TP03 PLC PROGRAMMING Manual

TP03 Programmable Logic Control

72-000002W **Version: 1.5** **2011-4-29** 0086-0510-8522-7555

Chapter I PLC Component

1 Product summary and corresponding programming language

Product summary

TP03 M type: 20 / 30 points

- Built-in Flash memory (8,000 Steps)
- Retentive data with lithium battery
- Can expand to 128 points
- Can expand 8 channels AD input & 2 channels DA output

TP03 H Type : 20/30 points

- Built-in EEPROM (8,000 Steps) ,Built-in RTC, RS485 communication
- Retentive data and RTC data with lithium battery
- Removable terminal block
- Can expand to 256 points with adding an external power supply
- Can expand 8 channels AD input & 2 channels DA output

TP03 H Type : 40/60 points

- Built-in EEPROM (16,000 Steps) ,Built-in RTC, RS485 communication
- Retentive data and RTC data with lithium battery
- Removable terminal block
- Can expand to 256 points with adding an external power supply
- Can expand 60 channels AD input & 10 channels DA output

TP03 S Type : 14/20/26/36 points

- Built-in EEPROM (4,000 Steps), RS485 communication
- Can expand to 80 points
- Can expand 8 channels AD input & 2 channels DA output

Programming mode

Instruction list (IL) programming

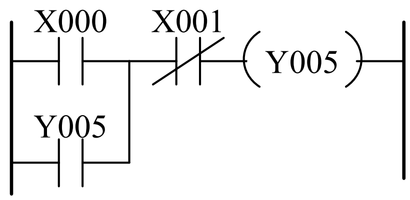
It refers to the sequential control instruction input with LD, AND, OUT and etc., and such mode is the basic input form for programming sequential control program, however, the content is hard to understand.

For example:

Step No.	Instruction	Device
0	LD	X000
1	OR	Y005
2	ANI	X001
3	OUT	Y005

Ladder Logic (LD) programming

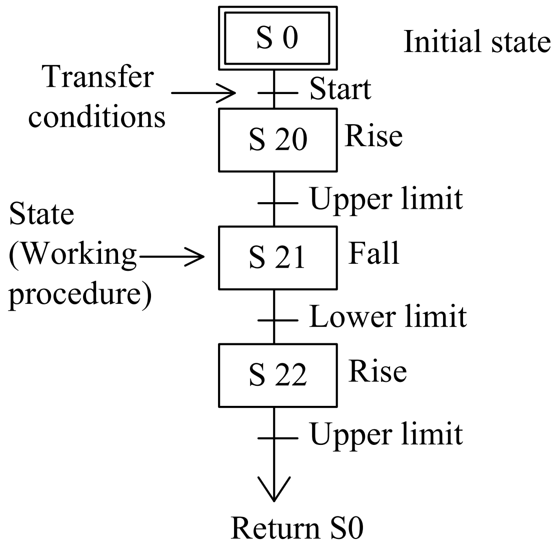
The ladder Logic programming employs sequential control signals and device numbers, and draws sequential control circuit on the drawing. Such method represents sequential control loop with contact symbol and coil symbol, so it is easy to understand the content. Meanwhile, the state displayed by the loop can be used to monitor action of PLC.



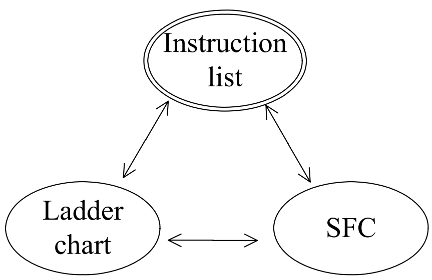
Ladder Logic is used to represent the above instruction list program

Sequential function chart (SFC) programming

SFC programming is the input method for sequential control design according to the mechanical actions. In the peripheral equipment with personal computer and other image, the chart below can be used to determine flow of the sequential control.



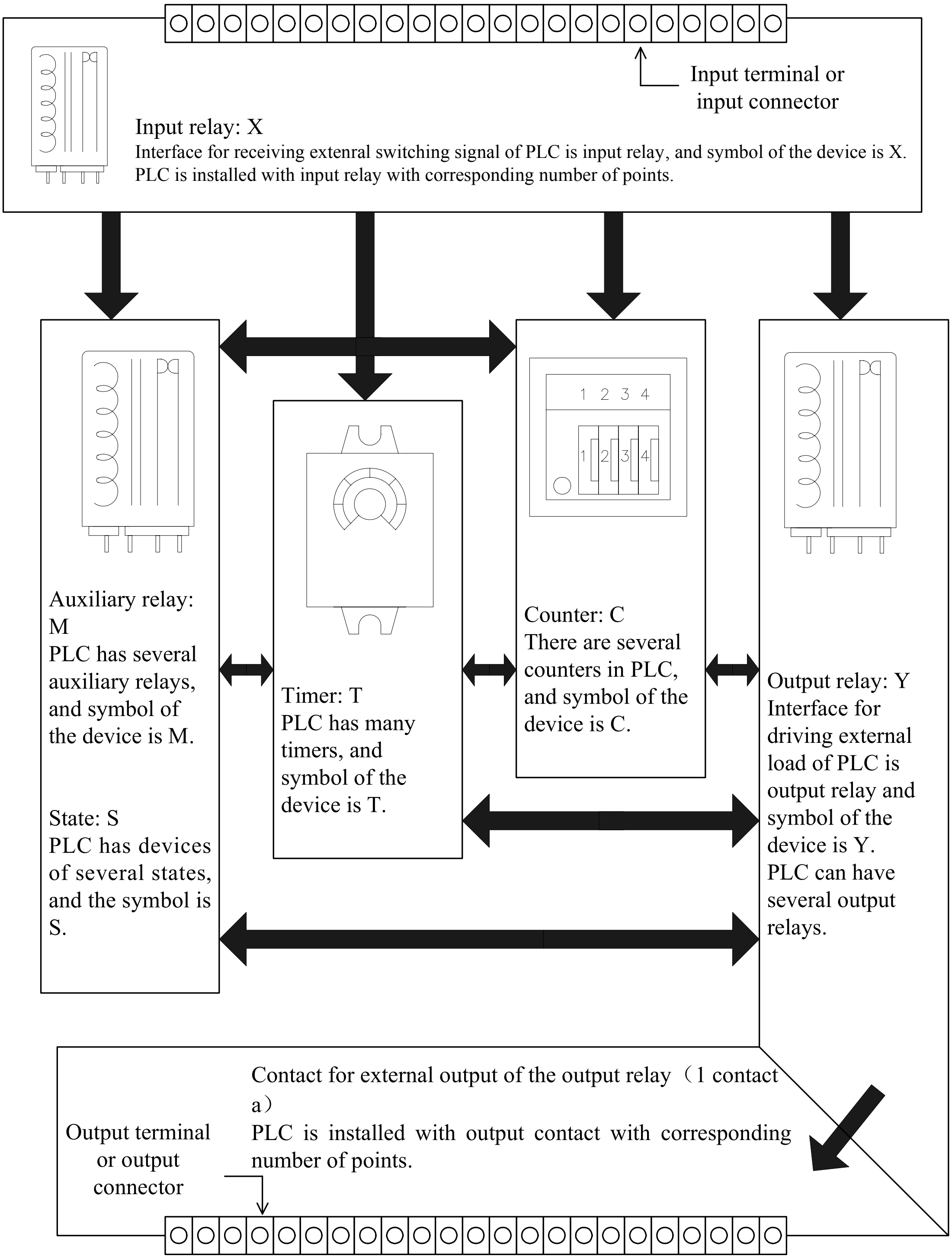
Of the above three programmed sequential control program, they are stored in memory of PLC in instruction list (content of the instruction list), therefore, representation and edition of the program in accordance with chart below can be exchanged (even it is instruction list program, according to SFC conversion rule, devices corresponding to SFC chart can be used to represent program based on instructions).



2 Devices for PLC

There are so many relays, counters and timers in PLC, and they have many contacts a (normally open contact) and contacts b (normally close contact). These contacts and coils make up sequential control loop. The arrow represents signal transfer.

In addition, there is memory device used to store data in PLC-data register (D).



Interpretation for devices:

Input and output relays (X, Y)

Address numbers of input relay, output replay and extension relay are distributed as per X000—X007, X010—X017, …Y000—Y007, Y010—Y017 and etc. in basic units in octal code, which follow the basic units and they are in correspondence with numbers of X and Y in octal code.

In addition to numbers of X and Y in octal code, the following device numbers are in decimal code

Auxiliary relay (M)

The auxiliary relay is the relay in PLC. The relay is different from input and output relays, which can not obtain external input and it can be used in program. Some relays can hold ON/OFF state in case of power failure for PLC.

Step relay (S)

It is the relay used as working step number represented by SFC. When it is not used as working step number, as the auxiliary relay, it can be programmed as common contact or coil. In addition, it can be used as signal alarm for diagnosing external failure.

Timer(T)

The timer executes clock pulses 1ms, 10ms, 100ms and etc. in PLC, when specified setting value is reached, the output contact acts. The timer based on clock pulse can be used to detect 0.001-3276.7 seconds.

For TP03 M/H type, T192-T199,T246~T249 are the special timer for sub-program and program interruption.

For TP03 S type, T196-T199,T246~T249 are the special timer for sub-program and program interruption (See chapter II Device for detail).

Drive input of timer coil of T246-T255 is OFF, the current value continues to act. Other timers are cleared 0.

Counter(C)

The counter is divided into the following types according to different applications.

Internal counting General use/Holding for power failure

16-bit counter: for increasing counting, range of counting: 1-32,767

32-bit counter: for increasing/decreasing counting, range of counting: -2,147,483,648~+2,147,483,647

These counters can be used as internal signal of PLC, with response speed less than 10Hz (0.1s).

High-speed counting Holding for power failure

32-bit counter: for increasing/decreasing counting, range of counting: -2,147,483,648~+2,147,483,647(Single-phase and single counting, single-phase and double-counting, double-phase and double counting) are distributed to input relay. The high-speed counter can perform 100kH counting, which has nothing to do with scan cycle of PLC.

Data registers (D)，(V)，(Z)

The data register is the device for storing data. Data register of PLC is 16-bit (the highest bit is the symbol mark), range of data: -32768~32767. Combine the two registers to execute 32-bit data processing (the highest bit is symbol mark). Range of data: -2,147,483,648~+2,147,483,647. Like other soft devices, the data register is classified for general use and holding in case of power failure.

Of data registers, Z and V registers for index (address index) are provided. See the following on combined use of Z and V registers and other devices.

If V0=3, Z0=5, D100V0=D103 C20Z0=C25 ← device number + values of V[ ] or Z[ ].

The data register and index register can be used for indirect specifying and applied instruction of the timer and counter.

Constant (K)，(H)

Of values used by PLC, K represents values of decimal system, H represents values of hexadecimal system and they are used as setting values of timer and counter or operand of applied instructions.

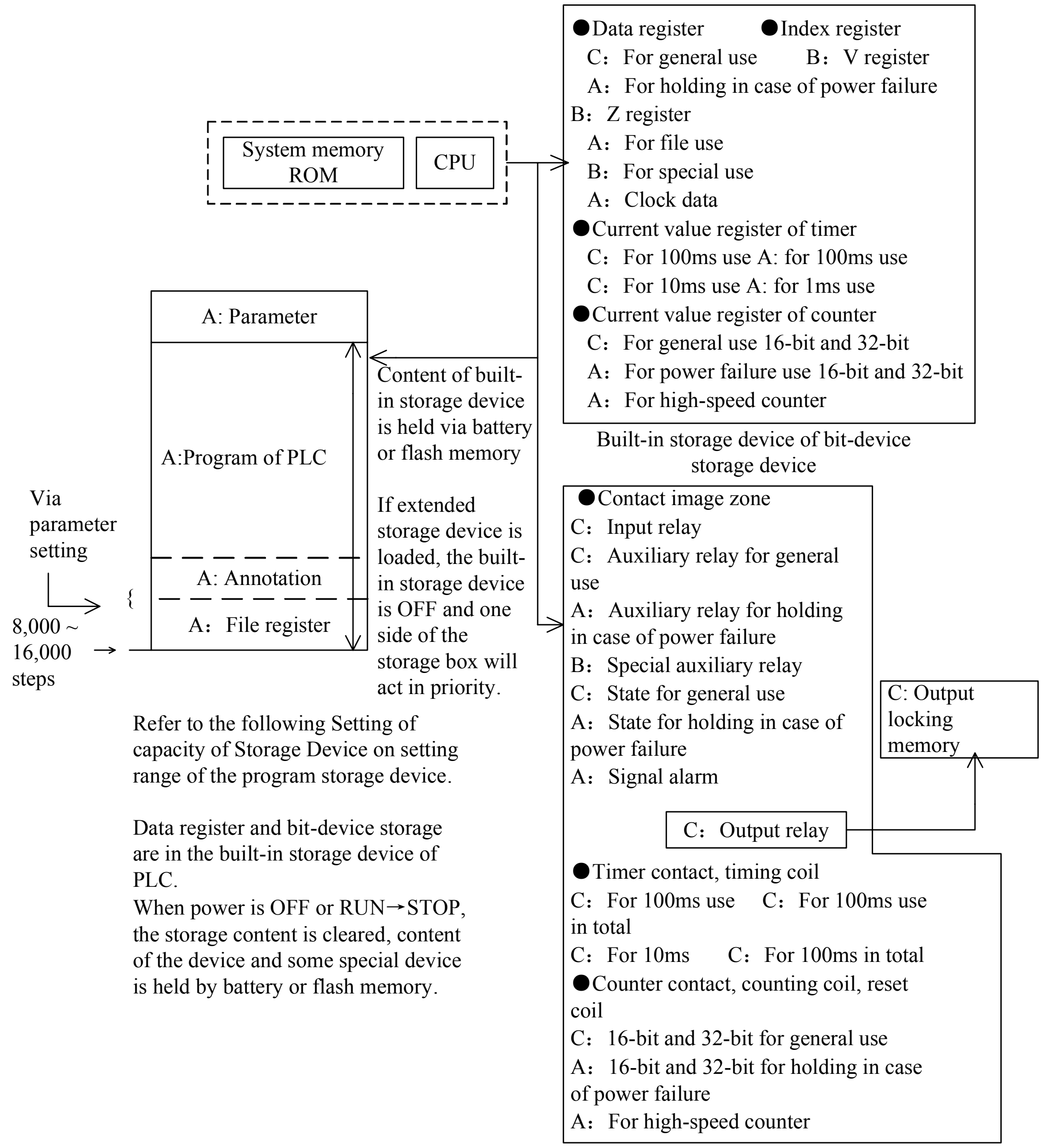
Pointer (P)，(I)

The pointer is used for branching and interruption. The pointer P for branching is used to specify F00(CJ) conditional jump or F01(CALL) sub-program jump. The pointer I for interruption is used to specify input interruption, timing interruption and counter interruption.

3 Program memory and parameter structure

Structure of storage device

See the figure on structure of storage device of PLC, in addition, devices of the storage device are divided into A, B and C according to content of initialization.



Type of storage devices	Power OFF	Power OFF→ON	STOP→ RUN	RUN→STOP
A: Battery, Flash memory backup supporting series storage device	No change			
B: Special M and D, index register	Clear	Setting of initial value ☆	No change☆	
C: Other non-backup supporting series storage device	Clear		No change	Clear
			No change of M8033 drive	

☆ The represented part will be cleared from STOP→ RUN, please pay attention.

Parameter structure

The parameters are used to specify range of holding in case of power failure and capacities of annotation and file registers, and setting and change of parameters can be executed via PC/PDA LINK. Regarding operation and its details, please refer to Help text of PC/PDA LINK. Refer to TP03 operating manual of functions of the parameters.

Parameter type and setting content

① Settings of storage device capacity: D8006.

② Settings of locking range: it is used to change range of holding in case of power failure of PLC.

③ Password registration Password can be set, which is used to error writing of programmed sequential control program or embezzlement, however, for online operation of programming software, password can be used to set 3 protection levels.

④ Other parameters: it is used to set validity/invalidity of RUN/STOP, specify non-battery operation mode and set PC general communication.

Initial values of parameter settings

TP03 M/H Type

Item		Initial value	PC/PDA device
Capacity of storage device	Program capacity	8K (M/H 20&30 points) 16K (40&60 points)	⊙
	Annotation capacity	0	⊙
Locking range (Holding range in case of power failure)	Auxiliary relay (M)	500-1023 (0-1023)	⊙
	State (S)	500-999 (0-1023)	⊙
	Counter (C) (16)	100-199 (0-199)	⊙
	Counter (C) (32)	200-255 (200-255)	⊙
	Data register (D)	200-511 (0-511)	⊙
Password		None	⊙
Input setting of terminal RUN		None	⊙
Input number of terminal RUN		None	⊙
PC general communication settings		None	⊙

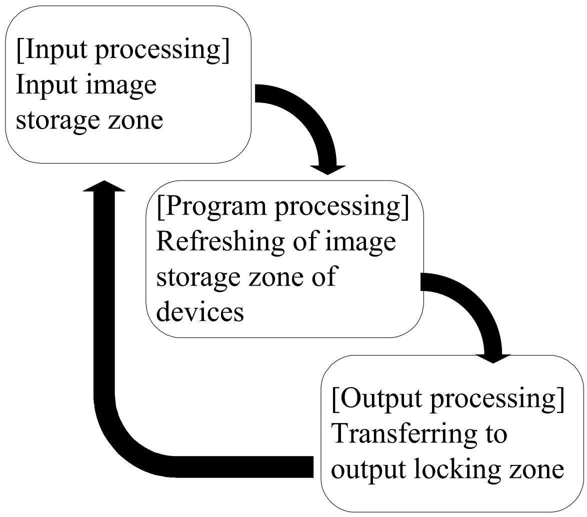
TP03 SR Type

Item		Initial value	PC/PDA device
Capacity of storage device	Program capacity	4K (S 14&20&26&36 points)	⊙
	Annotation capacity	0	⊙
Locking range (Holding range in case of power failure)	Auxiliary relay (M)	500-1023 (0-1023)	⊙
	State (S)	500-999 (0-1023)	⊙
	Counter (C) (16)	90-99 (0-99)	⊙
	Counter (C) (32)	220-255 (220-255)	⊙
	Data register (D)	400-511 (400-511)	⊙
Password		None	⊙
Input setting of terminal RUN		None	⊙
Input number of terminal RUN		None	⊙
PC general communication settings		None	⊙

⊙ : Change available

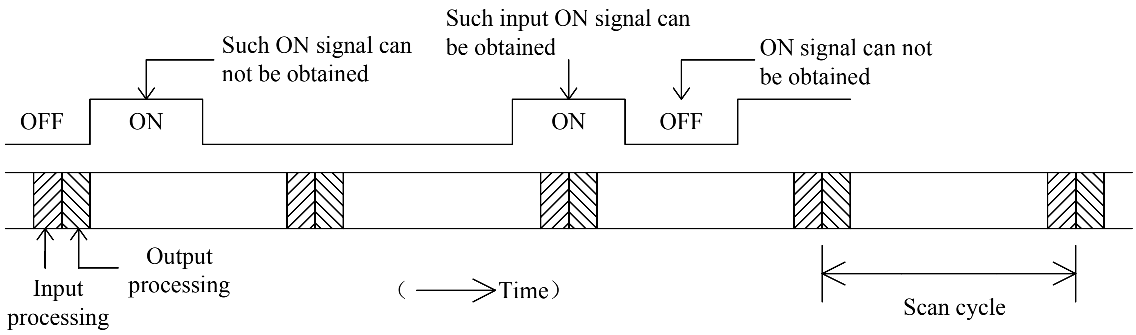
4 Notes(Input and output processing, response lagging, dual-coil)

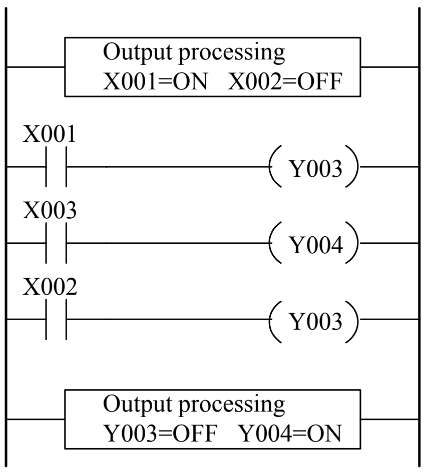
Action time sequence and response lagging of input and output relays



Restrictions on signal width of input pulse

Time width of input ON/OFF of PLC is longer than cycle time of PLC, if response lagging of input wave filter is 2ms, the cycle time is 10ms, time of ON/OFF needs 12ms respectively. Therefore, the input pulse of 1,000/(12+12)=40Hz and above can not be processed, however, if special function and applied instruction of PLC are used, such defect will be improved.





See the left chart, the same coil Y003 can be used at several points.

For example, take X001=ON, X002=OFF

For initial Y003, for X001 is ON, the image storage zone is ON, the output Y004 is ON.

However, for the secondary Y003, the input X002 is OFF. Therefore, the image storage zone is rewritten OFF. Therefore, actual external output Y003=OFF, Y004=ON.

Chapter II Device

1 Processing of digits, constants K and H

Processing of digits

Decimal system digit DEC	● Setting value K of timer and counter. ● Numbers of auxiliary relay M, timer T, counter C, status S and etc. (number of device). ● Specify digit and command action K in application command operation.
Hexadecimal system digit HEX	For decimal system digit, specify digital command action H in application command operation.
Binary digit BIN	● Specify digits of counter or data register with timer of decimal or hexadecimal system, however, such digit is processed with binary system digits within PLC. When monitor is performed on PC/PDA link, such device can be converted into decimal or hexadecimal system digit.
Octonary digit OCT	● Device numbers of the input/output relay are executed according to octonary digit system, therefore, it can be numbered 0~7, 10~17…70~77 and etc., there is no 8 and 9 in octonary system.
BCD code	● BCD code is used to express 0-9 of decimal system with 4-digit binary system. It is easy to process, therefore, it is used to digital switch of BCD output form or monitor control of 7 codes and etc.
Constants K, H	● K is the symbol of expressing integral of decimal system. ● H is the symbol of expressing integral of hexadecimal system. When operation for command digit and etc. is done on PC/PDA for programming, input with K for decimal digit and H for hexadecimal digit, such as K10, H102.

Conversion of digits

	Octonary digit OCT	Decimal system digit DEC	Hexadecimal system digit HEX	Binary digit BIN		BCD	
	0	0	00	0000	0000	0000	0000
	1	1	01	0000	0001	0000	0001
	2	2	02	0000	0010	0000	0010
	3	3	03	0000	0011	0000	0011
	4	4	04	0000	0100	0000	0100
	5	5	05	0000	0101	0000	0101
	6	6	06	0000	0110	0000	0110
	7	7	07	0000	0111	0000	0111
	10	8	08	0000	1000	0000	1000
	11	9	09	0000	1001	0000	1001
	12	10	0A	0000	1010	0001	0000
	13	11	0B	0000	1011	0001	0001
	14	12	0C	0000	1100	0001	0010
	15	13	0D	0000	1101	0001	0011
	:	:	:	:	:	:	:
	143	99	63	0110	0011	1001	1001
Main Application	Device numbers of input and output relays	Internal device numbers except constant, input and output relays	Constant H	Internal processing of PLC		BCD digit switch, 7 codes monitor	

2 List of Device Numbers

PLC of TP03 series has 4 basic programming elements. To identify varieties of programming elements, different symbols are specified. It is stated below:
X: Input relay, for storing on and off the external input circuit.
Y: Output relay, used to output physical signal from PLC directly.
M: Auxiliary relay and S: status relay: internal computation symbol of PLC.

List of device device:
TP03M/H machine types:

Input/ Output Type	TP03-20	TP03-30	TP03-40	TP03-60	Increasing expansion
Relay X X000~X377 255 points	X000~X013 12 points	X000~X017 16 points	X000~X027 24 points	X000~X043 36 points	X000~X177 128 points for M type X000~X377 256 points for H type
Relay Y Y000~Y377 256 points	Y000~Y007 8 points	Y000~Y005 14 points	Y000~Y017 16 points	Y000~Y027 24 points	Y000~Y177 128 points for M type Y000~Y377 256 points for H type

TP03SR machine type

Input/ Output Type	TP03SR-14	TP03SR-20	TP03SR-26	TP03SR-36	Increasing expansion
Relay X X000~X377 255 points	X000~X007 8 points	X000~X013 12 points	X000~X017 16 points	X000~X023 20 points	X000~X117 80 points
Relay Y Y000~Y377 256 points	Y000~Y005 6 points	Y000~Y007 8 points	Y000~Y011 10 points	Y000~Y017 16 points	Y000~Y117 80 points

There are 512 points for X and Y of relay, actual I/O module can be used for output and input, and the left points are used as auxiliary relay. Numbers of X and Y is octonary digital system, such as X000~X007, X010 after X007 not X008.

TP03M/H machine type

<table>
<tr><td>Auxiliary relay M</td><td colspan="2">M0~M499
500 points ※1 for general use</td><td>M500~M1023
524 points
※2 for keeping</td><td colspan="2">M1024~M7679
6656 points
※3 for keeping</td><td>M8000~M8511
512 points
※4 for special use</td></tr>
<tr><td>Status relay S</td><td colspan="2">S0~S499
500 points ※1
for general use
For initialization S0~S9
/Origin return S10~S19</td><td colspan="3">S500 ~ S1023
524 points ※2
for keeping
S900 ~ S999
100 points For alarm</td><td colspan="2">S1024 ~ S4095
3072 points
※3 for keeping</td></tr>
<tr><td>Timer T</td><td colspan="2">T0~T199 200 point
100ms
(T192~T199 for sub-program) ※5</td><td>T200~T245
46 points
10ms
※5</td><td>T246~T249 4 points
1ms in total
(T246~T249 for sub-program) ※3</td><td>T250~T255
6 points,
100ms in total
※3</td><td>T256~T511
256 points,
1ms ※5</td></tr>
<tr><td rowspan="2">Counter C</td><td colspan="2">16-bit plus counting</td><td>32-bit plus and minus counting</td><td colspan="3">32-bit high speed plus and minus</td></tr>
<tr><td>C0~C99
100 points
※1 for
general use</td><td>C100~C199
100 points
※2 for
keeping</td><td>C200~C234 35 points
※2 for keeping</td><td>C235~C245
1 phase 1
input
※2</td><td>C246~C249
1 phase 2
input
※2</td><td>C251~C254
2 phase
input
※2</td></tr>
<tr><td>Data register D</td><td>D0~D199
200 points
※1 for
general use</td><td>D200~D511
312 points
※2 for
keeping</td><td>D512~D7999 7488 points
※3 for file
(D2000~D3299 can be set as file register)</td><td colspan="2">D8000~D8511
512 points
※4 for special use</td><td>W0~W9999
10000 points
Supplementary register(Available since H/M V2.3)</td></tr>
<tr><td>Data register V,</td><td colspan="6">V0~V15, Z0~Z15
32 points (For indirect specifying)</td></tr>
<tr><td>Nested finger N,P,k</td><td>N0~N7
8 points
For main control circuit</td><td>P0~P255 256 points
Finger of jumping and sub-program</td><td>I00X~I50X
6 points
Pointer for interrupt input</td><td colspan="2">I6XX~I8XX
3 points
Pointer for time interrupt</td><td>I010~I060
6 points
Pointer for counting interrupt</td></tr>
<tr><td rowspan="2">Constant</td><td>K</td><td colspan="2">16-bit -32,768~32,767</td><td colspan="3">32-bit -2,147,483,648~2,147,483,647</td></tr>
<tr><td>H</td><td colspan="2">16-bit 0~FFFFH</td><td colspan="3">32-bit 0~FFFFFFFFH</td></tr>
</table>

※1 Non-keeping field for power failure Parameters are used to set and change the keeping field for the keeping field for power failure.

※2 Keeping field for power failure Parameters can be used to set and non-keeping field for power failure.

※3 Fixing range for keeping in case of power failure, and the keeping range can not be changed.

※4 Refer to list of special elements.

※5 Non-keeping field for power failure Parameters are not used to set and change the keeping field for the keeping field for power failure.

TP03SR machine type

<table>
<tr><td>Auxiliary relay
M</td><td colspan="2">M0~M499
500 points
※1 for general use</td><td>M500~M1023
524 points
※2 for keeping</td><td colspan="2">M1024~M1535
512 points
※3 for keeping</td><td>M8000~M8511
512 points
※4 for special use</td></tr>
<tr><td rowspan="2">Status relay
S</td><td colspan="2">S0~S499 500 points
※1 for general use</td><td colspan="4">S500 ~ S1023
524 points ※2 for keeping</td></tr>
<tr><td colspan="2">Initialization S0~S9
/Origin return S10~S19</td><td colspan="4">S900 ~ S999
100 points for alarm</td></tr>
<tr><td>Timer
T</td><td colspan="2">T0~T39, T196~T199
44 points 100ms
(Sub-program:
T196~T199) ※5</td><td>T200~T245
46 points
10ms
※5</td><td colspan="2">T246~T249 4 points
1ms in total
(Sub-program:
T246~T249) ※3</td><td>T250~T255
6 points
100ms in total ※3</td></tr>
<tr><td rowspan="2">Counter
C</td><td colspan="2">16-bit plus counting</td><td>32-bit plus and minus counting</td><td colspan="3">32-bit high speed plus and minus</td></tr>
<tr><td>C0~C89
90 points
※1 for general use</td><td>C90~C99
10 points
※2 for keeping</td><td>C220~C234
15 points
※2 for keeping</td><td>C235~C238
C241~C242
1 phase 1 input
※2</td><td>C246~C247
1 phase 2 input
※2</td><td>C251~C253
2 phase Input
※2</td></tr>
<tr><td>Data register
D</td><td>D0~D399
400 points
※1 for general use</td><td>D400~D511
112 points
※2 for keeping</td><td colspan="4">D8000~D8511
512 points
※4 for special use</td></tr>
<tr><td>Data register
V、Z</td><td colspan="6">V0~V15, Z0~Z15
32 points (For indirect specifying)</td></tr>
<tr><td>Nested finger
N、P、I</td><td>N0~N7
8 points
For main control circuit</td><td>P0~P127 128 points
Finger of jumping and sub-program</td><td>I00X~I30X
4 points
Pointer for interrupted input</td><td colspan="2">I6XX~I8XX
3 points
Pointer for time interrupt</td><td>I010~I060
6 points
Pointer for counting interrupt</td></tr>
<tr><td rowspan="2">Constant</td><td>K</td><td colspan="2">16-bit -32,768~32,767</td><td colspan="3">32-bit -2,147,483,648~2,147,483,647</td></tr>
<tr><td>H</td><td colspan="2">16-bit 0~FFFFH</td><td colspan="3">32-bit 0~FFFFFFFFH</td></tr>
</table>

※1 Non-keeping field for power failure Parameters are used to set and change the keeping field for the keeping field for power failure.

※2 Keeping field for power failure Parameters can be used to set and non-keeping field for power failure.

※3 Fixing range for keeping in case of power failure, and the keeping range can not be changed.

※4 Refer to list of special elements.

※5 Non-keeping field for power failure Parameters are not used to set and change the keeping field for the keeping field for power failure.

3 Number and function of input and output relays (X/Y)

Input and output relays

Number of input and output relays

Numbers of input and output relays are made up of inherent address No. of basic unit and address No. of expansion equipment of the above-mentioned number and these address numbers are expressed in octonary system. For example, in octonary system, 17 and 20 are adjacent integrals.

TP03M/H machine type

Input/ Output Type	TP03-20	TP03-30	TP03-40	TP03-60	Increasing expansion
Relay X X000~X377 255 points	X000~X013 12 points	X000~X017 16 points	X000~X027 24 points	X000~X043 36 points	X000~X177 128 points for M type X000~X377 256 points for H type
Relay Y Y000~Y377 256 points	Y000~Y007 8 points	Y000~Y005 14 points	Y000~Y017 16 points	Y000~Y027 24 points	Y000~Y177 128 points for M type Y000~Y377 256 points for H type

TP03SR machine type

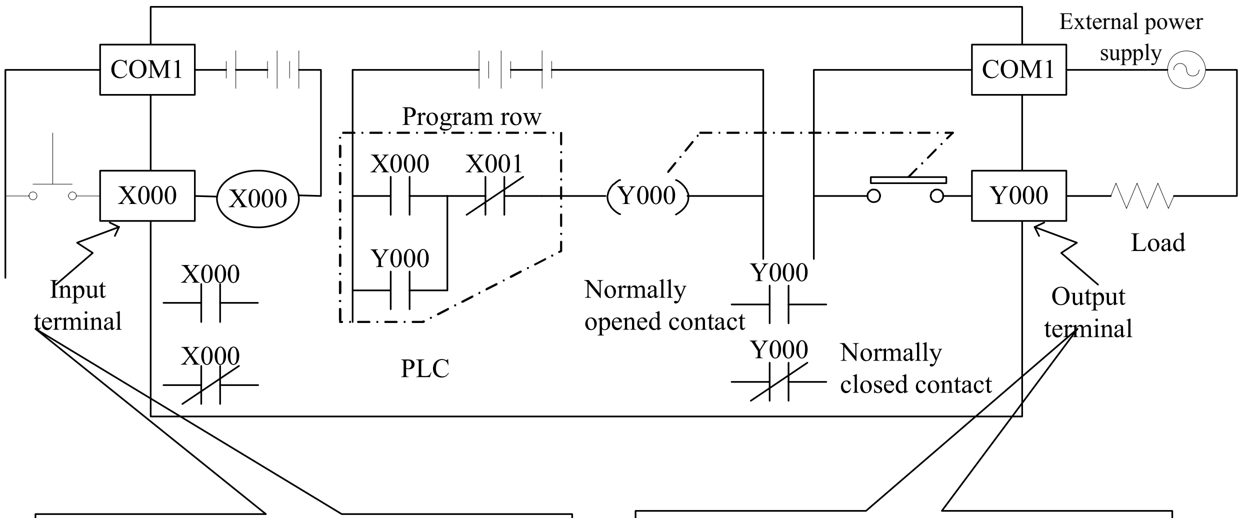
Input/ Output Type	TP03SR-14	TP03SR-20	TP03SR-26	TP03SR-36	Increasing expansion
Relay X X000~X377 255 points	X000~X007 8 points	X000~X013 12 points	X000~X017 16 points	X000~X023 20 points	X000~X117 80 points
Relay Y Y000~Y377 256 points	Y000~Y005 6 points	Y000~Y007 8 points	Y000~Y011 10 points	Y000~Y017 16 points	Y000~Y117 80 points

Input relay is the window for PLC receiving external switching quantity. PLC reads and stores external signal conditions into image memory. The input terminal is connected to external normally opened contact or closed contact and series connection or parallel connection circuits or electronic sensor made of several contacts or electronic sensor (such as, proximity switch). In Ladder Logic, normally closed contact and closed contact of input relay can be used for many times.

PLC of output relay is the window for PLC sending loading signal, the output relay is used to transmit output signal of PLC to the output module and the later one drives external loading.

Function

The following is the sketch map of PLC control system. When external input circuit of X000 terminal is get connected, the corresponding input image memory is “1” and “0” when it is disconnected. Status of input relay only depends on status of external input signal, which is not controlled by users’ program. Therefore, there will not be coil of input relay in Ladder Logic When Y000 coil is connected, normally opened contact of corresponding hardware relay of output module of relay is closed and the external loading works. Each relay in the output relay has only one contact, however, in Ladder Logic, normally opened contact and closed contact of each output relay can be used for many times.

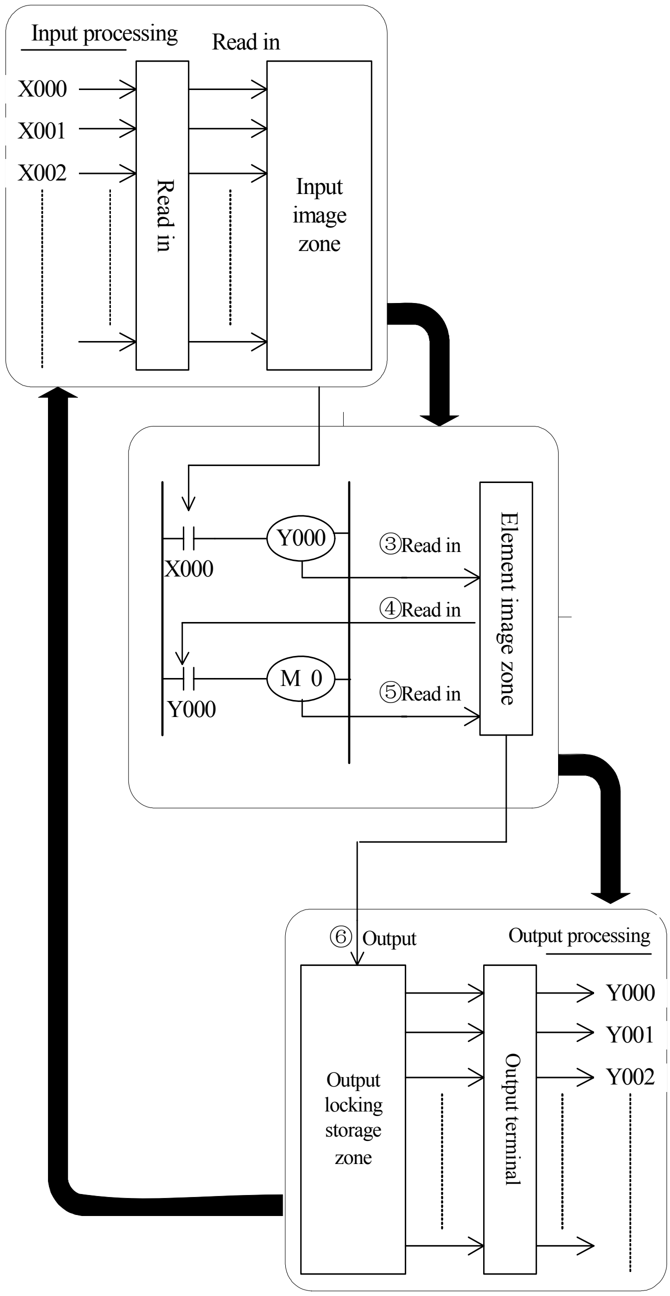


Input terminal is the terminal for receiving external input switch signal. PLC input terminal is connected to input component X which is an electronic relay with optical coupling features, and it owns several normally opened contacts (a point) and normally closed contact (b point). Theses points are used in PLC and input relay can not be driven in program.

Output terminal is the terminal connected to the outside. The output relay is the point for external output (transistor, double-direction flow, the relay is used for output terminal in PLC). The output relay on PLC has many normally opened/closed contacts and can be used. External output point and internal point can be seen below.

Action time sequence of input relay

PLC can perform sequence control by executing the following program repeatedly. When massive input and output is used, drive time and computation cycle of output wave filter and output assembly will be correspondingly delayed.



●Input processing

Before PLC executing program, all ON/OFF status of PLC will be read into the image zone.

During program executing, in case of input change, content of input image zone will not be changed, and when it is processed in the next cycle, such change will be read.

In addition, even ON→OFF and OFF→ON happen, before judging ON/OFF, there will be about 10ms delay for the input wave filter.

●Program processing

PLC reads out ON/OFF from input image zone or image zone of other image area according to command. It computes from step 0, then write the results into the image zone, therefore, image zone of the device element shall execute according to content of the image memory with the internal contacts.

●Output processing

Once all the commands have been executed, ON/OFF of image memory of output Y is transmitted to the locking zone, and it becomes actual output of PLC.

For contacts for external output of PLC, the response will be delayed according to the device element for output.

4 Number and function M of auxiliary relay M

Auxiliary relay

The auxiliary relay (M) is realized by device It can not accept external input signal and also can not drive external loading, it is an internal status sign

Number of auxiliary relay M is stated blow: the number is distributed according to decimal system.

TP03M/H machine type

For general use	Use for keeping in case of power failure	Special use for keeping in case of power failure	For special use
M0~M499 500 points ※1	M500~M1023 524 points ※2	M1024~M7679 6656 points ※3	M8000~M8511 512 points

TP03SR machine type

For general use	Use for keeping in case of power failure	Special use for keeping in case of power failure	For special use
M0~M499 500 points ※1	M500~M1023 524 points ※2	M1024~M1535 512 points ※3	M8000~M8511 512 points

※1 Non-keeping field for power failure Parameters are used to set and change the keeping field for the keeping field for power failure.

※2 Keeping field for power failure Parameters can be used to set and non-keeping field for power failure.

※3 Fixing range for keeping in case of power failure, and the keeping range can not be changed.

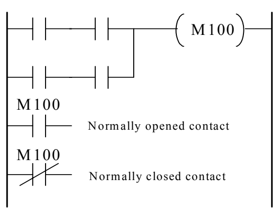
Case of function and act

PLC has many auxiliary relays. Like output relay, coil of such auxiliary relay is driven by contact of device elements in the PLC.

The auxiliary relay has many electronic normally opened contacts and closed contacts, and they can be used in PLC, however, such contact can not drive external loading, and drive of external loading shall be executed through the output relay.

For general use

When power down, the auxiliary relay, the output relay will be OFF. If it is powered a second time, besides the external input signal is ON, the others are still OFF.



Distribution of auxiliary relays for general use and keeping for power down in TP03 can be set and adjusted through PC/PDA link.

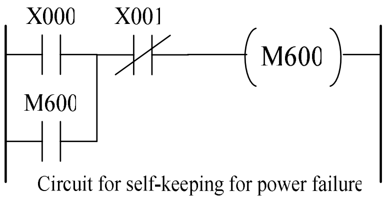
For keeping for power failure

Some control system memorizes status before power failure and such status will re-appear for a second operation.

Auxiliary relay for keeping for power supply is also called relay for keeping. It makes use of backup battery or flash memory in the PLC for keeping for power failure. It keeps the relay instant status in the first scanning cycle after PLC is powered on.

If the special relay for keeping for power supply is taken as general auxiliary relay, RST or ZRST can be used to clear at the front-most of the program.

In addition, when inter-PC link or parallel connection link is used, some auxiliary relay is occupied as link.



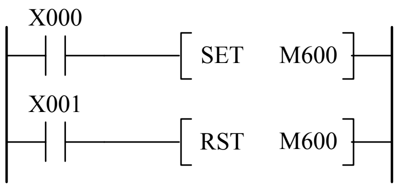
Circuit for self-keeping for power failure

The left figure displays the demonstration for keeping for power failure of M600. In the circuit, if X000 is connected, M600 acts, even X600 keeps acting, therefore, even X000 is open circuit caused by power failure, M600 will continue to act for a second operation.

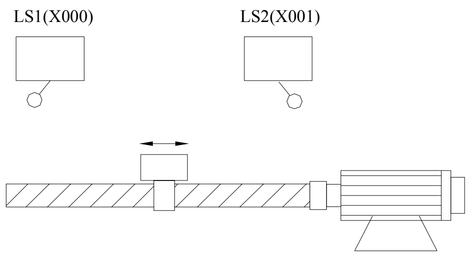
However, if normally closed contact of X001 is open circuit for a second operation, M600 will not act.

See the left figure for the commands SET, RST

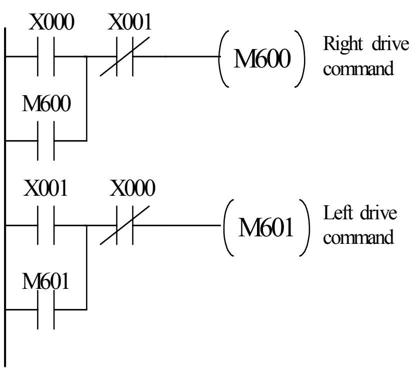
Example of keeping for power failure



Circuit for keeping for setting and clearing for power failure



When it is operated for a second time, direction of advance is the same as the direction before power failure.



X000=ON (Left limit switch) →M600=ON→ Right drive → Power off→ Platform stops → Operate a second time (M600=ON) →X001=ON (Right limit switch) →M600=OFF, M601=ON→ Left drive.

Special use

There are 512 special auxiliary relay in the PLC. These relays have its specified functions, which are divided into two types:

a. (Special auxiliary relay with contact functioning): drive coil of PLC is used, and the user can use such contact.

M8000: Operation monitor

M8002: Initial pulse

M8012: 100ms cycle oscillation

The user can not use undefined special auxiliary relay.

b. (Special auxiliary relay with coil drive): the users can drive these coils for specified operation.

M8033: Keep memory as required

M8034: All outputs forbidden

M8039: Constant scanning

Please note that there are two validities when driving and after executing END.

5 Number and function of status relay S

Status

The status relay is a kind of programming element for programming sequence control and it is used with commands of STL and RET described in chapter 4. The status relay for general use does not have the function of keeping in case of power failure. The relay for keeping in case of power failure can utilize the built-in backup battery or Flash memory of PLC for storing ON/OFF.

The status number S is stated blow (distributed according to decimal system).

TP03M/H machine type

Status relay S	S0~S499 500 points ※1 for general use	S500 ~ S1023 524 points ※2 for keeping	S1024 ~ S4095 3072 points ※3 for keeping
	S0~S9 for initialization / S10~S19 for origin return	S900 ~ S999 100 points for alarm	

TP03SR machine type

Status relay S	S0~S499 500 points ※1 for general use	S500 ~ S1023 524 points ※2 for keeping
	S0~S9 for initialization / S10~S19 for origin return	S900 ~ S999 100 points for alarm

※1 Non-keeping field for power failure Parameters are used to set and change the keeping field for the keeping field for power failure.

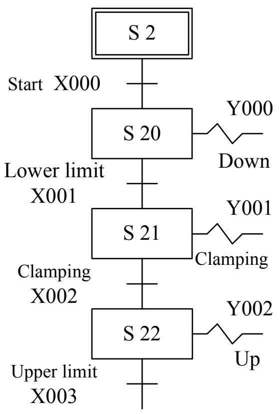
※2 Keeping field for power failure Parameters can be used to set and non-keeping field for power failure.

※3 Relevant features for keeping for power failure, which can not change with parameters.

Example of function and action

Status S is an important device for simple programming of step control of working procedure, which is used with step echelon command STL.

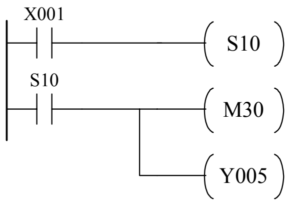
For general use/keeping



As the step control of working procedure described in the figure, if the starting signal X000 is on, the status S20 is on, the electromagnetic valve Y000 for down acts, the result is that: if the lower limit switch X001 is ON, the status S21 is on and electromagnetic valve Y001 for clamping acts.

If the limit switch X002 confirmed by clamping is ON, the status S22 is ON. With action moving, the status will return to original status.

After status relay for general use is powered off, it is OFF. The status relay for keeping in case of power failure can store the status before power failure. Therefore, it can be operated from the middle working procedure.

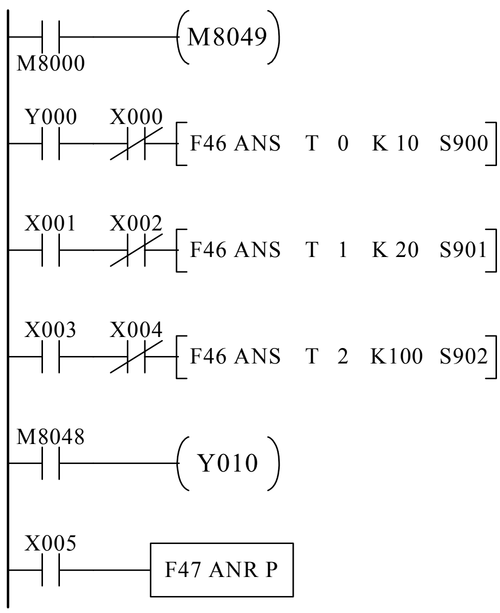


Like status relay, auxiliary relay has many normally opened and closed contacts, it can be used within the sequence control procedure.
In addition, if it is not used for step echelon command, status relay S and auxiliary relay M can be used in general sequence control.

PLC can change distribution for general use and power off use via setting of PC/PDA link parameters.

For signal alarm

The status for signal alarm can also be used as output of external failure diagnosis.
For instance, compile external failure diagnosis circuit in the figure below, monitor the special data register D8049 and display minimum number of S900~S999.
In case of several failures, clear the failure with minimum number to get the number of the next failure.



If special auxiliary relay M8049 is driven, the monitor is under effective condition.
After drive advance outputs Y00, if the advance terminal detects that X000 does not work within 1 second, S900 acts.
If the upper limit X001 and lower limit X002 do not work over 2 seconds, S901 acts.
For continuous operation mode input X003 is ON for machine with interval less than 10 seconds, if the action switch X004 does not work in one cycle, S902 acts.
If any of S900~S999 is ON, the special auxiliary relay M8048 acts, the failure display output Y010 will act. External failure diagnosis program is changed into OFF by the reset button X005. For each X005 ON, action of minimum number resets one by one.

6 Number and function of timer T

Number of timer

TP03M/H machine type

Timer T	T0~T199 200 points 100ms (For sub-program T192~T199)	T200~T245 46 points 10ms	T246~T249 4 points,1ms in total (For sub-program T246~T249)	T250~T255 6 points,100ms in total	T256~T511 256 points, 1ms

TP03SR machine type

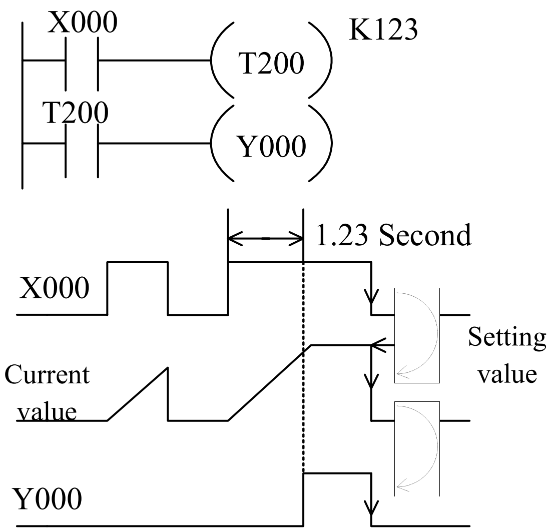
Timer T	T0~T39，T196~T199 44 points 100ms (For sub-program: T196~T199)	T200~T245 46 points 10ms	T246~T249 4 points,1ms in total (For sub-program T246~T249)	T250~T255 6 points,100ms in total

If it is not used as counter number of the counter, it can be used as data register for storing data.

Function

The timer accumulation is used for clock pulse of 1ms, 10ms, 100ms and etc. of PLC. When reaching specified setting, contact of the output acts. The setting value employs the constant K as setting value and data register D can be used for indirect specifying.

For general use

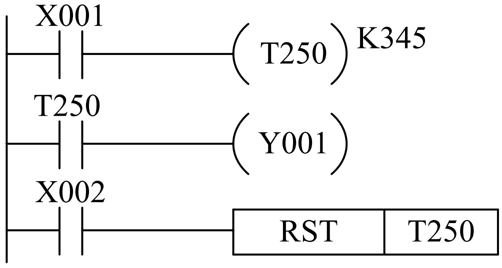


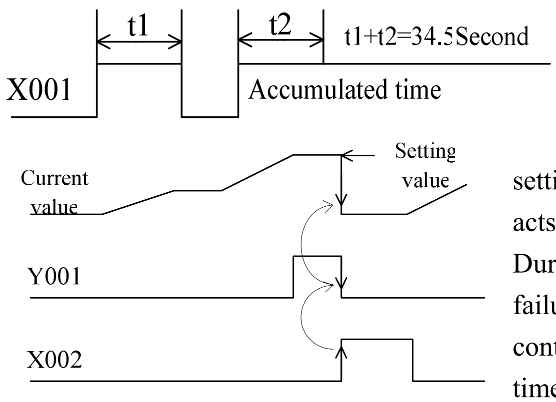
If drive input X000 of the timer coil T200 is ON, T200 employs the clock pulse totaled 10ms of the counter. If the value is equal to setting value K123, output contact of the timer acts.

The output contacts acts 1.23 seconds after coil drive.

Drive input X000 is off or powered off, the timer resets and the output contact resets.

For accumulated use





If drive input X001 of the timer coil T250 is ON, T250 employs the clock pulse totaled 100ms of the counter. If the value is equal to setting value K345, output contact of the timer acts.

During computation, even input X001 is off or failure of power supply, when it restarts, and it continues to compute and the accumulated action time is 34.5 seconds.

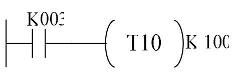
If reset input X002 is ON, the timer resets and the output contact also resets.

Special timers

T256~T511,these 256 points are special timers. When PLC is on ,they are used for general use. In case of RUN→STOP or power failure, they are used for accumulated use, the output contacts and data will keep.

Specifying method of setting value

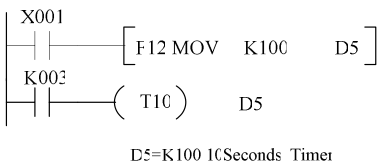
Specifying of constant



T10 is the counter taking 100ms (0.1S) as unit and 100 is specified as constant, so the timer 0.1S×100=10S works.

K is constant (integral of decimal system) 10 seconds timer

Indirect specifying D



Write content of indirect specified data register into program or input with digit switch. If it is specified as memory for keeping in case of power failure, please note that low voltage may lead to unstable setting value.

Processing of digital device

Current value of the counter can be used as value through application command and etc.

When it is used as data device, refer to number and function of internal counter.

Attentions in the procedure

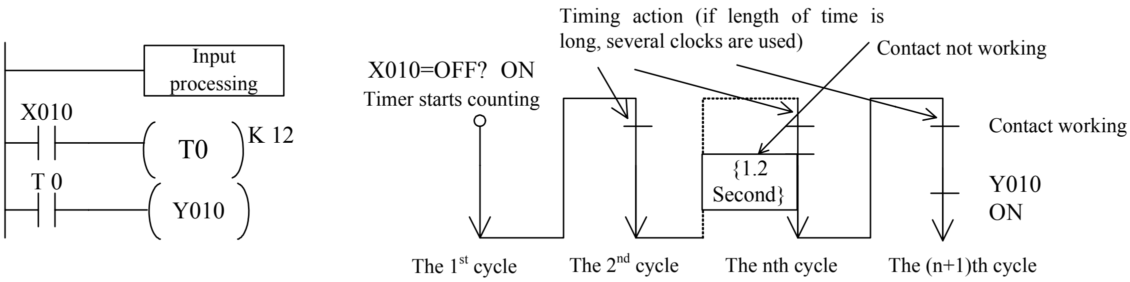
The timer of T192-T199 for sub-program and interrupted program and it starts timing when executing coil command or END command.

If timing reaches setting value, when executing coil command or END command, the output contact acts. The common timer executes coil command timing. Refer to act and precision of the timer in the following. Therefore, under some conditions, the coil command is used for executing sub-program or not timing for interruption and can not act.

If 1ms accumulated timer is used in the sub-program or interrupted program, when it reaches setting value, we must not that when executing initial coil command, the output contact acts.

Details and precisions of the timer

Besides the timer of interrupting executing, after the coil drive, the timer starts timing, after timing, the initial coil executes and the output contact acts.



Seen from the above figure, action precision of timer contact from driving coil to finishing, it can be expressed in the followings:

(T+To) ~ (T-α)

a: Correspondent with 1ms, 10ms and 100ms of the counter, namely 0.001, 0.01 and 0.1Second

T: Setting time of timer (S)

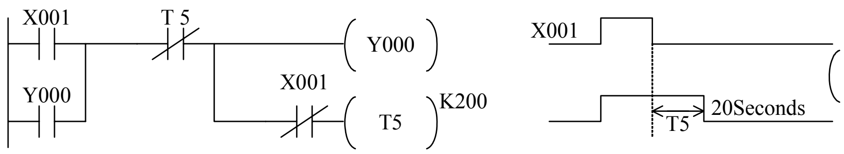
To: Scanning cycle (S)

When programming, the timer contact shall be written before the coil command, with maximum error +2T.

When setting value of the timer is 0 and the next coil command for scanning is executed, the output contact starts acting. In addition, after 1ms counter of interrupting execution executes coil command, 1ms clock pulse counting is executed in interruption mode.

Case of actions

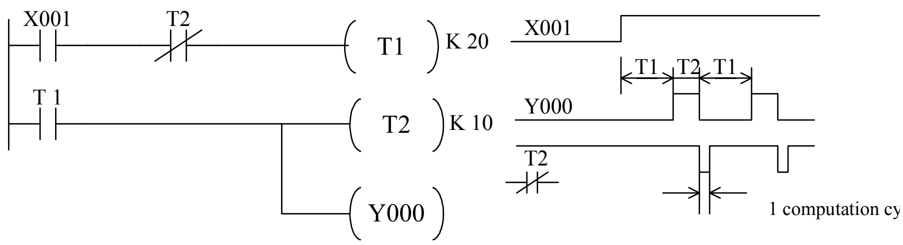
Output delay on and off timer



Sparkling point

When programming, the timer contact shall be written before the coil command, with maximum error +2T.

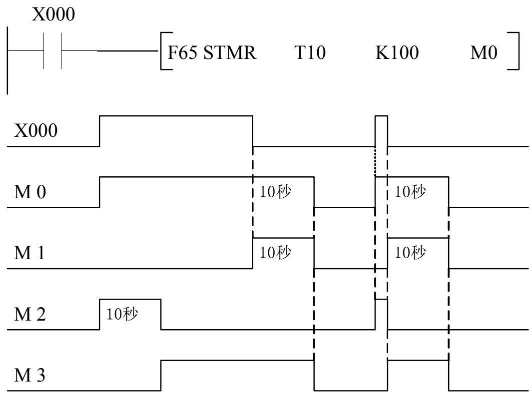
When setting value of the timer is 0 and the next coil command for scanning is executed, the output contact starts acting. In addition, after 1ms counter of interrupting execution executes coil command, 1ms clock pulse counting is executed in interruption mode.



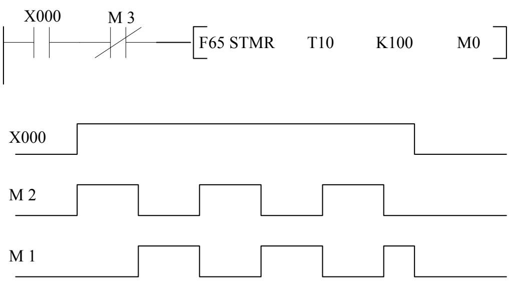
In addition, F66ALT command can be used for sparkling action.

Several counters by application command F65

Output on and off counter, single pulse output timer and sparkling timer are programmed.



- Specified value m is the setting value of the specified counter, 10 seconds in the case.
- M0 is OFF delay timer.
- M1 is ONE-SHOT timer after input point ON—OFF.
- M2 and M3 are sparkling actions. See the left figure.



- See the left figure on wiring of M3, and M1 and M2 have no sparkling.
- When X000 is changed into OFF, M0,M1 and M3 are changed into OFF, and T10 resets.
- Timer used here can not be used for other circuits.

In addition, if F64TTMR demonstration timer command is used, input time of the switch can be used to set time of the counter.

7 Number and function of counter C

Number of counter

Number of the counter is stated below and the number is distributed according to decimal system.

TP03M/H machine type

Counter C	16-bit in total		32-bit plus and minus	32-bit high speed plus and minus		
	C0~C99 100 points ※1 for general use	C100~C199 100 points ※2 for keeping	C200~C234 35 points ※2 for keeping	C235~C245 1 phase 1input ※2	C246~C249 1 phase 2 input ※2	C251~C254 2 phase input ※2

TP03SR machine type

Counter C	16-bit in total		32-bit plus and minus	32-bit high speed plus and minus		
	C0~C89 90 points ※1 for general use	C90~C99 10 points ※2 for keeping	C220~C234 15 points ※2 for keeping	C235~C238 C241~C242 1 phase 1input ※2	C246~C247 1phase2 input ※2	C251~C253 2phase input ※2

※1 Non-keeping field for power failure Parameters are used to set and change the keeping field for the keeping field for power failure.

※2 Keeping field for power failure Parameters can be used to set and non-keeping field for power failure.

Auxiliary relay numbers for 32-bit counter plus/minus switching

Counter No.	Direction switching	Counter No.	Direction switching	Counter No.	Direction switching	Counter No.	Direction switching
C200	M8200	C209	M8209	C218	M8218	C227	M8227
C201	M8201	C210	M8210	C219	M8219	C228	M8228
C202	M8202	C211	M8211	C220	M8220	C229	M8229
C203	M8203	C212	M8212	C221	M8221	C230	M8230
C204	M8204	C213	M8213	C222	M8222	C231	M8231
C205	M8205	C214	M8214	C223	M8223	C232	M8232
C206	M8206	C215	M8215	C224	M8224	C233	M8233
C207	M8207	C216	M8216	C225	M8225	C234	M8234
C208	M8208	C217	M8217	C226	M8226	——	——

Features of counter

Features of 16-bit counter and 32-bit counter are stated below. It can be used according to switching of counting direction and range of counting.

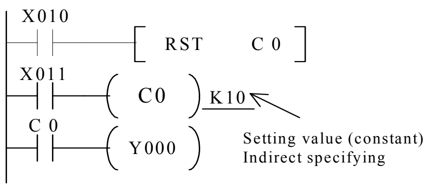
Item	16-bit counter	32-bit counter
Counting direction	Positive counting	Positive counting/negative counting switching (See the above sheet)
Setting value	0~32,767	-2,147,483,648~+2,147,483,647
Specified setting value	Constant K or data register	The same as left, one couple of memory after data completion (two)
Change of current value	Without change after positive counting	Change after positive counting (Circulating counter)
Output point	Keep acting after positive counting	Keep acting for positive counting and negative counting for reset
Reset actions	When executing RST, current value of the counter is zero, and the output point resets	
Current register	16 bits	32 bits

Example of function and action

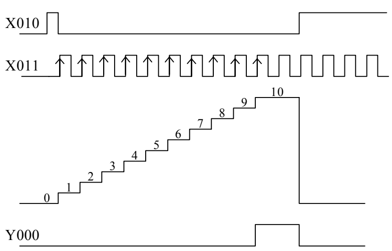
Distribution of status for counters for general use and keeping for power failure can be set and changed on PC/PDA link.

16-bit counter-for general use/keeping for power failure

16-bit binary plus counter, the effective setting value is K1~K32,767 (decimal constant). The setting values K0 and K1 have the same significances, namely, the output contacts acts at the first counting.



If power supply for PLC is cut off, counting value for general counter will be cleared, and the counter for keeping for power failure can store the counting value before power failure, therefore, the counter can continue to count from the last value before power failure.



The counting input X011 drives C0 coil one time, current value of the counter increases. When it executes the coil command the tenth time, the output contact acts. After that, if the counting input X011 acts a second time, current value of the counter will not change.

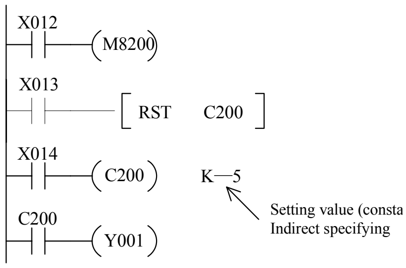
If reset input X010 is ON, RST command is executed. Current value of the counter is 0 and the output contact resets.

Setting value of the counter, besides specified by constant K, it can be specified by number of data register. For instance, D10 is specified, and D10 is 123, so it is the same as setting K123.

When the setting values are written into current data register with MOV and other commands, for the next input, the output coil is connected and the current memory is changed into setting value.

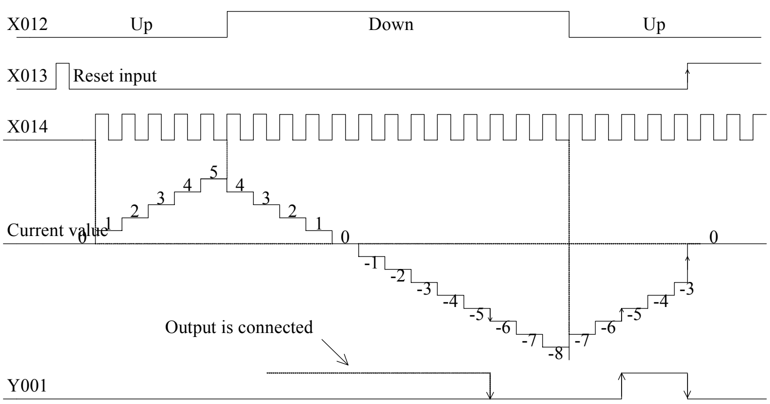
32-bit counter----for general use/keeping for power failure

Effective range of setting value of 32-bit binary plus/minus is -2,147,483,648~+2,147,483,647 (Decimal system constant). Special auxiliary relay M8200~M8234 is used to specify direction of minus/plus.



If C△△△ drives M8△△△, it is minus; otherwise, it is plus.

According to constant K and data register D, the setting value can be positive and negative, and content of data register with adjacent numbers are regarded as one couple and processed as 32-bit data. Therefore, when D0 is specified, D0 and D1 are processed as 32-bit setting value.



When counting input X014 is used to drive C200 coil, plus and minus are both available.

When current value of the counter is increased from -6-5, the output contact relocates, and when it is decreased from -5-6, the output contact resets.

For general use/keeping for power failure

Increasing/decreasing of the current value has nothing to do with action of output contact. If it is counted from 2,147,483,647, it will be changed to -2,147,483,648. Likewise, if it is minus from -2,147,483,648, it will be changed to 2,147,483,647. Such action is called ring counting.

If reset input X013 is ON, RST command is executed, current value of the counter is changed to 0 and the output contact resets.

When the counter for keeping for power failure is used, current value of the counter, action of the output contact and power failure for reset keep.

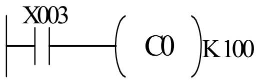
32-bit counter can be used as 32-bit data register. However, 32-bit counter can not be used as device element in 16-bit application command.

When the setting values are written into data register of current value with D-MOV command and etc., counting can be performed for the following counting input and the contact can not be changed.

Specifying of setting value

16-bit counter

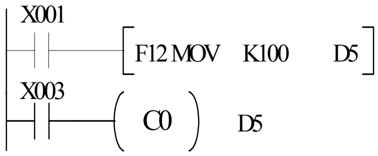
Specifying of constant K



Constant (Decimal system integral)

1-32,767 100 counts

Indirect specifying D



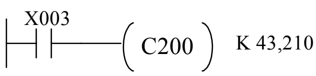
Write content of the indirectly specified data register into program in advance or input through digit switch.

When it is specified as the memory for keeping for power failure, please note that inadequate voltage may lead to unstable settings.

D5=K100 (100 counts)

32-bit counter

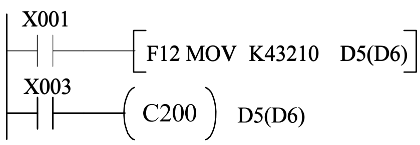
Specifying of constant K



Constant (decimal system integral)-2,147,483,648~ 2,147,483,647

43210 counting

Indirect specifying D



2 indirectly specified data memories are grouped to one group. While 32-bit command is written into setting value, do not repeat the data register on other program.

Response rate of the counter

When the counter executes circulating scanning and counting for X, Y, M, S, C and other contacts of PLC, for instance, X011 is taken as counting input, duration for getting through and disconnecting must be longer than scanning time of PLC (generally less than 10Hz). As for the mentioned high-speed counter executing counting with specified input for interrupted processing and counting for KHz, it has nothing to do with scanning time.

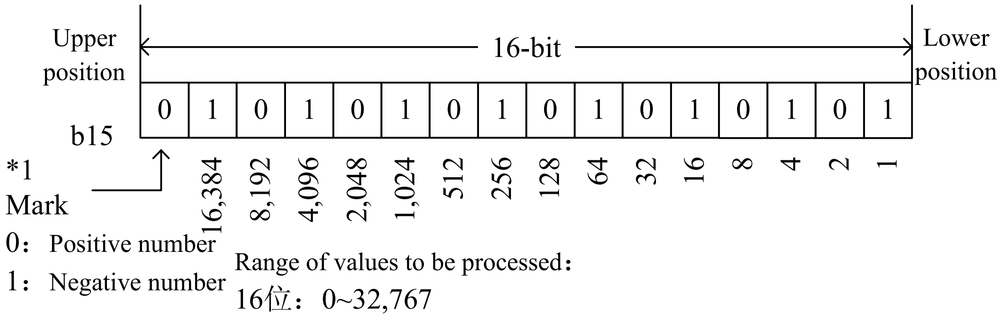
Processing of digital device element

The counter and timer act according to setting value. When the output contacts are used, counting value (current value) can be used as value for control.

Current value of the counter is the same as memory, processed as 16-bit or 32-bit data device elements.

16-bit（C）

Structure of current value memory and setting value memory of counter and timer (only limited to 16 bits)



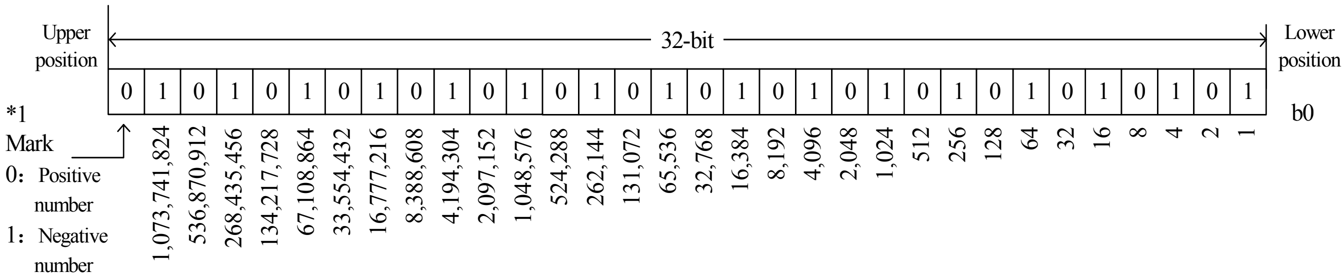
Range of values to be processed：

16位：0~32,767

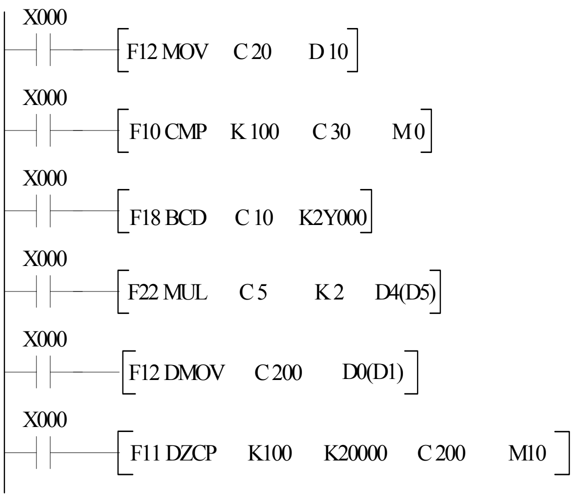
32位：-2,147,483,648~+2,147,483,647

*1: The above data valid only taken as data register.

32-bit（C）



Case of application command



C20(Current value)→D10 transmission

Compare decimal system integral 100 and C30 (current value), output the results to M0-M2.

Convert C10 (current value), output it to Y000-Y007. 7-code monitor.

Double C5 (current value), send it to D5 and D4.

C200(current value)→send it to D1, D0

Compare C200 (current value) and decimal system integral 100-20,000, output the results to M10-M11.

Case of application command describes how to use counter and timer as device element. Please refer to the following instructions.

8 Number and function of built-in counter C

Number of built-in high-speed counter

Built-in high-speed counter of PLC is expressed below:

It is distributed on input X000~X007 and X000~X007 according to number of the counter C, which can not be used repeatedly.

The input number which is not used as high-speed counter can be used as general input relay.

Besides, number of high-speed counter which is not used as high-speed counter can be used as 32-bit data register.

U: Plus input; D: Minus input A: A-phase input

B: B-phase input; R: Reset input; S: Starting input

TP03M/H machine type

	1 phase 1 counting input											1 phase 2counting input					2 phase 2 counting input				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249		C251	C252	C253	C254	
X000	U/D						U/D			U/D		U	U		U		A	A		A	
X001		U/D					R			R		D	D		D		B	B		B	
X002			U/D					U/D			U/D		R		R			R	A	R	
X003				U/D				R			R			U					B		
X004					U/D				U/D	S				D					R		
X005						U/D			R		S			R	S					S	

C250 / C255 Keep and unavailable

TP03SR machine type

	1 phase 1counting input											1 phase 2 counting input					2 phase 2counting input				
	C235	C236	C237	C238			C241	C242				C246	C247				C251	C252	C253		
X000	U/D						U/D					U	U				A	A			
X001		U/D					R					D	D				B	B			
X002			U/D					U/D					R					R	A		
X003				U/D				R											B		

C239~C240, C243~C245,C248~C250,C254~C255 Keep and unavailable

{Reading on the table}

Input X000, C235 single-phase and single input counting, without interruption reset and interruption starting input functions.

If C235 is used, C241, C244, C246, C247, C249, C251, C252, C254 and interruption pointer I00 are not used.

Refer to the operation manual 4 on the high-speed counter.

9 Number and function D of data register D

9.1 Data register D

Number of data register

Number of data register D is expressed as follows: (numbers are distributed according to decimal system):

TP03M/H machine type:

Data register D	D0~D199 200 points ※1 for general use	D200~D511 312 points ※2 for keeping	D512~D7999 7488 points For file ※3 (D2000~D3299 can be set as file register)	D8000~D8511 512 points ※4 for special use

TP03SR machine type

Data register D	D0~D399 400 points ※1 for general use	D400~D511 112 points ※2 for keeping	D8000~D8511 512 points ※4 for special use

※1 Non-keeping field for power failure Parameters are used to set and change the keeping field for the keeping field for power failure.

※2 Keeping field for power failure Parameters can be used to set and non-keeping field for power failure.

※3 Fixing range for keeping in case of power failure, and the keeping range can not be changed.

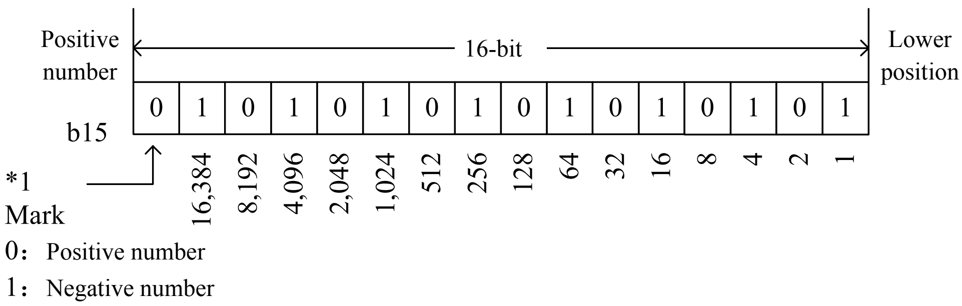
※4 Refer to list of special elements.

Structure and function of register

Data register is the device element for storing data and the type is expressed as follows. These register is 16-bit (positive and negative mark for the maximum digit bit). Combine the two data register to store 32-bit data (positive and negative mark for the maximum digit bit).

16-bit (D)

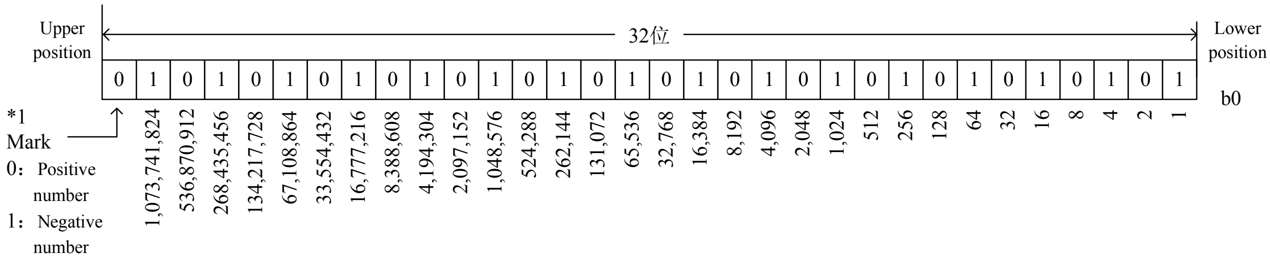
One data register (16-bit digit range) -32,768~+32,767



Readout and writing-in of data register employ application command. In addition, direct readout/writing-in can be executed from the unit (monitor) and programming equipment.

32-bit (D)

Two adjacent data registers is used to express 32-bit data (the high digit bit is a big number and the low digit bit is a small number, in Index register, V is a high digit and Z is low digit).
Therefore, we can process data of -2,147,483,648~+2,147,483,647.



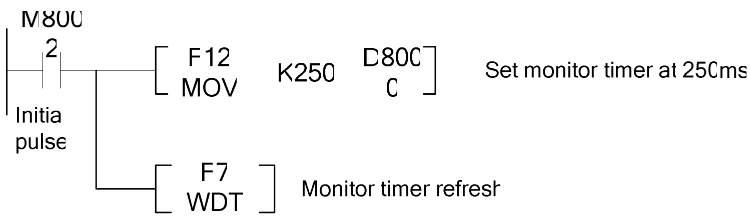
When 32-bit is specified, if the following digit bit (such as D0) is specified, the number after the high digit bit (such as D1) will be occupied automatically. The low digit bit can be specified any device element of odd or even. Considering monitoring of PC/PDA link, the following even device element number is recommended.

For general use/keeping for power supply

Once data is written into the data register, if other data will not be written, it will not change. However, in case of RUN→STOP or power failure, all the data will be cleared. If special auxiliary relay M8033 is driven, it can keep. Therefore, the data register for keeping for power failure can keep the content in case of RUN/STOP and power failure.
PC/PDA link parameter setting can be used to change distribution of PLC for general use and keeping for power failure, except for special device element.
When the special data memory for keeping for power failure is used for general use, please apply RST or ZRST commands to clear it content when starting.
When inter-PC simple link or parallel connection link is used, some data register is occupied by link.
Refer to document register D on usage of document register.

Special purpose

Data register for special purpose refers to writing in data for special purpose or writing special data into the data register in advance. When the power supply connected, it is set at the initial value and becomes 0 after clearing.ROM is used to write.
For instance, in D8000, time of monitoring timer is initially set by the system ROM. If it is to be changed, the transmission command F12 MOV is used to write in target time in D8000.



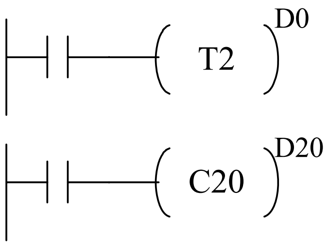
Refer to additional instructions for program memory and parameter structure for special data register for keeping for power failure.
Refer to additional instructions for basic functions on relevant description of data register type and function.

Case of action

Digit and data of data register can be used for control. Take basic command and application command for example, the data register can be used as expected. Refer to the following application commands.

<Data register for basic commands >

◆ Specified setting value of timer and counter



Timer/counter acts as per specified data register.

<Data register of application command> Such as F12(MOV) action case of command

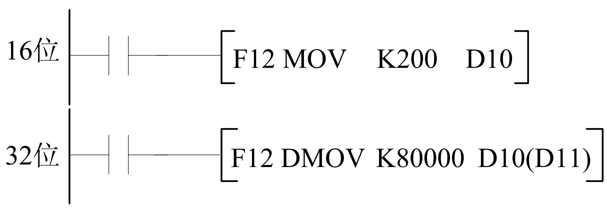
◆ Change current value of counter

◆ Read current value of timer/counter into the data register.

Current value of counter (C10) is sent to D4.

◆ Memorize the data in the data register.

◆ Store data in the data register.



Send 200 to D10 (Decimal system)

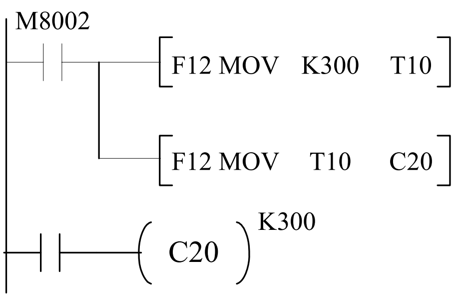
Send 80,000 to D10 (D11) (Decimal system).

Values exceeding 32,767 are 32-bit values, therefore, double (D) command is used. If low digit, bit (D10), the high digit bit (D11) is occupied.

◆ Send content of data register to other data registers.

Send content of D10 to D20.

<Unused timer and counter are regarded as data register > Take command F12(MOV) as example.



Send 300 to T10 (decimal system)

Send T10 to current value register of C20. At this time, T10 does not work as a timer but as data register.

When 32-bit is used, 2 16-bit data registers (such as C0 and C1) are used to express and 1 32-bit counter (such as C200 and etc.) is used to express.

9.2 Supplementary register W

Number of data register

Number of data register W is expressed as follows: (numbers are distributed according to decimal system):

TP03M/H 40/60 machine type:

Supplementary register W	W0~W9999　 10000points　※1

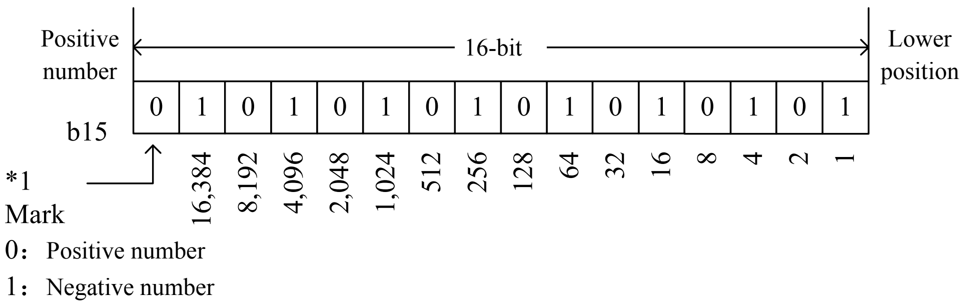
※1 In case of RUN→STOP or power failure, all the data of Supplementary register W are unsure.All the register W are used for general use.

Structure and function of register

Supplementary register is the device element for storing data and the type is expressed as follows. These register is 16-bit (positive and negative mark for the maximum digit bit). Combine the two data register to store 32-bit data (positive and negative mark for the maximum digit bit).

16-bit (W)

One data register (16-bit digit range) -32,768~+32,767

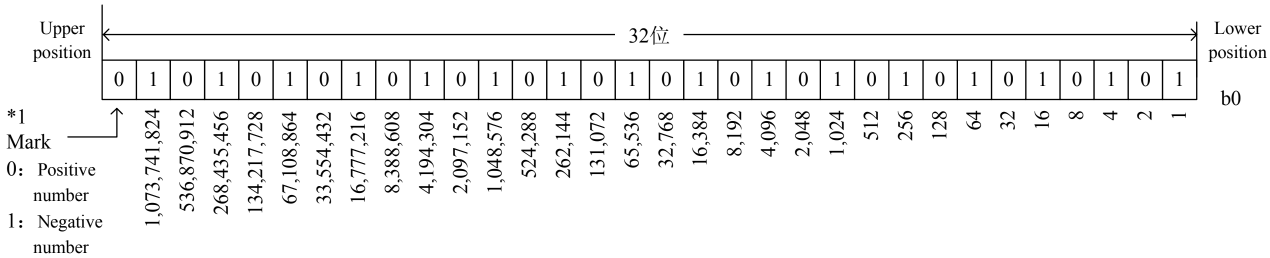


Readout and writing-in of data register employ application command. In addition, direct readout/writing-in can be executed from the unit (monitor) and programming equipment.

32-bit (W)

Two adjacent data registers is used to express 32-bit data (the high digit bit is a big number and the low digit bit is a small number, in Index register, V is a high digit and Z is low digit).

Therefore, we can process data of -2,147,483,648~+2,147,483,647.



When 32-bit is specified, if the following digit bit (such as W0) is specified, the number after the high digit bit (such as W1) will be occupied automatically. The low digit bit can be specified any device element of odd or even. Considering monitoring of PC/PDA link, the following even device element number is recommended.

9.3 Index register V， Z

Function and structure

Like common data register, the Index register V and Z is 16-bit data register for readout and writing-in data. There are 32 registers V0~V15 and Z0~Z15.

Besides the same using methods, such register can be used with other device element number or values in application command, and change device element number or value, so it is a special register.

In addition, pay attention that LD, AND, OUT and other basic sequence control command of PLC, or device element number of step echelon command and Index register may be used together.

16 bit

16bit	16bit
V	Z
V0~V15：8point	Z0~Z15：8point

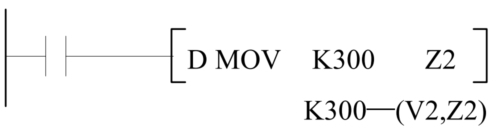
The two kinds of Index register have the same structure with the above data registers.

32 bit

32 bit	
V0(Upper position	Z0(Lower position)
V1(Upper position	Z1(Lower position)
V2(Upper position	Z2(Lower position)
V3(Upper position	Z3(Lower position)
V14(Upper position)	Z14(Lower position)
V15(Upper position)	Z15(Lower position)

When device element of 32-bit application command or values over 16 bits are process, Z0-Z15 must be used. See the combination of V and Z in the left figure. TP03 PLC takes Z as low digit bit of 32-bit digit. Therefore, even the high digit V0-V15 is specified, address-change can not be realized. In addition, if it is specified as 32-bit digit, for V (high digit) and Z (low digit) are referred simultaneously, if V is at high digit, and other rest digit may lead to error. Even 32-bit application command does not exceed 16-bit digit, for writing-in of Z, see the left figure. For DMOV and other 32-buit command, re-write 32-bit Index register for V (high) and Z (low).

Case of 32-bit search register writing:



K300—(V2,Z2)

Address-change of device element

For device element which may lead to address change, the content is described below:

Device element and value of decimal system: M, S, T, C, D, KnM, KnS, P, K.

For instance, V0=K5, when D20V0 is executed, the executed device element number is D2 (D20+5). In addition, the constant can be changed. For example, when K30V0 is specified, the executed element is the value K32(K30+5) of decimal system.

Device element of octonary system: X, Y, KnX, KnY.

For instance, Z1=K8, when X0Z1 is executed, the executed device element number is X10 (X0+8 plus of octonary system). As for device element address change of octonary system, content of V and Z are converted into octonary system digits, then plus computation is executed. Therefore, assume Z1=K10, X0Z1 is specified as X12, please not that this number is not X10.

Value of hexadecimal system: H

For instance, V5=K30 and specify H30, it is regarded as H4E (30H+K30). Besides, specify H30V5 with V5=H30, it is regarded as H60 (30H+30H).

Case of address change and attentions

Concerning address change of value of application commands and attentions, please refer to address change of value of Index register.

10 Number and function P/I of pointer

Pointer number

Pointer [P] and [I] are expressed as follows (distributed according to decimal system). When pointer for input interruption is used, the interrupted input code is distributed, which can not be used for high-speed counting and pulse wave density F56.

TP03M/H machine type

<table>
<tr><td>For branching</td><td colspan="3">For interruption input</td><td>For time interruption</td><td>For counting interruption</td></tr>
<tr><td rowspan="8">P0~P255
256
points</td><td>Input</td><td>Rising edge</td><td>Falling edge</td><td rowspan="8">I6□□
I7□□
I8□□
3 points</td><td rowspan="8">I010 I040
I020 I050
I030 I060
6 points</td></tr>
<tr><td>X000</td><td>I001</td><td>I000</td></tr>
<tr><td>X001</td><td>I101</td><td>I100</td></tr>
<tr><td>X002</td><td>I201</td><td>I200</td></tr>
<tr><td>X003</td><td>I301</td><td>I300</td></tr>
<tr><td>X004</td><td>I401</td><td>I400</td></tr>
<tr><td>X005</td><td>I501</td><td>I500</td></tr>
<tr><td colspan="3">6 points</td></tr>
</table>

TP03SR machine type

<table>
<tr><td>For branching</td><td colspan="3">For interruption input</td><td>For time interruption</td><td>For counting interruption</td></tr>
<tr><td rowspan="6">P0~P127
128
points</td><td>Input</td><td>Rising edge</td><td>Falling edge</td><td rowspan="6">I6□□
I7□□
I8□□
3 points</td><td rowspan="6">I010 I040
I020 I050
I030 I060
6 points</td></tr>
<tr><td>X000</td><td>I001</td><td>I000</td></tr>
<tr><td>X001</td><td>I101</td><td>I100</td></tr>
<tr><td>X002</td><td>I201</td><td>I200</td></tr>
<tr><td>X003</td><td>I301</td><td>I300</td></tr>
<tr><td colspan="3">4 points</td></tr>
</table>

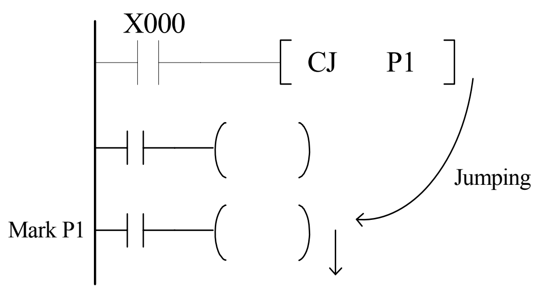
Case of function and action

Function and action of finger for branching pointer and interruption are stated below:

Almost all the fingers and application commands can be used together. Therefore, refer to the instruction manual for operation and instruction. Function and action of finger for branching pointer and interruption are stated below:

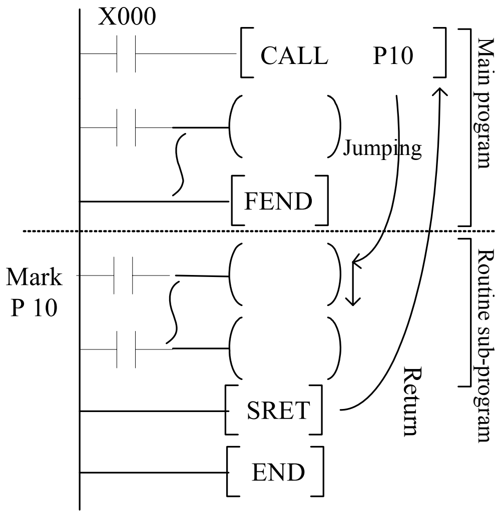
For branching

1 F00 (CJ)Conditional jumping



When X001 is ON, F00 (CJ) is jumped to specified position and execute the program

2 F01 (CALL)Accessing sub-program



For interruption

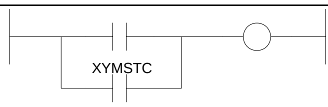
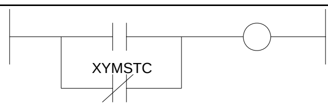
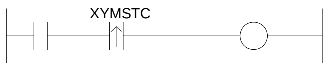
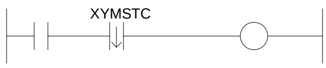
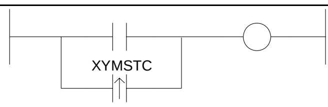
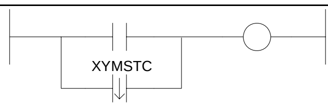
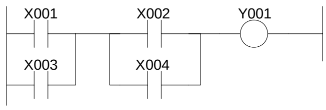
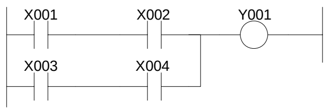
There are 3 types of pointer for interruption, application command FNC03 (IRET) for interruption return, FNC04(EI) interruption allowable and FNC05(DI) interruption forbidden.

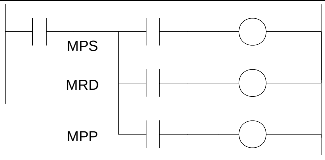
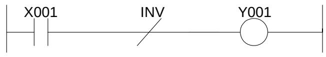
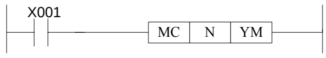
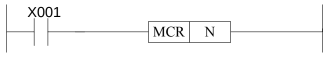
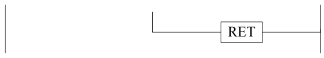
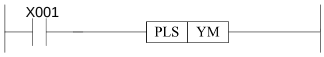
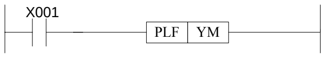
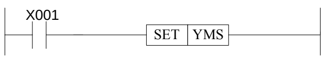
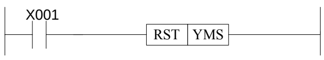
1. For input interruption: specified input number is not affected by PLC scanning cycle. When reading the signal, sub-program is interrupted. When input is interrupted, signal less than the scanning time can be read. During control of PLC, short time pulse wave signal can be processed in priority.
2. For time interruption: the specified interrupted time cycle (10ms-99ms), interrupt the sub-program and interruption processing program for fixed time outside PLC scanning time.
3. For counting interruption: compare the results from high-speed counter in PLC to interrupt sub-program, and prior control is realized by counting of high-speed counter.

Chapter III Interpretation of Basic Sequential control Sequence

1 List of basic instructions

List of basic instructions

Symbol	Function	Circuit form	Step
[LD]	Normally open contact for computation starting		1
[LDI]	Normally closed contract for computation starting		1
[AND]	Series connection normally open contact		1
[ANI]	Series connection normally closed contact		1
[OR]	Parallel connection normally open contact		1
[ORI]	Parallel connection normally closed contact		1
[LDP]	Computation starting of rising edge		2
[LDF]	Computation starting of falling edge		2
[ANDP]	Rising edge checks series connection		2
[ANDF]	Falling edge checks series connection		2
[ORP]	Pulse rising edge checks parallel connection		2
[ORF]	Pulse falling edge checks parallel connection		2
[ANB]	Parallel connection loop in series connection		1
[ORB]	Series connection loop in parallel connection		1

[MPS]	Computation and storage		1
[MRD]	Storage readout		1
[MPP]	Storage readout and reclosing		1
[INV]	Reverse		1
[MC]	Master control		3
[MCR]	Reclosing of master control		2
[NOP]	No action		
[END]	Programming scanning completes		
[STL]	Programming of step ladder style chart		1
[RET]	Completion of programming of step ladder style chart		1
[PLS]	Rising edge energizes coil		1
[PLF]	Falling edge energizes coil		1
[P]	Mark		——
[I]	Interruption mark		——
[OUT]	Coil		Y&M:1 S& specialM : 2 T:3 C:3(32bits) , 5(16bits)
[SET]	Setting coil		Y,M:1 S,special M:2 T,C:2 D&V&Z& special D:3
[RST]	Reclosing coil		
[SMCS]	Starting of loop branching		1
[SMCR]	Completing of loop branching		1
[JCS]	Starting of jumping branching		1
[JCR]	Completing of jumping branching		1

2 Interpretation of [LD]/ [LDI]/ [OUT]/[OUT I]

[LD]/ [LDI]/ [OUT]/[OUT I]

Interpretation of instructions

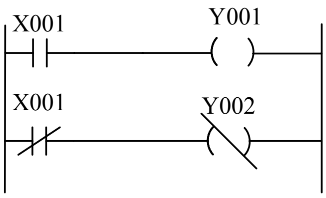
(1)[LD] Normally open contact and bus bar connection instruction, for X, Y, M, T, C and S.

(2)[LDI] Normally closed contact and bus wiring connection, for X, Y, M, T, C and S.

(3)[OUT] Coil drive instruction, for drive one specified coil with result of logic computation. For instance, output contact, auxiliary relay, step point, output coil of timer/counter, which can not be used for input coil X, only for Y, M, T, C and S.

(4)[OUT I] coil drive instruction is used for LDI of drive instruction [OUT], only for Y, M.

Programming case




```
LD    X001
OUT   Y001
LDI   X001
OUTI  Y002
```

Output of timer/counter

When instruction OUT is used for the timer and counter, the constants K and H are used to specify pre-setting vales and data memory can be used to specify pre-setting values of D, T, and C indirectly.

See the following table on setting ranges of time constant K and setting values of corresponding time:

Counter of timer	Setting range of value K	Actual setting values	Number of steps
1ms	1~32,767	0.001~32.676S	3
10ms	1~32,767	0.01~327.67S	3
100ms		0.1~3,276.7S	
16-bit counter	1~32,767	The same as left	3
32-bit counter	-2,147,483,648~+2,147,483,647	The same as left	5

3 Instructions AND, ANI

Instructions [AND]/[ANI]

Interpretation of instructions:

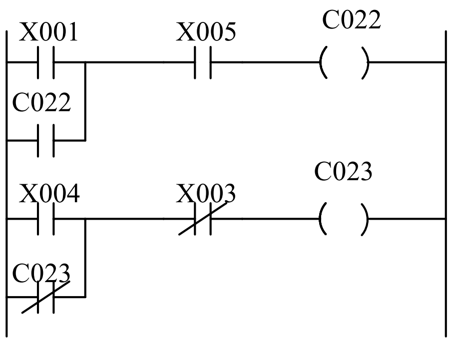
AND Normally open contact series instruction, for X, Y, M, T, C and S.

ANI Normally closed contact series connection instruction, for X, Y, M, T, C and S.

1) AND and ANI instructions are used for series connection of single contact, no restriction on quantity of series connection joint and using times.

2) [AND]/[ANI] instruction is used for series connection of single contact. If circuit block with two or more contacts in parallel connection is to be series connection, instruction ANB is used. Instruction ANB is the series connection instruction for parallel connection circuit block, without device followed.

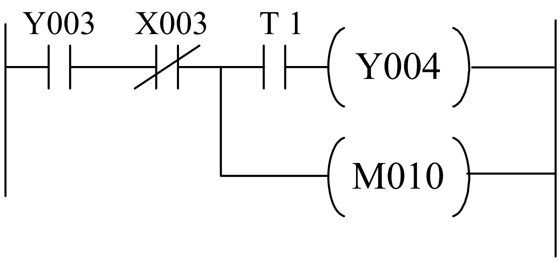
Programming case




```
LD    X001
OR    C022
AND   X005
OUT   C022

LD    X004
ORI   C023
ANI   X023
OUT   C023
```

Relation of MPS and MPP

If the ladder logic procedure is the following diagram, instructions MPS and MPP will be used.



4 Instructions OR， ORI

[OR]/[ORI]

Interpretation of instructions

[OR] is normally open contacts in parallel connection, for X, Y, M, T, C and S.

[ORI] is normally closed contacts in series connection, for X, Y, M, T,C and S.

When control circuits of the ladder logic is comprised of several contacts in parallel connection, instructions OR and ORI will be used.

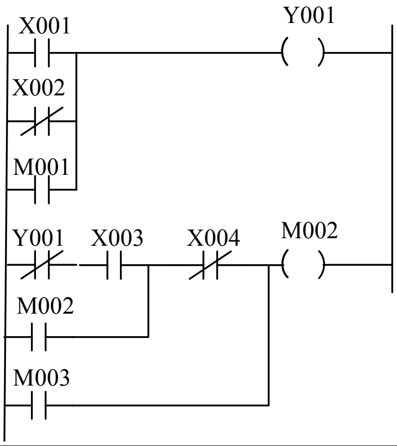
1) Instruction [OR]/[ORI] is used with said instruction [LD]/ [LDI] in parallel connection, no restriction on using times.
2) Instruction [OR]/[ORI] is only used for parallel connection of single contact. If circuit block with two or more contacts in series connection is to be parallel connection, instruction ORB is used. Instruction ORB is the parallel connection instruction for series connection circuit block, without device followed.

Parallel connection instruction of series connection circuit block ORB

ORB: make two or more series connection circuit blocks in parallel connection.

For circuit block with two or more contacts in series connection, when the series connection circuit block is to be parallel connection, instructions LD and LDI are used for starting end of the branch and instruction ORB is used for finishing end of the branch.

Programming

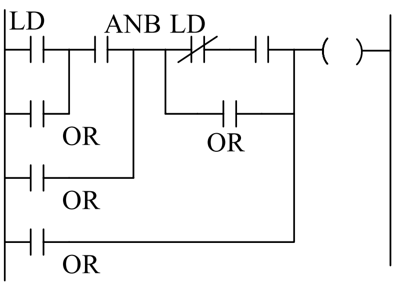



```
LD      X001
ORI     X002
OR      M001
OUT     Y001
LDI     Y001
AND     X003
OR      M002
ANI     X004
OR      M003
OUT     M002
```

Series connection instruction for parallel connection circuit block ANB

ANB: the instruction for connecting starting end of the parallel connection circuit block to the said circuit in series connection;

Circuit with two or more contacts in parallel connection is called parallel connection circuit block, and instruction ANB is used for connecting parallel connection circuit block in series connection.



5 Instructions LDP,LDF,ANDP,ANDF,ORP,ORF

Instructions

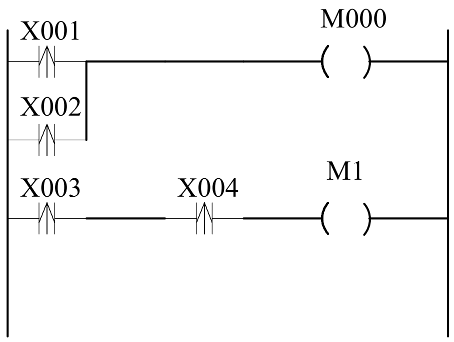
[LDP]/ [LDF]/ [ANDP]/ [ANDF]/ [ORP]/ [ORF]

Interpretation of instructions

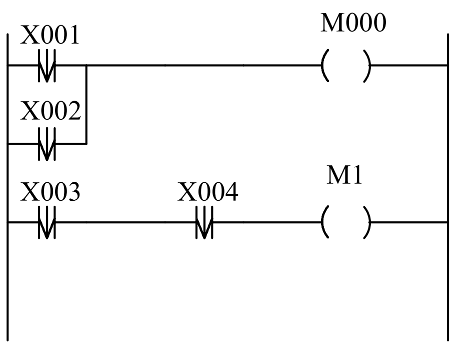
The instructions [LDP]/ [ANDP]/ [ORP] refer to the device operated by the instructions can connect one scanning cycle when triggering (OFF→ON) on the rising edge.

The instructions [LDF]/ [ANDF]/ [ORF] refer to the device operated by the instructions can be active one scanning cycle when triggering (ON→OFF) on the falling edge.

Programming



LDP X001
ORP X002
OUT M000
LDP X003
ANDP X004
OUT M1



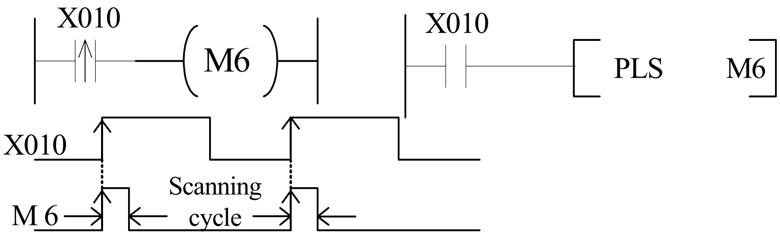
LDF X001
ORF X002
OUT M000
LDF X003
ANDF X004
OUT M1

In the above chart, when X001-X004 is switched from ON-OFF or OFF-ON, M0 or M1 is only connected to one scanning cycle.

Interpretation of actual drive of output coil:

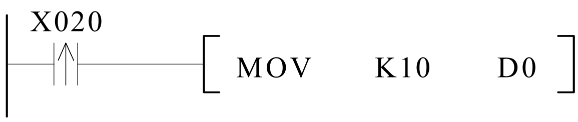
- The following circuits have the same drive effects:

OUT instruction and pulse instruction



In two circumstances, when X010 is switched from OFF-ON, M6 is only connected to one scanning cycle.

Pulse executing form of rising edge detecting and applied instruction



When X020 is changed from OFF-ON, data is D0 is transmitted for one time, and the two procedures have the same drive effect.

When previous condition logic results of the instruction MOV are ON, the data is transmitted continuously; when the logic results are OFF, the data transmission is stopped.

When previous condition logic results of the instruction MOVP is switched from OFF to ON, the data is transmitted, and such switches are in correspondence with times of data transmission.

6 Instruction ORB

Instruction [ORB]

Instruction interpretation

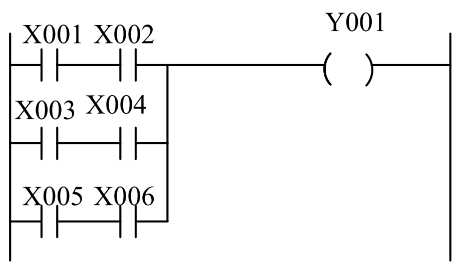
ORB: the instruction for parallel connection for two or more series connection circuits.

The circuit with more than two series connection contacts is called series connection circuit block. When parallel connection is used for series connection block, the instruction LD/LDI is used at the beginning of the branch and the instruction ORB is used at the ending of the branch.

1) [ORB] and [ANB] the same, it is a single instruction without device, without any device number followed.

2) In multi-parallel connection circuit, if each series connection circuit uses the instruction ORB, times of parallel connection is not restricted. The instruction ORB can be used continuously. At this time, times of repeated use of the instruction LD/LDI on one bus bar shall be less than 8 times.

Programming



LD X001
AND X002
LD X003
AND X004
ORB
LD X005
AND X006
ORB
OUT Y006

7 Instruction ANB

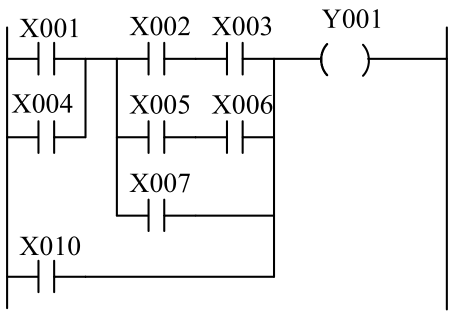
Instruction [ANB]

Instruction interpretation

ANB: the instruction for starting end of the parallel connection circuit block with previous circuit in series connection:

1) The circuit with more than two parallel connection contacts is called parallel connection block. The instruction ANB is required for series connection of parallel connection block; when it is connected to the previous circuit in series connection, the instructions LD and LDI can be used as starting end of the branch circuit, after finishing parallel circuit block of the branching circuit, the instruction ANB can be used for finishing series connection of the two circuits.
2) The instruction ANB does not follow any device, without any device number followed.
   When several circuits are in parallel connection, if each parallel connection block employs the instruction ANB for sequential series connection, quantity of parallel connection circuit is not restricted. The instruction ANB can be used collectively, but on the same bus bar, repeated use of the instructions LD and LDI must be less than 8 times.

Programming



LD X001
OR X004
LD X002
AND X003
LD X005
AND X006
ORB
OR X007
ANB
OR X010
OUT Y001

8 Instructions MPS,MRD and MPP

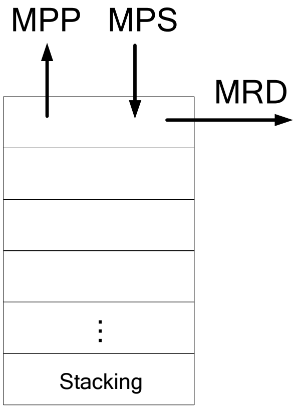
Instructions [MPS]/ [MRD]/ [MPP]

Instruction interpretation

(1) MPS (PUSH): push instruction.
(2) MRD (READ): read instruction.
(3) MPP (POP): pop instruction.

The group of the instructions can execute push protection for state of the contacts, when state of the contacts is required, the instruction pop is executed to ensure correction wiring with following circuit.

1) In the PLC, there are 8 memories for middle computation results, which are equal to stacking in microcomputer. One section of stacking is required according to the principle of first in and first out.



2) When the instruction MPS is used one time, computation results of the time will be positioned to the first stacking unit (called stacking top) of the stacking bottom. When MPS is used a second time, the computation results will be positioned to bottom top, and the former data will be positioned to the next stacking unit.

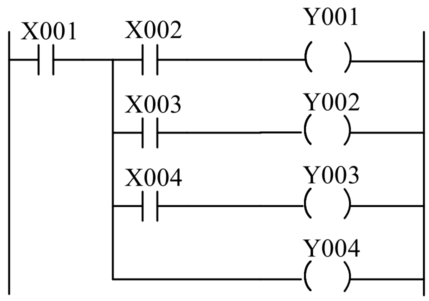
3) When the instruction MPP is used, the data is transmitted to the previous stacking. After it is sprang out, the data disappears from the stacking.

4) MRD is the special instruction for reading of stacking top, and data in the stacking will not be transmitted to the next or previous stacking.

5) Instructions MPS, MRD and MPP are without device numbers.

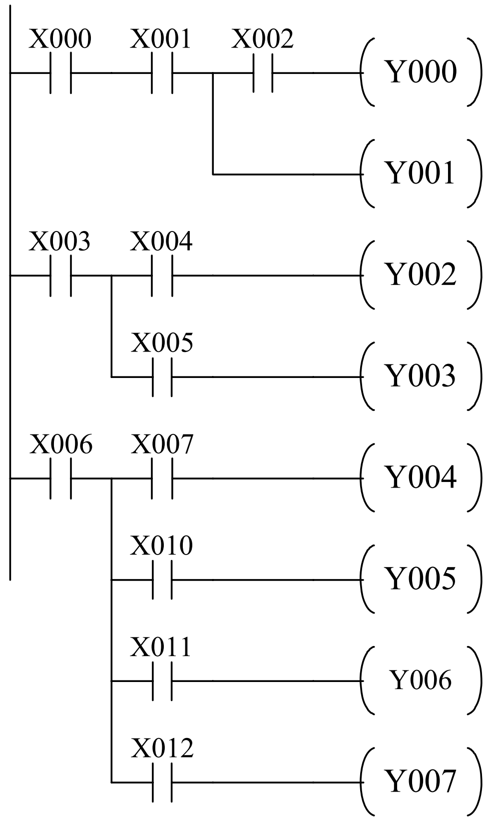
6) MPS and MPP shall be used in couple, and repeated Continuous use shall be less than 8 times.

Programming



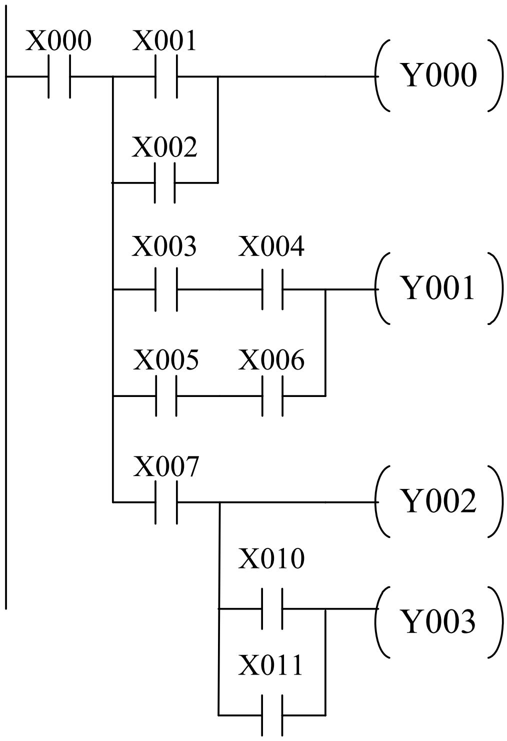

```
LD    X001
MPS
AND X002
OUT Y001
MRD
AND X003
OUT Y002
MRD
AND X004
OUT Y003
MPP
OUT Y004
END
```

Case of programming:
One section of stacking



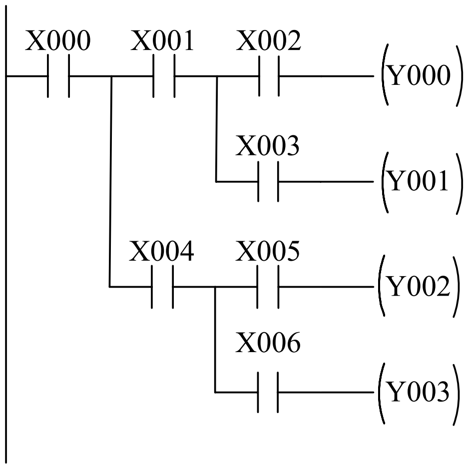
0 LD X000
1 AND X001
2 **MPS**
3 AND X002
4 OUT Y000
5 **MPP**
6 OUT Y001
7 LD X003
8 **MPS**
9 AND X004
10 OUT Y002
11 **MPP**
12 AND X005
13 OUT Y003
14 LD X006
15 **MPS**
16 AND X007
17 OUT Y004
18 **MRD**
19 AND X010
20 OUT Y005
21 **MRD**
22 AND X011
23 OUT Y006
24 **MPP**
25 AND X012
26 OUT Y007

One section of stacking, both applied of instructions ANB and ORB



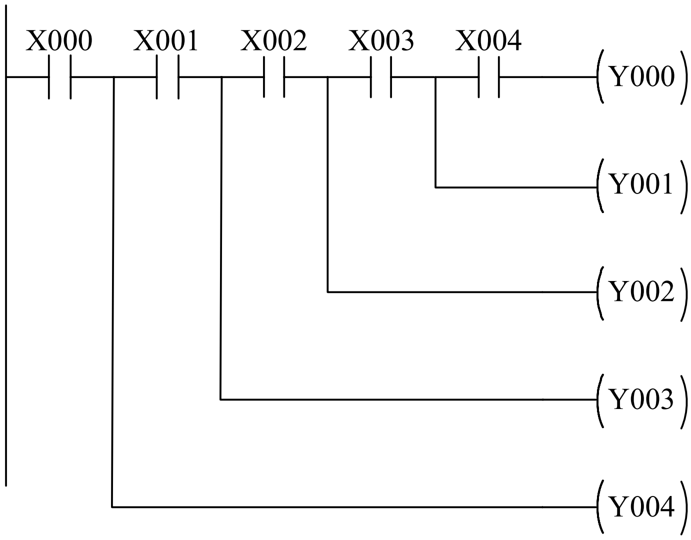
0 LD X000
1 **MPS**
2 LD X001
3 OR X002
4 ANB
5 OUT Y000
6 **MRD**
7LD X003
8 AND X004
9 LD X005
10 AND X006
11 ORB
12 ANB
13 OUT Y001
14 **MPP**
15 AND X007
16 OUT Y002
17 LD X010
18 OR X011
19 ANB
20 OUT Y003

Two sections of stacking

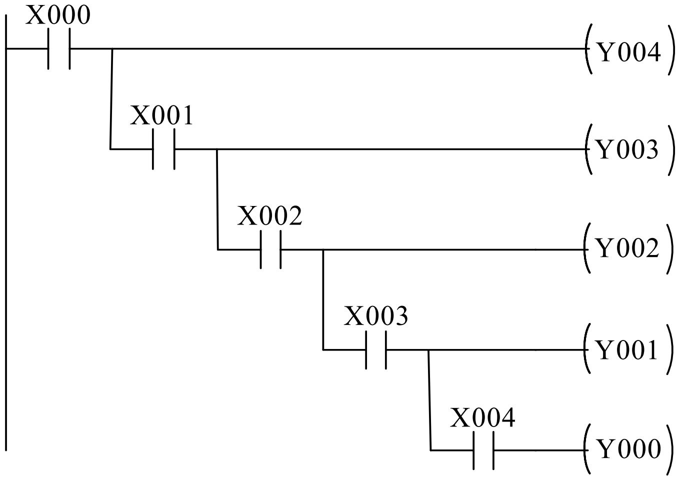


0 LD X000
1 MPS
2 AND X001
3 MPS
4 AND X002
5 OUT Y000
6 MPP
7 AND X003
8 OUT Y001
9 MPP
10 AND X004
11 MPS
12 AND X005
13 OUT Y002
14 MPP
15 AND X006
16 OUT Y003

Four sections of stacking




```
0  LD X000
1 MPS
2 AND X001
3 MPS
4 AND X002
5 MPS
6 AND x003
7MPS
8 AND X004
9 OUT Y000
10 MPP
11  OUT Y001
12 MPP
13 OUT Y002
14MPP
15 OUT Y003
16 MPP
17 OUT Y004
```




```
LD X000
OUT Y004
AND X001
OUT Y003
AND X002
OUT Y002
AND X003
OUT Y001
AND X004
OUT Y000
```

The above loop needs tri-MPS instruction programming.
However, if the following loop is used, the instruction MPS may not be used and it is to program.

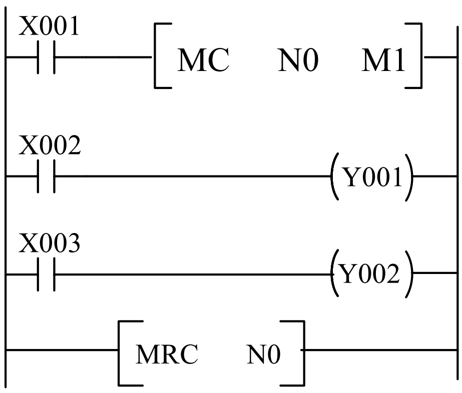
9 Instructions MC and MCR

Instructions [MC]/ [MCR]
(1) MC (Master control): connection instruction for public series connection contacts (new bus bar for public series connection contacts).
(2) MCR (Master control reset): reset instruction of MC instruction.
The two instructions are set at the starting point and ending point of the master control circuit block.

Instruction interpretation

1) In the chart below, when the input X001 is active, instructions between MC and MCR are executed; when X01 is inactive, devices between instructions MC and MCR are the following state: accumulative timer, counter and device driven by the instruction SET/RST keep current state; if non-accumulative timer and device driven by the instruction OUT, it shall be inactive.
2) After executing the instruction MC, the bus bar (LD and LDI) is transferred to MC contact. If it returns to original bus bar, the return instruction MCR is used. The instruction MC/MCR must be used in couple.
When using different device numbers Y and M, the instruction MC can be used repeatedly. If the same device number is used, like the instruction OUT, double coil output will occur.
3) The instruction MC can be used in nested way, namely, the instruction MC can be used the instruction MC. Number of the nested level is from small to big. When the instruction MCR is returned level by level, the number of nested level is from big to small.

Programming



LD X001
MC N0 M1
LD X002
OUT Y001
LD X003
OUT Y002
MRC N0

If the instruction MC is used in the instruction MC, number of the master control point shall be from small to big.
(N0→N1→N2→N3→N4→N5→N6→N7).When returning, the instruction MCR is released from big to small .(N7→N6→N5→N4→N3→N2→N1→N0)
the biggest number of nested level is 8 (N7)

10 **Instruction INV**

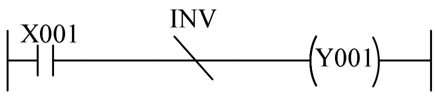
Instruction [INV]
The INV instruction is expressed with one short diagonal with an inclination angle of 45 degrees. The instruction INV is required before the computation results.

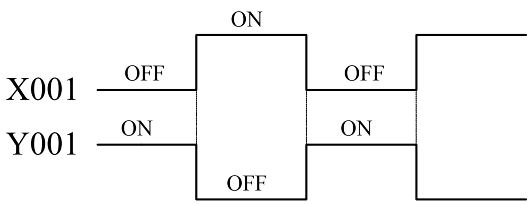
Instruction interpretation

The instruction INV has no device, without specified device number, and actions in the program are described below:

Computation results before executing the instruction INV	Computation results after executing the instruction INV
OFF→	ON
ON→	OFF

Programming





See the above chart, when the input relay X001 is inactive, the output coil Y001 is active. When X001 is active, Y001 is inactive. The instruction INV can be written at the same positions for inputting AND, ANI, ANDP and ANDF. The instruction INV can not be connected to bus bar like LD, LDI, LDP and LDF, and can not be used singly like OR, ORI, ORP and ORF.

11 Instructions PLS and PLF

Instructions [PLS]/ [PLF]

(1) PLS: differential output instruction, valid for rising edge;

(2) PLF: differential output instruction, valid for falling edge.

The two instructions are used for pulse output of goal objects. In case of input signal jumping, a pulse with width of one scanning cycle is generated.

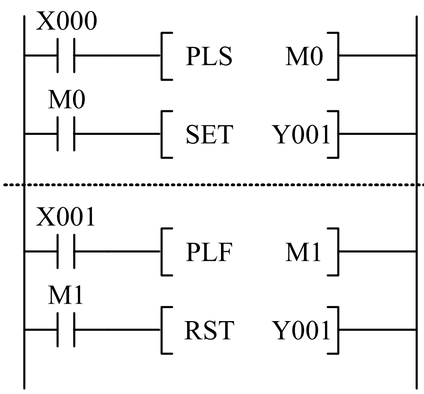
Instruction interpretation

When the instruction [PLS] is used, the drive input point is ON and the driven assembly has one scanning cycle.

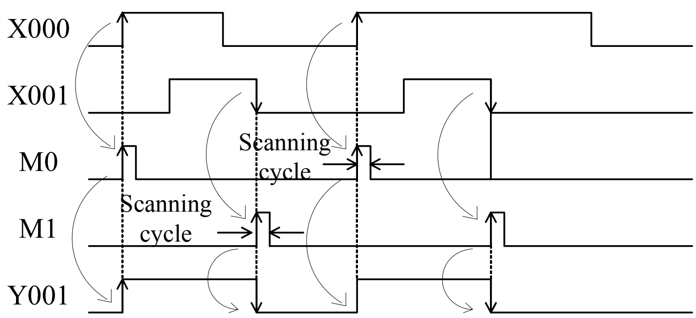
When the instruction [PLF] is used, the drive input point is OFF and the driven assemblies Y and M have one scanning cycle.

For instance, the input points X000 and X001 are active according to the following chart. When PLC is operated as per operation→ stop→ operation, it is known from logic relation of time sequence of analyzing program, when X000 is connected to the rising edge, M0 coil keeps powered and holds one scanning cycle, closing of M0 normally opened makes Y001 powered and reset 1; when X001 is connected to falling edge, M1coil is powered and holds one scanning cycle. Closing of M1 normally opened makes Y001 reset 0.

Programming




```
LD   X000
PLS  M0
LD   M0
SET  Y001

LD   X001
PLF  M1
LD   M1
RST  Y001
```



12 **Instructions SET and RST**

Instructions [SET]/[RST]

(1)SET (Set): Set instruction to keep the coil powered.

(2)RST (Reset): Reset instruction to keep the coil de-powered.

The instructions SET and RST used in applied program can set mark and clear mark for any state or time at any place for customers’ program.

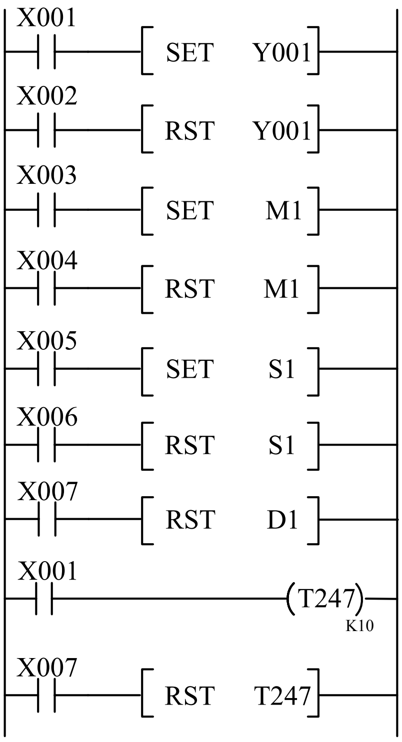
Indication interpretation

1) The instructions SET and RST have the function of self-holding. In the following procedure, when X001 is active, even if it is inactive, Y001 keeps active, once X002 is active, even if it is inactive, Y0001 keeps active.

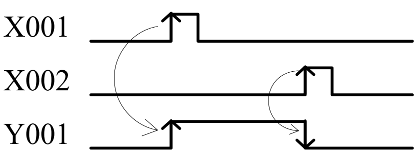
2) There is no restriction on use of the instructions SET and RST, other programs can be inserted into SET and RST, which is only effective when executing the last one.

3) In addition to Y, M and S for the instructions SET and RST, there are T, C and D, namely, clearing operation can be executed for data register D and index register and the timer T and counter C can be reset to clear the time and counting value.

Programming

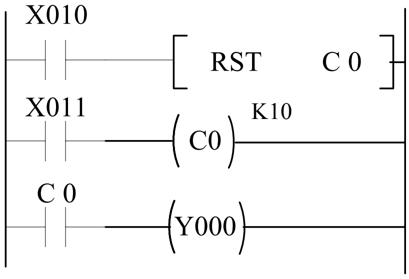



```
LD    X001
SET   Y001
LD    X002
RST   Y001
LD    X003
SET   M1
LD    X004
RST   M1
LD    X005
SET   S1
LD    X006
RST   S1
LD    X007
RST   D1
LD    X001
OUT   T247   K10
LD    X007
RST   T247
```



13 Instructions of output reset of counter (OUT, RST)

Programming

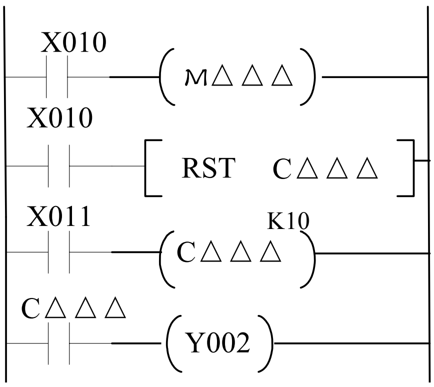


Interpretation of logic relation of the above program:

1) When the input point X011 from OFF—ON, the counter C0 starts increasing counting. When the counting value reaches setting value K10, the output contacts C0 acts and the output coil Y000 is active; when X011 is from OFF—ON, current value of the counter remains the same and the output coil Y000 is still active;

2) After the instruction OUT C, the constant K for counting is specified or the data memory is used for specifying indirectly. Only another input X010 is on, the counter C0 will be reset to 0 and the output contact Y000 returns.

Hi-speed counter programming



- When single-phase and single input counter of C235-C245 is used, special auxiliary relay M8235-M8245 shall be used for specifying the direction of counting. When X010 is ON, it is decreasing counting; when X010 is OFF, it is increasing counting.
- When X010 is ON, output contact of the counter C returns and current value of the counter returns to 0.
- If counter (C241 and C242) with the function of reset is used in the program, when the corresponding reset input is ON, the same effect with the above instruction can achieve through interruption and program is unnecessary for it.
- When X011 is ON, counting shall be executed for the counting input ON/Off of X000-X005 determined by number of the counter. With the counter (C244 and C245) with starting input, if the corresponding starting input point is not ON, counting can not be executed.

- Current value of the counter is increasing, when it reaches the setting value (constant K or content D), the output contact is SET; if it is lower than current value, it is RST.

14 **Instructions NOP and END**

Instructions [NOP]/ [END]

(1) NOP: No op instruction (or for deleting one instruction);

(2) END: program ending.

During debugging of the program, if the instructions NOP and END are appropriately used, it will bring convenience to users.

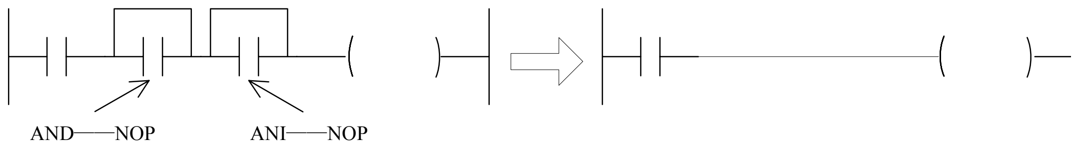
Instruction interpretation

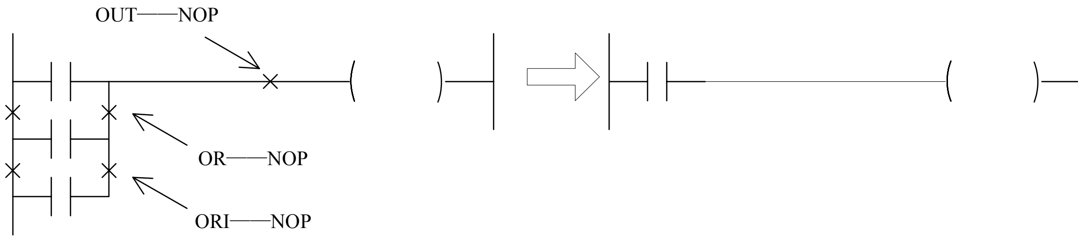
1) NOP is a no op instruction, and CPU will not execute the objective instruction. The instruction NOP shares one step sequence in the program, there is no corresponding device to express in the ladder-shaped chart, however, it can be reflected in the step sequence in the ladder-shaped chart.

2) After clearing all the executing instruction programs, all the instructions will be changed to NOP.

3) The program NOP can be inserted for minimizing times of change of step number when modifying or adding instructions.

As for finished program, when the instruction NOP is inserted, the program will not change. Please pay attention to it.





can be set by the instruction END in the program. The instruction END can be inserted by section, then debug section by section, after debugging, delete the instruction END.

15 Instructions SMCS and SMCR

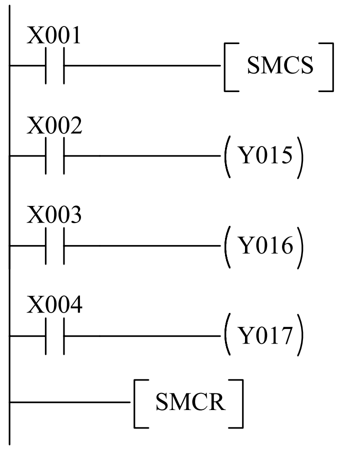
Instructions [SMCS] and [SMCR]

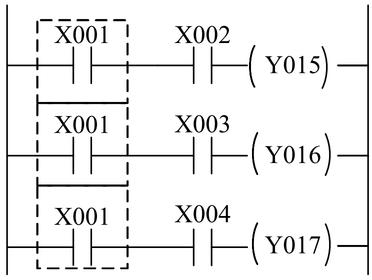
(1) [SMCS]: it is equal to one conditional bus bar, when the condition before the instruction is ON, the conditional bus bar is active.

(2) [SMCR]: the conditional bus bar returns.

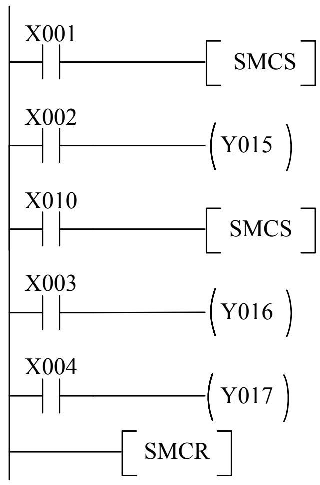
In the program SMCS and SMCR, the conditional bus bar must be used in couple. In the program, the instruction SMCS can be used for many times continuously or discontinuously, which is mainly used for the positions which require several occurrences in the circuit and it can be simplified.

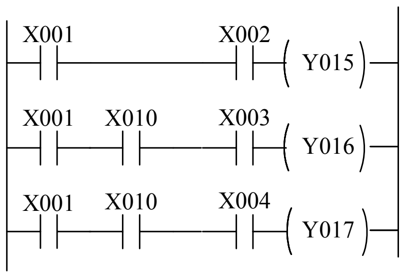
For example:





Instruction SMCS in the program can be used for many times, see the chart below. After it is used for one time, one condition for auxiliary bus bar is added. After the instruction SMCS has been used for many times, only one SMCR instruction can clear all the conditions.





Each instruction after SMCS and before SMCR can execute computation with the condition before SMCS.

When the common circuit is pretty complicated or reoccurs several times, such instruction can simplify the program.

Note: OUT, TMR, CNT and applied instructions can not be after the instruction SMCS directly.

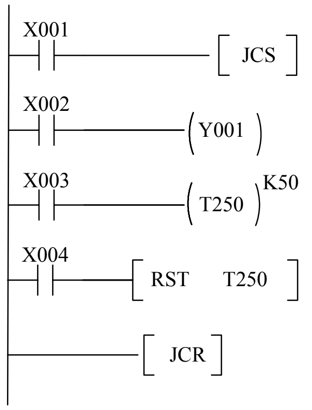
16 Instructions JCS and JCR

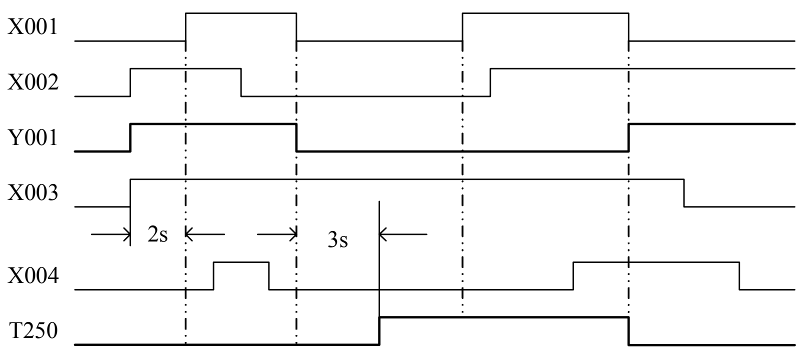
Instructions [JCS]，[JCR]

(1) [JCS] Jumping branching starting

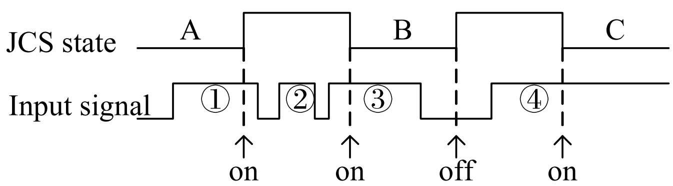
(2) [JCR] Jumping branching returning

All the instructions after JCS and before JCR will not be executed, namely, during JCS conditional input is ON, content of the memory will remain the same. The instruction END is not allowed between JCS and JCR, otherwise, the program will be error and the EER instruction light is ON.





Note1: Pay special attention to timing signal of the timer, and relative time relation of input signals of counter (switch from OFF→ON) and applied instruction and JCS ON/OFF state.



- When X002 is at OFF→ON of ①, Y001 will act, for state of JCS is OFF.
- When X002 is at OFF→ON of ②, Y001will not act, for state of JCS is ON.
- OFF→ON in ③ will not act, for state of JCS is ON.
- State of [JCS] will be switched into OFF when X002 is at ③ON, Y001 will not act; for in state (A) of JCS, Y001 will be switched from OFF to ON with input signal. When JCS is ON, it will not be affected by state change of ON→OFF or OFF→ON; for JCS is off (state B), it will be switched from OFF→ON, and Y001is switched from ON→OFF.

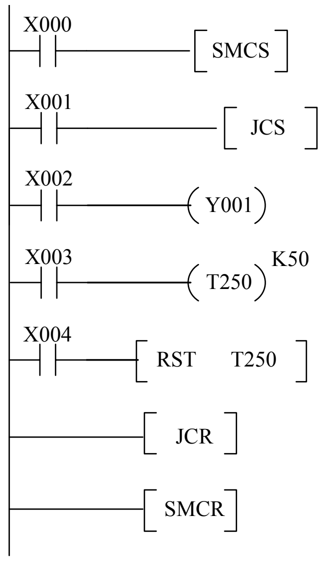
- When input signal in ④ is from OFF→ON, Y001 is still OFF and will not act. For state of JCS is ON.
- State of [JCS] is OFF when ④ is ON, Y001 acts.

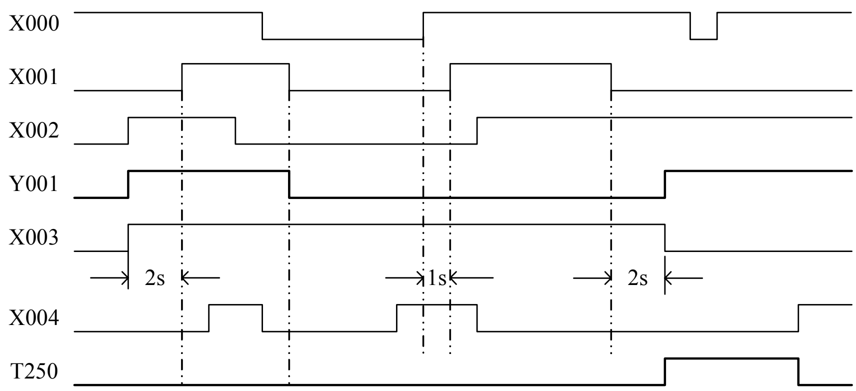
Note 2: when state of [JCS] is ON, instructions effecting positions between JCS and JCR will not be executed.

Note 3: the instruction between JCS and JCR will be executed certainly, which is not affected by state ON or OFF of JCS.

At this time, execution of the program will be suspended and the next scanning cycle is entered.

Note 4: instructions between JCS and JCR can be inserted between SMCS and SMCR.





Note 6: another JCS is inserted between JCS and JCR, and only one JCR can be used as state ending.

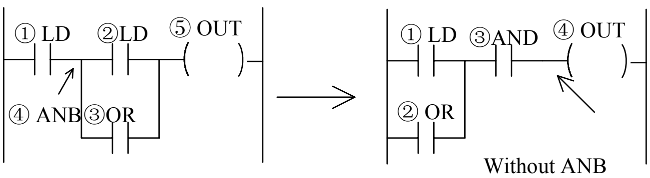
17 Attentions for programming

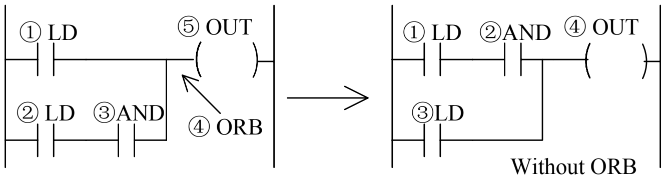
17.1 Step and executive sequence of program

1) Ladder in the ladder-shaped chart starts from the left bus bar and ends in the right bus bar. Each row at the left side is the combination of contacts, which represents the conditions for driving logic coil, and the logic coil representing the results can only located at the right bus bar. The contacts can not be at the right side of the coil.
2) The contacts shall be drawn on the horizontal line, not the vertical line.
3) When series connection and parallel connection are employed, branch with more contact shall be located at the left side of the ladder-shaped chart; when parallel connection is for blocks in series connection, the parallel connection branch with more contacts shall be located above the ladder-shaped charter.
4) Double-coil output is not recommended.

Structure and step consequence of contacts

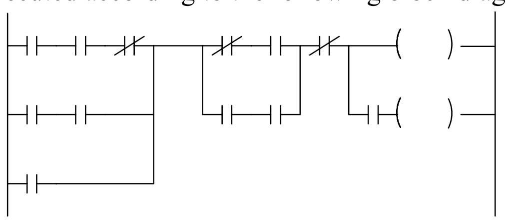
As for circuit of the same program, according to forming mode of the contacts, the program can be simplified and saved in capacity.

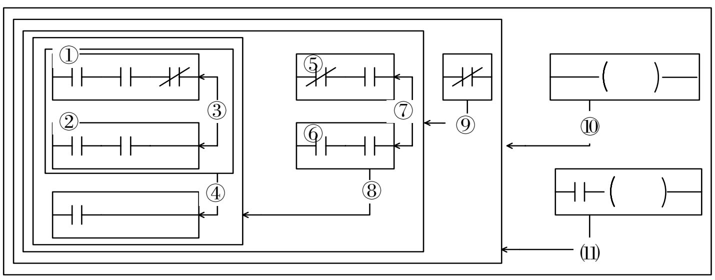




Executing sequence of program

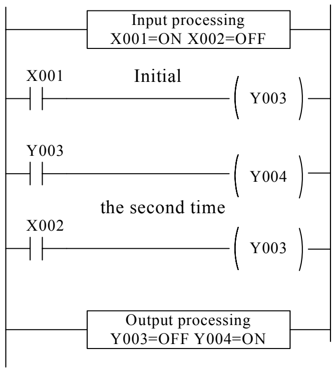
The program is processed from up to down and from left to right. Flow of the program instruction is executed according to the following block diagrams.





17.2 Action and countermeasures for double-coil of double output

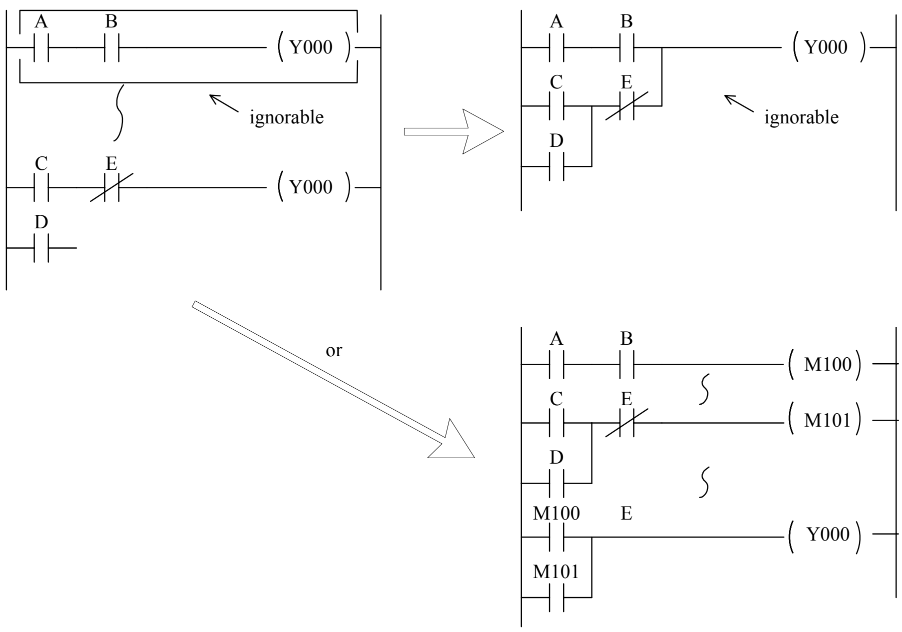
If double output (double-coil) is used in sequential control program, the later action shall execute in priority.



See the chart at the left side, pay attention to use of the same coil Y003 in several positions.
For instance, when 001=ON and X002=OFF,
At the beginning Y003, for X001 is active, the RAM is ON and the output Y004 is active.
Y003 is inactive for a second time by X002, and the RAM is OFF.
Therefore, actual external output is Y003= OFF and Y004=ON.

Countermeasures for double output

The double output does not disobey the rule in program, but the actions are very complicated, therefore, the following program is recommended.

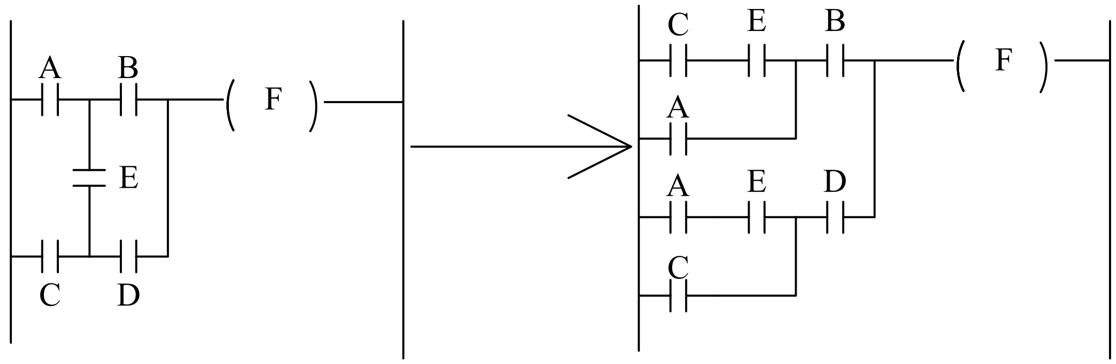


Skipping instruction and step ladder style instruction can be used and the above program is switched to the same output. When the step ladder style instruction is used, in the master program and state program, use of double output and the same output shall be paid attention to.

17.3 Loop and countermeasures for unavailable programming

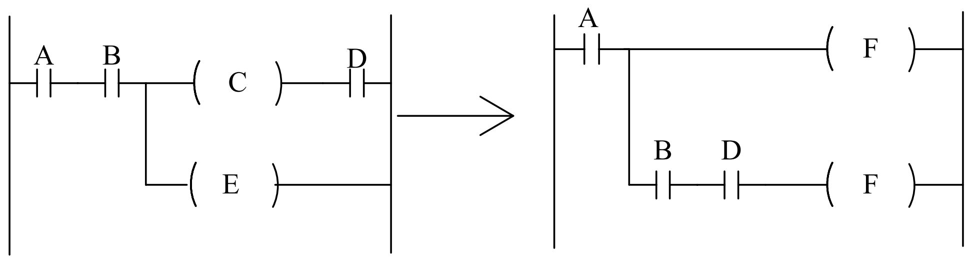
Bridge circuit

See the chart below: change direction of flow of the bi-directional loop (parallel connection for loops without D and B).



Connecting position of the coil

- Do not write contact at the right side of the coil.
- Coil among contacts shall be programmed first.



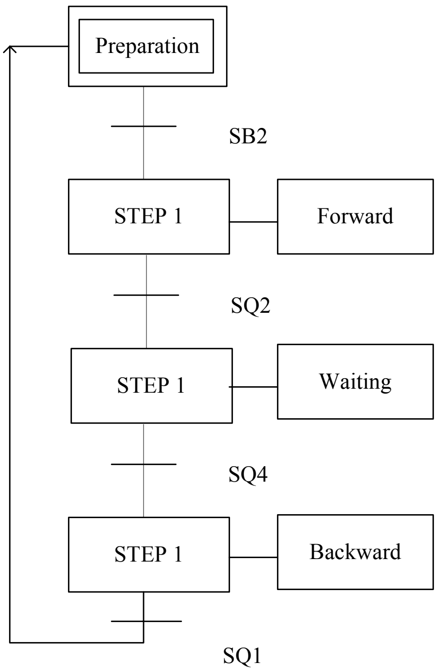
Chapter IV Instructions of step ladder style programming chart

Programming languages of PLC——Sequential function chart (SFC)

Sequential Function Chart is an illustrated method for describing sequential control function. As for complicated sequential control system, internal interlocking is very complicated. If it is programmed with step chart, the programming step will be too long and its readability is greatly reduced. SFL represented mechanical action, with state transfer programming, especially for complicated sequential control program.

Programming train of thought for programming with SFC:

(1) According to requirements of program determined by structure, one complicated control procedure can be divided into several working steps; such working step is called state. The state is separated by transferring. Adjacent states have differentiations. When transferring conditions of adjacent state are satisfied, transferring can be realized, namely, from the previous state to the next state.



(2) Element of SFC is made up of state, transferring and oriented segment.

① **State:** It represents one working step (action), the state symbol is represented in single line frame. The assembly number is in the frame. One control system shall have one initial state, which is correspondence with the operation origin and symbol of the initial state is double line frame.

② **Transfer:** It represents change from one state to another. Oriented segment is used to connect the transfer to indicate the direction. Vertical line on the oriented segment and the marked symbol represents conditions of state transfer.

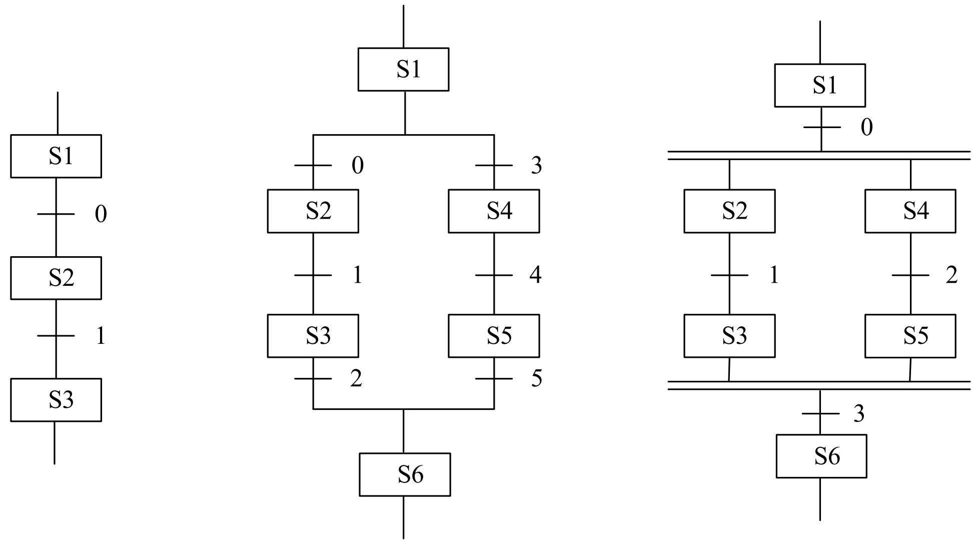
③ Actions in correspondence with state are represented with one or several rectangles at the right side.

(3) Basic forms of SFC, which is divided into three forms:

① **Single flow structure:** it refers to the state is operated one after another, each state is connected to one transfer and vice versa;

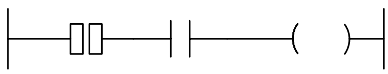
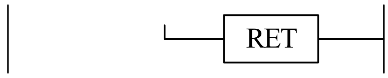
② **Selective structure:** it refers to several single flow branches after one state. When the corresponding transferring conditions are satisfied, one single flow branch can be selected one time. Transferring conditions of selective structure is one horizontal line is active after the state and the first transfer is active under the horizontal line. After ending of the single flow branch, one horizontal line shall be used to represent and there should be no transfer under it.

③ **Parallel structure:** it refers that under some transferring, if the transferring conditions are satisfied, several single flow branches can be triggered simultaneously, and these parallel sequential branches shall be written between the two lines.



(a) Single flow structure (b) Selective structure (c) Parallel structure

1 Step instructions STL, RET

Instruction symbol	Function	Circuit representations	Step
[STL]	Starting of step ladder		
[RET]	Finishing of step ladder		

STL and RET are a couple of step instructions, which means starting and ending of step instructions.

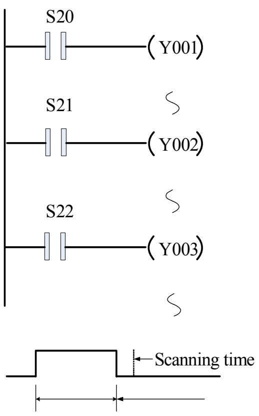
Instruction interpretation

(1) STL (Step ladder): connected to master bus bar normally open contact instruction, the contact instruction TL is used to represent with normally open contacts with two small rectangles, namely:

(2) RET (Return): return to bus bar instruction.

The step ladder is the starting of working step control for executing step point of internal devices. The initial state must start with S0-S9, and RET is the ending of the step point (S), meanwhile, it must be ended with S0-S9 and the program returns to the bus bar. At the end of the step ladder style programming chart, the instruction RET is required. In one program, totaled 10 step flows in maximum can be written and each step flow requires the instruction RET to end. Write the state adder style programming chart according to the following rules. SFC and ladder style programming chart are exchangeable.

Programming and actions



Repeated Use of state action and output

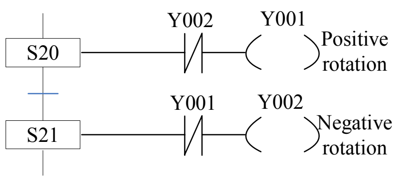
●Note: the state symbol S can not be used repeatedly.

●If the instruction RET is not written at the end of programming, editor of PC 06 software can perform self-check and add; automatically PC06 software When logic relation in SFC is correct, you can see the added instruction RET in the mnemonic view and ladder style programming chart.

●If the contact STL is active, there are some relevant loop actions.If the contact STL is inactive, there are not any relevant loop actions. After one scanning cycle, the instruction will not be executed (skipping state).

●See the left chart, during different status, the same coil (Y002) shall be output when programming. At this time, S21 or S22 is active, the coil Y002 will be active in different step sequence. In ladder-shaped programming, owing to complicated action of dual-coil, so it is not recommended.

In addition, when SFC is used for editing, when program the same output coil (Y002) for the state in the master program or vice versa, it shall be processed in dual-coil of the ladder style programming chart. Please pay attention.



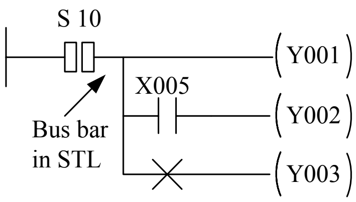
Output interlocking

During state transfer, it is active within shortest time (one scanning cycle). Therefore, to avoid failure of simultaneous active of a couple of outputs, interlocking can be set on PLC according to manual of the PLC.

Repeated use of timer

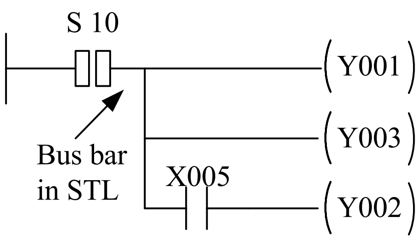
●In SFC, the timer coil can set points for the same timer under different states as the output coil, however, it can not be set under adjacent state; if the same timer is set under adjacent state, in case of state transfer, the timer coil will not be inactive and the current value can not be reset.

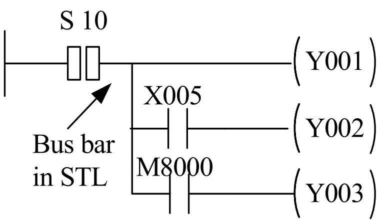
Instruction applied for state

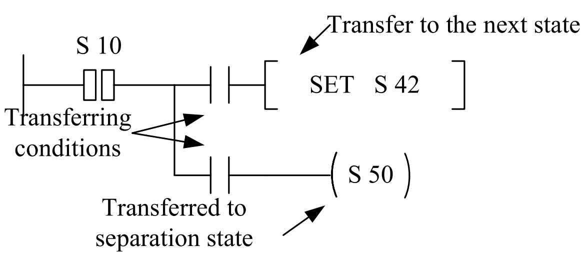


Driving method of output

See the chart on the left side, from bus bar in the state, once LD or LDI is written, instruction for not required contact shall not be programmed. Method in the following chart shall be used to modify such loop.







The instructions OUT and SET have the same functions for state (S) after the instruction STL and they will reset the transferring source. In addition, it has the function of holding. However, when the instruction OUT is used, it is used for transferring to separation state.

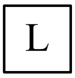
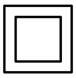
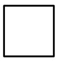
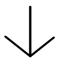
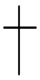
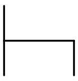
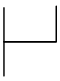
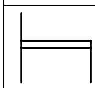
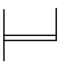
List of sequential control instruction in the state

State \ Instruction		LD/LDI/LDP/LDF, AND/ANI/ANDP/ANDF/ OR/ORI/ORF,INV,OUT SET/RST,PLS/PLF	ANB/ORB MPS/MRD/MPP	MC/MCR
Initial state/general state		Available	Available	Unavailable
Branching and merging	Output processing	Available	Available	Unavailable
	Transfer processing	Available	Unavailable	Unavailable

In interruption program and sub-program, the instruction STL can not be used.

In the instruction STL, the skipping instruction is not forbidden. However, the action is so complicated and not recommended.

Symbol interpretation in the SFC

Symbol	Description
 	Ladder-shaped mode, indicates internal edition program, only general ladder style programming chart not step ladder style programming program
	Initial step point, representing the chart for initial step, the available setting range is S0~S9.
	General step point, the available setting range is S10~S1023
	Step point skipping, the step point state transferred to non-adjacent step point
	Transferring conditions of step point, state transferring conditions between two step points.
	Select branching chart, from the same step point transferred to the corresponding step point under different transferring conditions
	Select merging chart, more than two different step point state transferred to the same step point with the same transferring conditions
	Parallel branching, from the same step point transferred to more than two step points with the same transferring conditions
	Parallel merging point, when more than two step points are established, they are transferred to the same step point with the same transferring conditions.

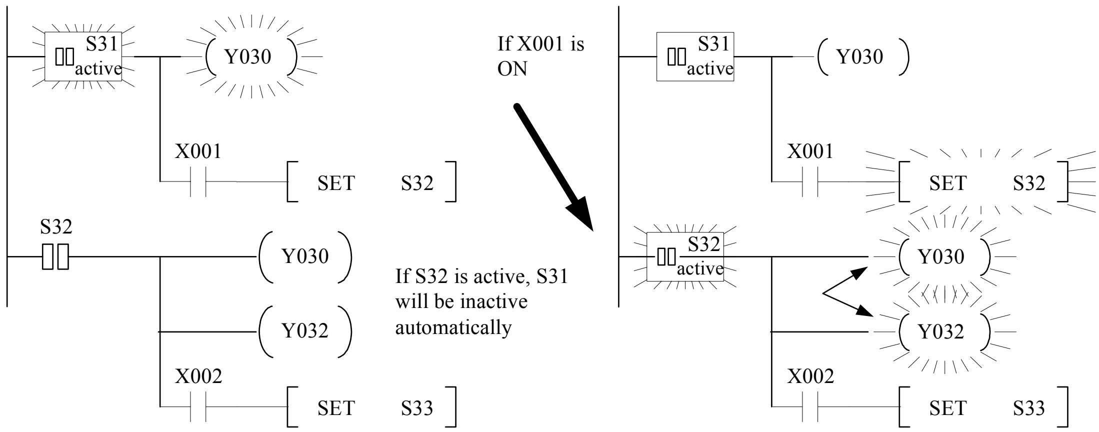
2 Action and SFC representation of step ladder logic

Function of instructions

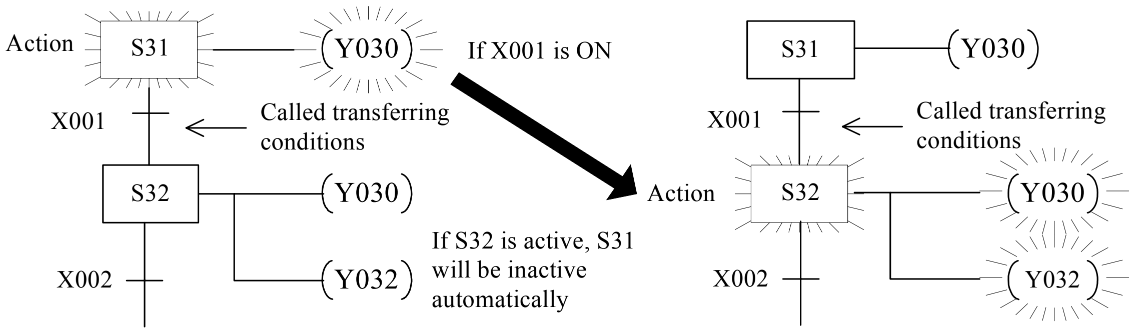
Sequential control edition of SFC is built in PLC of TP03. The SFC chart can be inverted into instruction sheet or ladder style programming chart program, and from instruction sheet or ladder-shape chart to SFC chart.

●In SFC, each state can be regarded as micro-control working step. The input conditions and output control can be programmed according to the sequence. The biggest feature of the control is that when current working step is in progress, the previous working step does not work, the working steps operate according to program to realize step control.

The step ladder style programming instruction can be represented in the following actions.



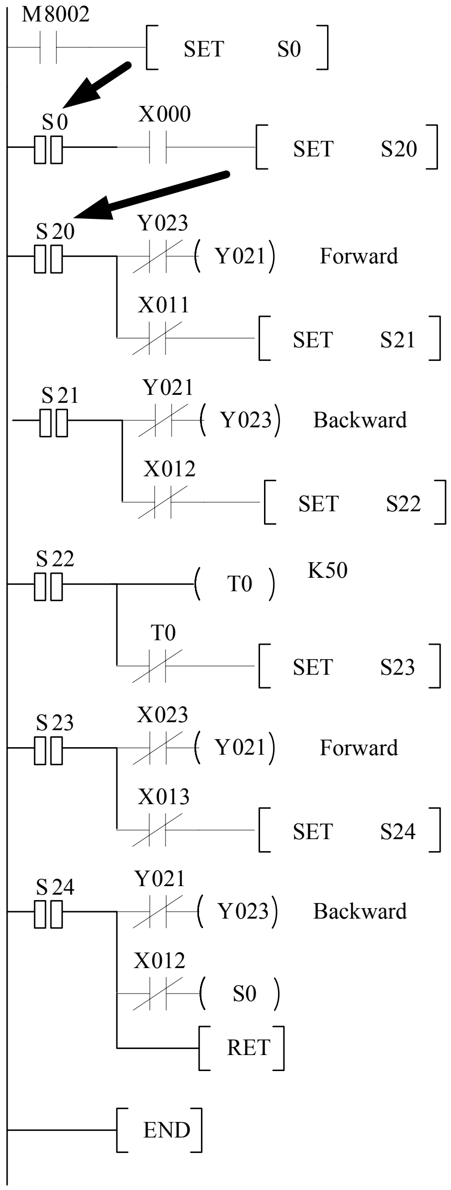
If the SFC chart indicates step ladder style programming loop in the left chart, see the following chart:

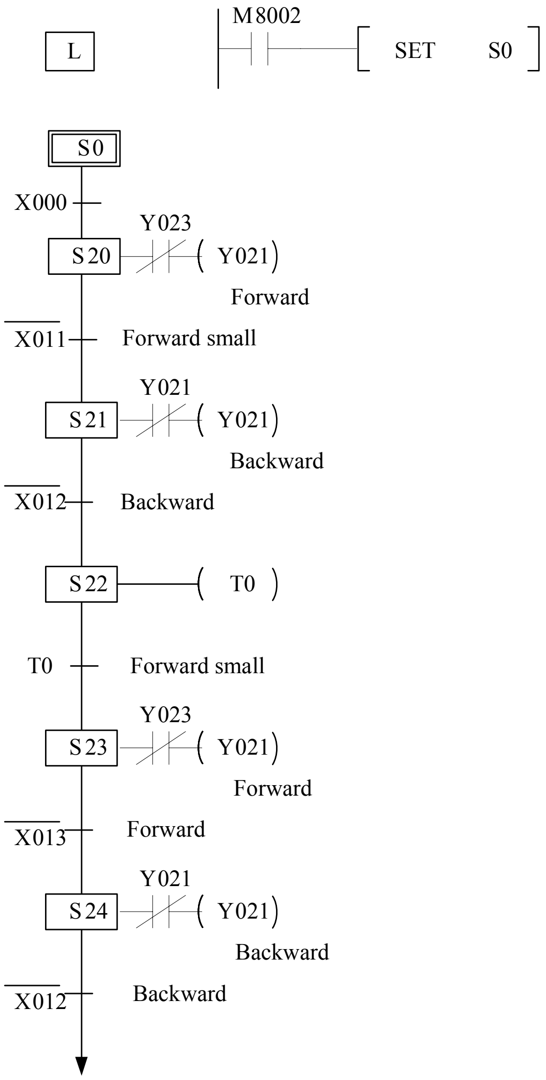


In the SFC chart, function of each piece of equipment and the whole process are understandable, the sequential control design is easy, even to the third person, it is used for maintenance, modification and troubleshooting. The SFC chart and step ladder style programming chart shall be programmed according to rules, which are exchangeable with the same content. The familiarized ladder style programming chart can be used. When editing SFC chart, corresponding peripheral equipment and programming software.

Actual representation of the instruction

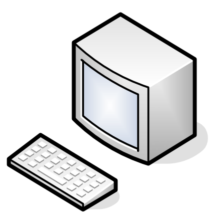
According to the above-mentioned statement, the step ladder style programming chart and SFC chart have the same content, with the following representations. STL chart is represented in ladder, and SFC is based on state (working procedure) to represent the flow in mechanical way.





Programming equipment

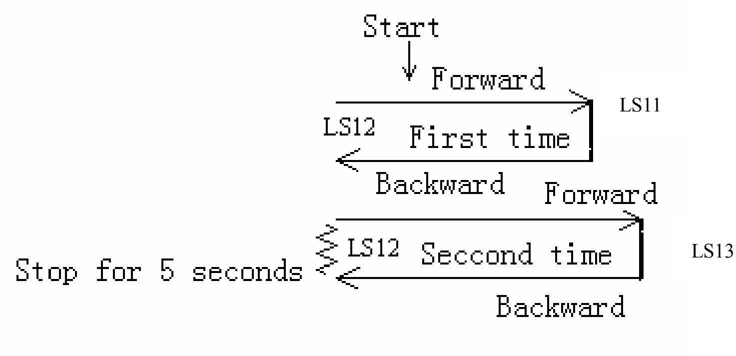
SFC can be programmed with personal computer (PC06 user's software); sequential control program programmed by SFC can be saved in TP03 PLC in the form of instruction.



3 Features of SFC

Simple action case

When mechanical action is filed for others' reading, it shall be programmed in the form of single program in accordance with time sequence chart or structural chart.



Actions

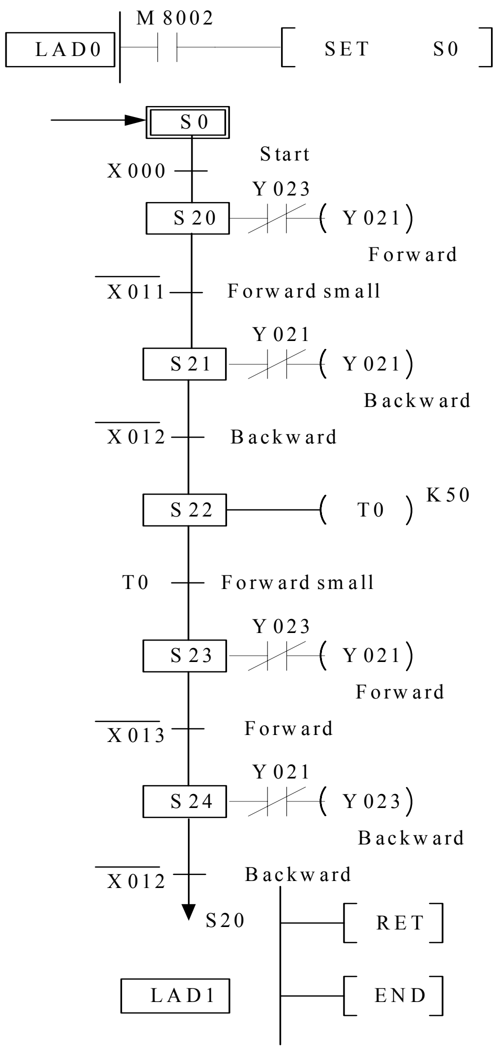
1. Press the starting button, the trolley moves forwards, when the inching switch LS11 acts, it moves backwards. (LS11 is normally ON, when it moves forward at limit, it is OFF, the same for other inching switches).
2. When moving backwards, the inching switch LS12 acts, after stopping for 5 seconds, it moves forwards, after the inching switch LS13 acts, it moves backwards.
3. When the inching switch LS12 acts a second time, drive motor of the trolley stops.

When actions are described correctly, complicated mechanical actions will be simple. The mechanical technicians and electric technicians can have deep discussions.

Basically, if electric technicians want to design PLC program, without action chart of PLC, it can not be designed.
However, the step action is very complicated for electric technicians, which required rich experience and designing time. And PLC chart seems pretty complicated from the third person, and such designer of PLC takes responsibility for mechanical maintenance.

Processing of single flow

Basic form of working step transfer is the control of single flow. In sequential control of single action, only single flow is adequate. With input conditions and operators, with following selective branching and parallel branching, complicated conditions can be treated in a simple way.



· In ladder circuit block LAD0, the auxiliary relay M8002 with action in a shortest time can make the initial state S0 reset (ON) during switching from stop-start switching of PLC.

· As for provided initial working step, such distributed PLC is called initial state device of S0-S9.

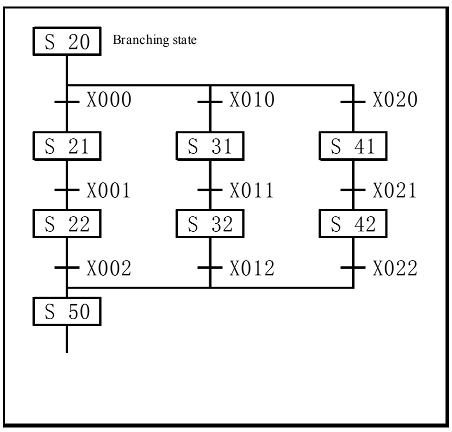
· As for distributed state S02-S889 for the working steps, including the state for holding in case of power failure, it can keep the state. In addition, when S10-S19 employs the instruction IST, they are also for special purposes.

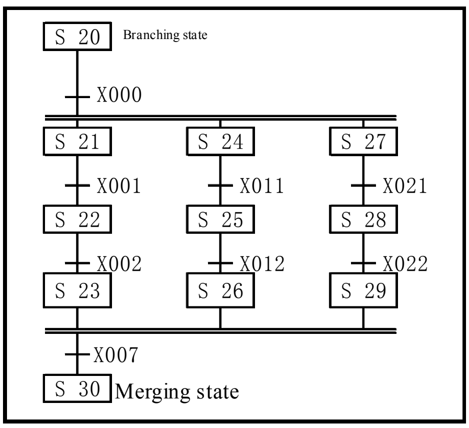
· Timer, counter and auxiliary relay and other devices are set in PLC, which can be used at your discretion. Such timer T0 takes 0.1 second as unit action. Therefore, the setting value is K50, after the coil is driven for 5 seconds, the output contact acts.

Selective processing and simultaneous processing of several working steps

When executing one of several flows, it is called selective branching.

When executing several flows simultaneously, it is called parallel branching.

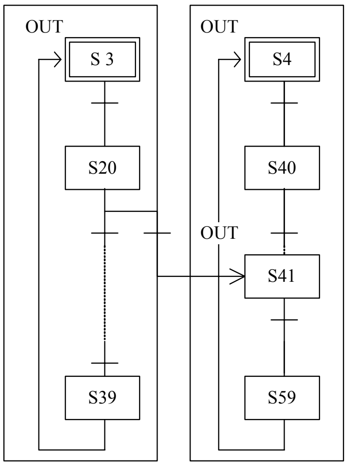




4 Prepared knowledge for programming SFC flow

Separation of flow

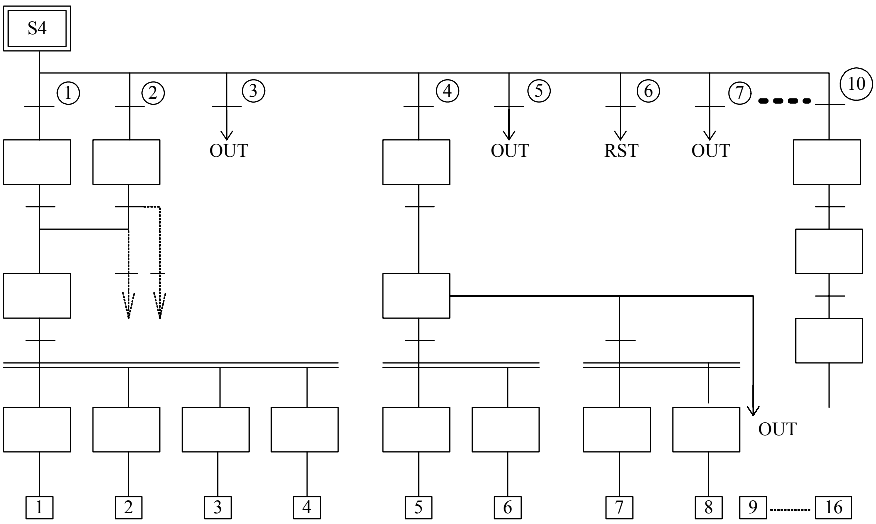
SFC program has several initial state, and each initial state is separated program.



Take the above chart as example, after the instruction STL of S20-S39 of the initial state S3 is executed, then relevant programs to S4 are executed then.
In the program, the instruction except the instruction STL can be used to execute other state.
In the left chart, the initial state S3 includes the instruction OUTS41. The initial state S4 includes the instruction LD S39. Do not confuse the instruction STL.

Restriction of branching loop quantity

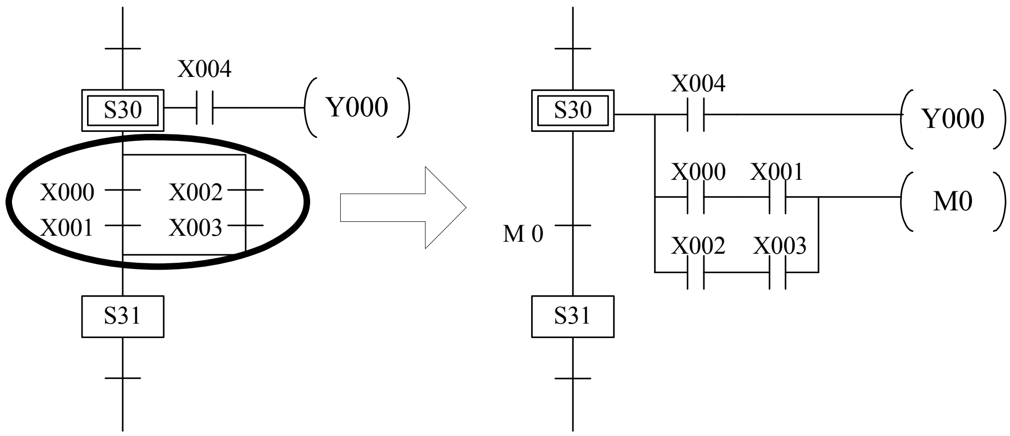
Quantity of one parallel branching or selective branching shall be less than 10.
However, if there are several parallel branching or selective branching, total loop quantity of each initial state may not exceed 16.



State from or before merging to the separation state by transferring or reset are not allowed. No op state must be set. Transferring and reset can only be executed from the branching.

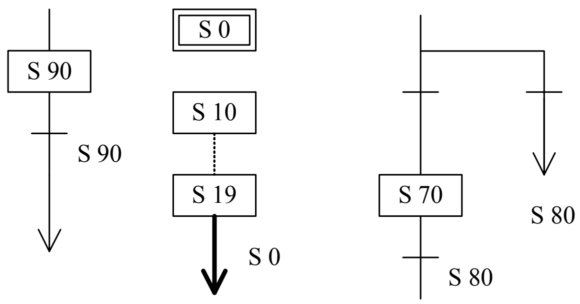
Program of complicated transitional conditions

There is only condition for each node in TP03 SFC program. If program shall be written as per the chart below at the left side, method at the right side shall apply.



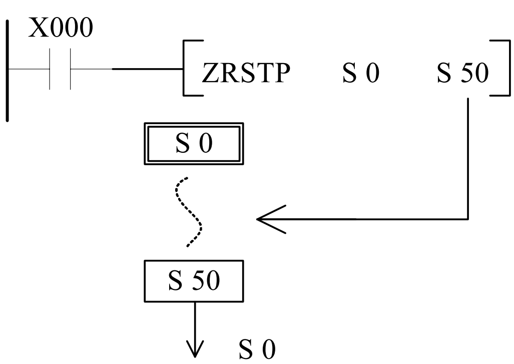
Instructions MPS, MRD and MPP can not be used in transitional conditional circuit. The right method shall be used for programming.

Actions of ▽ and ↓

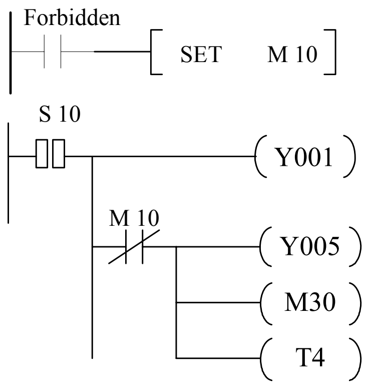


When state resetting is represented in the flow, it shall be represented with▽.

The symbol ↓ indicates transfer to upward state (repeat) or downward state (skipping), or transferred to state on other separated flow.



Clearing of state and output forbidden

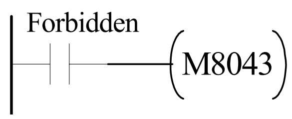


It is equal to forbidding output in emergency step stop. Please comply with Safety Attentions contained in the manual.

Clearing of state range

Forbid step random output in actions.

All the output relays (Y) PLC are off.



Special relays

To program SFC effectively, varieties of special relays shall be employed with content described below:

Component No.	Description	Function and purpose
M8000	Operation monitoring (contact a)	During operation of PLC, the relay needs always connecting, which can be used as input state for driving programming of operation state indication of PLC.
M8002	Starting pulse (contact b)	The relay is active in a short time (one scanning cycle) from stop to start of PLC. It is used for resetting of starting setting or initial state of the program.
M8040	Transfer forbidden	Driving the relay forbids transfer among state. Under the conditions of forbidding transferring, internal programs will still execute, therefore, the output coil will not be off automatically.
M8046	STL state action	When any state is active, M8046 is automatically active for avoiding simultaneous starting or for action sign of working step with other flows.
M8041	Step starts	Flag-sign for the instruction IST
M8047	STL monitoring is effective	Driving the relay, the programming function can read the executing state and display

Holding in case of power failure

Holding in case of power is used to keep its action with battery. During mechanical action in case of power failure, when power gets connected, such action can be continued.

Function of instruction RET

The instruction RET is finally programmed after a series of instruction STL.

Executing the instruction means completion of the step ladder-shaper loop. When expecting to interrupt a series of working steps and programming the master program, the instruction RET is also required.

The instruction RET can be programmed repeatedly.

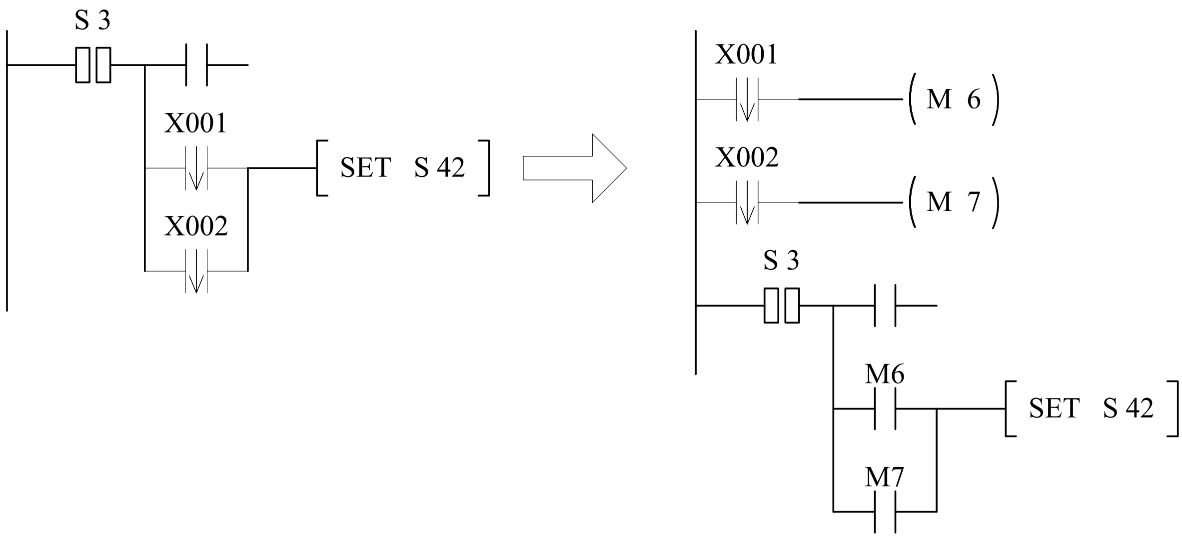
At the end of the instruction STL, if the instruction RET is not programmed, the software can add the instruction RET to the end of the program.

Attentions:

Attentions for detecting with rising edge and falling edge:

When rising edge and falling edge of LDP, LDF, ANDP, ANF, ORP and ORF, the changed contacts when the state is cut off will be detected when the state is active a second time.

As for changed conditions in case of the state is cut off, if rising edge and falling edge are used to detect, please modify the procedure according to the chart below.



If it is transferred to S42 through X001 falling edge, if X002 is falling, S3 is cut off, and X002 falling edge can not be detected. When S3 is active a second time, it is detected. Therefore, in case of second action of S3, it is transferred to S42.

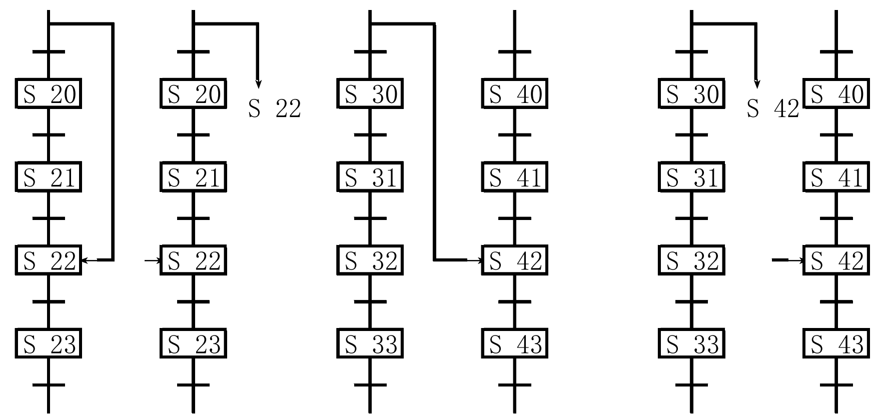
5 Form of SFC flow

5.1 Skipping and repeating flows

It indicates combined action mode of SFC single flow, selection branching and parallel branching.

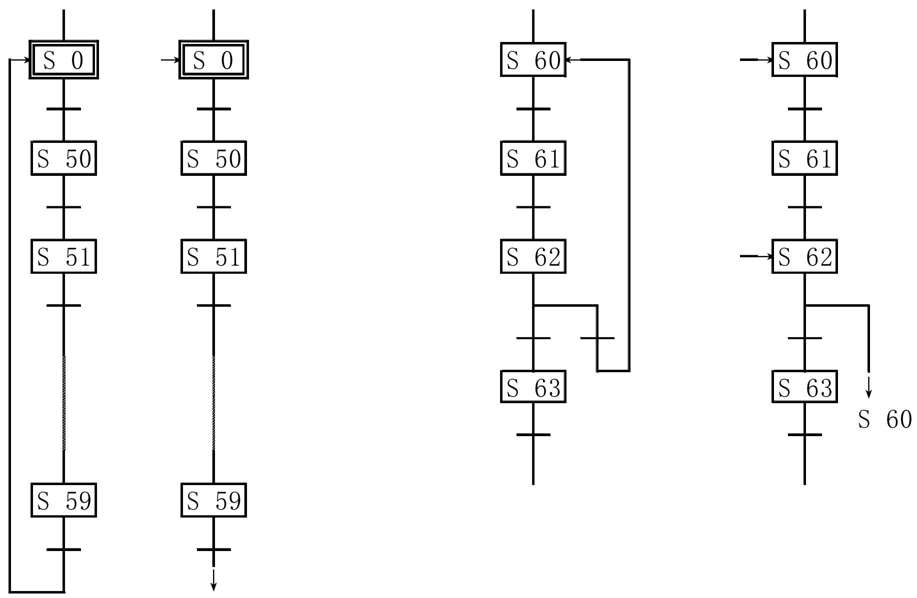
Skipping

Tranfer to the state below or out of the series is called skippng, with symbol ↓ to indicate transferred goal state.

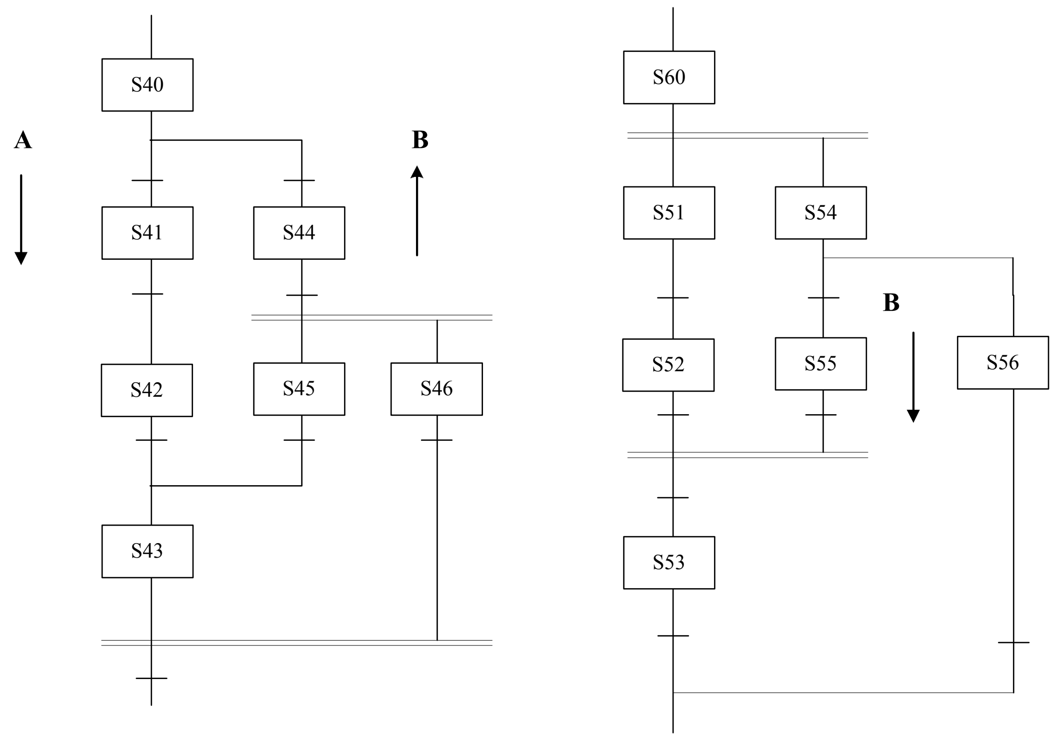


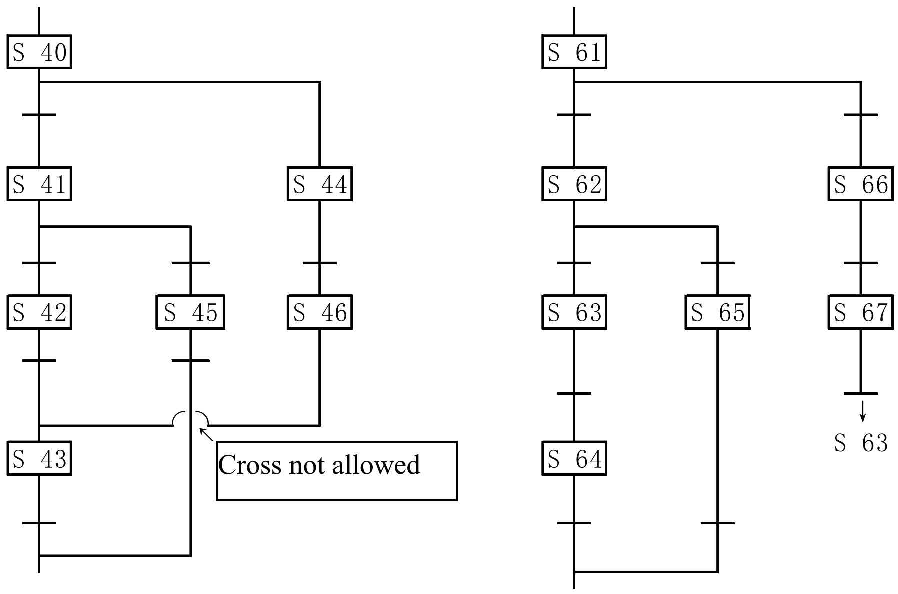
Repeating

Transfer to the above state is called repeating. The same as above, the symbol ↓ is used to indicate transferred goal state.



5.2 Combined flow of branching and merging





The SFC chart can not be used for flow cross. Flow in the above chart shall be re-programmed according to the program at the right side to realize reversal switching from program based on instruction to SFC chart.

6 Function of initial state

Use of initial state

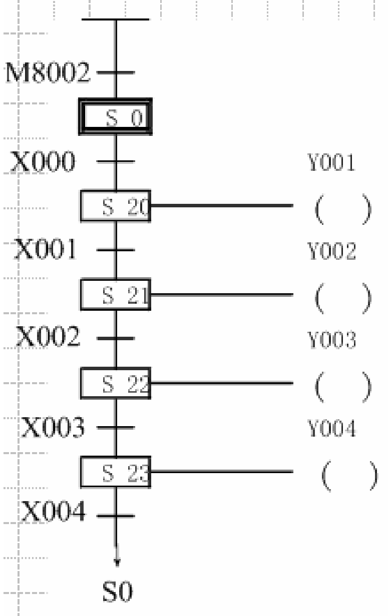
●The initial state is located at the frontmost position of SFC chart, which is represented in S0-S9.

●If the initial state is driven by other state, other method shall be used for driving when the operation starts.

Special auxiliary relay M8002 is used to drive during switching of stop-operation of PLC in the following cases.

●General state other than starting state shall be driven by STL of other state, and drive out of the drive shall not apply.

●The state driven by instructions other than STL is called initial state, which must be described in the frontmost of the flow. In addition, for the instruction STL in the initial state, it must be programmed before a series of instruction STL.




```
0   LD   M8002
1   SET  S   0
2   STL  S   0
3   LD   X   0
4   SET  S   20
5   STL  S   20
6   OUT  Y   001
7   LD   X   001
8   SET  S   21
9   STL  S   21
10  OUT  Y   002
11  LD   X   002
12  SET  S   22
13  STL  S   22
14  OUT  Y   003
15  LD   X   003
16  SET  S   23
17  STL  S   23
18  OUT  Y   004
19  LD   X   004
20  OUT  S   0
21  RET
22  END
```

With the instruction SET

Initial drive of initial state

Programming step S 0

Programming step S 20

Programming step S 21

Programming step S 22

Programming step S 23

Function of initial state

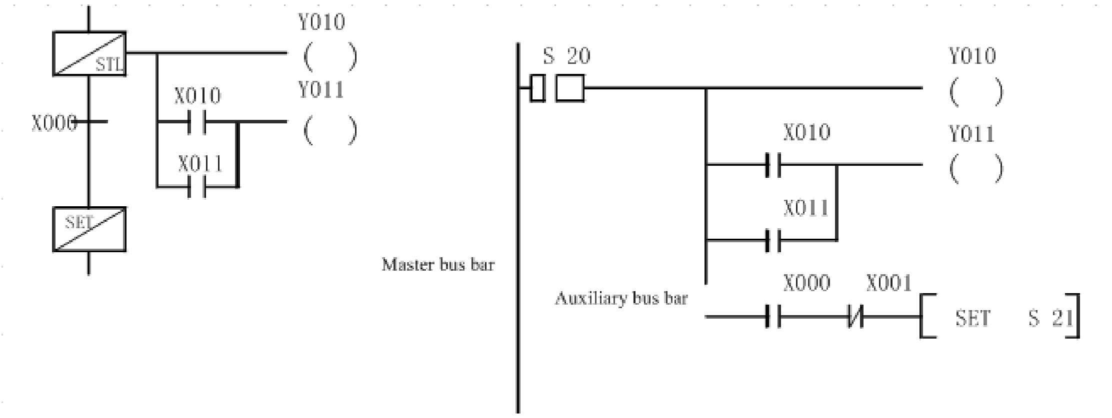
As the idenfied device for reversal change

● When reversal change is executed from instruction sheet to SFC chart, starting section of the flow shall be identified. Therefore, S0-S9 can be taken as initial state. If other numbers are used, reversal change can not be performed.

7 Intermediate state program

7.1 General flow without branching and merging

The following chart is a representative state from SFC. Each state has the functions of driving load, specifying transferring goal and transferring conditions. When sequential control of relay is used to represent SFC, it is the step ladder style programming chart below. It is programmed with SFC chart or step ladder style programming chart. The programming sequence is driving processing of load first, then transferring processing can be performed. Of course, there is driving processing without driving load.



0	STL	S 20
1	OUT	Y 010
2	LD	X 010
3	OR	X 011
4	OUT	Y 011
5	LD	X 000
6	ANI	X 001
7	SET	S 21

As for instructions SET and RST, it is a 2-step instruction

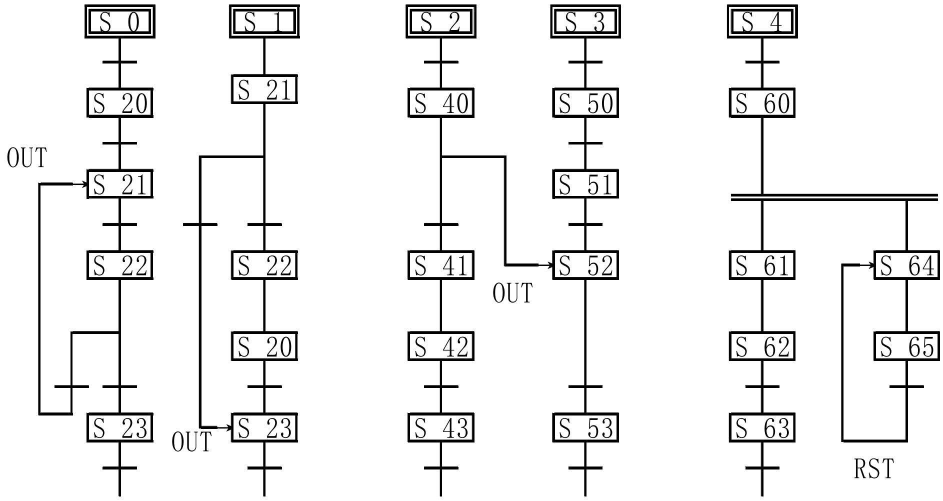
The instruction is used to represent program in the above chart, see the left chart.

The instruction STL is the normally open contact connected to main bus bar, in the following, coil is directly connected to auxiliary bus bar or the contact is used to drive the coil.

Contact connected to the auxiliary bus bar uses the instsruction LD(LDI). If it returns to original bus bar, the instruction RET shall be used. The contact STL drives the state S, it will be automatically reset before moving of the state S.

As for continuous SFC chart, if several state program shall be executed and all the statees are programmed, it will end after completing programming and the state sequential number can be selected at your discretion. However, initial state is required before a series of STL instructions, and the instruction RET shall be written finally.

7.2 General states with skipping and repeating

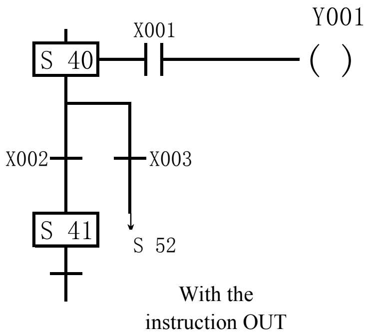


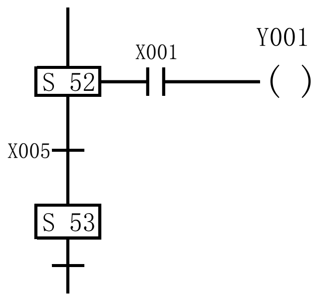
Repeating Skipping Jump out of the flow Reset processing

See the chart above, transferring to upward position (repeating), transferring to downward position (skipping) and transferring out of the flow and other separation state. See 4-5. The symbol ê represents the state symbol of transferred objective. See the chart below. The instruction OUT is programmed and cross flow in 4-5-1 is also the same.

Tranfering program

STL	S 40
LD	X001
OUT	Y001
LD	X002
SET	S 41
LD	X003
OUT	S52





Tranfering program

STL	S 52
LD	X004
OUT	Y002
LD	X005
SET	S 53

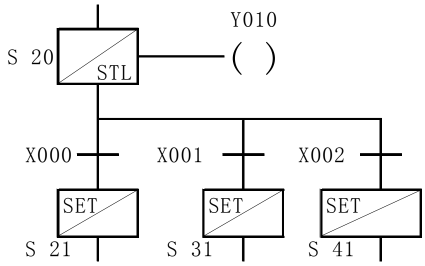
From S40, S52 is driven by X003, even the instruction OUT is used, when S52 holds its action, the transfering source S40 will be resetted automatically. The left chart indicates resetting of S65 throught X007 from S65.

From S65, although resetting of other state is the same, it it not transferring. S65 will not be resetted.

8 Branching and merging state program

8.1 Selective branching and merging state

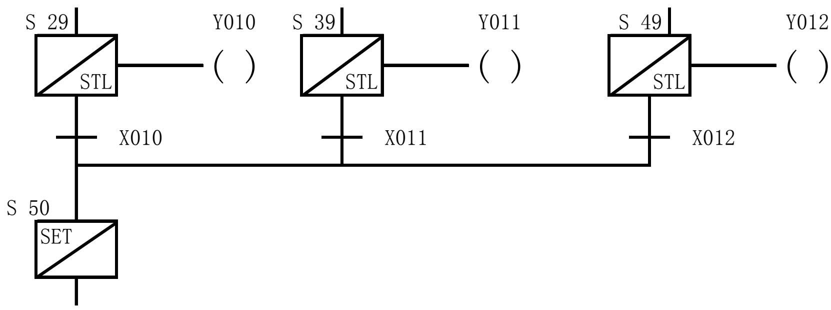
Case of selecting branching




```
STL   S 20
OUT   Y000   —Drive processing
LD    X000
SET   S 21   —Directly transferring to the state b
LD    X001
SET   S 31   —Transferred to 1st branching state
LD    X002
SET   S 41   —Tranferred to 2nd branching state
```

The same as programming of general state, driving processing shall be executed, then transfering processing shall be executed. All the transfering processing shall be executed according to the sequence.

Case of selecting merging




```
STL  S 29  —— Drive processing
OUT  Y010

STL  S 39  —— Drive processing
OUT  Y011

STL  S 49  —— Drive processing
OUT  Y012
```

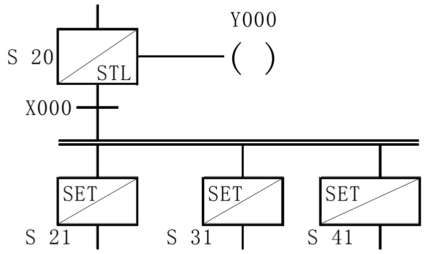
```
STL   S 29  ┐
LD    X010  │ Transferred to 1st merging point
SET   S50   ┘
STL   S 39  ┐
LD    X011  │ Transferred to 2nd merging point
SET   S 50  ┘
STL   S 49  ┐
LD    X012  │ Transferred to the next state
SET   S 50  ┘
```

State drive processing before merging shall be executed first, then continue transfering processing of merging state according to the sequence. It becomes the rules of reversal inversion towards SFC.

Pay attention to sequential numbers of the program, and the branching row and merging row cna not be crossed.

8.2 State of parallel branching and merging

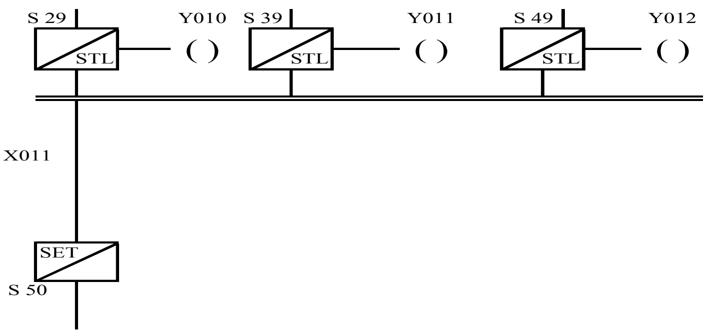
Case of parallel branching




```
STL   S 20
OUT   Y000    —Drive processing
LD    X000
SET   S 21    —Directly transferring to the state
SET   S 31
SET   S 41    —Transferred to 1st branching stat

              —Tranferred to 2nd branching stat
```

The same as program for general state, driving processing shall executed first, then transfering can be continued. All the transfer processing shall be executed according to the sequence.

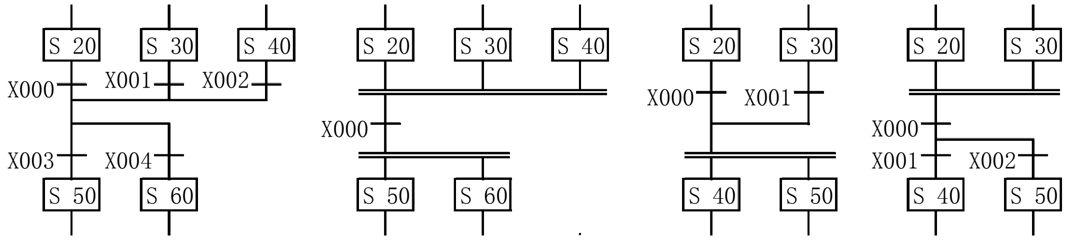
Case of parallel merging



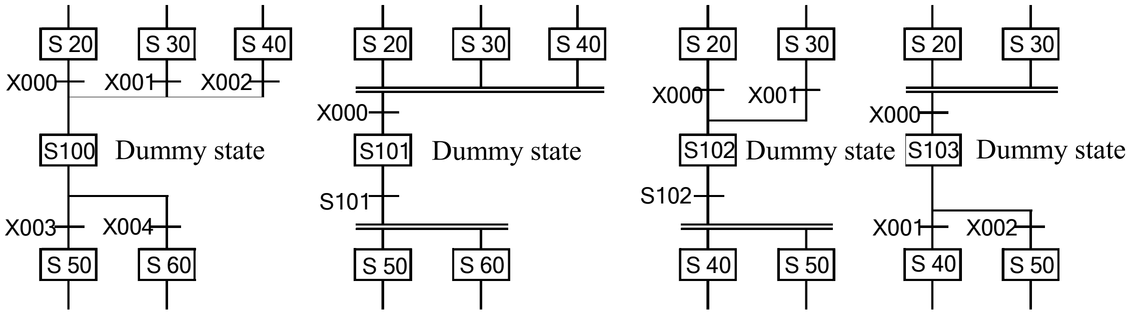

```
(
)
STL   S 29
OUT   Y010   —Drive processing
(
)
STL   S 39
OUT   Y011   —Drive processing
(
)
STL   S 49
OUT   Y012   —Drive processing
(
)
STL   S 29
STL   S39
STL   S49
LD    X010
AND   X011
AND   X012
SET   S 50
(
)
```

The continuous instruction STL represents parallel merging. Parallel branching is restricted less than 8.

8.3 Combination of branching and merging



Changed to the following forms:




```
STL   S20
LD    X000
SET   S100
STL   S 30
LD    X001
SET   S100
STL   S 40
LD    X002
SET   S100
STL   S100

LD    X003
SET   S 50

LD    X004
SET   S60
```

```
STL   S 20
STL   S 30
STL   S 40
LD    X000
SET   S101
STL   S101
LD    S101
SET   S 50
SET   S 60
```

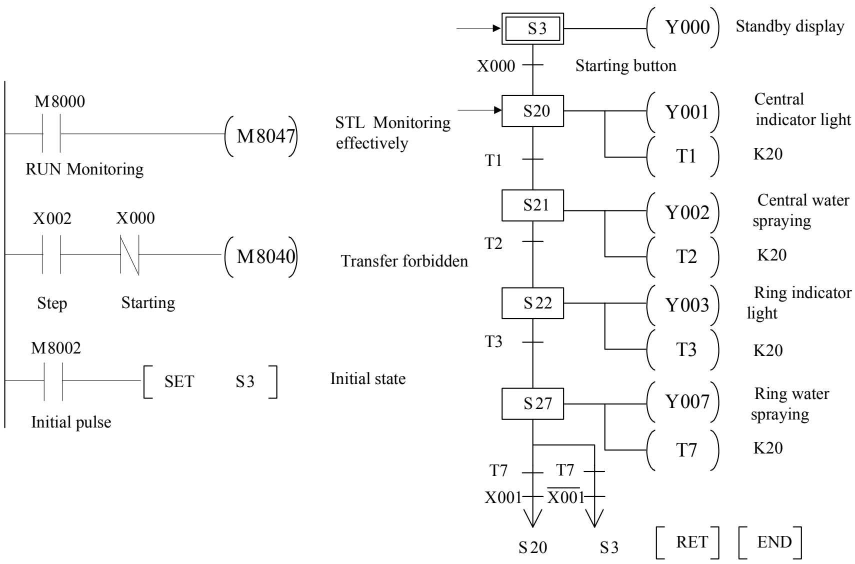
```
STL   S 20
LD    X000
SET   S102
STL   S 30
LD    X001
SET   S102
STL   S102
LD    S102
SET   S 40
SET   S 50
```

```
STL   S 20
STL   S 30
LD    X000
SET   S103
STL   S103

LD    X001
SET   S 40

LD    X002
SET   S50
```

Direct branch after merging is not a good way, and one virtual state point is easy to write.

9 Case of single flow

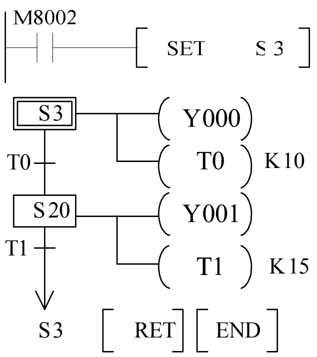
Case of water spraying control

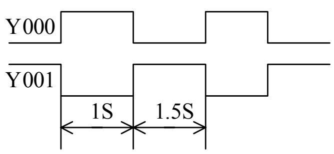


The above chart

1. Single operation (X001=OFF，X002=OFF). Press starting button, X000 acts, it acts as per the sequence according to Y000→Y001→Y002→Y003→Y007→Y000, and returns to standby state. Output is executed according to the sequence as per the 2-second timer.
2. Continuous operation (X001=ON), repeat actions Y001~Y007.
3. Step operation (X002=ON), press the starting button one time, output actions as per the sequence.

Case of sparkling loop



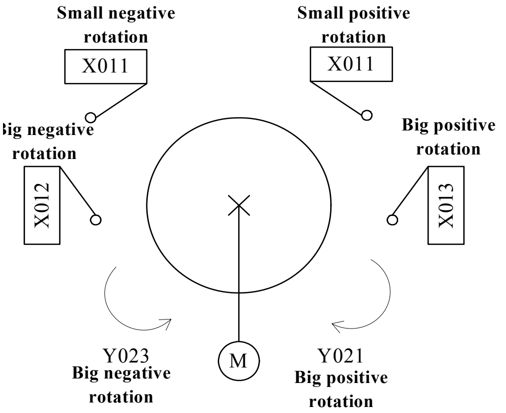


1 When PLC is operated, initial pulse（M8002） drives state S3.

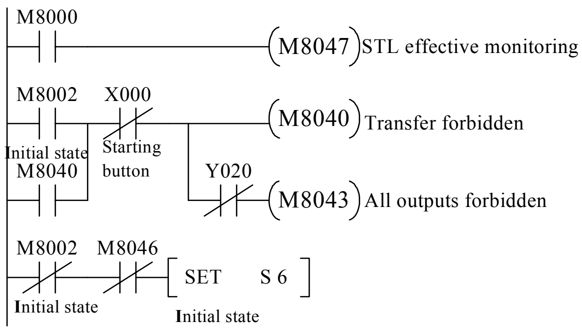
2 After the state S3 is ON, it outputs Y00, meanwhile, the timer starts timing. 1 second later, the counting will end and transfer to state S20.

3 When state S20 is changed to ON, it outputs Y001, the timer starts timing, 1.5 seconds later, it returns to S3.

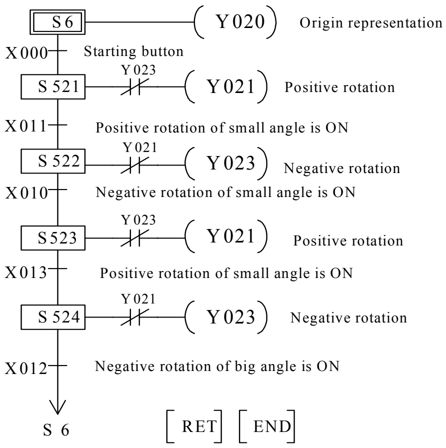
Rotation control of cam shaft



Limit switches X013 and X011 are set in the positions of the positive rotating angle, and limit switches X012 and X010 are set in the positions of the negative rotating angle.
Press the starting button, execute positive rotating small→ negative rotation small → negative rotation big→ negative rotation big and etc., then it stops.



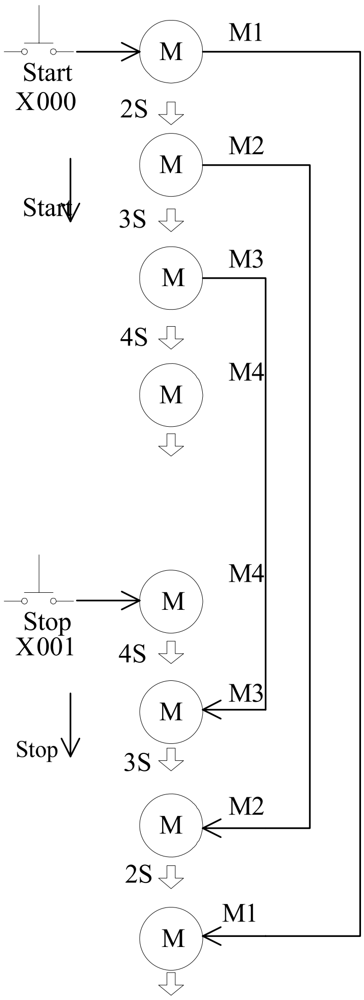
The limit switches X010~X013 are normally OFF. When the cam reaches the setting angle, it is ON.

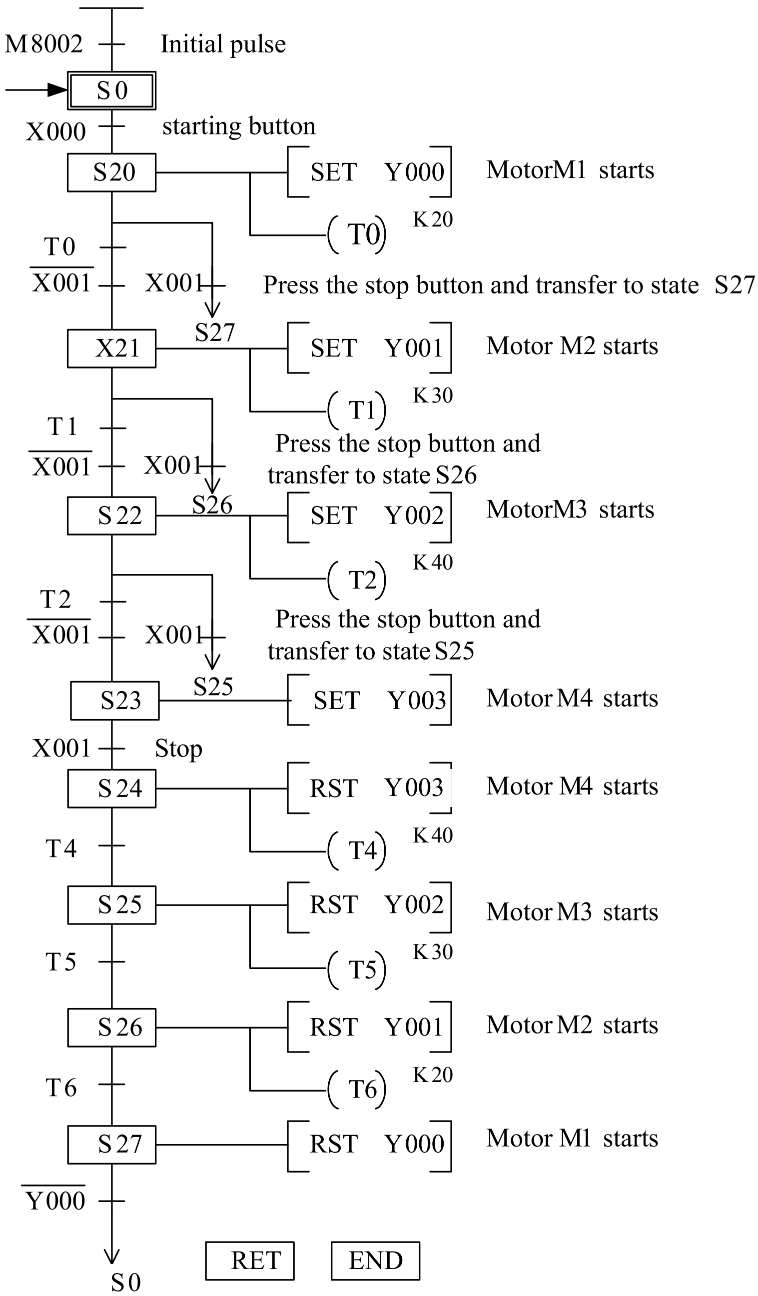


- When M8047 acts, the action state monitoring is effective. After S0-S899 act and the instruction END is executed, M8046 acts.
- State point of SFC is held by the battery. In case of power failure during action, when pressing the starting button a second time, it will continue acting. Before pressing the starting button, output actions below Y20 are totally forbidden.

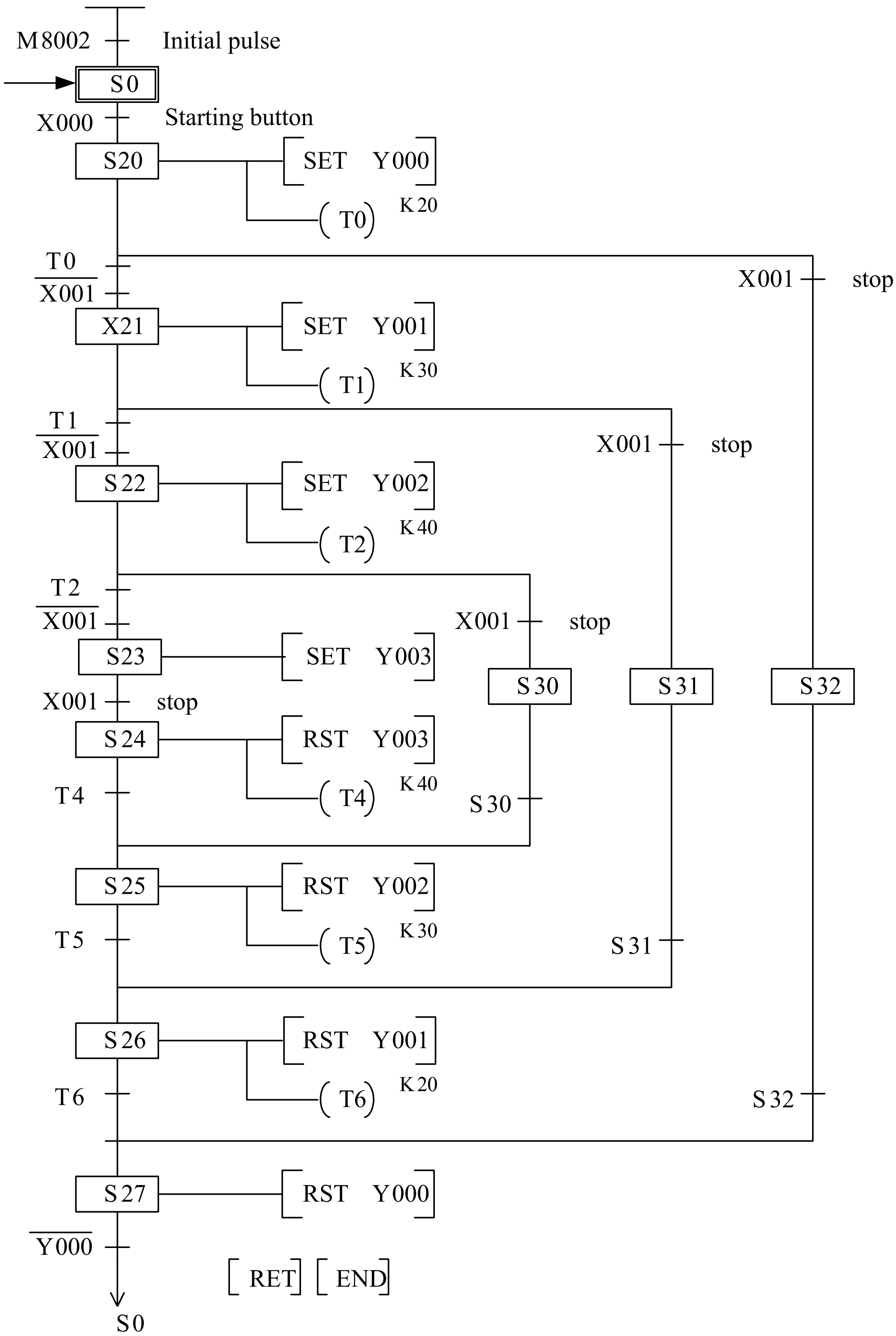
Case of sequential start and stop

The motor is started from M1 to M4 by the timer and stop with the reverse sequence.
Such SFC flow performs skipping based on single flow.





The skipping flow in the previous page can be represented in selective and merging flows described below. Flowing direction must be from up to down, which can not be crossed except branching and merging.

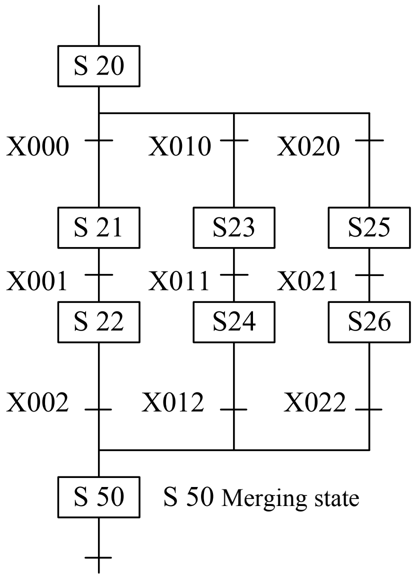


For instance, the state S20 acts, if X001 is active, the state S32 acts, then the contacts act, which skip to the state S27.

There should be more than one state in the branching; therefore, no op state shall be set.

10 Case of selective branching and merging

Action of selective branching

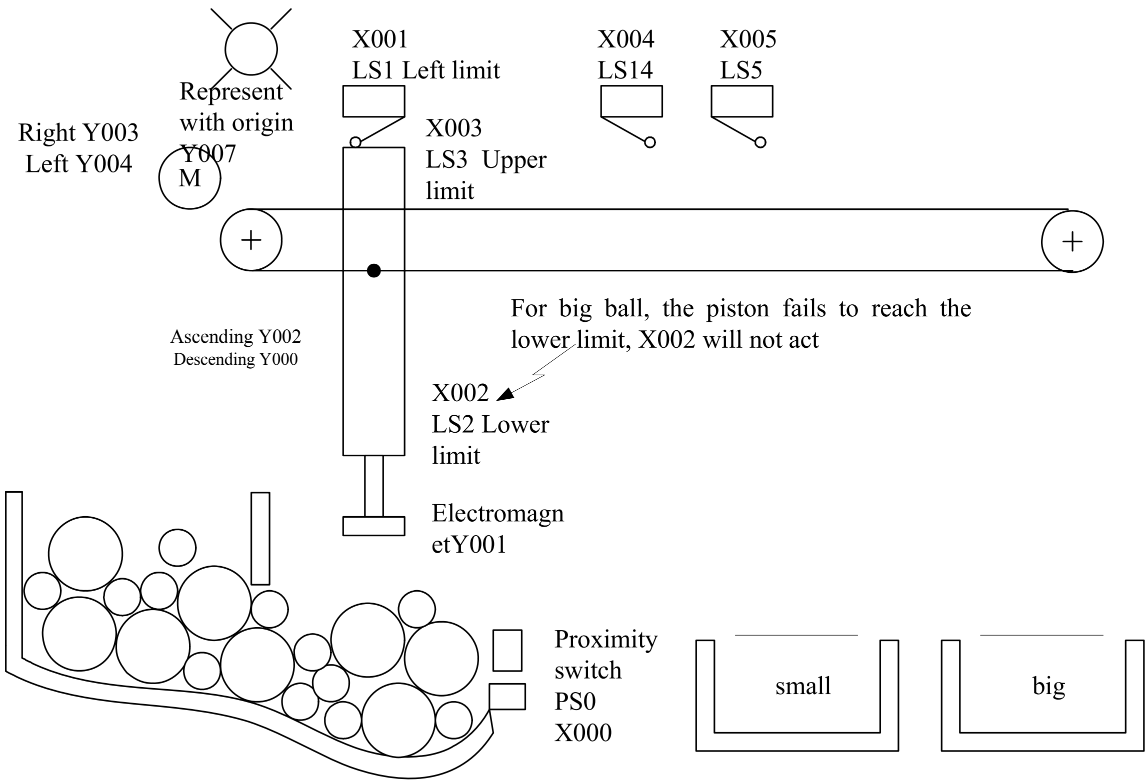


- One flow is executed from several flows is called selective branching.
- See the left chart, X000, X010 and X020 can not be active simultaneously.
- For instance, when S20 acts and X000 is active, the action state is transferred to S21, S20 will not act.
  Therefore, even X010 and X020 act, S23 and S25 do not act.
- Merging state S50 can be driven by either S22, S24 or S26.

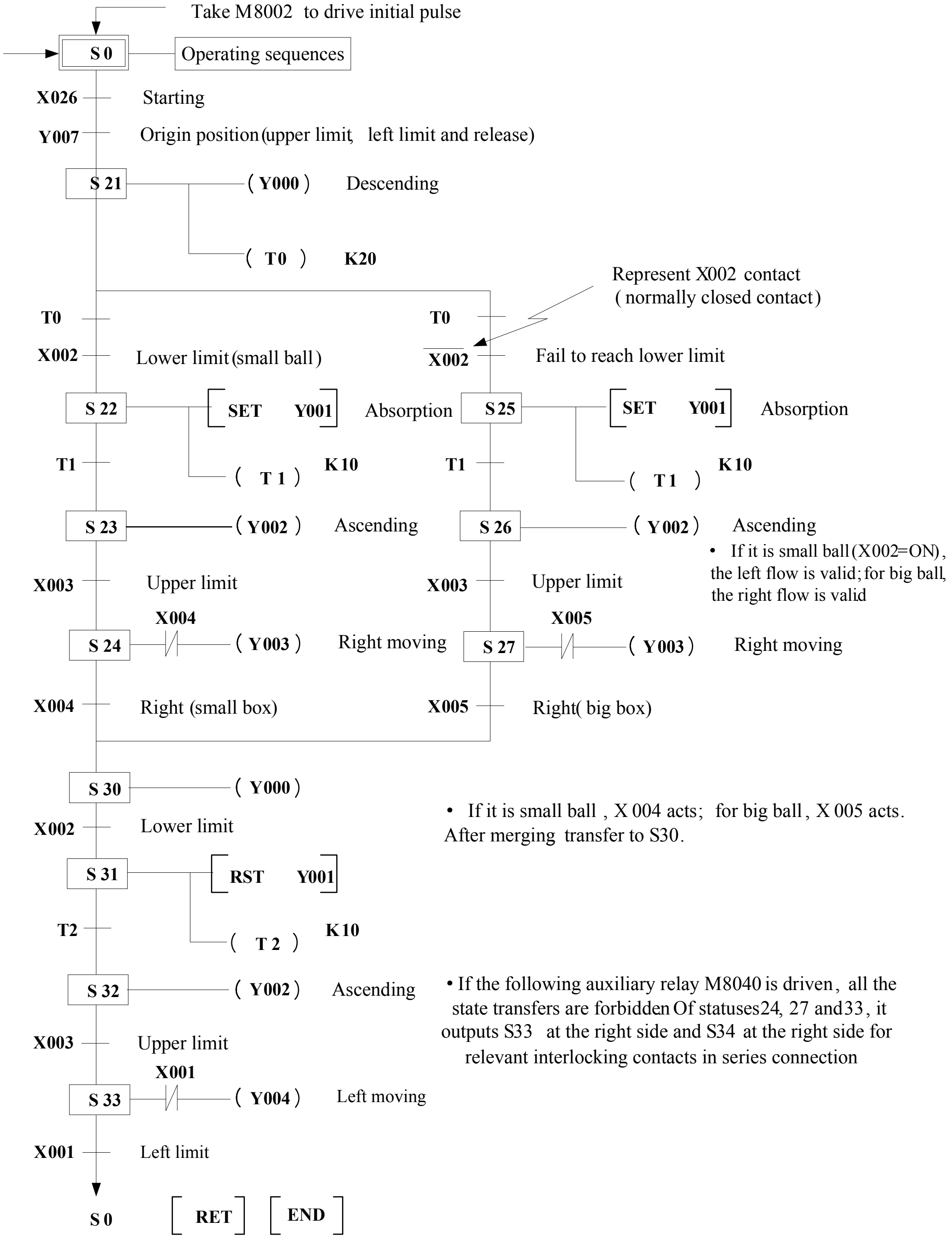
Action of selective branching

The following chart is the machine to convey classified big and small balls with the transmission points.

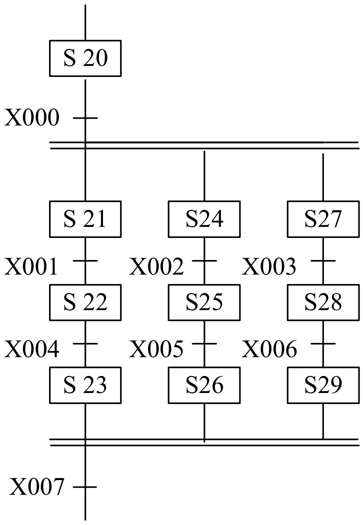
The top left is the origin and sequence of actions is descending, absorption, ascending, rightward moving, descending, ascending and leftward moving. In addition, the mechanical arm descends, when the electromagnet presses the big ball, the lower limit switch LS2 is inactive; when pressing the small ball, LS2 is active.



For such SFC for selecting size or judging acceptance or not, the following branching and merging SFC can be used to represent.



11 Case of parallel branching and merging flow



- Branches with several flows executed simultaneously are called parallel branching.
- Take the above chart for example, after S20 acts, X0 is on, state S21, S24 and S24 will be effective simultaneously and the flows start.
- After all the flows are completed, when X07 is active, merging state S30 acts and state S23, S26 and S29 do not act.
- Such merging is called waiting merging. (Flows complete earlier shall wait until all the flows are completed, then merging is continued).

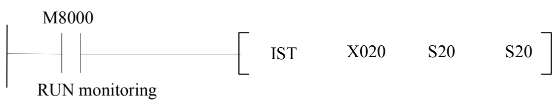
For instance, parts A, B and C are processed separately, which are assembled after processing, and this is a parallel branching and merging flow.

12 Fexible use of initial state (F60 IST) instruction

See the following chart on mechanical reversal mode, and part or whole of the mode can be used.

- Manual mod
  - Operation: independent button to execute the loads
  - Return of origin: press the reset button to return to origin.
- Automatic mode
  - Step: press the button one time, it moves forward.
  - Cycle mode: press the starting switch at the position of orgigin, automatic operation of one cycle is realized, then it stops at the origin. Press the stop button during operation, it stop. Press the starting button again, it continues from where it stops then it stops at the origin.
  - Continuous operation: press the starting button at the origin, it starts repeative operation. After pressing the switch button, it will stop at the origin.

In general, such control can be realized by writing step ladder style programming chart (SFC flow). In applied instruction of PLC of TP03 series, such method can control the mechanical and fixed instructions.



Applied instruction F60 (IST) is a complete set of instructions for state in the above mode or automatic control of special auxiliary relay.

IF the instruction IST is used, switching between modes and repeated control program sjhall be used. We should only focus on writing program of mechanical action in the state to complete the sequence design.
Refer to instruction F60(IST) in *Applied Instrucitons*.

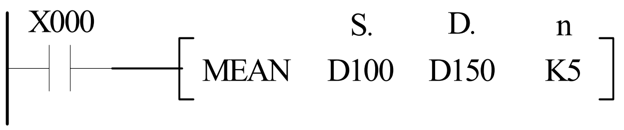
Chapter V Representation and Use of Applied Instructions

The chapter introduces PLC applied instructions and programming method of TP03 series. In general, one basic instruction can finish one specified operation and one applied instruction can finish one series of operations, which is equal to one sub-program, therefore, function of applied instruction is powerful. The basic instructions and ladder symbols are corresponding. The applied instruction employs ladder symbols for memory view to represent what the instruction is to do. Times of applied instruction in the whole program are restricted.

1 Representation and executing form of applied instructions

Instructions and operands

- PLC applied instruction can specify function No. F00-F □□□, and the instructions are represented with memory view. For instance, F45 is MEAN, representing “average value”.
- Applied instructions are made up of function No. and following memory view to form one complete instruction.



- MEAN: Memory view of the instruction, representing average value in mathematical way.

S.: Source operands, called source for short, after executing the instructions, operand of the content will not be changed. Under the conditions of changing device number with index, add “·” [S·] to represent and when the operand is not one, represent with [S1·], [S2·] and etc.

D.: Destination operand, called destination for short, after executing the instructions, operand of the content is changed. Like the source, index decoration shall apply; when the destination operand is more, represent with [D1·], [D2·] and etc.

m , n: other operands, which are used to represent constant or make additional interpretation for the source and destination. Decimal system figure is followed after K and hexadecimal system figure is followed and H.

Program step: the step for executing the program. Generally speaking, the function number and memory view occupy one step and each operand occupies 2-4 steps (2 steps for 16-bit operand and 4 steps for 32-bit operand).

Available device for operands

- X, Y, M, S and other bit devices can be used.
- Combine these devices, represent with KnX , KnY , KnM , KnS and other forms as data for processing. Refer to following *Use of Bit Device*.
- Current value registers of processing data buffer D, timers T or counter C.

The data register D is 16-bit, when 32-bit data is processed, one couple of data registers can be used.

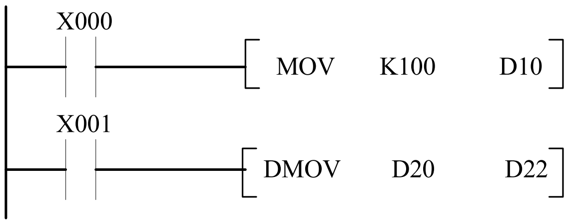
For instance, when the data register D0 is specified as operand of 32-bit instruction, 32-bit data (D0 and D1) is processed (D1 is high 16-bit and D0 is low 16-bit). Current registers of T and C can be used as general registers; however, point 1 of 32-bit counter of C200~C255 can process 32-bit data, which can not be used for operand of 16-bit instructions.

Form and executing form of the instructions

Of PLC of TP03 series, according to size of data to be processed, the applied instructions are divided into 16-bit instruction and 32-bit instruction. In addition, according to executing forms of the instructions, it has features of continuous executing and pulse executing and etc.

The applied instruction can be used together or independently.

- In applied instruction of data processing, according to bit length of the data, it is divided into 16-bit and 32-bit.



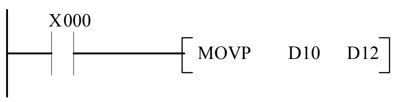
The decimal figure 100 is sent to hexadecimal system data register D10

Content of （D21, D20) is sent to（D23, D22）

- As for function instruction represents prefix D, it can process 32-bit data.
- 32-bit is comprised of 2 adjacent registers.
- One device of the 32-bit counter (C200-C255) is 32-bit, and it can not be used as 16-bit operand.

Pulse executing/continuous executing instructions

Pulse executing type



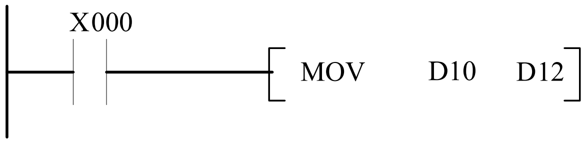
Under pulse executing, the instruction MOV execute on data sending from OFF→ON under the condition X000. To shorten scan time, pulse executing instruction must be used as much as possible.

The symbol P represents the instruction is available for pulse executing.

F24 (INC), F25 (DEC) and etc. shall be in accordance with the instructions. If continuous executing instruction is employed, each scan cycle and source content will change.

Note: pulse executing instruction does not execute in the first scan cycle.

Continuous executing type



The above figure is continuous executing instruction, PLC works in circulating scan. If the executing condition X001 is active, the above instruction is repeated one time in each scan cycle.

- When the drive inputs X0 and X1 are inactive, the instructions out of special symbols are not executed and the destinations do not change.

Processing of symbols

General symbols

- According to types of applied instructions, the following symbols will act.

(For example) M8020: zero symbol M8022: carrying symbol

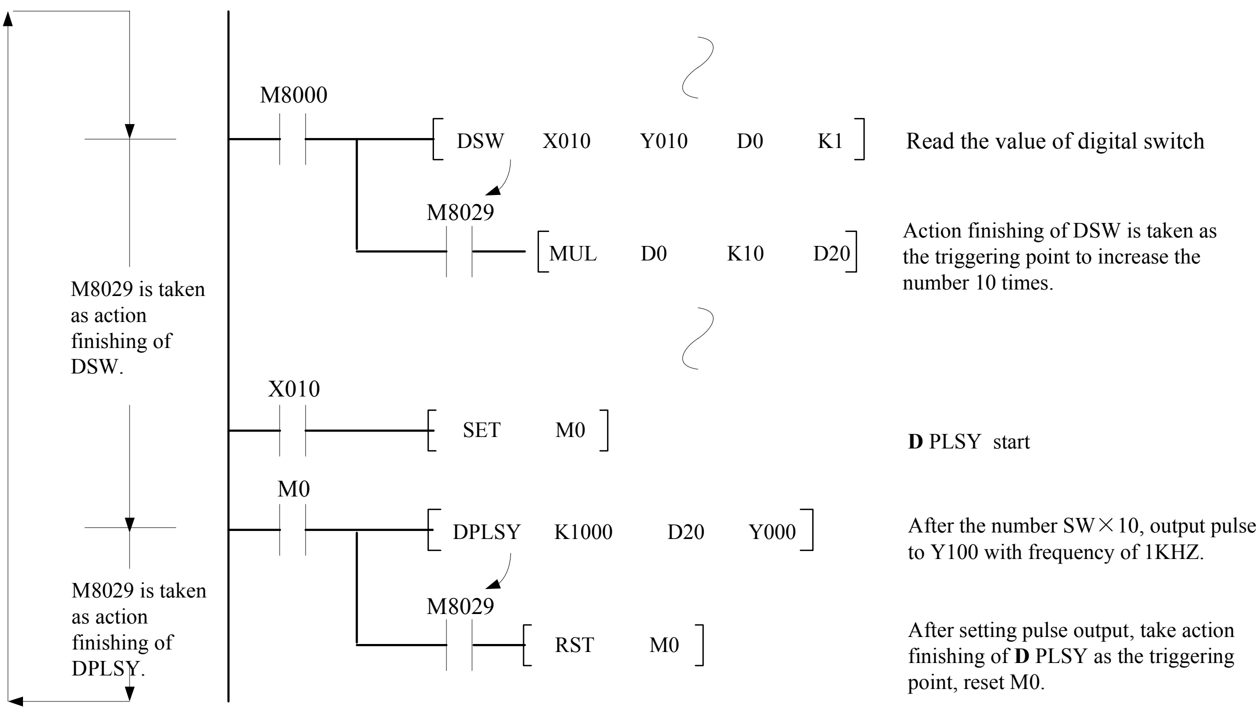
M8021: borrowing symbol M8029: executing results

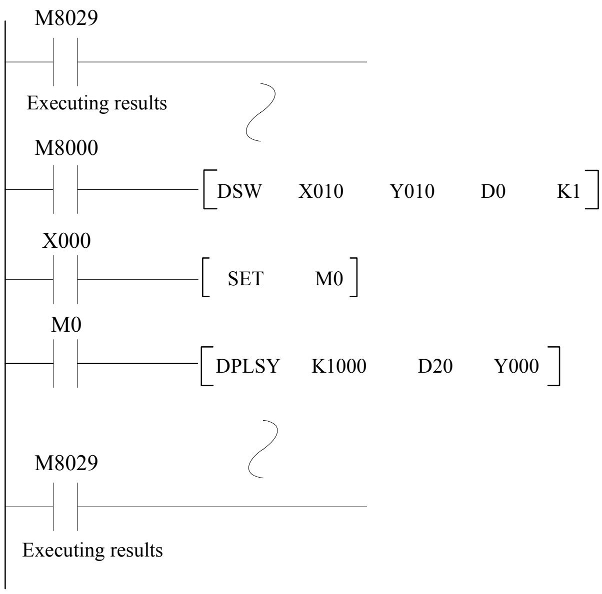
When the instructions are ON, the symbols will be active or inactive; for OFF, it will not change in case of error.

When there are too many instructions affecting the symbols, when executing the instructions each time, the active/inactive state will be changed. Please refer to the following symbol program case.

- Program case of several symbols (standard case of executing results)

When there are several applied instructions for the same symbol action, please write symbol contact-point after the instructions.





As for applied instructions employing the finishing symbol mark,for instance, DSW and DPLSY shall employ M9029 as the finishing mark. See the left chart on programming, and we can not judge which instruction is finished.

Symbol of computation errors

In case of errors in structure of applied instructions, available devices, number range and etc., it may lead to error in computation and the following symbol bit will act and record the error information.

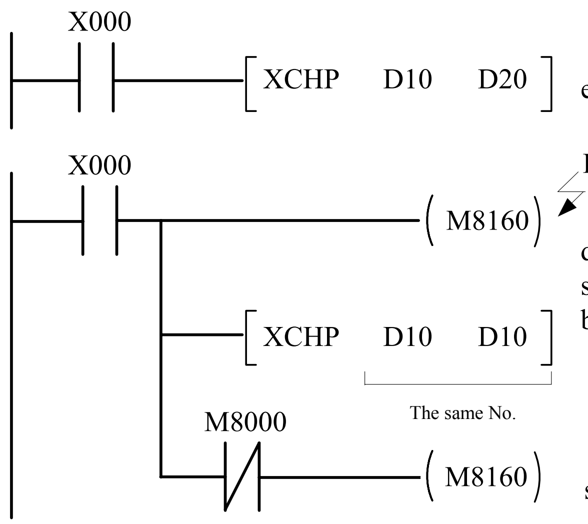
M8067 D8067

In case of abnormal computation, M8067 will act and hold and store code of abnormality in D8067.

In case of other new abnormality, D8067 will update new code of abnormality and abnormal step number (when the abnormality is released, it is OFF). When PLC from STOP→RUN, it is released.

Symbol for extended functions

In part of the instruction, the inherent special auxiliary relay determined by the applied instruction can be used for function extension and the following example shall apply:



• When the instruction X000 is active, exchange content of D10 and D11.

• When M8160 before the instruction XCH is driven, the computation object and result specify the same element, exchange the high 8-bit and low 8-bit.

• To return normal instruction XCH M8160 shall be inactive.

In addition, when the instruction for interrupting program needs the symbol for function extension, before drive of the symbol for function extension, write the DI instruction (interruption inhibition) and write the instruction EI (interruption permission) after inactiveness of the symbol for function extension.

Restriction of simultaneous drive of the instructions

In applied instruction, even some instructions can be programmed several times, and there are restrictions on action points.

Less than 6 instructions

F53 (DHSCS)，F54 (DHSCR)，F55 (DHSZ)

Less than 2 instructions

F72(DSW)，F74(SEGL)

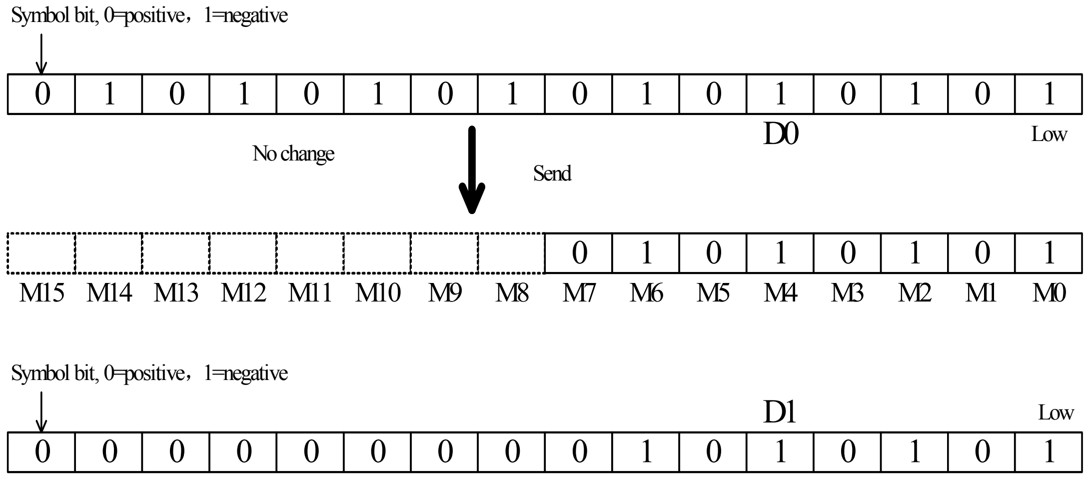
Less than 1 instruction

F52(MTR)，F57 (PLSY)，F58(PWM)，F59 (PLSR)，F60(IST)，F62(ABSD)，F68(ROTC)，F69(SORT)，F70(TKY)，F71(HKY)，F75(ARWS)，F80 (RS)，F87 (MBUS)，F156 (ZRN)，F157 (PLSV)，F158 (DRVI)，F159 (DRVA)，F190(DTLK)，F191(RMIO) ，F193(DTLK2)。

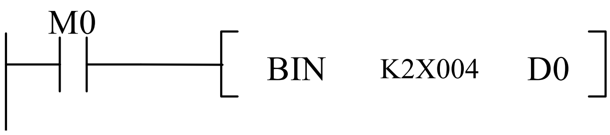
2 Use of data in the applied instructions

Use of bit element

- Like X, Y, M and S, there are two states, namely ON or OFF. The element which is represented with binary system is called bit element, and T, C, D and other devices for data processing are called word element. Even it is a bit element, it can be used to process data by combined use of the bit element. Under the circumstances, it represents with bit Kn and start device number.
- 4-bit is used as the unit, and number of bit is K1-K4(16-bit data)，K1-K8(32-bit data)，For instance, K2M0， M0~M7，2-bit data.



- When 16-bit data is sent to specified of K1M0~K3M0, the high-bit data (namely 4 bit in maximum) will not be sent according to insufficient specified data length; the same for 32-bit.
- In 16-bit (or 32-bit) computation, when the bit assembly is for specified number of bit K1-K3 (K1-K7), if the high bit is not sufficient, add 0 for processing, and the highest bit is 0, so the data is processed as positive.



- The specified bit element number can be specified freely. It is suggested that for X and Y, the lowest bit number shall be set 0 (X000, X010, X020…Y000, Y010 and Y020); for M and S, multiple of 8 is ideal. To avoid confusing, it shall be set M0, M10, M20… and etc.

Attached note

<Specifying of continuous character >

A series of data registers starting from D1 are D1, D2, D3, D4 and etc.

Through bit specifying, in the word situation, it can be used for a series of word processing. See the followings.

K1X000 K1X004 K1X010 K1X014……, K2Y010 K2Y020 K2Y030

K3M0 K3M12 K3M24 K3M36……, K4S16 K4S32 K4S48

Namely, do not skip the device. Use the devices according to unit of the bit.

However, for 32-bit computation, if K4Y000 is used, the upper 16-bit is 0. When 32-bit data is required, please specify K8Y000.

Use of floating-point decimal computation

In PLC, integral of PLC employs binary system.

In division computation of integral, for instance, 40/3=13 and 1.

In evolution computation of integral, decimal point is ignored.

Of PLC of TP03 series, to perform the computation more precisely, floating-point number computation is used.

- The floating-point number computation is valid for the following instructions.

F49(FLT), F110(DECMP), F111(DEZCP), F118(DEBCD),

F119D(EBIN), F120(DEADD), F121(DESUB), F122(DEMUL),

F123D(EDIV), F127(DESQR), F129(INT)

<Decimal system floating-point number >

- Binary floating-point number is hard for user to judge, therefore, it shall be converted into decimal floating-point number.
- A couple of data registers with continuous numbers can be used to process decimal system floating-point number, the smaller number is the mantissa section and the bigger number is the index section.

For instance, when the data registers (D1, D0) are used, the data is written into D0 and D1 by the instruction MOV.

Decimal system floating-point value = 「Mantissa D0」 $X10^{[\text{index DI}]}$

Mantissa D0= (1,000~9,999) or 0

Index D1= -41 ~ +35

The highest bits of D0 and D1are the bit for positive and negative symbols, which are processed as a complement code for 2.

Besides, in Mantissa D0, for instance, 100 does not exist. In situation for 100, it becomes 1000×10^{-1} (Mantissa 1000, index-1). Processing range of decimal floating-point is stated below:

Minimum absolute value 1175×10^{-41} Maximum absolute value 3402×10^{35}

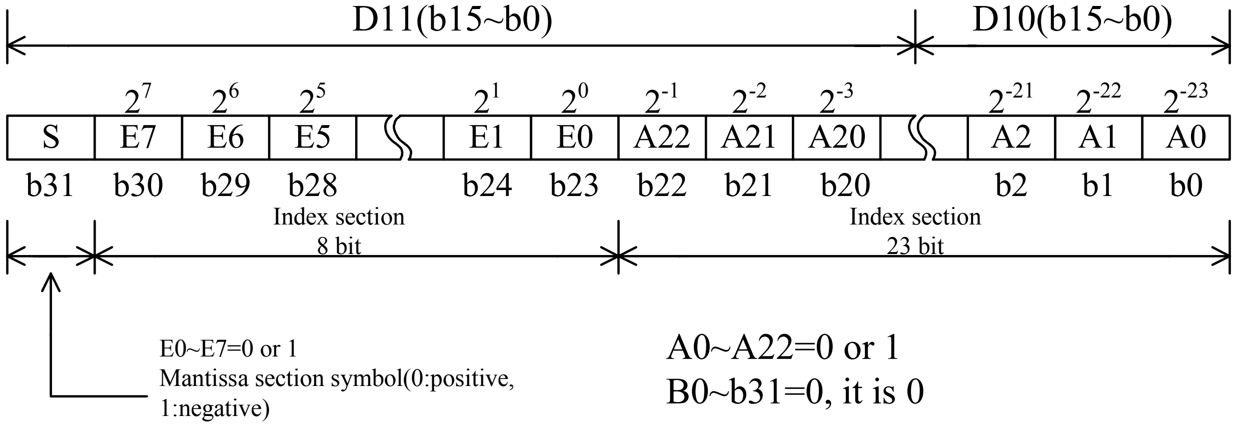
- Decimal floating-point number is valid in the following instructions.

Binary floating-point number→ Decimal floating-point number conversion F118 (DEBCD)

Decimal floating-point number→ Binary floating-point number conversion F119 (DEBIN)

Binary floating-point number

Binary floating-point number is a couple of data buffers using continuous numbers, such as situations (D11,D10), with results below:



Binary floating-point number $=\pm\ (2^{0}+A22\times 2^{-1}+A21\times 2^{-2}+\cdots\cdots+A0\times 2^{-23})$

$\times 2^{(E7\times 2^{7}+E6\times 2^{6}+\cdots\cdots+E0\times 2^{0})}\ /\ 2^{127}$

For example A22=1, A21=0, A20=1, A19~A0=0

E7=1, E6~E1=0, E0=1

Binary floating-point number $=\pm\ (2^{0}+1\times 2^{-1}+0\times 2^{-2}+1\times 2^{-3}+\cdots\cdots+0\times 2^{-23})$

$\times 2^{(1\times 2^{7}+0\times 2^{6}+\cdots\cdots+1\times 2^{0})}\ /\ 2^{127}$

$=\pm 1.625\times 2^{129}\ /\ 2^{127}=\pm 1.625\times 2^{2}$

The positive and negative symbols are determined by b31, and complement code can not be used.

- Use of zero symbol (M8020), borrowing symbol (M8021) and carrying symbol (M8022), see the symbol action on floating-point computation.

Zero symbol: when the result is 0, it is 1.

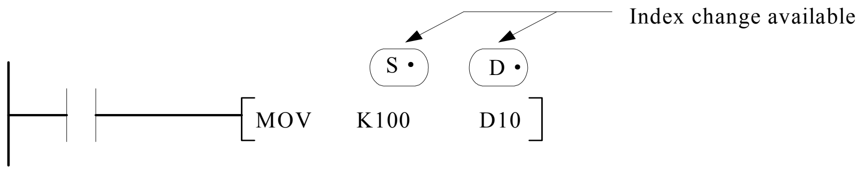
Negative symbol: the result is not the minimum and not 0, it is 1.

Carrying symbol: if the result exceeds available range of the absolute value, it is 1.

3 Change of operand with index register

Available applied instructions

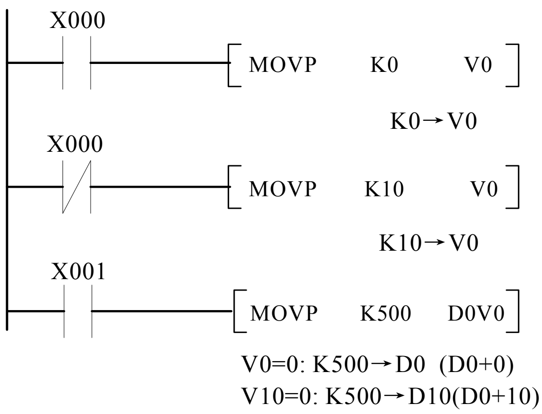
In interpretation of applied instructions, see the following figure on operand of index decoration. Add the mark "•" on the source S and destination D to distinguish the operand without change functions.



Case of index change

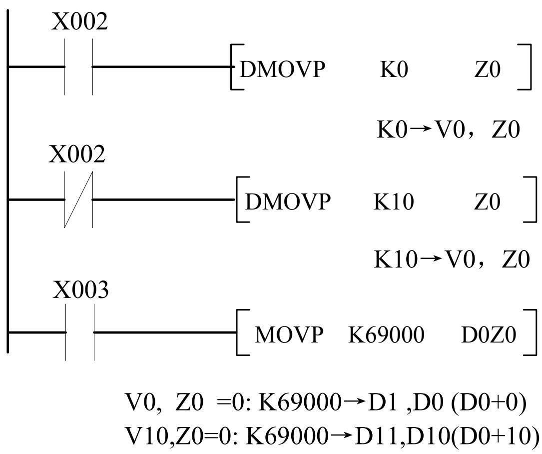
As for structure and function of the index register, please refer to [2-9-2 index register] mfor more information.

Number decoration of data register



Change of 16-bit instruction operand
Send content of K0 or K10 to the index register V0

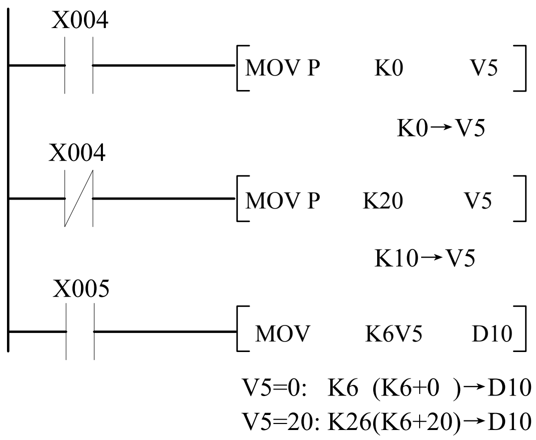
When X001 is active, and V0=0(D0+0=0), content of K500 is sent to D0. If V0=10(D0+10=D10), content of K500 is sent to D10.



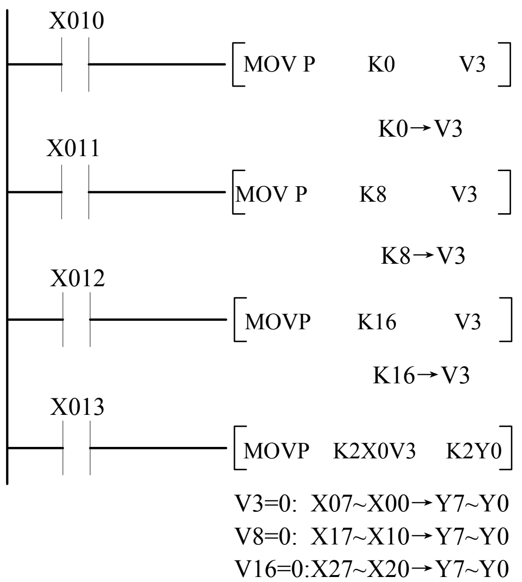
Change of 32-bit instruction operand
For the instruction DMOV is 32-bit instruction, therefore, the index register for the instruction shall be specified as 32-bit. In the 32-bit instruction, side Z (Z0-Z15) of the 32-bit instruction is specified, including side V (V0-V15), which is taken as actions of the 32-bit register.

*1 even data written in Z0 does not exceed 16 bits (0-32, 767), and V and Z shall be rewritten with the 32-bit instruction. If only size Z is written, and other data is left on side V, it may lead to great computation error.

Decoration of the constant K



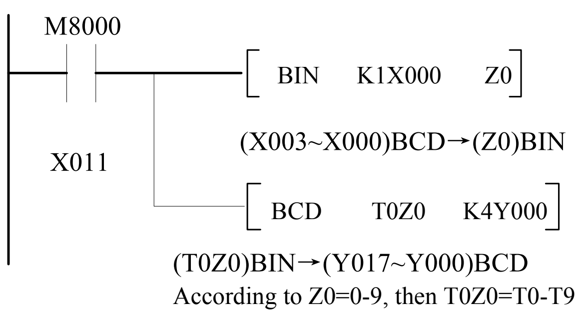
Change of the constant is the same as the device number.
If X005 is ON, if V5=0, [K6+0=K6], content of K6 is moved to D10.
If V5=20, [K6+20=K26], content of K26 is moved to D10.
Change of input and output relays (octal system device number)

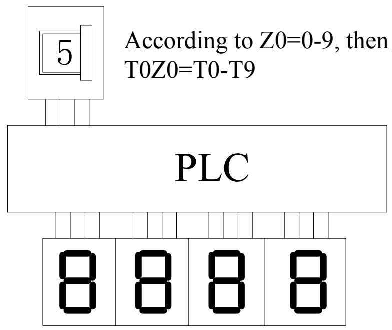


When X, Y, KnX and KnY and other octal system device number is executed for index change, content in the change register of corresponding device number shall be plus after octal system conversion.

See the left figure, output Y7-Y0 with the instruction MOV to change it into X7-X0, X17-X10 and X27-X20.

The conversion is by octal system conversion through [X0+0=X0], [X0+8=X10] and [X0+16=X20], with the device number, the input terminals are changed.
Display case of current value of the timer

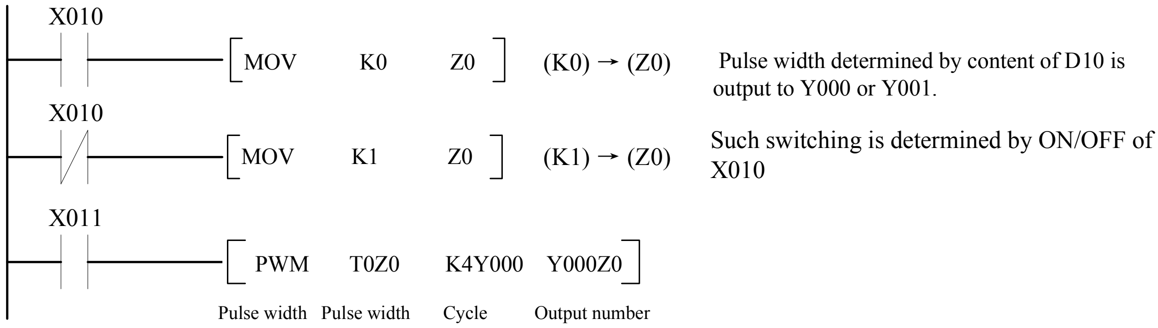




7-section code for display current value of the timer, output Y017-Y000.

Decoration of instructions with restricted times of use

The object assembly number is decorated with index buffers and the program can be used to change the object assembly number. As for instructions with restricted times of use, such method has the same effect with programming the same instruction for several times.



The instruction F58 can execute the programming instruction one time. Without driving several outputs at the same time, the controlled objects can be changed by changing the output numbers. In addition, during the instruction executing, even Z is changed, the above switching is invalid. To make better switching, please set condition of the drive instruction OFF one time.

Attentions:

- The 16-bit counter with index change can not be used as 32-bit counter. As results of index change, for 32-bit counter, please add Z0-Z15 after the counter C200.
- V and Z self or bit specifying employs Kn, and n can not be changed. (K4M0Z0 is valid and K0Z0M0 is invalid）
- Index change can not be executed for LD, AND, OUT and other PLC basic control instructions and step chart instructions.

4 Specification of Constants K, H and E

K: Decimal   H: Hexadecimal   E: Real Number

When handling constants in a sequence program, use constant K (decimal), H (hexadecimal) or E (floating point).

In peripheral equipment for programming, add “K” to a decimal number, “H” to hexadecimal number and “E” to a floating point (real number) for operations associated with numeric values in instructions. (Examples: K100 (decimal number), H64 (hexadecimal number) and E1.23 (or E1.23+10) (real number))

The roles and functions of constants are described below.

Constant K (decimal number):

“K” indicates a decimal integer, and is mainly used to specify the set value of timers and counters and numeric values as operands in applied instruction. (Example: K1234)

The decimal constant specification range is as follows:

- When word data (16 bits) is used … K-32768 to K32767
- When double data (32 bits) is used … K-2,147,483,648 to K2,147,483,647

Constant H (hexadecimal number):

“H” indicates a hexadecimal number, and is mainly used to specify numeric values a operands in applied instructions. (Example: H1234)

When using digits 0 to 9, the bit status (1 or 0) of each bit is equivalent to the BCD code, so BCD data can be specified also.

(Example: H1234 … When specifying BCD data, specify each digit of hexadecimal number in 0 to 9.)

The hexadecimal constant setting range is as follows:

- When word data (16 bits) is used … H0 to HFFFF (H0 to H9999 in the case of BCD data)
- When double data (32 bits) is used … H0 to HFFFFFFFF (H0 to H99999999 in the case of BCD data)

Constant E (real number):

“E” indicates a real num (floating point data), and is mainly used to specify numeric values as operands in applied instructions. (Example: E1.234 or E1.234+3)

The real number setting range is from -1.0×2^{128} to -1.0×2^{-126}, 0 and 1.0×2^{-126} to 1.0×2^{128}.

In a sequence program, a real number can be specified in two methods, “normal expression” and "exponent expression".

- Normal expression: ….. Specify a numeric value as it is.
  For example, specify "10.2345" in the form "E10.2345"
- Exponent expression: ... Specify a numeric value in the format "(numeric value) $\times 10^n$".
  For example, specify "1234" in the form "E1.234+3".
  "+3" in "E1.234+3" indicates "10^3".

Chapter VI Applied Instruction Interpretation

1 List of applied instruction

See the following table on types of applied instructions: form in accordance with function sequences

<table>
<tr><td rowspan="2">Classification</td><td colspan="3">Applied instruction</td><td rowspan="2">16/32 Bit</td><td rowspan="2">P</td><td colspan="2">Number of step</td></tr>
<tr><td>Instruction No.</td><td>Symbol</td><td>Instruction function</td><td>16bit</td><td>32bit</td></tr>
<tr><td rowspan="10">Program flow</td><td>00</td><td>CJ</td><td>Conditional jump</td><td>16</td><td>√</td><td>3</td><td>——</td></tr>
<tr><td>01</td><td>CALL</td><td>Sub-program call</td><td>16</td><td>√</td><td>3</td><td>——</td></tr>
<tr><td>02</td><td>SRET</td><td>Sub-program return</td><td>16</td><td></td><td>1</td><td>——</td></tr>
<tr><td>03</td><td>IRET</td><td>Interruption return</td><td>*1</td><td></td><td>1</td><td>——</td></tr>
<tr><td>04</td><td>EI</td><td>Interruption permitted</td><td>*1</td><td></td><td>1</td><td>——</td></tr>
<tr><td>05</td><td>DI</td><td>Interruption inhibited</td><td>*1</td><td>√</td><td>1</td><td>——</td></tr>
<tr><td>06</td><td>FEND</td><td>Main program end</td><td>*1</td><td></td><td>1</td><td>——</td></tr>
<tr><td>07</td><td>WDT</td><td>Timer</td><td>*1</td><td>√</td><td>1</td><td>——</td></tr>
<tr><td>08</td><td>FOR</td><td>Cycle loop start</td><td>*1</td><td></td><td>3</td><td>——</td></tr>
<tr><td>09</td><td>NEXT</td><td>Cycle loop end</td><td>*1</td><td></td><td>1</td><td>——</td></tr>
<tr><td rowspan="14">Sending and comparison</td><td>10</td><td>CMP</td><td>Comparison</td><td>16/ 32</td><td>√</td><td>7</td><td>13</td></tr>
<tr><td>11</td><td>ZCP</td><td>Inter-zone comparison</td><td>16/ 32</td><td>√</td><td>9</td><td>17</td></tr>
<tr><td rowspan="5">12</td><td rowspan="5">MOV</td><td>Sending</td><td>16/ 32</td><td>√</td><td rowspan="5">5</td><td rowspan="5">9</td></tr>
<tr><td>Constant sent to storage device, one character</td><td></td><td></td></tr>
<tr><td>Constant sent to storage device, two characters
Storage device sent to storage device, one character</td><td></td><td></td></tr>
<tr><td>Storage device sent to storage device, two characters</td><td></td><td></td></tr>
<tr><td>Non-character or word/special data range</td><td></td><td></td></tr>
<tr><td>13</td><td>SMOV</td><td>Shift moving</td><td>16</td><td>√</td><td>11</td><td>——</td></tr>
<tr><td>14</td><td>CML</td><td>Reverse moving</td><td>16/ 32</td><td>√</td><td>5</td><td>9</td></tr>
<tr><td>15</td><td>BMOV</td><td>Block moving</td><td>16</td><td>√</td><td>7</td><td>——</td></tr>
<tr><td>16</td><td>FMOV</td><td>Multi-point moving</td><td>16/ 32</td><td>√</td><td>7</td><td>13</td></tr>
<tr><td>17</td><td>XCH</td><td>Exchange</td><td>16/ 32</td><td>√</td><td>5</td><td>9</td></tr>
<tr><td>18</td><td>BCD</td><td>BCD conversion</td><td>16/ 32</td><td>√</td><td>5</td><td>9</td></tr>
<tr><td>19</td><td>BIN</td><td>BIN conversion</td><td>16/ 32</td><td>√</td><td>5</td><td>9</td></tr>
</table>

Classification	Applied instruction			16/32 Bit	P	Number of step	
	Instruction No.	Symbol	Instruction function			16bit	32bit
Arithmetic	20	ADD	BIN addition	16/ 32	√	7	13
	21	SUB	BIN subtraction	16/ 32	√	7	13
	22	MUL	BIN multiple	16/ 32	√	7	13
	23	DIV	BIN division	16/ 32	√	7	13
	24	INC	BIN plus 1	16/ 32	√	3	5
	25	DEC	BIN minus 1	16/ 32	√	3	5
	26	WAND	WAND	16/ 32	√	7	13
	27	WOR	WOR	16/ 32	√	7	13
	28	WXOR	WXOR	16/ 32	√	7	13
	29	NEG	NEG	16/ 32	√	3	5
Cyclic shift	30	ROR	Cyclic moving right	16/ 32	√	5	9
	31	ROL	Cyclic moving left	16/ 32	√	5	9
	32	RCR	Carrying cyclic moving right	16/ 32	√	5	9
	33	RCL	Carrying cyclic moving left	16/ 32	√	5	9
	34	SFTR	Bit moving right	16	√	9	——
	35	SFTL	Bit moving left	16	√	9	——
	36	WSFR	Word shift right	16	√	9	——
	37	WSFL	Word shift left	16	√	9	——
	38	SFWR	Shift write	16	√	7	——
	39	SFRD	Shift read	16	√	7	——
Data processing	40	ZRST	Batch return	16	√	5	——
	41	DECO	Decoding	16	√	7	——
	42	ENCO	Coding	16	√	7	——
	43	SUM	ON bit number	16/ 32	√	5	9
	44	BON	Check state of specified bit	16/ 32	√	7	13
	45	MEAN	Mean	16/ 32	√	7	13
	46	ANS	Signal alarm setting	16		7	——
	47	ANR	Signal alarm resetting	16	√	1	——
	48	SQR	Square	16/ 32	√	5	9
	49	FLT	BIN integral— floating-point number	16/ 32	√	5	9
High-speed processing	50	REF	Refreshing of input and output	16	√	5	——
	52	MTR	Matrix input	16		9	——
	53	HSCS	High speed counting setting	32		——	13
	54	HSCR	High speed counting resetting	32		——	13
	55	HSZ	High speed counting inter-zone comparison	32		——	17
	56	SPD	Pulse density	16		7	——

Classification	Applied instruction			16/32 Bit	P	Number of step	
	Instruction No.	Symbol	Instruction function			16bit	32bit
	57	PLSY	Pulse output	16/32		7	13
	58	PWM	Pulse adjustment	16		7	——
	59	PLSR	Pulse output with acceleration and deceleration	16/32		9	17
Convenient instructions	60	IST	Initializing state	16		7	——
	61	SER	Data searching	16/32		9	17
	62	ABSD	Absolute means for cam control	16/32		9	17
	63	INCD	Increment means for cam control	16		9	——
	64	TTMR	Demonstration teaching timer	16		5	——
	65	STMR	Special timer	16		7	——
	66	ALT	Alternative output	16	√	3	——
	67	RAMP	Slope signal	16		9	——
	68	ROTC	Rotating working bench control	16		9	——
	69	SORT	Data arrangement	16		11	——
Peripheral equipment input and output	70	TKY	Digit key input	16/32		7	13
	71	HKY	16-key input	16/32		9	17
	72	DSW	Digit switch	16		9	——
	73	SEGD	7-section decoding	16	√	5	——
	74	SEGL	7-section display as per time	16		7	——
	75	ARWS	Arrow switch	16		9	——
	76	ASC	ASCII code	16		11	——
	77	PR	ASCII code printing output	16		5	——
Peripheral Equipment SER	80	RS	Serial data sending	16		11	——
	81	PRUN	Octal code bit sending	16/32	√	5	9
	82	ASIC	HEX-ASCII conversion	16	√	7	——
	83	HEX	ASCII-HEX conversion	16	√	7	——
	84	CCD	Check code	16	√	7	——
	85	VRRD	Potential value read	16	√	5	——
	86	VRSC	Potential scale	16	√	5	——
	87	MBUS	MODBUS	16		11	——
	88	PID	PID control loop	16		9	——
	89	EPSC	Scale of extended card	16	√	9	——
9 Floating-point computation	110	ECMP	Floating-point comparison	32	√	——	13
	111	EZCP	Floating-point inter-zone comparison	32	√	——	17
	112	EMOV	Moving of floating-point number	32	√	——	9
	118	EBCD	Binary floating-point – Decimal floating-point conversion	32	√	——	9

Classification	Applied instruction			16/32 Bit	P	Number of step	
	Instruction No.	Symbol	Instruction function			16bit	32bit
	119	EBIN	Decimal floating-point – Binary floating-point conversion	32	√	——	9
	120	EADD	Floating-point addition	32	√	——	13
	121	ESUB	Floating-point subtraction	32	√	——	13
	122	EMUL	Floating-point multiple	32	√	——	13
	123	EDIV	Floating-point division	32	√	——	13
	124	EXP	Exponent arithmetic computation	32	√	——	9
	125	LOGE	Natural logarithm computation	32	√	——	9
	126	LOG10	Common logarithm computation	32	√	——	9
	127	ESQR	Floating-point square root	32	√	——	9
	128	ENEG	Binary floating-point numbers NEG computation	32	√	——	5
	129	INT	Binary floating-point –BIN integral conversion	16/32	√	5	9
	130	SIN	SIN	32	√	——	9
	131	COS	COS	32	√	——	9
	132	TAN	TAN	32	√	——	9
	133	ASIN	ASIN	32	√	——	9
	134	ACOS	ACOS	32	√	——	9
	135	ATAN	ATAN	32	√	——	9
	136	RAD	Radian computation	32	√	——	9
	137	DEG	Floating-point radian—>Angle	32	√	——	9
	147	SWAP	Upper and lower character conversion	16/32	√	3	5
Location	156	ZRN	Origin return	16/32		9	17
	157	PLSV	Pulse output with variable width	16/32		7	13
	158	DRVI	Relative location	16/32		9	17
	159	DRVA	Absolute location	16/32		9	17
Time computation	160	TCMP	Clock data comparison	16	√	11	——
	161	TZCP	Clock inter-zone comparison	16	√	9	——
	162	TADD	Clock data addition	16	√	7	——
	163	TSUB	Clock data subtraction	16	√	7	——
	166	TRD	Read RTC data	16	√	3	——
	167	TWR	Set RTC data	16	√	3	——
Peripheral equipment	170	GRY	Decimal system-Grey code conversion	16/32	√	5	9
	171	GBIN	Grey code- Decimal system conversion	16/32	√	5	9
Peripheral	188	CRC	CRC check	16	√	7	

Classification	Applied instruction			16/32 Bit	P	Number of step	
	Instruction No.	Symbol	Instruction function			16bit	32bit
communication	190	DTLK	Data Link	16		3	——
	191	RMIO	Remote I/O	16		3	——
	192	TEXT	OP07/08 text instruction	16	√	7	——
	193	DTLK2	Data Link2	16		7	——
Contact-point comparison	224	LD	(S1)=(S2)	16/32		5	9
	225		LD (S1)>(S2)	16/32		5	9
	226		LD (S1)<(S2)	16/32		5	9
	228		LD (S1)≠(S2	16/32		5	9
	229		LD (S1)≦(S2	16/32		5	9
	230		LD (S1)≧(S2	16/32		5	9
	232		AND (S1)=(S2)	16/32		5	9
	233		AND (S1)>(S2)	16/32		5	9
	234		AND (S1)<(S2)	16/32		5	9
	236		AND (S1)≠(S2	16/32		5	9
	237		AND (S1)≦(S2	16/32		5	9
	238		AND (S1)≧(S2	16/32		5	9
	240		OR (S1)=(S2)	16/32		5	9
	241		OR (S1)>(S2)	16/32		5	9
	242		OR (S1)<(S2)	16/32		5	9
	244		OR (S1)≠(S2	16/32		5	9
	245		OR (S1)≦(S2	16/32		5	9
	246		OR (S1)≧(S2	16/32		5	9

F00~F09 Program flow

Program chart

Function No.	Memory view	Name	Page
00	CJ	Conditional jump	1
01	CALL	Call sub-program	3
02	SRET	Sub-program return	3
03	IRET	Interruption return	5
04	EI	Interruption permitted	5
05	DI	Interruption inhibited	5
06	FEND	Main program end	10
07	WDT	Monitor timer	11
08	FOR	Cycle start	12
09	NEXT	Cycle end	12

F00 CJ Conditional jump

F 0		CJ	P	Conditional jump	

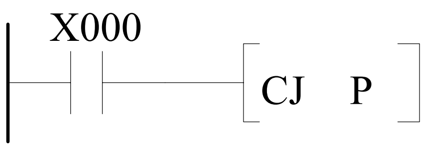
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

Instruction format:

P: destination symbol of conditional jump

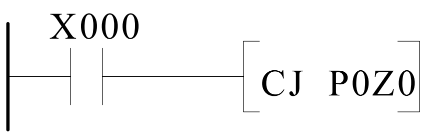
When PLC is used, if some of the program does not need executing every time, such instruction can be used to shorten the time of executing.

For instance,



Range of point (P) is: P0~P255 (TP03 M/H type)，P0~P127 (TP03 SR type). Of them, P63 refers to END, and do not program it. Otherwise, it may lead to error.

Index change can be done for the pointer number. See the figure below:

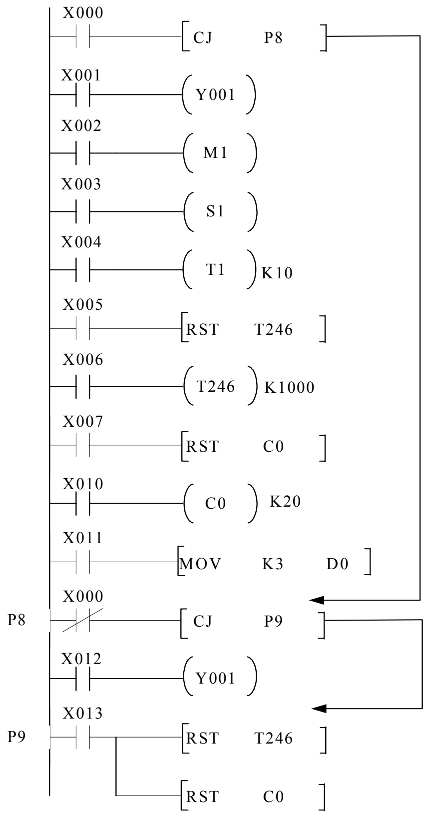


It is jumped to (0+ (Z0)) for executing.

During executing of conditional jump, description of the element actions:

1. Y, M and S keep the state before jump;
2. 10ms and 100ms timers will suspend timing;
3. 1ms timer T246~T249 will continue timing and the output contact-joint will act normally .
4. Timer T192~T199 for executing subprogram will continue timing and the output contact-joint will act normally;
5. High-speed counter during timing will continue counting and the output contact-joint will act normally;
6. General counter will stop counting;
7. If clear instruction for the integrating counter and timer is drive before jump, during jump executing, it is at clearing stage;
8. General applied instructions will not be executed.
9. Applied instructions during executing FNC53(DHSCS), FNC54(DHSCR), FNC55 (DHSZ), FNC56 (SPD), FNC57 (PLSY)and FNC58 (PWM) will continue executing.

For example:



The following table describes results of state change of the element during program jump:

Element	State of contact-joint before jump	Action of contact-joint during jump	Action of coil during jump
Y, M, S	X001, X002, X003 OFF	X001, X002, X003 ON	Y001, M1, S1 OFF
	X001, X002, X003 ON	X001, X002, X003 OFF	Y001, M1, S1 ON
10ms,100ms timer	X4 OFF	X4 ON	The timer dos not act
	X4 ON	X4 OFF	Timing stops, it continues after X0 OFF
1ms timer	X5, X6 OFF	X6 ON	The timer does not act
	X5 OFF, X6 ON	X6 OFF	Timing stops, it continues after X0 OFF
Counter	X007, X010 OFF	X010 ON	The counter does not count
	X007 OFF, X010 ON	X010 OFF	Counting stop, after X0 OFF, it continues counting
Applied instructions	X011 OFF	X011 ON	Applied instruction does not execute
	X011 ON	X011 OFF	The jumped applied instruction does not execute

- Y001 becomes dual-coil, no matter jump inside or outside, it is processed as general dual-coil.
- When reset instructions of the accumulated timer and counter jump outside, reset of the timing coil and counting coil (clearing of contact-joint recovery and current value) are valid.

F01 CALL Call sub-program

F 1		CALL	P	Call sub-program	

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

Instruction format:

[CALL P]

Range of pointer (P): P0~P255(TP03 M/H type)，P0~P127(TP03 SR type). Of them, P63 refers to END, which can not be used as pointer of FNC01 (CALL). Index change is available for pointer number.

F02 SRET Sub-program return

F 2		SRET		Sub-program return	

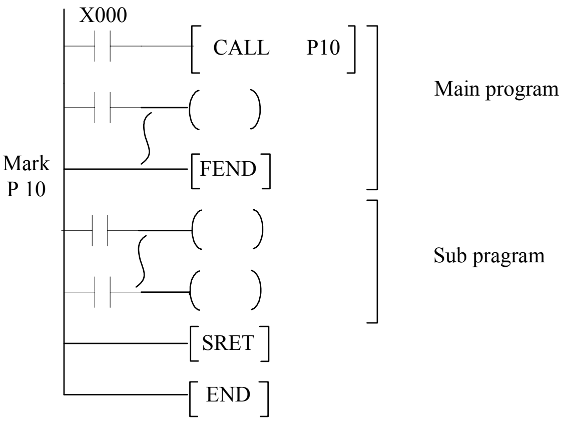
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

Instruction format:

[SRET]

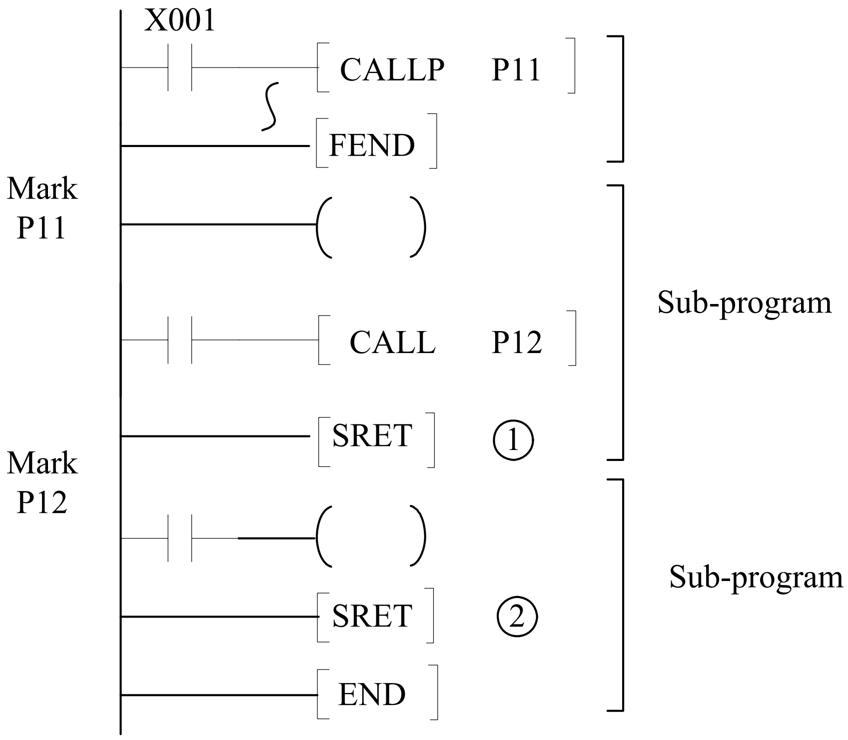
For sub-program return, there is no applicable device.

Example 1:



- When X000 is ON, the instruction CALL is executed, and it is executed after jumping to P10. Executing the sub-program here, when the instruction SRET is executed, it is returned to the original step.
- The pointer program shall be written after the instruction FEND.
- When it is used with the instruction CJ, the same number P can not be used.

Example 2:



- After X001=OFF→ON, the instruction CALL P P11 is executed one time, it jumps to P11.
- In the sub-program of P11, if the instruction CALL P12 is executed, sub-program of P12 is executed. After executing the instruction SRET, it returns to sub-program of P11, then the instruction SRET is executed to return to the main program.
- There are 16 layers of nesting in maximum.
- Timer in the sub-program employs T192~T199 or T246~T249.

F03 IRET Interruption return

F 3		IRET		Interruption return	

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

F04 EI Interruption permitted

F 4		EI		Interruption permitted	

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

F05 DI Interruption inhibited

F 5		DI		Interruption inhibited	

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

The above 3 instructions are single instructions without drive contact-joints.

- Normally, the PLC is under interruption inhibition. If interruption is required, the instruction FNC4 (EI) can be used for interruption.
- The pointer for interruption (I***) must be marked and programmed after the instruction. FEND.
- Types of interruption:
  1. Input interruption of external signal
  2. Interruption of timer
  3. Interruption of high speed counter

It is described below.

Input interruption of external signal

X000~X005 input signals are employed to execute routine program for interruption, applicable for pickup of adjustment control and short time pulse.

See the following table on number and action of interruption pointer of 6 points.

I x 0 x

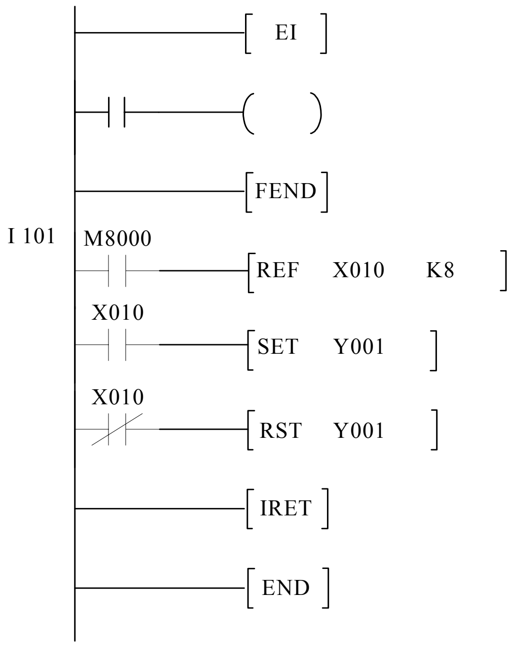
→ 0: Interruption of falling edge 1: Interruption of rising edge

→ X000~X005 are corresponding 0~5.

Input	Pointer number		Instruction of interruption inhibition
	Interruption of rising edge	Interruption of falling edge	
X000	I001	I000	M8050
X001	I101	I100	M8051
X002	I201	I200	M8052
X003	I301	I300	M8053
X004	I401	I400	M8054
X005	I501	I500	M8055

- The pointer number can not be repeatedly. As for the same outputs, the corresponding fall edge interruption and falling edge interruption numbers can not be used for the input.
- If M8050~M8055 is ON, interruption to corresponding input is inhibited.

For example: the interruption processing must be executed for latest input information.

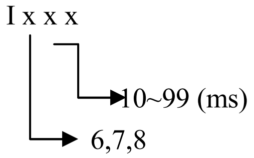


- Rising edge testing of X001is ON, routine program for interruption and input refreshing are executed. According to state of ON/OFF of X010, set Y001 or reset.

Interruption of the timer

It is not affected by scan cycle of the controller and it executes the interruption sub-program within an interval of 10ms~99ms. Under the situations of long computation cycle of the main program, the program to be determined needs a long time to process; or in the sequential control scan, it is appropriate for executing the program with an interval.

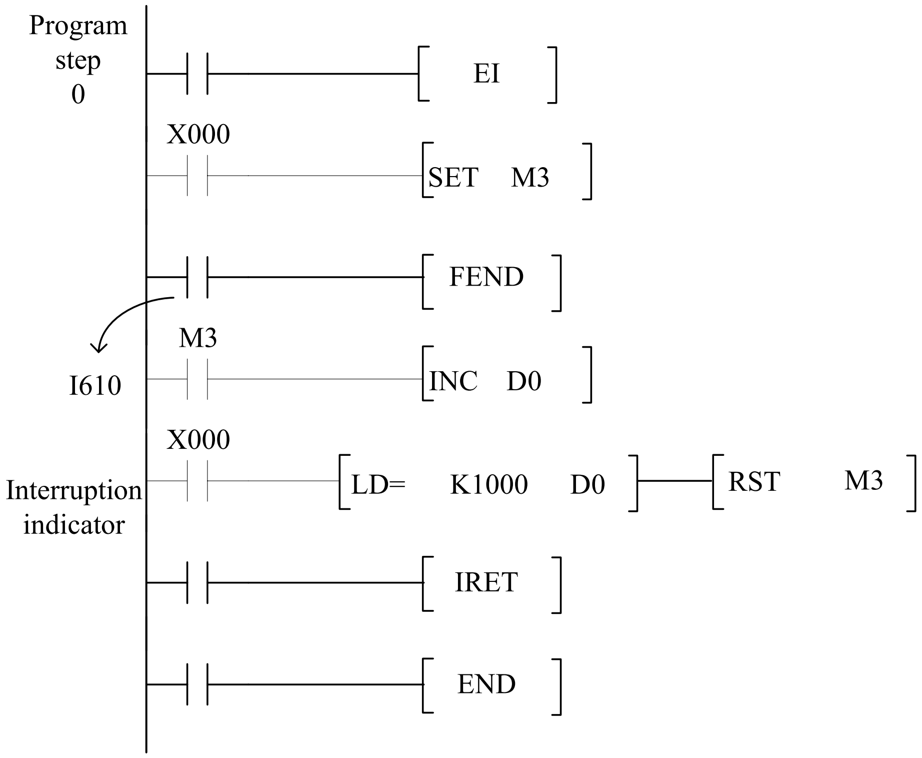
See the following table on numbers and actions of interruption pointers of corresponding 3 points:



Pointer No.	Interruption cycle	Instruction for interruption inhibition
I6xx	xx：integral of 10~99, representing interval of interruption	M8056
I7xx		M8057
I8xx		M8058

- Pointers (I6,I7,I8) can not be repeatedly.
- Set M8056~M8058 ON, and interruption is not allowed for corresponding timer.

For example: 1 is added on each 10ms, and compare with setting values.



- After each 10ms, 1 is added on current value of D0.
- When current value of D0 reaches 1000, M3 resets.

Interruption of high-speed counter

The interruption of current value of the high-speed counter is used with the comparison setting of FNC53 (DHSCS). When current value of the high-speed counter reaches specified values, the sub-program is interrupted.

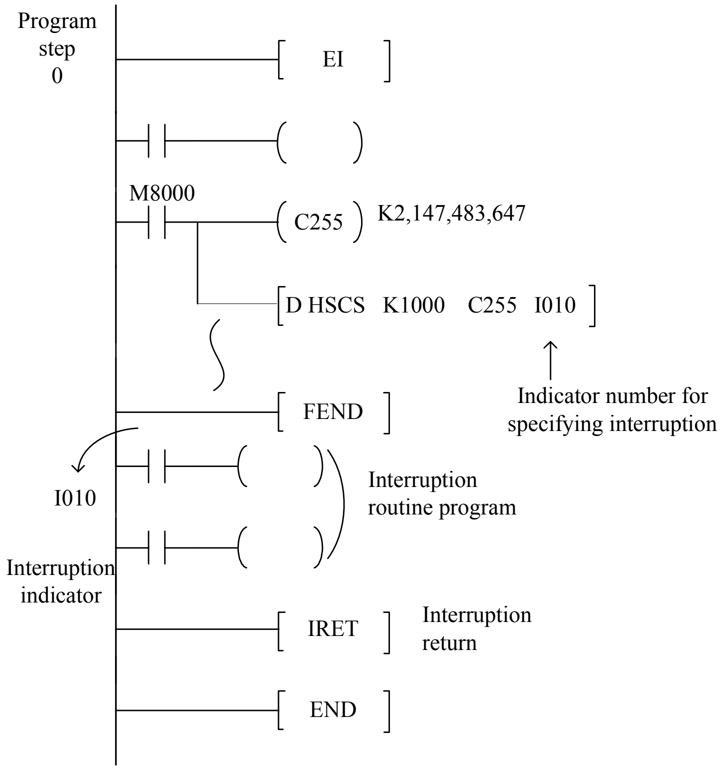
See the following table on interruption pointer numbers and actions of corresponding 6 points.

I 0 x 0

1~6

Pointer number	Instruction for interruption inhibition
I010	M8059
I020	
I030	
I040	
I050	
I060	

Example



- For the coil driving the high speed counter, it is used to specify the interruption pointer in the instruction FNC (DHSCS).
- When current value of C255 varies from 999~1000, sub-program interruption is executed.
- As for current values of the high speed counter, if active/inactive control can be done for the output relay or auxiliary relay, the instructions FNC53(DHSCS), FNC54(DHSCR), FNC55(DHSZ) can be used to simply the program.

Several interruption inputs

- In case of several interruptions, the earlier interruption shall be prior. In case of occurrence simultaneously, the smaller pointer number shall be prior.
- During executing of interruption of routine program, other interruptions are not allowed. Information of interruption during the period shall be kept and it will be executed after the routine program and 8 interruptions in maximum.

Pulse width of input interruption

- If external signal is used for executing input interruption, ON or OFF signal with pulse width more than 50us shall be input.

Recovery of input and output

- During interruption processing, when controlling the input relay and output relay, the recovery instruction FNC (REF) and output relays can be used to obtain latest input information or output computation results immediately to control without affecting by the computation cycle.

Attentions

- As the number of input relay for interruption pointer, please do not use the applied instructions for high speed counter, pulse density and etc. with the same input range for repeated numbers.
- Please use the timer T192-T199 for routine program for timer in the sub-program and routine program interruption. If general timer is used, timing can not be performed. When 1ms accumulated timer is used, pay attention to it.

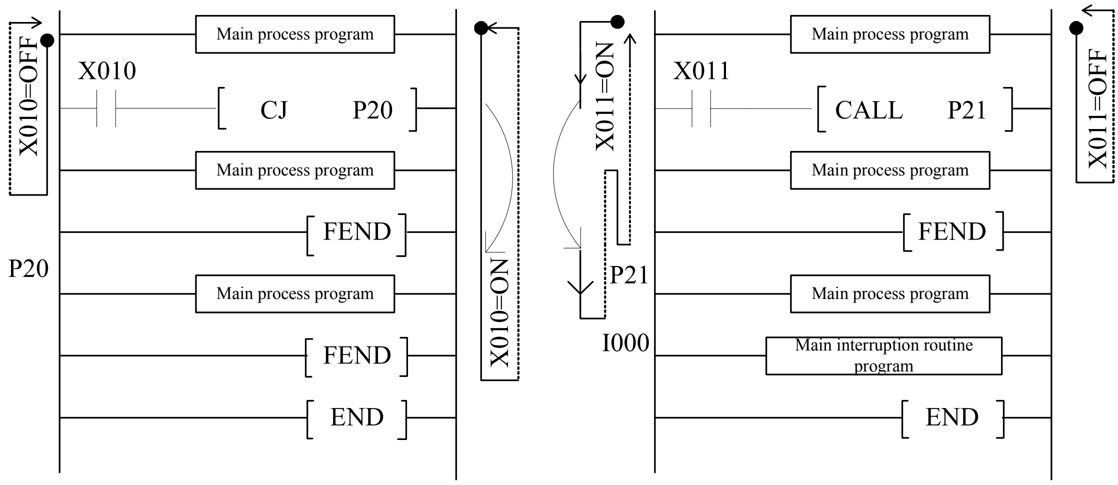
F06 FEND Main program end

F 6		**FEND**	Main program end	

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

The instruction is an independent instruction without driving contact-joint, representing end of the main program.

The instruction has the same effect with the instruction END, executing output processing, input processing, refreshing of monitor timer and returning to step 0 of the program.



- The instructions CALL and CALL P must be written after the instruction FEND, with the instruction SRET to end the sub-program. The interruption program must be written after the instruction FEND, with the instruction IRET to end the interruption program.
- After executing the instructions CALL and CALL P and before executing the instructions SRET and IRET; or after executing the instruction FOR and before executing the instruction NEXT; if the instruction FEND is executed, it may lead to abnormal program.
- If there are several FEND instruction, the sub-program and interruption program shall be written between the last FEDN and END instructions.

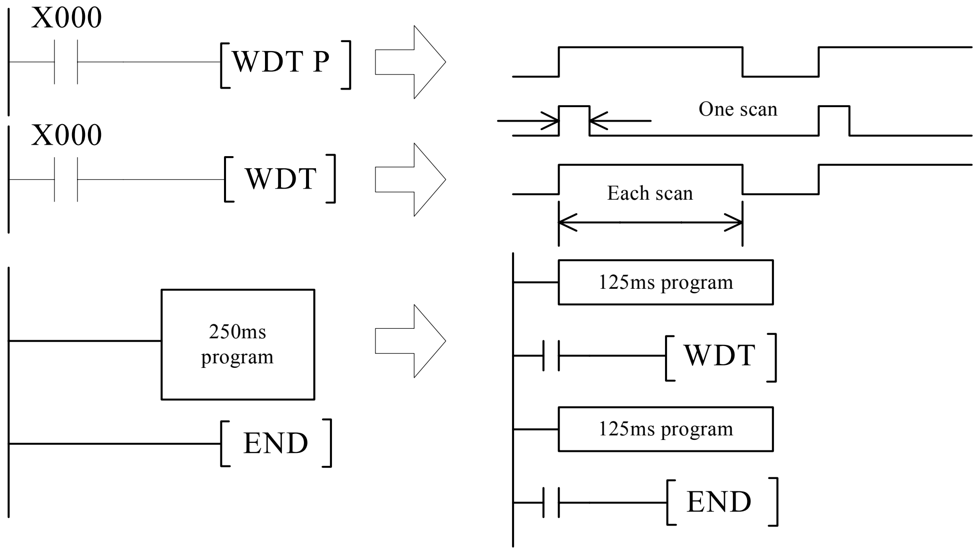
F07 WDT Monitor timer

F		**WDT**		Monitor timer	
7			P		

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

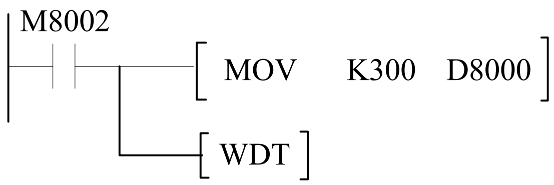
The instruction is used for refreshing of the monitor timer to avoid error of the controller caused by delay of the scan cycle.

For example: If the scan time exceeds specified values, PLC will stop operation. Under the circumstances, the instruction WDT will be inserted into appropriate program step to refresh the timer. Value of the monitor timer is set by D8000, with range of 200ms-1600ms.



If value of the monitor timer is set at 200ms, when scan time of the program is 250ms, it is divided into two parts. Insert WDT into it, and the first part and second part of the program is less than 200ms.

- Testing time of the monitoring timer can be changed by rewriting content of D8000, see the following figure:



- When the system is connected to many station locating, cam switch, ID interface, link, analog quantity and other special extended equipment, when the controller runs, initializing time of the buffing storage device will be extended to cause delay of the scan time. Besides, when executing several FROM/TO instructions for sending data to several buffing storage devices, the time will be delayed. Under the circumstances, it may cause abnormality of timeout monitor timer, at this time, input the above program near the starting step to extend time of the monitor timer.

F08 FOR Cycle start

F 8		FOR		Cycle start	S1 •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*

Instruction format:

—[FOR S •]

F09 NEXT Cycle end

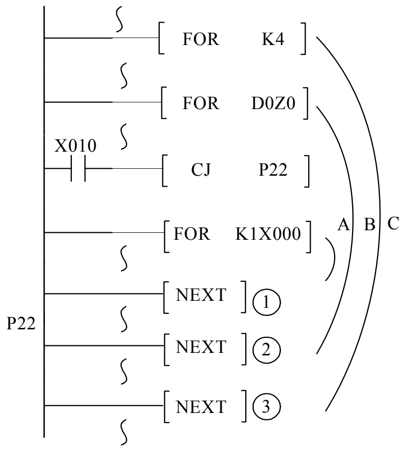
F 9		NEXT		Cycle end	

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

The instruction is a single instruction with driving contact-joint and used with FNC08(FOR).

After the instruction executes the instruction between FOR and NEXT for n times (specifying by the instruction FOR), then steps after NEXT can be processed. It is valid when n=1~32,767, when n= -32,767~0, it is processed as 1.

Example:



- After [c] program executed 4 times, it is moved to program after the instruction (3) of NEXT; if [c] program is executed, content of the data register D0Z0 is 6m the [B] program executes 6 times; the instruction CJ can be used to skip program between FOR~NEXT, like X010=ON.
- There are 16 layers of nesting, and FOR~NEXT must be in couple, otherwise, it may lead to error.
- Too many cycles may lead to delay of the scan cycle, which may cause error of the monitor timer and please pay attention.

F10~F19 Data moving and comparison

Data moving and comparison

Function No.	Memory view	Name	Page
10	CMP	Data comparison	1
11	ZCP	Zone comparison	2
12	MOV	Date moving	3
13	SMOV	Bit moving	4
14	CML	Contrary moving	5
15	BMOV	Batch moving	6
16	FMOV	Multi-point moving	7
17	XCH	Exchange	8
18	BCD	BIN→BCD	9
19	BIN	BCD→BIN	10

F10 CMP Data comparison

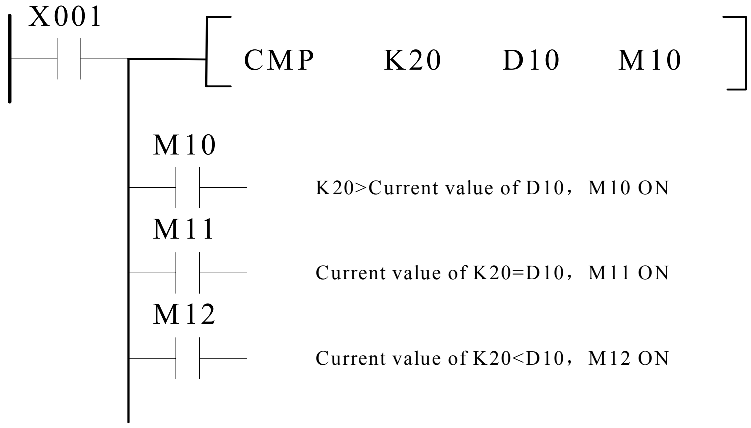
F 10	D	CMP	P	Data comparison	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •		*	*	*												

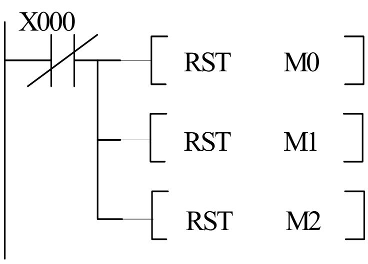
Instruction format:

[CMP S1 • S2 • D •]

S1•: Comparison value 1
S2•: Comparison value 2
D•: Comparison result, occupy continuous 3 points.
For example: compare the computation elements S1• and S2•, the results are stored in D•.



Compare S1·/S2· with the integrals with symbols.
The destination address occupies the following two. If Y001 is specified, Y002 and Y003 occupied automatically.
The 32-bit instruction destination operand can not specify V, and it can only specify Z. When Zn is specified, (Vn, Zn) make up 32-bit data. (The same for the following applied instruction if there is no special interpretation).
When the instruction is not executed, the destination data is not affected.
To clear the comparison results, the reset instruction or overall instruction shall be used.



F11 ZCP Zone comparison

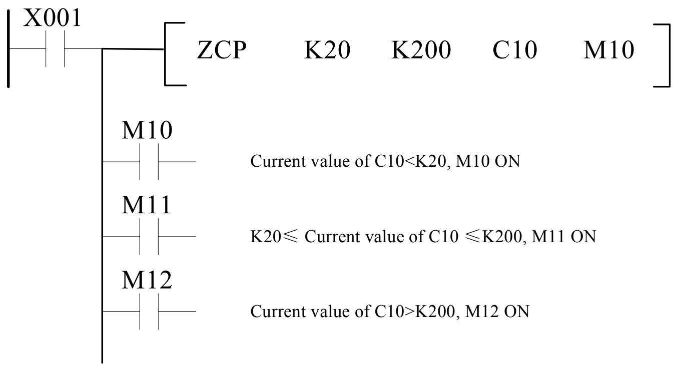
F 11	D	ZCP	P	Zone comparison	S1 •	S2 •	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •		*	*	*												

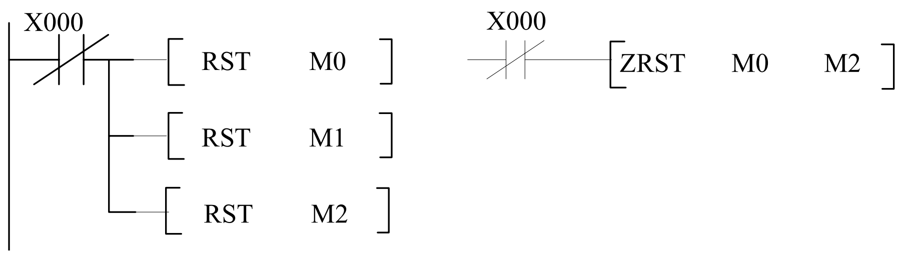
Instruction format:

[ZCP S1· S2· S· D·]

S1·: Lower limit of zone comparison
S2·: Upper limit of zone comparison
S·: Comparison value
D·: Comparison result, occupy continuous 3 points
The comparison value S· compares the lower limit value S1· and upper limit value, and the comparison result is stored in D·.
When the lower limit value S1·> upper limit value S2·, and the lower limit S1· can be used for comparison of upper and lower limits.
For example,



Data of S1·, S2· and S0· shall be compared by integral with symbols.
S1·≤S2· is required. When S2·<S1·, S2· is computed as S1·.
The destination address occupies 2 points automatically, if M0 is specified, it occupies M1 and M2.
When the instruction is not executed, the destination data is not affected.
If the results need to be cleared, the instructions RST or ZRST are used;



F12 MOV Date moving

F 12	D	MOV	P	Date moving	S • D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[MOV S • D •]

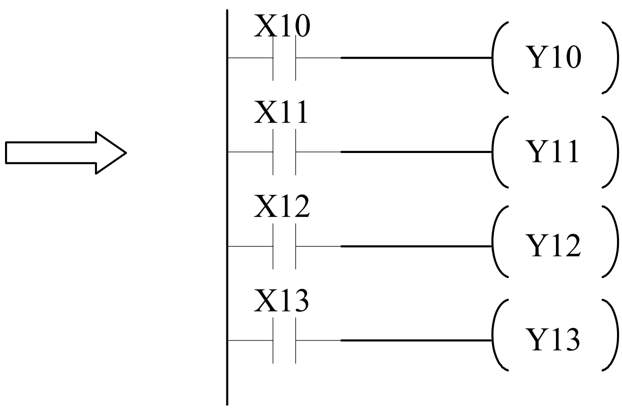
S•: Data source

D•: Destination address of data moving

Content of S• is directly sent to D•, when the instruction is not executed, content of D• will not be changed.

The 32-bit instruction employs DMOV instruction, and the operand occupies 2 characters automatically.

The 16-bit element information transmission, when the instruction is executed, 4 bit elements of X10-X13 are sent to Y10~Y13, which has the same function with the following program.



F13 SMOV Bit moving

F 13		SMOV	P	Bit moving	S •	m1	m2	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*	*	*
m1					*	*										
m2					*	*										
D •								*	*	*	*	*	*	*	*	*
n					*	*										

Instruction format:

[SMOV S • m1 m2 D • n]

S•: data source
m1: start bit number of data source.
m2: number of sent data source
D•: destination address of data moving
n: Start number of bit of destination address
m1/ m2/ n：1~4
For example:

[SMOV D10 K4 K2 D20 K3]

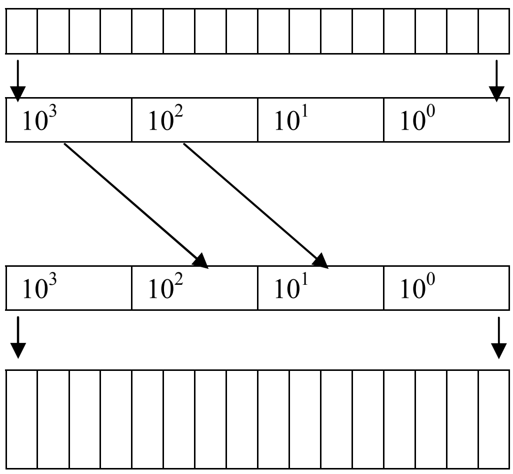
When M8168=OFF,

BIN code of D10

D10 converted into BCD code

BIN code of D20

Converted into BID code automatically

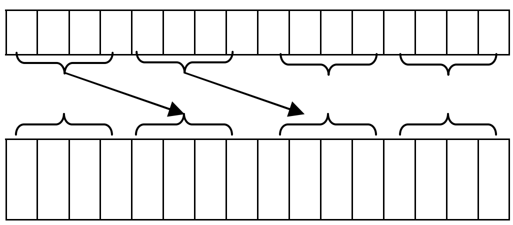


Conversion data of BCD of source data, the 2nd bit from the 4th bit is sent to the 3 bit of the D20.When BCD value of D10 exceeds 0~9,999, it will be error.

When M8168=ON:

BIN code of D10

BIN code of D20



BCD code is not executed. 4 bits are taken as one unit for bit moving.

F14 CML Contrary moving

F 14	D	CML	P	Contrary moving	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[CML S • D •]

S• : data source

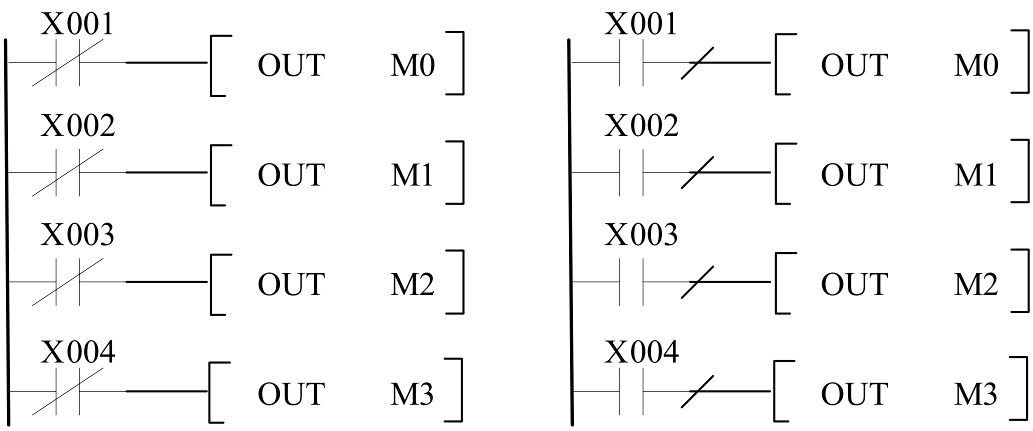
D• : destination address of data transmission

Content of S• is sent to D• contrarily (0→1, 1→0), if the content is the constant K, which is converted into BIN value automatically.

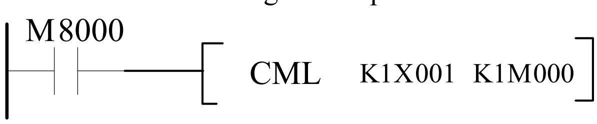
For instance,

D10													1	0	1	0

	Do not change												0	1	0	1
Y0	17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0



The above figure is equal to



F15 BMOV Batch moving

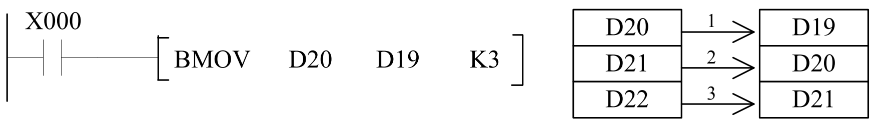
F 15		BMOV	P	Batch moving	S •	D •	n

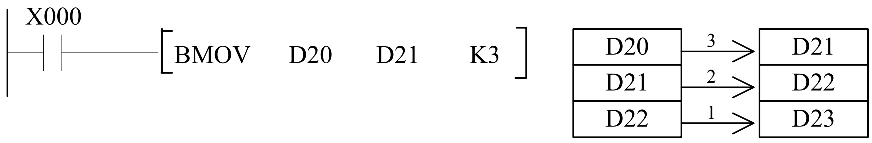
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*		
D •								*	*	*	*	*	*	*		
n					*	*							*	*		

Instruction format:

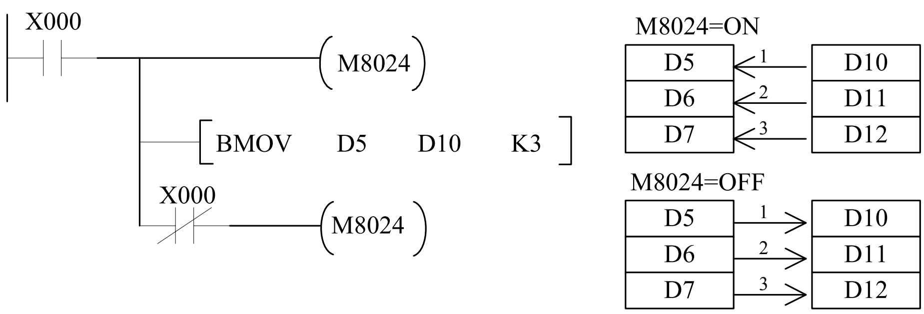
[BMOV S • D • n]

S•: start address of data source
D•: destination address of data moving
n: Length of moving block (n<=512)
The data of point n starting with specified device by the source starting address is transmitted to device of point n starting with devices specified by goal address. (If it exceeds number of range of the device, it will be transmitted to permissible scope).
See the following figure on the moving range. To avoid rewriting before moving, it is automatically moving as per the sequence 1-3.





Set M8024 ON, when executing the instruction, it rotates contrarily with the moving direction.



F16 FMOV Multi-point moving

F 16	D	FMOV	P	Multi-point moving	S •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*		
n					*	*										

Instruction format:

[FMOV S • D • n]

The multi-point moving instruction for the same data

n: n<=512

Content of S• is sent to the device beginning with n specified by D•. Content of device at point n is the same. When it is beyond range of the destination device, it will be moved to possible range.

Example:

X000

[FMOV D1 D10 K3]

Executing results: D1----------------->D10

D1----------------->D11

D1----------------->D12

Attentions

- The 16-bit instruction destination operand can not specify V and Z, the 32-bit instruction destination operand can not specify V, and it can only specify Z. When Zn is specified, (Vn, Zn) make up 32-bit data.

F17 XCH Exchange

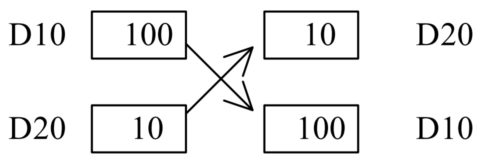
F 17	D	XCH	P	Exchange	D1 •	D2 •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D1 •								*	*	*	*	*	*	*	*	*
D2 •								*	*	*	*	*	*	*	*	*

Instruction format:

[XCH D1 • D2 •]

D1·: Exchange data 1
D2·: Exchange data 2
Specified content of D1· and D2· can be exchanged.
The instruction is pulse instruction XCHP in general.
For instance,



Please note that, when continuous executing instruction is used, data exchange is performed during each scan cycle.
When M8160= ON, D1·, D2· are the same device, exchange the low 8-bit and high 8-bit, the same for the 32-bit instruction.
When M8160= ON, D1·, D2· are different, the error mark is M8067 ON, D8067 writes error code, the instruction will not execute.
When M8160= ON, the executing function is the same as the instruction F147(SWAP).

F18 BCD BIN→BCD conversion

F 18	D	BCD	P	BIN→BCD conversion	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

S•: data source

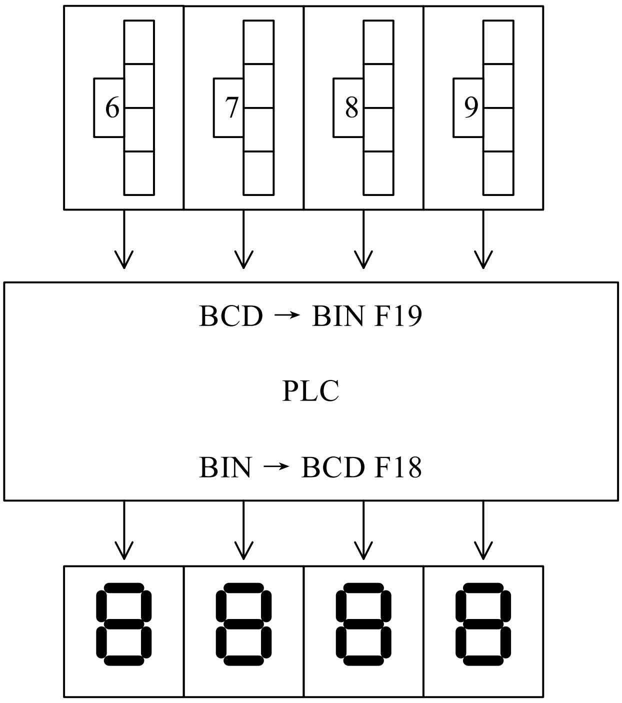
D•: Storage location

Function: data source S• executes conversion from BIN to BCD, which is stored in D•.

For 16-bit instruction, if the conversion result exceeds 0-9999, it is error.

For 32-bit instruction, if the conversion result exceeds 0~99999999, it is error.

In case of error, M8067= ON, D8067 writes error code, the instruction will not execute.



Arithmetic operation, increasing, decreasing and other instructions are executed in BIN in PLC.

When PLC reads data of external BCD digital switch, the conversion sending instruction FNC19(BCD→BIN) is used; and when it outputs to BCD seven-section display, the conversion sending instruction FNC18 (BIN→ BCD)is used.

When special instructions like FNC72(DSW), FNC74(SEGL), FNC75(ARWS) are used, it will execute BCD/BIN conversion automatically.

F19 BIN BCD→BIN Conversion

F 19	D	**BIN**	P	BCD→BIN Conversion	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[BIN S • D •]

S•: Data source

D•: Storage location

Function: content of data source S• (BCD: 0~9999 or 0~99999999) is converted into BIN format, and it is stored in D•.

If S• is not BCD code, error, M8067= ON, D8067 writes error code, and the instruction will not execute.

- It is used when PLC reads setting value of BCD digital switch. When data source is not BCD, it will be error.
- For the constant K is converted into binary system automatically, so it can not be the device for the instruction.

F20~F29 Arithmetic operation

Arithmetic operation

Function No.	Memory view	Name	Page
20	ADD	Addition computation	1
21	SUB	Subtraction computation	2
22	MUL	Multiple computation	3
23	DIV	Division computation	4
24	INC	Increasing computation	5
25	DEC	Decreasing computation	5
26	WAND	WAND computation	6
27	WOR	WOR computation	6
28	WXOR	WXOR computation	6
29	NEG	NEG computation	7

F20 ADD Addition computation

F 20	D	ADD	P	Addition computation	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[ADD S1 • S2 • D •]

S1· : Augend

S2· : Addend

D· : Sum

Symbol bit	Zero	M8020
	Borrowing	M8021
	Carrying	M8022

Function: S1· plus S2· with symbols, and the result is stored in D1.

If the computation result is 0, M8020 sets.

If the computation result is less than the minimum value, M8021is set.

If the computation result is more than the maximum value, M8022 is set.

Result of 16-bit computation is between -32,768 ~ +32,767.

Result of 32-bit computation is between-2,147,483,648 ~ +2,147,483,647.

- The two data sources are added in BIN and sent to the destination data zone, and the highest bit of the data is the positive (0) or negative (1) symbol bit, therefore, algebraic addition shall apply. (5+(-8)=-3)
- When the computation result is 0, the zero flag sign acts. When the computation result exceeds 32,767(16-bit computation) or 2,147,483,647(32-bit computation), the carrying flag sign will act. (Refer to the next page). If the computation result is less than -32,768(16-bit computation) or -2,147,483,648(32-bit computation), the negative flag sign will act. (Refer to the next page)
- For 32-bit computation, lower 16-bit element of Word element shall be specified and upper Word of specified number is used. To avoid repeated number, even number is used for specifying elements.
- Please specify the same number for the data source and destination data zone. If the continuous executing instruction (ADD,D ADD) is used, plus of each scan time is changing, and please pay attention to it.

X000 [ADD P D1 K1 D1]

- See the sequential control program in the above figure, when X000 is changed from OFF-ON, 1 is added on value of D1, which is similar with the following instruction INC P.

F21 SUB Subtraction computation

F 21	D	SUB	P	Subtraction computation	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[SUB S1 • S2 • D •]

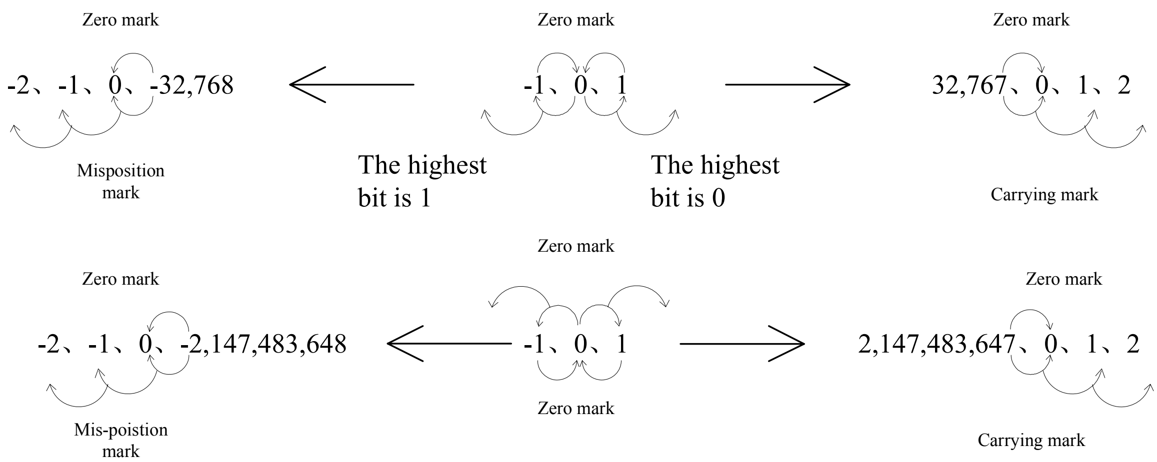
S1· : Minuend
S2· : Subtrahend
D· : Difference

Data in S1·and S2· are subtracted in BIN, and the result in D·.
For the highest bit of the data, 0 represents positive, 1 represents negative and algebraic subtraction is executed.

Example:

X000 [SUB D1 D2 D3]

- When X000 is ON, content of minuend D1 minus content of subtrahend D2, and the difference is stored in D3.
- Mark action, specifying method of 32-bit computation element, difference of the continuous executing and pulse executing and etc. are the same with the instruction ADD in previous page.
- See the following on relation of actions and values.



F22 MUL Multiple computation

F 22	D	MUL	P	Multiple computation	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[MUL S1 • S2 • D •]

S1· : Multiplicand
S2· : Multiplicator
D· : Product
Data in S1· and S2· are multiplied in BIN, and the result is stored in D·.
For the highest bit symbol of data, 0 represents positive and 1 represents negative, and algebraic multiplication is executed.

<16-bit computation >

X000 [MUL D0 D2 D4]

- The product computed from specified content of the data element is stored in specified element zone (the lower side), meanwhile, it occupies the upper element to make up 32-bit data. See the above figure, when D0)=8,(D2)=9, (D5,D4)=72.
- For the highest bit of the result, 0 for positive and 1 for negative.
- When D· is element, bit specifying for K1~K8 is required. When K4 is specified, low 16-bit of the product can be obtained.

<32-bit computation >

X000 [D MUL D0 D2 D4]

- See the figure below: (D1,D0)=8,(D3,D2)=9, (D7,D6,D5,D4)=72
- For 32-bit computation, if the result storage element is bit, the result is only lower position 32-bit and there is no upper position 32-bit and the Word element can be used for computation.
- When Word element is used, the computation result is stored in 64-bit, therefore, when Word element is used, the computation result is stored in 64-bit data and the result can not be viewed.
- D· can not specify Z element.

F23 DIV Division computation

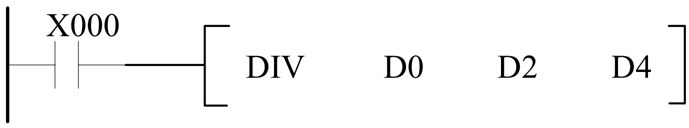
F		DIV		Division computation	S1 •	S2 •	D •
23	D		P				

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[DIV S1 • S2 • D •]

S1· : Dividend
S2· : Divisor
D· : Quotient
Data in S1· and S2· are divided by in, with result stored in D·.
For the highest bit symbol of the data, 0 represents positive, 1 represents negative and algebraic division is carried out.
<16-bit computation >



- See the figure above: D0 is the dividend, D2 is divisor, D4 stores the quotient by the computation and D5 stores the residual by the computation.

<32-bit computation >

- See the figure below, (D1,D0) are dividends, (D3,D2_are divisors,(D5,D4)store the quotient by the computation and (D7,D6) store the residual by the computation.
- D· can not specify Z element.

Note:
- When the divisor is 0, it may cause error and the instruction will not be executed D8067=6706.
- When D· is specified as bit element, residual can not be obtained.
- The uppermost of the quotient and residual is the symbol for positive (0) and negative (0). When the quotient is negative, any of the dividend and divisor is negative; if the residual is negative, the dividend is negative.

F24 INC Increasing computation

F 24	D	INC	P	Increasing computation	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[INC D •]

- The instruction is of pulse instruction in general. Otherwise, when the instruction is executed, 1 is added on each scan cycle D·, so please pay attention to it.
- For 16-bit computation, if 1 is added on +32,767, it becomes -32,768, and the mark does not act. For 32-bit computation, if 1 is added on +2,147,483,647, it becomes -2,147,483,648, and the mark does not act.

F25 DEC Decreasing computation

F 25	D	DEC	P	Decreasing computation	D •

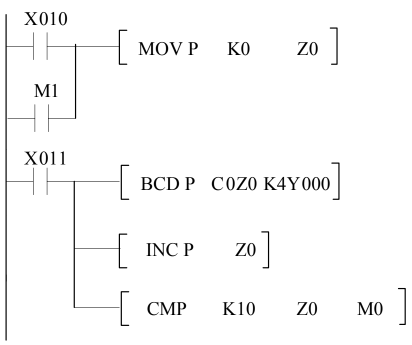
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[DEC D •]

- The instruction is of pulse instruction in general. Otherwise, when the instruction is executed, 1 is subtracted from each scan cycle D·, please pay attention to it.
- For 16-bit computation, if 1 is subtracted from -32,768, it becomes +32,767, the mark does not act.
  For 32-bit computation, if 1 is subtracted from -2,147,483,648, it becomes +2,147,483,647, the mark does not act.

<Application case>



- When current values of the counters C0~C9 are converted for BCD, it is output to K4Y000.
- Resetting input X010 is executed in advance to clear Z0.
- When X011 is ON one time, it outputs current values of C0, C1…C9.

F26 AND WAND computation

F 26	D	WAND	P	WAND computation	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[WAND S1 • S2 • D •]

F27 OR WOR computation

F 27	D	WOR	P	WOR computation	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[WOR S1 • S2 • D •]

F28 XOR WXOR computation

F 28	W D	WXOR	P	WXOR computation	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[WXOR S1 • S2 • D •]

S1· : Data source 1

S2· : Data source 2

D· : Computation result

F29 NEG NEG computation

F		NEG		NEG computation	D •
29	D		P		

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •								*	*	*	*	*	*	*	*	*

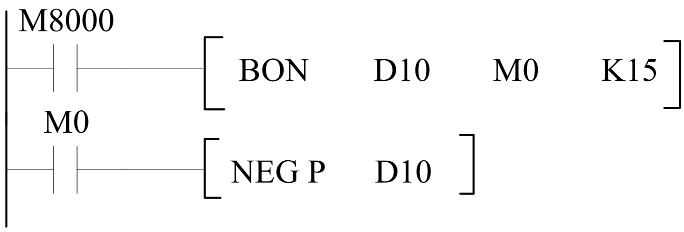
Instruction format:

[NEG D •]

- Reverse each bit of the content specified by D·, 1 is added on them, and the result is stored in D·.
- The instruction is of pulse instruction. The continuous executing instruction will execute the instruction after each scan cycle and please pay attention to it.
- When the instruction is used, the corresponding absolute value of negative BIN value shall be obtained.

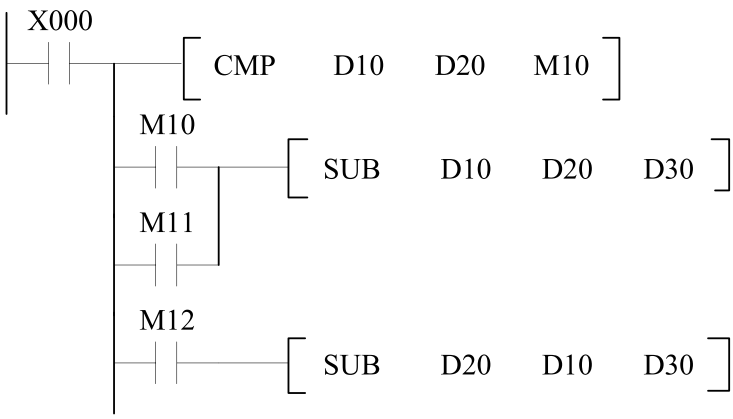
  Note: after computation is computed for -32,768, it is-32,768.

Absolute value processing of negative value of applied loop 1



When 15th of D10 is 1, M0 is set ON.
When M0 is ON, complement code for D10.

Absolute value processing of subtraction of applied loop 2



Even complement code is not used for the above loop, D30 represents absolute value of the difference in subtraction.

F30~F39 Rotating and shifting

Rotating and shifting

F 30 ROR Cyclic shifting right

F 30	D	ROR	P	Cyclic shifting right	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •								*	*	*	*	*	*	*	*	*
n					*	*										

Instruction format:

[ROR D • n]

F31 ROL Cyclic shifting left

F 31	D	ROL	P	Cyclic shifting left	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •								*	*	*	*	*	*	*	*	*
n					*	*										

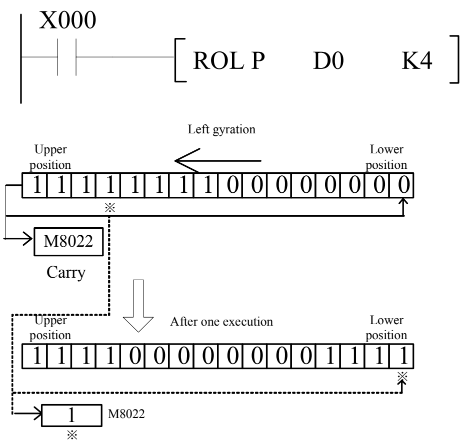
Instruction format:

[ROL D • n]

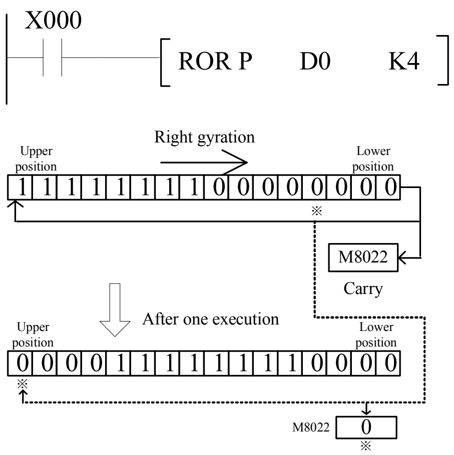
16-bit instruction n≤16; 32-bit instruction n≤32

The instruction for each bit of 16-bit or 32-bit data for left and right gyrations

Left gyration



Right gyration



Instruction interpretation: when X000 is Changed from OFF to ON, K4-bit left gyration is executed and the last bit is stored in the carrying flag sign M8022.

Instruction interpretation: when X000 is changed from OFF to ON, K4-bit right gyration is executed and the last bit is stored in the carrying flag sign M8022.

- The continuous executing instruction will gyrate after each scan cycle and please pay attention to it.
- It is the same for the 32-bit instruction.
- When bit is used to specify the element, only K4 (16-bit instruction) and K8 (32-bit instruction) are effective (Such as K4Y010 and K8M0)

F32 RCR Cyclic shifting right with carrying

F 32	D	RCR	P	Cyclic shifting right with carrying	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •								*	*	*	*	*	*	*	*	*
n					*	*										

Instruction format:

[RCR D • n]

F33 RCL Cyclic shifting left with carrying

F 33	D	RCL	P	Cyclic shifting left with carrying	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •								*	*	*	*	*	*	*	*	*
n					*	*										

Instruction format:

[RCL D • n]

16-bit instruction n≤16; 32-bit instruction n≤32

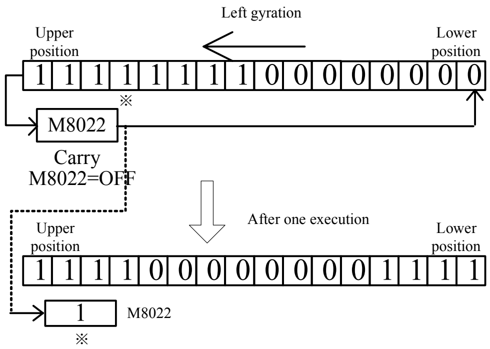
The left or right gyration instruction with carrying symbol M8022 for 16-bit or 32-bit data shall be used.

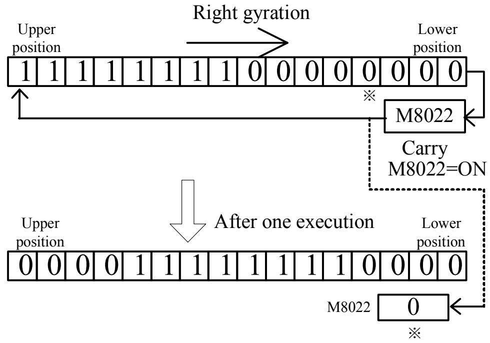
Left gyration with carrying

X000 [RCL P D0 K4]

Right gyration with carrying

X000 [RCR P D0 K4]





Instruction interpretation: when X000 is changed from OFF to ON, K4 with carrying symbol M8022 will gyrate to the left (right).

- The continuous executing instruction will gyrate after each scan cycle and please pay attention to it.
- It is the same for the 32-bit instruction.
- When the bit is used to specify the element, only K4 (16-bit instruction)and K8 (32-bit instruction) are valid.(such as K4Y010, K8M0)

F34 SFTR Bit moving right

F 34		SFTR	P	Bit moving right	S·	D·	n1	n2

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·	*	*	*	*												
D·		*	*	*												
n1					*	*										
n2					*	*										

Instruction format:

[SFTR S· D· n1 n2]

F35 SFTL Bit moving left

F 35		SFTL	P	Bit moving left	S·	D·	n1	n2

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·	*	*	*	*												
D·		*	*	*												
n1					*	*										
n2					*	*										

Instruction format:

[SFTL S· D· n1 n2]

S· : Start number of shift device

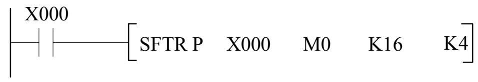
D· : Start number of device to be shift

n1: Data length to be shifted, n1=1~1024

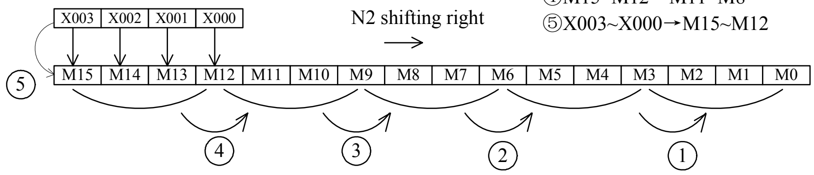
n2: Bit quantity for one shifting, n2=1~n1

- Data tandem with n1 word element (length of shifting buffer) started with D·, it moves right with n2 shifting number and number started with S· is moved to D· to fill the empty element with n2 word element.
- The instruction is of pulse instruction in general, when continuous executing instruction is used, it will execute for each scan cycle and pay attention to it.

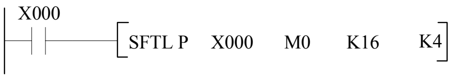
Bit shifting right



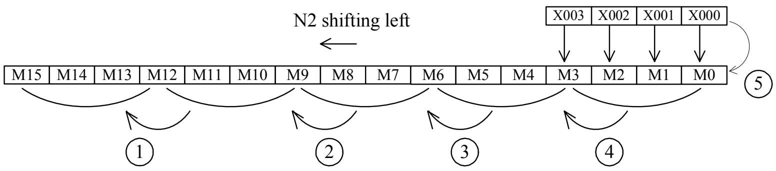
①M3~M0→Overflow
②M7~M4→M3~M0
③M11~M8→M7~M4
④M15~M12→M11~M8
⑤X003~X000→M15~M12



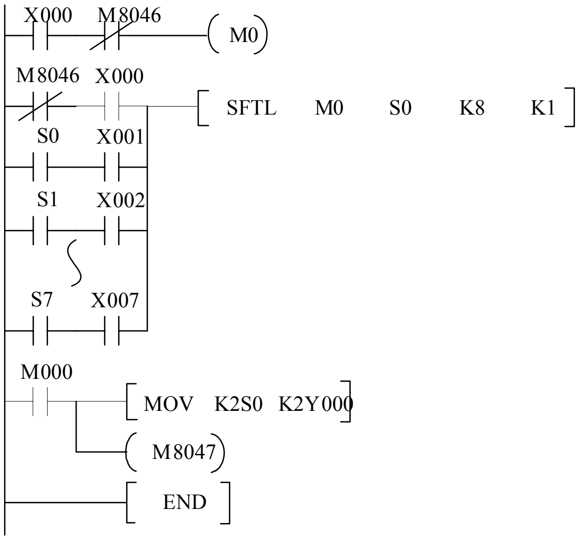
Bit shifting left



①M15~M12→Overflow
②M11~M8→M15~M12
③M7~M4→M11~M8
④M3~M0→M7~M4
⑤X003~X000→M3~M0



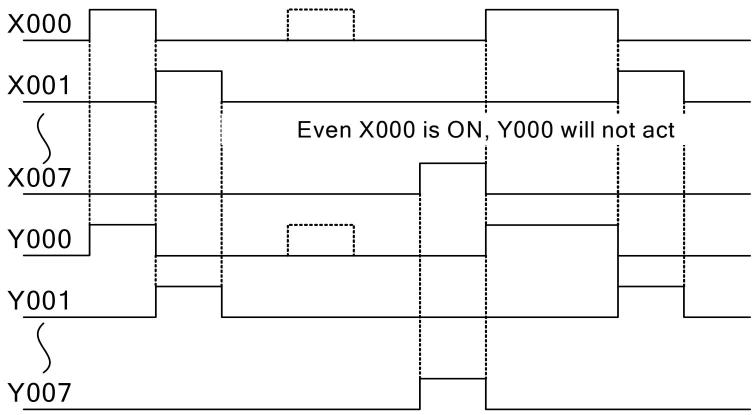
Condition step for 1-bit data



Start data for data input

1-bit of M0 taken as start input

Make 8-character length shifting register of S0~S7



F36 WSFR Word shifting right

F 36		WSFR	P	Word shifting right	S •	D •	n1	n2

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*		
D •								*	*	*	*	*	*	*		
n1					*	*										
n2					*	*										

Instruction format:

—[WSFR S • D • n1 n2]

F37 WSFL Word shifting left

F 37		WSFL	P	Word shifting left	S •	D •	n1	n2

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*		
D •								*	*	*	*	*	*	*		
n1					*	*										
n2					*	*										

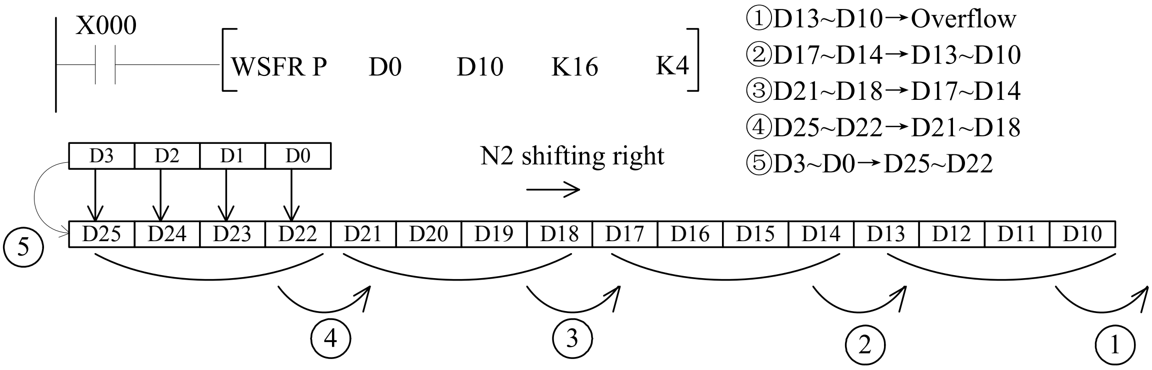
Instruction format:

—[WSFL S • D • n1 n2]

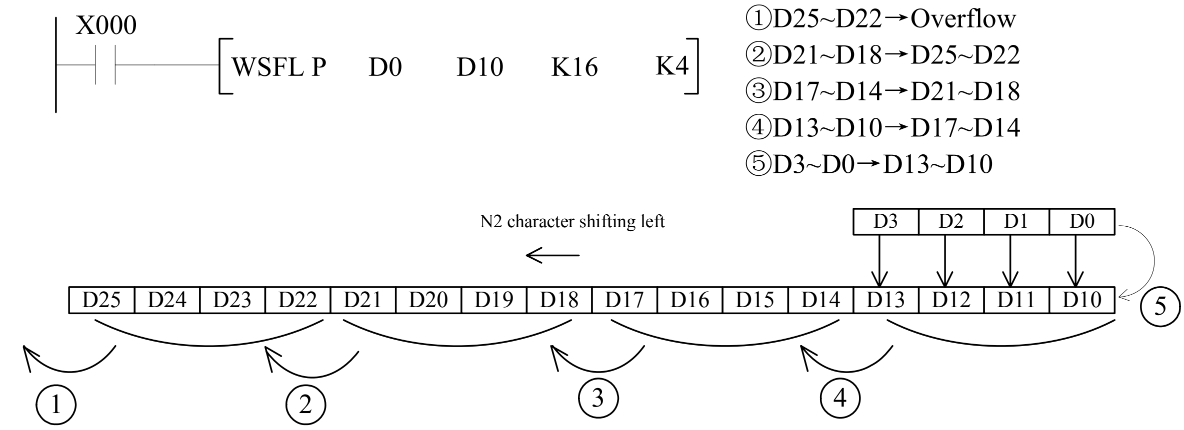
S· : Start number of shifting device
D· : Start number of device to be shifted
n1: Data length to be shifted with word as unit, n1=1~512
n2: Word element for one shift, n2=1~n1

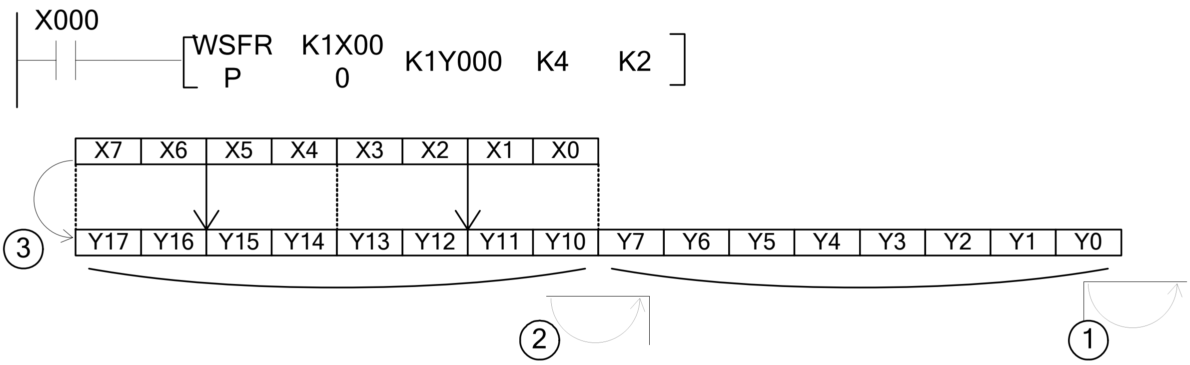
- Data tandem with n1 word element (length of shifting buffer) started with D·, it moves right (left) with n2 shifting number and number started with S· is moved to D· to fill the empty element with n2 word element.
- The instruction is of pulse instruction in general.
- When S·/ D· are specified as bit combined device, the same bit specifying is required.

Word shifting right



Word shifting left





The same bit specifying shall be done for the bit elements.

F38 SFWR Shift write

F 38		SFWR	P	Shift write	S ·	D ·	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·					*	*	*	*	*	*	*	*	*	*	*	*
D ·								*	*	*	*	*	*	*		
n					*	*										

Instruction format:

[SFWR S · D · n]

S· : Source device for shifting writing

D· : Start device for writing destination data tandem

n: Data length to be written, n=2~512 (Actual data length to be written is n-1, the start device D· is taken as the pointer for writing points).

X000 [SFWR D0 D1 K10]

- To control first in first out, the instruction shall be written.
- D shall be reset to 0 in advance.
- When the drive X000 is changed from OFF to ON, content of D0 is written into D2 and content of D1 is changed to 1. When X000 is changed from OFF to ON, content of D0 is stored in D3 and content of D1 is changed to 2. (When continuous executing instruction is used, it will be saved after one scan cycle).
- Content of the pointer D1 is taken as current points of written data. When content of D1 exceeds n-1, it stops executing and the carrying mark M8022 works.

F39 SFRD Shift read

F 39		SFRD	P	Shift read	S •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •								*	*	*	*	*	*	*		
D •								*	*	*	*	*	*	*	*	*
n					*	*										

Instruction format:

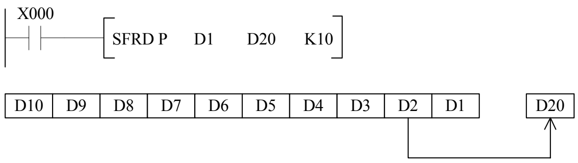
[SFRD S • D • n]

S· : Tandem start device of shifting read data

D· : Stored destination device for data tandem read

n: Data length for shifting read, n=2~512 (Actual length to be read is n-1, the read start device is taken as pointer for judging stopping instruction execution).

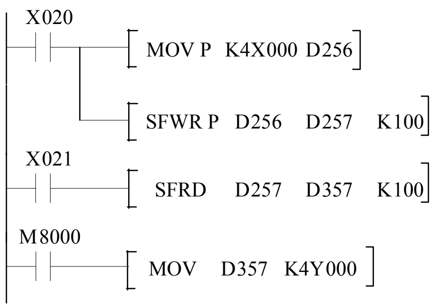
- To control the data first in first out, read the instruction in advance.



- When the drive X000 is ON from ON, content of D2 will be read to D20. At the same time, 1 is subtracted from content of the pointer D1. Data at the left side moves one bit to the right side (content of D10 is not changed). When X000 is ON from OFF, content of D2 is read to D20 and 1 is subtracted fro content of D1. (When continuous executing instruction is used, each scan cycle will execute one reading and shifting).
- When content of the pointer D1 is 0, the instruction stops executing, and the zero point mark M8020 acts.

Shifting reading and writing case for first in and first out control

The product number is logging while warehousing. To ensure first in first out, the case is stated below: The product is 4-bit number of hexadecimal system and the maximum number is below 99 points.



X000~017 are taken as input of product numbers, and moved to D256.

D257 is the index and D258~D356 are the data buffer for 99-point storage product numbers.

As for the numbers out of the warehouse, it is sent to D357.

The product number is represented with 4-bit of hexadecimal and moved to Y000~Y017.

F40~F49 Data processing

Data processing

Function No.	Memory view	Name	Page
40	ZRST	Full reset	1
41	DECO	Decoder	2
42	ENCO	Encoder	3
43	SUM	ON bit quantity	4
44	BON	ON Bit judging	5
45	MEAN	Mean	6
46	ANS	Warning coil setting	7
47	ANR	Warning coil resetting	8
48	SQR	BIN Square root computation	9
49	FLT	BIN Integral→ Binary floating-point	10

F40 ZRST Full reset

F 40		ZRST	P	Full reset	D1 •	D2 •

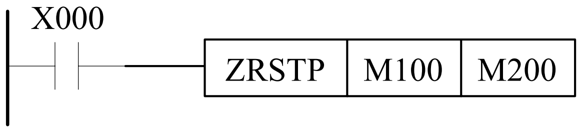
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D1 •		*	*	*							*	*	*	*		
D2 •		*	*	*							*	*	*	*		

Instruction format:

ZRST	D1 •	D2 •

D1·: Start device for all clearing

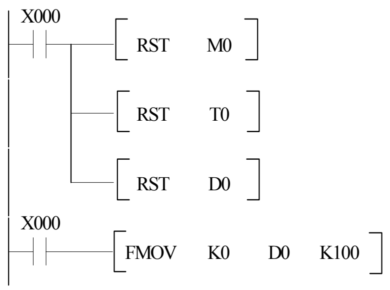
D2·: End device for all clearing, D1· number≤D2· number, and devices of the same type must be specified, otherwise, it may lead to instruction error, D8067=6705.



When X000 is from OFF to ON, M100 to M200 are fully cleared.

- When computation element number of D1·> computation element number of D2·, only the computation element specified by D·1 can be cleared.
- The instruction is executed with 16-bit, however, D1·/D2· can specify 32-bit counter, which can not be specified in a mixed way. For example, D1· is 16-bit counter, and D2· is 32-bit counter.
- The instruction is of pulse instruction in general.

As independent resetting instruction for the device, as for bit elements Y, M and S and word elements T, C and D, the instruction RST can be used. As the instruction F16 FMOV for writing the constant K0 in batches, it can be written into devices KnY,KnM,KnS,T,C,D.



Reset M0

Reset current value of T0

Rest D0

Write D0~D99 into K0

F41 DECO Decoder

F 41		DECO	P	Decoder	S •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •	*	*	*	*	*	*					*	*	*	*	*	*
D •		*	*	*							*	*	*	*		
n					*	*										

Instruction format:

[DECO S • D • n]

S· : Decoding source device

D· : Device for storing decoding results, when D· is bit device, n=1~8; when D is bit device, n=1~4.

n: Decoding bit length, n=1~8, when n=0, it is not processed and it will lead to error out of 0~8.

Low n bit of the source device is taken for decoding and result of 2^n bit length is stored in D·.

The instruction is of pulse instruction in general.

X004 [DECO X000 M0 K3]

X002	X001	X000
0	1	1

M7	M6	M5	M4	M3	M2	M1	M0
0	0	0	0	1	0	0	0

The data source is 1+2=3, the M3 is set 1, and other bits are reset.

D· is bit element, when n=8, 2^8=256 points.

D· is word element, when n=4, 2^8=16 points; when n<4, high bit of D· is used for zero extension.

F42 ENCO Encoder

F 42		ENCO	P	Encoder	S •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •	*	*	*	*							*	*	*	*	*	*
D •											*	*	*	*	*	*
n					*	*										

Instruction format:

[ENCO S • D • n]

S· : code source device

D· : Device for storing code results

n: Coding bit length, when S· is bit device, n=1~8; when S· is word device, n=1~4.

2 exp n bit length of the source device S· is used for coding and the result is stored in D·.

The instruction is pulse instruction in general.

X000

[ENCO M0 D0 K3]

M7	M6	M5	M4	M3	M2	M1	M0
0	0	0	0	1	0	1	0

b15 b0

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

When there are several bits in the data source are 1, 1 at low bit will not be processed. When the data sources are 0, it is error.

S· is bit element, when n =8, 2^8 =256 points.

S· is word element, when n =4, 2^4 =16 points.

F43 SUM ON bit quantity

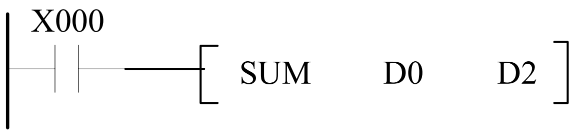
F 43	D	SUM	P	ON bit quantity	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[SUM S • D •]

S· : Source device
D· : Destination device for storing counting values



0	0	0	0	1	1	0	1	0	0	0	1	1	1	1	0	→	7

D0 → D2

- If 16 bits in D0 are 0, the zero mark M8020 will act.
- When 32-bit instruction is used, D· still occupies 2 buffers. See the figure above, number of 1 of 32-bit of (D1,D0)is written into D2 and D3 becomes 0.

F44 BON ON Bit judging

F 44	D	BON	P	ON Bit judging	S •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •		*	*	*												
n					*	*										

Instruction format:

[BON S • D • n]

S· : Source device

D· : Device for storing judging results

n: When specify judging bit, n=0~15 (16-bit instruction), n=0~31 (32-bit instruction).

When n^{th} bit of S· is 1, set D· at 1; for 0, set D· at 0.

Example:

X000

[BON D10 M0 K15]

b15 b0

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

D10 → M0=OFF

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

D10 → M0=ON

When 16-bit computation is executed, n=0~15; for 32-bit computation, n=0~31.

F45 MEAN Mean

F 45	D	MEAN	P	Mean	S •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*		
D •								*	*	*	*	*	*	*	*	*
n					*	*										

Instruction format:

[MEAN S • D • n]

S· : Source device
D· : Destination device for storing the mean
n: Specify number for mean, n=1~64

X000

[MEAN D100 D150 K5]

（D100+D101+D102+D103+D104）/ 5 ⟶ D150

- Mean (the algebraic sum is divided by n) of point n is stored in the destination address and the residual is ignored. If it exceeds the device number, minimum value of n is obtained in possible range.
- If n is out of 1~64, it may lead to error.

When range specified by S· exceeds n, the instruction will calculate the mean in effective range.

Example:

X000

[MEAN D8510 D0 K3]

（D8510+D8511）/ 2 ⟶ D0

F46 ANS Warning coil setting

F 46		ANS		Warning coil setting	S •	m	D •

	Bit element					Word element										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •											*					
m					*	*										
D •				*												

Instruction format:

[ANS S • m D •]

S· : Detecting alarm timer, T0~T199 can be used only.
m : Timing time, specify m=1~32,767 (Unit 100ms).
D· : Alarm point device, S900~S999 is available for D.
※ 1: For TP03SR machine type, S· only supports T0~39 and T196~T199.
It is the convenient instruction for driving signal alarm period.

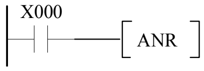
X000
[ANS T0 K10 S900]

- If X000 is active for 1 second, S900 is set. Even X000 is OFF (the timer is reset), S900 keeps acting.
  If it is less than 1 second, X000 or X001 is OFF, the timer resets.
- Preset M8049 (available signal alarm) ON, minimum number of the signal alarm S900~S999 ON is stored in D8049. In addition, if any of S900~S999, M8048 (the alarm acts) is ON.

F47 ANR Warning coil resetting

F		ANR		Warning coil resetting	No corresponding devices
47	D		P		

Instruction format:



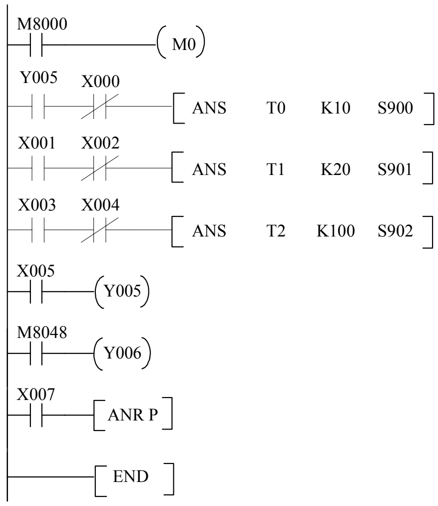
If X000 is active, the acting points of the signal alarm S900~S999 are reset.

If there are several alarm points simultaneously, the alarm point with minimum number will be reset. At this time, if the effective mark M8049 of the signal alarm is ON, content of the register D8049 will be updated timely and the minimum number of the alarm points will be left.

If X003 is active a second time, state of the next number will be reset.

If the instruction ANRP is used, it is reset in each scan cycle according to the sequence and pay attention to it.

The following figure describes the external failure diagnosis circuit. The special data buffer D8049 is used to monitor the minimum number of state numbers of S900~S999. In case of several failures, after the failure with minimum number is released, the next failure number is displayed.



- When the special auxiliary relay M8049 acts, monitoring can be performed.
- When the forward output Y005 is driven, the forward is detecting. If X000 fails to act within 1 second, S900 acts.
- In case of abnormality of DOG, if the upper limits X001 and X002 fail to act simultaneously over 2 seconds, S901 acts.
- When T2 is less than 10 seconds and the continuous operation input point X003 is ON, during 1 cycle, if the action switch X004 does not act, S902 acts.
- When there is ON between S900~S999, the special auxiliary relay M8048 acts, the failure represents the output Y006 acts.
- Before acting of the external failure diagnosis program, the state clearing key X007 is set OFF; when X007 is ON, the minimum number will be cleared in sequence.

<Alarm coil effective M8049>

When M8049 is driven, minimum number of acting state of S900~S999 is stored in D8049.

<Alarm coil acting M8048>

When M8049 is driven, in case of acting between S900~S999, M8048 will act.

F48 SQR BIN Square root computation

F 48	D	SQR	P	BIN Square root computation	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*							*	*		
D •													*	*		

Instruction format:

[SQR S • D •]

S· : Data source for square root

D· : Destination device for storing results

Square root is executed for data in S· and the data is stored in the device specified by D·.

Example:

X000

[SQR D0 D1]

When X000 is ON, after square root is executed for data in D0, the result is stored in D1.

- S· is valid for non-negative. If it is negative, it may lead to error, the mark M8067 is ON, and the instruction will not be executed.
- The computation result D· is integral, the decimal is ignored, and the borrowing mark signal M8021 is ON.
- When the computation result is 0, the zero mark M8020 is ON.

F49 FLT BIN Integral→ Binary floating-point conversion

F		FLT		BIN Integral→ Binary floating-point conversion	S •	D •
49	D		P			

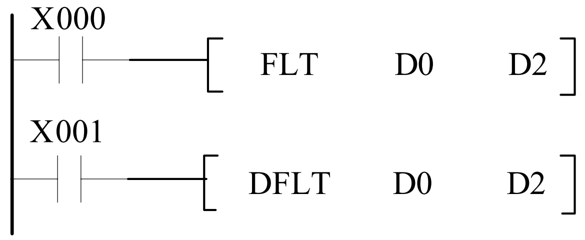
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •													*	*		
D •													*	*		

Instruction format:

[FLT S • D •]

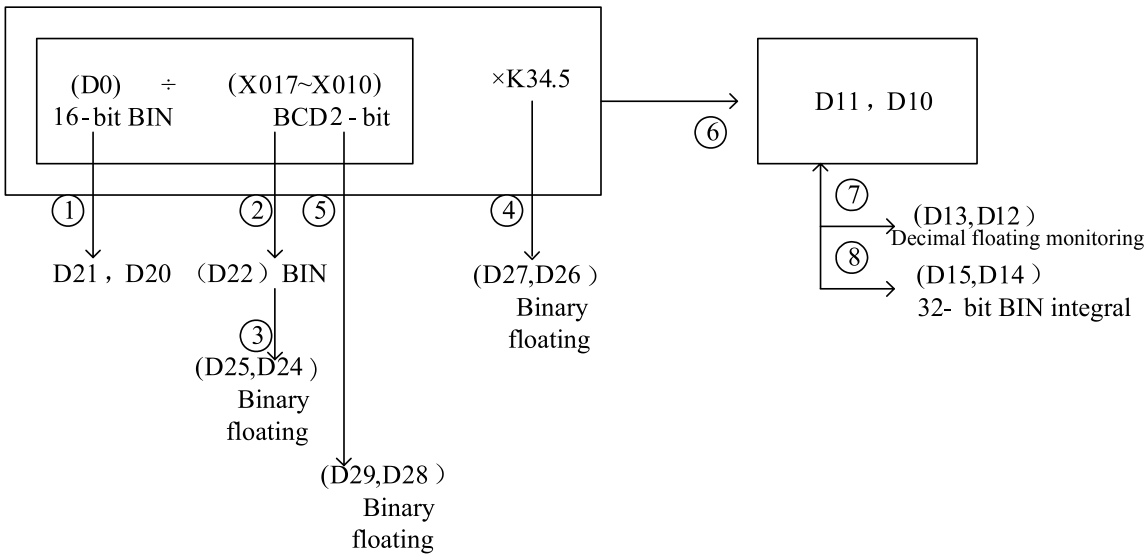
S· : Data source for conversion from BIN integral→ Binary floating-point numbers 2
D· : Destination device for storing results

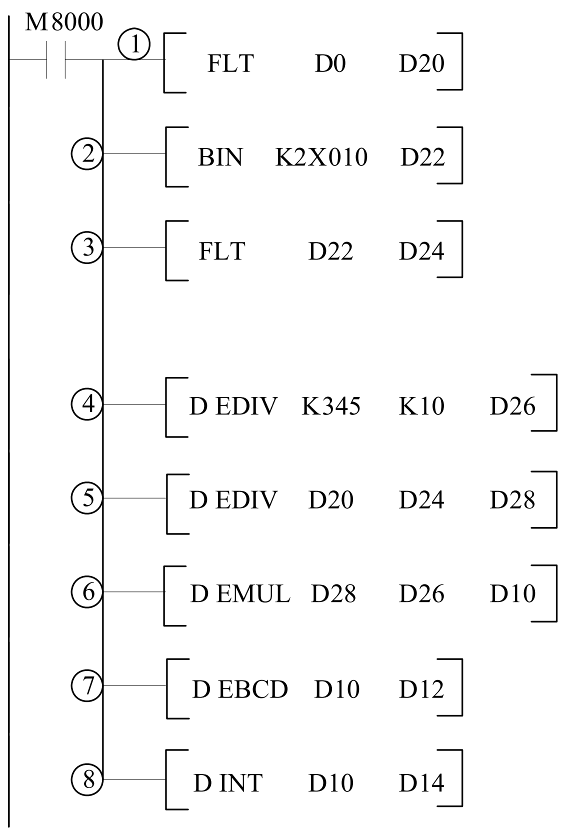
Example:



- Conversion instruction for BIN integral and binary floating-point values
  The constants K and H are automatically converted, and the instruction FLT can not be used.
- Reverse conversion instruction for the instruction is FNC129 (INT).

See the following sequential control on the floating-point computation case.


(D0) ÷ (X017~X010) ×K34.5
16-bit BIN
BCD2-bit
D11，D10
D21，D20
(D22) BIN
(D25,D24)
Binary floating
(D29,D28)
Binary floating
(D27,D26)
Binary floating
(D13,D12)
Decimal floating monitoring
(D15,D14)
32- bit BIN integral


M8000
FLT D0 D20
BIN K2X010 D22
FLT D22 D24
D EDIV K345 K10 D26
D EDIV D20 D24 D28
D EMUL D28 D26 D10
D EBCD D10 D12
D INT D10 D14

F50~F59 High speed counting processing

High speed counting processing

F50 REF Refreshing of input and output

F 50		REF	P	Refreshing of input and output	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •	*	*														
n					*	*										

Instruction format:

[REF D • n]

D· ：Start device number for input and output refreshing

n： Refreshing points

For X, X0~X17 can be refreshed, 16 points input in total

For Y, Y0~Y7 can be refreshed, 8 points output in total

※ 1：For TP03-14SR-A :

X can only refresh X0~X7, 8 points input in total

Y can only refresh Y0~Y5， 8 points output in total, when the refreshing point n is equal to 8, Y0~Y5 can be refreshed.

※ 2：For TP03-20SR-A

X can only refresh X0~X13, 12 points in total, when the refreshing point n is equal to 16, and refreshing starting from X0, and X0~X13 can be refreshed only. When the refreshing point n is equal to 8, the refreshing can be started from X10, and X10-X13 can be refreshed actually.

Y can only refresh Y0~Y7, 8 points output in total.

PLC employs input and output refreshing. The input terminal information is stored in input image storage zone before step 0 computation. After the output terminal executes the instruction END, it is output through the latching storage device from the output image storage zone. However, during computation, if latest input information and immediate output computation are expected, the input and output refreshing instruction can be used.

Example 1: input refreshing:

X000

[REF X010 K8] Refreshing, X010~X017, 8 points in total

Example 2: output refreshing:

X000

[REF Y000 K8] Refreshing, Y000~Y007, 8 points in total

- When specifying the start device number D·, set the lowest bit number 0, like X000, X010, Y000 and etc.
- The refreshing point n shall be 8 or 16 points, otherwise, it may lead to error.
- In general, the REF instruction can be used among the instructions FOR~NEXT and the instruction CJ.
- In the interruption processing with input and output actions, latest input information and timely output computation can be obtained by executing the instruction.

F52 MTR Matrix input

F 52	MTR	Matrix input	S •	D1 •	D2 •	n

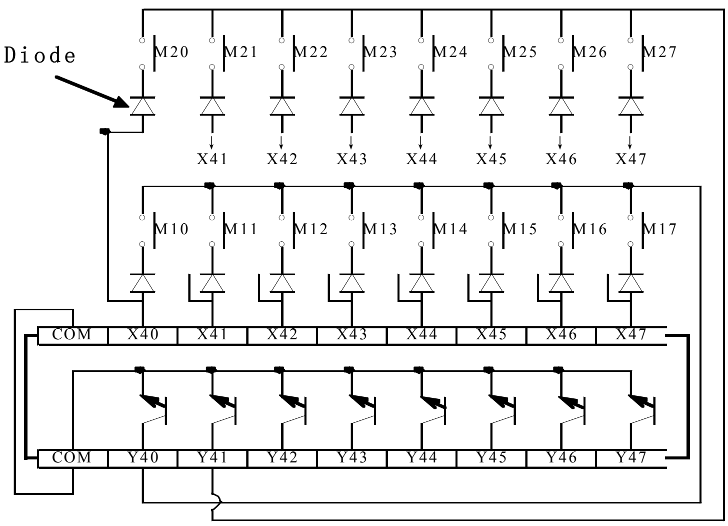
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •	*															
D1 •		*														
D2 •		*	*	*												
n					*	*										

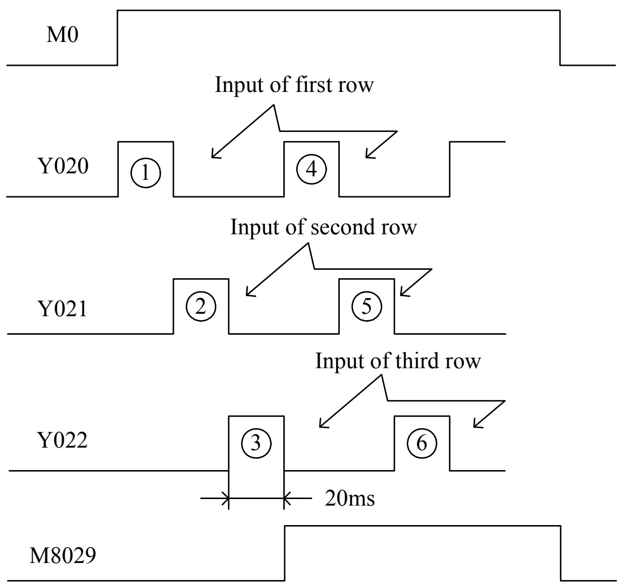
Instruction format:

MTR S • D1 • D2 • n

S· ：Start device for matrix scan input, the rightmost number shall be 0, such as X000, X010 and etc. and occupies continuous 8 points.
D1· ：Start device for matrix scan output, the rightmost number shall be 0.
D2· ：Start device for matrix scan value storage, the rightmost number shall be 0.
n：row of matrix scan, n=2~8 is valid.
The instruction uses 8 points input and n points output, and reads n rows of 8 point input signal instruction.
Example:

X010 MTR X040 Y040 M10 K3

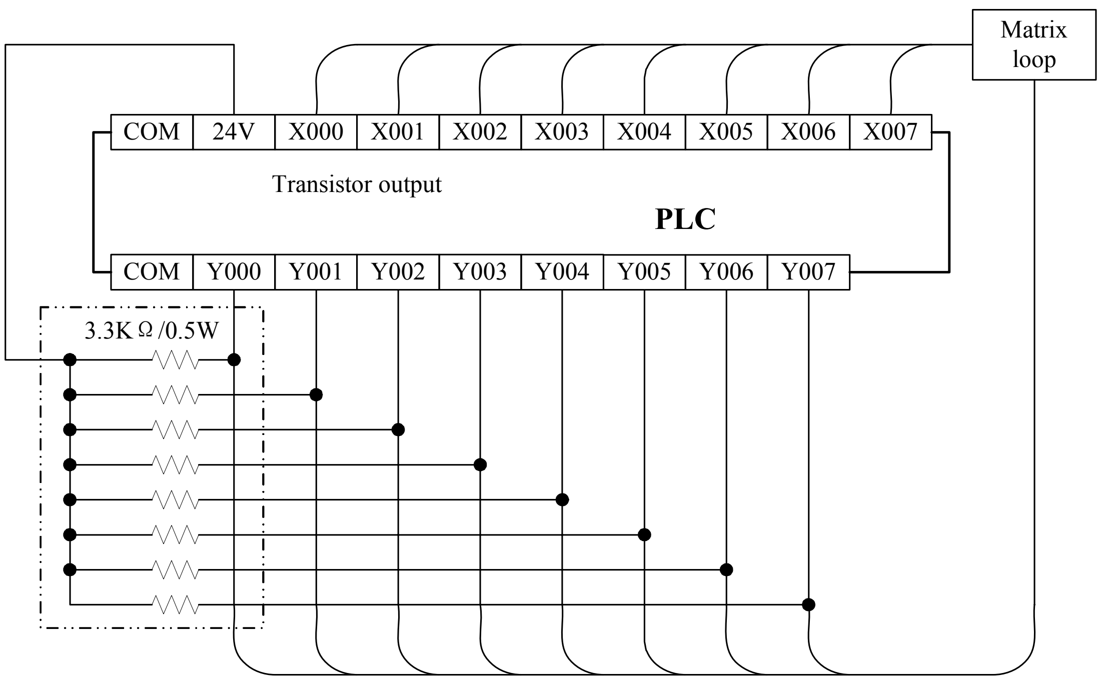




- Start with specified input S·, occupying 8 points input.
- Start with specified output D1·, occupying 3 points output.
- The figure is n=3 points outputs Y040, Y041 and Y042 repeatedly ON. Each repeat will obtain the first row, second row and third row inputs, which are stored in M10~M17, M20~M27 and M30~M37.
- The output is interrupted with an interval of 20ms for instant input and output processing.

Input numbers for the instruction MTR

1. Input number of the instruction MTR shall be used after X020. (16 点基本形为 X010 以后)
2. When the instruction MTR is used, the transistor output shall be connected to the resistor (3.3K/0.5W).



F53 HSCS Comparison setting

F 53	D	HSCS		Comparison setting	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*		*
S2 •												*				
D •		*	*	*												

Instruction format:

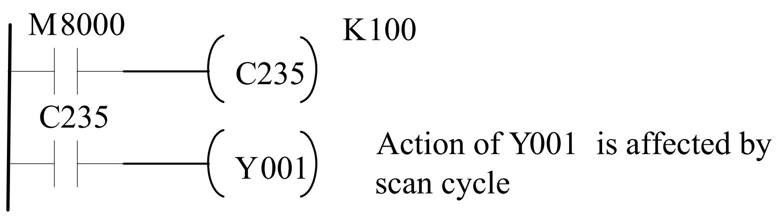
DHSCS S1 • S2 • D •

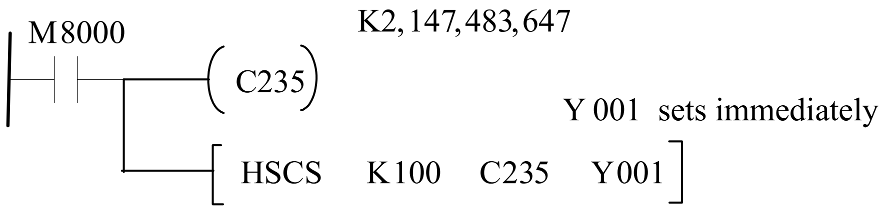
S1· ：Comparison setting

S2· ：Number of high speed counter, high speed counters C235~C249，C251~C254 must be specified.

D· ：Comparison results, and D· can specify interruption indicator I010~I060

Example:



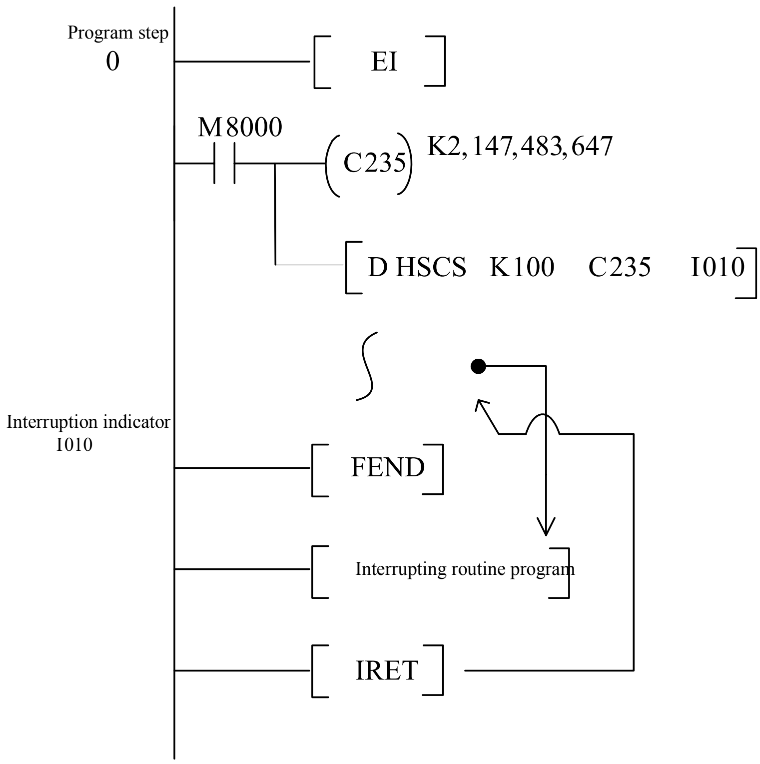


- The high speed counter works according to input OFF→ON in interruption mode. When current value of the counter is equal to the setting value, output contact-joint of the counter works immediately. If the instruction FNC53 is not used, the external output has something to do with the sequential control, so it is affected by scan cycle and it outputs after the END processing.
- The instruction FNC53 can be used for interrupting processing comparison and external output, therefore, when current value of C235 is becoming 99→100 or 101→100, Y001 sets immediately.

Attentions

- The instruction is 32-bit special instruction, which must be input as the instruction DHSCS.
- These instructions can compare results and actions in case of pulse input, therefore, even current value is changed by sending instruction, if there is no counting input, the comparison output will not change.
- F53, F54, F55 and other instructions can be used repeatedly, however, number of simultaneous driving of the instructions must be less than 6.
- If the instructions DHSCS, DHSCR, DHSZ and etc. are used, maximum permissible frequency of high speed controller of the PLC will be affected greatly.

Counting interruption



- D· of the instruction DHSCS can be specified as the interruption indicator I010~I060. (The number can not be used repeatedly)
- When current value of the high speed counter specified by S2· is changed into specified value of S1·, interruption program of specified mark by D· is executed.
- When special auxiliary relay M8059=ON, interruptions of I010~I060 are inhibited.

Regarding details of interruption processing, please refer to FNC03（IRET）~FNC05（DI）.

F54 HSCR Comparison resetting

F 54	D	HSCR		Comparison resetting	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*		*
S2 •												*				
D •		*	*	*								*				

Instruction format:

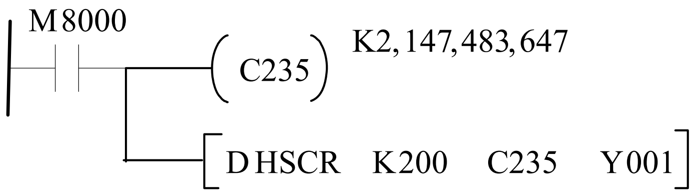
[DHSCR S1 • S2 • D •]

S1· ： Comparison value

S2· ： Number of high speed counter, the high speed counters C235~C249，C251~C254 must be specified.

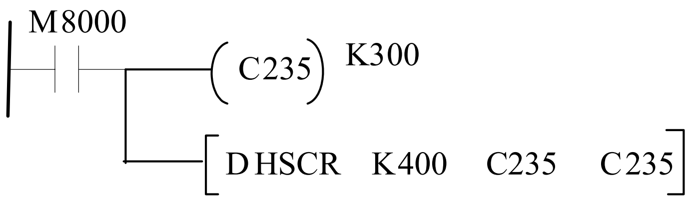
D· ： Comparison results, D· can specify number of the high speed counter with S2·.

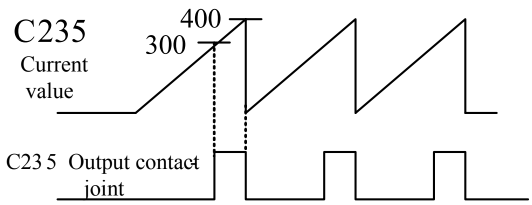
Example



- If the instruction F54 is used, for comparison and external output employ interruption processing, current values of C235 is becoming 199→200 or 201→200, which is not affected by scan cycle, Y001 immediately resets. Regarding effect of the scan cycle, please refer to the above FNC53.

Example of automatic reset loop





- When current value of C235 becomes 400, C235 resets immediately, the current value is 0 and the output contact does not work.
- If the instruction is 32-bit instruction, it must be used as input of the instruction DHSCR. Please refer to Attentions in FNC53 for other attentions.

F55 HSZ Inter-zone comparison

F 55	D	HSZ		Inter-zone comparison	S1 •	S2 •	S •	D •

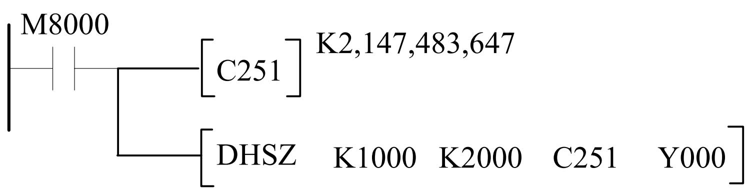
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*		*
S2 •					*	*	*	*	*	*	*	*	*	*		*
S •												*				
D •		*	*	*												

Instruction format:

[DHSZ S1 • S2 • S • D •]

S1· ：Lower limit value of zone comparison
S2· ：Upper limit value of zone comparison, S1·≦S2·
S· ：Number of high speed counter, high speed counter C235~C249，C251~C254 must be specified
D· ：Comparison results, occupying continuous 3 devices.

For instance,



Actions of comparison output:

K1000＞C251Current value	Y000 ON
K1000≦C251Current value≦K2000	Y001 ON
K1000＜C251Current value	Y002 ON

- Content of S1· and S2· is S1·≦S2·
- If the instruction FNC55 is used, interruption processing is executed for comparison and external output, which is not affected by scan cycle.
- When the instruction is enabled, output in the first scan cycle is output according to comparison result of current value of S· and S1· and S2·.
- If the instruction is 32-bit instruction, it must be used as input of the instruction DHSCR. Please refer to Attentions in FNC53 for other attentions.

F56 SPD pulse density

F 56		SPD		pulse density	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •	*															
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •											*	*	*	*	*	*

Number of instruction: 1 instruction for each input (decorated by index register)

Instruction format:

[SPD S1 • S2 • D •]

S1· ：External pulse input terminal, TP03 M/H machine type S1· can only specify X0~X5，TP03SR machine type can only specify X0~X3

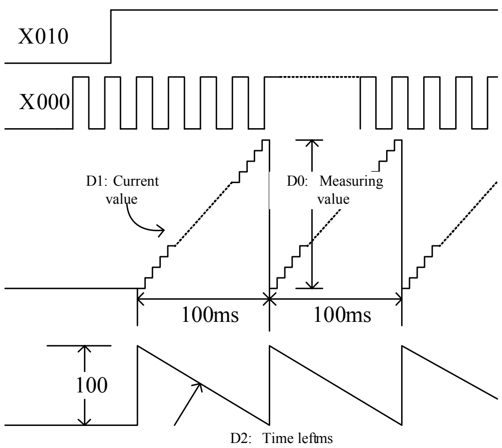
S2· ：Time of receiving pulse (the unit is ms)

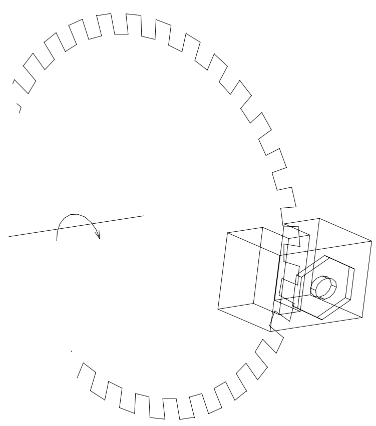
D· ： Result location, occupying continuous 3 devices.

Example:

X010

[SPD X000 K100 D0]





- Input pulse specified by S1· counts in specified time (ms) by S2·, and the result is stored in D· specified device.
- Through repeated operation, pulse density (the proportional value with rotating speed) in D· is obtained. D· occupies 3 points of devices.
- In the figure, when X010 is ON, D1 counts OFF—>ON of X000. After 100ms, the results are stored in D0. Then D1 resets, it counts actions of X000 a second time.
- D2 is used to measure time left.
- The specified input X000~X005 herein can not be used repeatedly with the high speed counter and interruption input.
- Maximum frequency of ON/OFF of input X000-X005 has the same processing with 1-phase high speed counting. When it is used high speed counting, instructions FNC57（PLSY）and FNC59（PLSR）, sum of the processing frequency shall be less than the specified frequency.

F 57 PLSY pulse output

F 57	D	PLSY		pulse output	S1 •	S2 •	D •

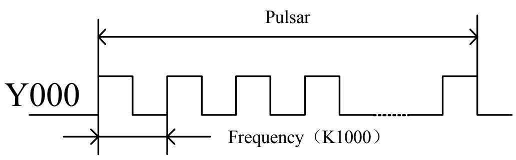
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •		*														

Instruction format:

[PLSY S1 • S2 • D •]

S1· ：Pulse output frequency
16-bit instruction: 1~32,767Hz
32-bit instruction: 1~100,000Hz
S2· ：Pulse output number
16-bit instruction: 1~32,767
32-bit instruction: 1~2,147,483,647
D· ：Pulse output device, specifying Y000 and Y001（Controller with transistor output shall be used）
※ 1：TP03SR machine type does not support the instruction.
Example:

X000
[PLSY K100 K1000 Y000]



- Specified frequency is used to generate fixed pulse frequency. During the instruction executing, if content of the word device specified by S1· is changed, the output frequency will vary.
- When value of S2· is specified 0, there is no restriction on created pulse. During the instruction executing, if specified word device by S2· is changed, it will execute the changed content from the next instruction drive.
- After X010 is OFF, the output stops; when it is ON a second time, it acts from the initial state, when it makes continuous beeper, X010 is OFF, and Y000 is also OFF.
- Duty ratio of the pulse is 0.5. The output control is not affected by scan cycle and interruption processing is employed.

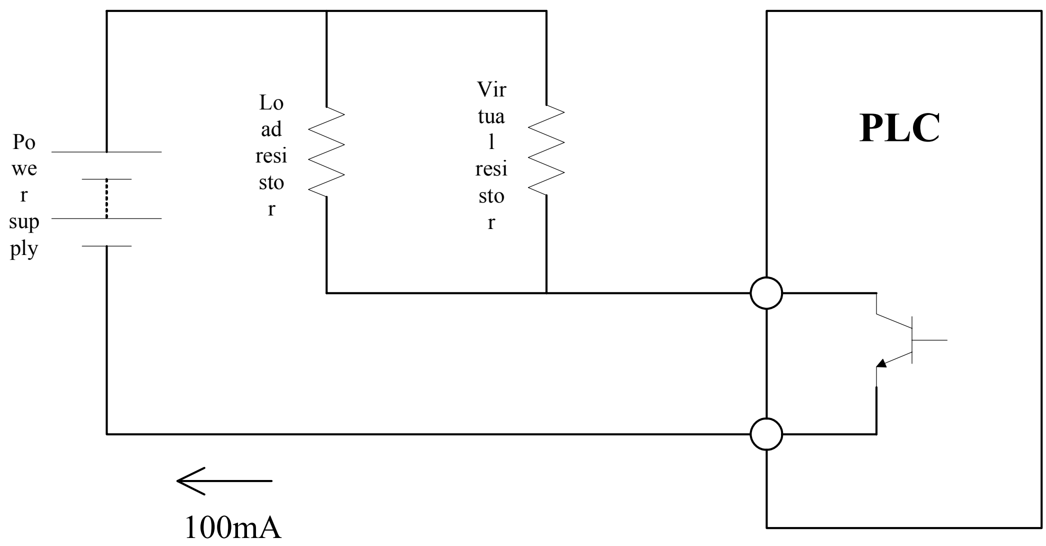
- After setting pulse finishes, the finish symbol M8029 acts.
- Controller with transistor output shall be employed.
- When programming 2 FNC57 (PLSY) instructions or 2 FNC59 (PLSR) instructions, independent pulse output can be done for Y000 and Y001.
- When programming 1 FNC57 (PLSY) instruction and 1 FNC59 (PLSR) instruction, independent pulse output can be done for Y000 and Y001.

<Attentions>

1. Do not drive the pulse output instruction using the same output relay (Y000 or Y001) simultaneously. If it is driven simultaneously, it leads to use of dual-coil and it can not operate with normal performance.
2. After drive contact-point of the pulse output instruction is OFF, a second drive shall be executed after the following conditions are established.

Conditions: after pulse output monitor Y000: [M8147] and Y001:[M8148]) of the previous drive is OFF, it can be driven after one calculation performance cycle.

For additional drive of the pulse output instruction required more than 1 calculation performance for OFF, if the drive is earlier than the conditions, the initial instruction performance may cause normal calculation performance, when it is driven by the 2nd instruction, it starts pulse output.



- It can not be repeated with specified output number by the instruction FNC58 (PWM).
- As for specified output numbers by the instruction FNC58 (PWM), it can not be repeated.

F58 PWM Pulse width modulating

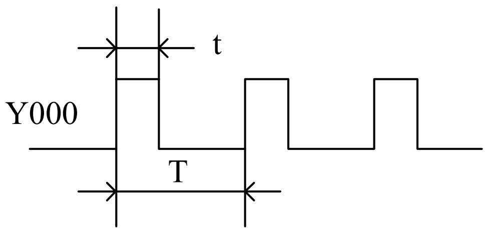
F 58		PWM		Pulse width modulating	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •		*														

Instruction format:

[PWM S1 • S2 • D •]

S1· ：Pulse output width, t=0~3000ms
S2· ：Pulse output cycle, T= 1~3000ms, but S1·≤S2·
D· ：Pulse output device, only specifying Y000 and Y001 (Please use controller with transistor output)
※ 1：TP03SR machine type does not support the instruction.
Example:



- The output ON/OFF can execute interruption processing.
- In the above case, when content of D10 exceeds 50, it may be error.
- When X010 is OFF, Y000 is OFF.

Attentions：
- Transistor output shall be used for PLC. For high frequency pulse output, as mentioned above, please offer load current. （FNC57（PLSY））
- Specified output number by FNC57（PLSY）or FNC59（PLSR）can not be used repeatedly.

F59 PLSR Pulse output with acceleration and deceleration

F 59	D	PLSR		Pulse output with acceleration and deceleration	S1 •	S2 •	S3 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
S3 •					*	*	*	*	*	*	*	*	*	*	*	*
D •		*														

Instruction format:

[PLSR S1 • S2 • S3 • D •]

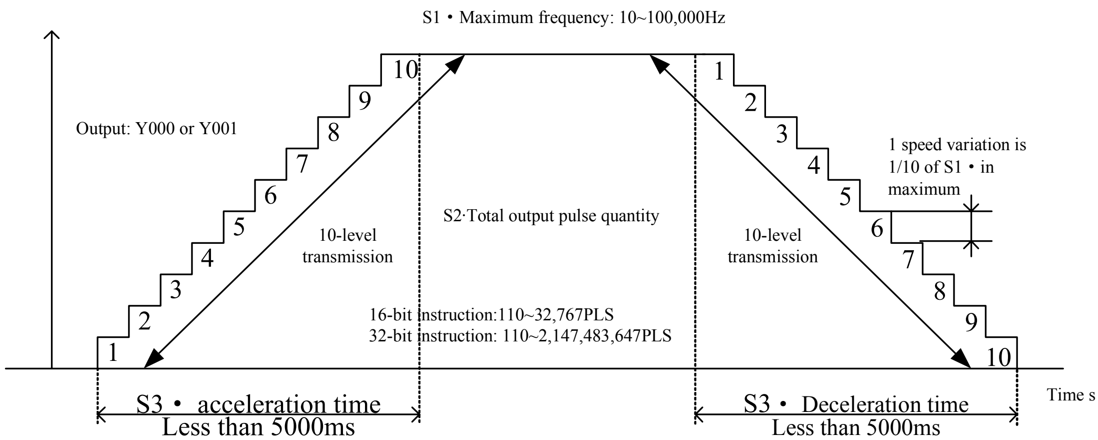
S1· ： Maximum frequency of pulse output
S2· ： Total quantity of output pulse
S3· ： Time of acceleration and deceleration ms
D· ：Pulse output device, only specifying Y000 and Y001(Please use controller with transistor output)
※ 1：TP03SR machine type does not support the instruction.
Example:

X010

	PLSR	K500	D0	K3600	Y000
		Maximum frequency (HZ)	**Total pulse output quantity (PLS)**	**Time of acceleration and deceleration (ms)**	**Output number (Y000 or Y001)**

- Pulse output instruction with functions of acceleration and deceleration with fixed sizes.

For specified maximum frequency, fixed acceleration is executed; after it reaches specified output pulse quantity, fixed deceleration is executed.



- Setting content of the operands:

S1· maximum frequency

Range of setting: 10~100,000（Hz）

The frequency is set with multiple of 10.

1/10 of specified value of maximum frequency can be set as one speed variation (frequency) during deceleration, therefore, please set the step motor in specified range.

S2· Total output pulse quantity

Range of setting: 16-bit computation: 110~32,767（PLS）

32-bit computation: 110~2147483647（PLS）

When it is less than 110, the pulse can not output normally.

When the instruction DPLSR is used, (D1 and D0) are used as 32-bit setting value.

S3· Time of acceleration and deceleration

Range of setting: less than 5000(ms), please comply with conditions of①~③.

Time of acceleration and deceleration shall act with the same values.

①Maximum time of scan time of possible PLC of acceleration and deceleration must be over 10 times, when it is less than 10 times, time sequence of acceleration and deceleration is uncertain.

②Formula of minimum time for acceleration and deceleration

$$S_3 \geq \frac{90000}{S_1} \times 5$$

③Formula of maximum time for acceleration and deceleration

$$S_3 \leq \frac{S_2}{S_1} \times 818$$

D· Pulse output numbers:

· Only specify Y000 or Y001.

· The output is transistor output.

- Output frequency of the instruction is 10~100,000Hz, when maximum speed and variable speed of acceleration and deceleration exceed the scope, it is lowered or carried in the range automatically.
- The output control is not affected by scan cycle for interruption processing.
- When X010 is OFF, output is interrupted. When it is set ON a second time, it acts from the initial position.
- During the instruction executing, even the operand is rewritten, the operation will not change. The content of change will take effect in the next instruction drive.
- When setting pulse output finishes, the finishing mark M8029 is set ON.

F60~F69 Convenient instructions

Convenient instructions

Function No.	Memory view	Name	Page
F60	IST	State initialization	1
F61	SER	Data searching	6
F62	ABSD	Control mode of cam control	8
F63	INCD	Cam control increment mode	10
F64	TTMR	Demonstrating teaching timer	12
F65	STMR	Special timer	13
F66	ALT	ON/OFF Alternative output	14
F67	RAMP	Slope signal	16
F68	ROTC	Rotating working bench control	18
F69	SORT	Data sort	19

F60 IST State initialization

F 60	IST	State initialization	S •	D1 •	D2 •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •	*	*	*													
D1 •				*												
D2 •				*												

Instruction format:

——[IST S • D1 • D2 •]

S • : Start input of operation mode
D1 • : Minimum serial number for state step point under automatic mode
D2 • : Maximum serial number for state step point under automatic mode
D1 • /D2 • Specified range of computation element: S20~S1023, and D2>D1.
The instruction IST is a specified convenient instruction for initializing state of step ladder control flow. For coordinating special relay, it forms convenient automatic control.

Example:

X000
—| |——[IST X030 S20 S20]
RUN
mnitoring

X030：Manual operation	X034：Continuous operation
X031：Origin return	X035：Origin return starts
X032：Step	X036：Automatic start
X033：One cycle	X037：Stop

When driving the instruction, the special relays will switch automatically, if the drive input is OFF, it will not change.

M8040：Transfer prohibited	S0：Initial state for manual operation
M8041：Transfer start	S1：Initial state for origin return
M8042：Start pulse	S2：Initial state for automatic operation
M8047：STL	

If the instruction is used, S10~S19 are used for origin return. Therefore, during programming, do not take these states as common states. In addition, when S0~S9 are taken as step points, S0~S2 are for the above manual operation, origin return and automatic operation. In the program, the step point circuits for the 3 states must be written, and S3~S9 can be used freely.
The instruction shall be programmed in priority than the states S0~S2.

To avoid the above X030~X034 are ON simultaneously, rotating switch must be used.
When origin return finishes（M8043）and there is no action, manual operation (X030) shall prevail.
If the origin returns （X031）and switches among（X032，X033，X034）, all the inputs and outputs are OFF. The automatic operation can be driven a second time after origin return finishes.

Special auxiliary relay for the instruction IST
The auxiliary relay for the instruction IST can be divided into automatic operation as per its state and program control for operation preparation and control purpose.
Automatic control for the instruction IST
Transfer prohibited M8040
When the auxiliary relay works, all the state transfers are prohibited.
Schlep: M8040 works frequently.
Recovery, one cycle, it keeps operation from pressing the stop button and pressing the start button.
Single step: M8040 works frequently. After pressing the start button, it becomes inactive and starts sending.
Other: for switching from PLC STOP→RUN, press the start button to release.

Transfer starts: M8041
The auxiliary relay for the sending conditions from the initial state S2 to the next state
Manual and recovery: do not act.
One single step and cycle: it acts when pressing the start button.
When pressing the start button continuously, it keeps acting; when pressing the stop button, it is released.

Start pulse: M8042
It acts instantly only pressing the start button.
Origin finishes M8043
In recovery mode, after the machine returns to the origin, the user makes the special relay with program.
Origin conditions M8044
Test origin conditions of the machine, and the special relay is driven. The full modes become effective signals.
All the output resets prohibited M8045
During switching among manual, recovering and automatic mode, if the machine is not at the origin location, all the outputs and action states shall reset. If M8045 is driven, only the action states need to reset.
STL monitoring effective M8047
After driving M8047, the state numbers S0~S899 in acting are stored in the special auxiliary relays D8040~D8047 from small to big. Therefore, it monitors 8 action state numbers. Besides, if any of the states acts, the special auxiliary relay M8046 will act.

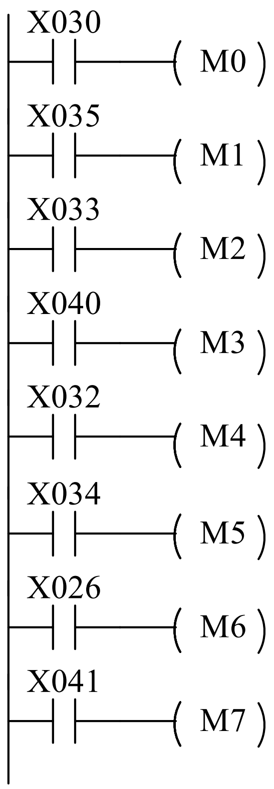
Distribution of mode selection input

If the instruction IST is used, input of the modes is described in the following distributed continuous input numbers. When discontinuous number and part of them are ignored, the following auxiliary relays can be used to specify the input initial element numbers.

X030：Individual operation
X031：Origin return
X032：Step
X033：One operation
X034：Continuous operation
X035：Origin return starts
X036：Automatic start
X037：Stop
X030~X034 do not act simultaneously (Selection switch is utilized)

Input is discontinuous number

Example: X030：individual operation
X035：Origin return
X033：Step
X040：One operation
X032：Continuous operation
X034：Origin return starts
X026：Automatic start
X041：Stop



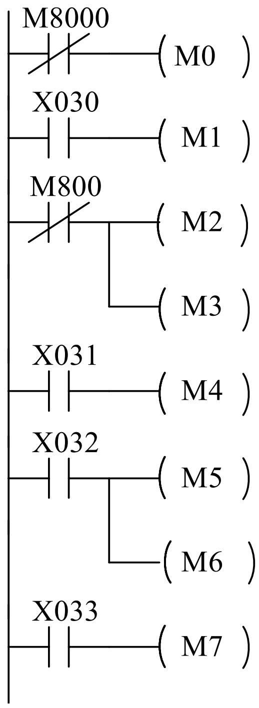
Only continuous/origin return mode

Example: X030：Origin return

X031：Continuous operation

X032：Automatic switch and origin return start

X033：Stop



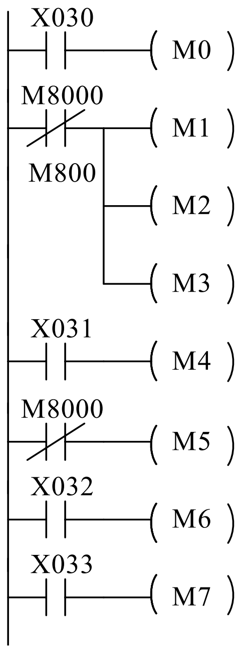
Only continuous/individual mode

Example: X030：individual operation

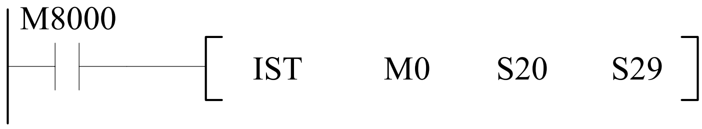
X031：Continuous operation

X032：Automatic switch

X033：Stop



In this case, M0 is taken as mode specified start input.



The auxiliary relay for the instruction IST is divided into self-generated action and program for operation preparation and control purpose.

Automatic control for the instruction IST

<Transition prohibited M8040>
When the auxiliary relay acts, all the state transitions are prohibited.
Individual: M8040 acts frequently.
One recovery: it keeps acting from pressing the stop button to the start button.
Individual: M8040 acts frequently, when the start button is pressed, it does not act, and transition is performed.
Other: it keeps acting during switching from STOP—RUN of the PLC. When the button is pressed, it is released. For transition prohibited state, output in the state continues holding.
<Transition start M8041>
The auxiliary relay for the initial state S2 transitioned to the next state
Individual and recovery: do not act.
Step, one time: it acts when pressing the start button.
Continuous: after pressing the start button, it keeps acting; after pressing the stop button, it is released.
<Start pulse M8042>
When the starting button is pressed, it acts instantaneously.
Regarding the control, refer to the next page.

For sequential control program

<Recovery finishing M8043>
After the recovery mode finishes at the origin, the user shall make the special auxiliary relay act with program.
<Origin conditions M8044>
The mechanical origin conditions are detected to drive the special auxiliary relay and the full mode is effective signal.
<All outputs cleared and prohibited M8045>
During switching of individual, recovery and automatic mode, when the machine is not at the origin, all the outputs and action states are cleared. When M8045 is driven, only the action states are cleared.
<STL monitoring effective M8047>
When M8047 is driven, the current state numbers (S0~S899) will be sequenced with smaller numbers, which are stored in D8040~D8047 and monitor 8 action states. In case of any action of the states, the special auxiliary relay M8046 will act.
The following out of the PLC circuit is control content of the fixed circuit, which shall be coordinated with other programs.

F61 SER Data searching

F 61	D	SER	P	Data searching	S1 •	S2 •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •							*	*	*	*	*	*	*	*		
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*		
n					*	*							*	*		

Instruction format:

[SER S1 • S2 • D • n]

S1 • ：Start element number for data searching zone

S2 • ：Data content searched by the instruction

D • ：Start device for storing checking list, occupying continuous 5 points

n：Length of data zone to be compared, n=1~256（16-bit instruction），n=1~128（32-bit instruction）.

The instruction is used for searching the same data, maximum value and minimum value in the data sheet.

Example:

X000 [SER D100 D0 D50 K10]

Structure and data case of search list:

Searched element	Case of searched data	Comparison data	Data location	Maximum	Sum	Minimum
D100	D100=K100	D0=K100	0			
D101	D101=K111		1			
D102	D102=K100		2		Same	
D103	D103=K98		3			
D104	D104=K23		4			Minimum
D105	D105=K66		5			
D106	D106=K100		6		Same	
D107	D107=K95		7			
D108	D108=K210		8	Maximum		
D109	D109=K88		9			

Check result sheet

Element No.	Content	Remark
D50	3	Number of the same data
D51	0	Location of the same data (Initial)
D52	6	Location of the same data (Final)
D53	4	Final location of minimum
D54	8	Final location of maximum

- It is algebraic comparison, namely, comparison with positive or negative symbols.
- When there are several maximum and minimum values, the back location shall display.
- When the 32-bit instruction is used, 32-bit shall be taken as unit storage and check results.
- In the 5-bit started with D・, see the table above, locations of the same data, minimum vale and maximum value are stored. When the same data do not exist, D50~D52=0 in the above case.

Note: WhenS1·，S2· and range of D· exceed the boundary, it may lead to error.

F62 ABSD Control mode of cam control

F 62	D	ABSD		Control mode of cam control	S1 •	S2 •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •							*	*	*	*	*	*	*	*		
S2 •												*				
D •		*	*	*												
n					*	*										

Note: when the instruction is 16-bit instruction, △=4; when it is 32-bit instruction,△=8.
Times for use of the instruction: one time (decorated by the index).

Instruction format:

[ABSD S1 • S2 • D • n]

S1 • : Comparison start device. When the computation elements specify KnX, KnY, KnM, KnS, 16-bit instruction shall specify K4; 32-bit instruction shall specify K8 and numbers of X, Y, M and S shall be multiple of 16.

S2 • : Counter number. 16-bit instruction shall specify C0~C199, and 32-bit instruction shall specify C200~C255.

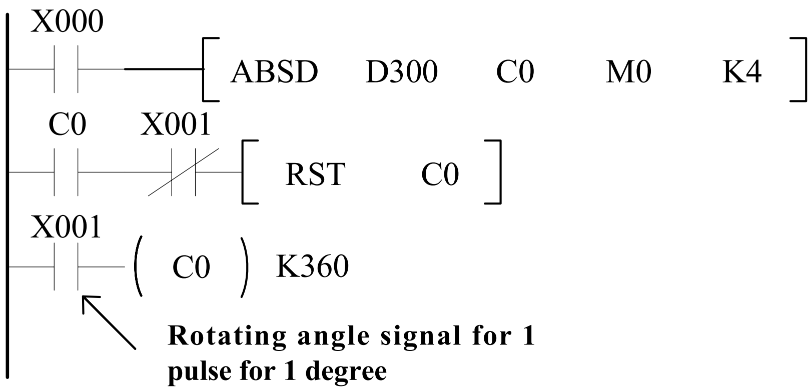
D • : Start number of comparison result output

n：Number of groups for multi-section comparison, n =1~64

※ 1：For S2 •counter number of TP03SR machine type, 16-bit instruction shall specify C0~C99, and 32-bit instruction shall specify C220~C255.

The instruction is used to generate several output waves for current values of the counter. Take the working bench rotating one cycle to control the auxiliary relay M0~M3 ON/OFF as example to describe in detail.

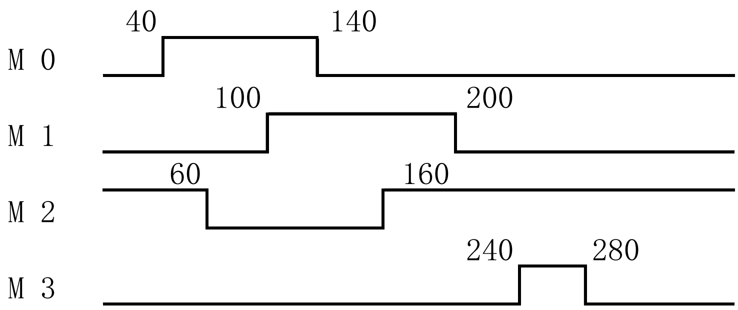
Example:



The following data is written into D300~D307 by using sending instructions in advance.

Rising point	Falling point	Object output
D300= 40	D301=104	M0
D302=100	D303=200	M1
D304=160	D305= 60	M2
D306=240	D307=280	M3

When X000 is ON, there are the following changes for M0~M3. Rising point/fall point may vary according to change of data of D300~D307.



- The value n determines number of points for output object.
- When X000 is OFF, the wave output remains the same.
- When the instruction DABSD is used, S2 • can specify high speed counter. However, the output wave may be delayed affected by the scan cycle. When high speed response is required, please use the instruction HSZ for comparing the performance.

F63 INCD Cam control increment mode

F 63		INCD		Cam control increment mode	S1 •	S2 •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •							*	*	*	*	*	*	*	*		
S2 •												*				
D •		*	*	*												
n					*	*										

Note: the instruction is 16-bit instruction,△=4

Instruction format:

[INCD S1 • S2 • D • n]

S1 • : Comparison start element, when KnX, KnY, KnM and KnS are specified, K4 shall be specified and number of X, Y, M and S shall be multiple of 16.

S2 • : Number of the counter, the 16-bit instruction shall specify C0~C198, occupying counters with 2 continuous numbers.

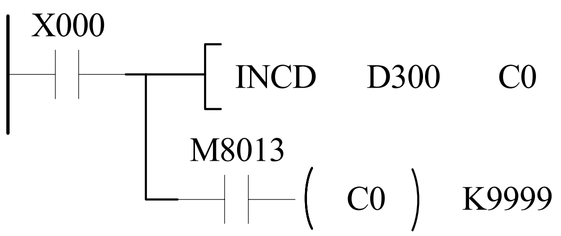
D • : Start number of comparison results

n: Number of groups for multi-section comparison n=1~64

※ 1: For S2 • counter number of TP03SR machine type, 16-bit shall specify C0~C98, occupying counters with 2 continuous numbers.

The instruction is used for several output waves generated by a couple of counters.

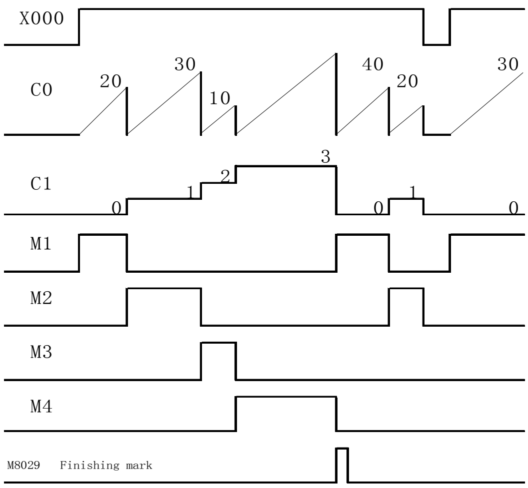
Example:



According to the time sequence sheet, take the control n=4 points for M0-M3 for interpretation.

- The following data is written into S1 • by the sending instructions in advance.

D300=20
D301=30
D302=10
D303=40



- When the counter C0 reaches setting values of D300~D303, it resets according to the sequence.
- The working counter C1 counts times of resetting.
- Current values of corresponding counters C1, M0-M3 works according to the sequence.
- When the output actions specified by n finish, the mark M8029 acts, it returns and starts the same actions.
- When X000 is OFF, C0 and C1 are cleared, M0~M3 is OFF. When X000 is set ON a second time, it works from initial state.

F64 TTMR Demonstrating teaching timer

F 64		TTMR		Demonstrating teaching timer	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •													*	*		
n					*	*										

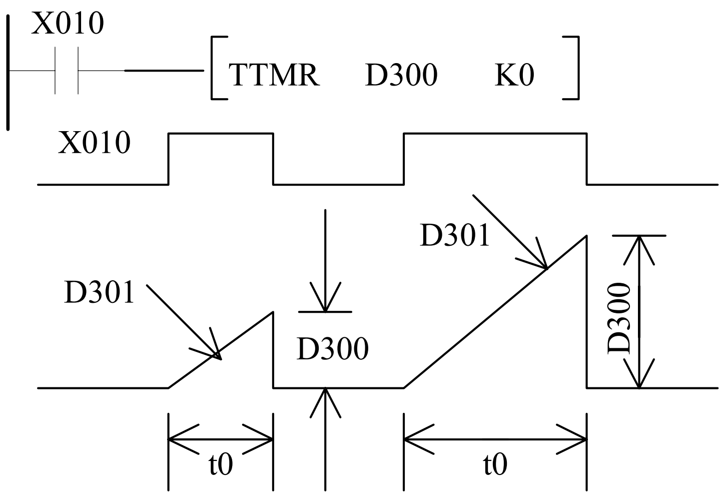
Instruction format:

[TTMR D • n]

D • ：Element number for the storage button switch ON, D • occupies devices with 2 continuous numbers

n：Setting of multiple, n=0~2

For example:



- Pressing time of the button X010 measured by D301 multiply by ratio specified by n and the result is stored in D300. Setting time of the timer can be adjusted by the button.
- Pressing time of the button X010 is τ 0 second, according to value of n, actual D300 is state below:

n	D300
K0	τ 0
K1	10 τ 0
K2	100 τ 0

- When X010 is OFF, D301 resets and D300 does not change.

F65 STMR Special timer

F 65		STMR		Special timer	S •	m	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •											*					
m					*	*										
D •		*	*	*												

Instruction format:

[STMR S • m D •]

S • ：Number of the timer, range of specifying: T0~T199

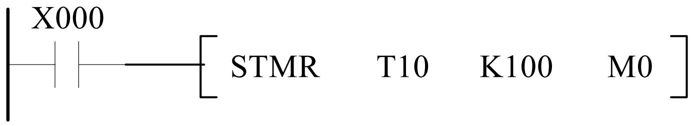
m：Setting value of the timer, unit 100ms, range: 1~32767

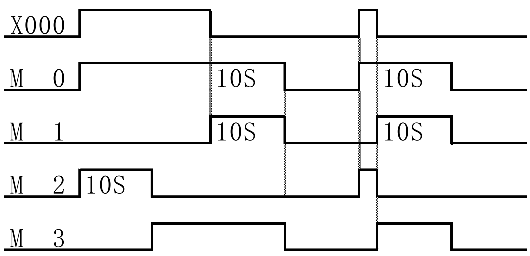
D • ： Start number of the output element, occupying 4 continuous number devices

※ 1：For S •counter number of TP03SR machine type, the range of setting: T0~T39，T196~T199

The instruction is for the delay timer, single triggering timer and twinkling timer.

Example:





- Specified value of m is the setting value of the specified timer, and it is 10 seconds in the case.
- M0 is a delayed timer.
- M1is a single-triggering timer after inputting ON→OFF.
- M2 and M3 are used for twinkling.
- When X000 is OFF, after setting the time, M0, M1 and M3 are OFF and T10 resets.
- Timer used here can not be used repeatedly in other general circuits.

F66 ALT ON/OFF Alternative output

F 66		ALT	P	ON/OFF Alternative output	D •

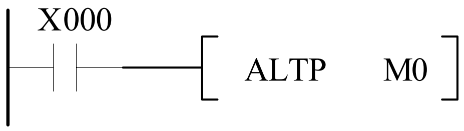
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •		*	*	*												

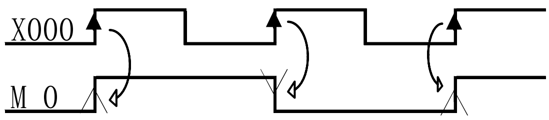
Instruction format:

[ALT D •]

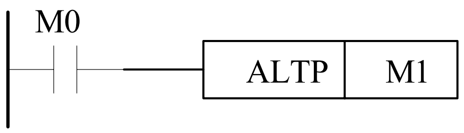
D • : Destination element

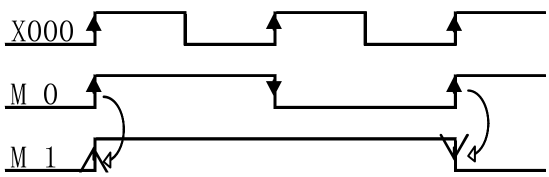
Example 1:





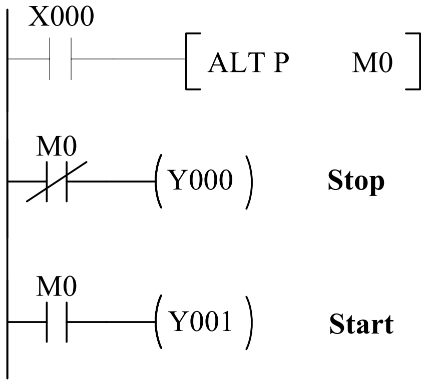
- When the drive input changes from OFF→ON each time, M0 is in reverse direction. When continuous executing instruction is used, each computation cycle performs reverse actions, please pay attention to it.
- M0 in the above figure is taken as input, when the instruction ALTP is used for driving M1, multi-level frequency dividing output can be obtained.





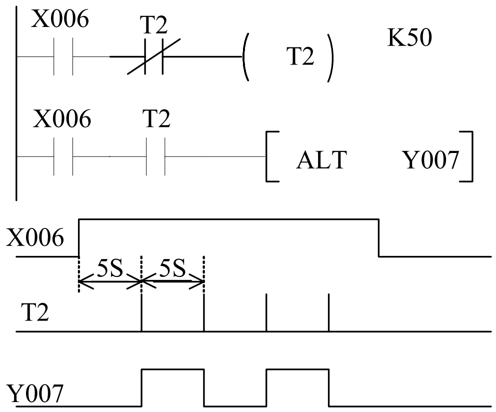
Application of repeated actions:

Start/stop from 1 input



- When pressing the button X000, the start output Y001 acts.
- When pressing the button X000, the stop output Y000 acts.

Twinkling action



- When the input X006 is ON, contact-joint of the timer T2 acts instantaneously with an interval of 5 seconds.
- When contact joint of T2 is ON each time, the output is ON/OFF alternatively.

F67 RAMP Slope signal

F 67	RAMP	Slope signal	S1 •	S2 •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •													*	*		
S2 •													*	*		
D •													*	*		
n					*	*										

Instruction format:

[RAMP S1 • S2 • D • n]

S1 • ：Element of initialized value of slope signal

S2 • ：Element of destination value

D • ： Process value

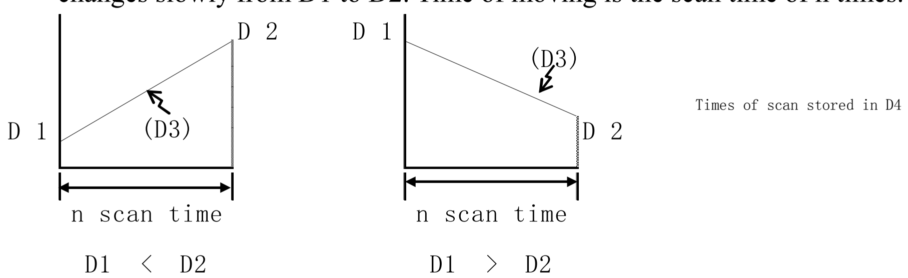
n：times of signal, n=1~32767

Example:

X000

[RAMP D1 D2 D3 K1000]

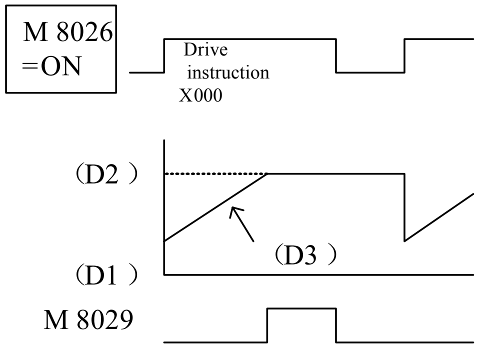
- Write preset initial value and destination value into D1 and D2. If X000 is ON, content of D3 changes slowly from D1 to D2. Time of moving is the scan time of n times.

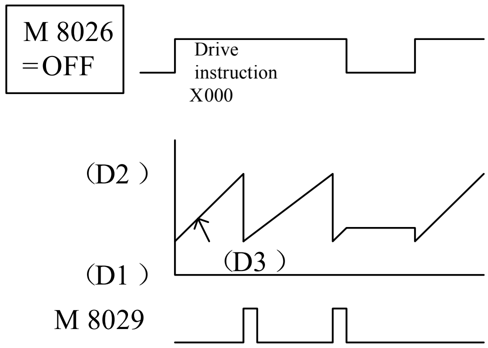


- The set scan time (a little longer than the actual scan time) is written into D8039, which is driven. The PLC is constant scan operation mode, if the value is 20ms, in the above case, after 20 seconds, D3 is changed from D1 to D2.
- During operation, if X000 is OFF, it becomes interrupted. When X000 is set ON a second time, D4 is cleared and it starts from D1.
- After executing, the mark M8029 is ON, value of D3 is back to D1.
- If the instruction combines the analog output, it outputs buffering start/stop instruction.
- X000 runs when it is ON, D4 is cleared in advance (D4 is for holding in case of power failure).

Actions of mode mark

In PLC, it acts according to the mode mark M8026, and content of D3 changes as follows:





F68 ROTC Rotating working bench control

F 68	ROTC	Rotating working bench control	S •	m1	m2	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •													*	*		
m1					*	*										
m2					*	*										
D •		*	*	*												

Instruction format:

[ROTC S • m1 m2 D •]

S：Register used in counting, use 3 consecutive devices.
m1：Number of encoder pulses per table revolution, m1=2~32,767.
m2：Distance to be traveled at low speed (in encoder pulses), m2=0~32,767（m1≥m2）.
D：Start element for signal output, uses 8 consecutive devices.
Example:

X010

[ROTC D200 K10 K2 M0]

The ROTC instruction is used to aid the tracking and positional movement of the rotary table as it moves to a specified destination.

- The ROTC instruction uses a built in 2-phase counter to detect both movement direction and distance traveled. Devices M0and M1 are used to input the phase pulses, while device M2is used to input the 'zero position' on the rotary table.
- D200 : Current position at the 'zero point' read only. D201 : Destination position (selected Station to be moved to) relative to the 'zero point' - User defined.D202: Start position selected station to be moved) relative to the 'zero point' -User defined.
- M0: A-phase counter signal - input
  M1: B-phase counter signal - input
  M2: Zero point detection - input
  M3: High speed forward - output
  M4: Low speed forward - output
  M5: Stop - output
  M6: Low speed reverse - output
  M7: High speed reverse - output
  Devices M3 to M7are automatically set by the ROTC instruction during its operation .These are used as flags to indicate the operation which should be carried out next.

F69 SORT Data sort

F 69	SORT	Data sort	S •	m1	m2	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •													*	*		
m1					*	*										
m2					*	*										
D •													*	*		
n					*	*							*	*		

Instruction format:

```
X000
—| |——[ SORT  S•  m1  m2  D•  n ]
```

S • : Start element of original data zone
m1：Number of groups of data，m1=1~32
m2：Number of rows of figure, m2=1~6
D • : Start element for storing sorting result data zone
n：Reference number of data sort n=1~m2
Example:

```
X020
—| |——[ SORT  D100  K5  K4  D200  D0 ]
```

When X020 is ON, the data starts sorting. After finishing, the mark M8029 ON stops operation. During operation, do not change the operand and data content. When it is operated a second time, set X020 OFF one time.

Working bench structure and data case

Number of row m2

Number of data m1

Line \ Row	1	2	3	4
	ID	Height	Weight	Age
1	D100 1	D105 150	D110 45	D115 20
2	D101 2	D106 180	D111 50	D116 40
3	D102 3	D107 160	D112 70	D117 30
4	D103 4	D108 100	D113 20	D118 8
5	D104 5	D109 150	D114 50	D119 45

- Start data register of the working bench is specified by S • .
- Input ID and other continuous numbers in the 1st line to identify the original line number.

For executing the instruction D0=K2

Row No. / Line No.	1	2	3	4
	ID	Height	Weight	Age
1	D200 4	D205 100	D210 20	D215 8
2	D201 1	D206 150	D211 45	D216 20
3	D202 5	D207 150	D212 50	D217 45
4	D203 3	D208 160	D213 70	D218 30
5	D204 2	D209 180	D214 50	D219 40

For executing the instruction D0=K3

Row No. / Line NO.	1	2	3	4
	ID	Height	Weight	Age
1	D200 4	D205 100	D210 20	D215 8
2	D201 1	D206 150	D211 45	D216 20
3	D202 2	D207 180	D212 50	D217 40
4	D203 5	D208 150	D213 50	D218 45
5	D204 3	D209 160	D214 70	D219 30

- Data of the computation result starts with the element specified by D • , occupying m1×m2 data registers. When S • and D • are the same elements, before operation finishing, do not change content of S • .
- Executing of the instruction needs m1 scan cycles, after data sorting finishes, the mark M8029 acts.

F70~F79 Peripheral equipment IO Instructions

Peripheral equipment IO instructions

F70 TKY Decimal key input

F 70	D	TKY		Decimal key input	S •	D1 •	D2 •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •	*	*	*	*												
D1 •								*	*	*	*	*	*	*	*	*
D2 •		*	*	*												

Times of use of the instruction: 1 time (Decorated by index register)

Instruction format:

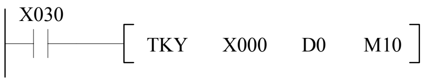
[TKY S • D1 • D2 •]

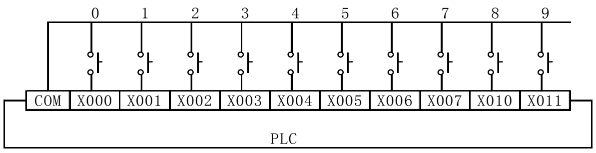
S• : Start device for key input, occupying continuous 10 points

D1• : Storage location for key input values

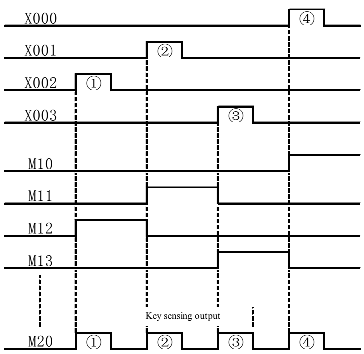
D2• : Key output signal

For example





- Input the decimal key according to the sequence①②③④ in the figure above, and content of D0 is changed to 2,130. Maximum value of the input is 9,999, and overflow will occur if it exceed (content of D0 is stored in binary system).
- When the instruction D TKY is used, D1and D0 buffers are used. When the value exceeds 99,999,999, overflow will occur.
- Start from pressing X002 to other keys, M12 keeps ON and other keys are the same.
- If corresponding X000~X011 act, M10~M19 will act.
- When pressing any button, during pressing, if M20 is detected, it will be ON. When several keys are pressed, the earlier pressed shall prevail.
- When the input X030 is OFF, content of D0 will change, and M10~M2 are OFF.



F71 HKY Hexadecimal key input

F 71	D	HKY		Hexadecimal key input	S• D1• D2• D3•

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •	*															
D1 •		*														
D2 •											*	*	*	*	*	*
D3 •		*	*	*												

Time of use of the instruction: 1 time (Decorated by the index register)

Instruction format:

[HKY S• D1• D2• D3•]

S• :Start device for key input, occupying continuous 4 points

D1• :Start device for key scan output, occupying continuous 4 points

D2• :Location for storing key input values

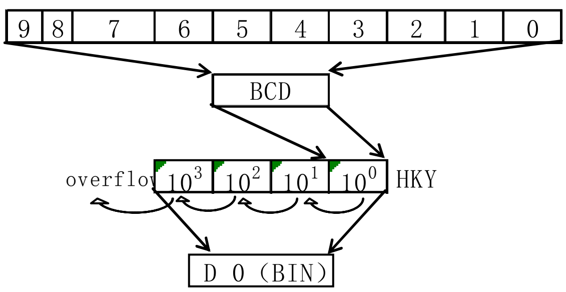
D3• :Key output signal

The instruction is used for writing numerical value and input functions with hexadecimal key.

Example:

X000 [HKY X010 Y010 D0 M0]

Digit keys



- When the digit key is pressed each time, values with upper limit 9,999 is stored in D0, if it exceeds the value, overflow will occur.
- When the instruction DHKY is used, D1 and D0 are valid between 0~99999999.
- When several keys are operated, the earlier pressed shall prevail.
- After one cycle scan for Y010~Y013, the finishing mark M8029 acts.

Function keys

F	E	D	C	B	A

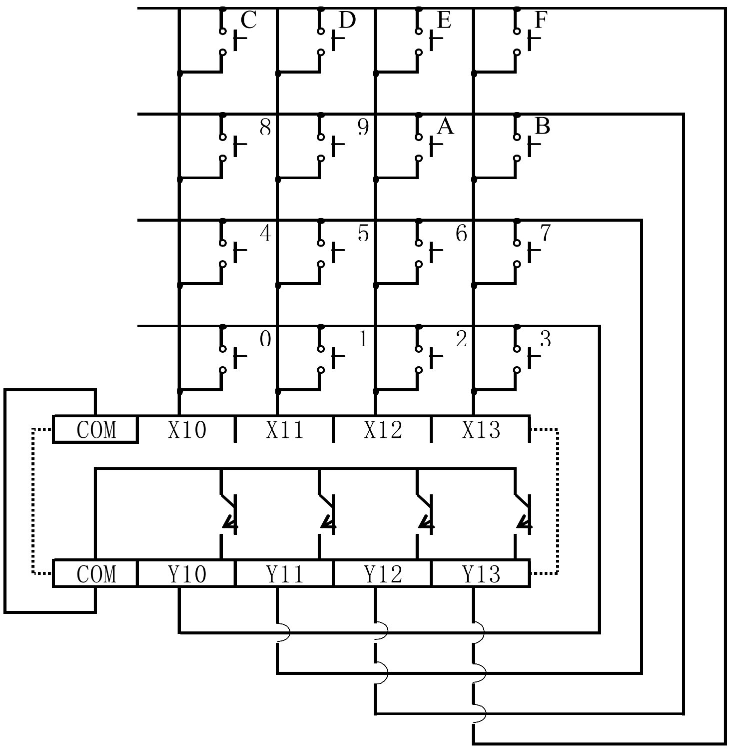
M5	M4	M3	M2	M1	M0

- When the key A is pressed, M0 keeps ON; when D is pressed, M0 is OFF and M3 keeps ON.
- When several keys are operated, the earlier pressed shall prevail.

Output of key testing

- When any key of A~F is pressed, it works only M6 is pressed.
- When any key of 0~9 is pressed, it works only M7 is pressed.
- When the drive input X000 is OFF, D0 does not change, and M0~M7 are OFF.

External circuit



Functions of the mark M8167:

When M8167=ON, the instruction HKY can input hexadecimal number of 0~F;

When M8167=OFF, the instruction HKY can be used as function key.

Additional interpretation:

When the instruction is executed, one input value of the key can be seized effectively after 8 scan cycles. Long or short scan cycle may lead to non-conformity of the key, therefore, the following skills can be used to overcome the difficulties.

1 If the scan cycle is too short, it may lead to untimely response of I/O and input value of the key, at this time, the scan time shall be fixed.

2 If the scan cycle is too long, it may cause longer response of the key, the instruction is written into the interruption subprogram to be executed the instruction at specified time.

F72 DSW Digit switch input

F 72	DSW	Digit switch input	S・	D1・	D2・	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S・	*															
D1・		*														
D2・											*	*	*	*	*	*
n					*	*										

Times of use of the instruction: 2 times（Decorated by the index register）

Instruction format:

[DSW S・ D1・ D2・ n]

S· : Start device for scan input of finger-dialing switch

D1· : Start device for scan output of finger-dialing switch

D2· : Location of setting value of finger-dialing switch

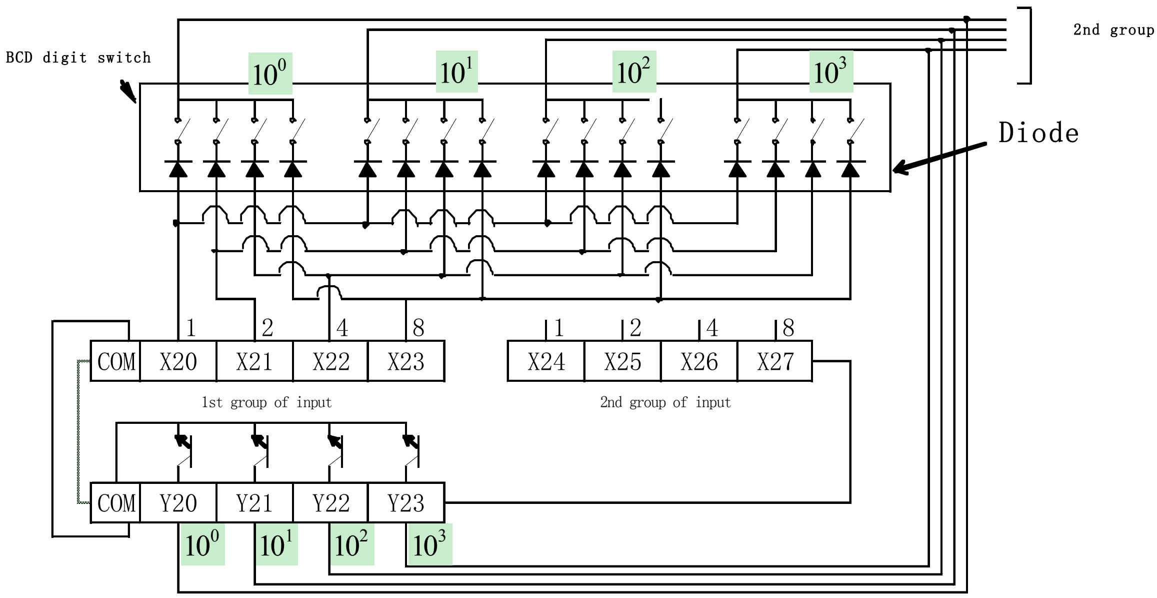
n: Number of groups connected by the finger-dialing switch, n=1~2

The instruction is used for read of BCD code set by the digit switch for 1(n=1) or 2 (n=2) groups of 4-bit numbers, when the input is not BCD code, it may lead to error.

Example:

X000

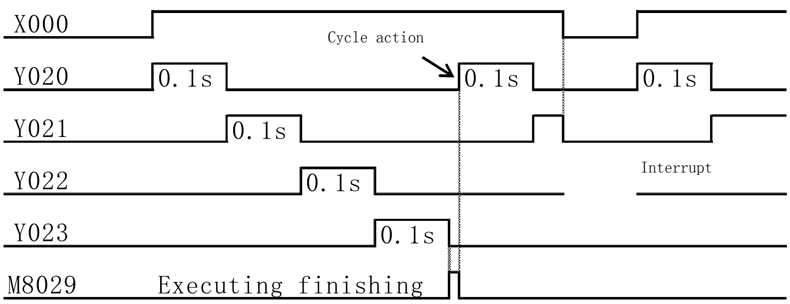
[DSW X020 Y020 D0 K1]



Note: when the input is not BCD code, it may be error.

The 1st group of input: the 4-bit BCD code digit switch connecting X020~X023 reads according to the sequence of Y020~Y023 and stored in D0.

The 2nd group of input: the 4-bit BCD code digit switch connecting X024~X027 reads according to the sequence of Y020~Y023 and stored in D1.(Valid when n=2)



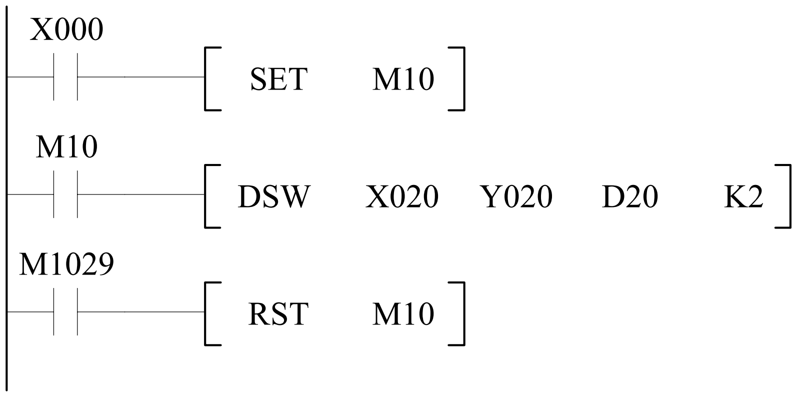
- When X000 is ON, Y010~Y013 work according to the sequence（100ms）. After one cycle, the finishing mark M8029 acts.
- When DSW values need continuous input, PLC with transistor output shall be used.

Additional interpretation:

When PLC is used as relay output, the following method can be used:

1 When X000=ON, the instruction DSW is executed; when X000 is OFF, M10 will continue to keep ON and it will be OFF until the scan terminal of DSW instruction finishes one cycle.

2 The condition contact-point X000 uses button switch, when it is used one time, M10 will be OFF after the scan terminal specified by the instruction DSW finishes one cycle output, the instruction stops output and the finger-dialing switch data will be fully read. Therefore, under the circumstances, even the scan terminal uses relay output, service life of the relay will not be shortened.



F73 SEGD 7-section decoding

F 73		**SEGD**	P	7-section decoding	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

SEGD S • D •

S· :Source device of decoding

D· :Output device after decoding

0-F (hexadecimal system number) of specified by low 4-bit of S· is decoded into D· of data in the 7-section code. High 8-bit of D· does not change.

M8273 specifies logic relation of output display.

When M8273=OFF,

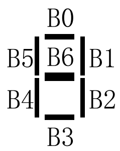
See the Seven-section Code Sheet

Source		7-section combined number	Pre-setting								Number represented
Hexadecimal system number	Bit combination format		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000	B0 B5 B6 B1 B4 B2 B3	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	1	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

Start of bit element(such as Y000)
Or last bit of word element or B0

When M8273=ON,

See the following on the 7-section decoding sheet.

Source		7-section combined number	Pre-setting								Number represented
Hexadecimal system number	Bit combined format		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000		1	1	0	0	0	0	0	0	0
1	0001		1	1	1	1	1	0	0	1	1
2	0010		1	0	1	0	0	1	0	0	2
3	0011		1	0	1	1	0	0	0	0	3
4	0100		1	0	0	1	1	0	0	1	4
5	0101		1	0	0	1	0	0	1	0	5
6	0110		1	0	0	0	0	0	1	0	6
7	0111		1	1	0	1	1	0	0	0	7
8	1000		1	0	0	0	0	0	0	0	8
9	1001		1	0	0	1	0	0	0	0	9
A	1010		1	0	0	0	1	0	0	0	A
B	1011		1	0	0	0	0	0	1	1	B
C	1100		1	1	0	0	0	1	1	0	C
D	1101		1	0	1	0	0	0	0	1	D
E	1110		1	0	0	0	0	1	1	0	E
F	1111		1	0	0	0	1	1	1	0	F

F74 SEGL 7-section code display

F 74	SEGL	7-section code display	S •	D •	n

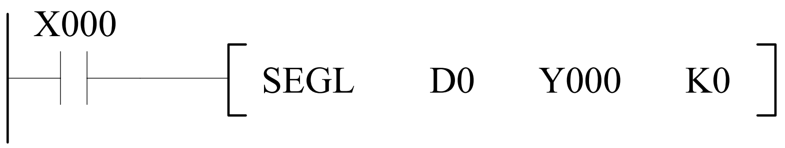
	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •		*														
n					*	*										

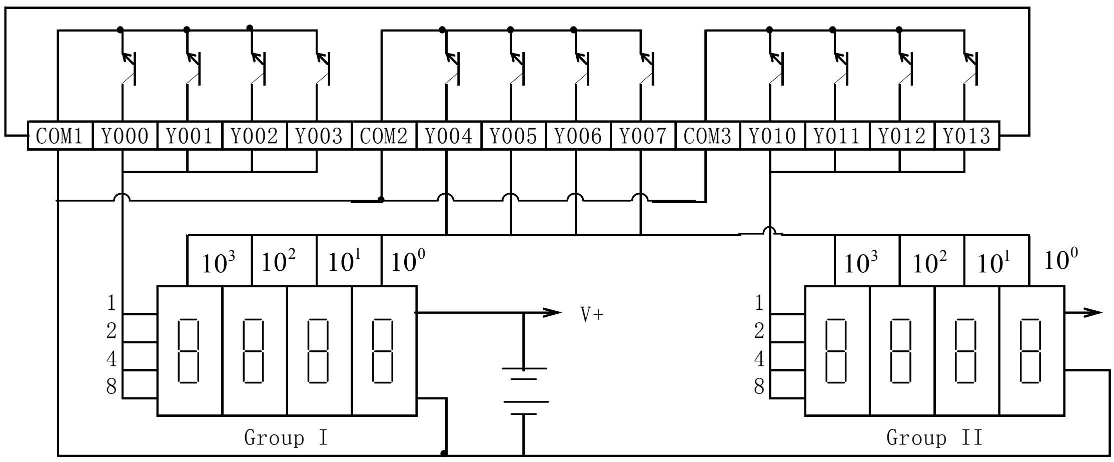
Times of use of the instruction: 2 times（Decorated by the index register）
Instruction format:

[SEGL S• D• n]

S• : Source device to display seven-section code
D• : Start device for seven-section monitor scan output
n: Polarity setting of output signal and scan signal, n=0~7

The instruction is used for controlling 1 or 2 groups of 4-bit seven-section code with locking.
Example:





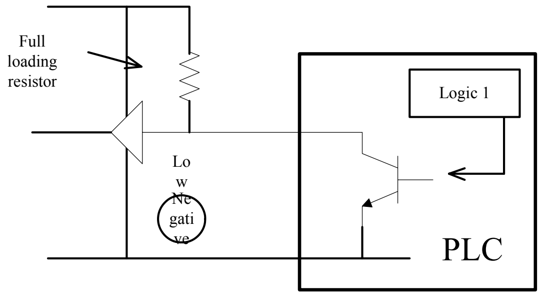
4-bit for one group, n=0~3

- Conversion result of D0 (binary system, after conversion in BCD, effective between 0~9,999), they are output to (Y000~Y003).
- The strobe pulse signals (Y004~Y007)are used to lock the 4-bit for one group 7-section code with locking.

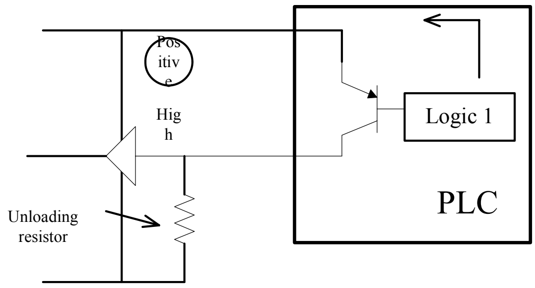
4-bit for 2 groups, n=4~7

- D0 is output to (Y000~Y003), D1 is output to (Y010~Y013) and D1 and D0 execute BCD conversion and it is effective between 0~9,999, otherwise, it will cause error.
- The strobe pulse signal (Y004~Y007)is shared by 2 groups.
- The instruction is used for 4-bit (1 or 2 groups) display, which requires 4 times of the computation cycle. After 4-bit output finishes, the finishing mark M8029 acts.
- When drive input of the instruction is ON, it executes repeated actions. Of one series of actions, when the drive input is OFF, it stops acting and restarts from the initial action.
- The instruction is simultaneously executed with the scan cycle of the PLC. To execute a series of displays, the scan cycle of the PLC needs more than 10ms. When it is less than 10ms, constant scan mode shall be used and scan cycle more than 10ms shall be operated.
- ON voltage of transistor output of the PLC is about 1.5V, the 7-section code shall use the corresponding output voltage.
- Setting of parameter n: it is used to set loop of positive or negative or the transistor, and the seven-section monitor is one group or two groups of 4-bit fingers.

Logic of PLC



NPN transistor output loop: when the internal signal is 1, it outputs low level, and it is called negative logic.



PNP transistor output loop: when internal signal is 1, it outputs high level, and it is called positive logic.

Logic of 7-section monitor

Description	Positive logic	Negative logic
Data input	High level converted into BCD data	Low level converted into BCD data
Strobe pulse signal	High level keeps the data of locking	Low level keeps the data of locking

Selection of parameter n

Number of groups for 7-section display	Group I				Group II			
Logic between PLC output terminal and monitor data input terminal	Same		Different		Same		Different	
Logic between PLC output terminal and monitor data scan signal	Same	Different	Same	Different	Same	Different	Same	Different
n	0	1	2	3	4	5	6	7

When logics of PLC transistor output and 7-section monitor are the same or not the same, it can be matched with setting value of the parameter n.
Assume PLC output is negative logic, data input terminal of 7-section monitor is also negative logic and strobe pulse signal of 7-section monitor is positive logic, if it is 4-bit for 1 group, n=1; for 4-bit for 2 groups, n=5.

F75 ARWS Direction switch

F 75	ARWS	Direction switch	S •	D1 •	D2 •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •	*	*	*	*												
D1 •											*	*	*	*	*	*
D2 •		*														
n					*	*										

Times of use of instructions: 1 time (decorated by index register)

Instruction format:

[ARWS S • D1 • D2 • n]

S• : Start device for key input, occupying continuous 4 points

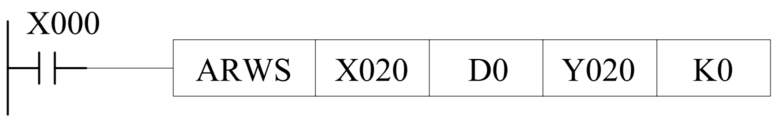
D1• : Device with data input by arrow switch

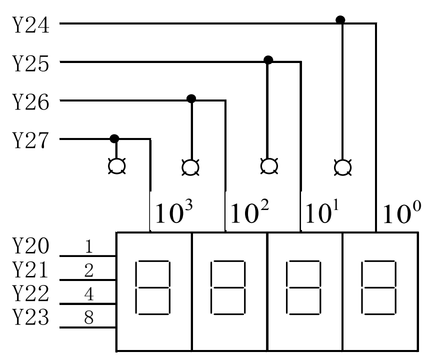
D2• : Start device with seven-section monitor scan output, occupying 8 points and it is used to set values by visual means.

n: the same as n in the instruction SEGL, with n=0~3

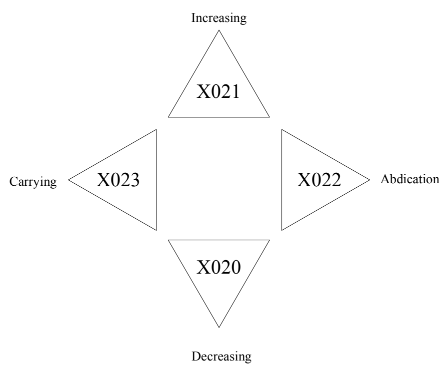
The instruction is used for inputting data by bit shifting and arrow keyboard switch for increasing and decreasing the data.

Example:





ection decoding monitor for visual sett



- 16-bit binary BCD (effective converted into 0-9,999) is stored in D0, for convenience, BCD code is used in the following interpretation.
- When the drive input X000 is ON, it is specified as 10^3-bit. When pressing the abdication key each time, it changes as per 10^3-10^2-10^1-10^0-10^3. In addition, when pressing the carrying key each time, it changes as per 10^3-10^0-10^1-10^2-10^3. The specified bit can display by LED according to the strobe pulse signals (Y024~Y027).
- As for specified bit, once the added key D0 is pressed, it changes as per 0-1-2-3-4-5-6-7-8-9-0-1; when the decreasing button is pressed, it changes as per 0-9-8-7-------1-0-9. The content is displayed by 7-section monitor.

  See the above, while you are viewing the monitor, you can write the destination value into D0.

Additional interpretation:

The output points Y20~Y27 specified by the instruction must use transistor output.

When the instruction is used, please fix the scan time or the instruction is inserted into the interruption subprogram for executing.

F76 ASC ASCII code conversion

F 76		ASC		ASCII code conversion	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •											*	*	*	*		

Instruction format:

[ASC S D •]

S : 8 letters and fingers input by the computer

D• : Start address of ASCII code device after storing and conversion

Example:

X000

[ASC ABCDEFGH D300]

Executing result of the instruction: After the letters A-H is converted by ASCII, and it is moved to D300~D303.

	High 8-bit	Low 8-bit
D300	42(B)	41(A)
D301	44(D)	43(C)
D302	46(F)	45(E)
D303	48(H)	47(G)

- The instruction is used for displaying error and other information on external monitor.
- After M8161 is set ON, when executing the instruction, low 8-bit is moved to D•, which occupies the devices with the same quantity of the character moved and the high 8-bit is 0.

	High8-bit	Low 8-bit	
D300	00	41	A
D301	00	42	B
D302	00	43	C
D303	00	44	D
D304	00	45	E
D305	00	46	F
D306	00	47	G
D307	00	48	H

F77 PR ASC II code printing

F 77	PR	ASC II code printing	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •											*	*	*	*		
D •		*														

Times of use of the instruction: 1 time (Decorated by the index register)

Instruction format:

[PR S • D •]

S• : Start element of ASCII code to be moved

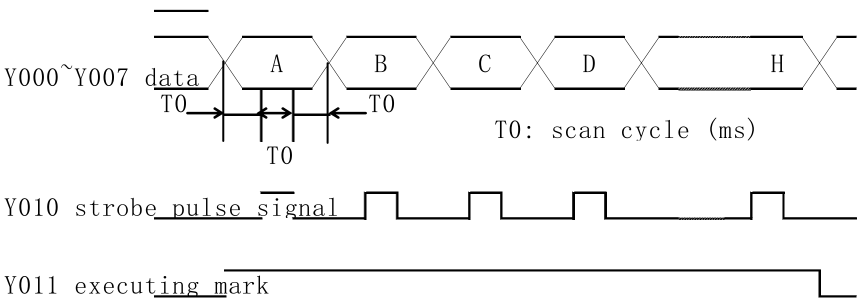
D• : External output point of outputting ASCII code, occupying continuous 10 points.

The instruction is used for moving ASCII code data to Y.

Example:

X000 —| |— [PR D300 Y000]

- See the previous page, if ASCII data is stored in D300 ~D303, sequence of moving starts with A and ends with H.
- The moving output is Y000 (Low bit) ~Y007 (high bit), and others include strobe pulse signal Y010 and executing mark Y011.

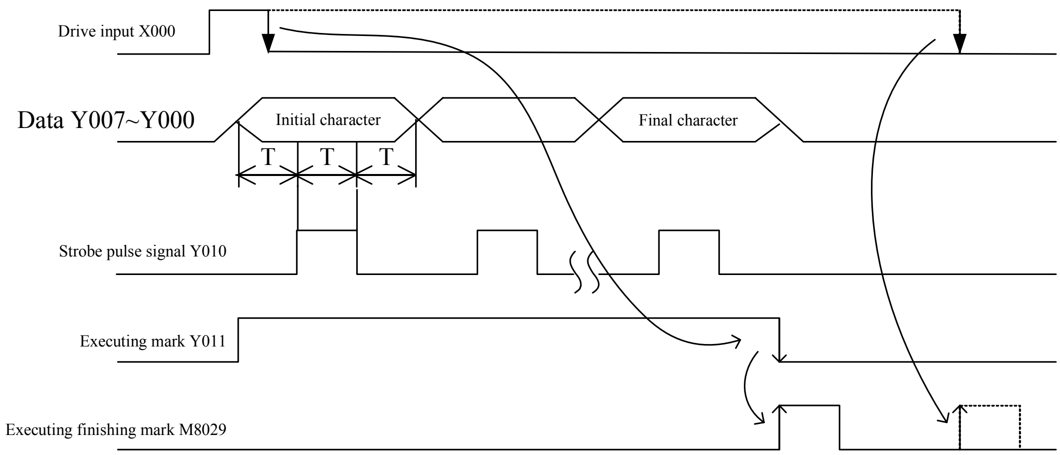


- When the drive input X000 is executing, if it is OFF, the moving is stopped immediately. When it is ON a second time, it acts from the initial state.
- When the instruction is executed with the scan cycle (T0 in the above figure), if the scan cycle is short, please use constant scan mode; if it is long, please use timing interruption mode.
- Transistor output must be used for the PLC.

16-character sequential output

FNC77 (PR) is the instruction for series connection output 8-bit parallel data in sequence. When the special auxiliary relay M8027=OFF, it is 8-character series connection output; when M8027=ON, it is 1-16 character series connection output. When moving the data, in case of 00H (NUL), it stops executing the instruction and the data left will not be output.

Action of the instruction PR, when M8027=ON,



T: Computation cycle or time of interruption

- When the instructions drive rising edge of X000, the instruction starts executing; in data output, if the drive X000 is OFF, the output is not stopped.
- The drive X000 is ON continuously, after one cycle output, it stops output. However, the mark 8029 will act until X000 is OFF.

F80~F89 Peripheral equipment SER

Peripheral equipment SER

F80 RS Serial data transmission

F 80		RS		Serial data transmission	S m D n K

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S													*	*		
m					*	*							*	*		
D													*	*		
n					*	*							*	*		
K					0,1	0,1										

Instruction symbol:

[RS S m D n K]

S: Address of data sent
m: Length of data sent (0~255)
D: Address of data received
n: Length of data received (0~255)
K: Serial port selection, constant 0~1
0: communication port of RS485; 1: communication port of extended card of RS485 or RS232.

The instruction is the communication port of extended cards of RS-232 and RS-485 to achieve sending and receiving of serial non-protocol data.

- Data sending format of RS485 communication port can be set by the special buffer D8120. During executing of the instruction RS, even settings of D8120 are changed, it is not accepted in fact.
- Data sending format of communication port of extended cards of RS485 or RS232 can be set by the special buffer D8320. During executing of the instruction RS, even settings of D8320 are changed, it is not accepted in fact.
- In the environment not for information sending, point of information sending is set at “K0”. In addition, in the environment not for information receiving, point of information receiving is set at (K0).
- Although one program can use a large number of RS, MBUS, DTLK, RMIO and other communication instruction, one communication instruction is driven at one serial port at the same time, time of OFF shall be equal to or more than one scan cycle during switching.

Communication format
<Communication format (D8120), (D8320)>
In addition to non-sequential communication from the instruction FNC80 (RS), when the communication formats D8120 and D8320 are for other communication instructions or calculator connection, special data buffer can be used.

When the instruction FNC80 (RS) used, relevant setting for other communication instructions or calculator connection will be unavailable, please comply with the format setting of attentions.

Bit number	Name	Content	
		0(Bit OFF)	1(Bit ON)
B0	Length of data	7 bit	8 bit
B1 B2	Parity	B2,B1 (0,0): None (0,1): (ODD) (1,0): (EVEN)	
B3	Stop bit	1 bit	2 bit
B4 B5 B6 B7	Communication speed (bps)	B7,B6,B5,B4 (0,1,1,1):9,600 (1,0,0,0):19,200 (1,0,0,1):38,400 (1,0,1,0):57,600	B7,B6,B5,B4 (1,0,1,1):76,800 (1,1,0,0):128,000 (1,1,0,1):153,600 (1,1,1,0):307,200
B8*1	Start character	None	Yes, initial value: STX(02H)
B9*1	Stop character	None	Yes, initial value: ETX(03H)
B10~B15*2	Unavailable		

*1: content of the start character and stop character can be changed by the user. When calculator connection is used, reset use shall apply.
*2:B10~B15 are the setting items for other communication instructions or calculator connection. When the instruction FNC80 (RS) is used, it shall be used with “0”.

- Setting example of communication format

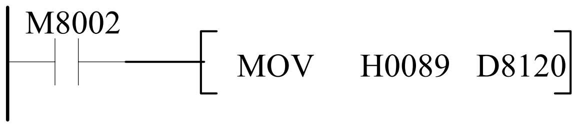
Length of data	8 bit
Parity	None
Stop bit	2 bit
Transmission speed	19,200 bps
Start character	None
Stop character	None
Communication port selection	RS485 communication port

Communication setting in the table above shall be set according to the following program or serial communication of peripheral machines.

	b15			b12	b11			b8	b7			b4	b3			b0
D8120	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1

↓

D8120 89H



Used special relay and data register:

a) When RS485 communication port is used:

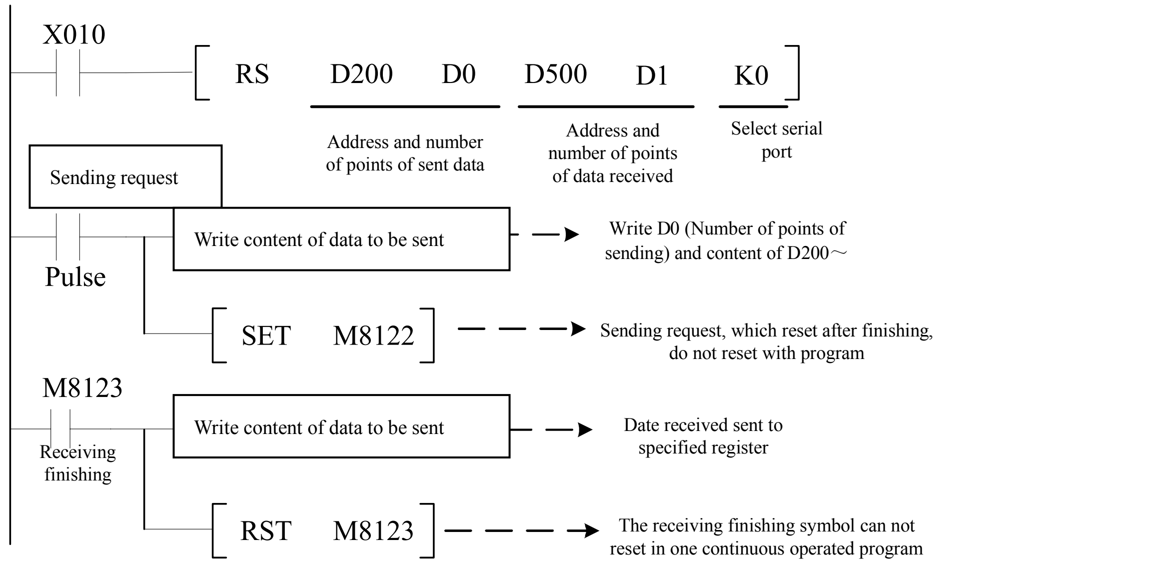
1) Sending waiting (M8121): if there is sending request during data receiving, 1 is set, after receiving and during data sending, it is cleared automatically.
2) Sending request (M8122): when M8122 is set by a pulse instruction under receiving waiting or receiving finishing, the data of length (m) is started to send from S. When sending finishes, M8122 resets automatically.
3) Receiving finishing (M8123): after receiving finishing, M8123 is ON, the data received is transmitted to specified register, then reset to enter receiving waiting.
4) Timeout judging (M8129): during data receiving, receiving will not be restarted, and the timeout output mark is ON, the receiving finishes; when M8123 is cleared, M8129 is automatically cleared.
5) Communication format setting (D8120): refer to RS instruction communication format described in previous text.
6) Number of residual data to be sent (D8122)
7) Number of data received (D8123)
8) Start character (D8124): it is for user to set the start character.
9) Stop character (D8125): it is for user to set the stop character.
10) Time of timeout judging (D8129): set the time for timeout judging (5~255)*10ms

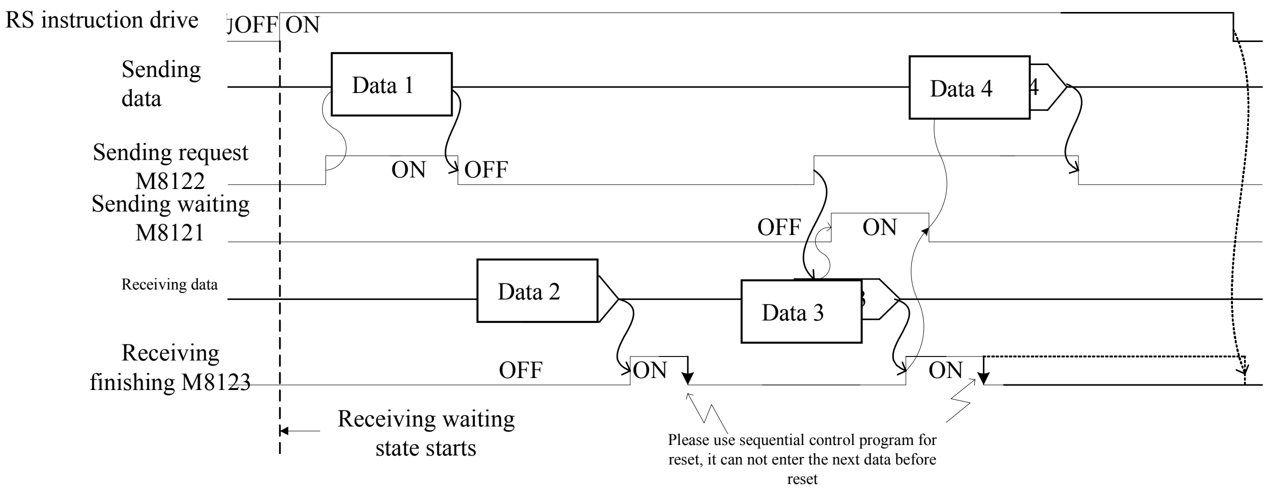
b) When communication ports of extended card of RS485 or RS232 are used (see the above text on definitions)

1) Sending waiting (M8321)
2) Sending request (M8322)
3) Receiving finishing (M8323)
4) Timeout judging (M8329)
5) Communication format setting (D8320)
6) Number of residual data to be sent (D8322)
7) Number of data received (D8323)
8) Start character (D8324)
9) Stop character (D8325)
10) Time of timeout judging (D8329)

Time sequence of sending and receiving

RS instruction does not stipulate first address and number of points of sent data from PLC, but also stipulate storage first address and maximum number of points of received data. See the following on sequence of sending and receiving data with RS instruction (RS485 communication port).





Sending request M8122

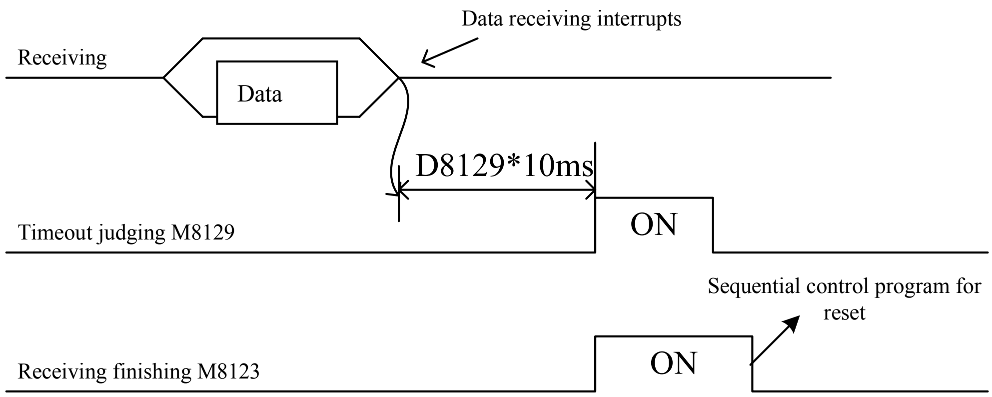
- The input condition X010 ON executes RS instruction, the controller enters receiving waiting.
- In case of receiving waiting or receiving finishing, M8122 is set ON by pulse signal, continuous D0 data is sent from D200. When sending finishes, M8122 will RESET OFF automatically.

Receiving finishing M8123

- When receiving finishing mark M8123 ON, all the data received will be transferred to other assembly for storing, and M8123 RESET is OFF.
- If M8123 RESET is OFF, it enters receiving waiting. The input condition X010 ON executes the RS instruction, the controller enters receiving waiting.
- When setting (D1) = 0, execute the MBUS instruction, M8123 will not act or enter receiving waiting. If after D1≧1, after M8123 ON is OFF, it enters receiving waiting.

Judging of timeout M8129

- When data receiving is interrupted, if data receiving is not restarted from the time to specified time by D8129, it is regarded as timeout and M8129 will be ON and becomes receiving finishing.
  M8129 will reset automatically with reset of M8123 program.
  By using the function, receiving finishing can be finished without stop character.

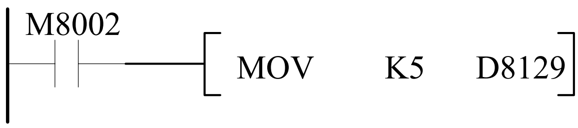


Time of judging timeout

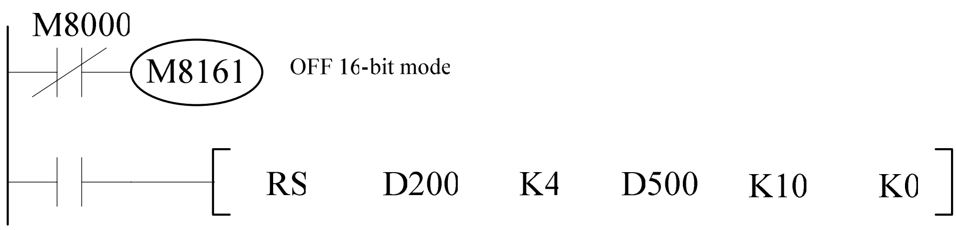
- Set using time of the above timeout judging

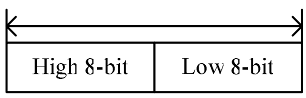
When setting the time, the setting value is X10ms, virtual value 5~255. When D8129 setting exceeds the range, it will become 50ms.

Take Time of timeout judging 50ms as example.



<Processing of 16-bit data> When M8161=OFF, (M8161 is shared by ASCI,HEX,CCD and other instructions)





The 16-bit data is divided into high and low 8-bit for data sending and receiving

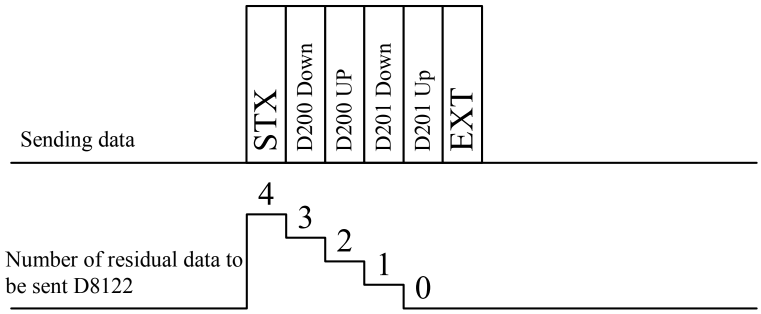
Sending data

STX	D200 Down	D200 Up	D201 Down	D201 Up	ETX
Start character	↑ S. Specified start address No. M specified number of byte to be sent				Stop character

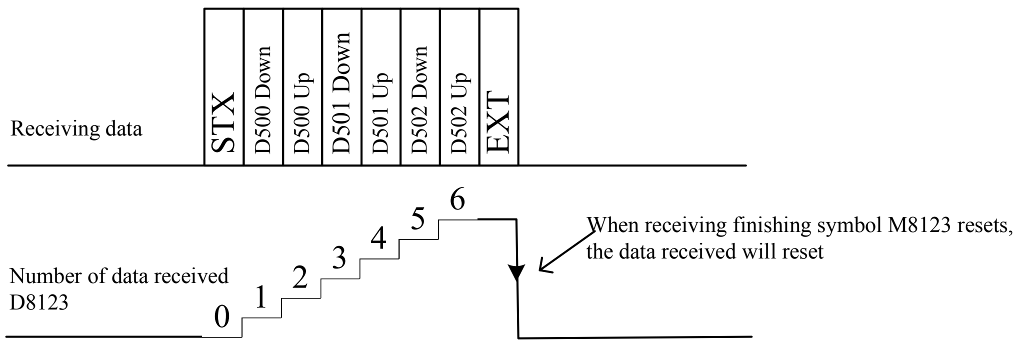
Receiving data

STX	D500 Down	D500 Up	D501 Down	D501 Up	D502 Down	D502 Up	ETX
Start character	↑ D. Specified start address No.						Stop character

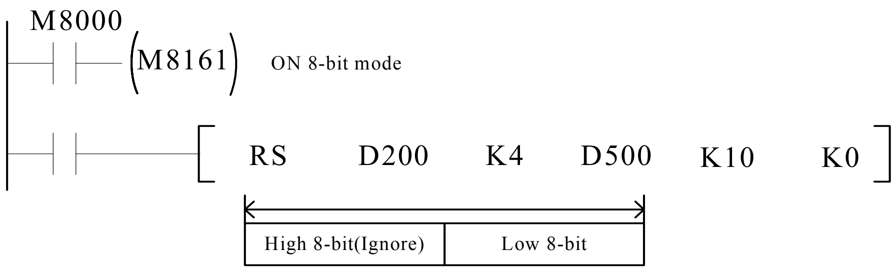
(1) Sending data and number of residual data to be sent



(2) Receiving data and number of data received



<Processing of 8-bit data (extension function)>When M8161=ON, (M8161 is shared by ASCI, HEX, CCD and other instructions)



16-bit data ignores high 8-bit, only low 8-bit is valid

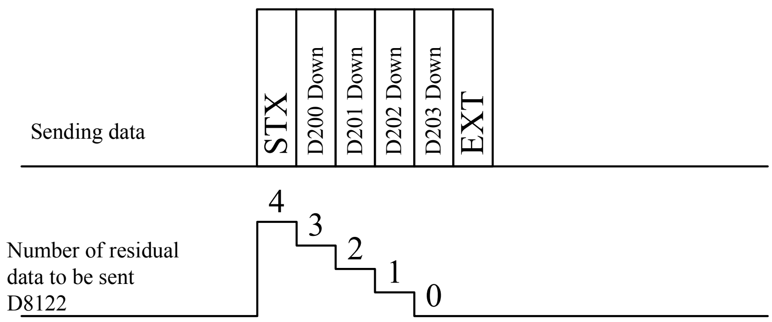
Sending data

STX	D200 down	D201 down	D202 down	D203 down	ETX
Start character	↑ S． specified start address No. M specifies number of byte to be sent				Stop character

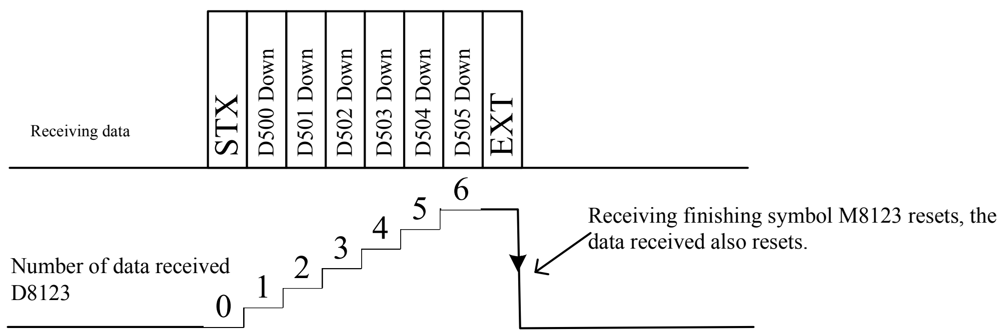
Receiving data

STX	D500 down	D501 down	D502 down	D503 down	D504 down	D505 down	ETX
Start character	↑ S. Specified start address No. It may not exceed maximum points specified by n, and the stop character EXT or points n receiving represents receiving finishing.						Stop character

（1）Sending data and number of residual data to be sent



（2）Receiving data and number of data received



F81 PRUN Octal code bit sending

F 81	D	**PRUN**	P	Octal code bit sending	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*		*							
D •								*	*							

Instruction format:

[PRUN S • D •]

S·: Sending source device, n of KnX, KnM is 1~8, and minimum bit of the specified device No. is 0.
D·: Sending destination device

The instruction is specified device number with octal code system and sends data.
Example 1:

X000
[PRUN K4X10 K4M0]

When instructions X10~X17→M0~M7，X20~X27→M10~M17 are executed, values of M8 and M9 will not be changed.

Example 2:

X000
[PRUN K4M0 K4Y10]

When instructions M0~M7→ Y10~Y17, M10~M17→ Y20~Y27 are executed, values of M8 and M9 will not be sent.

F82 ASCI HEX converted into ASCII

F 82		ASCI	P	HEX converted into ASCII	S •	D •	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*		
n					*	*										

Instruction format:

[ASCI S • D • n]

S·: Convert source device
D·: Convert destination device
n: Bit to be converted, n=1~256

The instruction is used to convert HEX code into ASCII code, with 16-bit and 8-bit conversion modes. For example:

X000

[ASCI D100 D200 K4]

Of them: D100=0ABCH
D101=1234H
D102=5678H.

ASCII:	[0]=30H	[1]=31H	[2]=32H	[3]=33H	[4]=34H	[5]=35H
	[6]=36H	[7]=37H	[8]=38H	[9]=39H	[A]=41H	[B]=42H
	[C]=43H	[D]=44H	[E]=45H	[F]=46H		

16-bit conversion mode, when M8161＝OFF (M8161 is shared by RS, ASCI, HEX, CCD and other instructions)

- After figures of HEX of S· are converted into ASCII code, they are sent to high 8-bit and low 8-bit of D· of S·, figure of conversion is set with n.
- D· is classified into low 8-bit and high 8-bit, which are used to store ASCII data.

Conversion results:

n / D·	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200 Low	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D200 High		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D201 Low			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D201 High				[C]	[B]	[A]	[0]	[4]	[3]
D202 Low					[C]	[B]	[A]	[0]	[4]
D202 High						[C]	[B]	[A]	[0]
D203 Low	Do	Not	Change				[C]	[B]	[A]
D203 High								[C]	[B]
D204 Low									[C]

8-bit conversion mode, when M8161=ON, (M8161 is shared by RS, ASCI, HEX, CCD and other instructions)

- After figures of HEX of S· are converted into ASCII code, they are sent to low 8-bit of D· and figures of conversion is set with n.
- High 8-bit of D· is 0.

16-bit data

0	Low 8-bit

Destination address

Conversion results:

n / D·	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D201		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D202			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D203				[C]	[B]	[A]	[0]	[4]	[3]
D204					[C]	[B]	[A]	[0]	[4]
D205						[C]	[B]	[A]	[0]
D206	Do	Not	Change				[C]	[B]	[A]
D207								[C]	[B]
D208									[C]

When BCD is output by printing and etc., BIN→BCD conversion instruction is used before the instruction.

F83 HEX ASCII converted into HEX

F 83		HEX	P	ASCII converted into HEX	S • D • n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*		
D •								*	*	*	*	*	*	*	*	*
n					*	*										

Instruction format:

[HEX S • D • n]

S·: Convert source device
D·: Convert destination device
n: bit of conversion, n=1~256

The instruction is used to convert ASCII code into HEX code, in 16-bit and 8-bit conversion modes. For example:

X000

[HEX D200 D100 K4]

16-bit conversion mode, when M8161=OFF, M8161 is shared by RS, ASCI, HEX, CCD and other instructions.

- After ASCII character data of high and low 8-bit of S· are converted into HEX data, it is sent to D· every 4 bits. Bit of the conversion is set with n.
- In HEX instruction, when the stored S· data is not ASCII code, it may be computation error and HEX conversion can not be executed. Especially M8161 is OFF, high 8-bit of S· needs to store ASCII code, please pay attention.

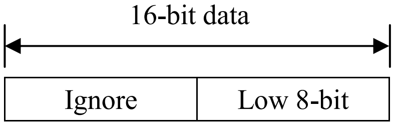
Conversion results:

S·	ASCII code	HEX conversion
D200 Low	30H	0
D200 High	41H	A
D201 Low	42H	B
D201 High	43H	C
D202 Low	31H	1
D202 High	32H	2
D203 Low	33H	3
D203 High	34H	4
D204 Low	35H	5

n \ D·	D102	D101	D100
1	Do not change · is 0		···0H
2			··0AH
3			·0ABH
4			0ABCH
5		···0H	ABC1H
6		··0AH	BC12H
7		·0ABH	C123H
8		0ABCH	1234H
9	···0H	ABC1H	2345H

8-bit conversion mode, when M8161=ON, (M8161 is shared by RS, ASCI, HEX, CCD and other instructions)

- After ASCII character data of low 8-bit of S· is converted into HEX data, it is sent to D· every 4 bits. Bit of the conversion is set with n.



Source address

Conversion results:

S·	ASCII code	HEX conversion
D200	30H	0
D201	41H	A
D202	42H	B
D203	43H	C
D204	31H	1
D205	32H	2
D206	33H	3
D207	34H	4
D208	35H	5

n \ D·	D102	D101	D100
1	Do not change		···0H
2	· is 0		··0AH
3			·0ABH
4			0ABCH
5		···0H	ABC1H
6		··0AH	BC12H
7		·0ABH	C123H
8		0ABCH	1234H
9	···0H	ABC1H	2345H

- When the input data is BCD code, after executing the instruction, conversion from BCD→ BIN shall be executed.

F84 CCD Check code

F 84		CCD	P	Check code	S • D • n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*		
D •								*	*	*	*	*	*	*		
n					*	*							*	*		

Instruction format:

[CCD S • D • n]

S·: Start device of information source
D·: Store destination results
n: number of data, n=1~256
It is the instruction for computing check code, in 8-bit and 16-bit conversion mode.
For example:

X000

[CCD D100 D0 K10]

16-bit conversion mode When M8161=OFF, (M8161 is shared by RS, ASCI, HEX, CCD and other instructions)

- Sum of specified high and low-bit data of point n as start by S· and horizontal check code are stored in D· and D·+1 devices.
- It is used for check of communication code.

Conversion of the above program:

S·	Data content
D100 Low	K100=01100100
D100 High	K111=01101111
D101 Low	K100=01100100
D101 High	K98=01100010
D102 Low	K123=01111011
D102 High	K66=01000010
D103 Low	K100=01100100
D103 High	K95=01011111
D104 Low	K210=11010010
D104 High	K88=01011000
Total	K1091
Horizontal check	10000101

Horizontal check: if number of 1 is odds, it is 1; it is even, it is 0.

8-bit conversion mode, when M8161=ON (M8161 is shared by RS, ASCI, HEX, CCD and other instructions)

- Sum of specified data of point n (only low 8-bit) as start by S· and horizontal check code are stored in D· and D·+1 devices.
- It is used for check of communication data.

16-bit data

Ignore	Low 8-bit

Source address

The above program conversion is stated below:

S·	Data content
D100	K100=01100100
D101	K111=01101111
D102	K100=01100100
D103	K98=01100010
D104	K123=01111011
D105	K66=01000010
D106	K100=01100100
D107	K95=01011111
D108	K210=11010010
D109	K88=01011000
Total	K1091
Horizontal check	10000101

F85 VRRD Readout of the potential value

F 85		VRRD	P	Readout of the potential value	S •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*										
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[VRRD   S •   D •]

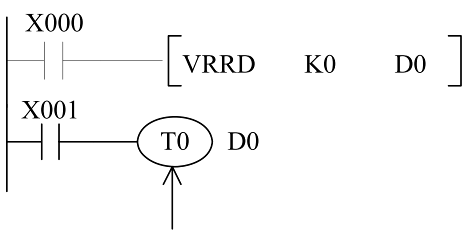
S·: number of potential

D·: device for storing scale of potential

※ 1: TP03SR machine type does not support the instruction.

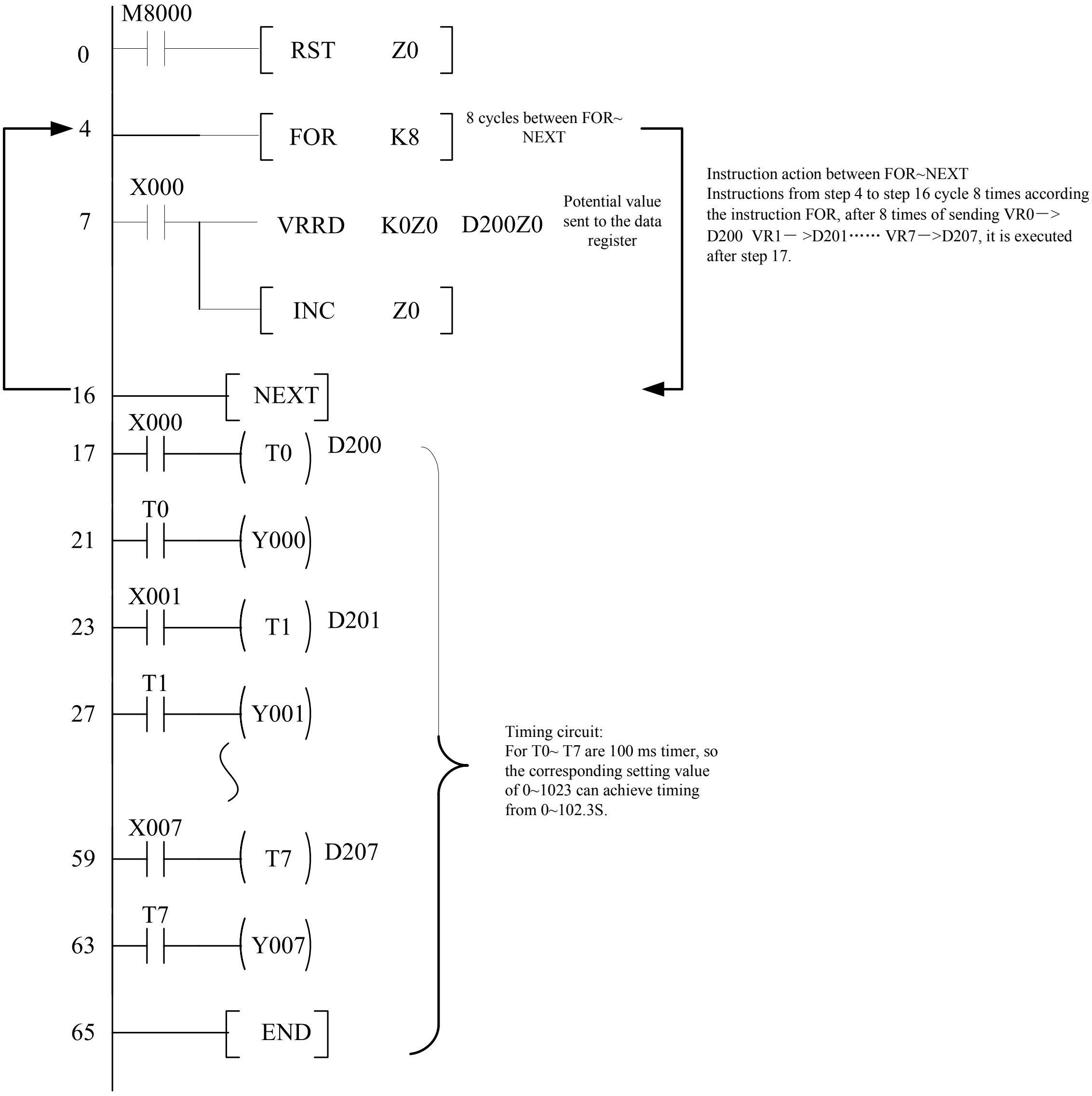
※ 2: The instruction VRRD is used to read scale values of the extended card (TP03-6AV), and the scale values are stored in D.

For example:



The case for analog timer

- Analog value of the potential No.0 is converted into decimal BIN value (0-1023), which is store in D0 and used as setting value of the timer.
- When setting value of the timer needs digits over 1024, the instruction FNC22 (MUL) shall be used. When the readout is taken as product, setting of the timer is done indirectly.
- The corresponding knobs VR0~VR7 and setting values of the instruction RRD are K0~K7. In the following program, it is decorated with the index buffer (Z0=0~7), and K0Z0=K0~K7. See the following chart:


M8000
0
RST Z0
4
FOR K8
8 cycles between FOR~ NEXT
X000
7
VRRD K0Z0 D200Z0
Potential value sent to the data register
INC Z0
Instruction action between FOR~NEXT
Instructions from step 4 to step 16 cycle 8 times according the instruction FOR, after 8 times of sending VR0—> D200 VR1— >D201…… VR7—>D207, it is executed after step 17.
16
NEXT
X000
17
T0
D200
T0
21
Y000
X001
23
T1
D201
T1
27
Y001
Timing circuit:
For T0~ T7 are 100 ms timer, so the corresponding setting value of 0~1023 can achieve timing from 0~102.3S.
X007
59
T7
D207
T7
63
Y007
65
END

F86 VRSC Scale of potential

F 86		VRSC	P	Scale of potential	S ·	D ·

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·					*	*										
D ·								*	*	*	*	*	*	*	*	*

Instruction format:

[VRSC S · D ·]

S·: Number of potential

D·: Location of storing scale of potential

※ 1: TP03SR machine type does not support the instruction.

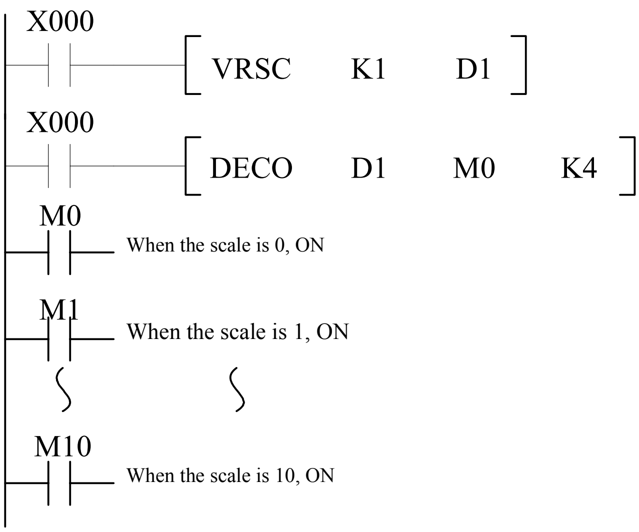
※ 2: The instruction VRSC is used to read scale values of the extended card (TP03-6AV), and the scale values are stored in D.

VRSC instruction is used to read 2 points of PLC host, numbered No.0, No.1 or 6 points of the function card, scale value of potential numbered No.2-No.7 (scale value 0-10) and the scale value is stored in D·.

For example:

- Scale 0-10 of the potential No.1 is stored in D1 with BIN value.
- When the knob rotates the scale, it is converted into integral by round down.

Use of rotating switch



- According to the potential scale 0~10, 1 point of the auxiliary relays M0~M10 is ON.
- With the instruction F41 DECO, auxiliary relays M0~M15 are occupied.

F87 MBUS computation

F 87	MBUS	MBUS computation	S m D n K

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S													*	*		
m					*	*							*	*		
D													*	*		
n					*	*							*	*		
K					0,1	0,1										

Instruction symbol:

[MBUS S m D n K]

S: Address of data sent
m: Length of data sent (0~255)
D: Address of data received
n: Length of data received(0~255)
K: Serial port selection, constant 0~1
0: communication port of RS485; 1:communication port of extended card of RS485 or RS232.

The instructions are communication ports (optional for all the models) for RS485 or RS232 extended cards, and RS485 communication port (only built-in for model H) to reach sending and receiving of serial Modbus protocol data. The above two ports can realize MBUS instruction communication simultaneously, and such function independently.
The instruction MBUS can be used as host station communication:
Stored instruction format of sent data register is HEX instruction code, including address, function code and communication data. The MBUS instruction sends according to setting mode like RTU mode plus CRC check code (2bytes); if it is ASCII mode, send plus start character (3AH), check code LRC (2bytes) and stop character (0DH+0AH) and the instruction code is converted into ASCII format to send to BUFF.
Stored instruction format in the received data register is address, function code and communication data, and start character, stop character and check code are not stored.

- Sending format of RS485 communication port data can be set with the special data buffer D8120. During executing of MBUS instruction, even settings of D8120 are changed, it will not be accepted.
- Sending format of communication port data of RS485 or RS232 extended card can be set with the special data buffer D8320. During executing of MBUS instruction, even settings of D8320 are changed, it will not be accepted.
- In the environment not for information sending, point of information sending is set at “K0”
- Although a large number of RS, MBUS, DTLK, RMIO and other communication instructions can be set in one program, only one communication instruction at one serial port at the same

time is driven, and OFF time shall be equal to or more than one scan cycle when switching.

Communication specifications

<Communication formats “D8120”, “D8320”>

In addition to Modbus protocol communication from the instruction FNC87 (MBUS), when the communication formats D8120 and D8320 are for other communication instructions or calculator connection, special data buffer can be used.

When the instruction FNC87 (MBUS) used, relevant setting for other communication instructions or calculator connection will be unavailable, please comply with the format setting of attentions.

Bit number	Name	Content	
		0(Bit OFF)	1(Bit ON)
B0	Length of data	7 bit	8 bit
B1 B2	Parity	B2,B1 (0,0): None (0,1): (ODD) (1,0): (EVEN)	
B3	Stop bit	1 bit	2 bit
B4 B5 B6 B7	Communication speed (bps)	B7,B6,B5,B4 (0,1,1,1):9,600 (1,0,0,0):19,200 (1,0,0,1):38,400 (1,0,1,0):57,600	B7,B6,B5,B4 (1,0,1,1):76,800 (1,1,0,0):128,000 (1,1,0,1):153,600 (1,1,1,0):307,200
B8~B12 *1	Unavailable		
B13	Modbus mode	(0) : RTU mode	(1) : ASCII mode
B14~B15*1	Unavailable		

*1:B8~B12，B14，B15 are setting items for other communication instructions or calculator connection. When the instruction FNC87(MBUS) is used, it must be used with “0”.

- Setting example of communication format

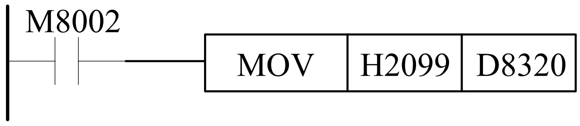
Length of data	8 bit
Parity	None
Stop bit	2 bit
Transmission speed	38,400 bps
Mode	ASCII mode
Serial port selection	RS485 expansion communication card

Communication setting in the table above shall be set according to the following program or serial communication of peripheral machines.

	b15			b12	b11			b8	b7			b4	b3			b0
D8320	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0	1

↓

D8320 2099H



Used special relay and data register:

a) When RS485 communication port is used:

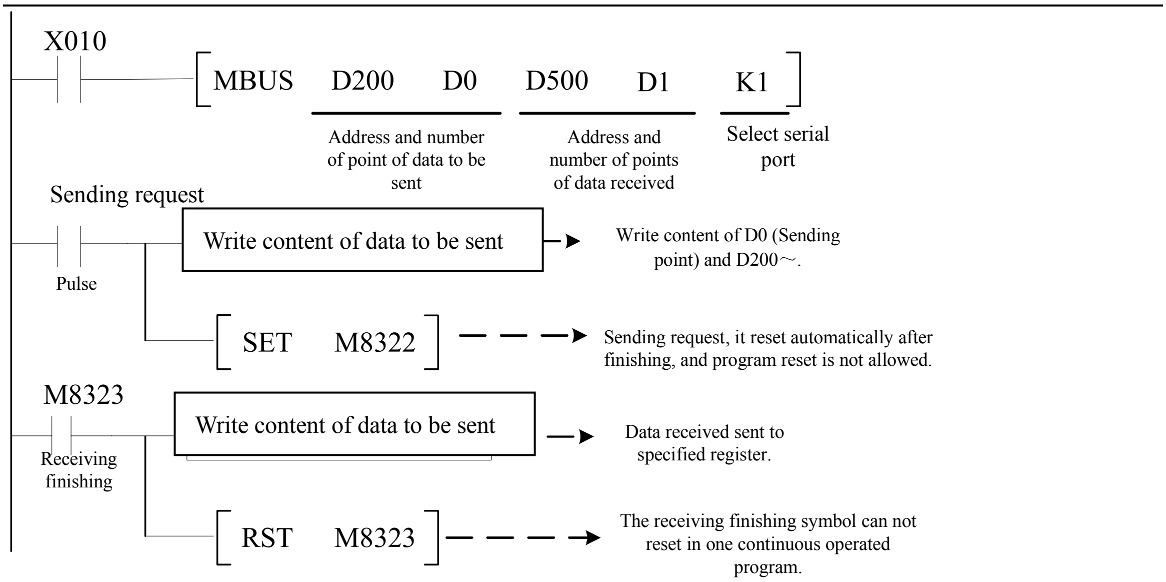
1) Sending waiting (M8121): if there is sending request during data receiving, 1 is set, after receiving and during data sending, it is cleared automatically.
2) Sending request (M8122): when M8122 is set by a pulse instruction under receiving waiting or receiving finishing, the data of length (m) is started to send from S. When sending finishes, M8122 resets automatically.
3) Receiving finishing (M8123): after receiving finishing, M8123 is ON, the data received is transmitted to specified register, then reset to enter receiving waiting.
4) Error indication (M8124): receiving error (error of CRC for RTU mode, error of LRC or stop character for ASCII mode).
5) Timeout judging (M8129): during data receiving, receiving will not be restarted, and the timeout output mark is ON, the receiving finishes; when M8123 is cleared, M8129 is automatically cleared.
6) Communication format setting (D8120): refer to MBUS instruction communication format described in previous text.
7) Number of residual data to be sent (D8122)
8) Number of data received (D8123)
9) Time of timeout judging (D8129): set the time for timeout judging (5~255)*10ms.

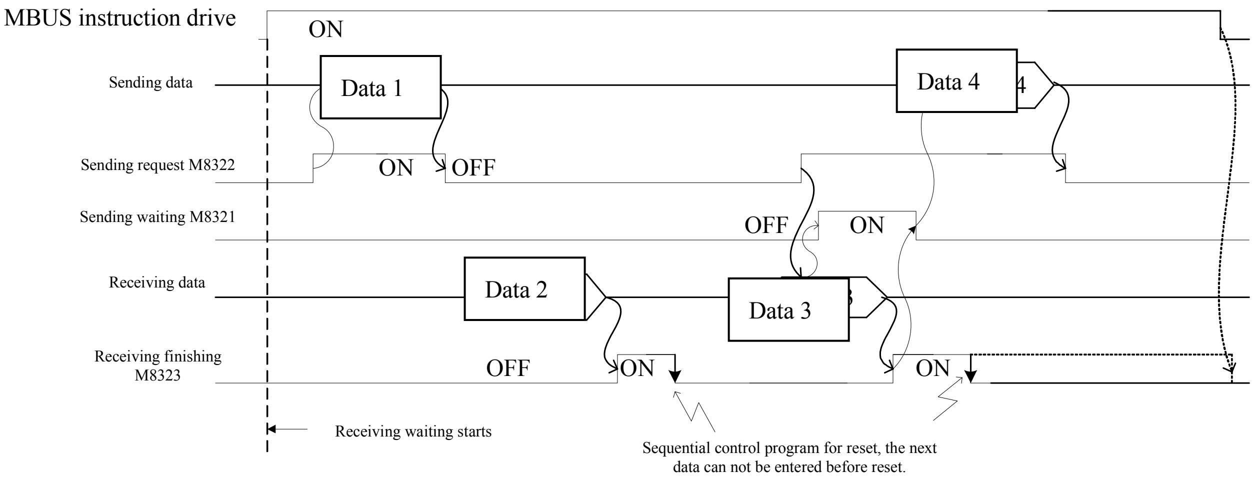
b) When communication ports of RS485 or RS232 extended card are used (refer to the above article on definitions):

1) Sending waiting (M8321)
2) Sending request (M8322)
3) Receiving finishing (M8323)
4) Timeout judging (M8329)
5) Error indication (M8324)
6) Communication format setting (D8320)
7) Number of residual data to be sent (D8322)
8) Number of data received (D8323)
9) Time of timeout judging (D8329)

Time sequence of sending and receiving

MBUS instruction does not stipulate first address and number of points of sent data from PLC, but also stipulates first address and maximum receiving point number of received data. See the following figure on sequence of sending and receiving data with MBUS instruction (for example, RS485 extended communication card is selected).





Sending request M8322

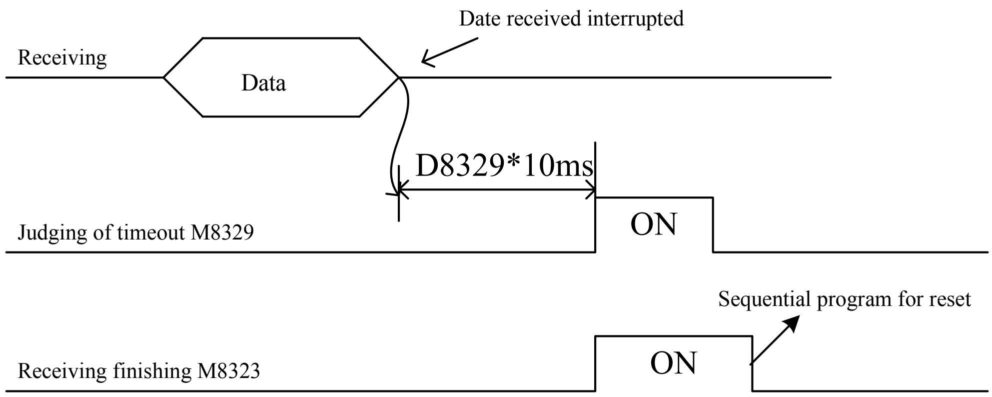
- When input condition X010 ON executes the MBUS instruction, it enters receiving waiting.
- In case of receiving waiting or receiving finishing, M8322 is set ON by pulse signal, continuous D0 data is sent from D200. When sending finishes, M8322 will RESET OFF automatically.

Receiving finishing M8323

- When receiving finishing mark M8323 ON, all the data received will be transferred to other assembly for storing, and M8323 RESET is OFF.
- If M8323 RESET is OFF, it enters receiving waiting. The input condition X010 ON executes the MBUS instruction, the controller enters receiving waiting.
- When setting (D1)=0, execute the MBUS instruction, M8323 will not act or enter receiving waiting. If after D1≧1, after M8323 ON is OFF, it enters receiving waiting.

Judging of timeout M8329

- When data receiving is interrupted, if the data receiving is not restarted from the time to setting time by D8329, it is regarded as timeout and M8329 will be ON and become receiving finishing. M8329 will reset automatically with reset of the program M8323.
  With the function, data receiving (ASCII mode) can be finished without stop character.



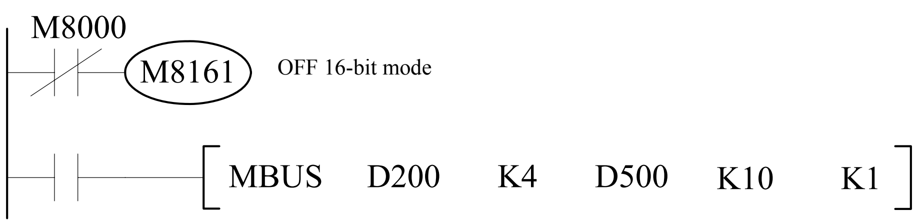
Time of judging timeout

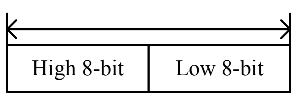
- Set using time of timeout judging.

When set the time, the setting value is X10ms, and the virtual value 5~255. When setting of D8329 exceeds the range, it becomes 50ms.

Take time of judging timeout 50ms as example.

<Processing of 16-bit data>When M8161=OFF, (M8161 is shared by RS, ASCI, HEX, CCD and other instructions)





16-bit data is divided into high and low
8-bit data for data sending and receiving.

Sending data（ASCII mode is different from RTU mode）

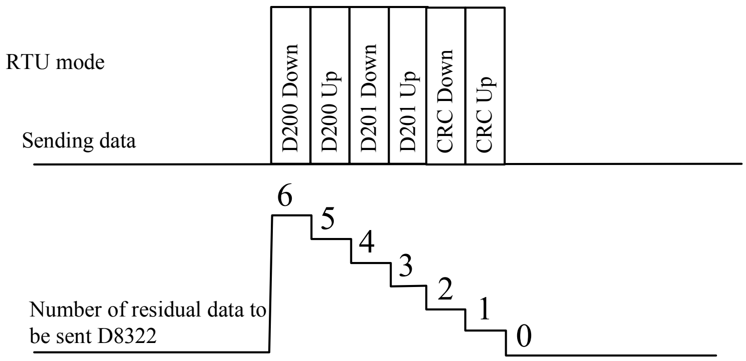
STX	D200 down	D200 up	D201 down	D201 up	Check code	ETX
Start character (3A)					LRC（ASCII）	Stop character (0D0A)
RTU mode （None）	↑ S．Specified start address No. M specified number of byte to be sent				CRC（RTU）	RTU mode （None）

Receiving data

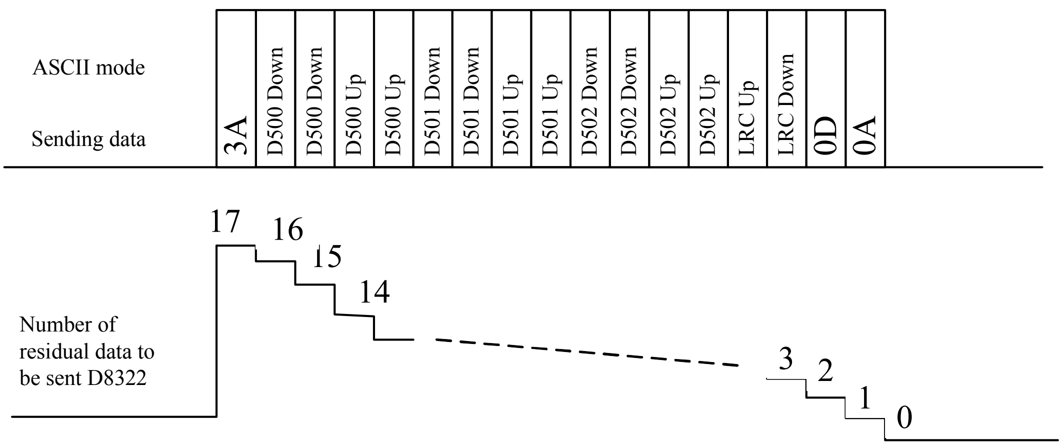
STX	D500 Down	D500 Up	D501 Down	D501 Up	D502 Down	D502 Up	Check code	ETX
Start character(3A)							LRC (ASCII)	Stop character (0D0A)
RTU (None)	↑ D．Specified start address No. It may not exceed maximum points specified by n, and the stop character EXT or points n receiving represents receiving finishing.						CRC (RTU)	

(3) Sending data and number of residual data to be sent

Under RTU mode

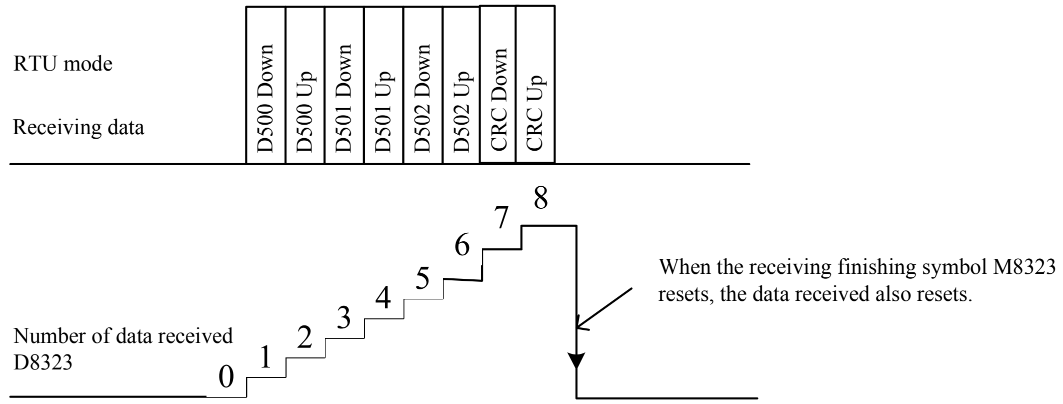


Under ASCII mode

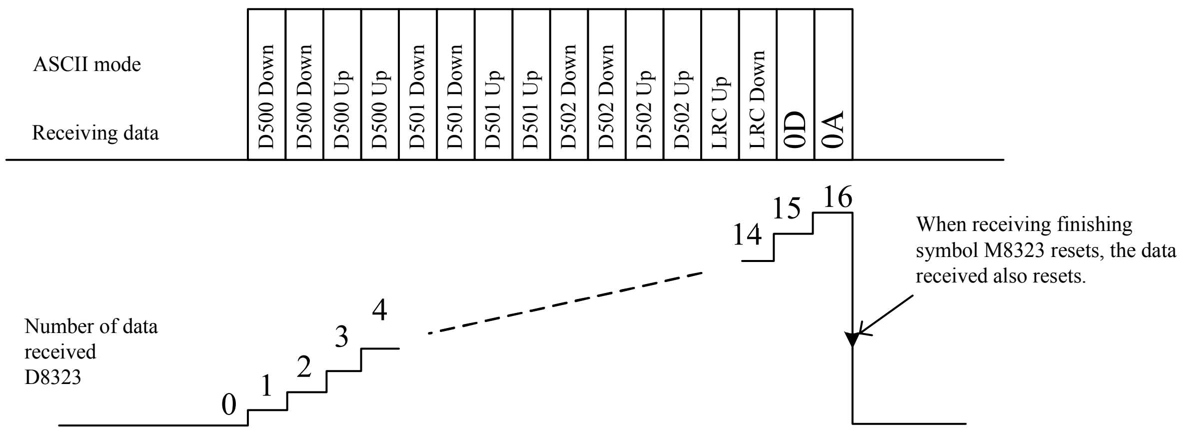


(4) Receiving data and number of data received

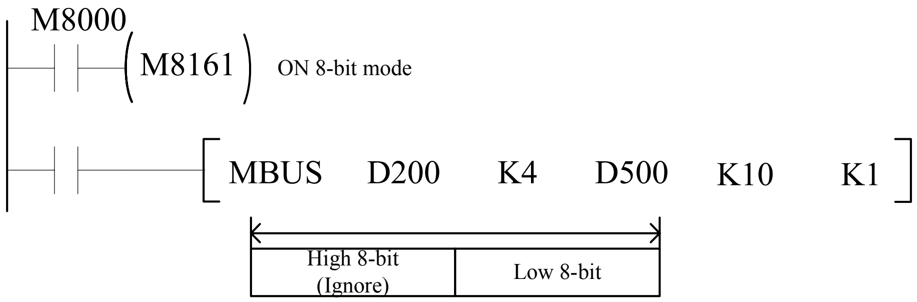
Under RTU mode:



Under ASCII mode



<Processing of 8-bit data (extension function)>When M8161=ON, (M8161 is shared by RS, ASCI, HEX, CCD and other instructions)



16-bit data ignores high 8-bit, and only low 8-bit is valid

Sending data（ASCII mode and RTU mode are different）

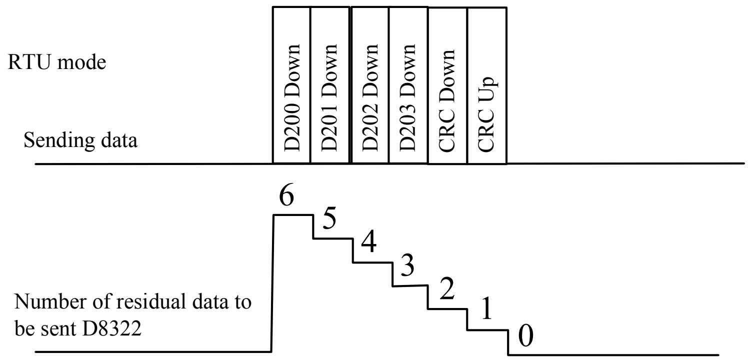
STX	D200 Down	D201 Down	D202 Down	D203 Down	Check code	ETX
Start character (3A)					LRC (ASCII)	Stop character (0D0A)
RTU mode (None)	↑ S. Specified start address No. M specified number of bytes to be sent				CRC (RTU)	RTU mode (None)

Receiving data

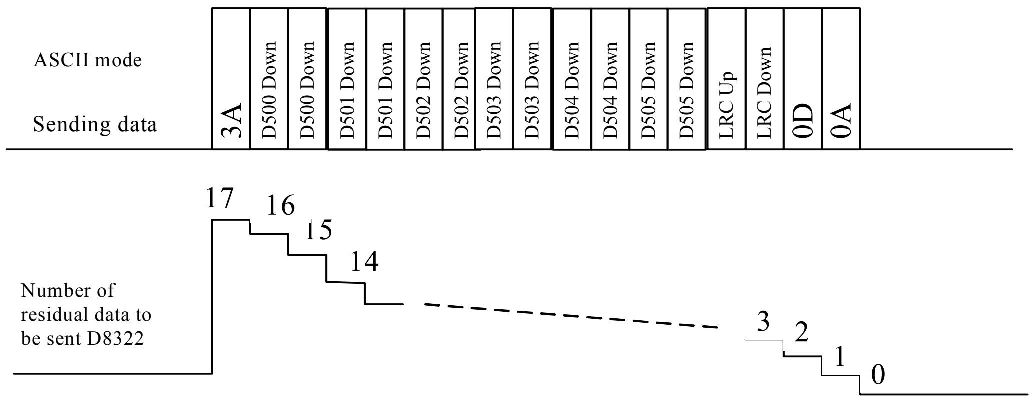
STX	D500 Down	D501 Down	D502 Down	D503 Down	D504 Down	D505 Down	Check code	ETX
Start character(3A)							LRC (ASCII)	Stop character (0D0A)
RTU(None)	↑ D. Specified start address No. It may not exceed maximum points specified by n, and the stop character EXT or points n receiving represents receiving finishing.						CRC (RTU)	

(1) Sending data and number of residual data to be sent

Under RTU mode

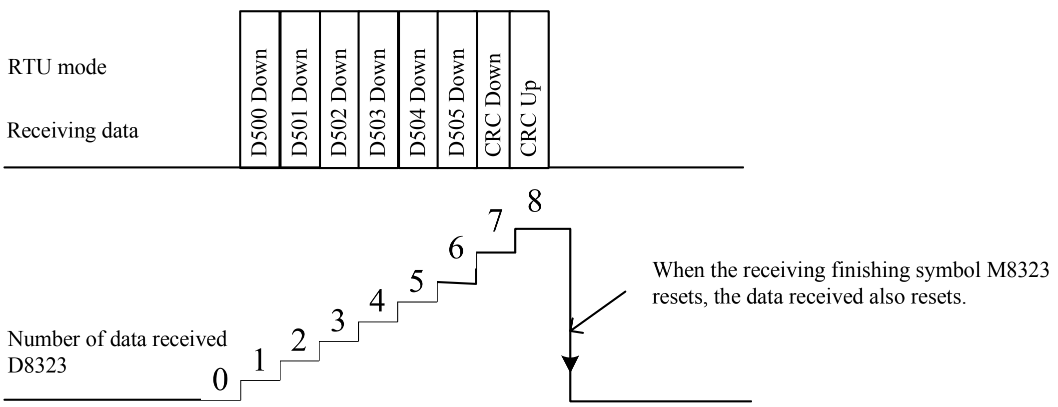


Under ASCII mode

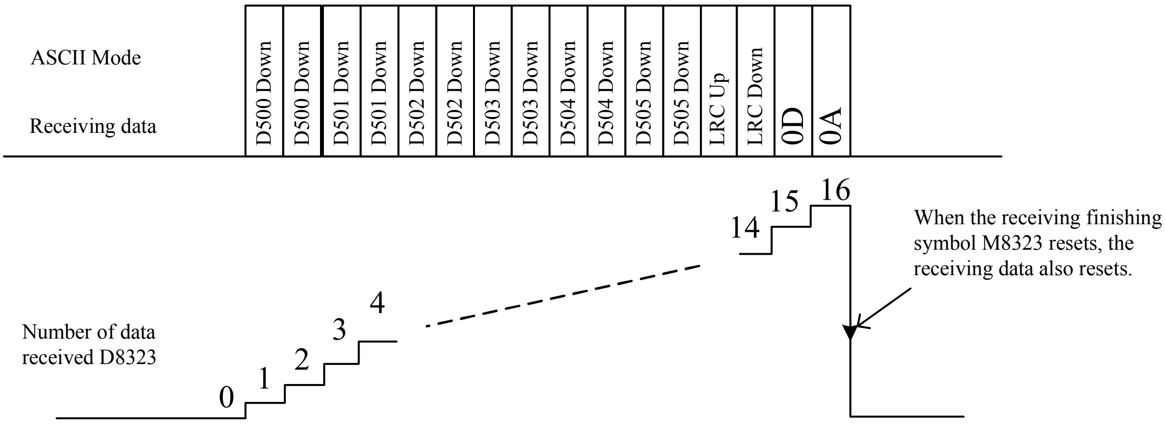


(2) Receiving data and number of data received

Under RTU mode:



Under ASCII mode



F88 PID computation

F 88		PID		PID computation	S1 S2 S3 D

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1													*			
S2													*			
S3													*			
D													*			

Instruction symbol

[PID S1 S2 S3 D]

S1: Destination value (SV)
S2: Current value (PV)
S3: Parameters, S3~S3+6 set control parameters.
D: Output value (MV)
※ 1: For TP03H/M machine type, S3 only uses D0~D7975.
※ 2: For TP03SR machine type, S3 only uses D0~D487

S1: Set current value SV
S2: Set measuring value PV
S3~S3＋6:Set control parameter

When executing the program, the computation results are stored in D.

The instruction is the special instruction for PID computation control, after the sampling time is up, the scanning time can reach PID computation. After setting all the parameters, PID instruction starts executing (before PID computation, parameter setting values for PID control must be written with MOV and other instructions), the results are temporarily stored in D. Content of D specifies temporary data storage without holding in case of power failure (if the register with holding in case of power failure, the register shall reset when RUN is added at the beginning of the program).

M8002

[RST D***]

D*** : It is the data register for holding in case of power failure

The program will occupy 25 data registers from S3 automatically, (the following page displays ACT setting of control parameters, when BIT1, BIT2 and BIT 5 are 0, only 20 data registers are occupied from S3).

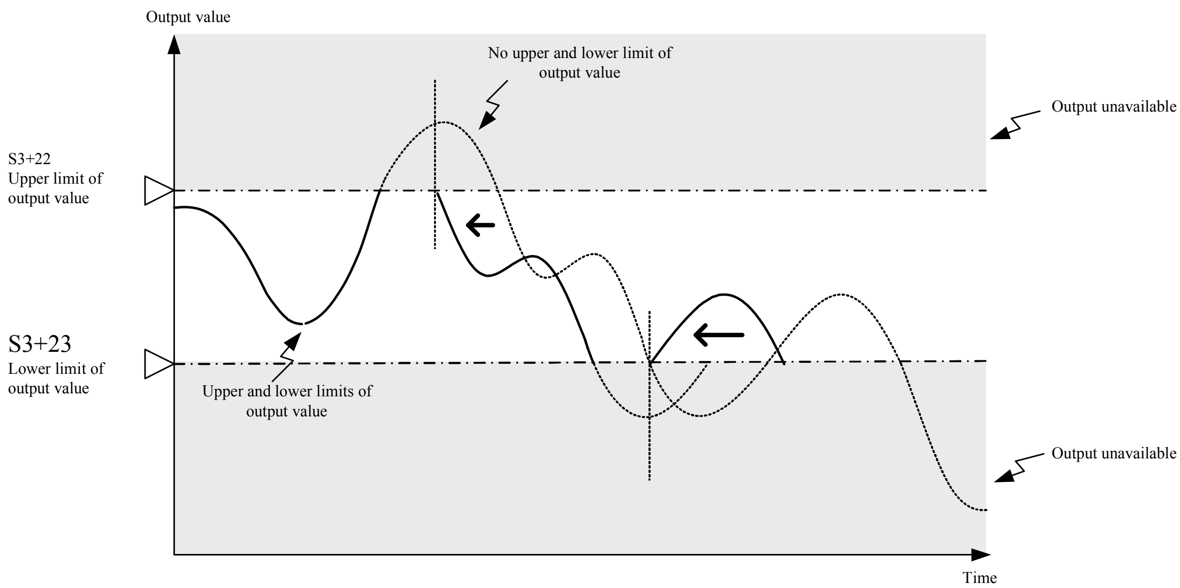
Parameter setting

Setting values of parameters for control must be written with MOV and other instructions before PID computation. In addition, when specifying data register in the zone for holding in case of power failure, the setting values are held after OFF of the PLC. Therefore, it requires no re-writing.

S3	Sampling time (Ts)	1~32767[ms](No shorter than the scanning time)
S3+1	Direction of action (ACT)	BIT0 0: positive action 1:Negative action BIT1 0: No input variation alarm; 1: Input variation alarm BIT2 0:No output variation alarm; 1:Output variation alarm BIT3 Unavailable BIT4 0:Auto turning function disable 1:Auto turning function enable BIT5 0: no upper and lower limits of the output values; 1: effective upper and lower limits of the output values. BIT6~BIT15 Unavailable BIT5 and BIT2 will not be ON simultaneously.
S3+2	Input wave filtering constant (a)	0~99[%] No input wave filtering for 0
S3+3	Proportional gain (Kp)	1~32767[%]
S3+4	Integration time (TI)	1~32767[x100ms] 0 is integration action
S3+5	Differential gain (KD)	0~100[%] 0 is no differential grain
S3+6	Differential time(TD)	1~32767[x10ms] 0 is no differential action
S3+7~S3+19	When PID computation is executed, it is used for internal processing.	
S3+20	Setting value of input variation (increasing)	0~32767(S3+1<ACT> bit1=1, it is effective)
S3+21	Setting value of input variation (decreasing)	0~32767(S3+1<ACT> bit1=1, it is effective)
S3+22	Setting value of output variation (increasing)	0~32767(S3+1<ACT> bit2=1,bit5=0, it is effective)
	Or setting value of upper limit of the output	–32768~32767(S3+1<ACT> bit2=1,bit5=0, it is effective)
S3+23	Setting value of input variation (decreasing)	0~32767(S3+1<ACT> bit2=1,bit5=0, it is effective)
	Or setting value of lower limit of the output	–32768~32767(S3+1<ACT> bit2=1,bit5=0, it is effective)
S3+24	Alarm output	BIT0 input variation (increasing) BIT1 input variation (decreasing) BIT2 output variation (increasing) BIT3 output variation (decreasing) (S3+1<ACT> bit1=1 or bit= 1, it is effective)

However, S3+20~S3+24 in S3+1<ACT>, bit1=1，bit2=2 or bit5=1 is occupied.
Although PID instruction can be executed for many times (no restriction on times of the loop), S3 or D used in the computation can not be used repeatedly.
PID instruction can be used in interruption of timer, sub-program, step charter and jump instruction.
Maximum error of sampling time T is between -(1 scan cycle+1ms) ~ +(1 scan cycle). When Ts is too small, such variation may lead to problems. At this time, please execute with constant scan mode or program in interruption of the timer.
If sampling time Ts≤1 computation cycle of PLC controller, the following computation abnormality can occur (K6740), and Ts= computation cycle for PID computation. At this time, it is suggested that PID instruction can be used in interruption (I6 口口~I8 口口) of the timer.
Input filter constant can be used to relieve variation of measuring values.
Increasing differential gain can be used to relieve dramatic variation of output values.
Direction of action (S3+1(ACT))

Direction of action [bit 0]
Direction of action of the system is specified with positive action and negative action.
Upper and lower limits settings of output values [bit5]
When settings of upper and lower limits of the output values are valid (S3+1(ACT) bit5=1), see the following figure on the output values. Use of the settings can be used to inhibit increasing of the PID control differential item. When the function is used, bit 2 of S3+1(ACT) is OFF.



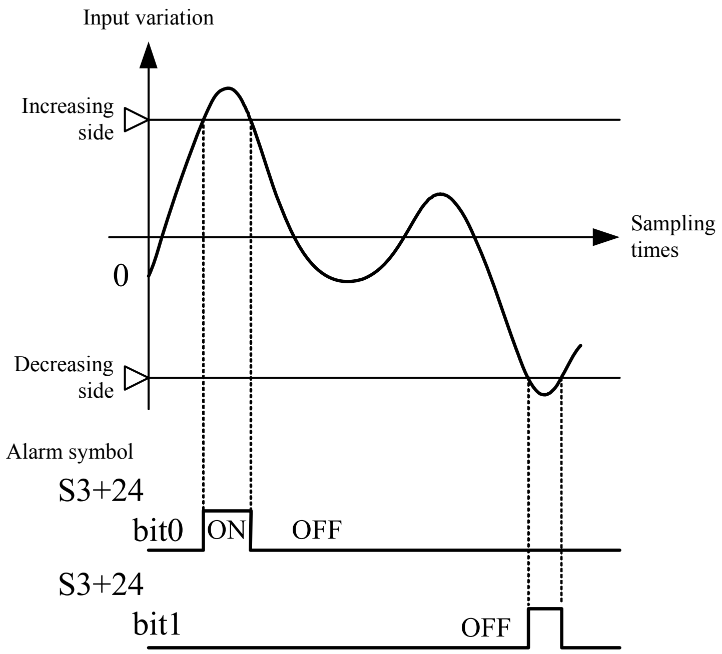
Alarm setting (Input variation and output variation) [bit1,bit2]
Make bit 1 and bit 2 of S3+1(ACT) ON, the operator can detect the input variation and output variation. The detection shall be executed according to values of S3+20~S3+23. If it exceeds the setting input variation, the bit elements of alarm symbols will be ON after PID instruction is executed (see the following figure).
However, when S3+21 and S3+23 are taken as alarm values, the setting values will be used as negative values. In addition, when the output variation is used for alarm, bit 5 of S3+1(ACT) should be OFF.

a) Variation

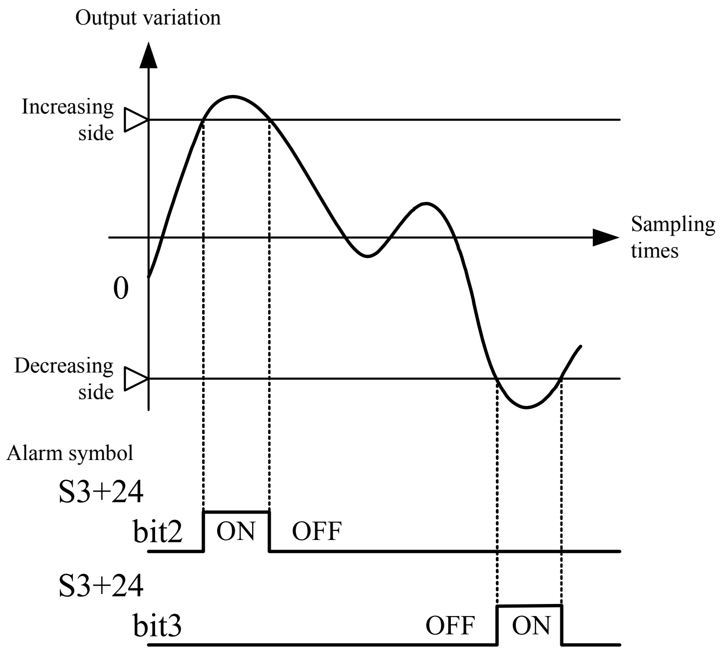
(Previous)- (Current) =Variation

b) Action of alarm symbols (S3+24)

i) Input variation (bit1=1)



ii)Output variation (bit2=1)

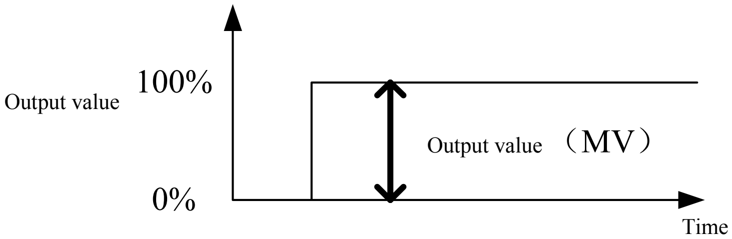


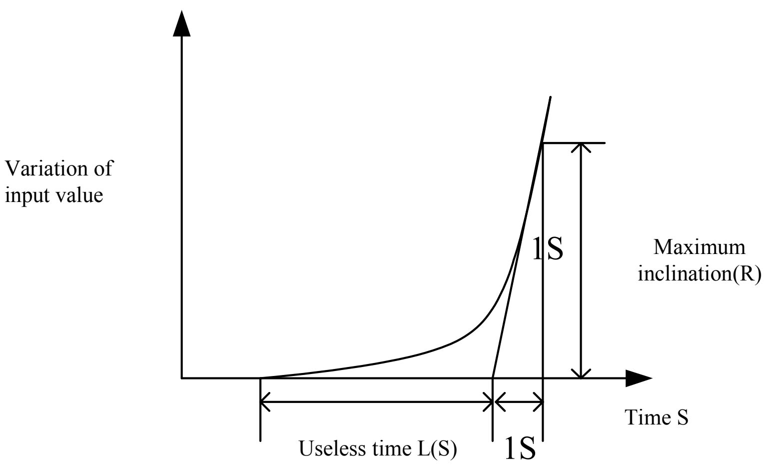
Mathematical method of 3 parameters of PID

To execute PID for better control, the optimum values of constants (parameters) for the control objects must be used.

There will be the optimum values of 3 constants to be obtained for PID (proportional gain Kp, integration time T1 and differential time TD).
Step response method is used for mathematical computation and the following is detailed description.

The step response method is to add 0-100% step output on the control system and judge features of actions from input variation (maximum inclination R and unavailable time L) to obtain 3 constants of PID.
Step 1 output can be obtained by 0-75% or 0-50%.
Features of actions





<Action features and 3 constants >

	Proportional grain Kp（%）	Integration time T1(×100ms)	Differential time TD(×100ms)
Only proportional control P	$\frac{1}{RL} \times$ Input values (MV)	——	——
PI control	$\frac{0.9}{RL} \times$ Input values (MV)	33L	——
PID control	$\frac{1.2}{RL} \times$ Input values (MV)	20L	50L

In case of error in setting values of the control parameters or PID computation, the computation error M8067 in ON. According to the error content, the following data created in D8067.

Code	Error	Processing state	Processing method
K6705	Operand of applied instruction is out of the object device	PID instruction computation stops	Please confirm content of control data
K6706	Range and data of device address No. of operand of applied instruction are out of the boundary		
K6730	Sampling time TS is out of the object boundary (Ts<0)		
K6732	Input wave filtering constant is out of the object boundary		
K6733	Proportional grain is out of the object boundary		
K6734	Integration time is out of the object boundary		
K6735	Differential grain is out of the object boundary		
K6736	Differential time is out of the object boundary		
K6740	Sampling time≤ Computation cycle	PID instruction computation continues	
K6742	Overflow of measuring value variation		
K6743	Overflow of deviated value		
K6744	Overflow of integration computation		
K6745	Overflow of differential grain leads to overflow of differential value		
K6746	Overflow of differential computation values		
K6747	Overflow of PID computation results		
K6750	SV-PVnf<150,or system is unstable	Auto turning stops	
K6751	Large Overshoot of the Set Value	Auto turning continues	
K6752	Large fluctuations during Auto tuning Set Process	Auto turning stops	

Key points:

Before executing PID computation, the correct measuring values shall be read into measuring value PV of PID, especially PID computation is executed for input value of input module of analog value, switching time shall be paid attention to.

Basic computation formula of PID instruction

PID computation is executed according to speed shape and differential shape of measuring values. PID executes computation formula of positive action or negative action according specified direction in S3.

In addition, values after S3 in computation, specify the used parameter content.

PID basic computation formula

Direction of action	PID computation method
Positive action	$\Delta MV = Kp\{(EV_n - EV_{n-1}) + \frac{Ts}{T1} EV_n + D_n\}$ $EV_n = PV_{nf} - SV$ $Dn = \frac{T_D}{Ts + \alpha_D . T_D}(-2PV_{nf-1} + PV_{nf} + PV_{nf-2}) + \frac{\alpha_D . T_D}{Ts + \alpha_D . T_D} . D_{n-1}$ $MV_n = \sum \Delta MV$
Reverse action	$\Delta MV = Kp\{(EV_n - EV_{n-1}) + \frac{Ts}{T1} EV_n + D_n\}$ $EV_n = SV - PV_{nf}$ $Dn = \frac{T_D}{Ts + \alpha_D . T_D}(2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{\alpha_D . T_D}{Ts + \alpha_D . T_D} . D_{n-1}$ $MV_n = \sum \Delta MV \Delta$

Mark interpretation
EVn: deviation of current sampling
EVn-1: deviation of one cycle
SV: destination value
PVnf: measuring value of current sampling (after wave filtering)
PVnf-1: measuring value one cycle before (after wave filtering)
PVnf-2: measuring value two cycles before (after wave filtering)
ΔMV: variation of the output
MVn : current operand
Dn : current differential quantity
Dn-1 : differential item one cycle before
Kp : proportional grain
Ts : sampling cycle
T1 : integration constant
TD : differential constant
αD: differential grain

PVnf is the value computed according to the read measuring value.
[Later measuring values PVnf]=PVn+L（PV_{nf-1}-PVn）
PVn: measuring value of current sampling
L: wave filtering coefficient
$PV_{nf-1:}$: measuring value one cycle before (after wave filtering)

F89 EPSC Scale of extended card

F 89		EPSC	P	Scale of extended card	S1 • S2 • S3 • D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*										
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
S3 •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[EPSC S1 • S2 • S3 • D •]

S1·: Channel No.(0~5)
S2·: Set minimum value of the scale
S3·: Set maximum value of the scale
D·: Store current scale value
※ 1: TP03SR machine type does not support the instruction.
※ 2: The instruction EPSC is used to read scale values of the extended card (TP03-2AI, TP03-2TI), and the scale values are stored in D.

For example:

X000
[EPSC K1 K0 K20 D1]

- Read channel 1 of the extended card to store in D1, the minimal value is 0, the maximal value is 20.
- When the knob is rotating on the rotating scale, it become integral from 0-20 by rounding down.

F110~F119，F120-F129，F130~F137 floating-point instructions

F110~F119，F120-F129，F130~F137 floating

F110 ECMP Comparison of binary floating-point number

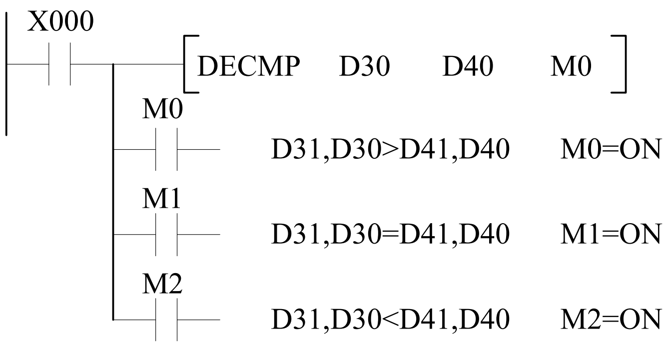
F 110	D	ECMP	P	Comparison of binary floating-point number	S1 • S2 • D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1 •					*	*							*	*			*
S2 •					*	*							*	*			*
D •		*	*	*													

Instruction format:

[ECMP S1 • S2 • D •]

S1•: Comparison value 1 of binary floating-point number
S2•: Comparison value 2 of binary floating-point number
D•: Comparison results, occupying continuous 3 points
Example:



When X000 is OFF, even the instruction ECMP is not executed, M0-M2 keeps the state of X000 not OFF.

Compare S1 and S2 floating-point number values, and the results (ON or OFF) are stored in 3-bit started with D. When the constants K and H are specified as source data, it is converted into binary floating-point number automatically for processing.

F111 EZCP Comparison of binary floating-point number zone

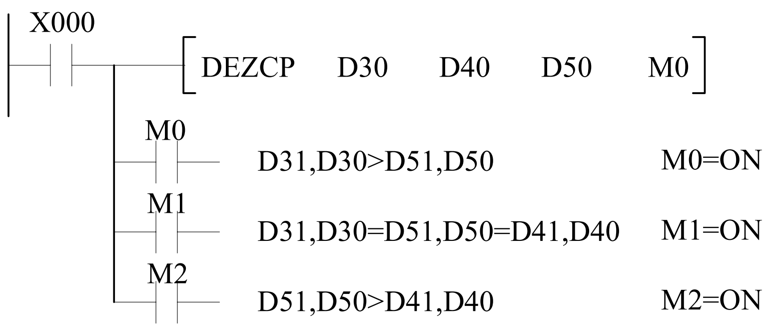
F		EZCP		Comparison of binary floating-point number zone	S1 • S2 • S • D •
111	D		P		

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1 •					*	*							*	*			*
S2 •					*	*							*	*			*
S •					*	*							*	*			*
D •		*	*	*													

Instruction format:

[EZCP S1 • S2 • S • D •]

S1•: Lower limit value of binary floating-point number for zone comparison
S2•: Upper limit value of binary floating-point number for zone comparison
S•: Comparison value of binary floating-point number
D•: Comparison result, occupying continuous 3 points
Note: when setting, S1•≤S2•**;** when S1•>S2•**,** value of S2• shall be taken as the same value of S1•**.**
Example:



When X000 is OFF, even the instruction ECMP is not executed, M0-M2 keeps the state of X000 not OFF.
Compare content of S and S+1 and range of the two specified binary floating-point number S1 and S2. The results (ON or OFF) are stored in the 3-bit started with D.
When the constants K and H are specified as source data, it is converted into binary floating-point number automatically for processing.

F112 EMOV Moving of binary floating-point number

F 112	D	EMOV	P	Moving of binary floating-point number	S •	D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S •													*	*			*
D •													*	*			

Instruction format:

[EMOV S • D •]

S•: Binary floating point data (transfer source) or device number storing data
D•: Device number receiving floating point data.
Content (binary floating-point number) of the transfer source (S•+1, S•) are transferred to (D•+1, D•). A real number (E) can be directly specified as S•.
Example:

X000 [DEMOV D10 D0]

(D11, D10) → (D1, D0)

X000 [DEMOV E-1.23 D0]

-1.23→ (D1, D0)

F118 EBCD Binary floating-point number→Decimal floating-point number

F		EBCD		Binary floating-point number→ Decimal floating-point number	S • D •
118	D		P		

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •													*	*		
D •													*	*		

Instruction format:

[EBCD S • D •]

S•: Data source (binary floating-point number)
D•: Conversion results (decimal floating-point number)

Example:

X000 [DEBCD D50 D40]

Convert binary floating-point number in the element specified by the source data into decimal floating-point number and store it in the destination address.

Binary floating-point number

D51	D50

Mantissa section 23-bit, the index section 8-bit and the symbol 1-bit

↓

Decimal floating-point number

Index	Mantissa
D41	D40

Mantissa section Index section

$D40 \times 10^{D41}$

The value of Decimal floating-point number = [Mantissa D40] $X10^{[\mathit{Index}D41]}$

Mantissa D40= (1,000~9,999) or 0

Index D41= -41 ~ +35

The decimal computation is executed based on the binary floating-point value in the PLC, for the binary floating-pint value, it is hard to judge, so it shall be converted into binary floating-point value. It is easy for the peripheral equipment to monitor.

F119 EBIN Decimal floating-point number→Binary floating-point number

F		EBIN		Decimal floating-point number→ Binary floating-point number	S • D •
119	D		P		

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •													*	*		
D •													*	*		

Instruction format:

[EBIN S • D •]

S•: Data source (decimal floating-point number)

D•: Conversion result (binary floating-point number)

Example:

X000

[DEBIN D50 D40]

The decimal floating-point number in the element D50 specified by source data is converted into binary floating-point numbers, which are stored in D41 and D40.

Decimal floating-point number

Index	Mantissa
D51	D50

Mantissa Index

D50×10^{D51}

The value of Decimal floating-point number =[Mantissa D50] X10$^{\text{[IndexD41]}}$

Mantissa D50= (1,000~9,999) or 0

Index D51= -41 ~ +35

Binary floating-point number

D41	D40

The mantissa section is 23-bit, the index section is 23-bit and the symbol is 1-bit.

F120 EADD Binary floating-point number addition computation

F		EADD		Binary floating-point number addition computation	S1 • S2 • D •
120	D		P		

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1 •					*	*							*	*			*
S2 •					*	*							*	*			*
D •													*	*			

Instruction format:

[EADD S1 • S2 • D •]

S1•: Summand
S2•: Addend

D•: Sum Log_e

Number on the buffer specified by S1• plus number on the buffer specified by S2•, the sum is stored in the buffer specified by D• and the addition computation is executed with binary floating-point number.
If number specified by S1• or S2• is constant, the instruction will convert the constant into binary floating-point number for addition computation.
S1• and S2• can specify the same buffer number, under general conditions, the pulse executing instruction DEADD P is used.
Example:

X000
[DEADD D50 D40 D10]

(D51, D50)+ (D41, D40) → (D11, D10)
When X000 ON, the binary floating-point numbers (D51, D50) plus binary floating-point numbers (D41, D40), the numbers are stored in (D11, D10).

F121 ESUB Subtraction computation of binary floating-point numbers

F		ESUB		Subtraction computation of binary floating-point numbers	S1 • S2 • D •
121	D		P		

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1 •					*	*							*	*			*
S2 •					*	*							*	*			*
D •													*	*			

Instruction format:

[ESUB S1 • S2 • D •]

S1•: Minuend
S2•: Subtrahend
D•: Difference
Number on the buffer specified by S1• minus number on the buffer specified by S2•, the difference is stored in the buffer specified by D• and the subtraction computation is executed with binary floating-point number.
If number specified by S1• or S2• is constant, the instruction will convert the constant into binary floating-point number for subtraction computation.
S1• and S2• can specify the same buffer number, under general conditions, the pulse executing instruction DESUB P is used.
Example:

X000
[DESUB P D50 D40 D10]

(D51, D50) — (D41, D40) → (D11, D10)
When X000 ON, the binary floating-point numbers (D51, D50) minus binary floating-point numbers (D41, D40), the numbers are stored in (D11, D10).

F122 EMUL Multiplication computation of binary floating-point numbers

F 122	D	EMUL	P	Multiplication computation of binary floating-point numbers	S1 • S2 • D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1 •					*	*							*	*			*
S2 •					*	*							*	*			*
D •													*	*			

Instruction format:

[EMUL S1 • S2 • D •]

S1•: Multiplicand

S2•: Multiplier

D•: Product

Number on the buffer specified by S1• multiple number on the buffer specified by S2•, the product is stored in the buffer specified by D• and the multiplication computation is executed with binary floating-point number.

If number specified by S1• or S2• is constant, the instruction will convert the constant into binary floating-point number for multiplication computation.

S1• and S2• can specify the same buffer number, under general conditions, the pulse executing instruction DEMUL P is used.

Example:

X000

[DEMUL P D50 D40 D10]

(D51, D50) × (D41, D40) → (D11, D10)

When X000 ON, the binary floating-point numbers (D51, D50) multiple the binary floating-point numbers (D41, D40), the product is stored in (D11, D10).

F123 EDIV Division computation of binary floating-point number

F		EDIV		Division computation of binary floating-point number	S1 • S2 • D •
123	D		P		

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1 •					*	*							*	*			*
S2 •					*	*							*	*			*
D •													*	*			

Instruction format:

[EDIV S1 • S2 • D •]

S1•: Dividend

S2•: Divisor (the number can not be 0, otherwise, it may be regarded as error computation, the instruction will not execute)

D•: Quotient

Number on the buffer specified by S1• divides by number on the buffer specified by S2•, the quotient is stored in the buffer specified by D• and the quotient computation is executed with binary floating-point number.

If number specified by S1• or S2•, the instruction can convert the constant into binary floating-point number for division computation.

S1• and S2• can specify the same buffer number, under general conditions, the pulse executing instruction DEDIV P is used.

Example:

X000

[DEDIV P D50 D40 D10]

(D51, D50) ÷ (D41, D40) → (D11, D10)

When X000 ON, the binary floating-point numbers (D51, D50) divide by the binary floating-point numbers (D41, D40); the quotient is stored in (D11, D10).

F124 EXP Exponent arithmetic computation

F 124	D	EXP	P	Exponent arithmetic computation	S •	D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S •													*	*			*
D •													*	*			

Instruction format:

[EXP S • D •]

S•: Head device number storing binary floating point data used in exponential operation.

D•: c

The exponent of (S•+1, S•) is calculated, and the operation result is stored to (D•+1, D•).

A real number can be directly specified as S•.

In the exponential operation, the base (e) is set to “2.71828”.

$e^{(S\bullet+1,S)}$ → (D•+1, D•)

Error condition:

An operation error in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

When the operation result is outside the following range (error code:K6706)

$2^{-126} \leq$|Operation result|< 2^{128}

Example:

(D11, D10) Binary floating-point number exponent value

↓

(D21, D20) Power value as binary floating-point number

Points:

1) The operation result becomes less than “2^{128}” when the BCD value set in D10 is 88 or less because of “Log_e^{128} =88.7”. If a value “89” or more is set, an operation error occurs and this operation will not be executed.
2) Conversion from natural logarithm into common logarithm
   In the CPU, operations are executed in natural logarithm.
   For obtaining a value in common logarithm, specify a common logarithm value divided by “0.4342945” in (S·+1, S·).

$$10^x = e^{\frac{x}{0.4342945}}$$

F125 LOGE Natural logarithm computation

F / 125	D	LOGE	P	Natural logarithm computation	S •	D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S •													*	*			*
D •													*	*			

Instruction format:

[LOGE S • D •]

S•: Head device number storing binary floating point data used in the natural logarithm operation.
D•: Head device number storing binary floating point data used in exponential operation.

Natural logarithm [logarithm whose base is “e (2.71828)”] of (S•+1, S•) is calculated, and the operation result is stored to (D•+1, D•). A real number can be directly specified as S•.

$Log_e(S\cdot+1, S\cdot) \rightarrow$ (D•+1, D•)

Only a positive value can be set in (S•+1, S•). (The natural logarithm operation cannot be executed for a negative value.)

Error condition:
An operation error in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.
When a negative value or 0 is specified in S•. (Error code:K6706)

Example:

X000

[DLOGE D10 D20]

(D11, D10) Binary floating-point number Power value

↓

(D21, D20) Exponent value as binary floating-point number

F126 LOG10 Common logarithm computation

F 126	D	LOG10	P	Common logarithm computation	S •	D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S •													*	*			*
D •													*	*			

Instruction format:

[LOG10 S • D •]

S•: Head device number storing binary floating point data used in the common logarithm operation.

D•: Head device number storing binary floating point data used in exponential operation.

Natural logarithm [logarithm whose base is “10”] of (S•+1, S•) is calculated, and the operation result is stored to (D•+1, D•). A real number can be directly specified as S•.

$Log_{10}(S\cdot+1, S\cdot) \rightarrow$ (D•+1, D•)

Only a positive value can be set in (S•+1, S•). (The common logarithm operation cannot be executed for a negative value.)

Error condition:

An operation error in the following case; The error flag M8067 turns ON, and the error code is stored in D8067.

When a negative value or 0 is specified in S•. (Error code:K6706)

Example:

X000

[DLOG10 D10 D20]

(D11, D10) Binary floating-point number Power value

↓

(D21, D20) Exponent value as binary floating-point number

F127 ESQR Square root of binary floating-point number

F		ESQR		Square root of binary floating-point number	S •	D •
127	D		P			

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S •					*	*							*	*			*
D •													*	*			

Instruction format:

[ESQR S • D •]

S•: source element to be square root(Only valid for positive numbers, if it is negative numbers, it is regarded as computation error, M8067 ON)

D•: square root

Square root is executed for numbers on the buffer specified by S•**,** the square root is stored in the buffer specified by D• and the square root computation is executed with binary floating-point numbers.

If number specified by S• is constant, the instruction will convert the constant into binary floating-point number for square root computation.

Example:

X000

[DESQR D50 D40]

$\sqrt{(D51, D50)} \rightarrow (D41, D40)$

When X000=ON, the binary floating-point numbers (D51, D50)are square root, and the square root is stored in (D41, D40).

F128 ENEG Binary floating-point numbers NEG computation

F		ENEG		Binary floating-point numbers NEG computation	D •
128	D		P		

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
D •													*	*			

Instruction format:

[ENEG D •]

D•: Head device number storing binary floating data whose sign is to be inverted.

The sign of binary floating point stored in (D•+1, D•) is inverted, and the negation result is stored to (D•+1, D•).

Example:

M0

[DENEGP D10]

(D11, D10)→ (D11, D10)

1.2345 → -1.2345

F129 INT Binary floating-point → BIN integral conversion

F		INT		Binary floating-point → BIN integral conversion	S •	D •
129	D		P			

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S •					*	*							*	*			
D •													*	*			

Instruction format:

[INT S • D •]

S•: Source to be converted

D•: Conversion result

The buffer content specified by S• is converted into BIN integral with the form of binary floating-point form and temporarily stored in the buffer by D•, and decimal of BIN integral is ignored.

Action of the instruction is opposite with the instruction F49 FLT.

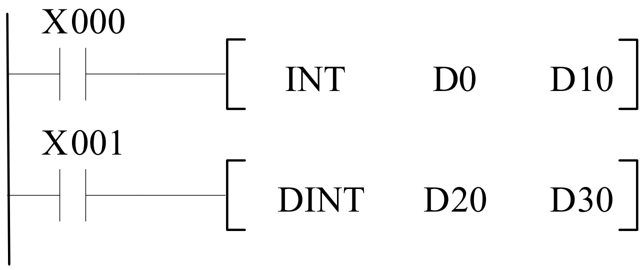
If the conversion result is 0, the zero symbol M8012=ON. If the decimal point is ignored in the conversion result, the borrowing symbol M8021=ON.

If the conversion results exceed the following range, the carrying symbol M8022=ON.

16-bit instruction: -32,768~32,767

32-bit instruction: -2,147,483,648~2,147,483,647

Example:



When X000=ON, the binary decimals(D1, D0)are converted into BIN integral, the result is stored in D10, decimal of the BIN integral is ignored.

When X001=ON, the binary decimals (D21, D20) are converted into BIN integral and the result is stored in (D31, D30), decimal of the BIN integral is ignored.

F130 SIN Computation of floating-point SIN

F 130	D	SIN	P	Computation of floating-point SIN	S •	D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S •													*	*			*
D •													*	*			

Instruction symbol:

[SIN S • D •]

S•: Specified RAD value

D•: Computation result of CSIN

RAD value specified by S• is equal to (angle×π/180), and SIN value is obtained and stored in buffer specified by D•.

Example:

X000

[DSIN D10 D20]

(D11, D10) Binary floating-point number RAD value

↓

(D21, D20) SIN value as binary floating-point number

F131 COS Computation of floating-point COS

F 131	D	COS	P	Computation of floating-point COS	S •	D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S •													*	*			*
D •													*	*			

Instruction format:

[COS S • D •]

S•: Specified RAD value

D•: Computation result of COS

RAD value 0(angle×π/180) specified by S• is obtained and stored in buffer specified by D•.

Example:

X000

[DCOS D10 D20]

(D11, D10) Binary floating-point number RAD value

↓

(D21, D20) COS value as binary floating-point number

F132 TAN Computation of floating-point TAN

F 132	D	TAN	P	Computation of floating-point TAN	S •	D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S •													*	*			*
D •													*	*			

Instruction format:

[TAN S • D •]

S•: Specified RAD value

D•: Computation result of TAN

RAD value (angle×π/180) specified by S• is obtained and stored in buffer specified by D•.

Example:

X000 [DTAN D10 D20]

(D11, D10) Binary floating-point number RAD value

↓

(D21, D20) TAN value as binary floating-point number

F133 ASIN Computation of floating-point ASIN

F		**ASIN**		ASIN Computation of floating-point ASIN	S •	D •
133	D		P			

	Bit element				Word element													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S													*	*			*
D													*	*			

Instruction symbol:

[ASIN S • D •]

S: Specify data source of ASIN

D: Computation result of ASIN

Content ARC SIN (inverse function of SIN) is taken as binary floating-point number and stored in D.

-1≤S<1

Example

(D11, D10) Binary floating-point number RAD value

↓

(D21, D20) ASIN value as binary floating-point number

F134 ACOS Computation of floating-point ACOS

F		ACOS		Computation of floating-point ACOS	S •	D •
134	D		P			

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S													*	*			*
D													*	*			

Instruction symbol:

[ACOS S • D •]

S: Specified ACOS data source

D: ACOS computation results

Content ARC COS (inverse function of COS) specified by S is taken as binary floating-point number and stored in D.

-1≤S<1

Example:

X000

[D ACOS D10 D20]

(D11, D10) Binary floating-point number RAD value

↓

(D21, D20) ACOS value taken as binary floating-point number

F135 ATAN Computation of floating-point ATAN

F		ATAN		Computation of floating-point ATAN	S •	D •
135	D		P			

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S													*	*			*
D													*	*			

Instruction symbol:

[ATAN S • D •]

S: Specify ATAN data source

D: ATAN computation results

Content ARC TAN (inverse function of TAN)specified by S is taken as binary floating-point number and stored in D.

-π/2~π/2

Example:

X000

[D ATAN D10 D20]

(D11, D10) Binary floating-point number RAD value

↓

(D21, D20) ATAN value as binary floating-point number

F136 RAD Angle converted into radian

F 136	D	RAD	P	Angle converted into radian	S • D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S													*	*			*
D													*	*			

Instruction symbol:

[RAD S • D •]

S and D are binary floating-point forms.

The instruction realizes conversion from angle unit into radian unit.

Example:

X000

[DRAD D10 D20]

F137 DEG Radian converted into angle

F 137	D	DEG	P	Radian converted into angle	S •	D •

	Bit element				Word element												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S													*	*			*
D													*	*			

Instruction format:

[DEG S • D •]

S and D are binary floating-point forms.

The instruction realizes conversion from angle unit to radian unit.

F147 Conversion instruction for upper and lower characters

F147 Conversion instruction for upper and lower characters

Function No.	Memory view	Name	Page
F147	SWAP	Conversion of upper and lower characters	1

F147 Conversion instruction for upper and lower characters

F147 SWAP Conversion of upper and lower characters

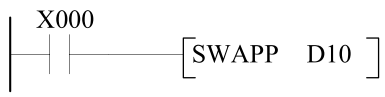
F 147	D	SWAP	P	Conversion of upper and lower characters	S・

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S								*	*	*	*	*	*	*	*	*

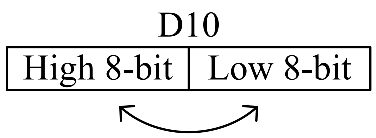
Instruction format:

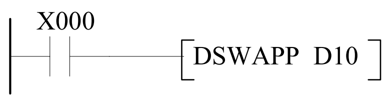
[SWAP S・]

S·: for executing exchange unit for the upper and lower 8-bit.

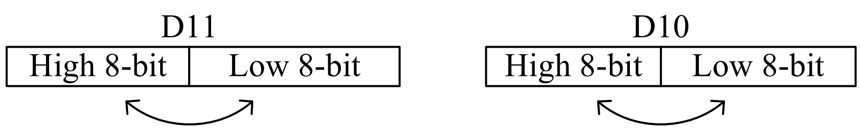


For 16-bit instruction, exchange the low 8-bit and high 8-bit.





For 32-bit instruction, exchange the low 8-bit and hig 8-bit.



It is noted that when the instruction is used as continuous executing instruction, the computation cycle needs conversing.
Function of the instruction is the same as extended function of F17 XCH.

F156~F159 Location instruction

F156~F159 Locating instruction

F156 ZRN Origin return

F 156	D	ZRN		Origin return	S1 • S2 • S3 • D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
S3 •	*	*	*	*												
D •		*														

Instruction format:

[ZRN S1 • S2 • S3 • D •]

- When executing relative location control of F158 (DRVI) and absolute location control of F159 (DRVA), PLC increases or decreases the current value with its self-generated positive/negative pulse, which are stored in current value register (Y000: [D8141, D8140], Y001: [D8143,D8142]). Therefore, the mechanical location always keeps, in case of power failure of the PLC, the location will disappear. For power on and initial operation, origin return must be executed and data of origin location of mechanical action shall be written in advance.

  S1·: Speed of origin return

  Specify speed of origin return start.

  [16-bit instruction]: 10~32,767 (Hz)

  [32-bit instruction]: 10~100,000 (Hz)

  S2·: Crawling speed

  Speed of low speed after DOG signal is ON.

  10~32,767(Hz)

  S3·: DOG signal

  Specify DOG signal input (a contact-point input)

  When specifying element out of the input relay (X), affected by mathematical performance cycle of the PLC, it may lead to greater deflection of the origin location.

  D·: Object number of pulse output

  Only Y000 or Y001 is specified. Output of the controller must be in the form of transistor.

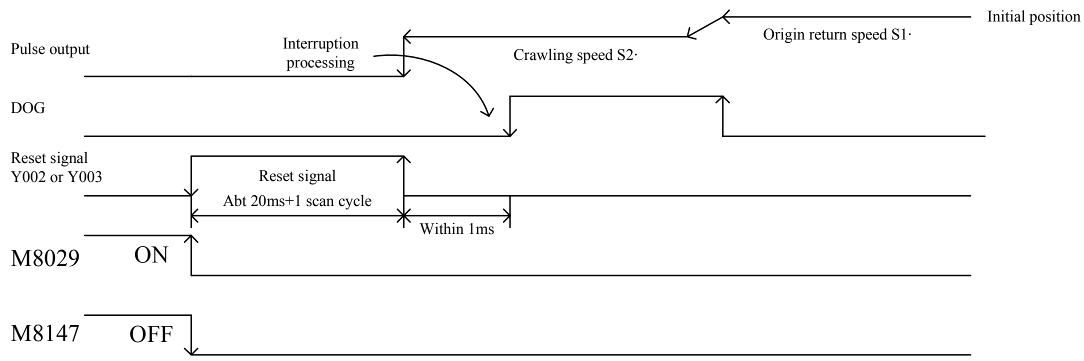
※ 1: TP03SR machine type does not support the instruction.

Output function of reset signal

- When M8140 is ON and origin return finishes, it outputs reset signal to the servo motor.
- Output number of the reset signal is determined by pulse output numbers.

  Pulse output[Y000]→Clear output[Y002]

  Pulse output[Y001] →Clear output[Y003]



Action of origin return

- Origin return shall be executed according to the following sequence.

  ① After the drive instruction, it move at the speed of origin return speed S1·.

  - During origin return, if the instruction drive contact-point is OFF, it will not decelerate but stop.
  - After the instruction drive contact-point is OFF, when pulse output monitors (Y000: M8147, Y001: M8148) are on, it will not receive another drive of the instructions.

  ② When DOG signal becomes ON from OFF, it is decelerated to the crawling speed S2·.

  ③ When DOG signal becomes OFF from ON, and pulse output stops, it writes 0 into the current value registers (Y000: [D8141, D8140], Y001: [D8143, D8142]). In addition, when M8140 (output performance for resetting signal) is ON, reset signal is output simultaneously. After that, when the finishing flag sign (M8029) acts, the pulse output monitors (Y000: [M8147], Y001: [M8148]) become OFF.

Relevant element address numbers

[D8141 (upper location), D8140 (lower location)]: pulse number output to Y000. (32-bit used)

[D8143 (upper location), D8142 (lower location)]: pulse number output to Y001 (32-bit used)

[M8145]: Y000 pulse output stops (Stop immediately)

[M8146]: Y001 pulse output stops (Stop immediately)

[M8147]: Y000 pulse output monitor (BUSY/READY)

[M8148]: Y001 pulse output monitor (BUSY/READY)

Attentions

- For it does not have the function of DOG searching, the origin return action shall be stated from front end of the DOG signal.
- In origin return, data of current value registers (Y000: [D8141, D8140], Y001: [D8143, D8142]) will change towards decreasing.

F157 PLSV Variable speed pulse

F 157	D	PLSV		Variable speed pulse	S • D1 • D2 •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D1 •		*														
D2 •		*	*	*												

Instruction format:

[PLSV S • D1 • D2 •]

- The instruction is the variable speed pulse output instruction with direction of rotation.

  S·: output pulse frequency

  [16-bit instruction]：1~32,767 (Hz), -1~-32,768 (Hz)

  [32-bit instruction]：1~100,000 (Hz), -1~-100,000 (Hz)

  D1·: Object number of pulse output

  Only Y000 or Y001 is specified. Output of the controller must be in the form of transistor.

  D2·: Output object number for rotating direction signal

  Actions are available for corresponding S·: when S· is positive, it is ON; when S· is negative, it is OFF.

※ 1: TP03SR machine type does not support the instruction.

- Even in pulse output, the output pulse frequency S· can be changed.
- For there is no acceleration or deceleration at start/stop, if buffering is required, instructions F67 (RAMP)and etc. can be used to change value of the pulse frequency S·.
- During pulse output, when the instruction drive contact-joint is OFF, it will not decelerate but stop.

Relevant element address number

[D8141 (upper location), D8140 (lower location)]: pulse number output to Y000. It will re reduced for reverse operation/ (32-bit used).

[D8143 (upper location), D8142 (lower location)]: pulse number output to Y001. It will re reduced for reverse operation/ (32-bit used).

[M8145]: Y000 pulse output stops (Stop immediately)

[M8146]: Y001 pulse output stops (Stop immediately)

[M8147]: Y000 pulse output monitor (BUSY/READY)

[M8148]: Y001 pulse output monitor (BUSY/READY)

Attentions

- Pay attention to driving time on instructions.

F158 DRVI Relative location control

F 158	D	**DRVI**		Relative location control	S1· S2· D1· D2·

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1·					*	*	*	*	*	*	*	*	*	*	*	*
S2·					*	*	*	*	*	*	*	*	*	*	*	*
D1·		*														
D2·		*	*	*												

Instruction format:

—[DRVI S1· S2· D1· D2·]

- The instruction is used for single-speed location control with relative drive mode.

S1·: Output pulse number (relative specifying)
[16-bit instruction]: -32,768~+32,767
[32-bit instruction]: -2,147,483,648~+2,147,483,647

S2·: Output pulse frequency
[16-bit instruction]: 10~32,767 (Hz)
[32-bit instruction]: 10~100,000 (Hz)
It can not be less than the frequency in the formula in the next page.

D1·: Object number of the pulse output
Only Y000 or Y001 is specified. Output of the controller must be in the form of transistor.

D2·: Output object number for rotating direction signal

Actions are available for corresponding S·: when S· is positive, it is ON; when S· is negative, it is OFF.

※ 1: TP03SR machine type does not support the instruction.

- Corresponding locations of output pulse number are described below:
  Output to Y000: [D8141 (high location), D8140 (low location)] (32-bit used)
  Output to Y001: [D8143 (high location), D8142 (low location)] (32-bit used)
  When it rotates oppositely, content of current value register will be decreased.
- During the instruction executing, even content of the operand is changed, it can not be reflected to current operation and it will take affect in the next instruction drive.
- During instruction executing, when the instruction drive contact-joint is OFF, it stops deceleration. At this time, the finishing flag sign M8029 does not act.
- The so-called absolute control means taking the origin location as basic point, move with a certain pulse number with rotating direction.
- Minimum frequency of actual output pulse frequency is determined by the following formula.

Minimum frequency of output pulse frequency =

$$\sqrt{\text{Maximum speed[D8147, D8146]Hz} \div (2 \times (\text{Time for acceleration and deceleration[D8148]ms} \div 1000))}$$

Relevant element interpretation

[D8145]: Basic speed for executing the instructions FNC158 (DRVI), FNC159(DRVA) and etc.

When controlling the step motor, considering resonance zone and self-starting frequency of the step motor for setting speed.

Range of setting: less than 1/10 of maximum speed (D8147, D8146).

When it exceeds the range, it is reduced to 1/10 of maximum speed automatically.

[D8147 (high location), D8146(low location)]:

Maximum speed for executing the instructions FNC158 (DRVI) and FNC159 (DRVA).

Output pulse frequency specified by S2· must be less than the maximum speed.

Range of setting: 10~100000(Hz)

[D8148]: Time of acceleration and deceleration for executing the instructions FNC158 (DRVI) and FNC159 (DRVA).

Time of acceleration and deceleration means the required time for reaching the maximum speed (D8147, D8146).

Therefore, when the output pulse frequency S2· is lower than the maximum speed (D8147 D8146), the actual time of acceleration and deceleration will be shortened.

Range of setting: 50~5000(ms)

[M8145]: Y000 pulse output stops (Stop immediately)

[M8146]: Y001 pulse output stops (Stop immediately)

[M8147]: Y000 pulse output monitoring (BUSY/READY)

[M8148]: Y001 pulse output monitoring (BUSY/READY)

F159 DRVA Absolute location control

F 159	D	DRVA		Absolute location control	S1 • S2 • D1 • D2 •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
D1 •		*														
D2 •		*	*	*												

Instruction format:

[DRVA S1 • S2 • D1 • D2 •]

- The instruction is used for executing single-seed location control with absolute drive mode.

S1·: Destination location (absolute specifying)

[16-bit instruction]: -32,768~+32,767

[32-bit instruction]: -2,147,483,648~+2,147,483,647

S2·: Output pulse frequency

[16-bit instruction]: 10~32,767 (Hz)

[32-bit instruction]: 10~100,000 (Hz)

D1·: Object number of the pulse output

Only Y000 or Y001 is specified. Output of the controller must be in the form of transistor.

D2·: Output object number for rotating direction signal

Actions are available for differences of corresponding S· and current locations: when the difference is positive, it is ON; and is OFF for negative.

※ 1: TP03SR machine type does not support the instruction.

- Corresponding relative locations of output pulse quantity S1· are stated below:

  Output to Y000: [D8141 (high location), D8140 (low location)] (32-bit used)

  Output to Y001: [D8143 (high location), D8142 (low location)] (32-bit used)

  When it rotates oppositely, content of current value register will be decreased.
- During the instruction executing, even content of the operand is changed, it can not be reflected to current operation and it will take affect in the next instruction drive.
- During instruction executing, when the instruction drive contact-joint is OFF, it stops deceleration. At this time, the finishing flag sign M8029 does not act.
- The so-called absolute control means taking the origin location as basic point, move with a certain pulse number with rotating direction.
- Minimum frequency of actual output pulse frequency is determined by the following formula.

  Minimum frequency of output pulse frequency =

$$\sqrt{\text{max } speed[\text{D8147, D8146}]\text{Hz} \div (2 \times (a\text{cc/dec time}[\text{D8148}]\text{ms} \div 1000))}$$

Relevant element interpretation

[D8145]: Basic speed for executing the instructions FNC158 (DRVI), FNC159(DRVA) and etc.

When controlling the step motor, considering resonance zone and self-starting frequency of the step motor for setting speed.

Range of setting: less than 1/10 of maximum speed (D8147, D8146).

When it exceeds the range, it is reduced to 1/10 of maximum speed automatically.

[D8147(high location), D8146(low location)]:

Maximum speed for executing the instructions FNC158 (DRVI) and FNC159 (DRVA).

Specified output pulse frequency of S2· must be less than the maximum speed.

Range of setting: 10~100000(Hz)

[D8148]: Time of acceleration and deceleration for executing the instructions FNC158 (DRVI) and FNC159(DRVA).

Time of acceleration and deceleration means the required time for reaching the maximum speed (D8147, D8146).

Therefore, when the output pulse frequency S2· is lower than the maximum speed (D8147 D8146), the actual time of acceleration and deceleration will be shortened.

Range of setting: 50~5000(ms)

[M8145]: Y000 pulse output stops (Stop immediately)

[M8146]: Y001 pulse output stops (Stop immediately)

[M8147]: Y000 pulse output monitoring (BUSY/READY)

[M8148]: Y001 pulse output monitoring (BUSY/READY)

F160~F167 Clock computation

F160~F167 Clock computation

F160 TCMP Clock data comparison

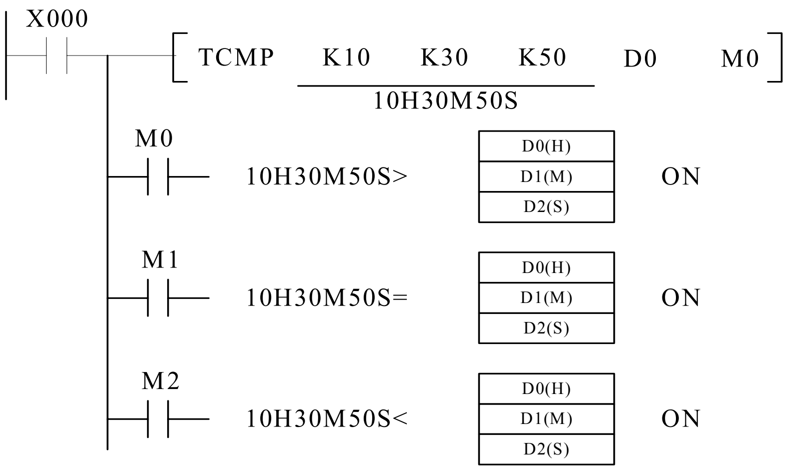
F 160		TCMP	P	Clock data comparison	S1 • S2 • S3 • S • D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •					*	*	*	*	*	*	*	*	*	*	*	*
S2 •					*	*	*	*	*	*	*	*	*	*	*	*
S3 •					*	*	*	*	*	*	*	*	*	*	*	*
S •											*	*	*	*		
D •		*	*	*												

Instruction format:

[TCMP S1 • S2 • S3 • S • D •]

S1·: Specify "H" of the comparison time, range of specifying (0~23)
S2·: Specify "M" of the comparison time, range of specifying (0~59)
S3·: Specify "S" of the comparison time, range of specifying (0~59)
S·: Specify "H" of the time data, range of specifying (0~23)
S·+1: Specify "M" of the time data, range of specifying (0~59)
S·+2: Specify "S" of the time data, range of specifying (0~59)
D·: Comparison results, occupying 3 continuous bit elements.
※ 1: TP03SR machine type does not support the instruction.
Example:



Even X000 is used to stop executing the instruction TCMP, M0~M2 shall keep the state of X000 not OFF.

- Compare time of source data (S1·, S2·, S3·)and 3-point time data started with S·, 3-point ON/OFF state is output according to the comparison results.

F160~F167 Clock computation

F161 TZCP Clock zone comparison

F 161		TZCP	P	Clock zone comparison	S1 • S2 • S • D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •											*	*	*	*		
S2 •											*	*	*	*		
S •											*	*	*	*		
D •		*	*	*												

Instruction format:

[TZCP S1 • S2 • S • D •]

S1·: Specify lower limit value of the comparison time

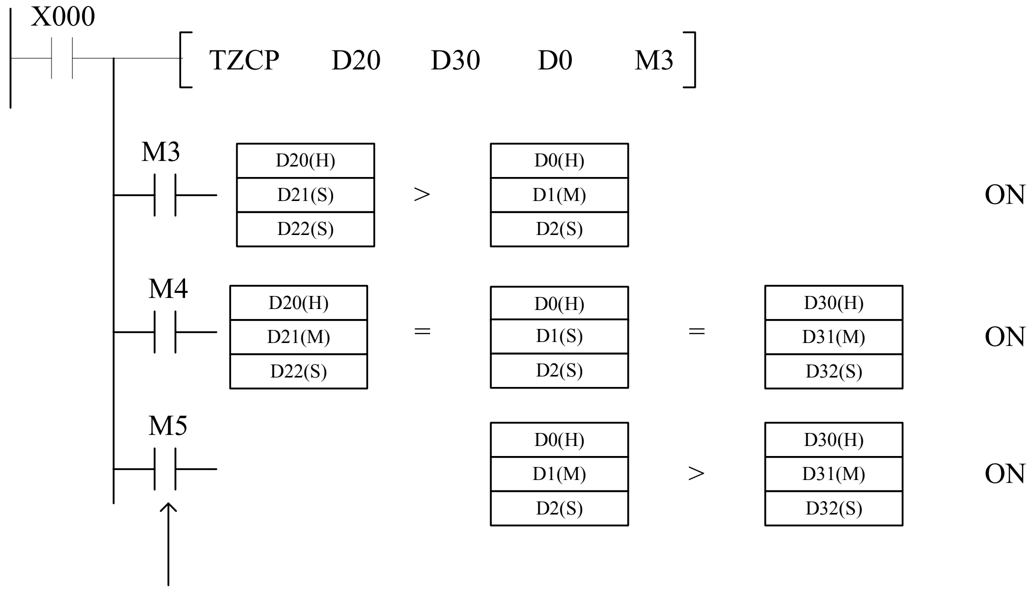
S2·: Specify upper limit value of the comparison time

S·: Specify time data

D·: Comparison result, occupying 3 continuous bit elements

※ 1: TP03SR machine type does not support the instruction

Example



Even X000 is used to stop the instruction TZCP, M3~M5 shall keep the state of X000 not OFF.

- Compare 3-point time data started with S· and time zones specified by the upper and lower points S1· and S2·. 3-point bit element ON/OFF state started with D· is output according to the comparison results.

  S1·, S1· +1, S1· +2: “H”, “M” and “S” of lower setting value of the comparison time.

  S2·, S2· +1, S2· +2, “H”, “M” and “S” of upper setting value of the comparison time

  S·, S ·+1, S ·+2: “H”, “M” and “S” of the specified time

  The lower limit value S1·may not be more than the upper limit value S2·. When the lower limit value S1·>the upper limit value S2·, the lower limit value S1· shall be taken as the upper and lower limit values for comparison.

F160~F167 Clock computation

F162 TADD Clock data plus computation

F 162		TADD	P	Clock data plus computation	S1 • S2 • D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •											*	*	*	*		
S2 •											*	*	*	*		
D •											*	*	*	*		

Instruction format

[TADD S1 • S2 • D •]

S1·: Summand of time
S2·: Addend of time
D·: Sum of time
※ 1: TP03SR machine type does not support the instruction
Example:

X000
[TADD D10 D20 D30]

D10 10(H)		D20 3(H)		D30 13(H)		
D11 30(M)	+	D21 10(M)	→	D31 40(M)		
D12 10(S)		D22 5(S)		D32 15(S)		

10H30M10S 3H10M5S 13H40M15S

- H, M and S of calendar data specified by S1· plus H, M and S of calendar data specified by S2·, and the results are stored in H, M and S of buffer specified by D·.
- If the result is over 24H, the carrying flag sign M8022=ON. The computation result minus 24H, and the final result is obtained and stored.
  Example:

18(H)		10(H)		4(H)
10(M)	+	20(M)	→	30(M)
30(S)		5(S)		35(S)

- If the computation result is 0 (0 H 0 M 0S), and the zero flag sign M1020=ON.

F160~F167 Clock computation

F163 TSUB Minus computation of clock data

F 163		TSUB	P	Minus computation of clock data	S1 •	S2 •	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •											*	*	*	*		
S2 •											*	*	*	*		
D •											*	*	*	*		

Instruction format:

[TSUB S1 • S2 • D •]

S1·: Minuend of time
S2·: Subtrahend of time
D·: Difference of time
※ 1: TP03SR machine type does not support the instruction.
Example:

X000
[TSUB D10 D20 D30]

D10 10(H)		D20 3(H)		D30 7(H)
D11 30(M)	-	D21 10(M)	→	D31 20(M)
D12 10(S)		D22 5(S)		D32 5(S)
10H30M10S		3H10M5S		7H20M5S

- H, M and S of calendar data specified by S1· minus H, M and S of calendar data specified by S2·, and the results are stored in H, M and S of buffer specified by D·.
- When the result is less than 0, the carrying flag sign M8022=ON. The computation result plus 24H, and the final result is obtained and stored.
  Example:

5(H)		18(H)		11(H)
20(M)	-	10(M)	→	10(M)
40(S)		5(S)		35(S)
5H20M40S		18H10M5S		11H10M35S

- If the computation result is equal to 0 (0 H 0 M 0 S), the zero flag sign M1020=ON

F166 TRD Clock data reading

F 166	TRD	P	Clock data reading	D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D •											*	*	*	*		

Instruction format:

[TRD D •]

D·: start device after readout of current time of the calendar, occupying 7 points

※ 1: TP03SR machine type does not support the instruction.

Example:

X000 [TRD D0]

- Read real-time clock data of the controller according to the following format. The reading source is the special data registers (D8013~D8019)for holding the clock data.

	Element	Item	Clock data		Element	Item
Special data register for real-time clock	D8018	Y (Solar calendar)	2000~2099	→	D0	Y (Solar calendar)
	D8017	M	1~12	→	D1	M
	D8016	D	1~31	→	D2	D
	D8015	H	0~23	→	D3	H
	D8014	M	0~59	→	D4	M
	D8013	S	0~59	→	D5	S
	D8019	Week	0(Sun)~6(Sat)	→	D6	Week

F160~F167 Clock computation

F167 TWR Writing-in of clock data

F 167		TWR	P	Writing-in of clock data	S •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •											*	*	*	*		

Instruction format:

[TWR S •]

S·: Store start device of new setting value of the calendar

※ 1: TP03SR machine type does not support the instruction.

Example:

X000

[TWRP D10]

- Write data for setting clock into the real-time clock of the controller. To write clock data, 7-point element started with element address numbers specified by S· must be specified.

	Element	Item	Clock data		Element	Item	
Data for clock setting	D10	Y (Solar calendar)	2000~2099	→	D8018	Y (Solar calendar)	Real-time clock for special data register
	D11	M	1~12	→	D8017	M	
	D12	D	1~31	→	D8016	D	
	D13	H	0~23	→	D8015	H	
	D14	M	0~59	→	D8014	M	
	D15	S	0~59	→	D8013	S	
	D16	Week	0(Sun)~6(Sat)	→	D8019	Week	

F170~F172 Peripheral equipment

F170~F172 Pheripheral equipment

Function No.	Memory view	Name	Page
F170	GRY	Conversion of BIN—GRY codes	1
F171	GBIN	Conversion of GRY—BIN codes	2

F170 GRY Conversion of BIN—GRY codes

F 170	D	GRY	P	Conversion of BIN—GRY codes	S • D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[GRY S • D •]

S·: Source device:

D·: Device for storing grey code

Example:

X000

[GRY K1234 K3Y10]

BIN 1234 (b15 … b0)

b15															b0
0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0

↓

GRY 1234

Y23			Y20	Y17							Y10
0	1	1	0	1	0	1	1	1	0	1	1

- Convert BIN data to grey code and send data.
- 32-bit grey code conversion can be executed in maximum.
- As for values of S·, it is only valid in the following range.
  16-bit computation: 0~32,767
  32-bit computation: 0~2,147,483,647

F171 GBIN Conversion of GRY—BIN codes

F / 171	D	GBIN	P	Conversion of GRY—BIN codes	S • D •

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •					*	*	*	*	*	*	*	*	*	*	*	*
D •								*	*	*	*	*	*	*	*	*

Instruction format:

[GBIN S • D •]

S·: Source device
D·: Device for storing reversal conversion of grey code.
Reversal conversion is executed for specified device by S· to BIN value and store it in specified device by D·.
Example:

GRY 1234

Y13			Y10	X7							X10
0	1	1	0	1	0	1	1	1	0	1	1

↓

BIN 1234

b15															b0
0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0

- Convert grey code into BIN data and send data.
- 32-bit grey code reversal conversion can be executed in maximum.
- As for values of S·, it is only valid in following range.
  16-bit computation: 0~32,767
  32-bit computation: 0~2,147,483,647

F188~F192 Peripheral communication instruction

Peripheral communication instruction

F188 CRC Cyclical Redundancy Checking

F 188		CRC	P	Cyclical Redundancy Checking	S D n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S													*	*		
D													*	*		
n					*	*							*	*		

Instruction format:

CRC	S	D	n

S·: Source device

D·: Device for storing CRC calculation value

n: Number of source device

It is the instruction for computing CRC code, in 8-bit and 16-bit conversion mode.

8-bit conversion mode when M8161 ON

16-bit conversion mode when M8161 OFF

Reference: the main CRC expressions:

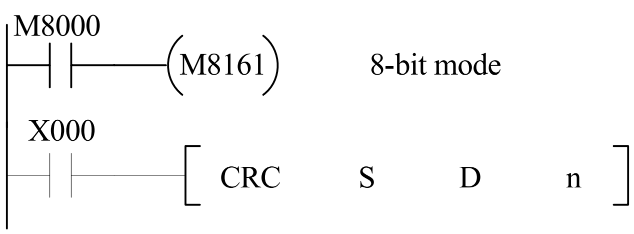
name	expression
CRC-12	$X^{12}+X^{11}+X^{3}+X^{2}+X+1$
CRC-16	$X^{16}+X^{15}+X^{2}+1$
CRC-32	$X^{32}+X^{26}+X^{3}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}\ X^{8}+X^{7}+X^{5}+X^{4}X^{2}+X+1$
CRC-CCITT	$X^{16}+X^{12}+X^{5}+1$

For example:

8-bit conversion mode[M8161=ON]

In 8-bit mode, only low byte of the element source would be operated

The result will be kept in the element of D and D+1, the low byte kept in D and the high kept in D+1



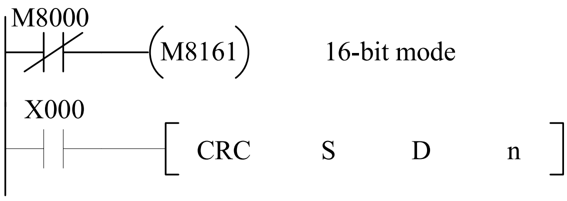
When S=D100, D=D0, n=6

			element	value
Address of source device	S	Low byte	D100 low byte	01H
	S+1	Low byte	D101 low byte	01H
	S+2	Low byte	D102 low byte	03H
	S+3	Low byte	D103 low byte	CDH
	S+4	Low byte	D104 low byte	6BH
	S+5	Low byte	D105 low byte	05H
	∫	∫	——	
	S+n-1	Low byte	——	
Address of preserve CRC value	D	Low byte	D0 low byte	42H
	D+1	Low byte	D1 low byte	82H

16-bit conversion mode [M8161=OFF]

In 16-bit mode, the low byte and the high of element source would be operated

The result will be kept in the element of D



When S=D100,D=D0,n=6

			element	value	
				8 bit	16 bit
Address of source device	S	Low byte	D100 low byte	01H	0101H
		High byte	D100 high byte	01H	
	S+1	Low byte	D101 low byte	03H	CD03H
		High byte	D101 high byte	CDH	
	S+2	Low byte	D102 low byte	6BH	056BH
		High byte	D102 high byte	05H	
	∫	∫	——		
	S+n/2-1	Low byte	——		
		High byte			
Address of preserve CRC value	D	Low byte	D0 low byte	42H	8242H
		High byte	D0 high byte	82H	

F190 DTLK Data Link

F 190		DTLK		Data Link	K

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
K					0, 1	0, 1										

Instruction format:

```
——[ DTLK     K ]
```

K,H:0,1

0: for built in RS485 port

1: for RS485 or RS232 expansion card

Operation:

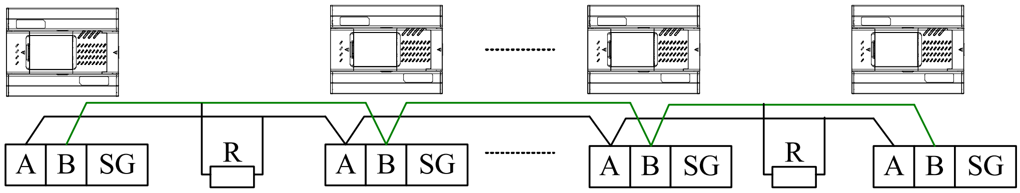
This instruction F190 DTLK used by PLC can setup a small network which enables PLC controlling other 15 PLC.

While two communication ports are ready for DTLK, only one firstly enabled is available. Communication frame and baud rate is set through D8120 or D8320, which is controlled by the different port.

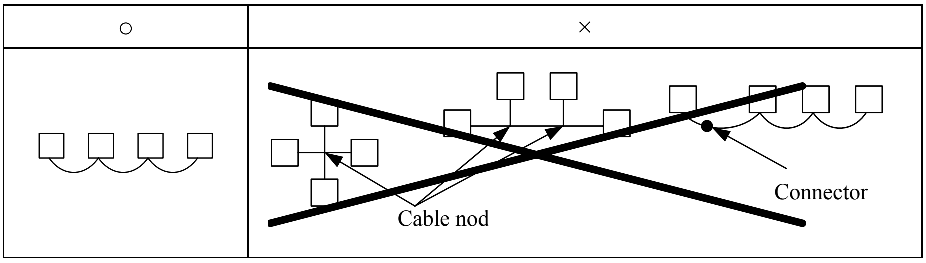
Both the port RS485/ RS232 expansion card (all type is available for expansion), RS485 port (only built-in port in H type) are available for Data Link. However, both of them can not be enabled simultaneously.

Item	Specification
Communication standard	EIA RS-485
Baud rate	9600bps~307200bps
Number of slaves	Max 15 slaves
Related devices	D0~D157, M2000~M3023
Data length for each slave	Max 64 bits+8 word
Communication cable	Insulated twisted cable, 2 lines type, Total length: 500m (76800bit/s), 1km(38400bit/s)

Wiring:



- Note 1: SHL terminal should be 3 class ground or the production will be interrupted to error operation because of noise.
- Note 2: Branch of communication cable should not exceed 3.
- Note 3: R represents terminal resistor (120Ω,1/4W).



Related devices:

(1) Special relays

Special relays	Feature	Function	Description	Respond from
M8400	Read-only	Master error	The relay will be on as master is error.	L
M8401	Read-only	Slave 1 error	The relay will be on as slave 1 is error.	M/L
M8402	Read-only	Slave 2 error	The relay will be on as slave 2 is error.	M/L
…	…	…	…	..
M8414	Read-only	Slave 14 error	The relay will be on as slave 14 is error.	M/L
M8415	Read-only	Slave 15 error	The relay will be on as slave 15 is error.	M/L
M8416	Read-only	state	The relay will be on as DTLK is enabled.	M/L
M8417	Read-only	Data Link mode	The relay will be on as expansion card is in Data Link.	M/L
M8418	Read-only	Data Link mode	The relay will be on as RS485 port is in Data Link.	M/L

(2) Data register

Special relays	Feature	Function	Description	Respond from
D8173	Read-only	Address number	Saving its own address number	M/L
D8174	Read-only	The number of slaves	Saving the number of slaves	M/L
D8175	Read-only	Refreshing range	Saving refreshing range (Data Link)	M/L
D8176	Write	Slave address setting	Setting its own address number	M/L
D8177	Write	Slavers number setting	Setting the number of slaves	M
D8178	Write	Data Link setting	Setting refreshing range (Data Link)	M
D8179	Read/ write	Retry times	Setting retry timess	M
D8180	Read/ write	Time-out setting	Setting communication time-out (Time-Out)	M
D8401	Read-only	Current communication scan time	Saving current communication scan time	M/L
D8402	Read-only	Max communication scan time	Saving Max communication scan time	M/L
D8403	Read-only	Error times for master	Error times for master	L
D8404	Read-only	Error times for slave 1	Error times for slave 1	M/L
D8405	Read-only	Error times for slave 2	Error times for slave 2	M/L
…	…	…	…	..
D8411	Read-only	Error times for slave 8	Error times for slave 8	M/L
…	…	…	…	..
D8417	Read-only	Error times for slave 14	Error times for slave 14	M/L
D8418	Read-only	Error times for slave 15	Error times for slave 15	M/L
D8419	Read-only	Error code for master	Error code for master	L
D8420	Read-only	Error code for slave 1	Error code for slave 1	M/L
D8421	Read-only	Error code for slave 2	Error code for slave 2	M/L
…	…	…	…	..
D8427	Read-only	Error code for slave 8	Error code for slave 8	M/L
…	…	…	…	..
D8433	Read-only	Error code for slave 14	Error code for slave 14	M/L
D8434	Read-only	Error code for slave 15	Error code for slave 15	M/L

Setting:

When the program is in operation, or TP03 is power ON, all the setting for Data Link will take effect.

(1) Setting the slaver address (D8176)

Set 0~15 to the special data register D8176, 0 is for master, and 1~15 is for slave.

(2) Setting the slavers number (D8177)

Set 1~15 to the special data register D8177(default: 7). It is unnecessary for slavers,The slavers number should be set according to different condition in order to raise the refreshing speed.

(3) Setting the refresh range (D8178)

Set 0~2 to special data register D8178 (default: 0). It is unnecessary for slaves.

D8178		0	1	2
Data Link mode		Mode 0	Mode 1	Mode 2
Refreshing range	Bit device (M)	0 point	32 point	64 point
	Word device (D)	4 point	4 point	8 point

The devices to be refreshed under different mode:

Address	Mode 0		Mode 1		Mode 2	
	(M)	(D)	(M)	(D)	(M)	(D)
No 0	—	D0~D3	M2000~M2031	D0~D3	M2000~M2063	D0~D7
No 1	—	D10~D13	M2064~M2095	D10~D13	M2064~M2127	D10~D17
No 2	—	D20~D23	M2128~M2159	D20~D23	M2128~M2191	D20~D27
No 3	—	D30~D33	M2192~M2223	D30~D33	M2192~M2255	D30~D37
No 4	—	D40~D43	M2256~M2287	D40~D43	M2256~M2319	D40~D47
No 5	—	D50~D53	M2320~M2351	D50~D53	M2320~M2383	D50~D57
No 6	—	D60~D63	M2384~M2415	D60~D63	M2384~M2447	D60~D67
No 7	—	D70~D73	M2448~M2479	D70~D73	M2448~M2511	D70~D77
No 8	—	D80~D83	M2512~M2543	D80~D83	M2512~M2575	D80~D87
No 9	—	D90~D93	M2576~M2607	D90~D93	M2576~M2639	D90~D97
No A	—	D100~D103	M2640~M2671	D100~D103	M2640~M2703	D100~D107
No B	—	D110~D113	M2704~M2735	D110~D113	M2704~M2767	D110~D117
No C	—	D120~D123	M2768~M2799	D120~D123	M2768~M2831	D120~D127
No D	—	D130~D133	M2832~M2863	D130~D133	M2832~M2895	D130~D137
No E	—	D140~D143	M2896~M2927	D140~D143	M2896~M2959	D140~D147
No F	—	D150~D153	M2960~M2991	D150~D153	M2960~M3023	D150~D157

(4) setting retry times (D8179)

Set 0~10 to special data register D8179 (default: 3). It is unnecessary for slaves. If the master retry communication with the slave for more than the set times, the slave will be in communication error.

(5) setting time out (D8180)

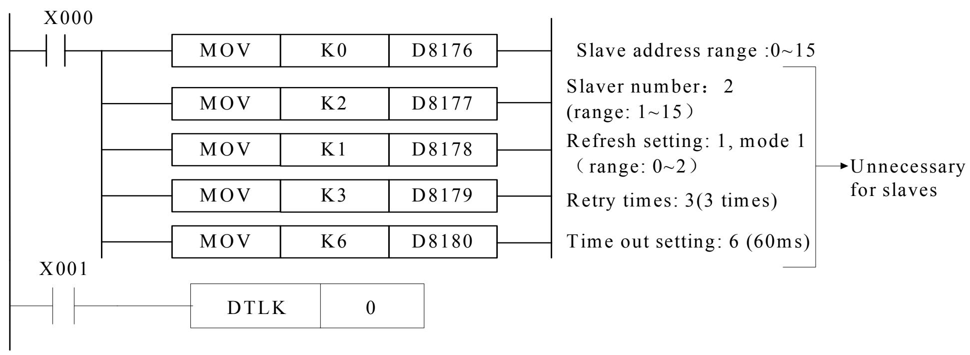
Set 5~255 to special data register D8180 (default: 5), the product of such value and 10 is the waiting time for communication time out (ms).

(6) Current communication scan time (D8401)

The product of such value and 10 is the current communication scan time (ms).

(7) Max communication scan time (D8402)

The example program for setting the said devices:



Error code:

When there is error, the special relays M8400~M8415 will indicates the error condition and the error code will be stored in special data registers (D8419~D8434).

Error code	Error	Error address	Check address	Description	Check point
01H	Communication time out error	L	M	There is no responding as the master sends the request to slave and time out.	Wiring, power supply and run/ stop state
02H	Communication number error	L	M	Address is not set according to the certain relations between master and slave	Wiring
03H	Communication counting error	L	M	The data in communication counter does not conform to according to the certain relations between master and slave	Wiring
04H	Communication frame error	L	M, L	Communication frame of slave is error	Wiring and DTLK setting
11H	Communication over time error	M	L	After the slave responses to master, the master does not send another request to slavers.	Wiring, power supply and run/ stop state
14H	Communication frame error	M	L	Communication frame of master is error	Wiring and DTLK setting
21H	Without slave	L	L *1	Address in the net is wrong	Address setting
22H	Address error	L	L *1	Slave address does not comply with the certain relations between master and slave	Wiring
23H	Communication counting error	L	L *1	The data in communication counter does not conform to according to the certain relations between master and slave	Wiring
31H	Receiving communication parameter error	L	L *2	Master send request before the slave accepts the set parameter.	Wiring, power supply and run/ stop state
32H	Other error	L	L *1	Communication instruction error	Net setting

M: master L: slave

1: another slave 2: Individual slave

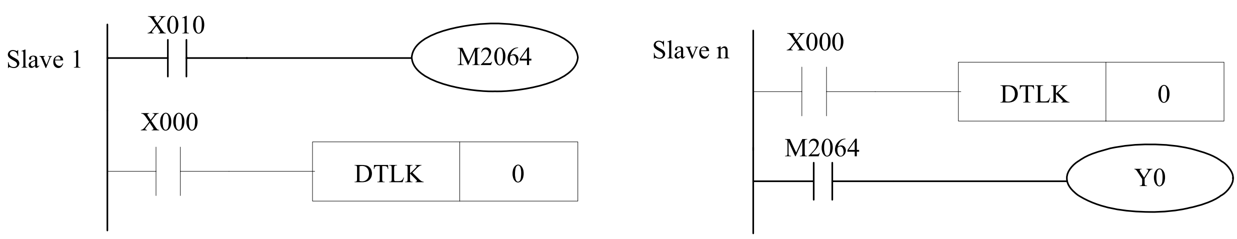
Communication Timing Sequence and the Time Required for Transmission

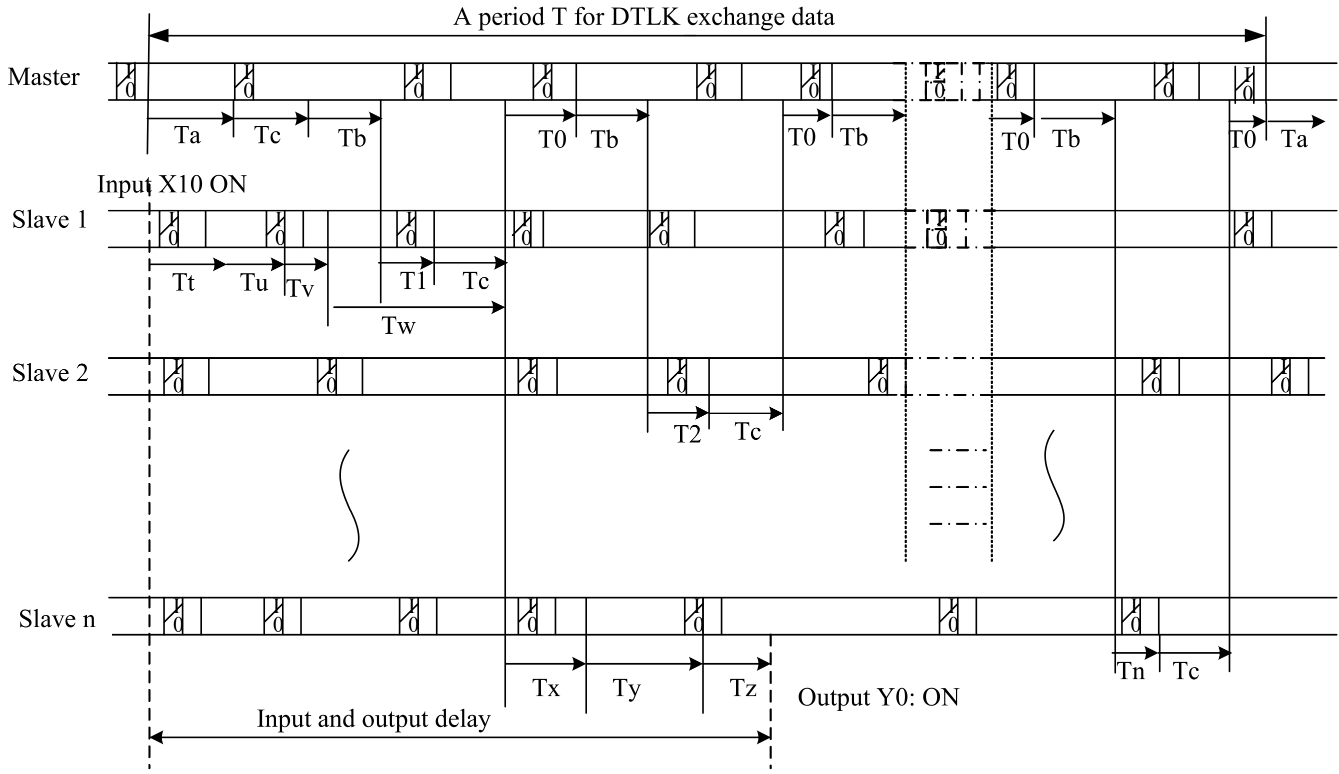
- The communication for master-station and slave-stations is not synchronous with the scanning cycle of master-station.
- The master station will perform the linked data exchange and update the communication flag at the scan cycle after the communication completed.

Communication timing sequence diagram and communication delay diagram.

In Data Link net, there will be delay for receiving data. Please refer to following figure for communication timing sequence:

For example: the M2064 for slave 1 is controlled by X010. The state of M2064 will be sent to other nod of the net as the instruction DTLK is enabled.





The time required to complete transmission

In data-link mode, the time T required for the master-station to complete communication with all slave-station can be devised as follows (not spend the SCAN TIME of master-station):

T=Ta + Tc + [Tb + Tn + Tc + T0]*n1*{+[Tb+Tn+D8180*10]*n2}*;

T_a : the transmission time for master sending instruction for net configuration to slave.

T_b : the transmission time for master sending instruction for data-exchange to slave.

T_c : the transmission time for the net exchanging data (differs from different DTLK mode).

T_0 : the time for master detecting communication states (0~1 SCAN TIME)

T_n: the time for slave detecting communication states (0~1 SCAN TIME)

(n1+n2): the number of DTLK slave set in master (D8177=1~15), n1: actual slave number, n2: the number of the slave which is not recognized by master (0~15).

D8180 is timeout value.

Delay time:

Tu: the time required for PLC to detect the input status (max. 1 SACN TIME)

Tv: the time between the PLC received input state and program started to be scanned.

Tw: the time for operation result send out (max net scan time T);

Tx: the time between the data received and data written to registers (max. 1 scan time);

Ty: the time between program operated to output (1scan time);

Tz: output port delay

The transmission time under different Baud rate:

Baud rate(bps)	Ta (ms)	Tb (ms)	Tc (ms)		
			DTLK mode 0	DTLK mode 1	DTLK mode 2
9600	21.8	12.6	31.0	40.1	67.6
19200	10.9	6.3	15.5	20.1	33.8
38400	5.5	3.2	7.8	10.0	16.9
57600	3.7	2.1	5.2	6.7	11.3
76800	2.8	1.6	3.9	5.0	8.5
128000	1.7	1.0	2.4	3.0	5.1
153600	1.4	0.8	2.0	2.5	4.3
307200	0.7	0.4	1.0	1.3	2.2

F191 RMIO Remote IO

F 191		RMIO		Remote IO	K

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
K					0, 1	0, 1										

Instruction format:

[RMIO K]

K,H:0,1

0: for built in RS485 port

1: for RS485 or RS232 expansion card

Operation:

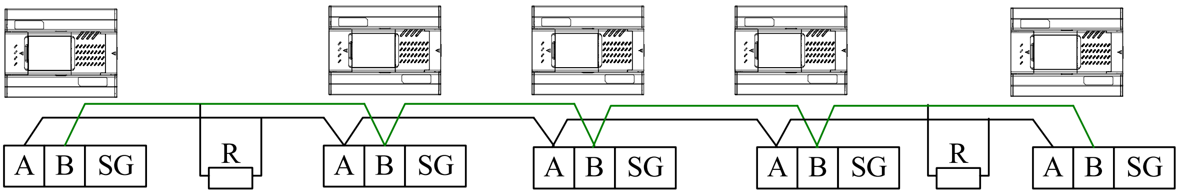
This instruction F191 RMIO used by PLC can setup a small network which enables PLC controlling other 4 PLCs.

While two communication ports are ready for RMIO, only the one firstly enabled is available.

Communication frame and baud rate is set through D8120 or D8320, which is controlled by the different port.

- Note 1: When a PLC is set as a slave in RMIO mode, it is used as a expansion I/O for master and only RMIO instruction is available for operation.
- Note 2: As long as PLC as a slave in RMIO mode, only stop the operation of program can switch the RMIO to other mode.

In Remote I/O mode, the master PLC can control other 4 PLCs.



Item	Description	
standard	EIA RS485	
Baud rate	9600bps~307200bps	
Number of slaves	Max 4 slave	
Related devices	Slave 1	Input: 36 points (M4200~M4235); Output: 24point (M4600~M4623)
	Slave 2	Input: 36 points (M4240~M4275); Output: 24point (M4624~M4647)
	Slave 3	Input: 36 points (M4280~M4315); Output: 24point (M4648~M4671)
	Slave 4	Input: 36 points (M4320~M4355); Output: 24point (M4672~M4695)
Cable	Insulated twisted cable, 2 lines type, Total length: 500m (76800bit/s), 1km(38400bit/s)	

Both the port RS485/ RS232 expansion card (all type is available for expansion), RS485 port (only built-in port in H type) are available for Data Link. However, both of them can not be enabled simultaneously.

Note: Only basic unit can be set as a slave in RMIO mode.

Related devices:

(1) Special relays

Special relays	Feature	Function	Description	Respond from
M8335	Read only	Communication state	ON as RMIO communication is enabled	M/L
M8336	Read only	Master error	ON as master error	L
M8337	Read only	Slave 1 error	On as slave 1 error	M/L
M8338	Read only	Slave 2 error	On as slave 2 error	M/L
M8339	Read only	Slave 3 error	On as slave 3 error	M/L
M8340	Read only	Slave 4 error	On as slave 4 error	M/L
M8341	Read only	RMIO mode	Expansion card is in RMIO mode	M/L
M8342	Read only	RMIO mode	RS485 port is in RMIO mode	M/L

(2) Data register D

Special relays	Feature	Function	Description	Respond from
D8373	Read only	Address number	Saving its own address number	M/L
D8374	Read only	The number of slaves	Saving the number of slaves	M/L
D8376	Write	Address number setting	Setting its own address number	M/L
D8377	Write	Setting the number of slaves	setting the number of slaves	M
D8379	Read/write	Retry times	Setting retry times	M
D8380	Read/write	Time-out setting	Setting communication time-out (Time-Out)	M/L
D8331	Read only	Current communication scan time	Saving current communication scan time	M
D8332	Read only	Max communication scan time	Saving Max communication scan time	M
D8333	Read only	Master error times	Master error times	L
D8334	Read only	Slave 1 error times	Slave 1 error times	M/L
D8335	Read only	Slave 2 error times	Slave 2 error times	M/L
D8336	Read only	Slave 3 error times	Slave 3 error times	M/L
D8337	Read only	Slave 4 error times	Slave 4 error times	M/L
D8338	Read only	Master error code	Master error code	L
D8339	Read only	Slave 1 error code	Slave 1 error code	M/L
D8340	Read only	Slave 2 error code	Slave 2 error code	M/L
D8341	Read only	Slave 3 error code	Slave 3 error code	M/L
D8342	Read only	Slave 4 error code	Slave 4 error code	M/L

Setting:

When the program is in operation, or PLC is power ON, all the setting for Remote I/O will take effect.

(1) Setting the slaver address (D8376)

Set 0~4 to the special data register D8376, 0 is for master, and 1~4 is for slave.

(2) Setting the slavers number (D8377)

Set 1~4 to the special data register D8377(default: 4). It is unnecessary for slavers

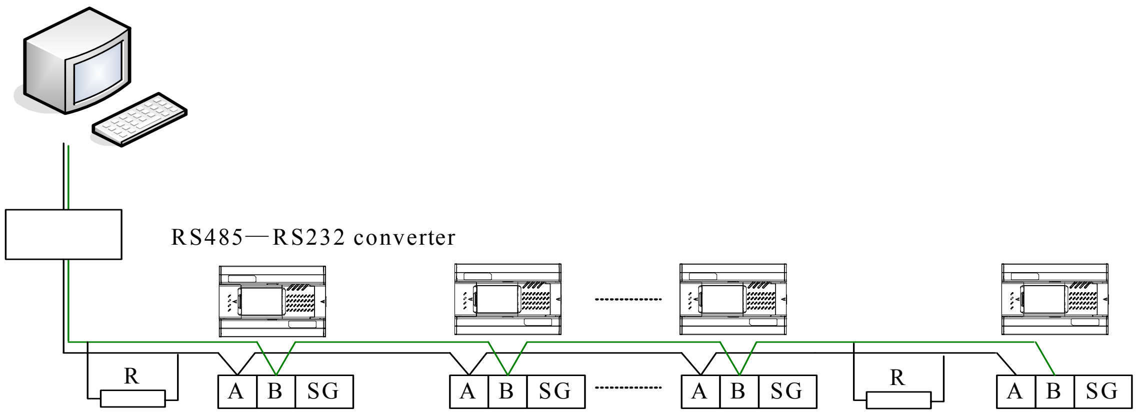
The slavers number should be set according to different condition in order to raise the refreshing speed.

The related devices for Remote I/O:

In Remote I/O mode, the related devices for master:

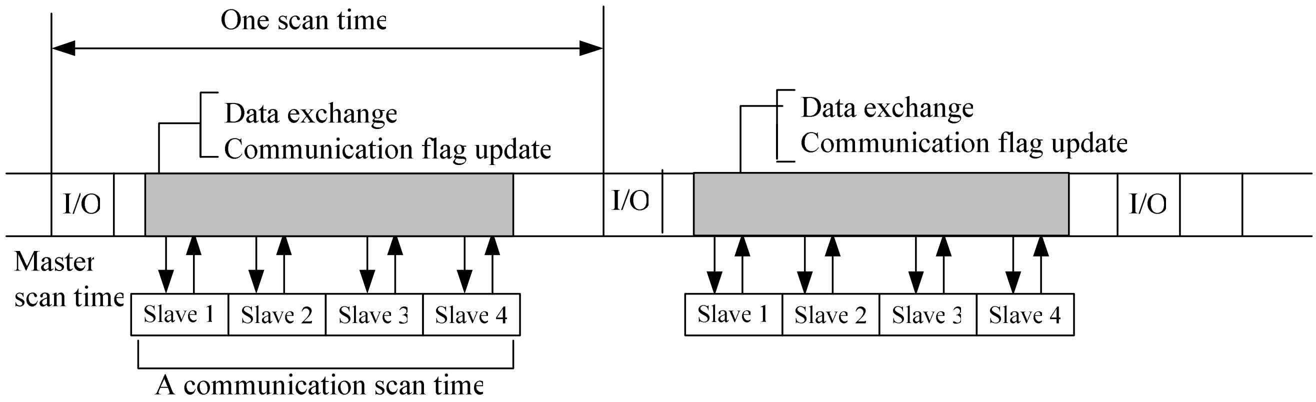
		Master	Slave 1	Slave 2	Slave 3	Slave 4
Slave 1	Input	M4200~M4235	X000~X043			
	Output	M4600~M4623	Y000~Y027			
Slave 2	Input	M4240~M4275		X000~X043		
	Output	M4624~M4647		Y000~Y027		
Slave 3	Input	M4280~M4315			X000~X043	
	Output	M4648~M4671			Y000~Y027	
Slave 4	Input	M4320~M4355				X000~X043
	Output	M4672~M4695				Y000~Y027

Wiring:



- Note 1: SHL terminal should be 3 class ground or the production will be interrupted to error operation because of noise.
- Note 2: Branch of communication cable should not exceed 3.
- Note 3: R represents terminal resistor (120Ω,1/4W).

Communication sequence and the time required for transmission



The Time Required for Transmission

The communication of master-station to slave-station, the data exchange of remote I/O and the update of communication flag are synchronous with the scan cycle of master station.

The process (1 communication period) will increase the SCAN TIME of master-station

When there is error in communication between master and slave, Remote I/O communication and PLC operation will stop and enter abnormal condition.

When an error occurs on the communication between the master station and slave-station, the remote I/O communication and PLC operation will be stopped and enter error mode. Besides, all communication flag of master-station and slave-station are set to OFF.

Possible cause of error is as follows:

① CRC error

② Slave in STOP mode or ERROR mode

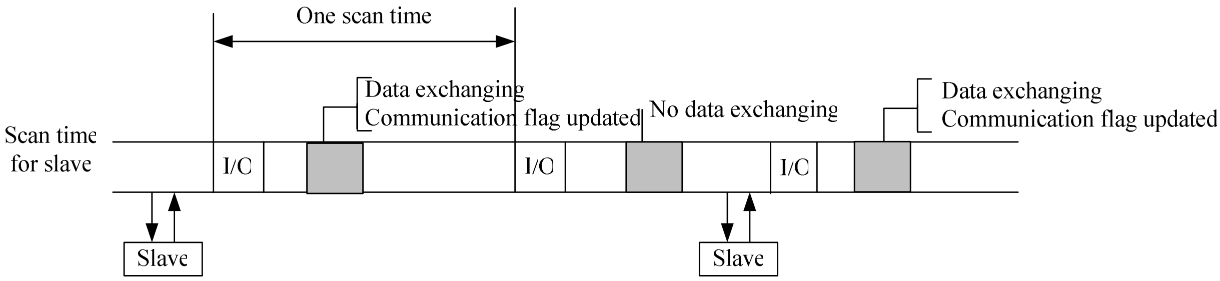
③ Slave not connected or connection wire broken

When the master-station is in STOP mode or ERROR mode, it will not communicate with any slave-station. The settings for communication format between master and slave are not same.

Communication sequence for slave

The communication of slave to master is asynchronous with the scan time of slave.

After communication between master and slave is finished, the Remote I/O data and communication flag will be refreshed, which will last about 0.2ms.



The time required for transmission

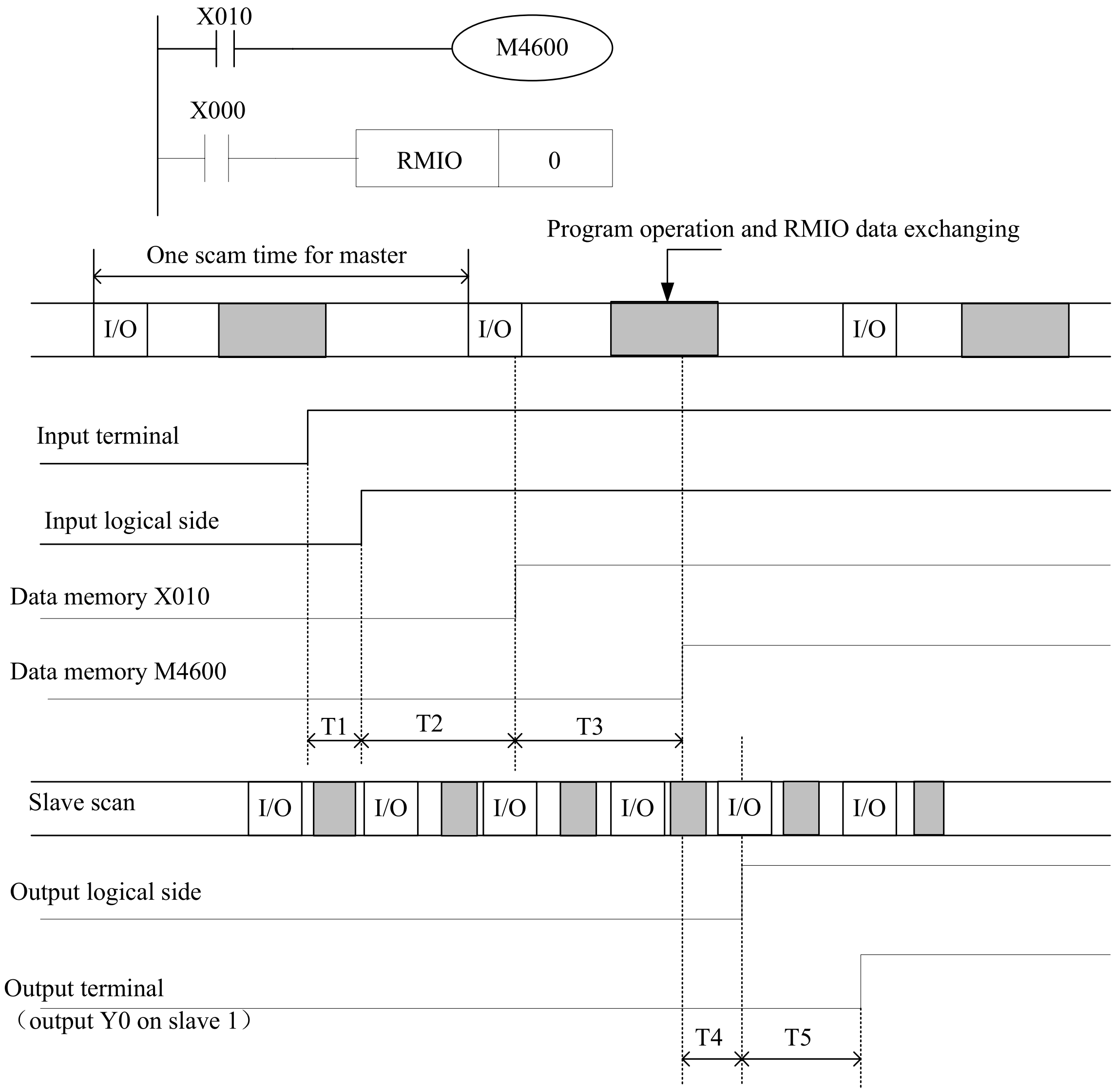
In remote I/O mode, the time T (the communication period, this period will be included in the master station SCAN TIME) required for master-station to complete the communication with all slave-stations is as follows :

Baud Rate (bps)	Communication time for each slave, Tn (ms)	Time out, t (ms)	Communication time for master, T(ms)	Normal communication time for master and 4 slaves (ms)
9600	42	D8380*10	Tn*n1＋t*n2 (n1: normal slave number; n2: slave number for time out)	168
19200	21			84
38400	11			44
57600	7			28
76800	6			24
128000	4			16
153600	3			12
307200	2			8

If there is communication error in slave, the communication time will be increased repeatedly (Tn will be added to the time for each error)

Delay time:

When the remote I/O is receiving data, there will be some delay as in the following figure.



T1: delay for input (response time for OFF to ON)

T2: time for master writing data to coil register (max 1 scan time)

T3: program operation and output time

T4: time between the slave received data to output terminal

T5: delay for output (response time for ON to OFF)

Error code:

When there is error, the special relays M8400~M8415 will indicates the error condition and the error code will be stored in special data registers (D8419~D8434).

Error code	Error	Error address	Check address	Description	Check point
01H	Communication time out error	L	M	There is no responding as the master sends the request to slave and time out.	Wiring, power supply and run/ stop state
02H	Communication number error	L	M	Address is not set according to the certain relations between master and slave	Wiring
03H	Communication counting error	L	M	The data in communication counter does not conform to according to the certain relations between master and slave	Wiring
04H	Communication frame error	L	M, L	Communication frame of slave is error	Wiring and RMIO setting
11H	Communication over time error	M	L	After the slave responses to master, the master does not send another request to slavers.	Wiring, power supply and run/ stop state
14H	Communication frame error	M	L	Communication frame of master is error	Wiring and RMIO setting
21H	Without slave	L	L *1	Address in the net is wrong	Address setting
22H	Address error	L	L *1	Slave address does not comply with the certain relations between master and slave	Wiring
23H	Communication counting error	L	L *1	The data in communication counter does not conform to according to the certain relations between master and slave	Wiring
24H	Communication frame error	L	L *1	Communication frame of slave is error	Wiring and RMIO setting

M: master L: slave

*1: another slave

F192 TEXT OP07/08 TEXT

F 192		TEXT	P	OP07/08 TEXT	D	S	n

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D					*	*							*	*		
S													*	*		
n					1, 2	1, 2										

Instruction format:

```
——[ TEXTP    D    S    n ]
```

D: number of character for display

S: number of data register for display

n: display line of OP07/08

Operation:

This instruction should be used with OP07/08. After F192 is enabled, the value 13 will be written to data register D8284, after OP07/08 saving the ‘13’ in D8284, the value 13 also will be written to D8285 by OP07/08 itself.

As F192 is enabled, the certain text file will be saved to D8280 and D8281 (D8280 is for the file to be displayed in the first line of OP07/08, D8281 is for the second one) and the value to be displayed will be saved toD8295 and D8296.

The value in D8295 will be displayed in the ‘#’position of the first line, while the value in D8296 of the second line.

When there is ‘?’ on LCD, you can input data, the input data for first line will be saved in the D register (Number =value in D8295 + 1). As for the second line, the input data in position ‘?’ will be saved in D register (Number =value in D8296 + 1).

‘#’ and ‘?’ can be placed anywhere in the text file. However, only the former 5 ones can be set as inputs or outputs.

Example:

LCD position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Text file 1:

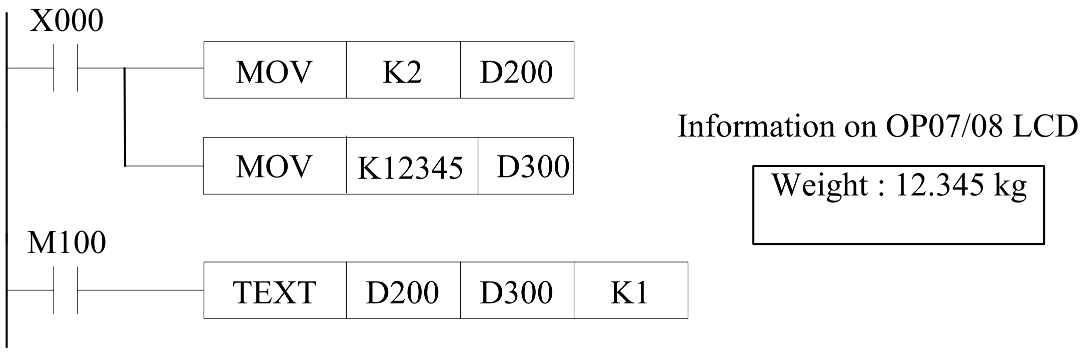
D register	2000		2001		2002		2003		2004		2005		2006		2007		2008		2009	
Content	L	e	n	g	t	h		:		#	#	.	#	#	#		c	m		

Text file 2:

D register	2010		2011		2012		2013		2014		2015		2016		2017		2018		2019	
Content	W	e	i	g	t	h		:		#	#	.	#	#	#		k	g		

Text file 3:

D register	2020		2021		2022		2023		2024		2025		2026		2027		2028		2029	
Content	U	n	i	t		p	r	i	c	e		:		$	?	?	?	?	?	

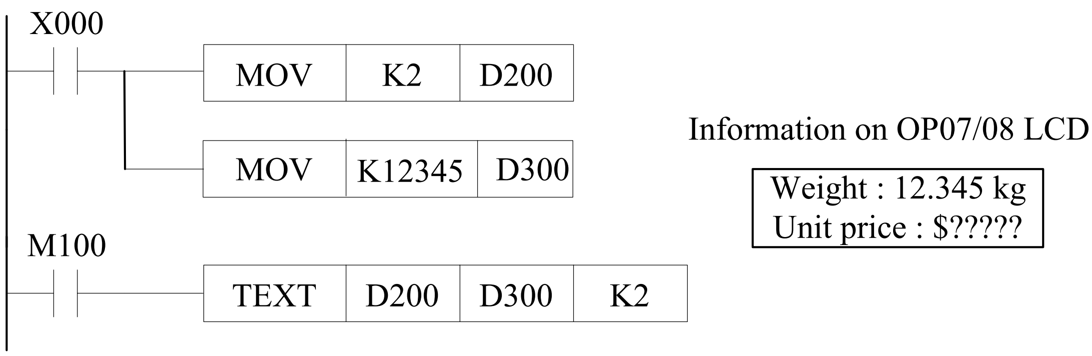


Description:

1, X000 ON, 2 will be moved to D200 while 12345 will be moved toD300;

2, when M100 is ON, TEXT instruction is enabled. D8284 defaults 13, D200 will be written to D8280; D300 to D8285.Then OP07/08 will enter F192 mode.

3, F192 will operate for the first time. As D8280= D200= 2, OP07/08 will display the file 2 on the first line of LCD. Because there is a ‘#’ in the file 2, 12345 in D300 will be displayed in the place of ‘#’.



Description:

1, X000 ON, 2 will be moved to D200 while 12345 will be moved to D300;

2, When M100 is ON, TEXT instruction is enabled. D8284 defaults 13, D200 will be written to D8280; sum of data in D200 and 1 will be written to D8281, D300 to D8285, D8286.Then OP07/08 will enter F192 mode.

3, F192 will operate for the first time. As D8280= D200= 2, D8281=3, OP07/08 will display the file 2 on the first line of LCD and file 3 on the second line. Moreover, 12345 in D300 will be displayed in the place of ‘#’ and the input data by the keys will be stored in D301.

F193 DTLK2 Data Link 2

F 193		DTLK2		Data Link 2	S1 • S2 • K

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 •													*	*		
S2 •					*	*							*	*		
K					0,1	0,1										

Instruction format:

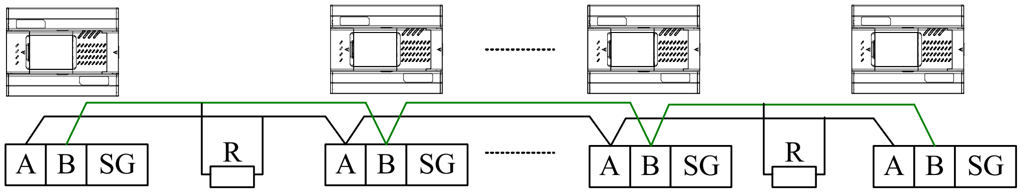
[DTLK2 S1 • S2 • K]

S1: start address of data source (D0~D7999)
S2: Length of data source (1~40)
K,H:0,1
0: for built in RS485 port
1: for RS485 or RS232 expansion card

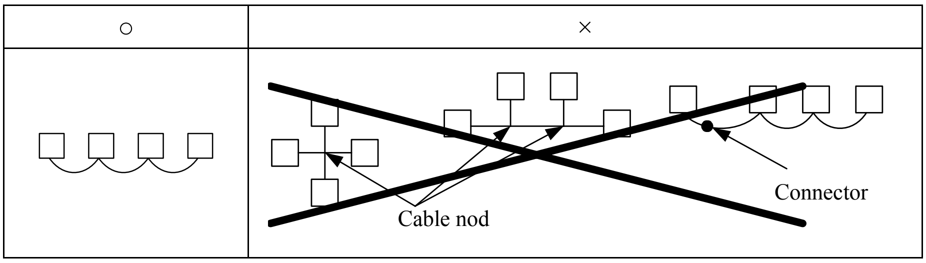
Operation:
This instruction F193 DTLK2 used by PLC can setup a small network which enables PLC controlling other 15 PLC.
While two communication ports are ready for DTLK2, only one firstly enabled is available. Communication frame and baud rate is set through D8120 or D8320, which is controlled by the different port.
Both the port RS485/ RS232 expansion card (all type is available for expansion), RS485 port (only built-in port in H type) are available for Data Link. However, both of them can not be enabled simultaneously.

Item	Specification
Communication standard	EIA RS-485
Baud rate	9600bps~307200bps
Number of slaves	Max 15 slaves
Related devices	D0~D7999,first address is decided by DTLK2
Data length for each slave	Max 40word, length is decided by DTLK2
Communication cable	Insulated twisted cable, 2 lines type, Total length: 500m (76800bit/s), 1km(38400bit/s)

Wiring:



- Note 1: SHL terminal should be 3 class ground or the production will be interrupted to error operation because of noise.
- Note 2: Branch of communication cable should not exceed 3.
- Note 3: R represents terminal resistor (120Ω,1/4W).



Related devices:

(1) Special relays

Special relays	Feature	Function	Description	Respond from
M8400	Read-only	Master error	The relay will be on as master is error.	L
M8401	Read-only	Slave 1 error	The relay will be on as slave 1 is error.	M/L
M8402	Read-only	Slave 2 error	The relay will be on as slave 2 is error.	M/L
…	…	…	…	..
M8414	Read-only	Slave 14 error	The relay will be on as slave 14 is error.	M/L
M8415	Read-only	Slave 15 error	The relay will be on as slave 15 is error.	M/L
M8416	Read-only	state	The relay will be on as DTLK2 is enabled.	M/L
M8417	Read-only	Data Link2 mode	The relay will be on as expansion card is in Data Link2.	M/L
M8418	Read-only	Data Link2 mode	The relay will be on as RS485 port is in Data Link2.	M/L

(2) Data register

Special relays	Feature	Function	Description	Respond from
D8173	Read-only	Address number	Saving its own address number	M/L
D8174	Read-only	The number of slaves	Saving the number of slaves	M/L
D8175			Preserve	
D8176	Write	Slave address setting	Setting its own address number	M/L
D8177	Write	Slavers number setting	Setting the number of slaves	M
D8178			Preserve	
D8179	Read/ write	Retry times	Setting retry timess	M
D8180	Read/ write	Time-out setting	Setting communication time-out (Time-Out)	M
D8401	Read-only	Current communication scan time	Saving current communication scan time	M/L
D8402	Read-only	Max communication scan time	Saving Max communication scan time	M/L
D8403	Read-only	Error times for master	Error times for master	L
D8404	Read-only	Error times for slave 1	Error times for slave 1	M/L
D8405	Read-only	Error times for slave 2	Error times for slave 2	M/L
…	…	…	…	..
D8411	Read-only	Error times for slave 8	Error times for slave 8	M/L
…	…	…	…	..
D8417	Read-only	Error times for slave 14	Error times for slave 14	M/L
D8418	Read-only	Error times for slave 15	Error times for slave 15	M/L
D8419	Read-only	Error code for master	Error code for master	L
D8420	Read-only	Error code for slave 1	Error code for slave 1	M/L
D8421	Read-only	Error code for slave 2	Error code for slave 2	M/L
…	…	…	…	..
D8427	Read-only	Error code for slave 8	Error code for slave 8	M/L
…	…	…	…	..
D8433	Read-only	Error code for slave 14	Error code for slave 14	M/L
D8434	Read-only	Error code for slave 15	Error code for slave 15	M/L

Setting:

When the program is in operation, or TP03 is power ON, all the setting for Data Link2 will take effect.

(1) Setting the slaver address (D8176)

Set 0~15 to the special data register D8176, 0 is for master, and 1~15 is for slave.

(2) Setting the slavers number (D8177)

Set 1~15 to the special data register D8177(default: 7). It is unnecessary for slavers,The slavers number should be set according to different condition in order to raise the refreshing speed.

(3) setting retry times (D8179)

Set 0~10 to special data register D8179 (default: 3). It is unnecessary for slaves. If the master retry communication with the slave for more than the set times, the slave will be in communication error.

(4) setting time out (D8180)

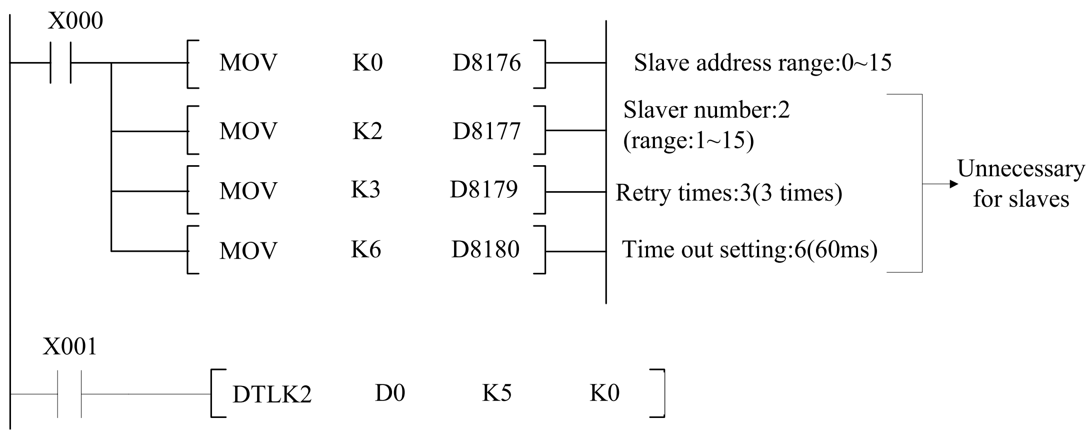
Set 5~255 to special data register D8180 (default: 5), the product of such value and 10 is the waiting time for communication time out (ms).

(5) Current communication scan time (D8401)

The product of such value and 10 is the current communication scan time (ms).

(6) Max communication scan time (D8402)

The example program for setting the said devices:



Error code:

When there is error, the special relays M8400~M8415 will indicates the error condition and the error code will be stored in special data registers (D8419~D8434).

Error code	Error	Error address	Check address	Description	Check point
01H	Communication time out error	L	M	There is no responding as the master sends the request to slave and time out.	Wiring, power supply and run/ stop state
02H	Communication number error	L	M	Address is not set according to the certain relations between master and slave	Wiring
03H	Communication counting error	L	M	The data in communication counter does not conform to according to the certain relations between master and slave	Wiring
04H	Communication frame error	L	M, L	Communication frame of slave is error	Wiring and DTLK2 setting
11H	Communication over time error	M	L	After the slave responses to master, the master does not send another request to slavers.	Wiring, power supply and run/ stop state
14H	Communication frame error	M	L	Communication frame of master is error	Wiring and DTLK2 setting
21H	Without slave	L	L *1	Address in the net is wrong	Address setting
22H	Address error	L	L *1	Slave address does not comply with the certain relations between master and slave	Wiring
23H	Communication counting error	L	L *1	The data in communication counter does not conform to according to the certain relations between master and slave	Wiring
31H	Receiving communication parameter error	L	L *2	Master send request before the slave accepts the set parameter.	Wiring, power supply and run/ stop state
32H	Other error	L	L *1	Communication instruction error	Net setting

M: master L: slave

1: another slave 2: Individual slave

F224~246 Contact comparison instruction

F224~F246 Contact comparison instructions

F224~230 Contact-joint state comparison LD

F 224~230	D	LD		Contact-joint state comparison LD	S1 S2

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1					*	*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*	*

S1: Comparison value 1

S2: Comparison value 2

The instruction for comparing contents of S1 and S2, when the comparison results are equal, the instruction is active; when they are not equal, the instruction is not active.

The instruction LD can be used with bus wire.

F No	16-bit element instruction	32-bit element instruction	Active conditions	Inactive conditions
224	LD=	D LD=	S1=S2	S1≠S2
225	LD>	D LD>	S1>S2	S1≤S2
226	LD<	D LD<	S1<S2	S1≥S2
228	LD<>	D LD<>	S1≠S2	S1=S2
229	LD≤	D LD≤	S1≤S2	S1>S2
230	LD≥	D LD≥	S1≥S2	S1<S2

When leftmost of S1 and S2 (16-bit instruction: b15, 32-bit instruction: b31) is 1, the comparison value is regarded as negative value.

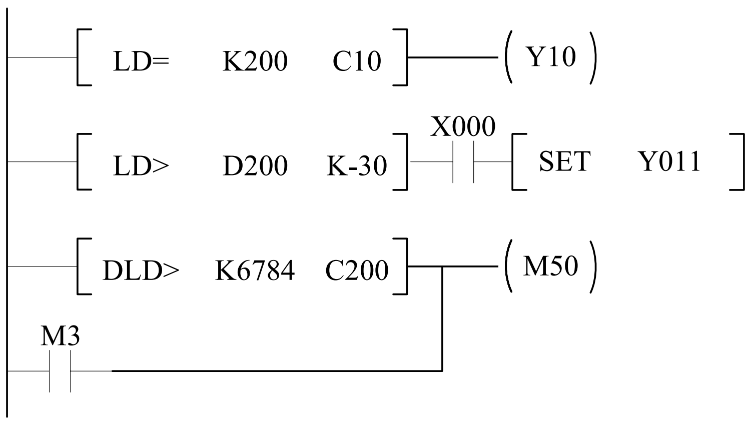
When 32-bit length counter (C200~) is used in the instruction for comparison, 32-bit instruction must be used (DLD※); if 16-bit instruction (LD※, CPU judges “program error”, red indicator light on the host panel is twinkling and CPU can not run.

Program case

When C10 is equal to K200, Y10= On

When D200 is more than -30 and X0= On, Y11= On and hold.

When C200 is less than 6,784 or M3= On, M50= On



F232~238 Contact-joint state comparison AND

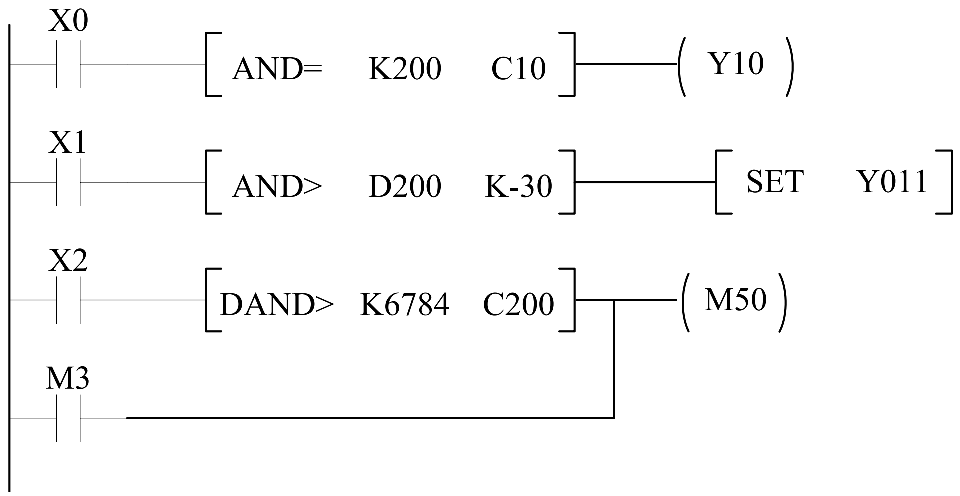
F 232~238	D	AND		Contact-joint state comparison AND	S1 S2

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1					*	*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*	*

S1: Comparison value 1
S2: Comparison value 2
The instruction for comparing contents of S1 and S2, when the comparison results are equal, the instruction is active; when they are not equal, the instruction is not active. The instruction LD is the comparison instruction for series connection to the contact-joint.

F No	16-bit element instruction	32-bit element instruction	Active conditions	Inactive conditions
224	AND=	D AND=	S1=S2	S1≠S2
225	AND>	D AND>	S1>S2	S1≤S2
226	AND<	D AND<	S1<S2	S1≥S2
228	AND<>	D AND<>	S1≠S2	S1=S2
229	AND≤	D AND≤	S1≤S2	S1>S2
230	AND≥	D AND≥	S1≥S2	S1<S2

When leftmost of S1 and S2 (16-bit instruction: b15, 32-bit instruction: b31) is 1, the comparison value is regarded as negative value.
When 32-bit length counter (C200~) is used in the instruction for comparison, 32-bit instruction must be used (DLD※); if 16-bit instruction (LD※), CPU judges “program error”, red indicator light on the host panel is twinkling and CPU can not run.
Program case
When X0= on, current value of C10 is K200, Y10= on.
When X1=on and buffer D200 is more then -30, Y11= On and hold.
When X2= on and current value of C200 is less than 6,784,or M3=on, M50=on.



F240~246 Contact-point state comparison OR

F 240~246	D	OR		Contact-point state comparison OR	S1 S2

	Bit element				Word element											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1					*	*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*	*

S1: Comparison value 1

S2: Comparison value 2

The instruction for comparing contents of S1 and S2, when the comparison results are equal, the instruction is active; when they are not equal, the instruction is not active. The instruction is the comparison instruction for parallel connection to the contact-joint.

F No	16 –bit element instruction	32-bit element instruction	Active conditions	Inactive conditions
224	OR=	D OR=	S1=S2	S1≠S2
225	OR>	D OR>	S1>S2	S1≤S2
226	OR<	D OR<	S1<S2	S1≥S2
228	OR<>	D OR<>	S1≠S2	S1=S2
229	OR≤	D OR≤	S1≤S2	S1>S2
230	OR≥	D OR≥	S1≥S2	S1<S2

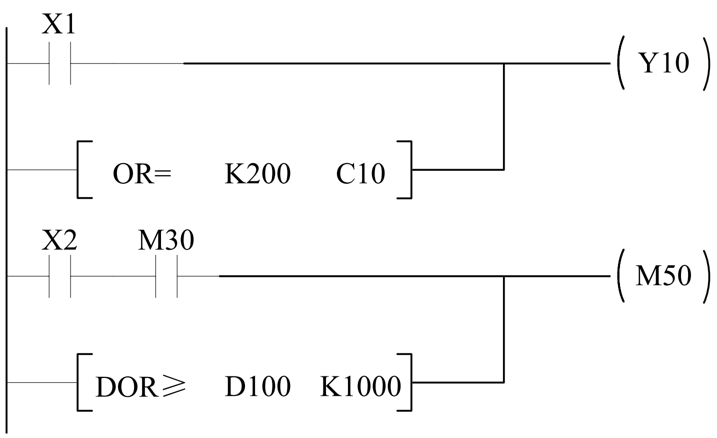
When leftmost of S1 and S2 (16-bit instruction: b15, 32-bit instruction: b31) is 1, the comparison value is regarded as negative value.

When 32-bit length counter (C200~) is used in the instruction for comparison, 32-bit instruction must be used (DLD※); if 16-bit instruction (LD※), CPU judges “program error”, red indicator light on the host panel is twinkling and CPU can not run..

Program case

When X1=on, or current value of C10 is equal to K200, Y10= On.

When X2 and M30 are equal to On or 32-bit buffer D101 and D100 is larger than or equal to K1,000, M50=On.

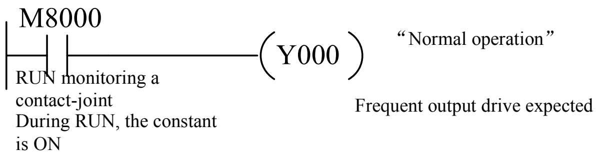


Chapter VII Additional Interpretation of Basic Functions

1 Additional interpretation for special devices

Operation symbol of PLC

RUN of operation state of PLC is used to M8000 and M8001, which are taken as drive conditions for the instruction and display in normal operation.



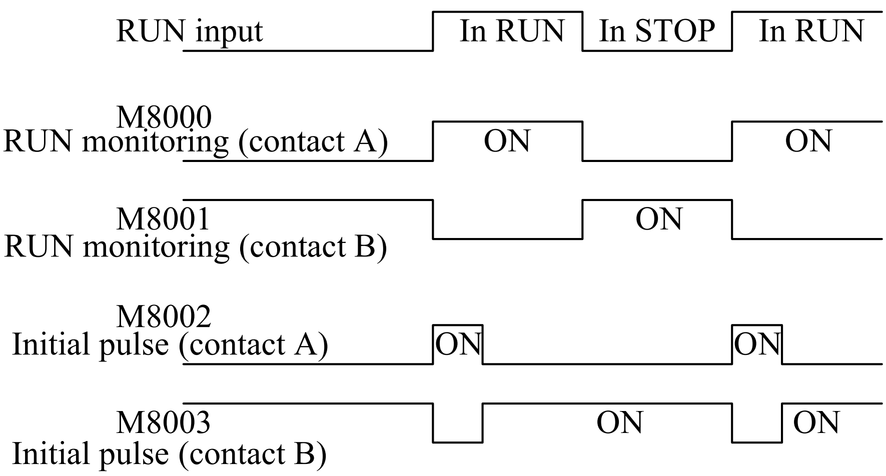
When M8001 is RUN, the constant is OFF.

After the initial pulse M8002 starts operation in PLC, only one computation is ON and it keeps OFF in the left time.

The pulse is taken as initial setting signal for program initialization, writing specified values and etc.

M8003 is OFF for one computation cycle after RUN, and it keeps ON in other left time.

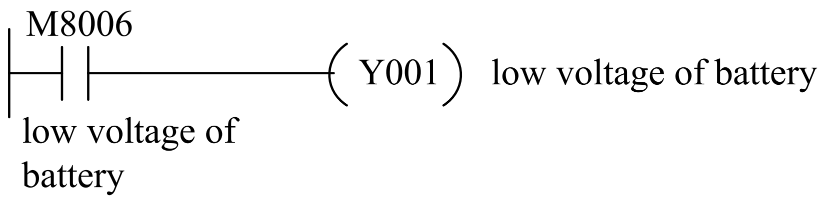
Time sequence of mark action



Testing of super-low voltage of battery

Super-low battery voltage testing and external output

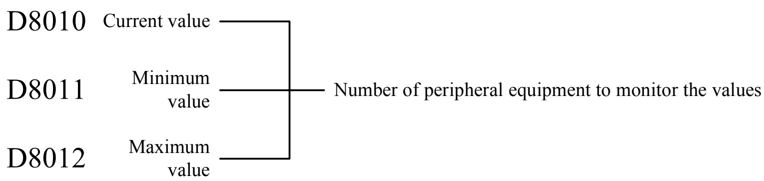
The device is used to test super-low battery voltage of backup lithium battery of the storage device. When the PLC tests the battery voltage is super low, BATT error indication light is ON. The sequential control program is used to report to the outside.



M8007 used for state locking for low voltage of battery

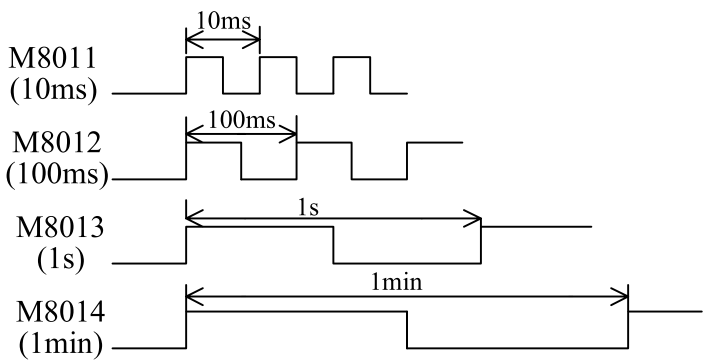
Computation time (monitoring of scan time)

Computation time of PLC is stored in D8010~D8012.



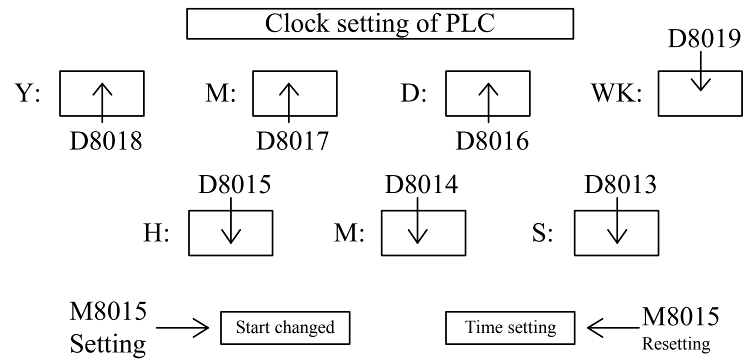
The numerical values include the waiting time of constant scan time stated in the following.

The machine offers the following four internal clocks. When the PLC is powered on, the four types of time will keep oscillating.



Note: even the PLC is STOP, the clock still keeps operating. Therefore, falling edge and starting time of clock monitored by RUN (M8000) are not synchronous.

Real-time clock

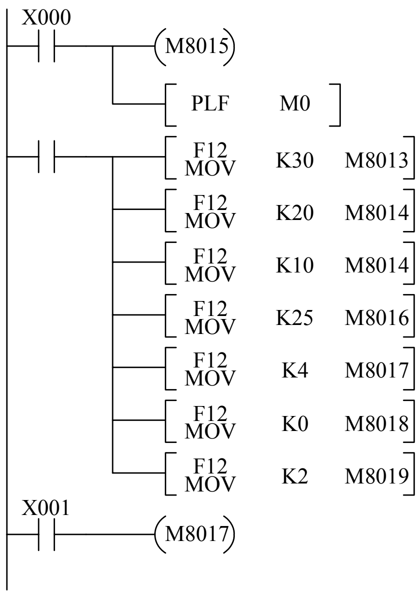


Details of using programming equipment

Forced setting

Current value of the data register is used to change the function. The clock data expected to be calibrated is input to the data register. When reaching the setting time, it sets forcibly.

Common program executing



Attentions for clock calibration

When M8015 is OFF, no change can be done to the time register.

When M8015 is ON, a new time is input.

When the setting time is input, the time several minutes earlier than current time shall be set.

When it reaches the setting time, M8015 is changed from ON——OFF, the state changes and the new time takes effect.

When the input time does not exist, the time can not be changed and the correct time data shall be input a second time.

Values 2000~2099 of D8018 represent the years 2000~2099.

Holding stop of the storage device

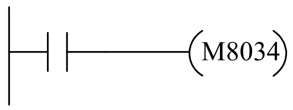
Output holding in STOP

The special auxiliary relay M8033 is driven in advance, even after the PLC from RUN——STOP, it still keeps the output state in operation.

For instance, if drive heating of the PLC is required, stop the PLC for the drive heater and other equipment. After the executing program is changed, it can be performed a second time.

Instruction for all outputs prohibited

The output lock storage device is cleared by driving M8034, and all the output relays become OFF and the PLC still operates on the image storage device.



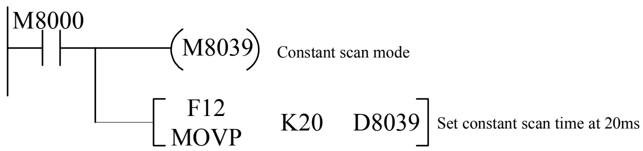
All outputs prohibited
（Image RUN）

Constant scan mode

Fixing of computation processing time

The auxiliary relay M8039 is driven, and the destination scan time is written into the data register M8039 with 1ms as unit in advance, and computation cycle of the PLC will not be less than the value.

Even the computation ends earlier, it will wait in the left time, and return to the step 0.



When the scan synchronous instructions like FNC67 (RAMP), FNC71 (HKY), FNC74 (SEGL), FNC75 (ARWS), FNC77 (PR) and etc. are executed, constant scan mode or timing interrupted by the timer are used to drive.

Especially the instruction FNC71 (HKY) is used, filtration wave for button input will lead to delay response, and the scan time must be set above 20ms.

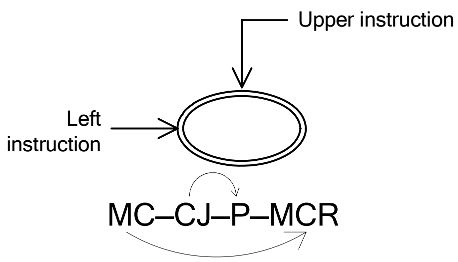
Note: in the scan time of D8010~8012, it includes the specified time of constant scan mode.

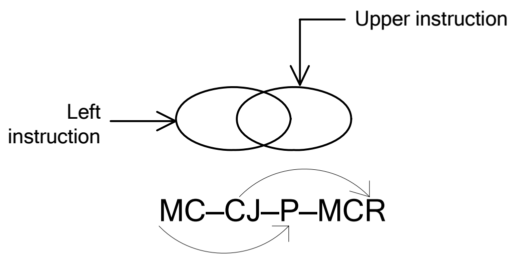
Transfer prohibited during states

After driving M8040, even all the transfer conditions are provided, state transfer can not be performed and output in the stop state will continue acting. Please refer to the step instruction interpretation on output reset.

2 Relation of control instructions of program flow

Relation of the instruction MC-MCR and the instruction CJ has been described in the instruction F00CJ. See the following on relations of other instructions.
In the following figure, represents the relation of inclusion and represents repeating of the front and back inter-zones.





	MC-MCR	CJ-P	EI-DI	FOR-NEXT	STL-RET
MC-MCR	○	○ example 1	○	○	○
	△	△ example 2	△	×(6607)	×(6605)
CJ-P	○	○	○	○	○
	△	△	△	△	△
EI-DI	○	○	○	○	○
	△	△	△	△	△
FOR-NEXT	×(6607)	○	○	○ 16 layers	×(6607)
	×(6607)	△	△	△ *2	×(6607)
STL-RET	×(6605)	○	○	○ In 1 STL	○
	×(6605)	△	△	×(6607)	△
P-SRET	×(6606)	○	○	○	×(6606)
	×(6608)	△	△	×(6607)	×(6605)
I-IRET	×(6606)	○	○	○	×(6606)
	×(6606)	△	△	×(6607)	×(6606)
FEND-END	○	○	○	○	○
	×(6608)	△ ×(6701)	△	×(6607)	×(6605)
O-FEND	○	○	○	○	○
	△ ×(6608)	△	△	×(6607)	×(6605)
O-END	○	○	○	○	○
(No FEND)	×(6608)	×(6701)	△	×(6607)	×(6605)

○: It can be used without problem.
×: Combined use prohibited, number ()is error code.
△ : Although it is not strictly prohibited, it may lead to complication of actions, which shall be avoided as much as possible.

P-SRET	I-IRET	FEND-END	Remark
×(6608)	×(6608)	×(6608)	1 No abnormality display, and DI state forgotten. 2 R FOR NEXT NEXT, such as real line action. 3 Only valid for FEND and END, not all the programs to be written or executed. No abnormality display. Besides some instructions, the instructions of inclusion relation can be used in combined form and the following exceptions must be paid attention to. 1.MC-MCR can not be used in FOR~NEXT, STL-RET, P-SRET, I-IRET and other instructions. 2.STL-RETcan not be used in FOR~NEXT, P-SRET, I-IRET and other instructions. 3.MC-MCR, FOR-NEXT, P-SRET, I-IRET can not use I, IRET, SRET, FEND, END and other instructions.
×(6606)	×(6606)	×(6608)	
△	△	△	
△	△	△	
○	○	○	
○	○	○	
×(6607)	×(6607)	×(6607)	
×(6701)	×(6607)	×(6607)	
×(6605)	×(6605)	×(6605)	
×(6606)	×(6606)	×(6605)	
×(6606)	×(6606)	×(6709)	
×(6606)	×(6606)	×(6709)	
×(6606)	×(6606)	×(6606)	
×(6606)	×(6606)	×(6606)	
○	○		
×(6709)	×(6709)		
×(6606)	×(6606)		
×(6709)	×(6606)		
×(6606)	×(6606)		
×(6709)	×(6706)		

3 ASC II Character arrangement

<ASC II code list (Representation of 7-bit code and hexadecimal system)>

Hexadecimal system	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		DLE	SP	0	@	P	、	p.								
1	SOH	DC1	!	1	A	Q	a.	q.								
2	STX	DC2	”	2	B	R	b.	r.								
3	ETX	DC3	#	3	C	S	c.	s.								
4	EOT	DC4	$	4	D	T	d.	t.								
5	ENO	NAK	%	5	E	U	e.	u.								
6	ACK	SYN	&	6	F	V	f.	v.								
7	BEL	ETB	’	7	G	W	g.	w.								
8	BS	CAN	(	8	H	X	h.	x.								
9	HT	EM	)	9	I	Y	i.	y.								
A	LF	SUB	*	:	J	Z	j.	z.								
B	VT	ESC	+	;	K	[	k.	{								
C	FF	FS	,	<	L	\	l.									
D	CR	GS	—	=	M	]	m.	}								
E	SO	RS	□	>	N	^	n.	~								
F	SI	US	/	?	O	_	o.	DEL								

<Case of ASC II codes>

Decimal system number	ASC II (Hexadecimal system number)
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39

English letter	ASC II (Hexadecimal system number)	English letter	ASC II (Hexadecimal system number)
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	K	58
L	4C	Y	59
M	4D	Z	5A

Code	ASC II (Hexadecimal system number)
STX	02
ETX	03

4 List of error codes

M register		D register		Continue to operate
No.	Function	No.	Error code description	
8060	Expansion card error	8060	Error code :200x: Expansion card not adapt to x: card install in fact 1:TP03-6AV 2:TP03-485RS 3:TP03-232RS 4:TP03-2AI 5:TP03-2TI 6:TP03-10P 6006：no communication card	Y
8061	PC hardware check	8061	Error code 0000: No error 6101: RAM error 6102: ROM error 6103: Basic unit I/O bus error 6104: User program error 6105: **watchdog time detecting over time** 6106: RAM address error	N
8063	Communication error	8063	6301：DTLK error 6302：RMIO error	Y
8064	Parameter error	8064	Error code 0000: No error 6401: program and parameter is not corresponding 6402: register capacity set error 6409: other error	N
8065	Syntax error	8065	Error code 0000: no error 6501: instruction address error 6504: pointer repeated 6505: device address is beyond range 6506: using undefined instruction 6507: Pointer error 6508: Interruption pointer error 6509: other	N
8066	Program error	8066	Error code 0000: no error 6603: MPS continuously used for more than 8 times 6604: MPS MRD MPP relation error 6605: STL continuously used for more than 10 times 6606: no known label 6607: Main program has I and SRET	N

				6609: CALL has more than 16 levels nest. 6610: for next are not corresponding 6611: with JCS and without JCR 6612: with STL and without RET 6613: with MC and without MCR 6614: with SMCS and without SMCR 6615: with I and without IRET 6616: MC, MCR I, SRET between 6617:for,next has more than 16 levels nest. 6618: no end 6621: other error	
8067	Operation error	8067	Error code	6705: address error 6706: parameter error 6730: sampling time out of range(Ts<0) 6732: PID input filter out of range 6733: PID proportional gain out of range (Kp) 6734: PID integral time const out of range.(TI) 6735: PID Derivative gain out of range.(KD) 6736: PID Derivative time const out of range 6740: PID sampling time ≤ scan cycle 6742: Overflow of variational value about measuring in PID 6743: Overflow of deviated value in PID 6744: Overflow of integration computation in PID 6745: Overflow of differential grain leads to overflow of differential value in PID 6746: Overflow of differential computation values in PID 6747: Overflow of PID computation results 6750: SV-PVnf<150,or system is unstable	Y

			6751: Large Overshoot of the Set Value 6752: Large fluctuations during Autotuning Set Process	
8069	I/O bus error	8069	6903: expansion I/O error 6904: expansion A/D error 6905:expansion A/D unit is power off	Y