN instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
PRUN	Transmission of Octal bits	16	No	PRUN S D	5
PRUNP		16	Yes		5
DPRUN		32	No		9
DPRUNP		32	Yes		9

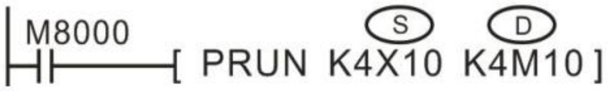
The instruction is used for coping the bit variables (the width unit is of octal) of the continuous addresses starting with S to the bit variable set starting with D in batch.

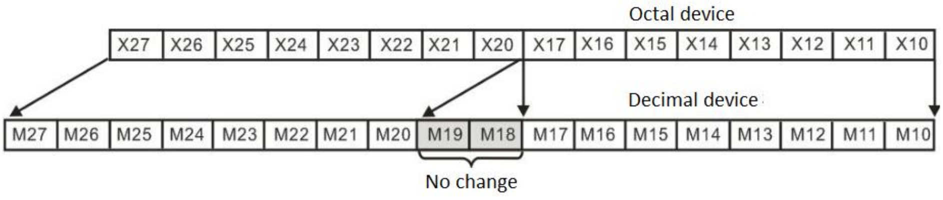
- S: The starting address of the bit variables to be copied, where the unit digit of the addresses must be 0, such as X10, M20;
- D: The starting address of the target bit variables, where the unit digit of the addresses must be 0, such as X10, M20;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√		√						
D									√	√						

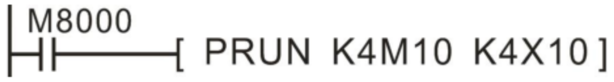
2) Program example

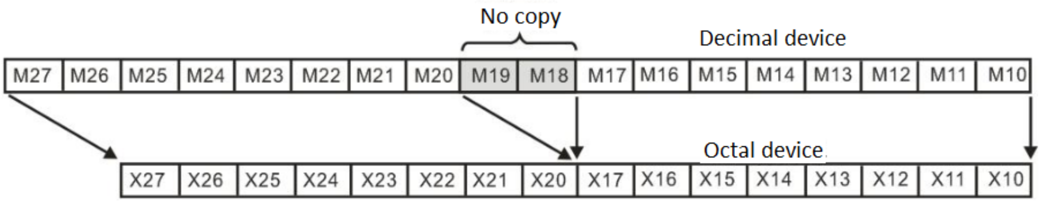
Example 1




Octal device
X27 X26 X25 X24 X23 X22 X21 X20 X17 X16 X15 X14 X13 X12 X11 X10
Decimal device
M27 M26 M25 M24 M23 M22 M21 M20 M19 M18 M17 M16 M15 M14 M13 M12 M11 M10
No change

Example 2




No copy
Decimal device
M27
M26
M25
M24
M23
M22
M21
M20
M19
M18
M17
M16
M15
M14
M13
M12
M11
M10
Octal device
X27
X26
X25
X24
X23
X22
X21
X20
X17
X16
X15
X14
X13
X12
X11
X10

ASCI instruction

1) Instruction description

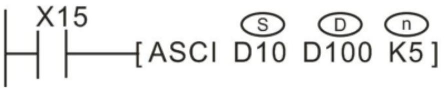
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
ASCI	Converts a data value from hexadecimal to ASCII	16	No	ASCI S D n	7
ASCIP		16	Yes		7

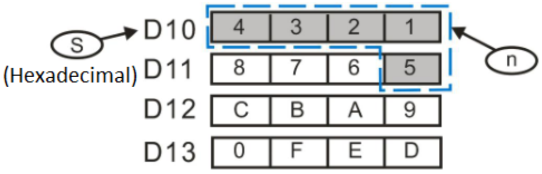
This instruction reads n hexadecimal data characters from head source address (S) and converts them in to the equivalent ASCII code.

- S: The source address;
- D: The store address;
- n: The data length;

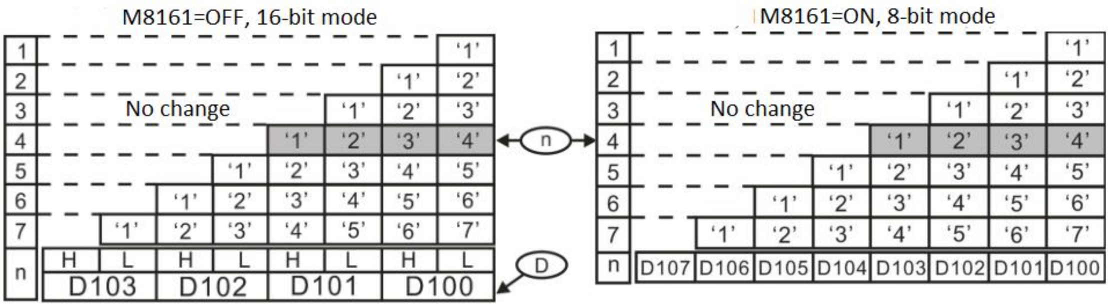
Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√		
n	Constant, n=1~256															

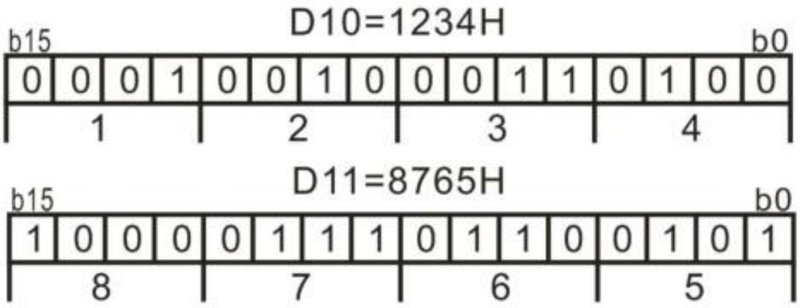
2) Program example



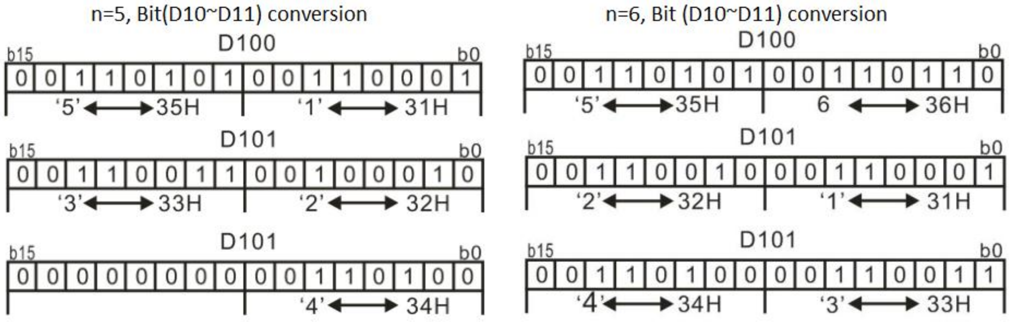


The M8161 flag determines the width mode of the target variable for calculation result storage. When M8161=OFF, it is 16bit mode, which means the higher byte and lower byte are saved respectively. When M8161=ON, it is 8bit mode, which means that only the lower byte is used to save result and the actual variable range length is longer.



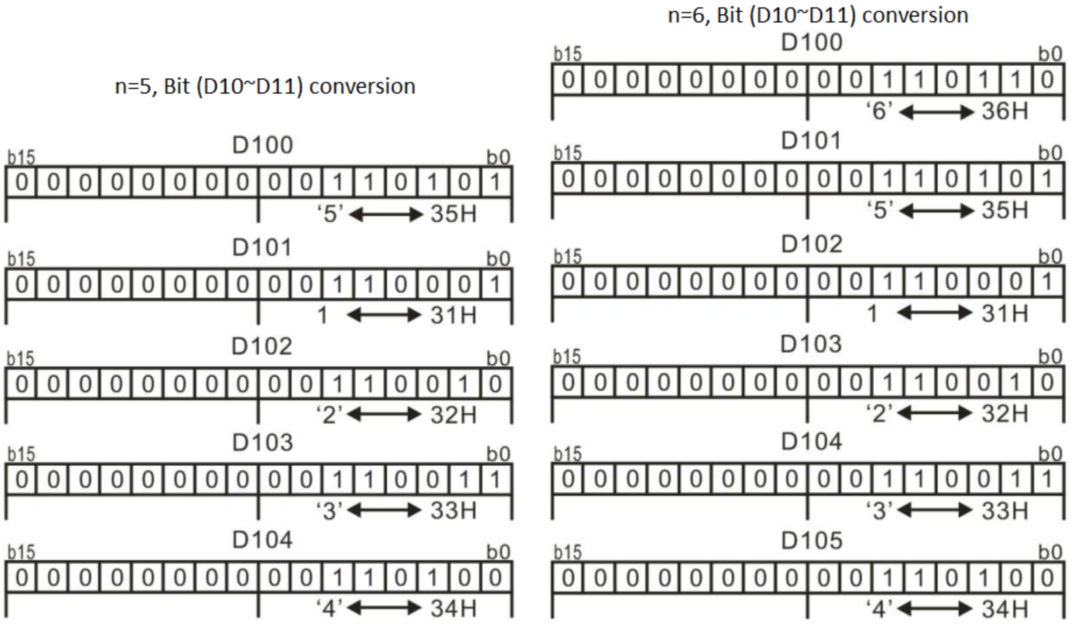


- When M8161=OFF


n=5, Bit(D10~D11) conversion

n=6, Bit (D10~D11) conversion

When M8161=ON


n=6, Bit (D10~D11) conversion

n=5, Bit (D10~D11) conversion

3) Points to note

Instructions such as RS / HEX / ASCI / CCD share the M8161 mode flag, please pay attention on it when programming.

HEX instruction

1) Instruction description

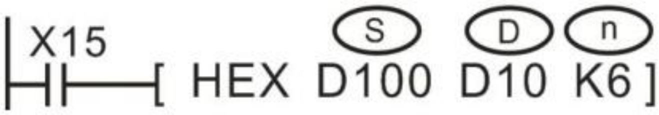
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
HEX	Converts a data value from hexadecimal to ASCII	16	No	HEX S D n	7
HEXP		16	Yes		7

This instruction reads n ASCII data bytes from head source address (S) and converts them in to the equivalent Hexadecimal character, and saved result in D.

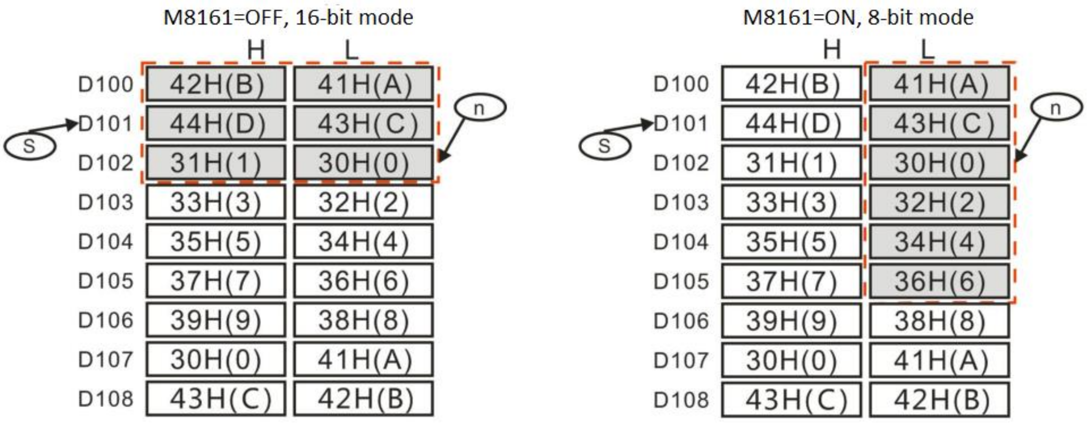
- S: The variable address or constant to be converted. If it is a register variable, the conversion interval will has a width of a 32bit variable.
- D: The starting address for storing the ASCII code.
- n: The data length;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√		√	√	√	√	√	√	√	√	√
D									√	√	√	√	√	√		
n	Constant, n=1~256															

2) Program example



For example, the following data is starting from D100.

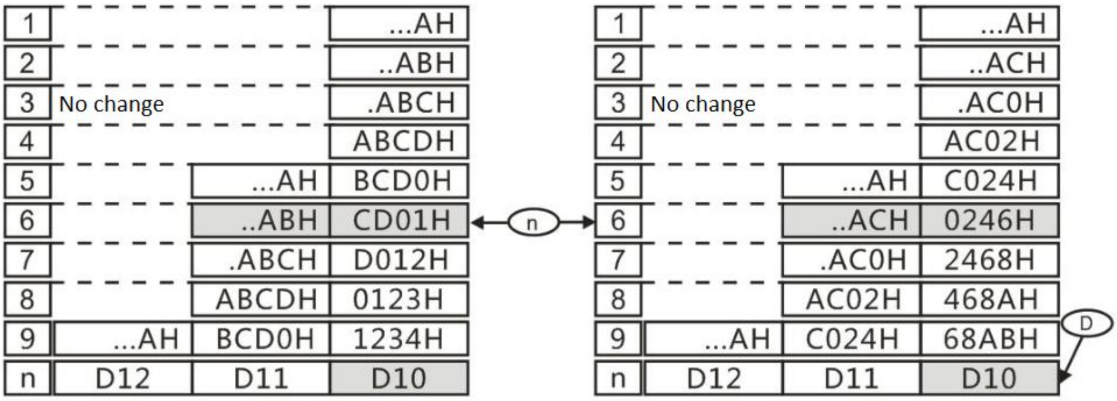


M8161=OFF, 16-bit mode

	H	L
D100	42H(B)	41H(A)
D101	44H(D)	43H(C)
D102	31H(1)	30H(0)
D103	33H(3)	32H(2)
D104	35H(5)	34H(4)
D105	37H(7)	36H(6)
D106	39H(9)	38H(8)
D107	30H(0)	41H(A)
D108	43H(C)	42H(B)

M8161=ON, 8-bit mode

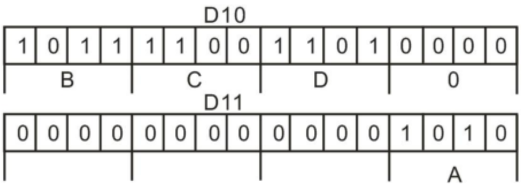
	H	L
D100	42H(B)	41H(A)
D101	44H(D)	43H(C)
D102	31H(1)	30H(0)
D103	33H(3)	32H(2)
D104	35H(5)	34H(4)
D105	37H(7)	36H(6)
D106	39H(9)	38H(8)
D107	30H(0)	41H(A)
D108	43H(C)	42H(B)



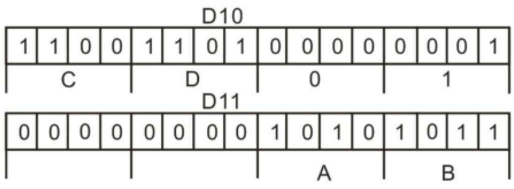
The M8161 flag determines the width mode of the target variable for calculation result storage. When M8161=OFF, it is 16bit mode, which means the higher byte and lower byte are saved respectively. When M8161=ON, it is 8bit mode, which means that only the lower byte is used to save result and the actual variable range length is longer.

- When M8161=OFF

n=5, bit conversion (D100~D102)

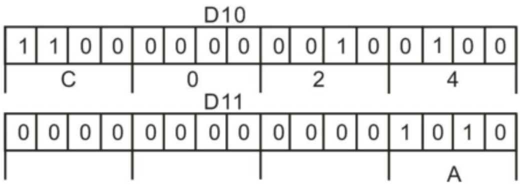


n=6, bit conversion (D100~D102)

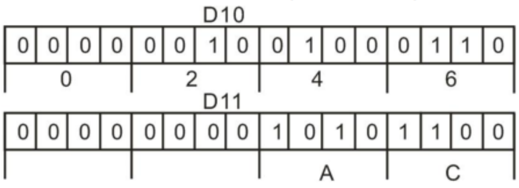


- When M8161=ON

n=5, bit conversion (D100~D104)



n=6, bit conversion (D100~D105)



3) Points to note

- Instructions such as RS / HEX / ASCI / CCD share the M8161 mode flag, please pay attention on it when programming;
- S data area of the source data must be ASCII characters, or error occur when conversion;
- If the output data is in BCD format please do BCD-BIN conversion after HEX conversion, then users can get correct value;

CCD instruction

1) Instruction description

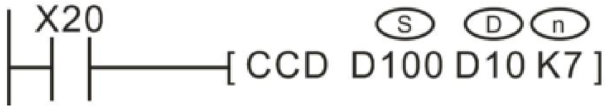
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
CCD	Check Code	16	No	CCD S D n	7
CCDP		16	Yes		7

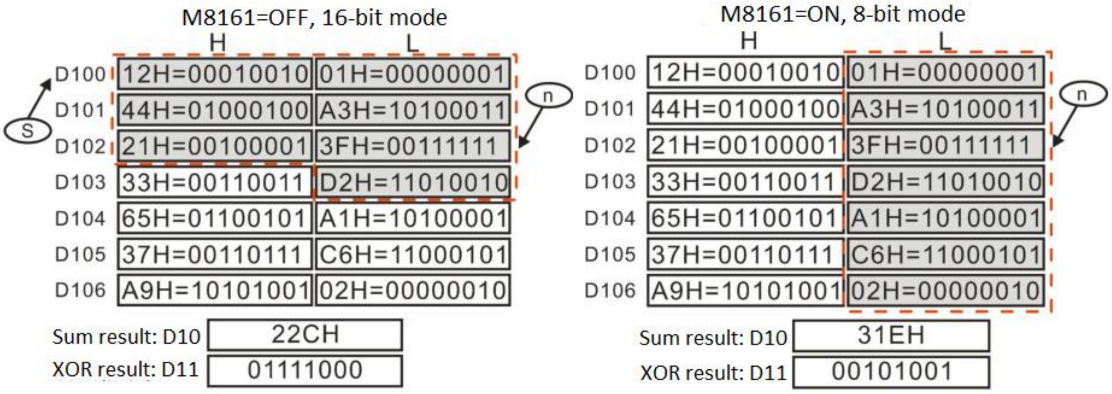
This instruction looks at a byte stack of data from head address (S) for n bytes and checks the vertical bit pattern for parity and sums the total data stack. These two pieces of data are then stored at the destination (D).

- S: The starting address of variables, which are to be checked and calculated;
- D: Respectively used for saving “SUM” result (D+1) is respectively used for saving “XOR” result;
- n: The bit number occupied by variables for checking

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S								√	√	√	√	√	√	√		
D									√	√	√	√	√	√		
n	Constant, n=1~256															

2) Program example




M8161=OFF, 16-bit mode

	H	L
D100	12H=00010010	01H=00000001
D101	44H=01000100	A3H=10100011
D102	21H=00100001	3FH=00111111
D103	33H=00110011	D2H=11010010
D104	65H=01100101	A1H=10100001
D105	37H=00110111	C6H=11000101
D106	A9H=10101001	02H=00000010

Sum result: D10 22CH
XOR result: D11 01111000

M8161=ON, 8-bit mode

	H	L
D100	12H=00010010	01H=00000001
D101	44H=01000100	A3H=10100011
D102	21H=00100001	3FH=00111111
D103	33H=00110011	D2H=11010010
D104	65H=01100101	A1H=10100001
D105	37H=00110111	C6H=11000101
D106	A9H=10101001	02H=00000010

Sum result: D10 31EH
XOR result: D11 00101001

The M8161 flag determines the width mode of the target variable for calculation result storage. When M8161=OFF, it is 16bit mode, which means the higher byte and lower byte are saved respectively. When M8161=ON, it is 8bit mode, which means that only the lower byte is used to save result and the actual variable range length is longer.

The “SUM” is quite simply a summation of the total quantity of data in the data stack.

The "XOR" logical calculation means:

- The involved variables are converted to binary format.
- Then it counts the number of variables with bit0=1. If it is even, the calculation result of bit0 is 0. If it is odd, the calculation result of bit0 is 1.
- Then it counts the number of variables with bit1=1. If it is even, the calculation result of bit1 is 0; if it is odd, the calculation result of bit1 is 1.
- In the same way, calculation is implemented from bit2 to bit7. After that, the binary HEX value converted from binary is the “XOR” result (polarity value).

3) Note for use

- Instructions such as RS / HEX / ASCI / CCD share the M8161 mode flag, please pay attention on it when programming;

PID instruction

1) Instruction description

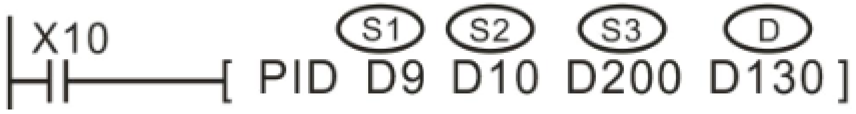
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
PID	PID operation	16	No	PID S_1 S_2 S_3 D	9

This instruction is for PID operation, it is used for control of close-loop system parameter. PID control is widely used in mechanical equipment, pneumatic equipment, constant pressure water supply, electronic equipment and so on.

- S_1: The predefined set value;
- S_2: The current value;
- S_3: The operation parameter, it takes the next 25 addresses, the value range is D0 ~ D7975, it is best to specify the retentive memory, for saving parameter when power OFF;
- D: The destination device, it is better to specify the non-retentive memory, otherwise users need to initialize it before executing instruction;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2														√		
S_3														√		
D														√		

2) Program example



D9 is target value, D10 is current value, the unit for D9 and D10 must be the same.

D200~D224 are used for storing the set value and process value of PID operation. These values must be set item by item before executing PID operation.

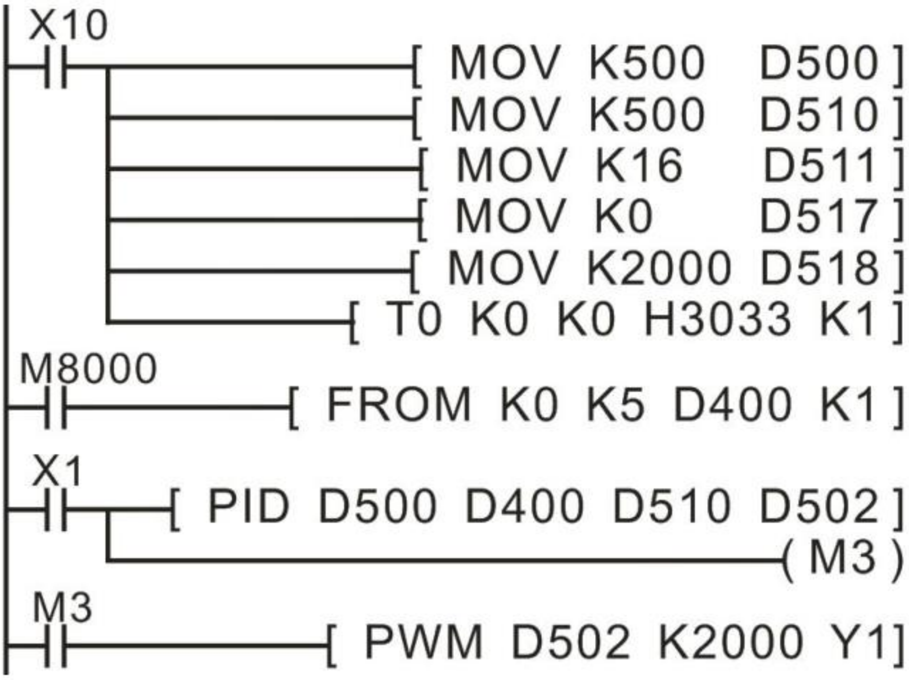
D130 is used for storing the calculated value, it is used for controlling the implementation of the action.

Operation parameters (S_3+N)		
Unit	**Function**	**Description**
S_3	Sample time(Ts)	Setting range 1～32767(ms), but must longer than scanning cycle of plc program

S_3+1	Reaction direction(ACT)	bit0: 0＝positive action; 1＝negative action;
		bit3: 0＝one way; 1＝two way;
		bit4: 0＝disable self-tuning; 1＝enable self-tuning;
		Others cannot be used.

S_3+2	Maximum climbing(Delta T)	Setting range 0~320
S_3+3	Proportional gain(Kp)	Setting range: 0~32767, note:this value is magnified 256 times, actual value is Kp/256
S_3+4	Integral gain(Ki)	Setting range: 0~32767, Ki＝16384Ts/Ti, Ti is integral time
S_3+5	Derivative gain(Kd)	Setting time: 0~32767, Kd≈Td/Ts, Td is derivative time
S_3+6	Filter (C0)	Range: 0~1024
S_3+7	Output lower limit	Recommended range: -2000~2000, when S3+1 bit3=0, please set 0； S3+1 bit3=1, please set -2000
S_3+8	Output upper limit	Recommended values: 2000
S_3+9	Retain	Retain

- Self-tuning example



3) Error code

If an error occurs in the set value of the control parameters or in the PID operation, the operation error flag M8067 turns on and the following data is stored in D8067 according to the error details.

Error code	Error content	State	Processing method
K6705	Operand of application instruction outside of target device	Stop PID operation	Please check data for PID operation
K6706	Operand of application instruction outside of target device		
K6730	(TS< 0) Sampling time(TS) outside of target device (TS< 0)		
K6732	Filter (C0) outside (C0<0 or 1024≤C0)		
K6732	Maximum rate of raise(DeltaT) outsideΔT<0 or 320≤ΔT		
K6733	Proportional gain(KP) outside of target range		
K6734	Integral gain (KI)outside of target range(KI<0)		
K6735	Derivative gain outside of target range(KD<0)		
K6740	Sampling time≤ operation cycle	Continue PID operation	
K6742	Variation of measured value exceed ((PV<-32768 or 32767 < (PV)		
K6751	Direction of Self-tuning isn't match	Continue PID self-tuning	The action direction between set value and current value are not match. Please correct the target value, self-tuning

			output, estimated value, then self-tuning.
K6752	Self-tuning action is improper	Self-tuning	Self-tuning measured value cannot be correct action, due to changes in the upper and lower. Please make the sampling time is much greater than the output change cycle, increase the input filter constant. After changing the setting, please perform auto-tuning again.

4) Note for use

The correct measured value must be read into the PID measured value (PV) before the PID operation is executed. In particular, pay attention to the conversion time when performing PID operation on the value of the analog input module.

PID instruction can be used multiple times and executed at the same time, but variable area of PID instruction cannot overlap; it also can be used in step instruction, jump instruction, timer interruption, subroutine, but please delete S_3+9 cache unit before execute PID instruction.

The maximum error of sampling time TS is - (1 execution cycle+ l ms)~ +(1 execution cycle). If sampling time TS≤1 execution cycle OF PLC, then will have below PID operational error (K6740),and execute PID algorithm as TS = execution cycle, in that case, it is better to use constant scanning mode or use the PID instruction in timer interrupt (16□□~18□□)

5.2.13 Floating calculation

DECMP instruction

1) Instruction description

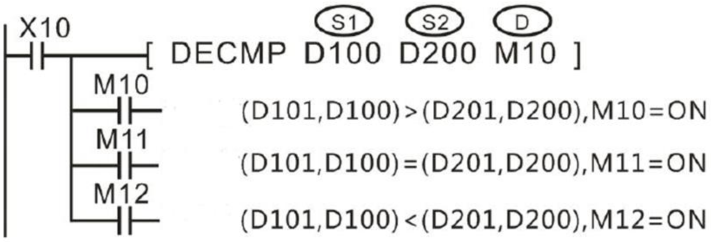
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DECMP	Compares two floating point values - results of <, = and > are given	32	No	DECM S_1 S_2 D	13
DECMP P		32	Yes		13

The data of S_1 is compared to the data of S_2. The result is indicated by 3 bit devices specified with the head address entered as D. The bit devices indicate:

- S_2 is less than < S_1 - bit device D is ON
- S_2 is equal to = S_1 - bit device D +1 is ON
- S_2 is greater than > S_1 - bit device D+2 is ON

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√	√							√		
S_2					√	√	√							√		
D		√	√	√												

2) Program example


X10
S1
S2
D
[DECMP D100 D200 M10]
M10
(D101,D100)>(D201,D200),M10=ON
M11
(D101,D100)=(D201,D200),M11=ON
M12
(D101,D100)<(D201,D200),M12=ON

- When X10 is ON, one of M10~M12 will be on.
- When X10 turns from ON to OFF, DECP is not executed. M10~M12 keep the initial value. User could use RST or ZRST to reset M10~M12.
- If S1 and S2 are not floating number, they will be converted into floating number automatically.

DEZCP instruction

1) Instruction description

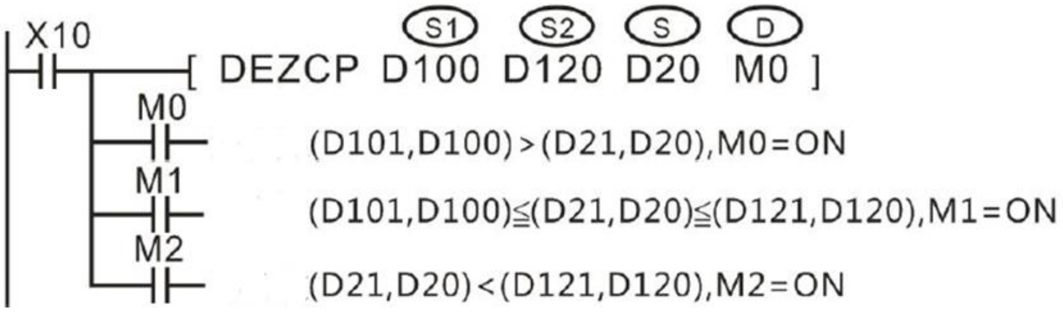
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DEZCP	Compares a float range with a float value.	32	No	DECM S_1 S_2 S D	17
DEZCPP		32	Yes		17

The instruction compares the inter-zoning variables of binary floating-points, and then exports the result to the three (3) initiative variables

- S_1: the inter-zoning minimum of the binary floating-point variables.
- S_2: the inter-zoning maximum of the binary floating-point variables.
- S: the binary floating-point variable that is to be compared.
- D: the storage unit for comparison results, occupying three variable units.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√	√							√		
S_2					√	√	√							√		
S					√	√	√									
D		√	√	√												

2) Program example


S1
S2
S
D
X10
DEZCP D100 D120 D20 M0
M0
(D101,D100)>(D21,D20),M0=ON
M1
(D101,D100)≦(D21,D20)≦(D121,D120),M1=ON
M2
(D21,D20)<(D121,D120),M2=ON

DEBCD instruction

1) Instruction description

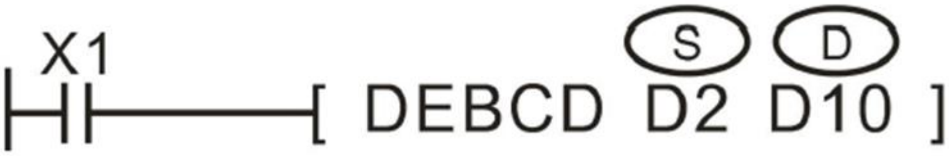
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DEBCD	Converts binary floating point to decimal floating point	32	No	DBCD S D	9
DEBCDP		32	Yes		9

It converts a floating point value at S into separate mantissa and exponent parts at D and D+1(decimal floating).

- S: The binary floating variable;
- D: The storage unit for converted decimal floating result;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							√							√		
D														√		

2) Program example



The binary floating value in (D3, D2) is converted to decimal floating value and the saved to (D11, D10).

There are 23 bits real number, 8 bits exponent, and 1 bit signal in binary floating (D3, D2), which will be converted to decimal floating (D11, D10), and it could be expressed with science formula of $D2*10^{D3}$.

The floating data calculation is PLC is all in binary format, and it is converted to decimal for ease of monitoring.

DEBIN instruction

1) Instruction description

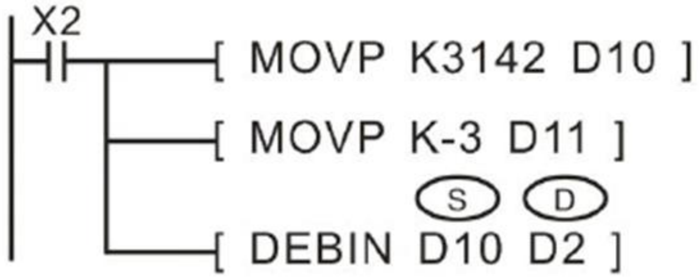
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DEBIN	Converts decimal floating point to binary floating point	32	No	DBIN S D	9
DEBINP		32	Yes		9

This instruction converts decimal floating to binary floating

- S: The decimal floating variable.
- D: The storage unit for converted binary floating result.

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							√							√		
D														√		

2) Program example



The decimal floating 3.142, which is saved in D11, D10, is converted to binary floating and then saved in (D3, D2).

DEADD instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DEADD	Adds two floating point numbers together	32	No	DEADD S_1 S_2 D	13
DEADDP		32	Yes		13

The floating point values stored in the source devices S_1 and S_2 are algebraically added and the result is stored in the destination device D.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√	√							√		
S_2					√	√	√							√		
D														√		

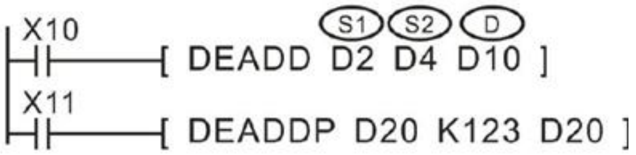
The instruction must use the double word format; i.e., DEADD or DEADDP. All source data and destination data will be double word.

K or H will be regarded as being in floating point format and the result stored in the destination will also be in floating point format.

If the result of the calculation is larger than the largest floating point number then the carry flag, M8021 is set ON and the result is set to the largest value.

If the result of the calculation is smaller than the smallest floating point number then the borrow flag, M8022 is set ON and the result is set to the smallest value.

2) Program example



For DEADD, when X10 is ON, the add operation will be executed in every scanning cycle. For DEADDP, when X11 is ON, the add operation will be executed only once.

DESUB instruction

1) Instruction description

Name	Function	Bits	Pulse type	Instruction format	Step
DESUB	Subtracts one floating point number from another	32	No	DESUB S_1 S_2 D	13
DESUBP		32	Yes		13

The floating point value of S_2 is subtracted from the floating point value of $S_1$and the result stored in destination device D.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√	√							√		
S_2					√	√	√							√		
D														√		

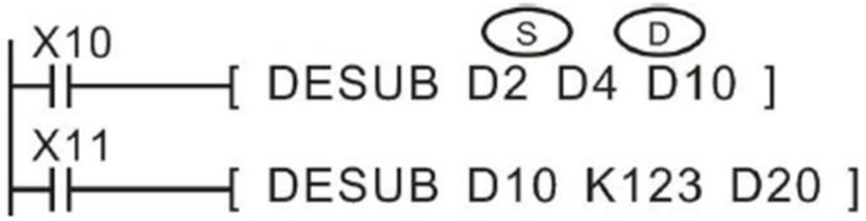
K or H will be regarded as being in floating point format and the result stored in the destination will also be in floating point format.

If the result of the calculation is 0, M8020 will set ON.

If the result of the calculation is larger than the largest floating point number then the carry flag, M8022 is set ON and the result is set to the largest value.

If the result of the calculation is smaller than the smallest floating point number then the borrow flag, M8021 is set ON and the result is set to the smallest value.

2) Program example



When X10 = ON, after the binary floating-point (D3, D2) subtracts the other binary floating-point (D5, D4), the difference result will be stored in (D11, D10).

When X11 turns from OFF to ON, the value of the binary floating-point requires to subtract 123.The constant K123 is automatically converted to binary floating value before calculation.

The storing unit for the subtraction difference can be seemed as same one unit with the subtrahend and minuend. Please use the pulse execution instruction DESUBP under this circumstance. Otherwise, if selected the progressive execution instruction, the subtraction operation will be carried out again every time when the program is scanned.

DEMUL instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DEMUL	Multiplies two floating point numbers together	32	No	DEMUL S_1 S_2 D	13
DEMULP		32	Yes		13

The floating point value of S_1 is multiplied with the floating point value of S_2. The result of the multiplication is stored at D as a floating point value.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√	√							√		
S_2					√	√	√							√		
D														√		

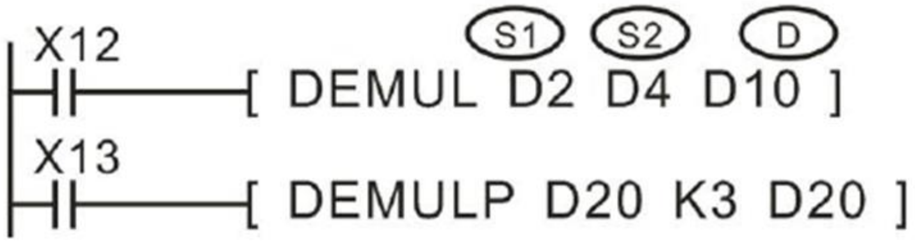
K or H will be regarded as being in floating point format and the result stored in the destination will also be in floating point format.

If the result of the calculation is 0, M8020 will set ON.

If the result of the calculation is larger than the largest floating point number then the carry flag, M8022 is set ON and the result is set to the largest value.

If the result of the calculation is smaller than the smallest floating point number then the borrow flag, M8021 is set ON and the result is set to the smallest value.

2) Program example



When X12 = ON, after the binary floating-point (D3, D2) multiplies the other binary floating-point (D5, D4), the product will be stored in (D11, D10).

When X13 turns from OFF to ON, the binary floating-point (D21, D20) value will be multiplied by 3 (three) and saved back in (D21, D20) the constant K3 has already been automatically converted to a binary floating-point value prior to the calculation.

The storing unit for the multiplication product can be treated as one unit with the multiplicand and the multiplier. Please use the pulse execution instruction DEMULP under this circumstance. Otherwise, if selected the progressive execution instruction, the multiplication operation will be carried out again every time when the program is scanned.

DEDIV instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DEDIV	Divides one floating point number by another	32	No	DEDIV S_1 S_2 D	13
DEDIVP		32	Yes		13

The floating point value of S_1 is divided by the floating point value of S_2. The result of the division is stored in D as a floating point value. No remainder is calculated.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√	√							√		
S_2					√	√	√							√		
D														√		

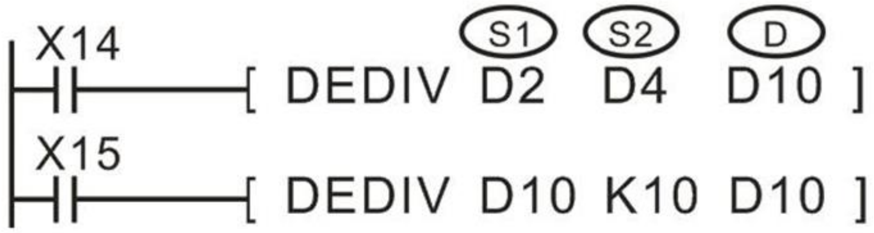
K or H will be regarded as being in floating point format and the result stored in the destination will also be in floating point format.

If the result of the calculation is 0, M8020 will set ON.

If the result of the calculation is larger than the largest floating point number then the carry flag, M8022 is set ON and the result is set to the largest value.

If the result of the calculation is smaller than the smallest floating point number then the borrow flag, M8021 is set ON and the result is set to the smallest value.

2) Program example



```
[ DEDIV D2 D4 D10 ]
[ DEDIV D10 K10 D10 ]
```

When X14=ON and the binary floating variable (D3, D2) are divided by the binary floating variable (D5, D4), the result will be saved in (D11, D10).

When X15 is set from OFF to ON, the binary floating (D11, D10) is divided by 10 and then the result is saved back to (D11, D10).The constant K10 is automatically converted to a binary floating value before calculation.

The storage unit for the result could be the storage unit for the dividend or divisor, in which the pulse-type DEDIVP instruction is recommended, or the continued implementation instruction will be applied, in which the calculation will be implemented every time when the program is scanned.

DESQR instruction

1) Instruction description

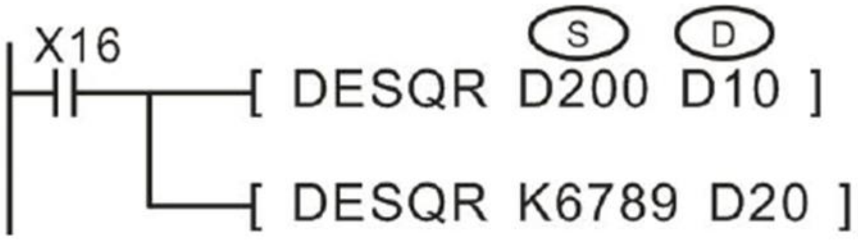
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DESQR	Calculates the square root of a floating point value.	32	No	DESQR S D	9
DESQRP		32	Yes		9

A square root is performed on the floating point value of S, the result is stored in D.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					√	√	√							√		
D														√		

- If S is K or H, it will be regarded as being in floating point format.
- If the result of the calculation is 0, M8020 will set ON.
- S must be greater than 0, if not, M8067 and M8068 will be set ON.

2) Program example



- Solution 1: The binary floating radiation result is saved to (D11, D10)
- Solution 2: The binary floating number K6789 is implemented with radiation calculation and then the result is saved to (D21, D20), where the constant K6789 is automatically converted to binary floating data before implementation;

DINT instruction

1) Instruction description

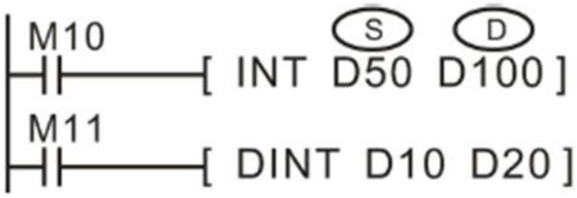
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
INT	Converts a number from floating point format to decimal format	16	No	INT S D	5
INTP		16	Yes		5
DINT		32	No		9
DINTP		32	Yes		9

The floating point value of S is rounded down to the nearest integer value and stored in normal binary format in D.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							√							√		
D														√		

- If the result is 0, M8020 will be set ON.
- If the source data is not a whole number it must be rounded down. In this case the borrow flag M8021 is set ON to indicate a rounded value.
- If the resulting integer value is outside the valid range (16 bit: -32768~32767, 32 bit: -2147483648~2147483647); for the destination device then an overflow occurs. In this case the carry flag M8022 is set on to indicate overflow.

2) Program example



When M10 is triggered, (D51, D50) are rounded down to the nearest integer value and stored in normal binary format in D100.

When M11 is triggered, (D11, D10) are rounded down to the nearest integer value and stored in normal binary format in (D21, D20).

DSIN instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DSIN	Calculates the sine of a floating point value	32	No	DSIN S D	9
DSINP		32	Yes		9

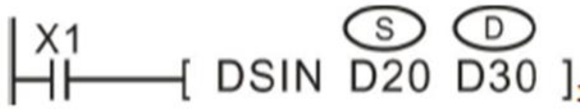
This instruction performs the mathematical SIN operation on the floating point.

- S: The angle variable that needs to be calculated in order to obtain SIN value. The unit is in RAD, and the value is expressed in binary floating points.ValueRange0<=α<=2π;
- D: The storage unit for the SIN calculation results after its conversion. It is in binary floating point format.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example

Example 1:

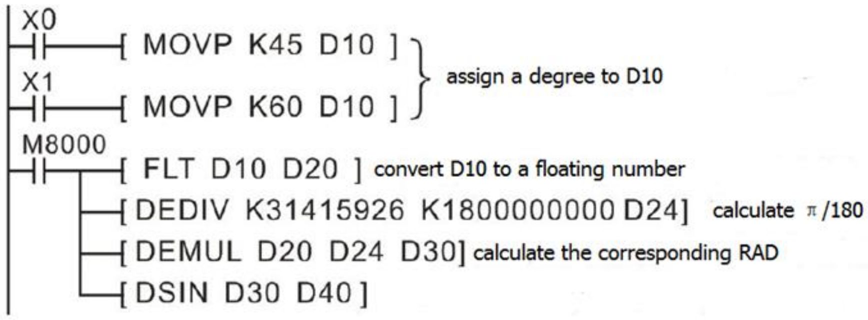


```
X1
─┤├──[ DSIN D20 D30 ]
          S    D
```

The unit of S is rad, the range is from 0 to 2π.

RAD=DEGREE*π/180°

Example 2:



```
X0
─┤├──[ MOVP K45 D10 ]    } assign a degree to D10
X1
─┤├──[ MOVP K60 D10 ]    }
M8000
─┤├──[ FLT D10 D20 ] convert D10 to a floating number
   ├─[ DEDIV K31415926 K1800000000 D24]  calculate π/180
   ├─[ DEMUL D20 D24 D30] calculate the corresponding RAD
   └─[ DSIN D30 D40 ]
```

DCOS instruction

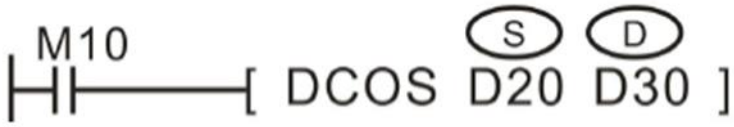
1) Instruction description

Name	Function	Bits (bits)	Pulse type	Instruction format	Step
DCOS	DCOS Calculates the cosine of a floating point value	32	No	DCOS S D	9
DCOSP		32	Yes		9

This instruction performs the mathematical cos operation on the floating point value in S. The result is stored in D.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



When M10 is triggered, radian (D21, D20) are implement with COS calculation and saved to (D31, D30).

The calculated source data and COS results are in binary floating format.

RAD (radian) value= angle×π/180°,for example, the radian correspondingtoangle360°=360°×π/180°=2π.

For the program instruction for the COS calculation of an angle, please refer to examples in the SIN instruction.

DTAN instruction

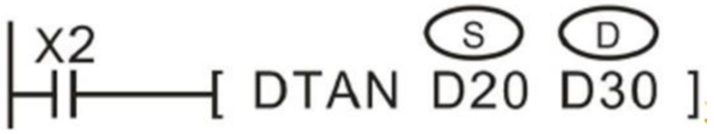
1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
DTAN	Calculates the tangent of a floating point value	32	No	DTAN S D	9
DTANP		32	Yes		9

This instruction performs the mathematical TAN operation on the floating point value in S. The result is stored in D.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



When X2 is triggered, it calculations the TAN value of radian (D21, D20) and saved it to (D31, D30).

The calculated source data and SIN results are all in binary floating point value format.

RAD(radian)value＝angle × π/180°,for example, the radian corresponding to angle 360° = 360°× π/180° = 2π.

In regards to the programming statements used to calculate the TAN value, please refer to the example in the SIN instruction section.

DASIN instruction

1) Instruction description

Name	Function	Bits (bits)	Pulse type	Instruction format	Step
DASIN	Calculate the corresponding radian value based on the SIN value	32	No	DASIN S D	9
DASINP		32	Yes		9

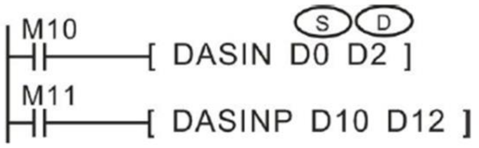
This instruction performs the mathematical ARCSIN operation on the floating point.

- S: The value of SIN, it is in binary floating-point format, the range is-1.0<=α<= 1.0;
- D: It used for store result, the range is -0.5π~+0.5π

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example

Example 1:



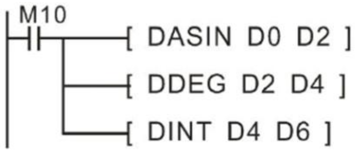
When M0= 0, SIN $^{-1}$ operation (D1, D0) carried out, and result is saved in (D3, D2).

SIN $^{-1}$ (D1, D0) -> (D3, D2)

The source data and results are binary floating-point format.

Angle in radians=angle in degree*π/180°

Example 2:



If (D1, D0) is 0.707106781, when M10 turns from OFF to ON, (D3, D2) will be 0.78539815, (D5,D4) will be 45, (D7,D6) will be 45.

DACOS instruction

1) Instruction description

Name	Function	Bits (bits)	Pulse type	Instruction format	Step
DACOS	Calculate the corresponding radian value based on the COS value	32	No	DACOS S D	9
DACOSP		32	Yes		9

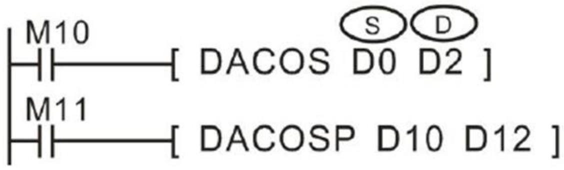
Calculate radian value, according to the corresponding value of COS.

- S: The value of COS, it is in binary floating-point format, the range is-1.0<=α<= 1.0;
- D: It used for store result, the range is 0~π

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example

Example 1:



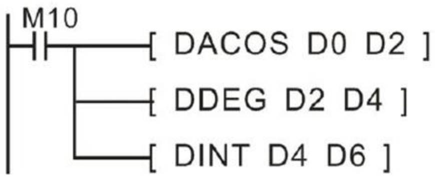
When M0=0, COS-1 operation (D1, D0) carried out, and result is saved in (D3, D2).

COS^{-1}(D1, D0)->(D3, D2).

The source data and results are binary floating-point format.

Angle in radians=angle in degree*π/180°

Example 2:



If (D1, D0) is 0.866025404, when M10 turns from OFF to ON, (D3, D2) will be 0.52359877, (D5, D4) will be 30, (D7, D6) will be 30.

DATAN instruction

1) Instruction description

Name	Function	Bits (bits)	Pulse type	Instruction format	Step
DATAN	Calculate the corresponding radian value based on the TAN value	32	No	DATAN S D	9
DATANP		32	Yes		9

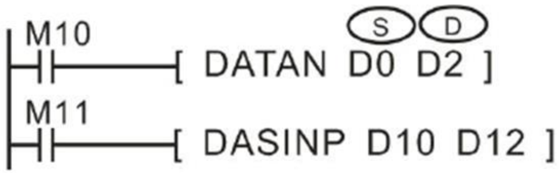
Calculate radian value, according to the corresponding value of TAN.

- S: The value of TAN, it is in binary floating-point format;
- D: It used for store result, the range is –π/2~+π/2

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example

Example 1:



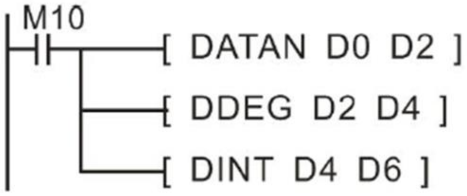
When M0= 0, TAN -1 operation (D1, D0) carried out, and result is saved in (D3, D2).

TAN -1 (D1、D0) -> (D3、D2)

The source data and results are binary floating-point format.

Angle in radians=angle in degree*π/180°

Example 2:



If (D1, D0) is 1.732050808, when M10 turns from OFF to ON, (D3, D2) will be 1.04719753, (D5, D4) will be 60, (D7, D6) will be 60.

DSINH instruction

1) Instruction description

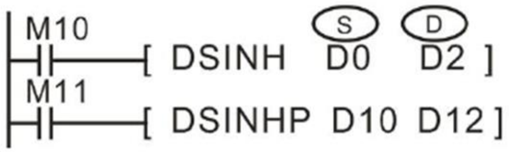
Name	Function	Bits (bits)	Pulse type	Instruction format	Step
DSINH	The operation of Hyperbolic Sine function SINH (Binary floating)	32	No	DSINH S D	9
DSINHP		32	Yes		9

This instruction performs the mathematical SINH operation on the floating point value. D= $(e^{S}-e^{-S})/2$

- S: The binary floating-point for SINH;
- D: It used to save result;

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



When M10 turn ON, calculate value (D1, D0) of SINH, and saved the result in (D3, D2). DSINH (D1, D0) → (D3, D2)

DCOSH instruction

1) Instruction description

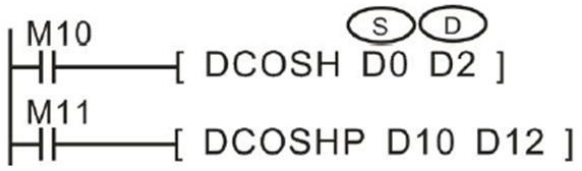
Name	Function	Bits (bits)	Pulse type	Instruction format	Step
DCOSH	Binary floating point hyperbolic sine function COSH operation	32	No	DCOSH S D	9
DCOSHP		32	Yes		9

This instruction performs the mathematical COSH operation on the floating point. D=(e^{S}+e^{-S})/2

- S: The binary floating-point for COSH;
- D: The used to store result;

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



When M10 turn ON, calculate value (D1, D0) of COSH, and saved the result in (D3, D2).

DCOSH (D1, D0) → (D3, D2)

DTANH instruction

1) Instruction description

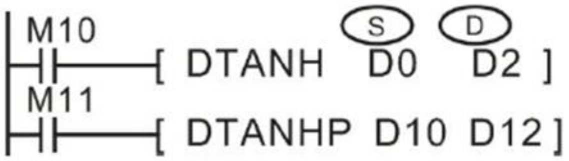
Name	Function	Bits (bits)	Pulse type	Instruction format	Step
DTANH	The operation of Hyperbolic Tangent function TANH	32	No	DTANH S D	9
DTANHP		32	Yes		9

This instruction performs the mathematical TANH operation on the floating point. D= $(e^{s}-e^{-s})/ (e^{s}+e^{-s})$

- S: The binary floating-point for TANH;
- D: It used to save result;

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



```
M10
─┤├──[ DTANH  D0  D2 ]
M11
─┤├──[ DTANHP D10 D12]
```

When M10 is triggered, calculate value (D1, D0) of TANH, and saved the result in (D3, D2). DTANH (D1, D0) → (D3, D2)

DDEG Instruction

1) Instruction description

Name	Function	Bits (bits)	Pulse type	Instruction format	Step
DDEG	The operation of converting radian to angle (Binary floating-point)	32	No	DDEG S D	9
DDEGP		32	Yes		9

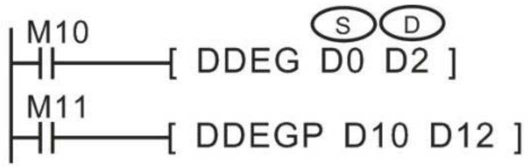
This instruction is used for converting radian to angle (Binary floating-point). The formula is RAD value= Angle*π/180°

- S: The radian;
- D: It used to save result;

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example

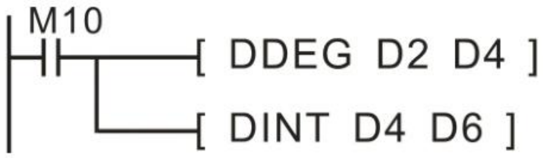
Example 1



When M10 turn ON, convert radian (D1, D0) to angle, and saved the result in (D3, D2). DDEG (D1, D0) → (D3, D2)

RAD (radian) value= angle×π/180°, for example, the radiancorrespondingtoangle360°=360°×π/180°= 2π.

Example 2



If the value of (D1, D0) is 3.1415926, when M10 turn ON, the value of (D3, D2) is 180; the value of (D5, D4) is 180.

DRAD instruction

1) Instruction description

Name	Function	Bits (bits)	Pulse type	Instruction format	Step
DRAD	The operation of Hyperbolic Tangent function TANH	32	No	DTANH S D	9
DRADP		32	Yes		9

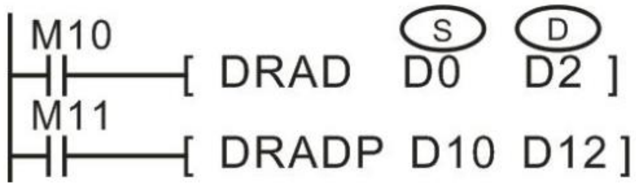
This instruction is used for converting angle to radian (Binary floating-point). The formula is RAD value= Angle*π/180°

- S: The angle;
- D: It used to save result.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example

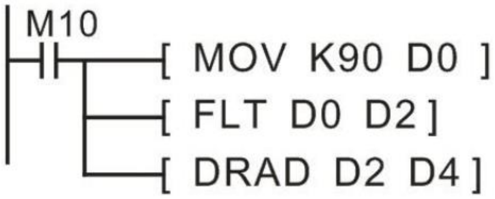
Example 1:



When M10 turn ON, convert angle (D1, D0) to radian, and saved the result in (D3, D2). DRAD (D1, D0) → (D3, D2)

RAD (radian) value= angle×π/180°,for example, the radian corresponding to angle 360°= 360°× π/ 180°= 2π.

Example 2:



When M10 is ON, (D5, D4) is π/2, i.e. 1.570796

DEXP instruction

1) Instruction description

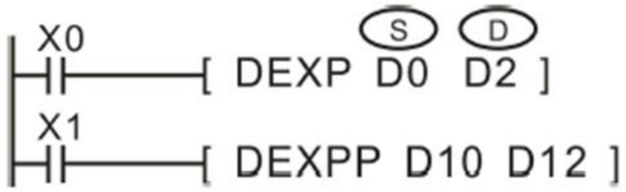
Name	Function	Bits	Pulse type	Instruction format	Step
DEXP	Exponential operation of binary floating number with e as the base number	32	No	DEXP S D	9
DEXPP		32	Yes		9

This instruction performs the exponential operation on S with e(2.71828) as the base number and store the result in D.

When D is not within 2^{-126} ~2^{128}, an error will occur. The error code is K6707 that is stored in D8067, and M8067 will set ON.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



When X0 is triggered, E (D1, D0) →(D3,D2). Because loge2128=88.7, so when (D1, D0) is greater than 88.7, D8067 is k6706, M8067 will set ON.

DLOG10 instruction

1) Instruction description

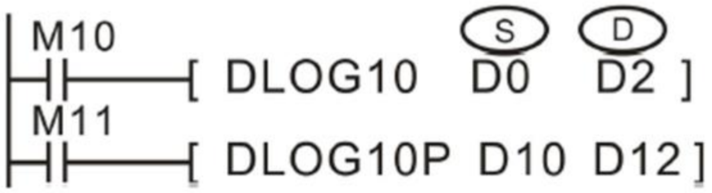
Name	Function	Bits	Pulse type	Instruction format	Step
DLOG10	Logarithmic operations of binary floating numbers with 10 as the base number	32	No	DLOG10 S D	9
DLOG10P		32	Yes		9

This instruction performs common logarithm operation of binary floating-point number to base 10.

- S: the binary floating-point variables of common logarithm in exponent binary floating-point number.
- D: the storage unit for saving the operation result of common logarithm
- Note: the value in can only be positive number. Operational error will occur when the value in is 0 or negative number. Error code K6706 is saved in D8067 and error flag M8067 turns ON.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



When M10 turns ON, the common logarithm operation of binary floating-point number in (D1, D0) to base 10 is performed, and save the result in (D3, D2)

Log_{10} (D1, D0) →(D3, D2).

DLOGE instruction

1) Instruction description

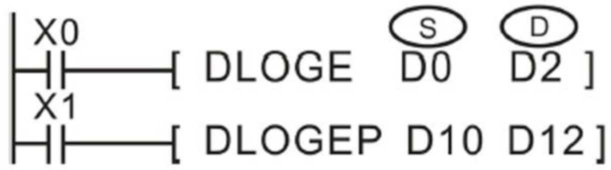
Name	Function	Bits	Pulse type	Instruction format	Step
DLOGE	The natural logarithmic operation of binary floating number with e (2.71828) as the base number	32	No	DLOGE S D	9
DLOGEP		32	Yes		9

This instruction performs common logarithm operation of binary floating-point number to base 10.

- S: the binary floating-point variables of common logarithm in exponentiated binary floating-point number.
- D: the storage unit for saving the operation result of common logarithm
- Note: the value in S can only be positive number. Operational error will occur when the value in S is 0 or negative number. Error code K6706 is saved in D8067 and error flag M8067 turns ON.

Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S														√		
D														√		

2) Program example



When X0 is triggered, $\text{Loge}^{(D1,D0)}$ →(D3,D2).

The conversation between natural logarithmic operation and common logarithmic operation is as below:

$$10^{X} = e^{\frac{X}{0.4342945}}$$

5.2.14 Circular interpolation instruction

G90G01 instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
G90G01	Absolute position line interpolation	16	No	G90G01 S_1 S_2 D_1 D_2	21

- S_1: Target position, will occupy 6 continue address. The range is -2147483648 ~ 2417482647. Target position can be specified by a absolute address; S_1: Target position of X axis, S_1+2: Target position of Y axis, S_1 +4:Target position of Z axis.
- S_2: The output frequency of the synthesis;
- D_1: High speed pulse output port, only Y0 can be specified, occupy 3 continue address (Y0, Y1, and Y2);
- D_2: The operating direction output port, occupy 3 continue address;

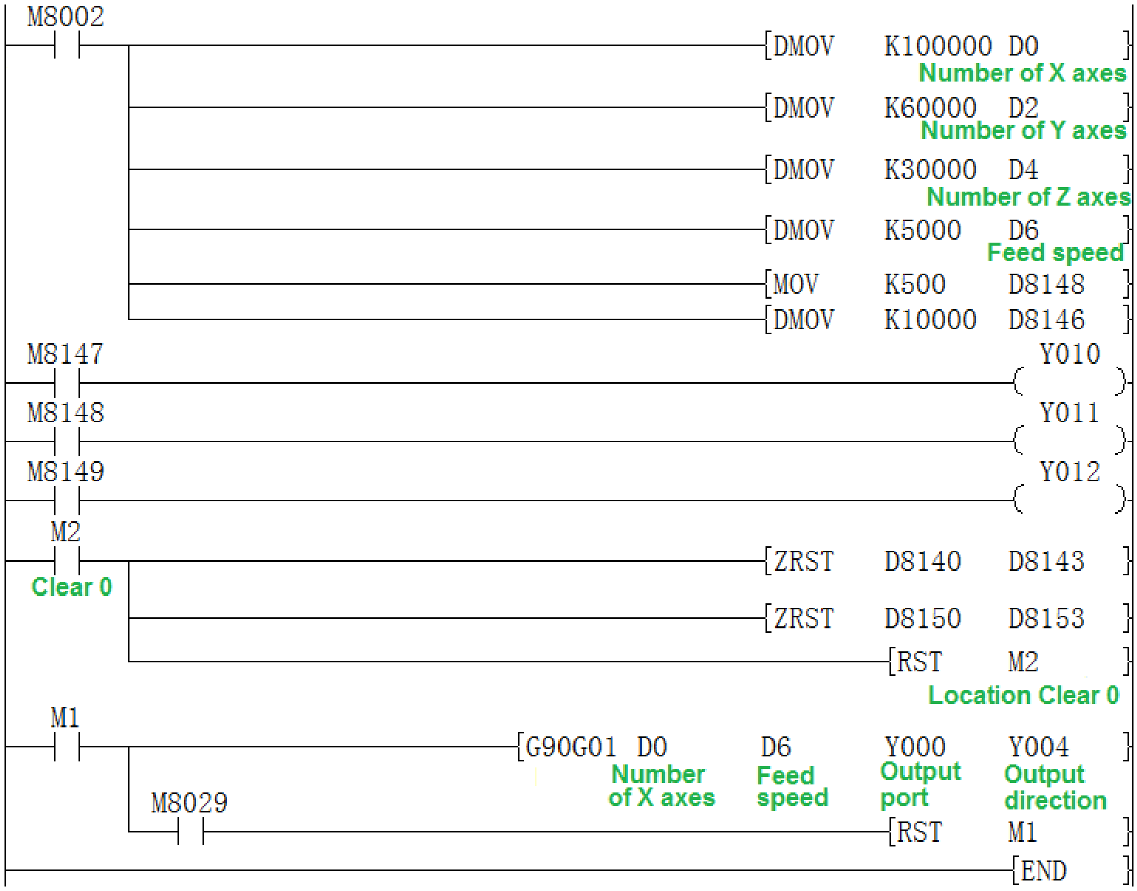
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2					√	√								√		
D_1		√														
D_2		√	√													

2) Program example

The system variables when execution of instruction:

Instruction	G90G01		
Function	Absolute position line interpolation		
Axis	3		
Output port	Y0	Y1	Y2
Present position (double byte)	D8140	D8142	D8150
Time of ACC/DEC (byte)	D8148		
	D8104	D8105	D8106
Basic velocity (double)	D8145		

Maximum velocity (double byte)	D8146		
Pulse output interrupt	M8145	M8146	M8152
BUSY/ READY	M8147	M8148	M8149
ACC/ DEC	Trapezium AC/DE		



3) Note for use

a) When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

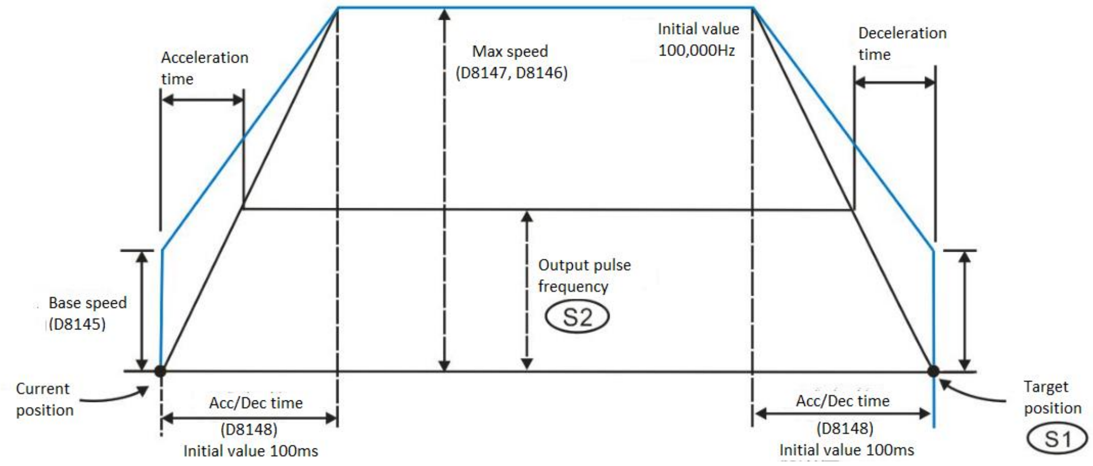
b) Only support trapezoidal acceleration and deceleration;

c) The actuality output synthesized frequency (lowest frequency),the computation formula is as follows:

$$V_{min}=\sqrt{\frac{\text{Maximum operating frequency(D8146)}}{2*\text{ACC\&DEC time}/1000}}$$

d) The output frequency range of interpolation (not synthesized frequency):10~100 KHz;

e) Frequency calculation:


Acceleration time
Max speed (D8147, D8146)
Initial value 100,000Hz
Deceleration time
Output pulse frequency
S2
Base speed (D8145)
Current position
Acc/Dec time (D8148) Initial value 100ms
Acc/Dec time (D8148) Initial value 100ms
Target position
S1

G91G01 instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
G91G01	Relative position line interpolation	16	No	G91G01 S_1 S_2 D_1 D_2	21

- S_1: Target position, will occupy 6 continue address. The range is -2147483648 ~ 2417482647. Target position can be specified by a relative address; S_1: Target position of X axis, S_1+2: Target position of Y axis, S_1 +4:Target position of Z axis.
- S_2: The output frequency of the synthesis;
- D_1: High speed pulse output port, only Y0 can be specified, occupy 3 continue address (Y0, Y1, and Y2);
- D_2: The operating direction output port, occupy 3 continue address;

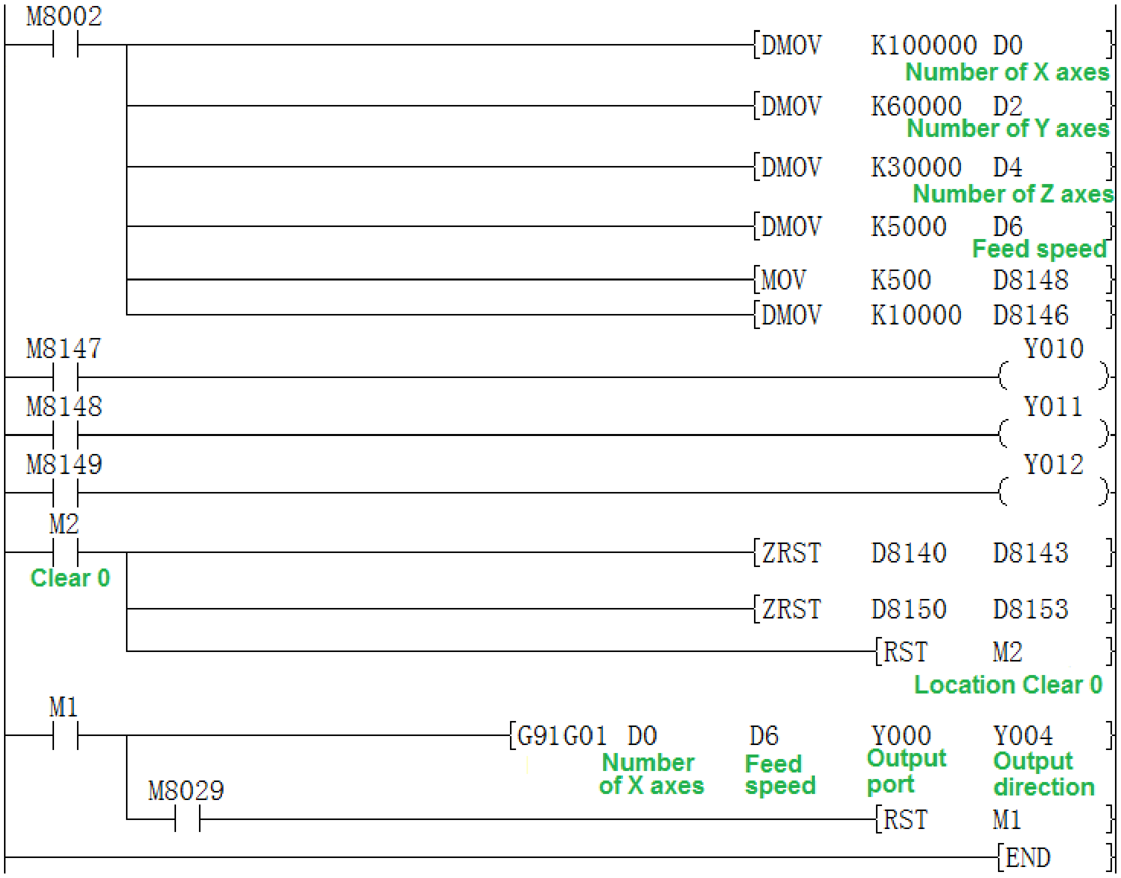
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2					√	√								√		
D_1		√														
D_2		√	√													

2) Program example

The system variables when execution of instruction:

Instruction	G91G01		
Function	Relative position line interpolation		
Axis	3		
Output port	Y0	Y1	Y2
Present position (double byte)	D8140	D8142	D8150
Time of ACC/DEC (byte)	D8148		
	D8104	D8105	D8106
Basic velocity (double)	D8145		
Maximum velocity (double byte)	D8146		
Pulse output interrupt	M8145	M8146	M8152

BUSY/ READY	M8147	M8148	M8149
ACC/ DEC	Trapezium AC/DE		



3) Note for use

a) When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

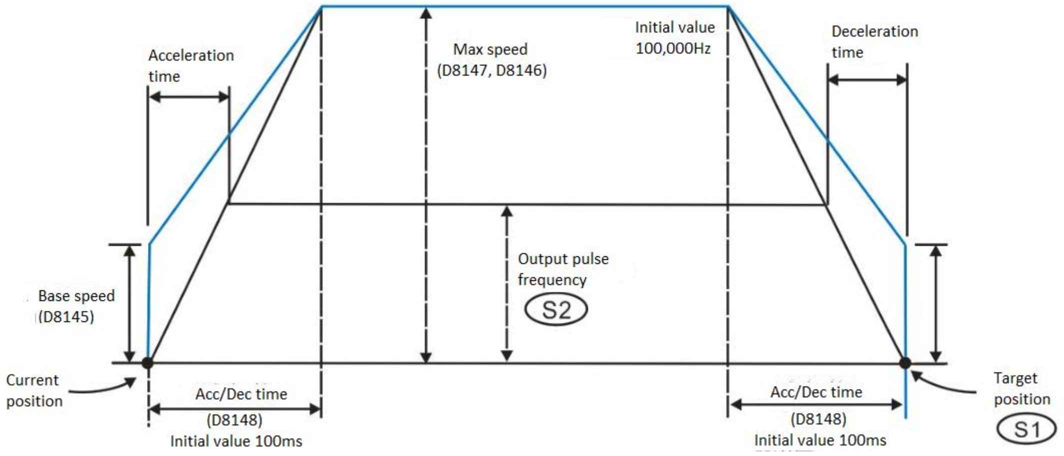
b) Only support trapezoidal acceleration and deceleration;

c) The actuality output synthesized frequency (lowest frequency),the computation formula is as follows:

$$V_{min}=\sqrt{\frac{\text{Maximum operating frequency(D8146)}}{\text{2*ACC\&DEC time/1000}}}$$

d) The output frequency range of interpolation (not synthesized frequency):10~100 KHz;

e) Frequency calculation:


Acceleration time
Max speed (D8147, D8146)
Initial value 100,000Hz
Deceleration time
Output pulse frequency
S2
Base speed (D8145)
Current position
Acc/Dec time (D8148) Initial value 100ms
Acc/Dec time (D8148) Initial value 100ms
Target position
S1

G90G02 instruction

1) Instruction description

Name	Function	Bits (bits)	Pulse type	Instruction format	Step
G90G02	Absolute position of the clockwise circular interpolation	16	No	G91G02 S_1 S_2 S_3 D_1 D_2	21

- S_1: Target position, will occupy 6 continue address. The range is -2147483648 ~ 2417482647. Target position can be specified by a absolution address; S_1: Target position of X axis, S_1+2: Target position of Y axis.
- S_2: Radius/r, occupy 4 bit variable in the continue address .The radius will always treated as relative address; S_2+0:The pulse output D-value between center coordinate and present position ,or the pulse amount of radius “R"; S_2+2:The pulse output D-value between center coordinate and present position .When we use radius mode ,the value must be 0x7FFFFFFF.The range is -2,147,483,648 ～ 2,147,483,647.
- S_3: The output frequency of the synthesis;
- D_1: High speed pulse output port, only Y0 can be specified, occupy 3 continue address (Y0, Y1, and Y2);
- D_2: The operating direction output port, occupy 2 continue address;

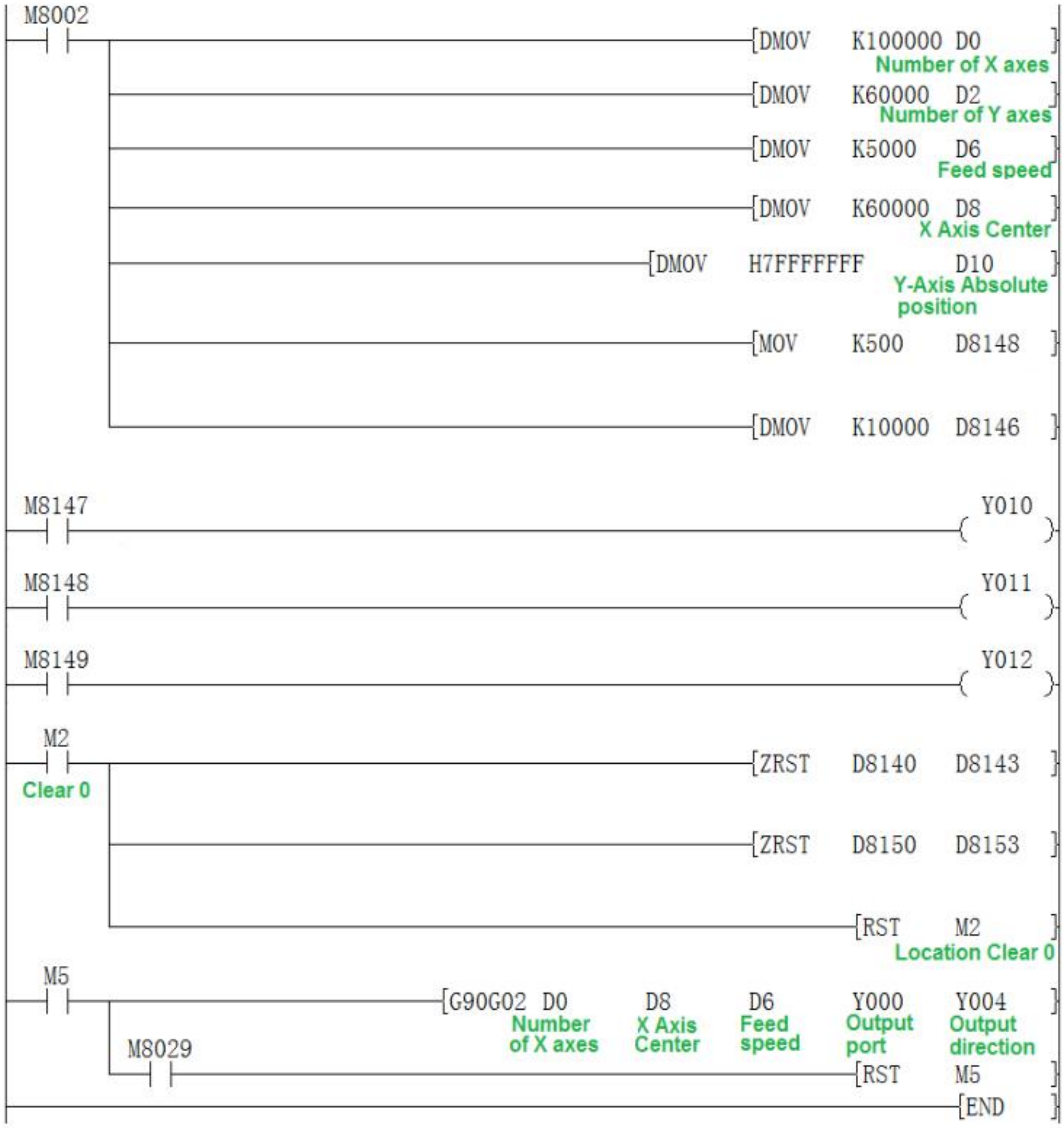
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2														√		
S_3					√	√								√		
D_1		√														
D_2		√	√													

2) Program example

The system variables when execution of instruction:

Instruction	G91G02
Function	Absolute position clockwise circular interpolation
Axis	2

Output port	Y0	Y1
Present position (double byte)	D8140	D8142
Time of ACC/DEC (byte)	D8148	
	D8104	D8105
Basic velocity (double)	D8145	
Maximum velocity (double byte)	D8146	
Pulse output interrupt	M8145	M8146
BUSY/ READY	M8147	M8148
ACC/ DEC	Trapezium AC/DE	



3) Note for use

a) The arc should be more than 20 pulses in circular interpolation, otherwise there will be error;

b) The maximum radius of circular interpolation support is 8000000 pulse;

c) The selection of S_2 have two modes: IJ mode (center coordinates) and R mode(radius mode).When set the value of S_2+2 as 0x7FFFFFFF,it is R mode (radius mode), otherwise it is IJ mode (center coordinates) mode .

d) IJ mode: S_1 only express the relative position between present position and center coordinates regardless absolute or relative interpolation mode. It should be calculated with pulse deviation value.

e) R mode (radius mode),when the value of R is positive, it is express a circular arc less than 180 degrees ;inversely ,it is express a circular arc greater than 180 degrees .In R mode , you can’t generate the full circle ,because it has infinitely solution.

f) When S_1 express the relative address of target position, the target position should be logical to insure create a correct path of target of arc .When both S_1+0 and S_1+2 equal 0, it will generate a full circle.

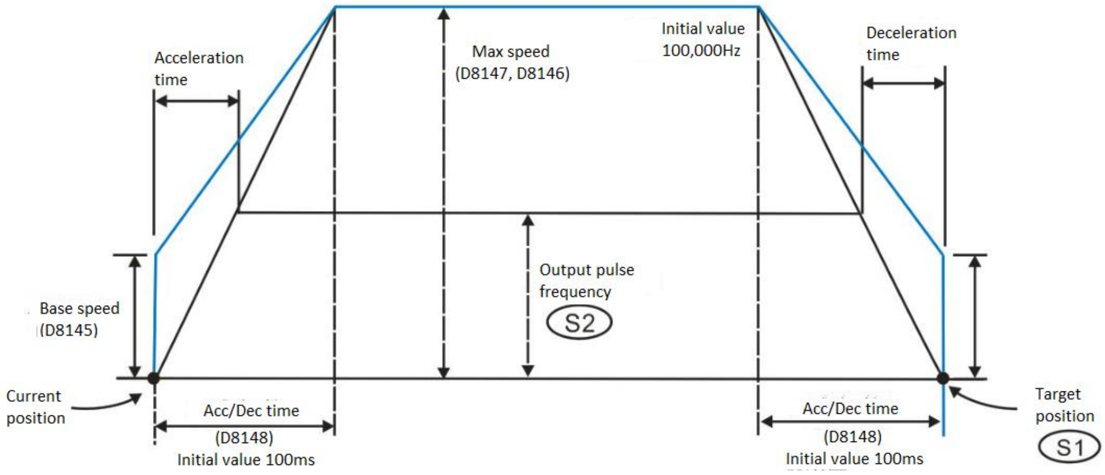
g) When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

h) The actuality output synthesized frequency (lowest frequency),the computation formula is as follows:

$$V_{min}=\sqrt{\frac{\text{Maximum operating frequency(D8146)}}{\text{2*ACC\&DEC time/1000}}}$$

i) The output frequency range of interpolation (not synthesized frequency):10~100 KHz;

j) Frequency calculation:



G91G02 instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
G91G02	Relative position clockwise circular interpolation	16	No	G91G02 S_1 S_2 S_3 D_1 D_2	21

- S_1: Target position, will occupy 6 continue address. The range is -2147483648 ~ 2417482647. Target position can be specified by a absolution address; S_1: Target position of X axis, S_1+2: Target position of Y axis.
- S_2: Radius/r, occupy 4 bit variable in the continue address .The radius will always treated as relative address; S_2+0:The pulse output D-value between center coordinate and present position ,or the pulse amount of radius "R"; S_2+2:The pulse output D-value between center coordinate and present position .When we use radius mode ,the value must be 0x7FFFFFFF.The range is -2,147,483,648 ~ 2,147,483,647.
- S_3: The output frequency of the synthesis;
- D_1: High speed pulse output port, only Y0 can be specified, occupy 3 continue address (Y0, Y1, and Y2);
- D_2: The operating direction output port, occupy 2 continue address;

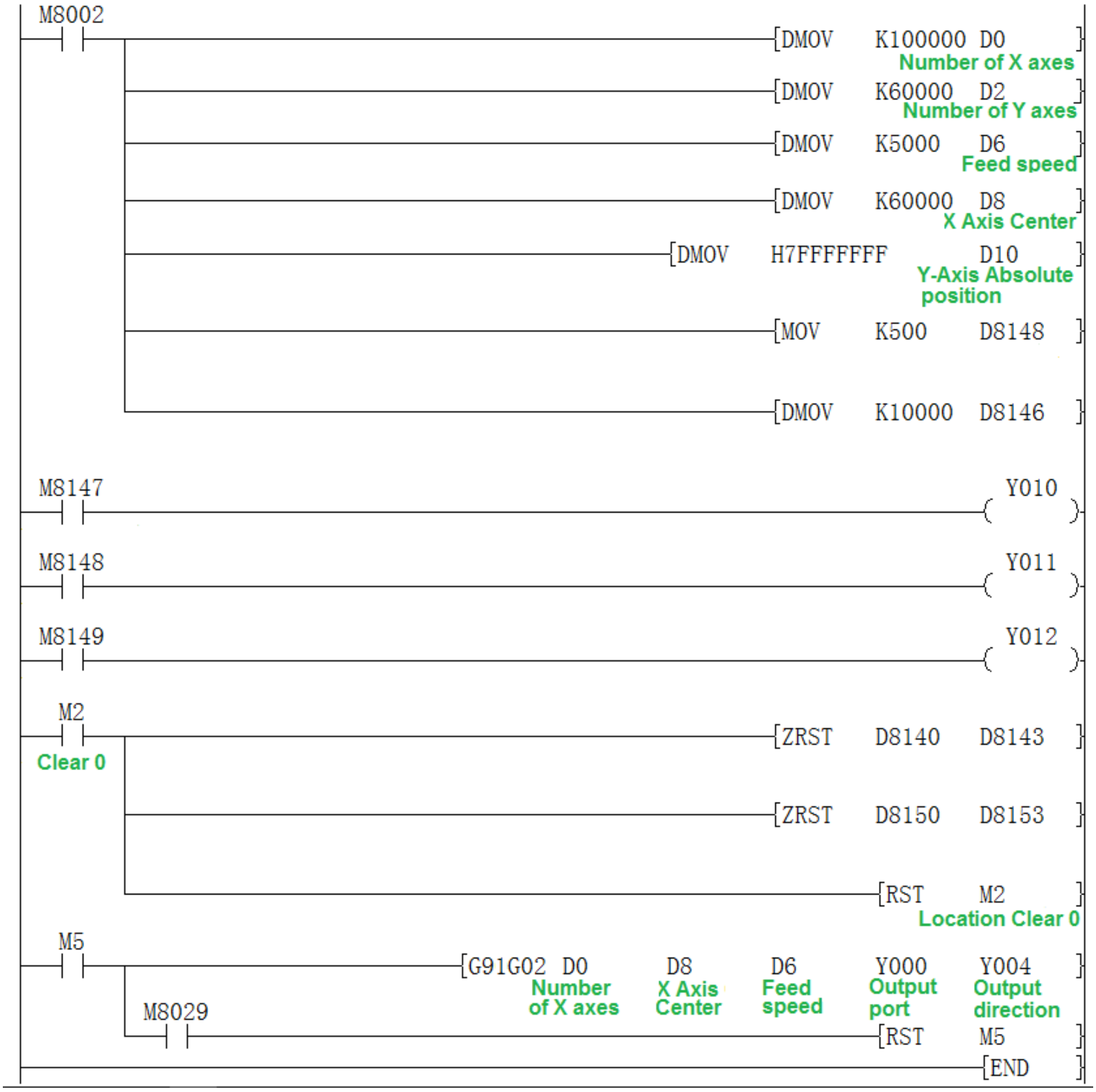
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2														√		
S_3					√	√								√		
D_1		√														
D_2		√	√													

2) Program example

The system variables when execution of instruction:

Instruction	G91G02
Function	Absolute position clockwise circular interpolation
Axis	2

Output port	Y0	Y1
Present position (double byte)	D8140	D8142
Time of ACC/DEC (byte)	D8148	
	D8104	D8105
Basic velocity (double)	D8145	
Maximum velocity (double byte)	D8146	
Pulse output interrupt	M8145	M8146
BUSY/ READY	M8147	M8148
ACC/ DEC	Trapezium AC/DE	



3) Note for use

a) The arc should be more than 20 pulses in circular interpolation, otherwise there will be error;

b) The maximum radius of circular interpolation support is 8000000 pulse;

c) The selection of S_2 have two modes: IJ mode (center coordinates) and R mode(radius mode).When set the value of S_2+2 as 0x7FFFFFFF,it is R mode (radius mode), otherwise it is IJ mode (center coordinates) mode .

d) IJ mode: S_1 only express the relative position between present position and center coordinates regardless absolute or relative interpolation mode. It should be calculated with pulse deviation value.

e) R mode (radius mode),when the value of R is positive, it is express a circular arc less than 180 degrees ;inversely ,it is express a circular arc greater than 180 degrees .In R mode , you can’t generate the full circle ,because it has infinitely solution.

f) When S_1 express the relative address of target position, the target position should be logical to insure create a correct path of target of arc .When both S_1+0 and S_1+2 equal 0, it will generate a full circle.

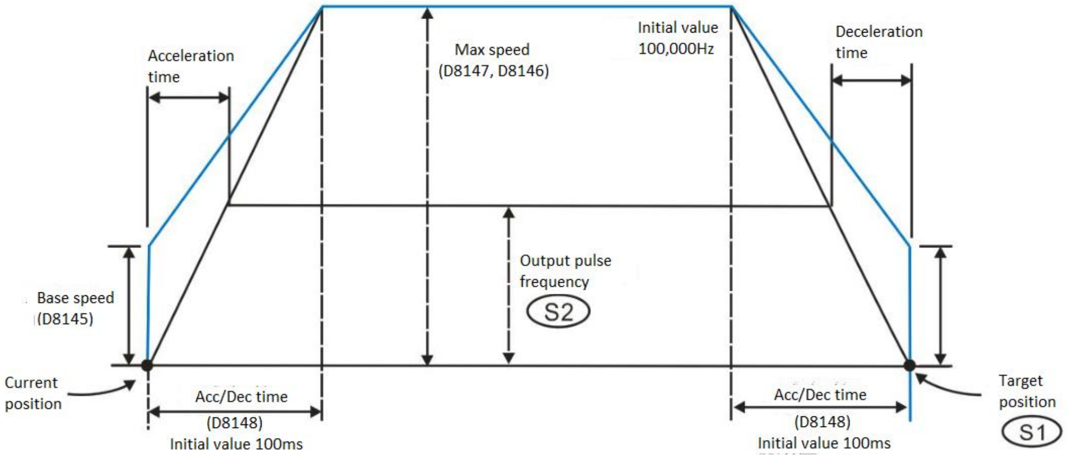
g) When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

h) The actuality output synthesized frequency (lowest frequency),the computation formula is as follows:

$$V_{min}=\sqrt{\frac{\text{Maximum operating frequency(D8146)}}{\text{2*ACC\&DEC time/1000}}}$$

i) The output frequency range of interpolation (not synthesized frequency):10~100 KHz;

j) Frequency calculation:



G90G03 instruction

1) Instruction description

Name	Function	Bits (bits)	Pulse type	Instruction format	Step
G90G03	Relative position anticlockwise circular interpolation	16	No	G90G03 S_1 S_2 S_3 D_1 D_2	21

- S_1: Target position, will occupy 6 continue address. The range is -2147483648 ~ 2417482647. Target position can be specified by a absolution address; S_1: Target position of X axis, S_1+2: Target position of Y axis.
- S_2: Radius/r, occupy 4 bit variable in the continue address .The radius will always treated as relative address; S_2+0:The pulse output D-value between center coordinate and present position ,or the pulse amount of radius "R"; S_2+2:The pulse output D-value between center coordinate and present position .When we use radius mode ,the value must be 0x7FFFFFFF.The range is -2,147,483,648 ～ 2,147,483,647.
- S_3: The output frequency of the synthesis;
- D_1: High speed pulse output port, only Y0 can be specified, occupy 3 continue address (Y0, Y1, and Y2);
- D_2: The operating direction output port, occupy 2 continue address;

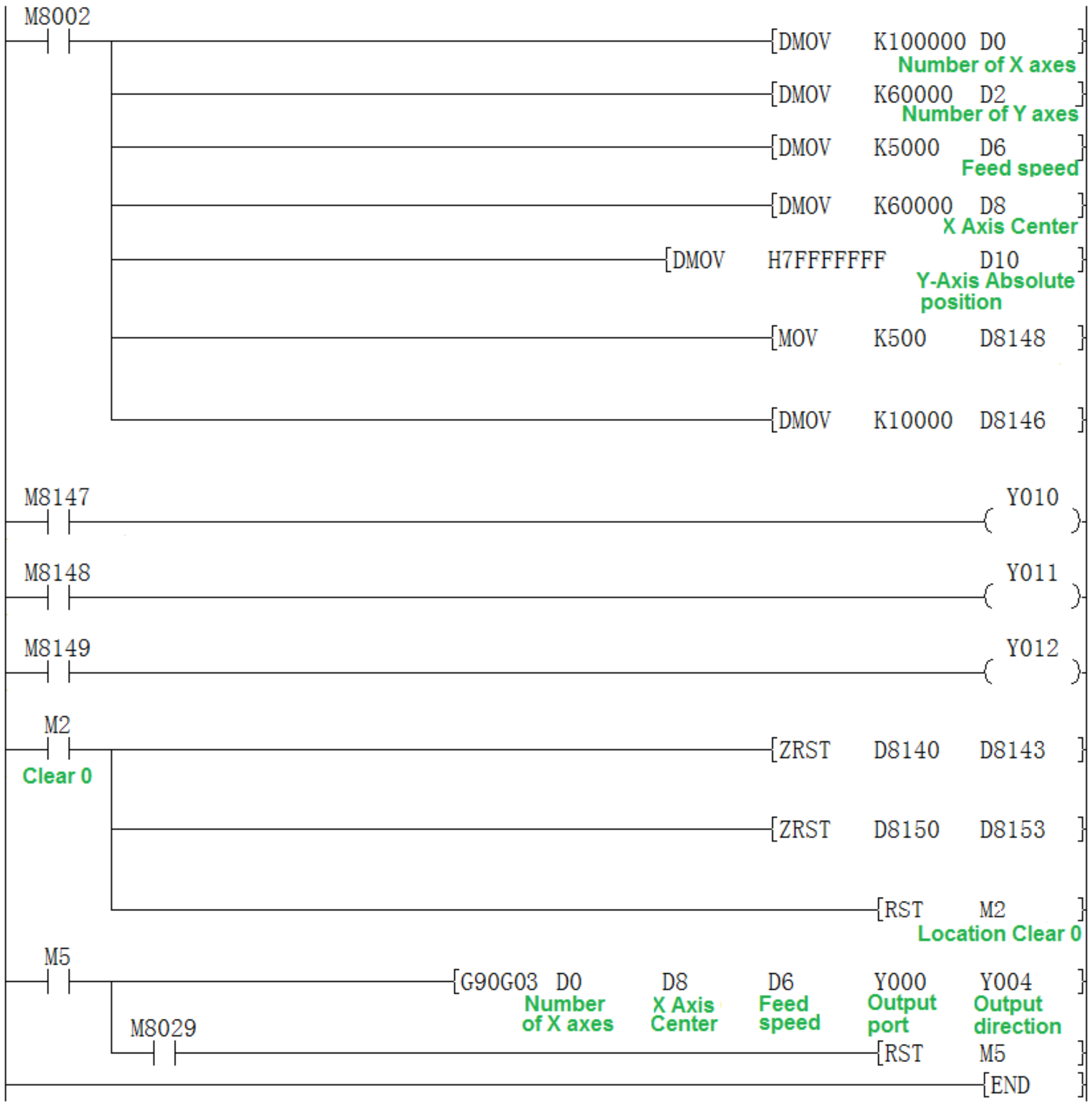
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2														√		
S_3					√	√								√		
D_1		√														
D_2		√	√													

2) Program example

The system variables when execution of instruction:

Instruction	G90G03
Function	Absolute position anticlockwise circular interpolation

Axis	2	
Output port	Y0	Y1
Present position (double byte)	D8140	D8142
Time of ACC/DEC (byte)	D8148	
	D8104	D8105
Basic velocity (double)	D8145	
Maximum velocity (double byte)	D8146	
Pulse output interrupt	M8145	M8146
BUSY/ READY	M8147	M8148
ACC/ DEC	Trapezium AC/DE	



3) Note for use

a) The arc should be more than 20 pulses in circular interpolation, otherwise there will be error;

b) The maximum radius of circular interpolation support is 8000000 pulse;

c) The selection of S_2 have two modes: IJ mode (center coordinates) and R mode(radius mode).When set the value of S_2+2 as 0x7FFFFFFF,it is R mode (radius mode), otherwise it is IJ mode (center coordinates) mode .

d) IJ mode: S_1 only express the relative position between present position and center coordinates regardless absolute or relative interpolation mode. It should be calculated with pulse deviation value.

e) R mode (radius mode),when the value of R is positive, it is express a circular arc less than 180 degrees ;inversely ,it is express a circular arc greater than 180 degrees .In R mode , you can't generate the full circle ,because it has infinitely solution.

f) When S_1 express the relative address of target position, the target position should be logical to insure create a correct path of target of arc .When both S_1+0 and S_1+2 equal 0, it will generate a full circle.

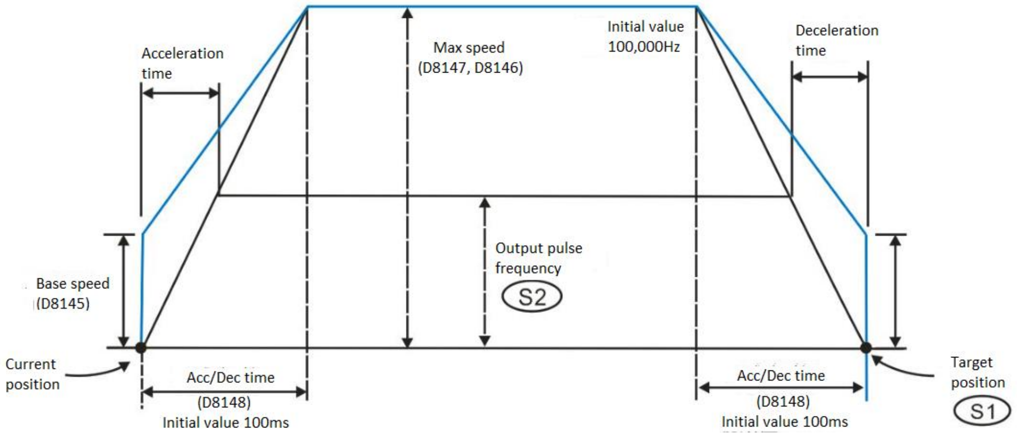
g) When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

h) The actuality output synthesized frequency (lowest frequency),the computation formula is as follows:

$$V_{min}=\sqrt{\frac{\text{Maximum operating frequency(D8146)}}{2^{*}\text{ACC\&DEC time/1000}}}$$

i) The output frequency range of interpolation (not synthesized frequency):10~100 KHz;

j) Frequency calculation:



G91G03 instruction

1) Instruction description

Name	Function	Bits (bits)	Pulse type	Instruction format	Step
G91G03	Relative position anticlockwise circular interpolation	16	No	G91G03 S_1 S_2 S_3 D_1 D_2	21

- S_1: Target position, will occupy 6 continue address. The range is -2147483648 ~ 2417482647. Target position can be specified by a absolution address; S_1: Target position of X axis, S_1+2: Target position of Y axis.
- S_2: Radius/r, occupy 4 bit variable in the continue address .The radius will always treated as relative address; S_2+0:The pulse output D-value between center coordinate and present position ,or the pulse amount of radius “R"; S_2+2:The pulse output D-value between center coordinate and present position .When we use radius mode ,the value must be 0x7FFFFFFF.The range is -2,147,483,648 ～ 2,147,483,647.
- S_3: The output frequency of the synthesis;
- D_1: High speed pulse output port, only Y0 can be specified, occupy 3 continue address (Y0, Y1, and Y2);
- D_2: The operating direction output port, occupy 2 continue address;

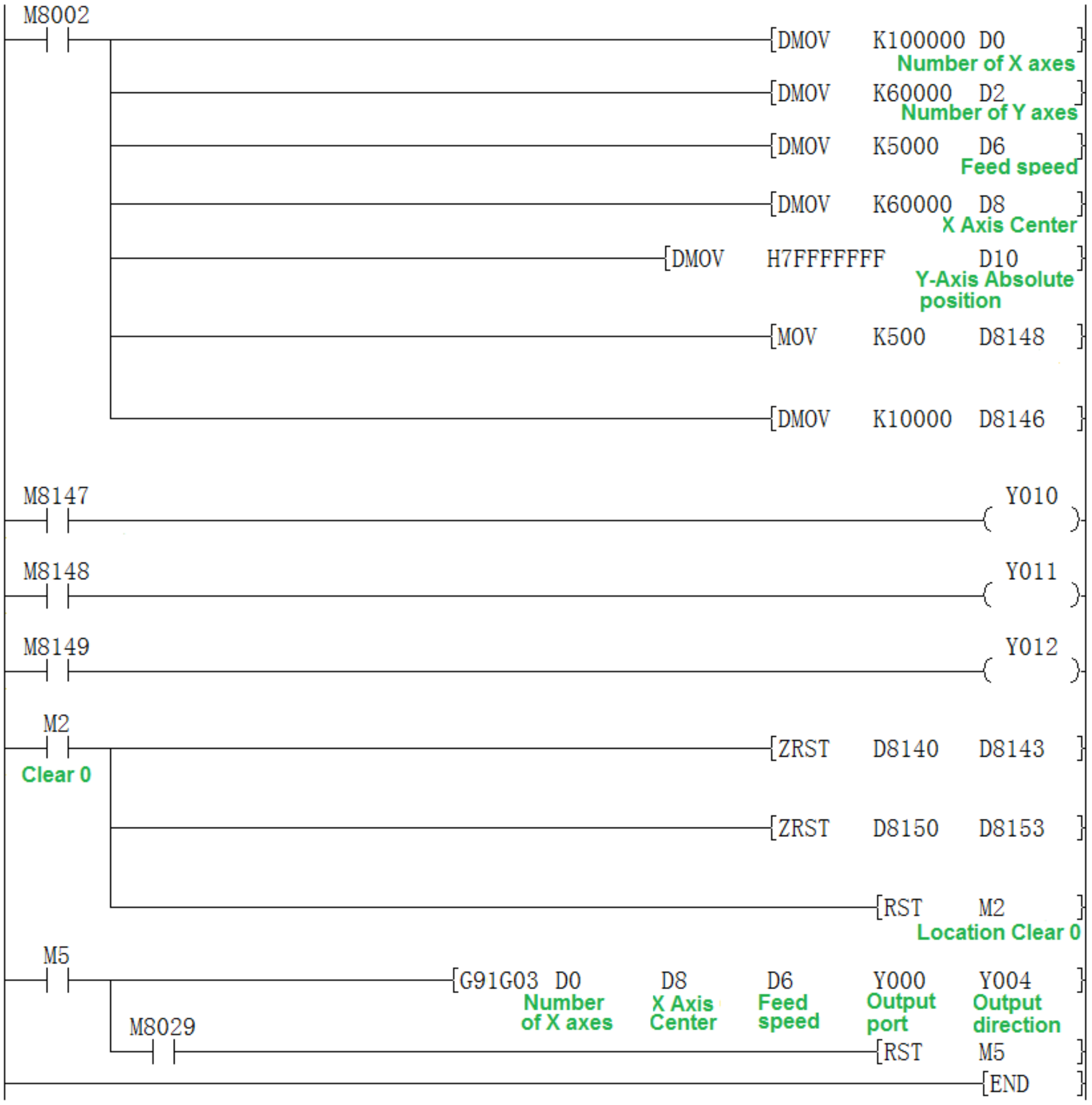
Operand	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1														√		
S_2														√		
S_3					√	√								√		
D_1		√														
D_2		√	√													

2) Program example

The system variables when execution of instruction:

Instruction	G91G03
Function	Absolute position clockwise circular interpolation
Axis	2

Output port	Y0	Y1
Present position (double byte)	D8140	D8142
Time of ACC/DEC (byte)	D8148	
	D8104	D8105
Basic velocity (double)	D8145	
Maximum velocity (double byte)	D8146	
Pulse output interrupt	M8145	M8146
BUSY/ READY	M8147	M8148
ACC/ DEC	Trapezium AC/DE	



3) Note for use

a) The arc should be more than 20 pulses in circular interpolation, otherwise there will be error;

b) The maximum radius of circular interpolation support is 8000000 pulse;

c) The selection of S_2 have two modes: IJ mode (center coordinates) and R mode(radius mode).When set the value of S_2+2 as 0x7FFFFFFF,it is R mode (radius mode), otherwise it is IJ mode (center coordinates) mode .

d) IJ mode: S_1 only express the relative position between present position and center coordinates regardless absolute or relative interpolation mode. It should be calculated with pulse deviation value.

e) R mode (radius mode),when the value of R is positive, it is express a circular arc less than 180 degrees ;inversely ,it is express a circular arc greater than 180 degrees .In R mode , you can't generate the full circle ,because it has infinitely solution.

f) When S_1 express the relative address of target position, the target position should be logical to insure create a correct path of target of arc .When both S_1+0 and S_1+2 equal 0, it will generate a full circle.

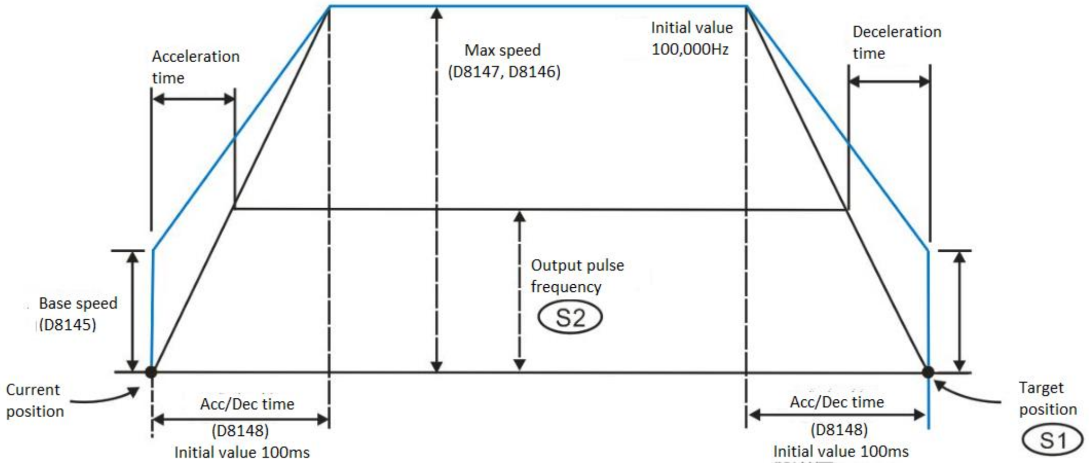
g) When using the interpolation instruction, parameter settings (such as celebration/deceleration time and so on) are subject to the X axis (Y0);

h) The actuality output synthesized frequency (lowest frequency),the computation formula is as follows:

$$V_{min}=\sqrt{\frac{\text{Maximum operating frequency(D8146)}}{\text{2*ACC\&DEC time/1000}}}$$

i) The output frequency range of interpolation (not synthesized frequency):10~100 KHz;

j) Frequency calculation:



5.2.15 Compare instruction

LD compare instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
LD=	Active when S_1=S_2	16	No	LD S_1 S_2 compare instructions include =, >, <, >=, <=, <>	5
LDD=	Active when S_1=S_2	32	No		9
LD>	Active when S_1>S_2	16	No		5
LDD>	Active when S_1>S_2	32	No		9
LD<	Active when S_1<S_2	16	No		5
LDD<	Active when S_1<S_2	32	No		9
LD<>	Active when S_1≠S_2	16	No		5
LDD<>	Active when S_1≠S_2	32	No		9
LD<=	Active when S_1<=S_2	16	No		5
LDD<=	Active when S_1<=S_2	32	No		9
LD>=	Active when S_1>=S_2	16	No		5
LDD>=	Active when S_1>=S_2	32	No		9

The value of S_1 and S_2 are tested according to the comparison of the instruction. If the comparison is true then the LD contact is active. If the comparison is false then the LD contact is not active.

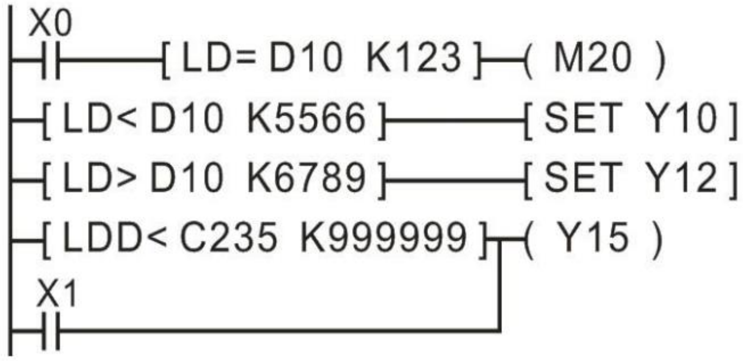
- S_1: The source data or variable data 1 for comparison;
- S_2: The source data or variable data 2 for comparison;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√

2) Program example

- If the content of D10=K123 and X0=ON, then M20 set ON;
- If the content of D10<K5566, then Y10 set ON and holds;
- If the content of D10>K6789, then Y12 set ON and holds;

- If the content of C235<K999999 or X1=ON, then Y15 set ON;
- If the operands are 32-bit counters, an error occurs if users don't use 32-bit LD instruction. C200~C255 are 32-bit counters.



AND compare instruction

1) Instruction description

Name	Function	Bits(bits)	Pulse type	Instruction format	Step
AND=	Active when S_1=S_2	16	No	AND S_1 S_2 compare instructions include =, >, <, >=, <=, <>	5
ANDD=	Active when S_1=S_2	32	No		9
AND>	Active when S_1>S_2	16	No		5
ANDD>	Active when S_1>S_2	32	No		9
AND<	Active when S_1<S_2	16	No		5
ANDD<	Active when S_1<S_2	32	No		9
AND<>	Active when S_1≠S_2	16	No		5
ANDD<>	Active when S_1≠S_2	32	No		9
AND<=	Active when S_1<=S_2	16	No		5
ANDD<=	Active when S_1<=S_2	32	No		9
AND>=	Active when S_1>=S_2	16	No		5
ANDD>=	Active when S_1>=S_2	32	No		9

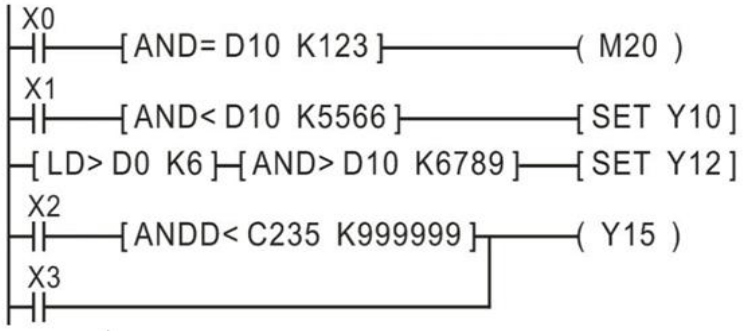
The value of S_1 and S_2 are tested according to the comparison of the instruction. If the comparison is true then the AND contact is active. If the comparison is false then the AND contact is not active.

- S_1: The source data or variable data 1 for comparison;
- S_2: The source data or variable data 2 for comparison;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√

2) Program example

- When X0=ON and D10=K123, then M20 set ON;
- When X1=ON and D10<K5566, then Y10 set ON and holds;
- When D0>K6 and D10>K6789, then Y12 set ON and holds;
- When X2=ON and C235<K999999, or X3=ON, then Y15 set ON;
- If the operands are 32-bit counters, an error occurs if users don’t use 32-bit LD instruction. C200~C255 are 32-bit counters.


X0
[AND= D10 K123]
(M20)
X1
[AND< D10 K5566]
[SET Y10]
[LD> D0 K6]
[AND> D10 K6789]
[SET Y12]
X2
[ANDD< C235 K999999]
(Y15)
X3

OR compare instruction

1) Instruction description

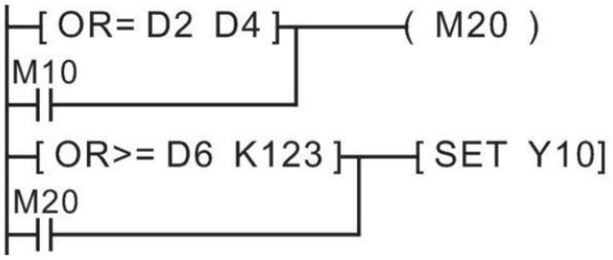
Name	Function	Bits(bits)	Pulse type	Instruction format	Step
OR=	Active when S_1=S_2	16	No	OR S_1 S_2 compare instructions include =, >, <, >=, <=, <>	5
ORD=	Active when S_1=S_2	32	No		9
OR>	Active when S_1>S_2	16	No		5
ORD>	Active when S_1>S_2	32	No		9
OR<	Active when S_1<S_2	16	No		5
ORD<	Active when S_1<S_2	32	No		9
OR<>	Active when S_1≠S_2	16	No		5
ORD<>	Active when S_1≠S_2	32	No		9
OR<=	Active when S_1<=S_2	16	No		5
ORD<=	Active when S_1<=S_2	32	No		9
OR>=	Active when S_1>=S_2	16	No		5
ORD>=	Active when S_1>=S_2	32	No		9

The value of S_1 and S_2 are tested according to the comparison of the instruction. If the comparison is true then the OR contact is active. If the comparison is false then the OR contact is not active.

- S_1: The source data or variable data 1 for comparison;
- S_2: The source data or variable data 2 for comparison;

Operands	Bit device				Word device											
	X	Y	M	S	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					√	√		√	√	√	√	√	√	√	√	√
S_2					√	√		√	√	√	√	√	√	√	√	√

2) Program example



- WhenM10=ON, or D2=ON, then M20 set ON;
- When M20=ON or D6>=K123, then Y10 set ON and holds;
- If the operands are 32-bit counters, an error occurs if users don’t use 32-bit LD instruction. C200~C255 are 32-bit counters.

5.3 Step control instructions

STL, RET instructions

1) Instruction description

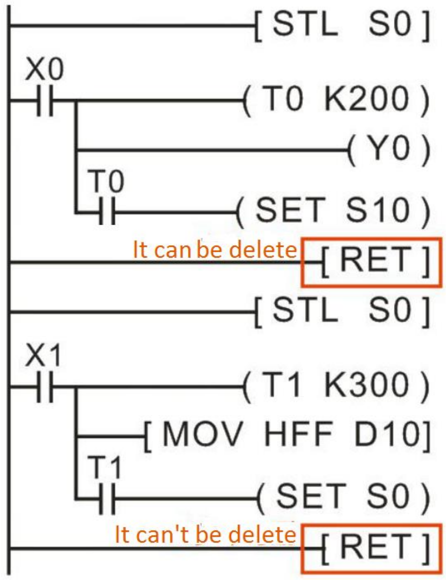
Name	Function	Device	Step
STL	STL programming start instruction	S	1
RET	STL programming end instruction	--	1

Step Control (STL) is controlled by several operating procedures (S0, S1……Sn).

Step Control method's feature is that after taken into considerations for each control step and divided the complex procedure into successive steps, it greatly reduces the interdependence between each step and the complexity involved in programming.

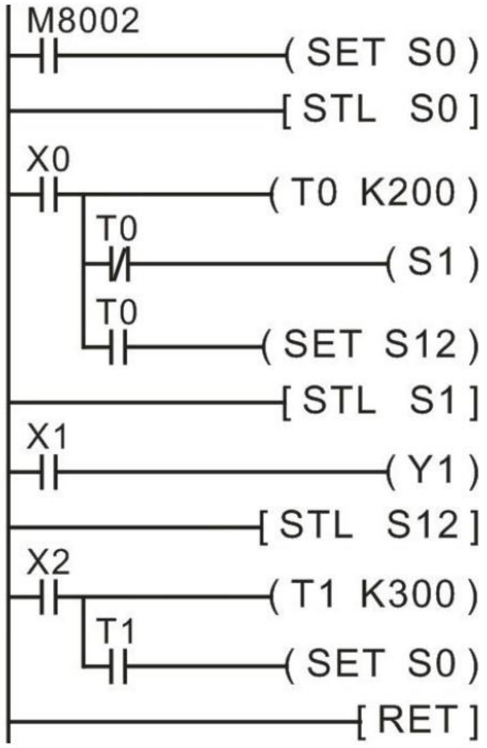
2) Program example

Example 1



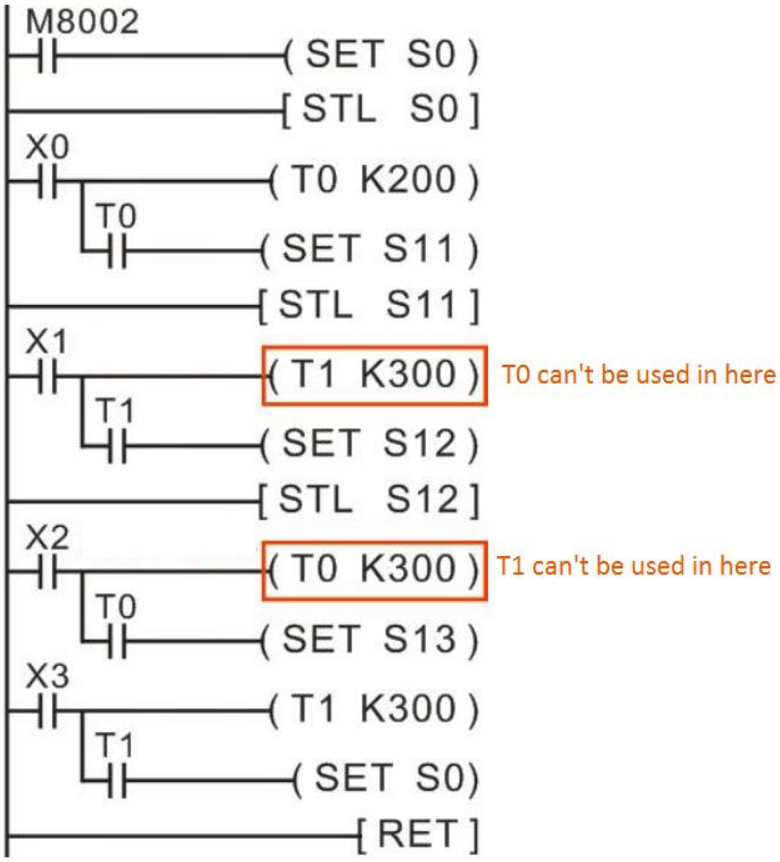
In example 1, RET will be omitted between each step procedures. Therefore, it will seem a RET is shared by several STL. When STL is programmed and RET procedure is not, error message will appear.

Example 2



State transfer can only use the SET instruction, cannot use OUT instructions.

Example 3



Time relay T can be reused, but the adjacent two states cannot be reused using the same time relay.

3) Note for use

- STL---RET instructions cannot be used in sub-programs.
- When transition is happening from current status (S0) to next status (S1), the actions under the two scanning cycle conditions will both be executed; when the next scanning cycle is being executed, current status (S0) will be reset by the next status (S1), and the actions under the current status (S0) will not be executed. All OUT components' inputs will be interrupted.
- Generally speaking, RET will be omitted between each step procedures. Therefore, it will seem a RET is shared by several STL. When STL is programmed and RET procedure is not, error message will appear.

6. Shortcut list

6.1 Common shortcuts list

The following table lists the common shortcuts.

Shortcuts	Corresponding menu	Description
Ctrl + N	New	Create a new project
Ctrl + O	Open	Open an existing project
Ctrl + S	Save	Save the project
Ctrl + X	Cut	Cut the selected data
Ctrl + C	Copy	Copy the selected data
Ctrl + V	Paste	Paste the cut/copied data at the cursor position
Ctrl + Z	undo	Cancel the previous operation
Ctrl + Y	Redo	Perform the operation canceled by [Undo]
Ctrl + F	Find Device	Search for a device
Ctrl + F1	Show/Hide toolbar menu	Show/hide toolbar menu
F3	Start monitoring	Start monitoring the window being operated.
Ctrl + F3	Stop monitoring	Stop monitoring the window being operated
F4	Transform/transform + compilation	Compile (Transform) current program
Alt+F4	Exit	Close the project being edited and exits WECON PLC Editor

6.2 Shortcuts list in programming area

The following table lists the shortcuts in programming area.

Shortcuts	Corresponding menu	Description
F5	Open contact	Insert an open contact at the cursor position
Shift + F5	Open branch	Insert an open contact branch at the cursor position
F6	Close contact	Insert a closed contact at the cursor position
Shift + F6	Close branch	Insert a closed contact branch at the cursor

		position
F7	Coil	Insert a coil at the cursor position
F8	Application instruction	Insert an application instruction at the cursor position
F9	Horizontal line	Insert a horizontal line at the cursor position
F11	Vertical line	Insert a vertical line at the cursor position
Ctrl + F9	Delete horizontal line	Delete the horizontal line at the cursor position
Ctrl + F11	Delete vertical line	Delete the vertical line at the cursor position
Shift + F7	Rising pulse	Insert a rising pulse at the cursor position
Shift + F8	Falling pulse	Insert a falling pulse at the cursor position
Ctrl + Alt + F7	Rising pulse branch	Insert a rising pulse branch at the cursor position
Ctrl + Alt + F8	Falling pulse branch	Insert a falling pulse branch at the cursor position
Ctrl + Alt + F11	Invert operation results	Insert an operation result inversion at the cursor position
Ctrl + Shift + Insert	Insert line statement	Insert statement line statement at the cursor position
Shift + Insert	Insert row	Insert a row at the cursor position
Shift + Delete	Delete row	Delete the row at the cursor position
Ctrl + Insert	Insert column	Insert a column at the cursor position
Ctrl + Delete	Delete column	Delete the column at the cursor position
Ctrl + →	Enter/Delete HLine rightward	Enter/delete a line at the right of the cursor position
Ctrl + ←	Enter/Delete HLine leftward	Enter/delete a line at the left of the cursor position
Ctrl + ↓	Enter/Delete VLine downward	Enter/delete a line at the downward of the cursor position
Ctrl + ↑	Enter/Delete VLine upward	Enter/delete a line at the upward of the cursor position
Ctrl + /	Switch open/close contact	Switch an open contact to closed contact, and vice versa

Ctrl + G	Jump	Display the specified row
Ctrl + F5	Comment	Display device comments
Ctrl + F7	Statement	Display statements
F1	Open the instructions help	Display the instructions help

7. Communication example

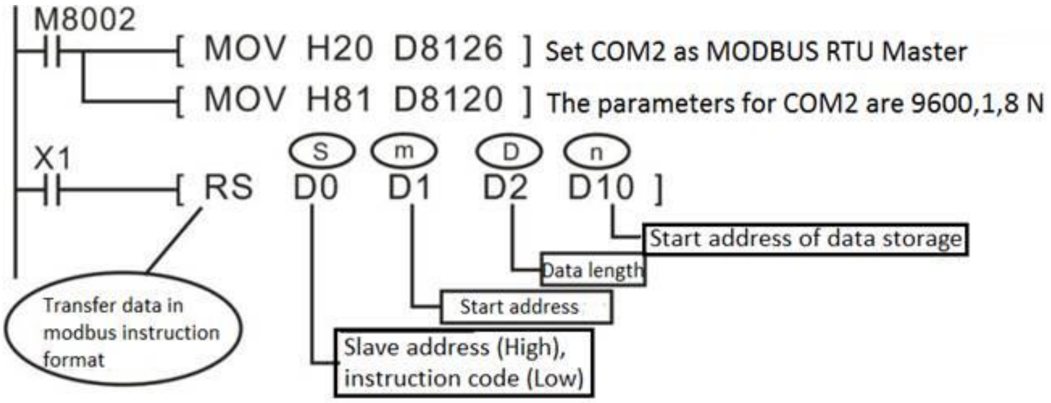
7.1 MODBUS communication

The communication application of MODBUS master station

Communicational port COM 2 in series of LX PLC can be used for MODBUS-RTU and MODBUS-ASCII protocol by setting value in D8126. Users can use RS instruction to achieve MODBUS RTU and ASCII protocol.

1) Instruction description

Write H20 to D8126, it configures MODBUS-RTU master protocol to COM2. RS instruction will send data in the MODBUS protocol format. In process of communication, engrossed register definition is different from standard RS instruction, please pay attention to it:


M8002
[MOV H20 D8126] Set COM2 as MODBUS RTU Master
[MOV H81 D8120] The parameters for COM2 are 9600,1,8 N
X1
[RS D0 D1 D2 D10]
S
m
D
n
Start address of data storage
Data length
Start address
Slave address (High), instruction code (Low)
Transfer data in modbus instruction format

The definitions of each operand in the RS (MODBUS mode) instruction are different from those of a standard RS instruction (user-defined protocol).

- S: Slave address (high byte), communication command (low byte, defined by MODBUS protocol);
- m: Start address of accessing slave;
- D: Data length, unit: word;
- n: Start address of data storage, the take up length of the subsequent address defined by D;

In RS (MODBUS mode) instruction, variable type that each of operand support are as following table:

Operands	Word device											
	K	H	E	KnX	KnY	KnM	KnS	T	C	D	V	Z
S										√		
m	√	√								√		
D										√		
n										√		

It requires setting some parameters before executing RS instruction, once started, the system will automatically calculate the CRC check, organize the communication frame to finish sending data, receive operation.

The HEX-ASC format conversion of sending and receiving data is done automatically by the PLC system program. The user's method of using the RS (MODBUS mode) instruction is exactly the same as that of using the MODBUS-RTU protocol.

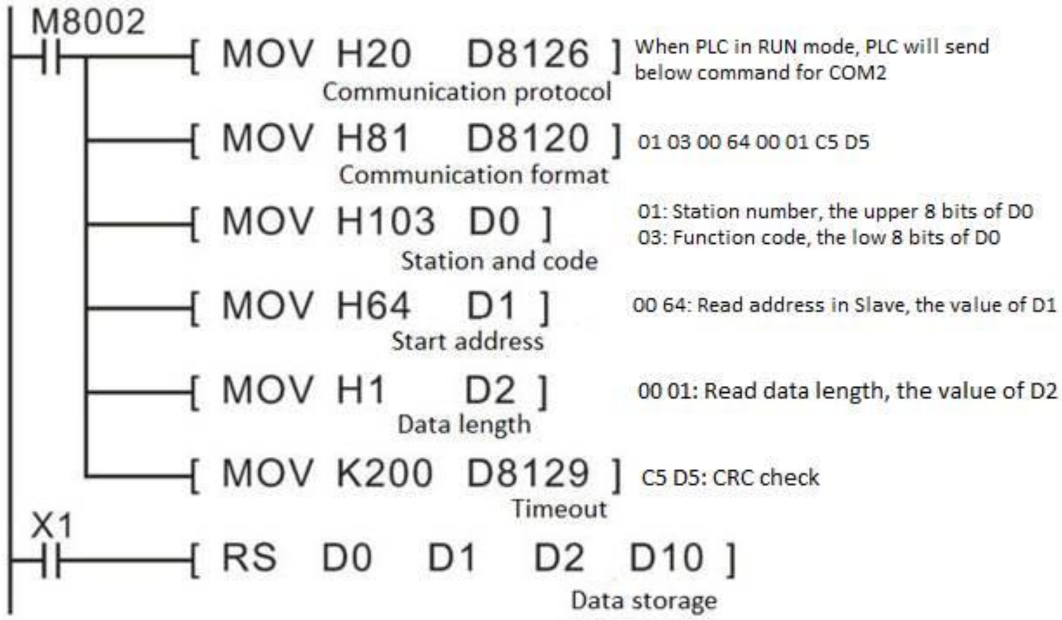
[Function code in MODBUS Master]

Function code	Function	Description
0x01	Coil read	Coil read out (continuous operation)
0x02	Word register read	Input read (continuous operation)
0x03	Latched register read	Latched register read (continuous operation)
0x04	Register read	Register read (continuous operation)
0x05	Coil write	Coil write (single coil)
0x06	Register write	Register write (single register)
0x0F	Write continuous coils	Write continuous coils
0x10	Write continuous register	Write continuous register

2) Program example

When X1 is triggered, PLC reads data from address 100 in the Slave 1, and stores data in D10.

In the user program, the fewer RS (MODBUS) commands to be executed cyclically, the more frequently the communication data is updated, the faster the refresh rate of readings is, the real-time performance is improved, and the reading frequency of some non-important parameters can be reasonably arranged to improve the communication effect.



```
M8002
[ MOV H20   D8126 ]
[ MOV H81   D8120 ]
[ MOV H103  D0 ]
[ MOV H64   D1 ]
[ MOV H1    D2 ]
[ MOV K200  D8129 ]
X1
[ RS  D0  D1  D2  D10 ]
```

When PLC in RUN mode, PLC will send below command for COM2

01 03 00 64 00 01 C5 D5

01: Station number, the upper 8 bits of D0
03: Function code, the low 8 bits of D0

00 64: Read address in Slave, the value of D1

00 01: Read data length, the value of D2

C5 D5: CRC check

The use of special variables M8129, can also determine the communication timeout fault, you can make the appropriate protection or alarm processing.

The communication application of MODBUS slave station

In some industrial applications, the PLC controller, as part of an industrial automation system, is monitored by the automation control network; in this case, the communication port of PLC needs to communicate with the host computer by MODBUS slave protocol.

1) Modbus Slave setting

M8120	Reserved	D8120	Com2 port setting, the default value is 0
M8121	Sending and waiting (RS instruction)	D8121	Station number settings, the default value is 1
M8122	Sending flag (RS instruction) Instruction execution status (MODBUS)	D8122	Amount of remaining data to be transmitted (Only for RS instruction) unit:0.1ms
M8123	Receiving complete flag (RS) Communication error flag (MODBUS)	D8123	Amount of data already received (Only to RS instruction)
M8124	Receiving (only to RS instruction)	D8124	Start character STX (Only to RS instruction)
M8125	Reserved	D8125	End character ETX (Only to RS instruction)
M8126	Reserved	D8126	Communication protocol setting, the default value is 0
M8127	Reserved	D8127	Starting address for PC protocol
M8128	Reserved	D8128	Data length for PC protocol
M8129	Timeout judgement	D8129	Timeout judgement, default value is 10 (100ms)

2) Communication Format (D8120)

Item	Parameter	Bit value of D8120							
		b7	b6	b5	b4	b3	b2	b1	b0
Baud rate (Bps)	115200	1	1	0	0	-	-	-	-
	57600	1	0	1	1	-	-	-	-
	38400	1	0	1	0	-	-	-	-

	19200	1	0	0	1	-	-	-	-
	9600	1	0	0	0	-	-	-	-
	4800	0	1	1	1	-	-	-	-
Stop bit	1 bit	-	-	-	-	0	-	-	-
	2 bit	-	-	-	-	1	-	-	-
Parity	None	-	-	-	-	-	0	0	-
	Odd	-	-	-	-	-	0	1	-
	Even	-	-	-	-	-	1	1	-
Data bit	7 bit	-	-	-	-	-	-	-	0
	8 bit	-	-	-	-	-	-	-	1
Example: the communication format is 9600.1.8.None, b7b6b5b4=1000, b3=0, b2b1=00, b 0=1. D8120=81H ($(10000001)_2$=81H, 81H means hexadecimal number)									

3) Modbus Slave operation

LX3V Series PLC as MODBUS slave station, supports MODBUS 0x01,0x03,0x05,0x06,0x0f, 0x10 and other communication operation function codes; through these codes, can read and write PLC coil M, S, T, C, X Read), Y and other variables; register variables have D, T, C.

a) Function code 0x01(01): read coil (bit address)

Frame format: Station number of slave&0x01 + start address + number of coils + CRC

No.	Data	Number of byte	Instruction
1	Station number of slave	1 byte	Value range 1~247, set by D8121
2	0x01(function code)	1 byte	Read coil
3	Start address	2 bytes	
4	Number of coils	2 bytes	
5	CRC	2 bytes	

b) Function code 0x03(03): read register (word address)

Frame format: Station number of slave&0x03 + start address+ number of registers + CRC

No.	Data	Number of byte	Instruction
1	Station number of slave	1 byte	Value range 1~247, set by D8121
2	0x03 (function code)	1 byte	Read register
3	Start address	2 bytes	
4	Number of registers	2 bytes	
5	CRC	2 bytes	

c) Function code 0x05(05): write single coil

Frame format: Station number of slave&0x05 + address + state of coil + CRC

No.	Data	Number of byte	Instruction
1	Station number of slave	1 byte	Value range 1~247, set by D8121
2	0x05 (function code)	1 byte	Write single coil
3	address	2 bytes	
4	State of coil	2 bytes	
5	CRC	2 bytes	

d) Function code 0x06 (06): Write single register

Frame format: Station number of slave&0x06 + address + value + CRC

No.	Data	Number of byte	Instruction
1	Station number of slave	1 byte	Value range 1~247, set by D8121
2	0x06 (function code)	1 byte	Write single register
3	address	2 bytes	
4	Value of register	2 bytes	
5	CRC	2 bytes	

e) Function code 0x0f(15): Write continuous coils

Frame format: Station number of slave&0x0f + start address + number of coils + length + state of coil + CRC

No.	Data	Number of byte	Instruction
1	Station number of slave	1 byte	Value range 1~247, set by D8121
2	0x0f (function code)	1 byte	Write contunuous coils
3	Start address	2 bytes	
4	Number of coil	2 bytes	
5	Length	1 bytes	
6	State of coils	[(N+7)/8] bytes	
7	CRC	2 bytes	

f) Function code 0x10 (10): Write continuous registers

Frame format: Station number of slave&0x10 + start address + number of registers + length + value of register + CRC

No.	Data	Number of byte	Instruction
1	Station number of slave	1 byte	Value range 1~247, set by D8121
2	0x10 (function code)	1 byte	Write continuous registers
3	Start address	2 bytes	
4	Number of registers	2 bytes	

5	Length	1 bytes	
6	Value of register	N*2 bytes	
7	CRC	2 bytes	

4) WECON PLC - MODBUS (Slave) addresses rules

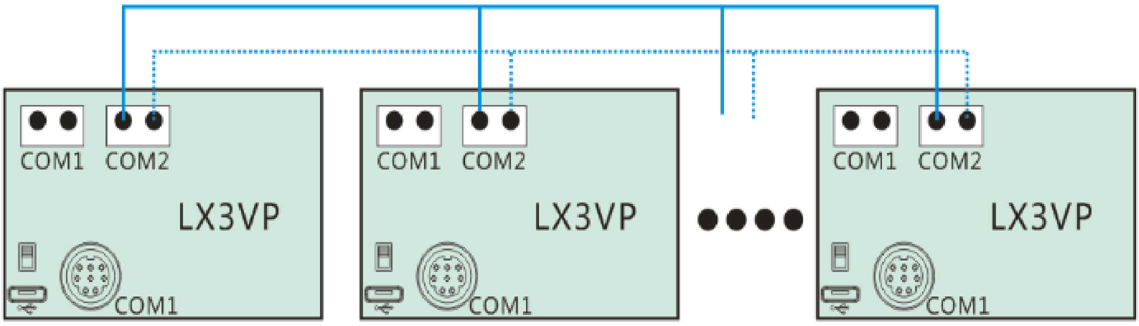
PLC Bit Address		
PLC Address	MODBUS Address	
	Hex	Decimal
M0 ~ M3071	0 ~ 0xBFF	0 ~ 3071
M8000 ~ M8256	0x1F40 ~ 0x2040	8000 ~ 8256
S0 ~ S999	0xE000 ~ 0xE3E7	57344 ~ 58343
T0 ~ T256	0xF000 ~ 0xF100	61440 ~ 61696
C0 ~ C255	0xF400 ~ 0xF4FF	62464 ~ 62719
X0 ~ X255	0xF800 ~ 0xF9FE	63488 ~ 63998
Y0 ~ Y255	0xFC00 ~ 0xFDFE	64512 ~ 65022
PLC Word Address		
PLC Address	MODBUS Address	
	Hex	Decimal
D0 ~ D8255	0 ~ 0x203F	0 ~ 8255
T0 ~ T255	0xF000 ~ 0xF0FF	61440 ~ 61695
C0 ~ C199	0xF400 ~ 0xF4C7	62464 ~ 62663
C200 ~ C255	0xF700 ~ 0xF7FF	63232 ~ 63487

7.2 N: N network

LX3VP COM2 N:N Application

1) LX3VP COM2 N:N Connection

PLC built-in N:N connection protocol provides a effective way to exchange data among multiple PLC (Max. 8 devices). Technically, it only requires the twisted pair RS485 cable to connect with each PLC COM2 (in parallel).



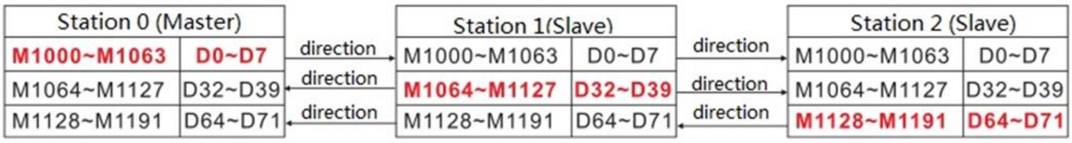
2) COM2 N:N Instructions

User should put data to specified register unit. Exchanging the data among PLCs, This function only requires putting data in preset register block, all the data in this register block would share with other PLCs. There are five modes for choice, according to data volume and communication speed. See the preset register block from table below.

Station Number		Mode 0		Mode 1		Mode 2	
		Bit (M)	Word(D)	Bit (M)	Word(D)	Bit (M)	Word(D)
		0	4	32	4	64	8
Master	NO. 0	--	0-3	1000-1031	0-3	1000-1063	0-7
Slave	NO. 1	--	32-35	1064-1095	32-35	1064-1127	32-39
	NO. 2	--	64-67	1128-1159	64-67	1128-1191	64-71
	NO. 3	--	96-99	1192-1223	96-99	1192-1255	96-103
	NO. 4	--	128-131	1256-1287	128-131	1256-1319	128-135

	NO. 5	--	160-163	1320-1351	160-163	1320-1383	160-167
	NO. 6	--	192-195	1384-1415	192-195	1384-1447	192-199
	NO. 7	--	224-227	1448-1479	224-227	1448-1511	224-231
Station Number		**Mode 3**		**Mode 4**			
		Bit (M)	**Word(D)**	**Bit (M)**	**Word(D)**		
		64	**16**	**64**	**32**		
Master	NO. 0	1000-1063	0-15	1000-1063	0-31		
Slave	NO. 1	1064-1127	32-47	1064-1127	32-63		
	NO. 2	1128-1191	64-79	1128-1191	64-95		
	NO. 3	1192-1255	96-111	1192-1255	96-127		
	NO. 4	1256-1319	128-143	1256-1319	128-159		
	NO. 5	1320-1383	160-175	1320-1383	160-191		
	NO. 6	1384-1447	192-207	1384-1447	192-223		
	NO. 7	1448-1511	224-239	1448-1511	224-255		

Communication between each PLC (up to 8 PLC), please see the connection construction below (For 3 PLC interconnection).



3) The special devices in N: N network

Register	Description
D8120	Communication format settings
D8126	COM2 communication protocol settings, 40h means N:N Master

	Device, 04h means N:N Slave device
D8176	Station number, from 0 to 7, 0 means master device
D8177	Total station number, from 1 to 7, only required for master device.
D8178	Register block setting, from 0 to 5, only required for master device.
D8179	Retry count settings, only required for master device.
D8180	Timeout setting, unit: 10ms， only required for master device.
D8201	Current connection scan time
D8202	Maximum connection scan time
D8203	Master error counter
D8204~D8210	Slave error counter
D8211	Master N:N error code
D8212~D8218	Slave N:N error code
M8183	Master data transfer sequence error
M8183~M8190	Communication error flag: M8183 - No.0 (Master); M8184 - No. 1 (Slave 1) ......
M8191	Processing sending data

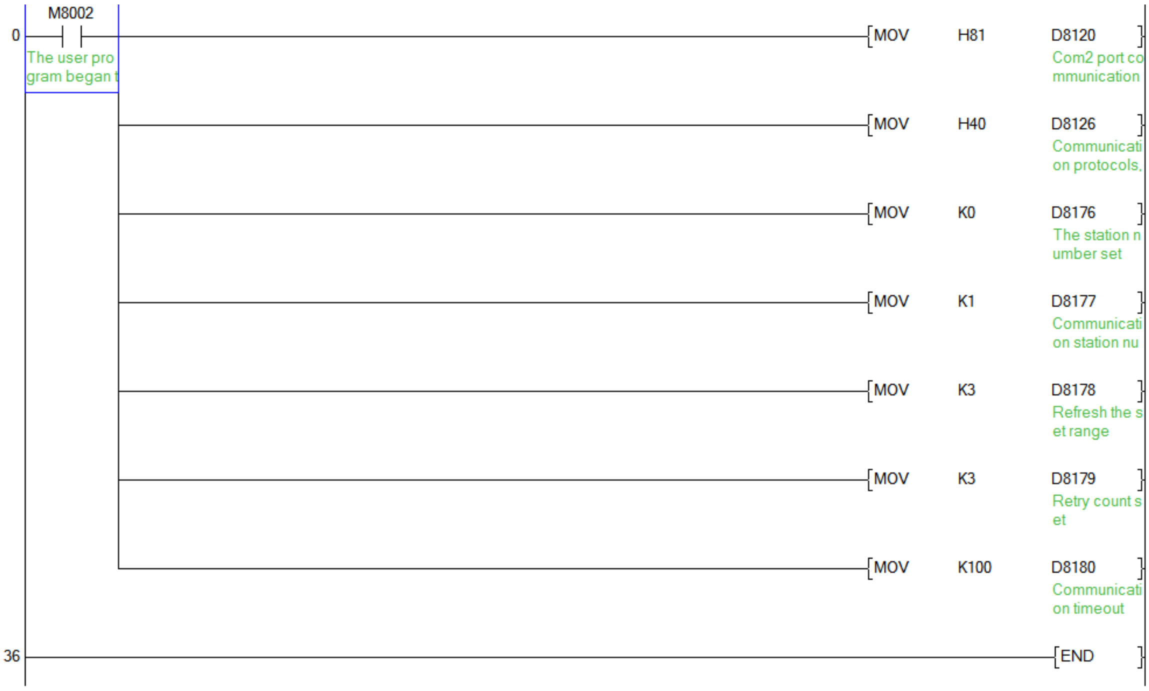
4) Communications format:

Item	Parameters	b15(RS2)	b14-b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit mode	8 bit	0	Retention	-	-	-	-	-	-	-	-
	16 bit	1		-	-	-	-	-	-	-	-
Baud rate (Bps)	115200	-		1	1	0	0	-	-	-	-
	57600	-		1	0	1	1	-	-	-	-
	38400	-		1	0	1	0	-	-	-	-
	19200	-		1	0	0	1	-	-	-	-
	9600	-		1	0	0	0	-	-	-	-
	4800	-		0	1	1	1	-	-	-	-
Stop bit	1 bit	-		-	-	-	-	0	-	-	-
	2 bit	-		-	-	-	-	1	-	-	-
Parity	None	-		-	-	-	-	-	0	0	-
	Odd	-		-	-	-	-	-	0	1	-
	Even	-		-	-	-	-	-	1	1	-
Data bit	7 bit	-		-	-	-	-	-	-	-	0
	8 bit	-		-	-	-	-	-	-	-	1

5) Program example

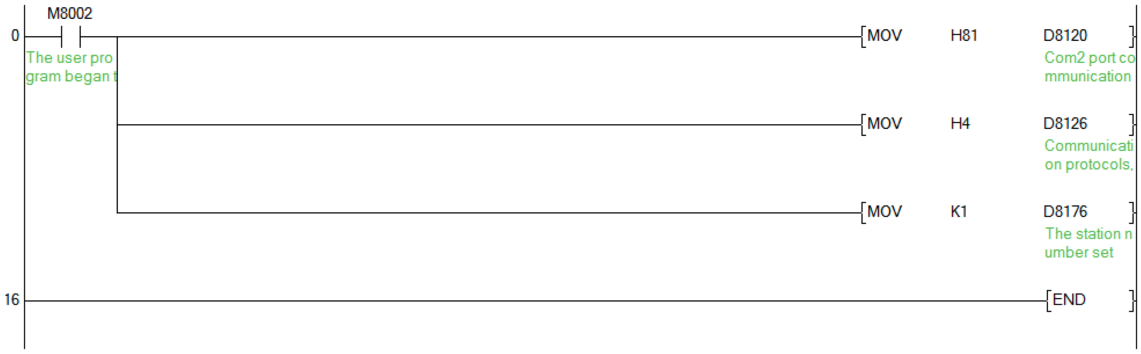
LX3VP COM2 port communication parameters: 9600, 1, 8, NONE. Register block mode: 3

- PLC as master



```
0   M8002                      [MOV  H81   D8120 ]  Com2 port communication
    The user program began     [MOV  H40   D8126 ]  Communication protocols,
                               [MOV  K0    D8176 ]  The station number set
                               [MOV  K1    D8177 ]  Communication station nu
                               [MOV  K3    D8178 ]  Refresh the set range
                               [MOV  K3    D8179 ]  Retry count set
                               [MOV  K100  D8180 ]  Communication timeout
36                             [END              ]
```

- PLC as slave



```
0   M8002                      [MOV  H81   D8120 ]  Com2 port communication
    The user program began     [MOV  H4    D8126 ]  Communication protocols,
                               [MOV  K1    D8176 ]  The station number set
16                             [END              ]
```

6) Note for use

- There are two modes of N:N protocol configuration. one is LX3VP built-in N:N protocol, the other one is LX3V N:N protocol (LX3V-2RS485-BD required).
- In LX3VP series PLC, only one kind of N:N configuration available. Second mode would be disabled when LX3VP built-in N:N protocol configured.

Version: V1.0.1

Date: April 2019