pLocon

Ex1　Ex2

Programmable Logic Controller

User's　Manual

健昇科技股份有限公司

JS AUTOMATION CORP.

台北縣汐止市中興路100號6樓

6F, No.100, Chungshin Rd.

Shitsu, Taipei (221), Taiwan, R.O.C.

TEL：+886-2-2647-6936

FAX：+886-2-2647-6940

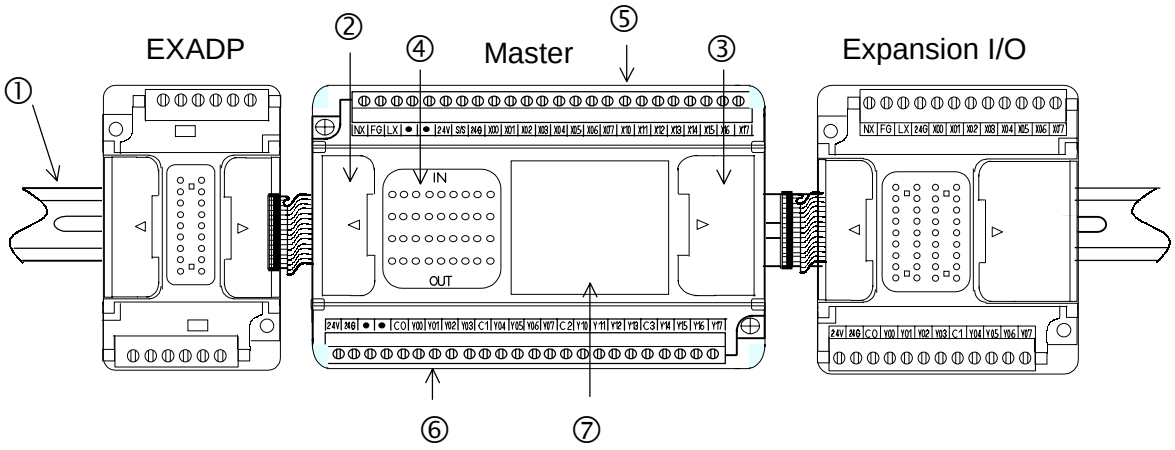
http://www.automation.com.tw/

Email : control.cards@automation.com.tw

CONTENTS

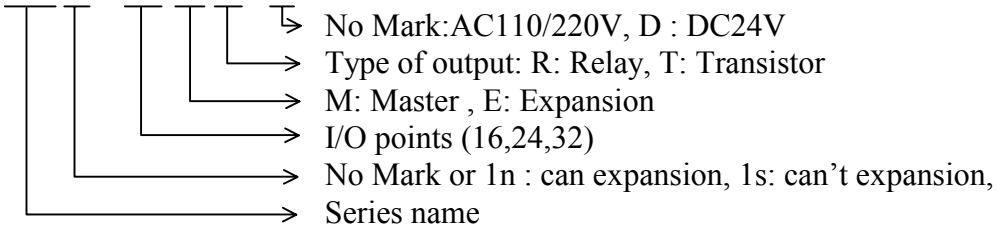
1. Specifications

◎ Master Unit & Expansion Unit

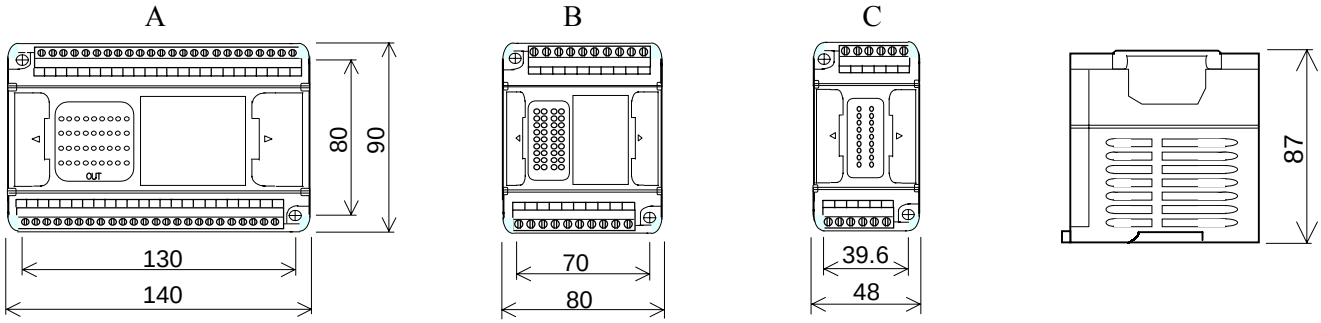


- Master unit, Expansion I/O unit, Expansion module and EXADP communication module all can assembly to ① (35mm)
- Open ③ connector cover, connected master unit and expansion i/o unit or expansion module.
- Open ② connector cover, connected master unit and EXADP communication module.
- ④ is the LED monitor of input relay, output relay, power, run status and error status.
- ⑤ is the terminal of input relay, ⑥ is the terminal of output relay.
- ⑦ is EEPROM card.

EX - 32 M R -



◎ Dimension (mm)



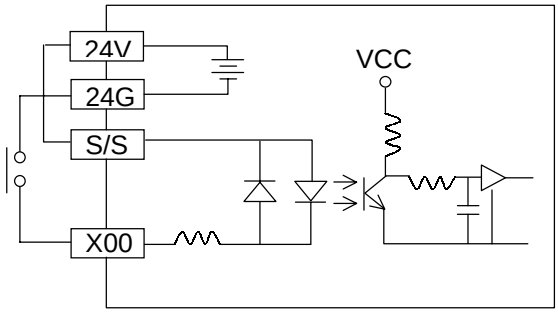
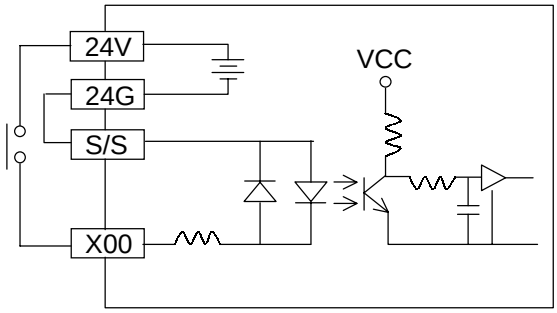
◎ Performance Specification

ITEM		Ex1s	Ex1n	Ex2n
Operating control method		Cyclic operation by stored program		
I/O control method		Batch processing method (when END instruction is executed)		
Operation time		Basic instruction 0.5us, Applied instruction from 2us to several 100us.		
Programming language		Relay symbolic language + Step ladder		
Program capacity / memory		8000 steps (built in EEprom)		
Number of instruction	General	Basic instruction:27, Step ladder instruction:2		
	Applied	106	108	119
Input Relay		Sink/Source DC24V 7mA photo coupler isolation		
		X00~X17	X000 ~ X177	
Output Relay		Relay : AC250V/1A or Transistor : DC30V/0.5A		
		Y00~Y17	Y000 ~ Y177	
Auxiliary Relay (M)	Latched	M000 ~ M499 (EEprom backup)		
	General	M500 ~ M1535 (no backup)		
	Special	M8000 ~ M8255 (no backup)		
State Relay (S)	Latched	S000 ~ S499 (EEprom backup)		
	General	S500 ~ S999 (no backup)		
Timer (T)	100 msec	T000 ~ T199 (no backup)		
	10 msec	T200 ~ T245 (no backup)		
	1 ms integration	4 points, T246 ~ T249 (EEPROM backup)		
	100 ms integration	6 points, T250 ~ T255 (EEPROM backup)		
	Analog	2 points (Define by user)		
Counter (C)	16bits Counter	Latched C00 ~ C31 (EEprom backup)		
		General C32 ~ C199		
	32bits Counter	General C200 ~ C215		
		Latched C216 ~ C255 (EEprom backup)		
	High Speed Counter	6 points : X0 ~ X5 ; X0 or X1 for 1 phase 60KHz , X2 ~ X5 for 1phase 10KHz		
		X0 and X1 for 2 phase 30KHz , X2 ~ X5 for 2phase 5KHz		
Data Register	Latched	D000 ~ D255 (EEprom backup)		
	General	D256 ~ D3999 (can used FNC(12) MOV stored at EEPROM)		
	Special	D8000 ~ D8255 (no backup)		
Index		V0 ~ V7, Z0 ~ Z7		
Pointer (P)	JMP,CALL	P000 ~ P127		
Pointer (I)	Interrupt (I)	I0xx ~ I8xx		
Nest	Nest (N)	N0 ~ N7		
Communication Interface The 2-nd port (Option)		RS-232C & RS-232C/RS-422,RS-485		
Calendar	(Option)	Week, Year, Month, Day, Hour, Minute, Second		
Constant(K)	Decimal	16 bits: -32,768 ~ +32,767		
		32 bits: -2,147,483,648 ~ +2,147,483,647		
Constant(H)	Hexadecimal	16 bits: 0000h ~ FFFFh		
		32 bits: 00000000h ~ FFFFFFFFh		

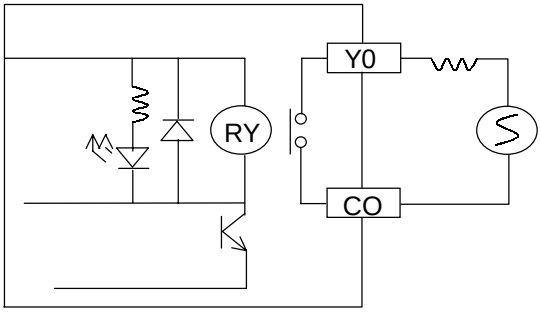
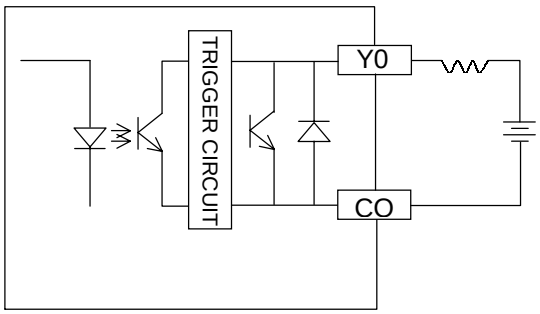
◎ General Specification

Item	Description
Source Voltage	AC 85~264 V 50/60 Hz
Supply current	24VDC / 800 mA
Momentary power failure	Keep operation in 10 ms
Breakdown voltage	AC1500/1min (between output terminal and frame ground terminal)
Isolation resistance	DC500v/5mΩ
Noise Impedance	Noise voltage: 1000Vp-p, noise width: 1 us
Grounding	Class 3 ground
Ambient Temperature	0 ~ 55°C
Ambient humidity	35 ~ 85 RH (without condensation)
Atmosphere	Must be free from corrosive gasses

◎ Input Specification

Item	DC input (Sink)	DC input (Source)
Circuit		
Input voltage	DC24V+10%, -15%	DC24V+10%, -15%
Input current	7mA / DC24V	7mA / DC24V
Impedance	3.3 KΩ	3.3 KΩ
Response time	About 10 ms (X00~X07 High Speed)	About 10 ms (X00~X07 High Speed)
Input pattern	No voltage contact or NPN open collector	No voltage contact or PNP open collector
Circuits isolation	Photo coupler	Photo coupler

◎ Output Specification

Item	Relay output	Transistor output
Circuit		
Load voltage	Under AC250V DC30V	DC5V ~ 30V
Rated current	2A / 1 point	0.5A / 1 point
Rated capacity	100W	12W
Response time	About 10ms	Under 1 ms
Circuits isolation	Machine isolation	Photo coupler

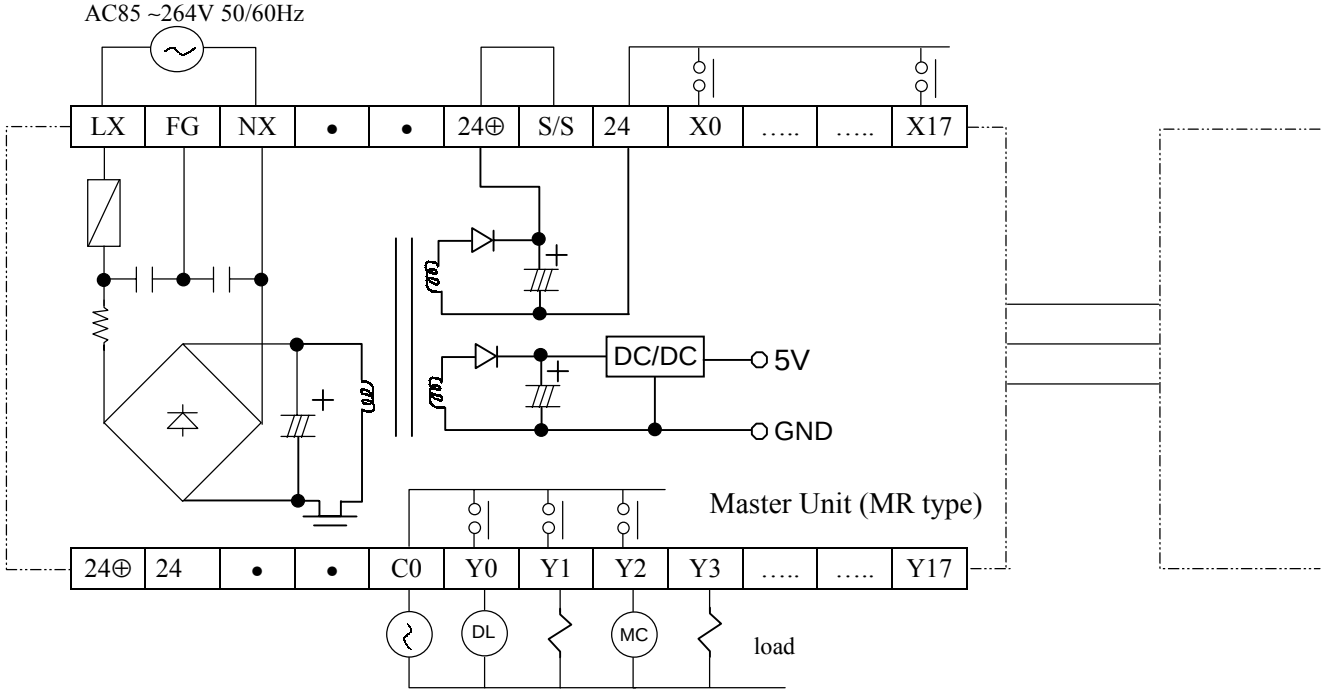
◎ Note of Output Specification

Ex1s Ex1n Transistor output module haven't pulled high resistor 2.2K
Ex32MT-P Transistor output module have pulled high resistor 2.2K

◎ Source Power Wiring Diagram (NPN Mode)

(24⊕, 24 is output power source from PLC)

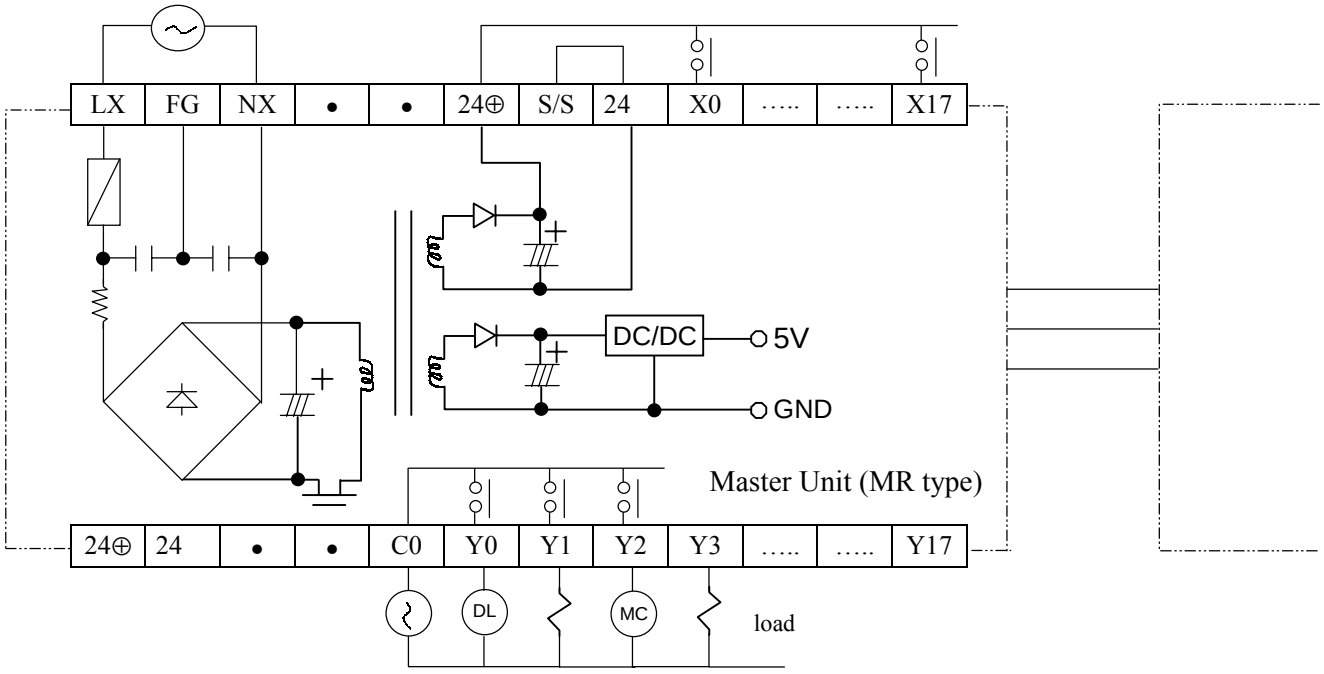
AC85 ~264V 50/60Hz



◎ Source Power Wiring Diagram (PNP Mode)

(24⊕, 24 is output power source from PLC)

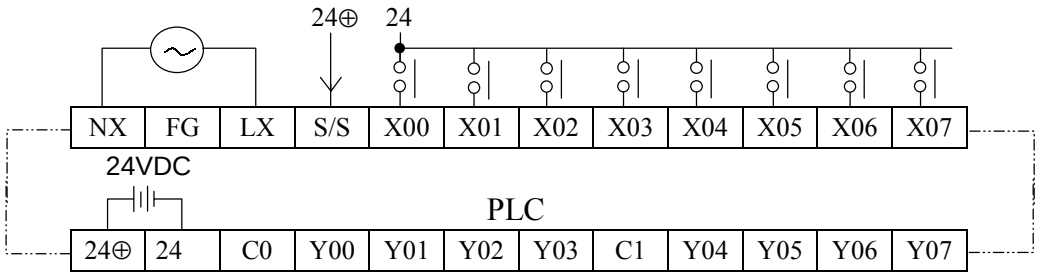
AC85 ~264V 50/60Hz



◎ 16MR Type Terminal Signal (24⊕ → S/S is NPN mode，24 → S/S is PNP mode)

(24⊕, 24 are output power source from PLC)

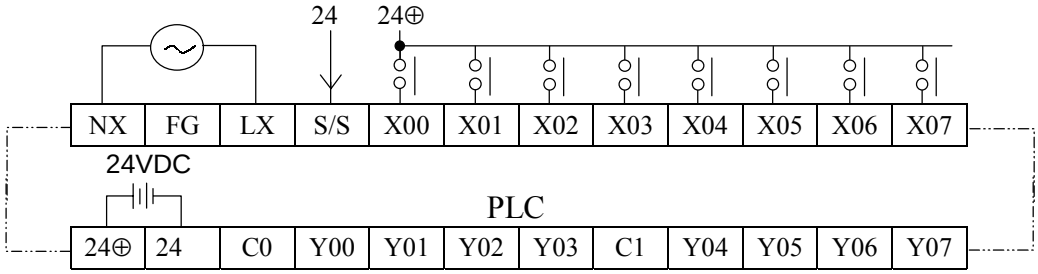
AC85 ~264V 50/60Hz



◎ 16MR Type Terminal Signal (PNP mode Source)

(24⊕, 24 are output power source from PLC)

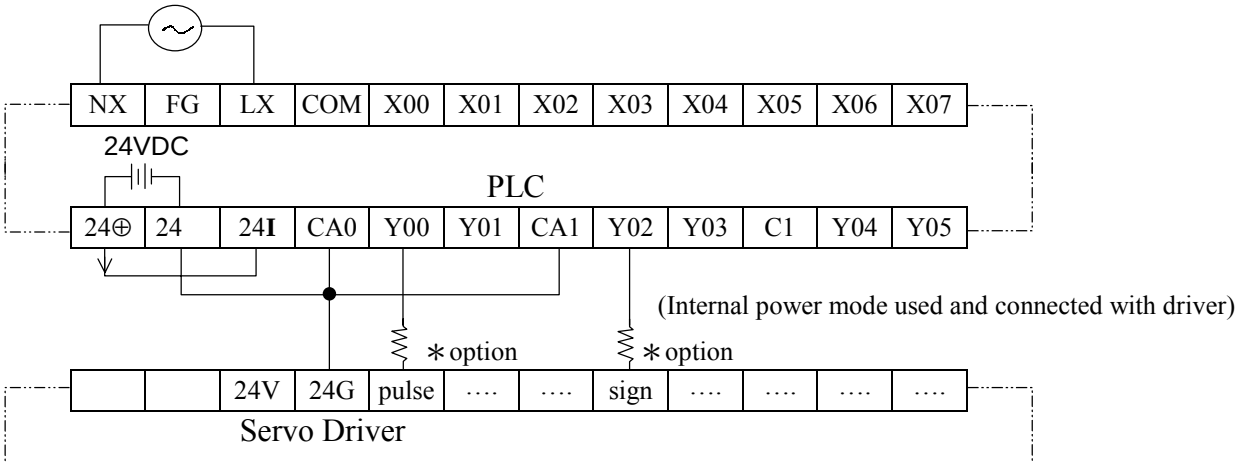
AC85 ~264V 50/60Hz



◎ 14MT Type Terminal Signal and Wiring Diagram

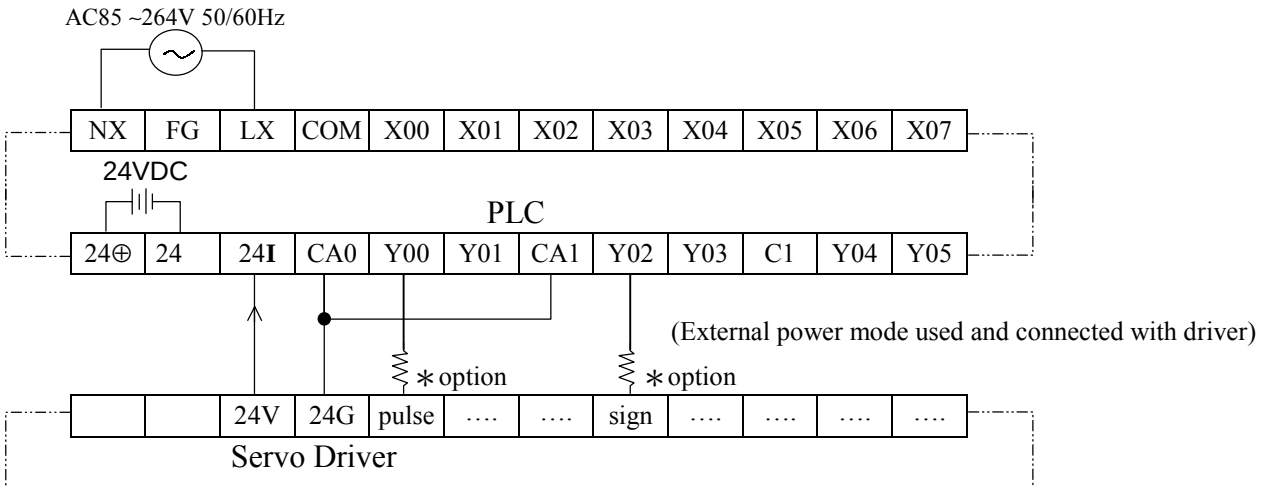
(24⊕, 24 are output power source from PLC)

AC85 ~264V 50/60Hz



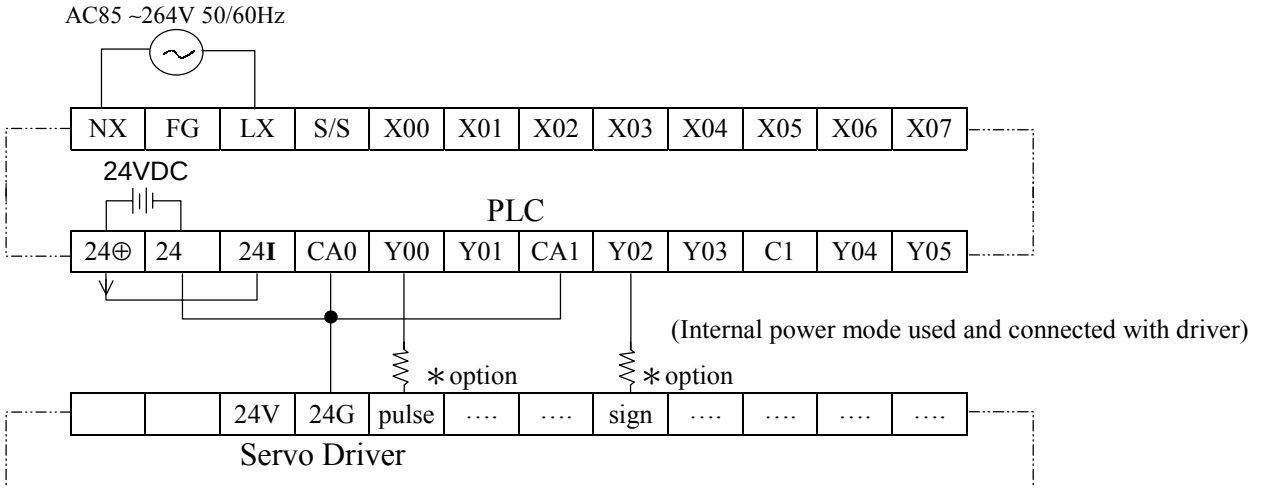
◎ 14MT Type Terminal Signal and Wiring Diagram

(24⊕, 24 are output power source from PLC)



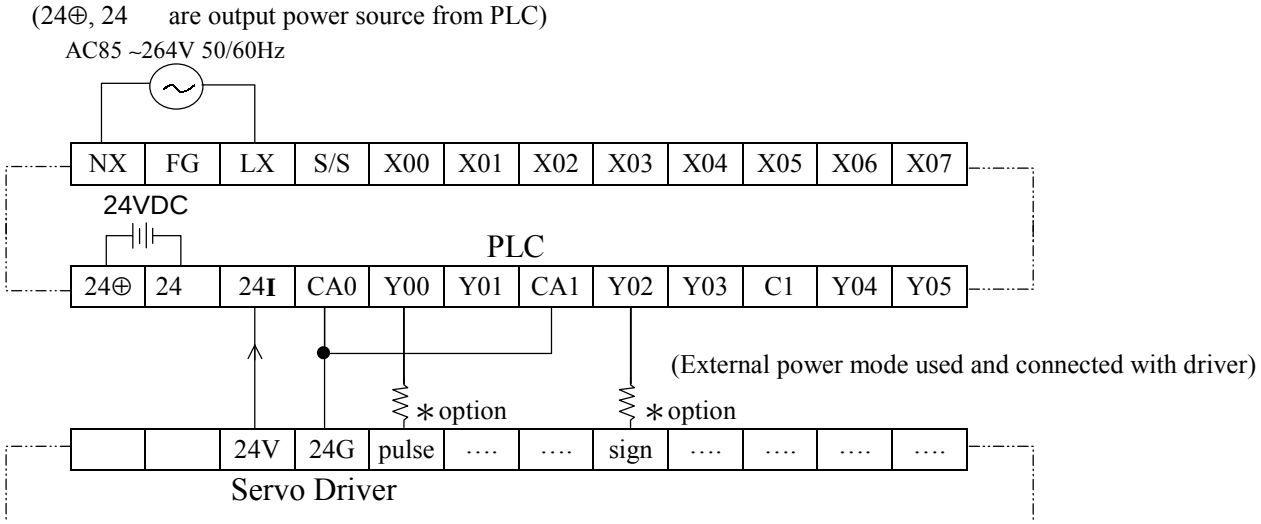
◎ 14MT Type Terminal Signal and Wiring Diagram (24⊕ → S/S is NPN mode, 24 → S/S is PNP mode)

(24⊕, 24 are output power source from PLC)



◎ 14MT Type Terminal Signal and Wiring Diagram (24⊕ → S/S is NPN mode, 24 → S/S is PNP mode)

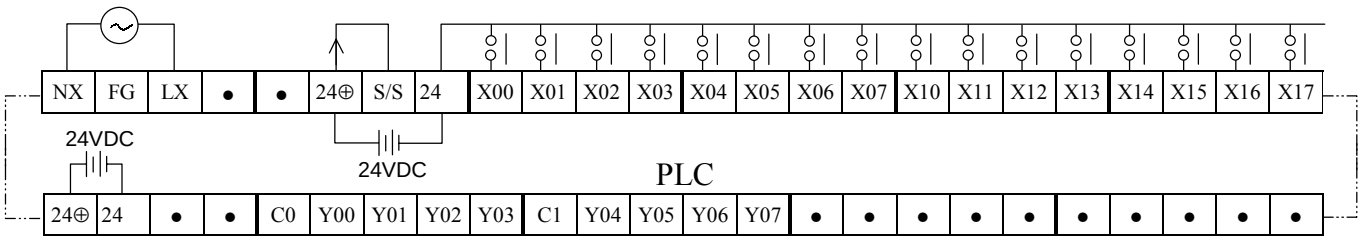
(24⊕, 24 are output power source from PLC)



◎ 24MR Type Terminal Signal (24⊕ → S/S is NPN mode, 24 → S/S is PNP mode)

(24⊕, 24 are output power source from PLC)

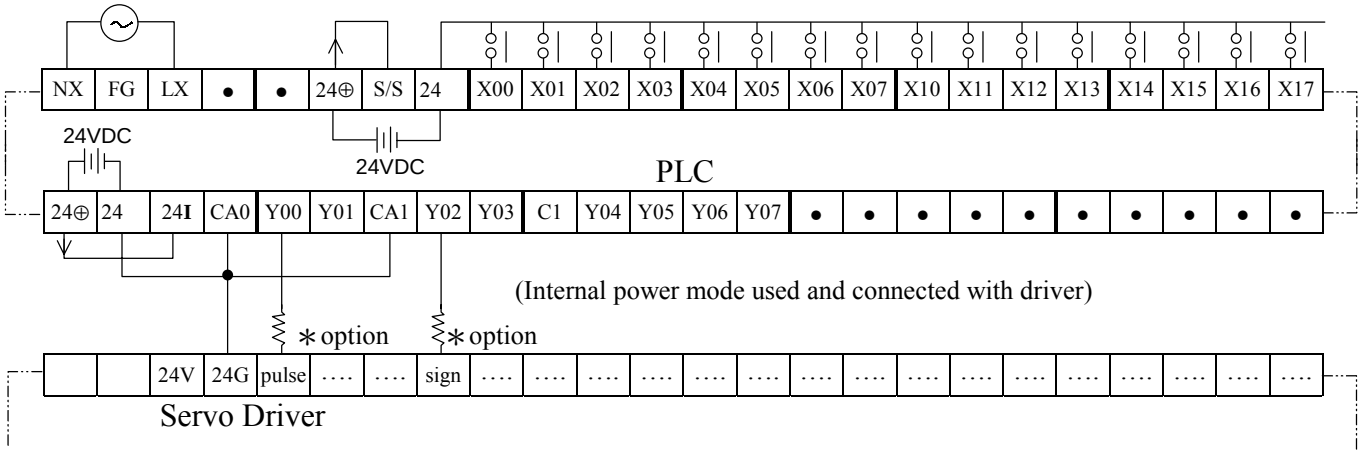
AC85 ~264V 50/60Hz



◎ 24MT Type Terminal Signal and Wiring Diagram (24⊕ → S/S is NPN mode, 24 → S/S is PNP mode)

(24⊕, 24 are output power source from PLC)

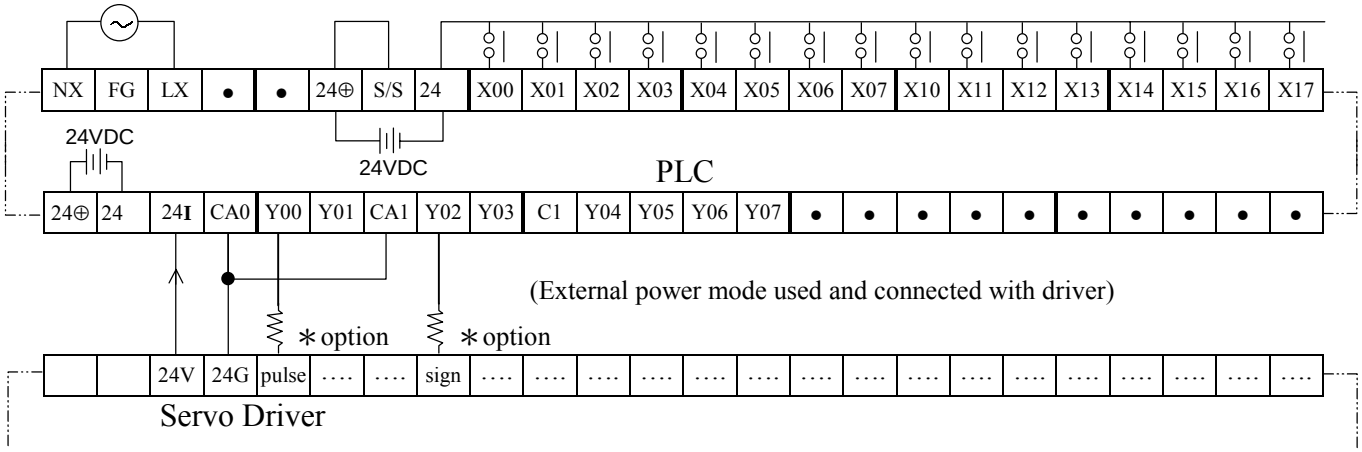
AC85 ~264V 50/60Hz



◎ 24MT Type Terminal Signal and Wiring Diagram (24⊕ → S/S is NPN mode, 24 → S/S is PNP mode)

(24⊕, 24 are output power source from PLC)

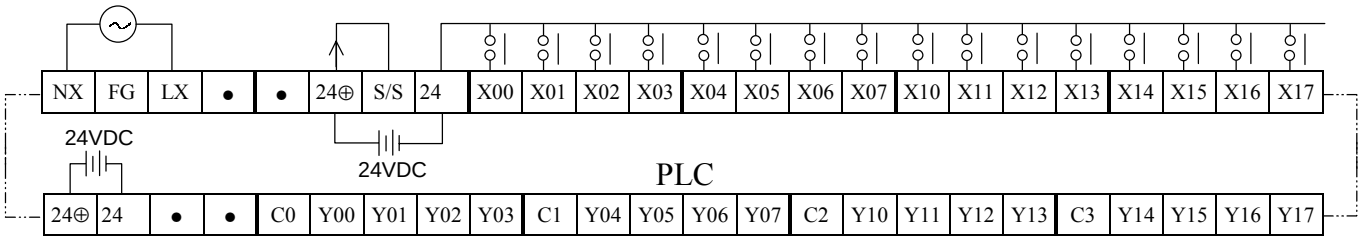
AC85 ~264V 50/60Hz



◎ 32MR Type Terminal Signal (24⊕ → S/S is NPN mode, 24 → S/S is PNP mode)

(24⊕, 24 are output power source from PLC)

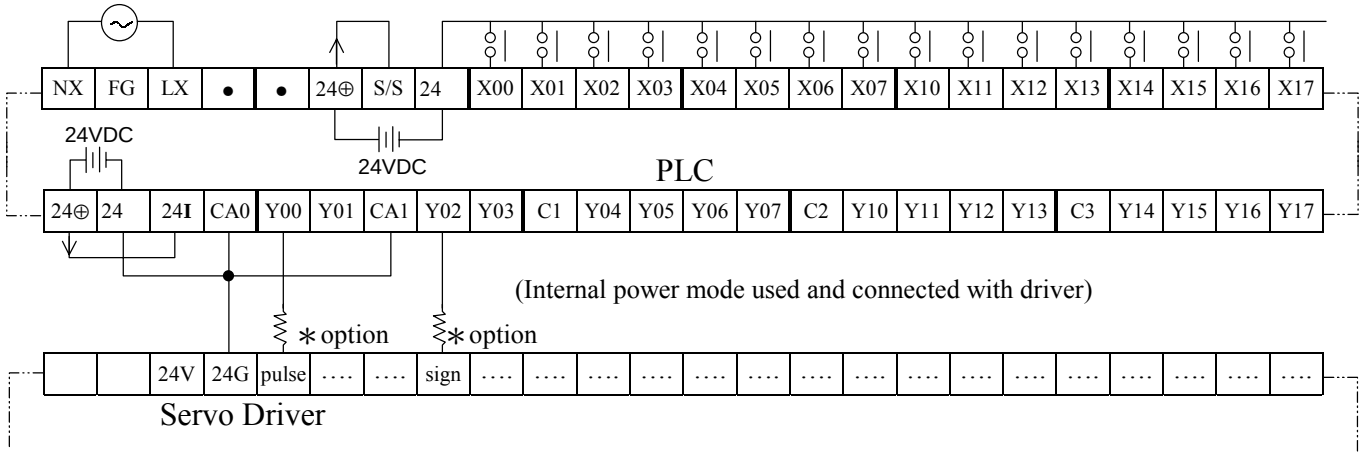
AC85 ~264V 50/60Hz



◎ 32MT Type Terminal Signal and Wiring Diagram (24⊕ → S/S is NPN mode, 24 → S/S is PNP mode)

(24⊕, 24 are output power source from PLC)

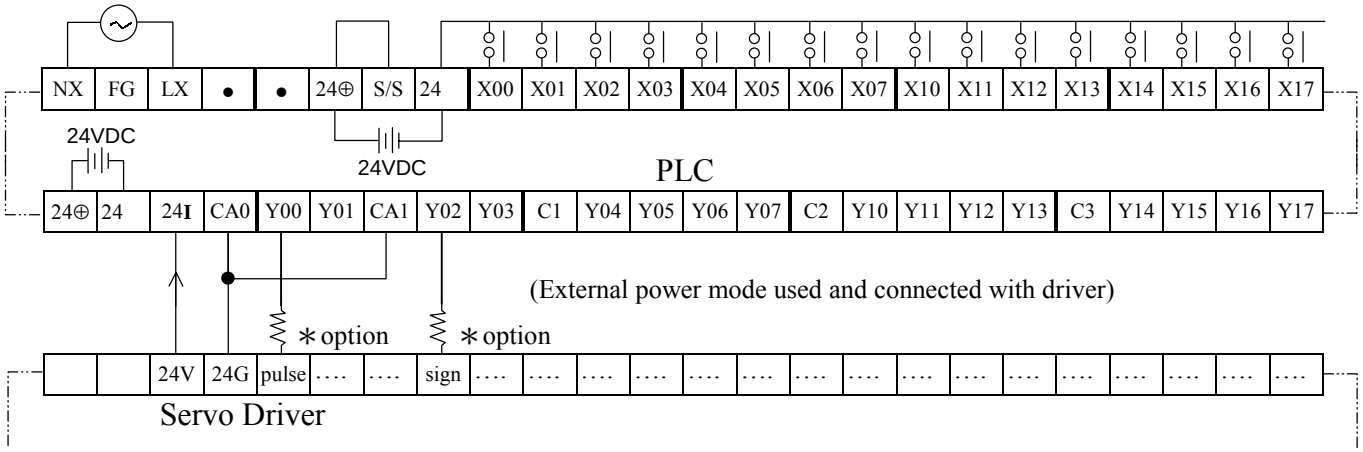
AC85 ~264V 50/60Hz



◎ 32MT Type Terminal Signal and Wiring Diagram (24⊕ → S/S is NPN mode, 24 → S/S is PNP mode)

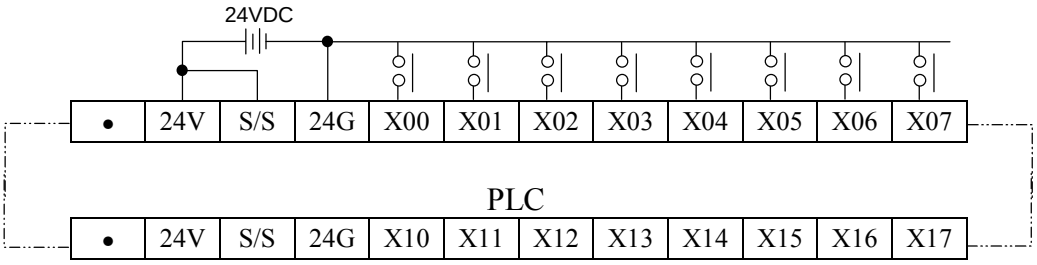
(24⊕, 24 are output power source from PLC)

AC85 ~264V 50/60Hz



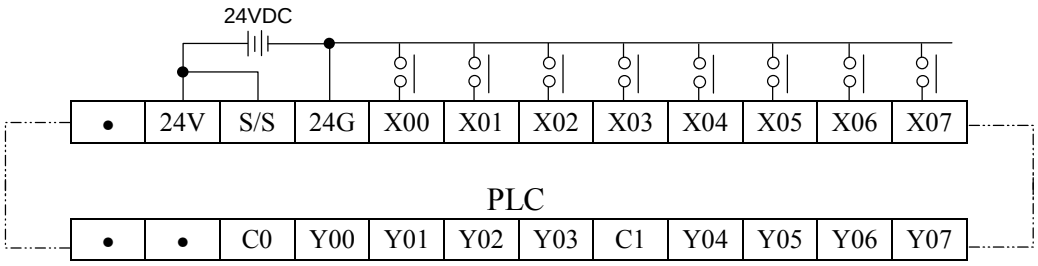
◎ 16EX Type Terminal Signal (24V → S/S is NPN mode, 24G → S/S is PNP mode)

(24V, 24G are external power source input terminal)

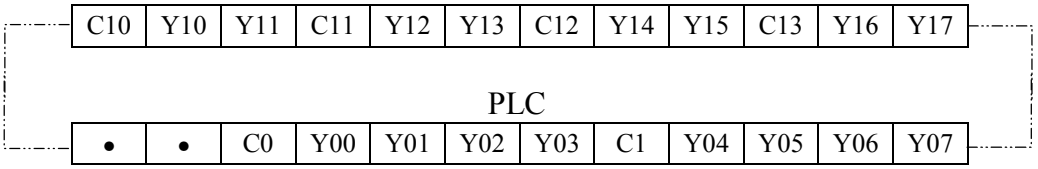


◎ 16ER, 16ET Type Terminal Signal (24V → S/S is NPN mode, 24G → S/S is PNP mode)

(24V, 24G are external power source input terminal)

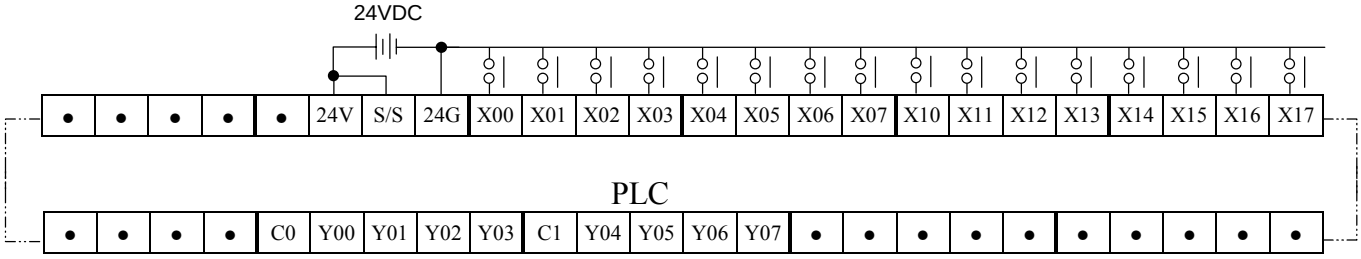


◎ 16EYR, 16EYT Type Terminal Signal (not need external power source input)



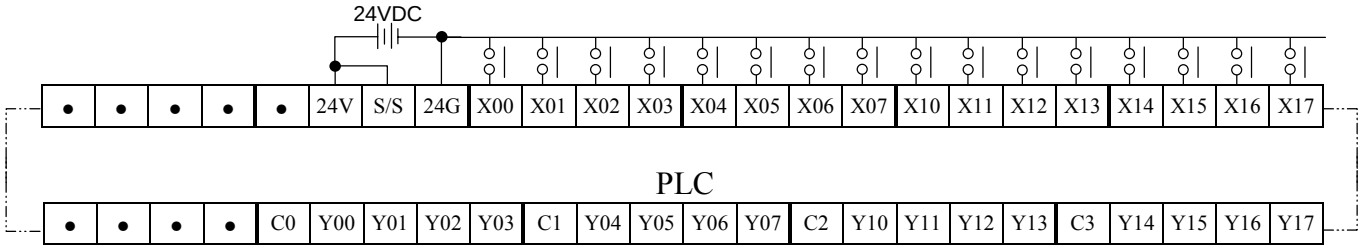
◎ 24ER, 24ET Type Terminal Signal (24V → S/S is NPN mode, 24G → S/S is PNP mode)

(24V, 24G are external power source input terminal)



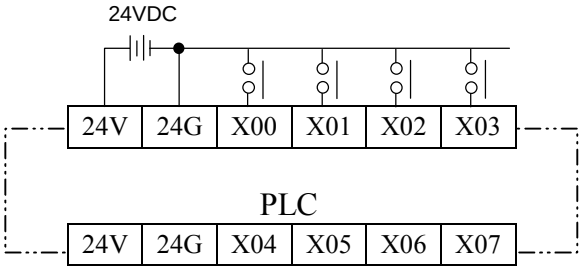
◎ 32ER, 32ET Type Terminal Signal (24V → S/S is NPN mode, 24G → S/S is PNP mode)

(24V, 24G are external power source input terminal)



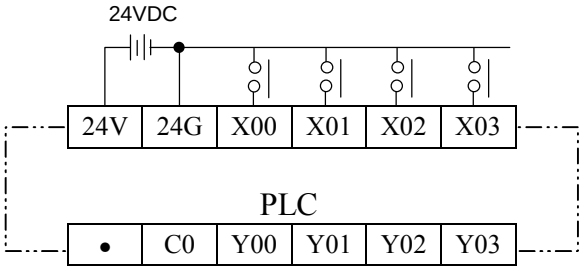
◎ 8EX Type Terminal Signal

(24V, 24G are external power source input terminal)



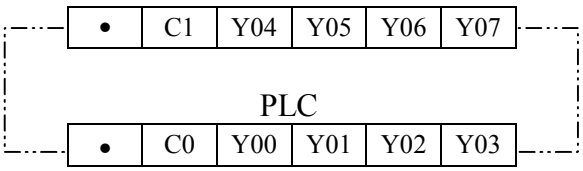
◎ 8ER, 8ET Type Terminal Signal

(24V, 24G are external power source input terminal)



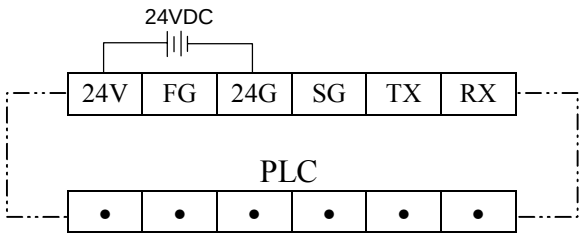
◎ 8EYR, 8EYT Type Terminal Signal

(not need external power source input)



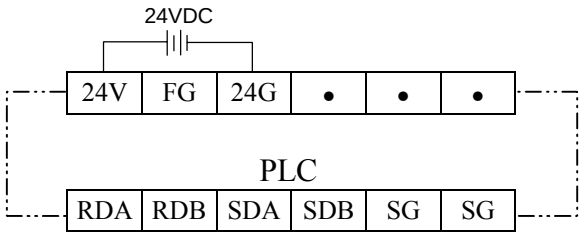
◎ 232ADP Type Terminal Signal

(24V, 24G are external power source input terminal)



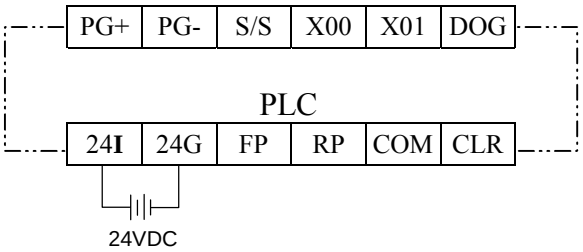
◎ 485ADP Type Terminal Signal

(24V, 24G are external power source input terminal)

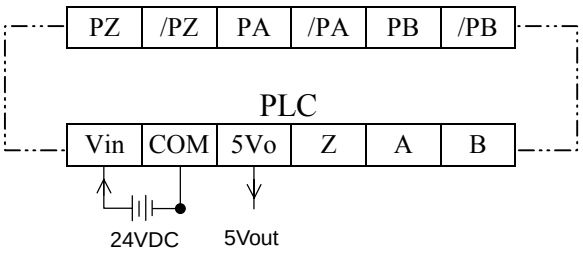


◎ 1PG Type Terminal Signal

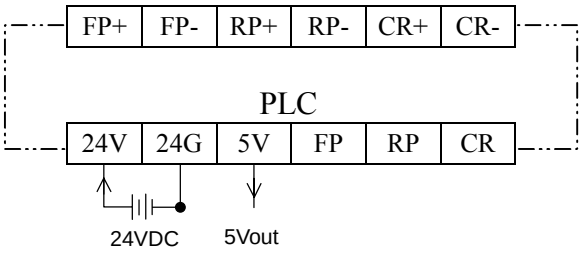
(24V → S/S are NPN mode, 24G → S/S are PNP mode)



◎ LTOC Type Terminal Signal

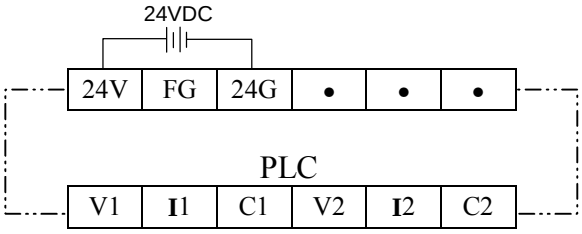


◎ CTOL Type Terminal Signal



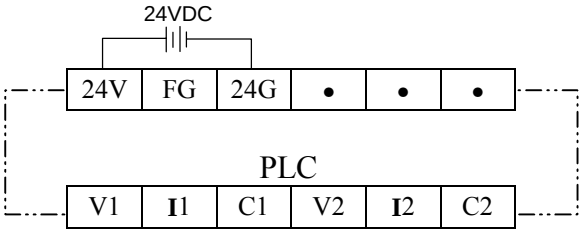
◎ 2DA Type Terminal Signal

(24V, 24G are external power source input terminal)



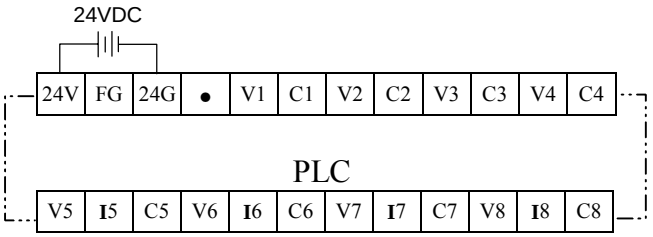
◎ 2AD Type Terminal Signal

(24V, 24G are external power source input terminal)



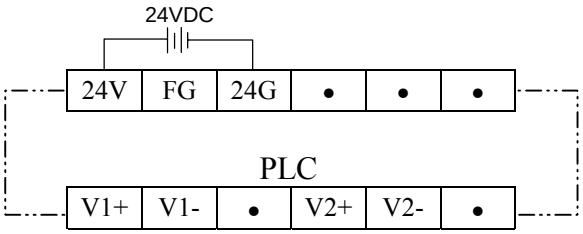
◎ 8AD Type Terminal Signal

(24V, 24G are external power source input terminal)



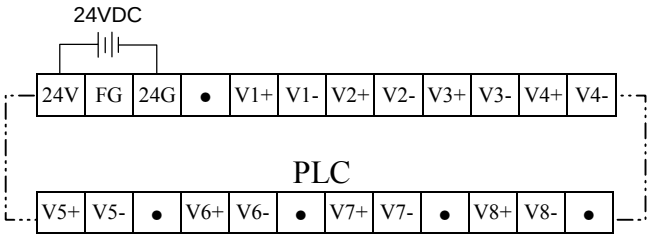
◎ 2TC Type Terminal Signal

(24V, 24G are external power source input terminal)



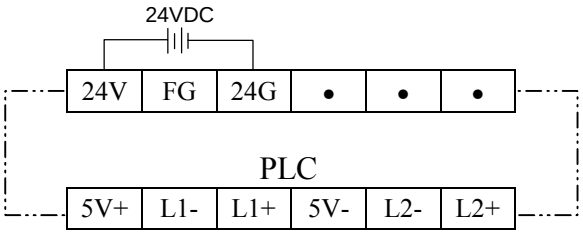
◎ 8TC Type Terminal Signal

(24V, 24G are external power source input terminal)



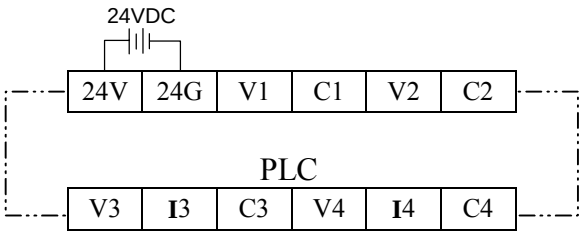
◎ 2LD Type Terminal Signal

(24V, 24G are external power source input terminal)



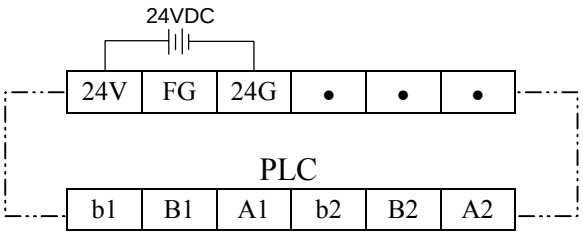
◎ 4AD Type Terminal Signal

(24V, 24G are external power source input terminal)



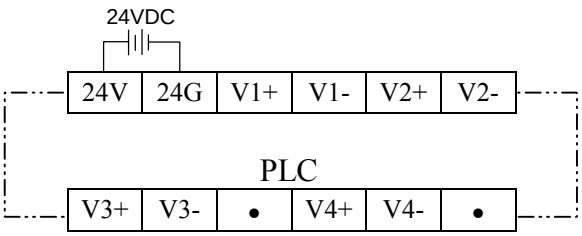
◎ 2PT Type Terminal Signal

(24V, 24G are external power source input terminal)

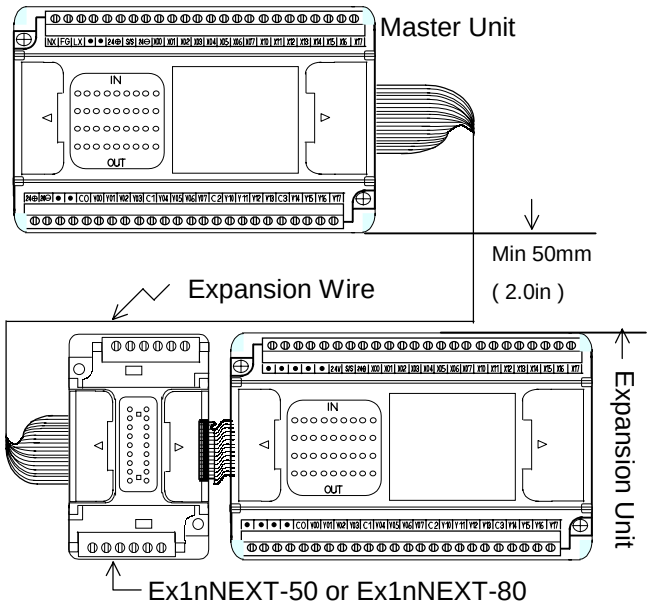


◎ 4TC Type Terminal Signal

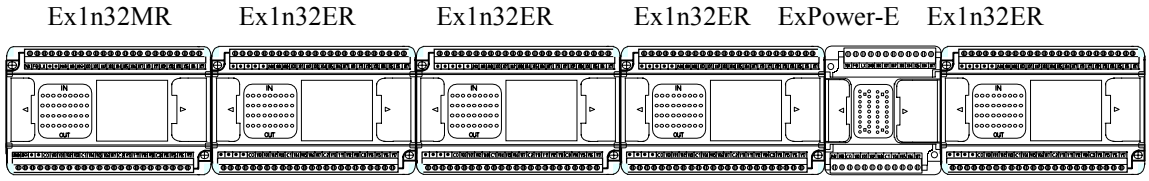
(24V, 24G are external power source input terminal)



◎ Note for Wiring

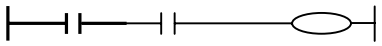
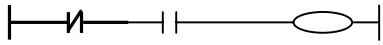
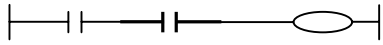
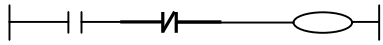
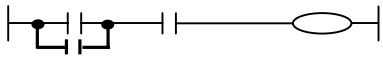
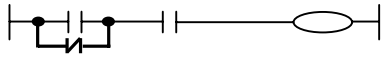
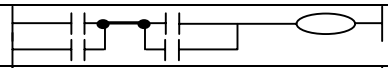
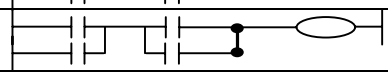
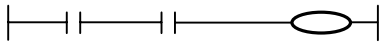
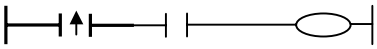
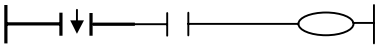
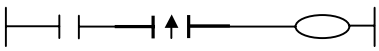
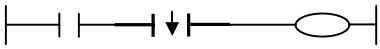
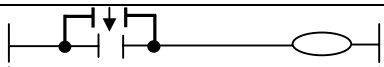
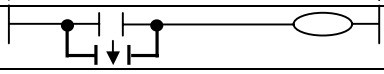
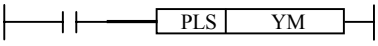
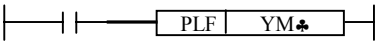
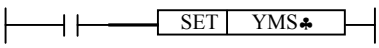
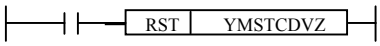
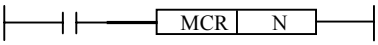
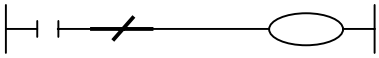
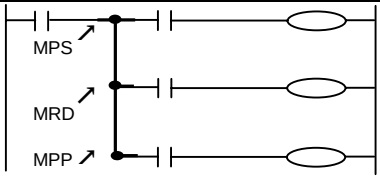


- ♦ Don't wire it to the mark (•) of terminal.
- ♦ Can't use the same cable for the signal wire of input and output.
- ♦ Don't put the signal cable of input and output with power cable at the same tube.
- ♦ The expansion module with power device, so can't wire the 24⊕ of expansion module to the 24⊕ of master.
- ♦ There is no power device in expansion I/O unit, so have to connect 24⊕ of master unit to 24V or 24I of expansion I/O unit, otherwise can't input signal.
- ♦ If there is no enough space, but have to arrange it to two lines, then can install wire extension module (50cm length of Ex1nNEXT-50 or 80cm length of Ex1nNEXT-80), as left picture.
- ♦ In principle, when system is more than 128 points, then have to install power extension module (ExPower-E), as below picture.



2. Basic Instructions

◎ List of Basic Instruction

Symbol	Function	Circuit & Devices
LD LoaD	Each logic start A contact	 X,Y,M,S,T,C
LDI LoaD Inverse	Each logic start B contact	 X,Y,M,S,T,C
AND AND	Serial connection A contact	 X,Y,M,S,T,C
ANI ANd Inverse	Serial connection B contact	 X,Y,M,S,T,C
OR OR	Parallel connection A contact	 X,Y,M,S,T,C
ORI OR Inverse	Parallel connection B contact	 X,Y,M,S,T,C
ANB ANd Block	Serial connection of Parallel circuit	
ORB OR Block	Parallel connection of serial circuit	
OUT OUT	Final operation coil drive	 Y,M,S,T,C
LDP LoaD rising Pulse	Initial logical operation Rising edge pulse	 X,Y,M,S,T,C
LDF LoaD Falling pulse	Initial logical operation Falling edge pulse	 X,Y,M,S,T,C
ANDP AND Pulse	Serial connection of Rising edge pulse	 X,Y,M,S,T,C
ANDF AND Falling	Serial connection of Falling edge pulse	 X,Y,M,S,T,C
ORP OR Pulse	Parallel connection of Rising edge pulse	 X,Y,M,S,T,C
ORF OR Falling	Parallel connection of Falling edge pulse	 X,Y,M,S,T,C
NOP NOP	No operation	N/A
PLS PULSE	Rising edge pulse	 PLS YM
PLF PLF	Falling edge pulse	 PLF YM♣
SET SET	Set a bit device Permanently ON	 SET YMS♣
RST ReSeT	Reset a bit device Permanently OFF	 RST YMSTCDVZ
MC Master Control	Denote the start of Master control block	 MC N YM
MCR Master Control Reset	Denote the end of Master control block	 MCR N
INV INVerse	Invert the current result of the internal PLC operations	
MPS PuSh	Push the result of operation to stack	 MPS MRD MPP
MRD ReaD	Read the result of operation from stack	
MPP PoP	Pop & remove the Result from stack	
END END	Main program end	Forced the current program scan to step 0

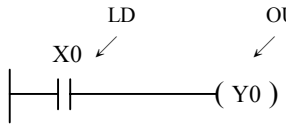
♣ : Special Auxiliary Relay

◎ Load & Load Inverse & Out Instruction

EX	EX_{1S}	EX_{1N}	EX_{2N}

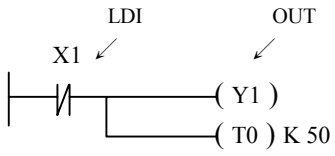
Mnemonic	Instruction	Symbol & Device	Step number
LD	LoaD	X,Y,M,S,T,C	1
LDI	LoaD Inverse	X,Y,M,S,T,C	1
OUT	OUT	Y,M,S,T,C	1

♦ If each logic line start an NO contact, use the LD instruction.




```
LD    X   0
OUT   Y   0
```

♦ If each logic line start an NC contact, use the LDI instruction.




```
LDI   X   1
OUT   Y   1
OUT   T   0
(SP)  K   50
```

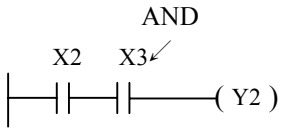
♦ When use hand held programmer, the space key needs to be pressed to enable the constant to be entered.

◎ AND & AND Inverse Instruction

EX	EX_{1S}	EX_{1N}	EX_{2N}

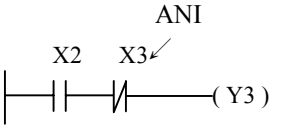
Mnemonic	Instruction	Symbol & Device	Step number
AND	AND	X,Y,M,S,T,C	1
ANI	ANd Inverse	X,Y,M,S,T,C	1

♦ If an NO contact is connected in series, use the AND instruction.

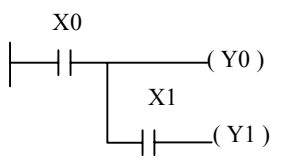



```
LD    X   2
AND   X   3
OUT   Y   2
```

♦ If an NC contact is contacted in series, use the ANI Instruction.



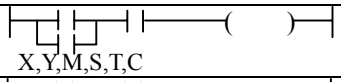
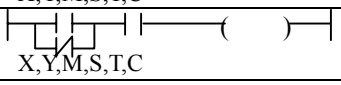

```
LD    X   2
ANI   X   3
OUT   Y   3
```



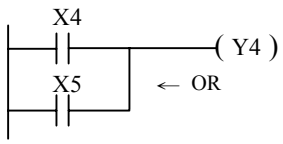

```
LD    X   0
OUT   Y   0
AND   X   1
OUT   Y   1
```

◎ OR & OR Inverse Instruction

EX	EX_{1S}	EX_{1N}	EX_{2N}

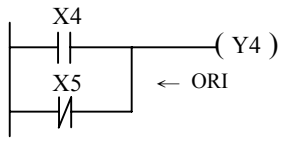
Mnemonic	Instruction	Symbol & Device	Step number
OR	OR		1
ORI	OR Inverse		1

♦ If an NO contact is connected in parallel, use the OR instruction.




```
LD    X   4
OR    X   5
OUT   Y   4
```

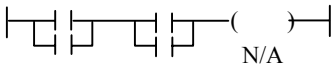
♦ If an NC contact is connected in parallel, use the ORI instruction.



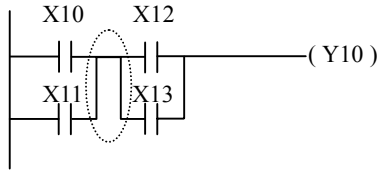

```
LD    X   4
ORI   X   5
OUT   Y   4
```

◎ ANB Instruction

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
ANB	ANd Block		1

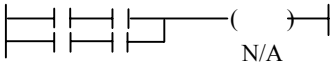
♦ Serial connection of parallel circuit, use the ANB



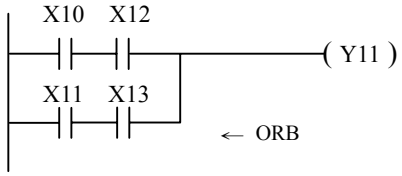

```
LD    X   10
OR    X   11
LD    X   12
OR    X   13
ANB
OUT   Y   10
```

◎ ORB Instruction

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
ORB	OR Block		1

♦ Serial connection of parallel circuit, use the ORB

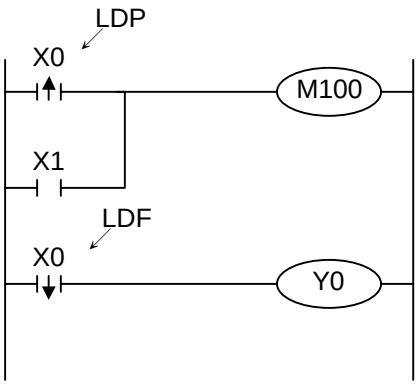



```
LD    X   10
AND   X   12
LD    X   11
AND   X   13
ORB
OUT   Y   11
```

◎ Load Pulse & Load Falling Pulse Instruction

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
LDP	LoaD rising Pulse	X,Y,M,S,T,C	2
LDF	LoaD Falling pulse	X,Y,M,S,T,C	2



☞ 0 LDP X 0
2 OR X 1
3 OUT M 100
☞ 4 LDF X 0
6 OUT Y 0

Basic points to remember:

- ♦ Connect LDP and LDF instructions directly to the left hand bus bar.
- ♦ Or use LDP and LDF instructions to define a new block of program when use ORB and ANB instructions (see later sections).
- ♦ LDP is active for one program scan after the associated device switches from OFF to ON.
- ♦ LDF is active for one program scan after the associated device switches from ON to OFF.

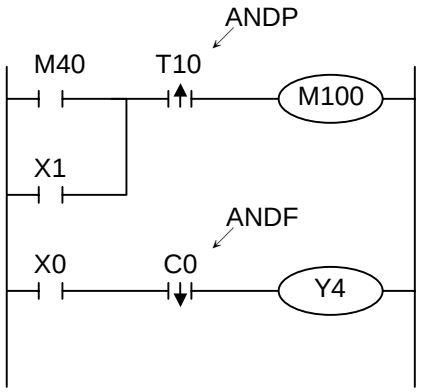
Single Operation flags M2800 to M3071

- ♦ When the pulse operation instructions used with auxiliary relays M2800 to M3071, only activate the first instruction encountered in the program scan, after the point in the program where the device changes. Any other pulse operation instructions will remain inactive.
- ♦ This is useful for use in STL programs (see chapter 3) to perform single step operation using a single device.
- ♦ Any other instructions (LD, AND, OR, etc.) will operate as expected.

◎ AND Pulse & AND Falling Pulse Instruction

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
ANDP	AND Pulse	X,Y,M,S,T,C	2
ANDF	AND Falling pulse	X,Y,M,S,T,C	2




```
    0   LD    M   40
    1   OR    X   1
☞   2   ANDP  T   10
    4   OUT   M   100
    5   LDF   X   0
☞   6   ANDF  C   0
    8   OUT   Y   4
```

Basic points to remember:

- ♦ Use the ANDP and ANDF instructions for the serial connection of pulse contacts.
- ♦ Usage is the same as for AND and ANI; see earlier.
- ♦ ANDP is active for one program scan after the associated device switches from OFF to ON.
- ♦ ANDF is active for one program scan after the associated device switches from ON to OFF.

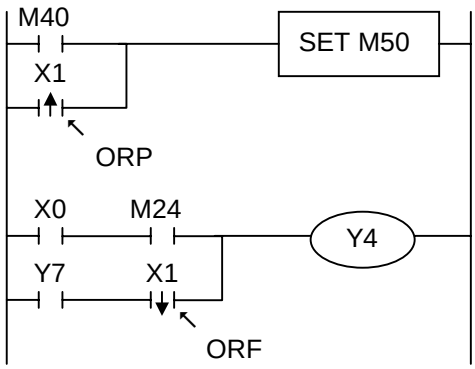
Single operation flags M2800 to M3071:

- ♦ When used with flags M2800 to M3071, only the first instruction will activate.

◎ OR Pulse & OR Falling Pulse Instruction

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
ORP	OR Pulse	X,Y,M,S,T,C	2
ORF	OR Falling pulse	X,Y,M,S,T,C	2




```
0   LD   M 40
1   ORP  X 1
3   SET  M 50
4   LD   X 0
5   AND  M 24
6   LD   Y 7
7   ORF  X 1
9   ORB
10  OUT  Y 4
```

Basic points to remember:

- ♦ Use the ORP and ORF instructions for the parallel connection of pulse contacts.
- ♦ Usage is the same as for OR and ORI; see earlier.
- ♦ ORP is active for one program scan after the associated device switches from OFF to ON.
- ♦ ORF is active for one program scan after the associated device switches from ON to OFF.

Single operation flags M2800 to M3071

- ♦ When used with flags M2800 to M3071, only the first instruction will activate.

◎ NOP & END Instruction

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
NOP	NOP	N/A	1
END	END	END	1

NOP Instruction

- ♦ After the program "all clear operation" is executed, all instructions in the program are over written with NOP's.

END INSTRUCTION

- ♦ Insert this instruction at the end of a program and program return to step 0.
- ♦ If missing this instruction, then program can't be executed.

NOTE :

1 : Program a circuit from its up to down and left to right.

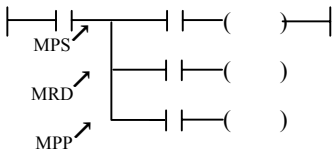
2 : Output relay can't be connected directly from the bus bar.
If necessary, connect it through the N/C contact of special auxiliary relay M8000.

3 : I/O relay, inside auxiliary relay, TIM/OUT the number of contact that can be used per output relay is not limited.

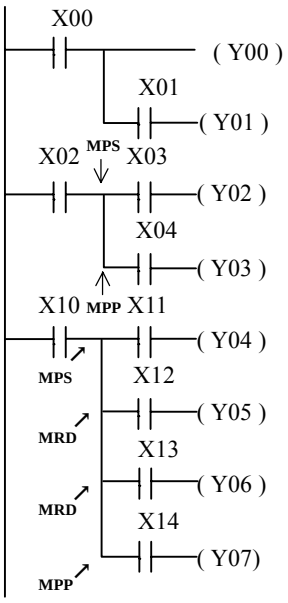
4 : Behind the output coil can not in addition contact; Two or more output coils can be connected in parallel.

◎ Multiplex output circuit

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
MPS	Memory PuSh		1
MRD	Memory ReaD		1
MPP	Memory PoP		1

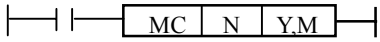
- In pLocon EX series, there are (11) stack memory space can store operation result, so MPS instruction may be used up to (11) times continuously.
- MPS: Push the operation result into stack and the stack pointer increment by 1.
- MRD: Read the operation result from stack and the stack pointer unchanged.
- MPP: Pop the operation result from stack. First the stack pointer decrement by 1.
- MPS,MRD,MPP are all no operand.

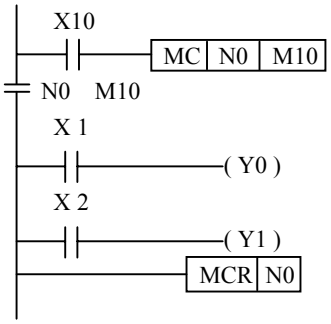


LD	X	00
OUT	Y	00
AND	X	01
OUT	Y	01
LD	X	02
MPS		
AND	X	03
OUT	Y	02
MPP		
AND	X	04
OUT	Y	03
LD	X	10
MPS		
AND	X	11
OUT	Y	04
MRD		
AND	X	12
OUT	Y	05
MRD		
AND	X	13
OUT	Y	06
MPP		
AND	X	14
OUT	Y	07

◎ Master Control (MC/MCR)

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
MC	Master control		2
MCR	Master Control Reset		1



LD	X	10
MC	N	0
SP	M	10
LD	X	1
OUT	Y	0
LD	X	2
OUT	Y	1
MCR	N	0

- ♦ N is the nesting level number.
- ♦ The MC/MCR instructions are used in pairs when branch a circuit to plural OUT instruction.
- ♦ When the MC condition is ON, the state of each relay is the same as in an ordinary circuit with out MC/MCR instruction.
- ♦ When the MC condition is OFF, the state of each relay between the MC and MCR instruction is as following:

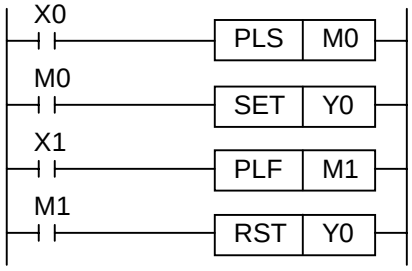
Time, Device for OUT	Reset & OFF
Counter, Device for SET	Hold present state

- ♦ Be sure that an LD/LDI instruction will be always following MC/MCR instruction.

◎ Inverse (INV)

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
INV	INVerse	N/A	1




```
0   LD   X 0
1   PLS  M 0
3   LD   M 0
4   SET  Y 0
5   LD   X 1
6   PLF  M 1
8   LD   M 1
9   RST  Y 0
```

Basic points to remember:

- The INV instruction is used to change (invert) the logical state of the current ladder network at the inserted position.
- Usage is the same as for AND and ANI; see earlier.

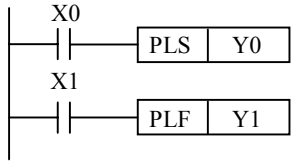
Usages for INV

- Use the invert instruction to quickly change the logic of a complex circuit.
  It is also useful as an inverse operation for the pulse contact instruction LDP, LDF, ANP, etc.

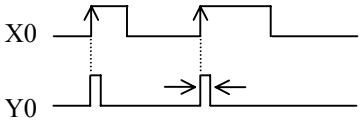
◎ PLS / PLF (Pulse Output)

EX	EX_{1S}	EX_{1N}	EX_{2N}

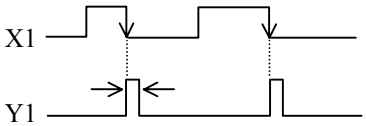
Mnemonic	Instruction	Symbol & Device	Step number
PLS	PuLSe	PLS Y.M.	2
PLF	PuLse Falling	PLF Y.M.	2




```
LD   X  0
PLS  Y  0
LD   X  1
PLF  Y  1
```



♦ When X0 pulse OFF→ON, the specified devices of PLS is enabled "1" scan cycle.



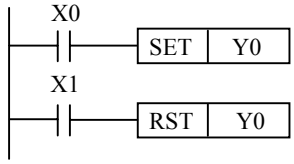
♦ When X1 pulse ON→OFF, the specified devices of PLF is enabled "1" scan cycle.

♦ The special auxiliary relay can't be for PLS/PLF used.

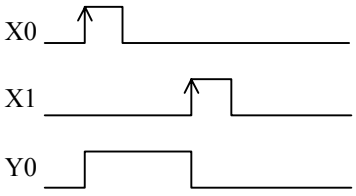
◎ SET/RST

EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
SET	SET	SET Y,M,S	Y.M. :1 Special M,S Coils :2
RST	ReSeT	RST Y,M,S,D	D, special D, registers, V and Z :3




```
LD   X  0
SET  Y  0
LD   X  1
RST  Y  0
```



♦ SET : While operation result is on the specified device is enabled. Once enabled, the specified device remains enabled even if the operation result is disabled.

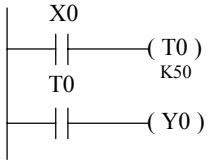
♦ RST : While operation result is on the specified device is reset, word device cleared to "0".

◎ TIMER & COUNTER

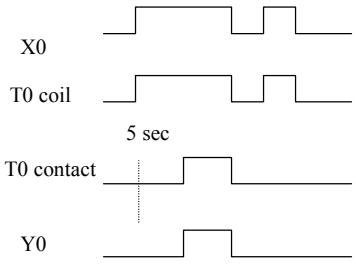
EX	EX_{1S}	EX_{1N}	EX_{2N}

Mnemonic	Instruction	Symbol & Device	Step number
OUT	OUT	(T.C) K..	32 bit counter : 5 Others :3
RST	RST	RST T,C	T.C : 2

<< Timer >>



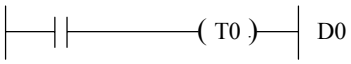

```
LD    X  0
OUT   T  0
      K  50
LD    T  0
OUT   Y  0
```



♦ When X0 on, T0 active after 5 seconds T0 contact ON, and keep current data even through X0 keep ON.

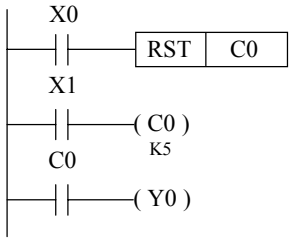
♦ When X0 off, then clear T0 to "0" and contact off,

♦ Timer can be set directly by using constant K or indirectly by using data register (D).

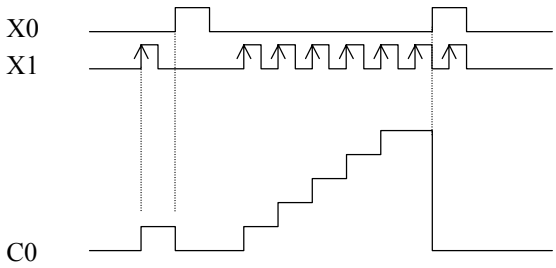


♦ All of the timers (T000~T255) are unlatched.

<< Counter >>



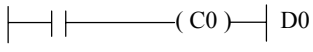

```
LD    X  0
RST   C  0
LD    X  1
OUT   C  0
      K  5
LD    C  0
OUT   Y  0
```



♦ When X0 ON, clear C0 current data to"0" and contact off.

♦ C0 count up the signal of X1 (OFF→ON), after 5 counts then keep current value and the contact ON.

♦ Counters can be set directly using constant K or indirectly by using data register (D).



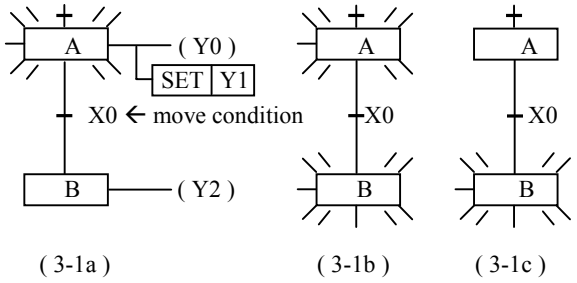
♦ All of the counters (C0 ~ C255) are latched.

♦ The high speed counters refer to chapter 4.

3. Step Ladder Instructions

◎ How STL Operates

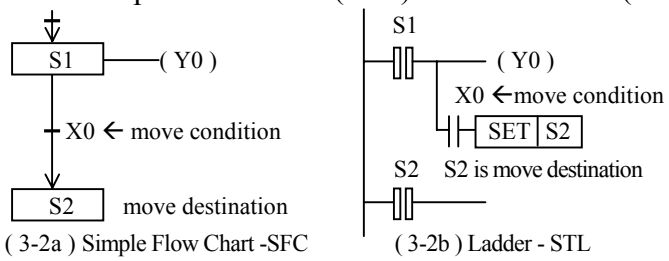
3.1 The state activate & move condition



(3-1a) (3-1b) (3-1c)

- ♦ When (3-1a) state A ON, execute the program belonged to state A, i.e. Y0 ON, Y1 ON, Y2 OFF. State B OFF, the program belonged to state B not executing.
- ♦ When move condition X0 ON (don't need to keep), then state B ON, i.e. state A ON and state B ON in one cycle time (3-1b) Y0,Y1,Y2 all ON.
- ♦ After one cycle state A auto OFF, state B still ON (3-1c) i.e. Y0 OFF, Y1 ON (SET), Y2 ON.
- ♦ Once the current STL state activates a second following state, the source STL state will auto reset.

3.2 Simple Flow Chart (SFC)& Ladder Chart (STL)



(3-2a) Simple Flow Chart -SFC (3-2b) Ladder - STL

- ♦ (3-2a) is Simple Flow Chart, (3-2b) is Ladder Chart.
- ♦ The state (S) can be connected to Output Relay directly.
- ♦ To Activate an STL state, need to drive the STL coil first.
  In the EX-series, the SET is used to drive an STL state to make it active.
- ♦ The formula is used M8002 & ZRST to initial STL state, and used M8002 & SET to start STL program.
- ♦ The RET instruction is end of STL state, let program return to ladder sequence.

3.3 STL＆RET Operands : S0 ~ S999

3.3.1 : Single Flow Mode

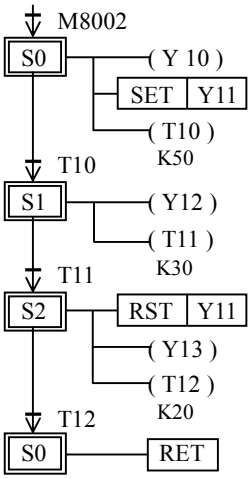


Fig 3-3-1a
Simple Flow Chart - SFC

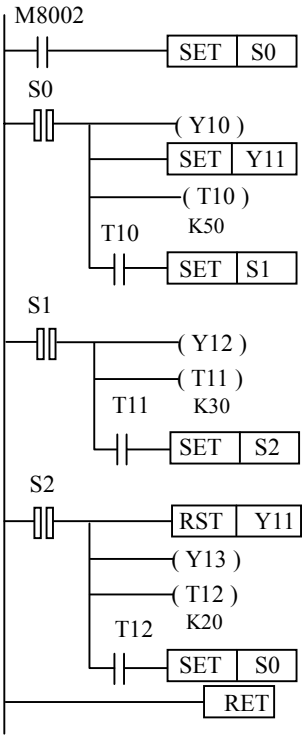


Fig 3-3-1b. Ladder - STL

LD	M	8002		K	30
SET	S	0	LD	T	11
STL	S	0	SET	S	2
OUT	Y	10	STL	S	2
SET	Y	11	RST	Y	11
OUT	T	10	OUT	Y	13
	K	50	OUT	T	12
LD	T	10		K	20
SET	S	1	LD	T	12
STL	S	1	SET	S	0
OUT	Y	12	RET		
OUT	T	11			

♦ The end of STL program area need added RET instruction, let program return to original bus bar.

3.3.2 : Selective Branch Programming

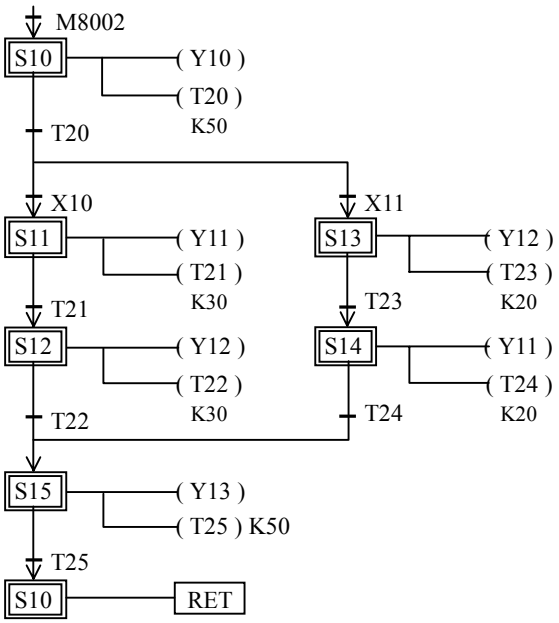


Fig 3-3-2a. Simple Flow Chart - SFC

♦ This type of program construction can created many flow but only one flow can enable, i.e. X10,X11 can't ON at the same time.

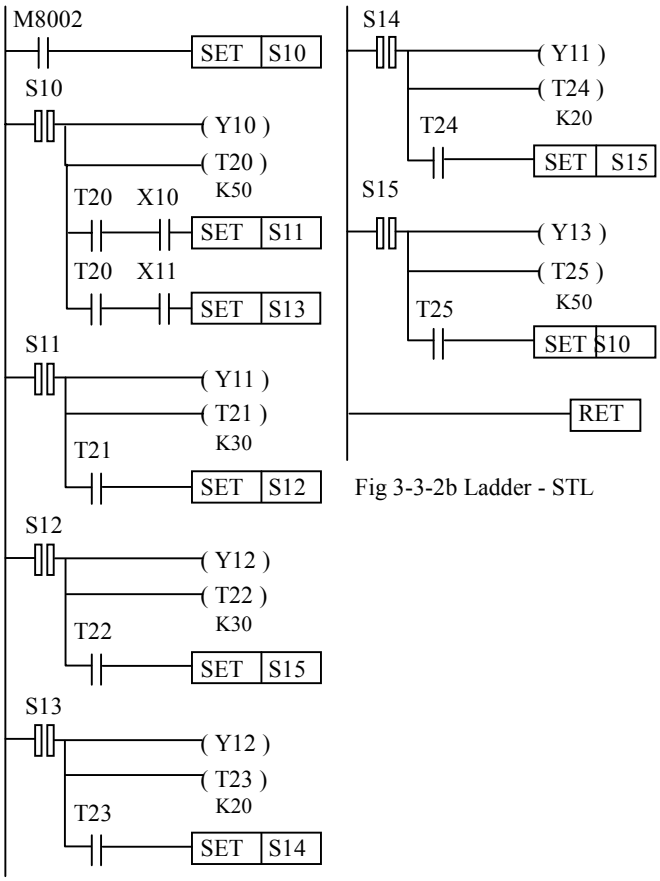


Fig 3-3-2b Ladder - STL

0000	LD	M	8002	0022	LD	T	22
0001	SET	S	10	0023	SET	S	15
0002	STL	S	10	0024	STL	S	13
0003	OUT	Y	10	0025	OUT	Y	12
0004	OUT	T	20	0026	OUT	T	23
0005		K	50	0027		K	20
0006	LD	T	20	0028	LD	T	23
0007	AND	X	10	0029	SET	S	14
0008	SET	S	11	0030	STL	S	14
0009	LD	T	20	0031	OUT	Y	11
0010	AND	X	11	0032	OUT	T	24
0011	SET	S	13	0033		K	20
0012	STL	S	11	0034	LD	T	24
0013	OUT	Y	11	0035	SET	S	15
0014	OUT	T	21	0036	STL	S	15
0015		K	30	0037	OUT	Y	13
0016	LD	T	21	0038	OUT	T	25
0017	SET	S	12	0039		K	50
0018	STL	S	12	0040	LD	T	25
0019	OUT	Y	12	0041	SET	S	10
0020	OUT	T	22	0042	RET		
0021		K	30	0043			

(3-3-2c)

3.3.3 : Multiple Flows Simultaneously

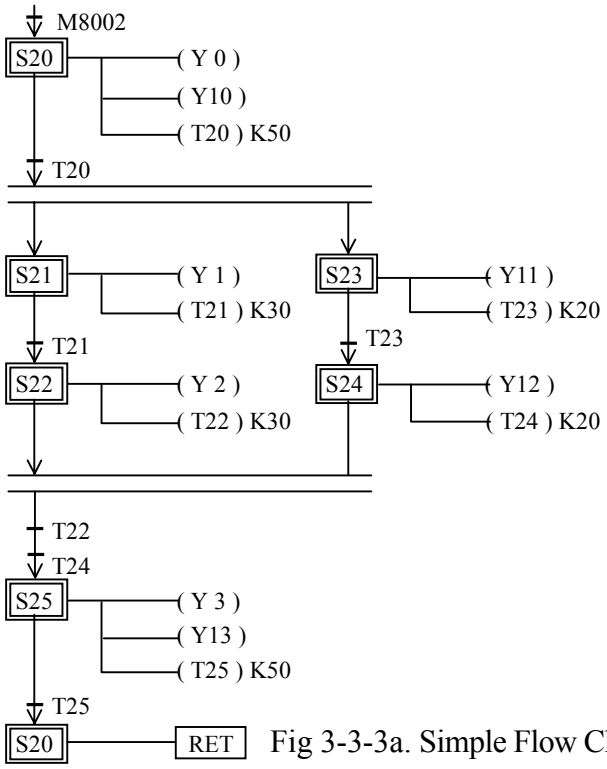


Fig 3-3-3a. Simple Flow Charting

♦ This type of program construction can enable multiple flow at the same time.

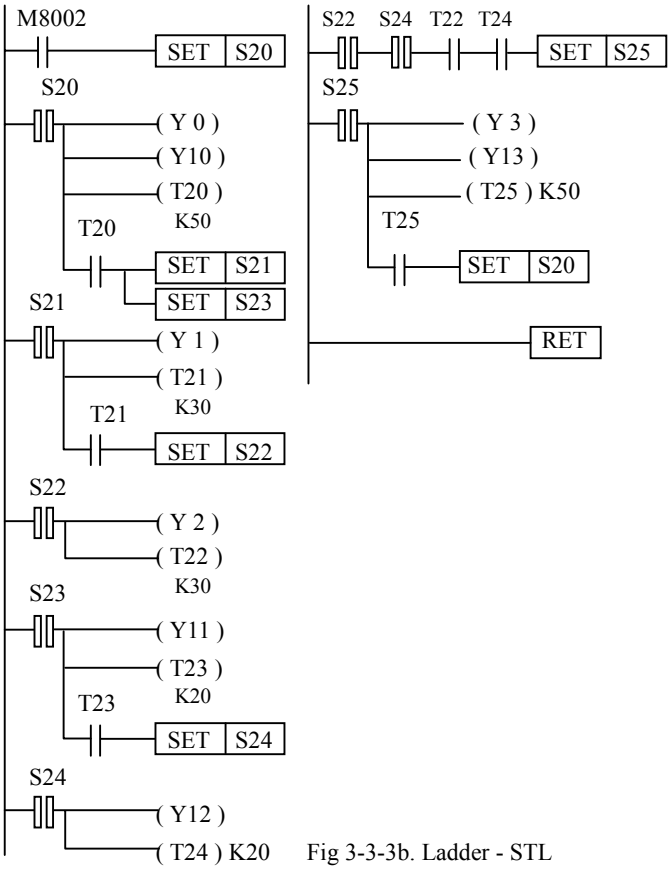


Fig 3-3-3b. Ladder - STL

0000	LD	M	8002	0022	OUT	T	23
0001	SET	S	20	0023		K	20
0002	STL	S	20	0024	LD	T	23
0003	OUT	Y	0	0025	SET	S	24
0004	OUT	Y	10	0026	STL	S	24
0005	OUT	T	20	0027	OUT	Y	12
0006		K	50	0028	OUT	T	24
0007	LD	T	20	0029		K	20
0008	SET	S	21	0030	STL	S	22
0009	SET	S	23	0031	STL	S	24
0010	STL	S	21	0032	LD	T	22
0011	OUT	Y	1	0033	AND	T	24
0012	OUT	T	21	0034	SET	S	25
0013		K	30	0035	STL	S	25
0014	LD	T	21	0036	OUT	Y	3
0015	SET	S	22	0037	OUT	Y	13
0016	STL	S	22	0038	OUT	T	25
0017	OUT	Y	2	0039		K	50
0018	OUT	T	22	0040	LD	T	25
0019		K	30	0041	SET	S	20
0020	STL	S	23	0042	RET		
0021	OUT	Y	11	0043			

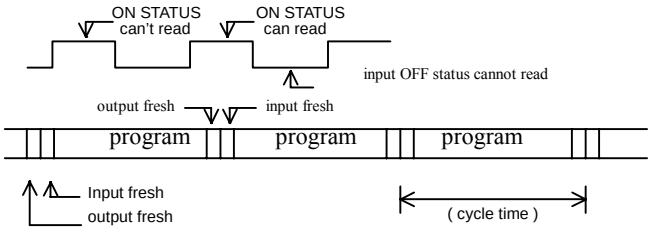
4. Advanced Devices

◎ Input Relays (X) & Output Relays (Y)

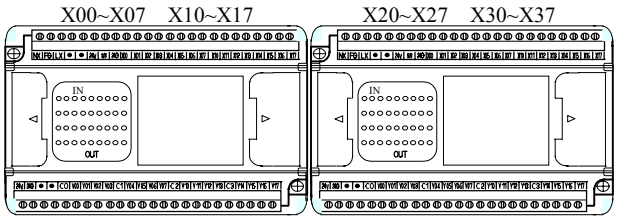
♦ Input Relay (X000 – X177) octal 128 points receive external switch signal, photo coupler isolation, and connect to input terminal directly.

♦ Output Relay (Y000 – Y177) octal 128 points output the signal to drive load, relay or photo coupler isolation and connect to output terminal directly.

➢ The timing of Input (X) & Output (Y)



➢ The number of Input (X) & Output (Y)



➢ Auxiliary Relay (M)

♦ General auxiliary relay (M000~M511) decimal 512 points

♦ Latch auxiliary relay (M512~ M1023) decimal 512 points

♦ Special auxiliary relay (M8000~M8255) 256 points

➢ State Relay (Mnemonic S)

♦ State relay (S000~S999) decimal 1000 points.

♦ All of state relays (S000~S999) are latched.

♦ This type relay is for SFC used.

➢ Pointer (Mnemonic P,I)

♦ The pointer (P00-P63) decimal 64 points

♦ The pointer (P) is for CJ, CALL branch used.

♦ The number of pointer can't use duplicate.

♦ The pointer (I) is for interrupt used.

➢ Constant (Mnemonic K/H)

♦ Decimal constant (K) range

16 bits: -32,768 ~ +32,767.

32 bits: -2,147,483,648 ~ +2,147,483,647

♦ Hexadecimal constant (H) data range

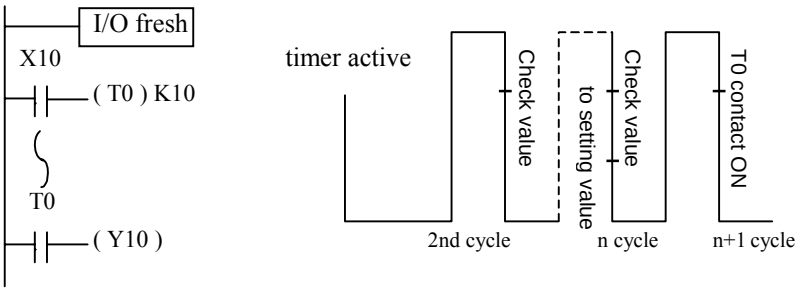
16 bits: 0000 ~ FFFF.

32 bits: 00000000 ~ FFFFFFFF

◎ Timer (Mnemonic T) T000 ~ T255

- ♦ All of timers belong to count up internal clock pulse (10ms,100ms). When count data reaches the setting value, the contacts activated.
- ♦ When the drive condition OFF, the current value reset to "0" and the contact OFF, except integration Timer.
- ♦ Setting value of timers can constant K or can use using data register (D) indirectly.
- ♦ 100ms Timer: T000 ~ T199 (200 points) setting range: 0.1 ~ 3,276.7 seconds
- ♦ 10ms Timer : T200 ~ T245 (46 points) setting range: 0.01 ~ 327.67 seconds
- ♦ 1ms integration Timer : T246 ~ T249 (4 points) setting range: 0.001 ~ 32.767 seconds
- ♦ 100ms integration Timer : T250 ~ T255 (6 points) setting range: 0.1 ~ 3276.7 seconds

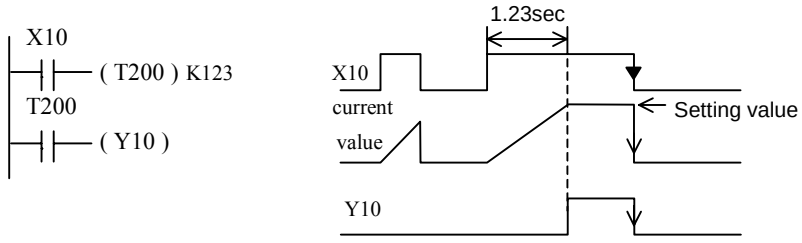
➢ Timer contact active condition and accuracy



♦ From above diagram, if the timer contact position put before timer coil, then the bad accuracy is "+2t".

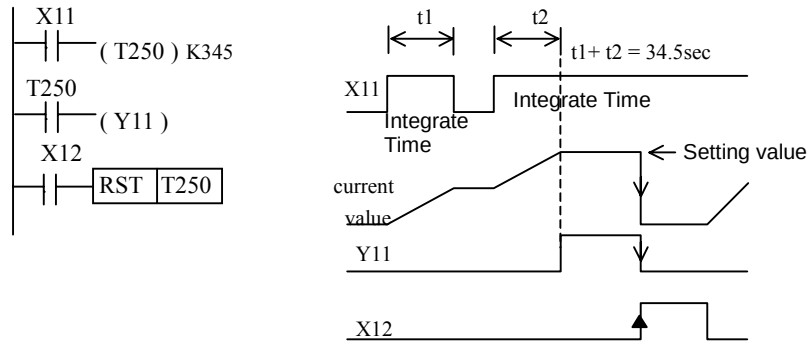
➢ The timing of Timer detailed action

♦ The timing of unlatched timer (General)



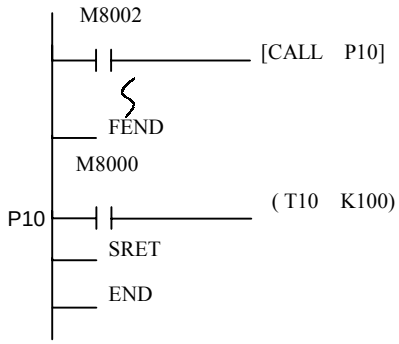
When input contact X10 ON, T200 start to count by 100ms counting method. When count value reaches to setting value, the contact act.
When count in the midway, input contact X10 OFF, the count current value will clear to "0".
When count reaches to input contact X10 OFF, the current value of count will clear to "0" and contact returns.

♦ The timing of latched timer (Integration)



When input contact X11 ON, T250 count by 100ms counting method. When counting value reaches to setting value, the contact act.
When count in the midway, input contact X11 OFF, the unchanged current value of timer (t1) input to contact and then ON, then from current value to count up until to setting value, and contact act.
Integration timer needs to use RST instruction to clear the content and the contact.

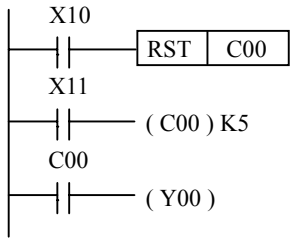
♦ Circular Timer



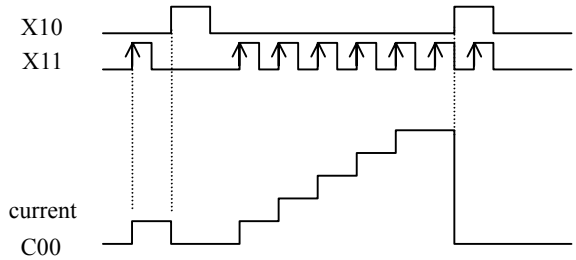
The program starts, enable T10 start to count, i.e. not count time to check if setting value reaches. At this time, timer will become to circular timer, circulating count by 0 → 32767 → 65535 → 0

◎ Counter (Mnemonic C) C000 ~ C255

➢ 16 bits up counter (C000~ C199, range: 1 ~ 32,767)



LD	X	10
RST	C	00
LD	X	11
OUT	C	00
	K	5
LD	C	00
OUT	Y	00



♦ The counter can be reset by RST at any time, value of counter is set to "0", and contact signal is OFF.

♦ When X10 ON, clear C00 current value to "0" and contact turned OFF.

♦ When counter count the pulse (OFF→ON) number of X11, and when current value to the setting value, then the contact turned ON, and keep the current value.

♦ Counters can be set directly by using constant K or indirectly by using data register (D).

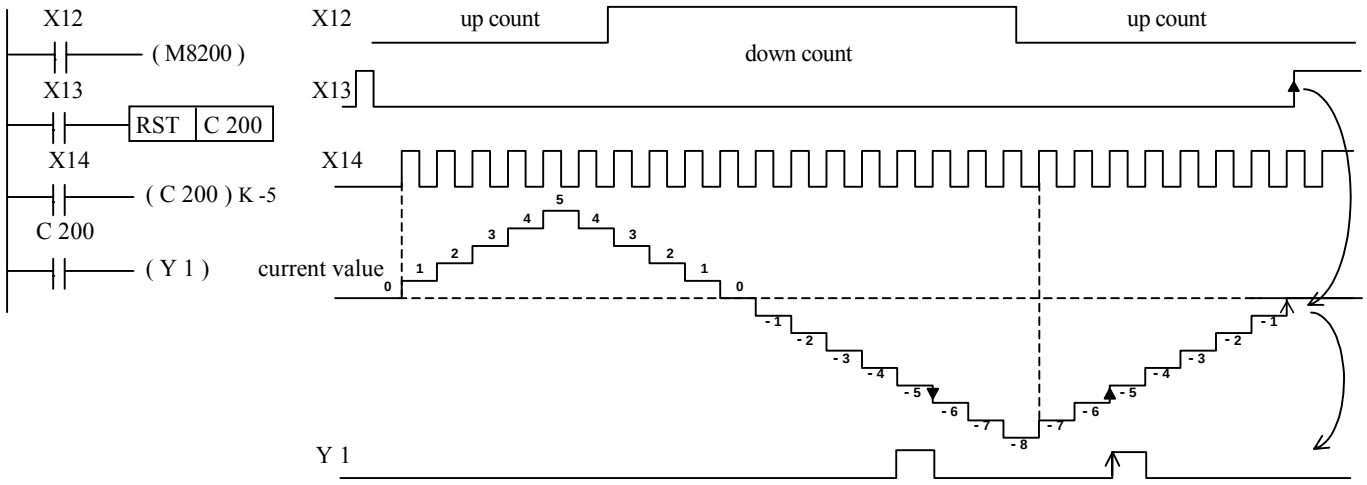
♦ Use data registers or special data registers to let content of data register become to setting value for the counter.

♦ The counter input signal ON or OFF must greater than program scan cycle time.

♦ If use other instruction to write a data which is greater than setting value to current value register, then when next count input is ON, counter output contact act and current value register will become to setting value.

♦ High Speed Counter operated by the principle of interrupt; this means they are event triggered and independent of cycle time.

➢ 32 bits up/down counter (C200~ C234) range: (-2,147,483,648 ~ +2,147,483,647)



♦ Through X14 count input to drive C200's coil one time, the current value of counter will be increment or decrement. When the current value from "-6" increase to "-5" or from "-4" decrease to "-5", the output contact set to ON. If from "-5" decrease to "-6" or from "-5" increase to "-4", then output contact OFF ;i.e., current value = setting value ON, others are OFF.

♦ If the current value is +2,147,483,647, when increment by "1" will change to -2,147,483,648. If the current value is -2,147,483,648, when decrement by "1" will change to +2,147,483,647. This counter we called it to circular counter.

♦ The counting direction assigned by special auxiliary relay M8200 - M8234. If M8xxx ON, then belong to down counter. If M8xxx OFF, then belong to up counter.

♦ If use other instruction to write a data which is greater than setting value to current value register, then when next count input, counter will still count as usual, but output contact will not be changed.

➢ High Speed Counter (C235~ C255) (High speed counter operated by interrupt and independent cycle time)

♦♦♦ 32 bits up/down latched counter ♦♦♦

ITEM	1 phase 1 direction	1 phase bi-direction	2 phase bi-direction
Counter number	C235 ~ C245	C246 ~ C250	C251 ~ C255
Counter direction	According to ON/OFF by M8235 ~M8245 to decided direction	According different input point to decided up count or down count	When A-phase ON, B-phase: 0→1: Up, 1→0:Down
Direction monitor	- - -	Monitor M8246 ~ M8255 status, then can know counter direction	

♦♦♦Difference of the 16 bit / 32 bit counters ♦♦♦

ITEM	16 bit counter	32 bit counter
Direction	Up counter	Up / down counter direction can be change
Value range	0 ~ 32,767	-2,147,483,648 ~ +2,147,483,647
Setting method	Constant (16 bit) or data register	Constant (32 bit) or a pair of register
Current value	No change to maximum value	Change to maximum value (ring counter)
Output contact	To maximum value set and keep status	Up counter: keep status, down counter: reset
reset	When RST instruction be driven, the value of counter reset to zero and output contact OFF	

➢ NOTE

♦ The input signal of high speed counter cannot be higher than counting speed.

♦ If an input is already being used by a high speed counter, it cannot be used for any other high speed counters or for any other purpose, like as an interrupt input.

♦♦♦Device Table of High Speed Counter ♦♦♦

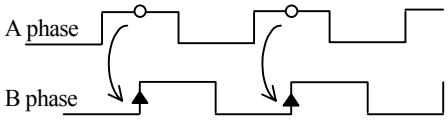
Input	1 Phase without start/reset						1 Phase with start/reset					1 Phase bi-direction					A-B Phase counter				
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X0	U/D						U/D			U/D		U	U		U		A	A		A	
X1		U/D					R			R		D	D		D		B	B		B	
X2			U/D					U/D			U/D		R		R			R		R	
X3				U/D				R			R			U		U			A		A
X4					U/D				U/D					D		D			B		B
X5						U/D			R					R		R			R		R
X6										S					S					S	
X7											S					S					S

U: up counter input, D: Down counter input, A: A-phase input, B: B-phase input, R: Reset input, S: Start input

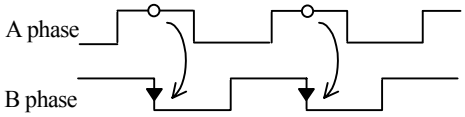
♦ Inputs X0 ~X7 cannot be used for more than one counter. For example:
If C235 is used the following counters (C241, C244, C246, C247, C249, C251, C252, C254, I0xx & SPD) cannot be used.

♦ X6 & X7 also as high speed input point, but only as START or RESET input used, cannot as counter input.

♦♦Following is 2 Phase Encoder Forward & Reverse Pulse Conduction, Need used AB Phase Counter ♦♦



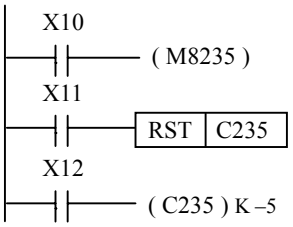
Forward Pulse, Count Up.



Reverse Pulse, Count Down.

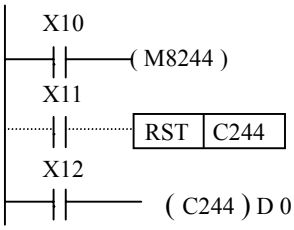
➢ 1 Phase High Speed Counter (High speed counter operated by interrupt and independent cycle time)

- 1 Phase 1 Input without start/reset C235~C240 (control M8xxx ON/OFF status, then can set Cxxx counting direction)



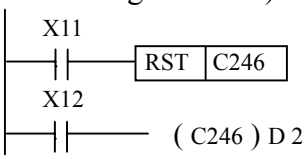
♦This example used X10 control C235 counting direction, X00 is counting input.
♦When X12 ON, C235 start counting the input pulse (OFF→ON) of X00.
♦When X11 ON, reset C235 current value to ”0”, and the contact turned OFF.

- 1 Phase 1 Input with start/reset C241~C245 (control M8xxx ON/OFF status, then can set Cxxx counting direction)



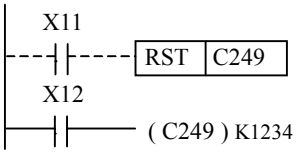
♦X10 control counting direction, X00 counting input, X01 reset input, X06 start input.
♦When X12 ON and X06 ON, C244 start counting the input pulse (OFF→ON) of X00.
♦When X11 or X01 ON, reset C244 current value to ”0”, and the contact turned OFF.
♦This line instruction can ignored, because can used X01 reset C244.
♦This example the content of (D1, D0) is the setting value.

- 1 Phase 2 Input without start/reset C246 (monitor M8xxx ON/OFF status, then can know Cxxx counting direction)



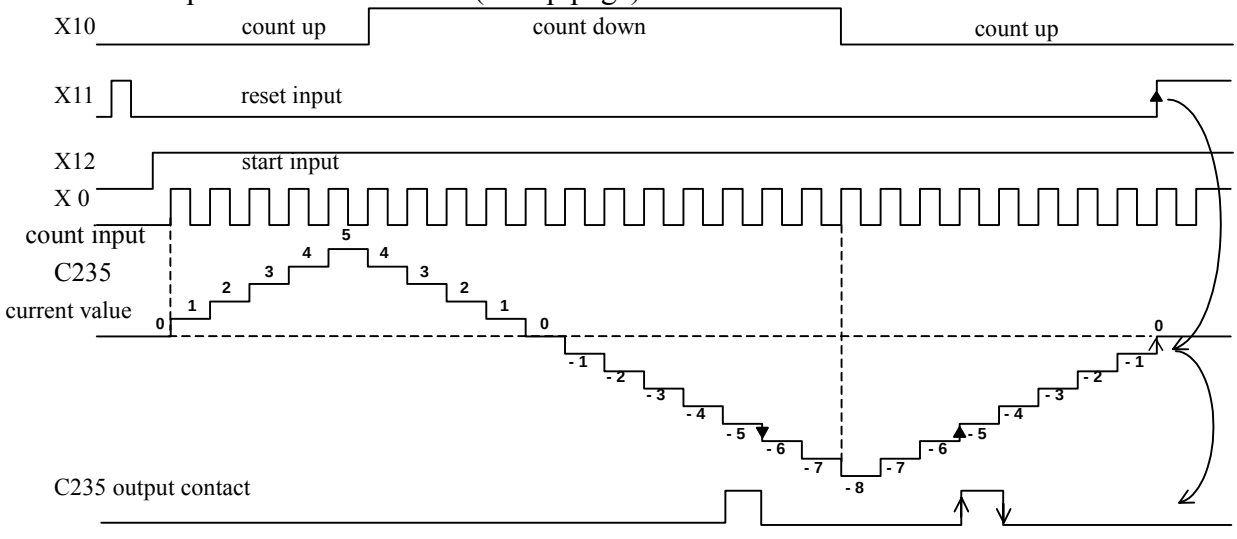
♦X00 as count up input point, X01as count down input point.
♦When X12 ON, C246 start counting the input pulse (OFF→ON) of X00 and X01.
♦When X11 ON, reset C246 current value to ”0”, and the contact turned OFF.

- 1 Phase 2 Input with start/reset C247~C250 (monitor M8xxx ON/OFF status, then can know Cxxx counting direction)



♦X00 count up input point, X01 count down input point, X02 reset input,X06 start input.
♦When X12 and X06 ON, C249 counting the input pulse (OFF→ON) of X00 and X01.
♦When X11 or X02 ON, reset C249 current value to ”0”, and the contact turned OFF.

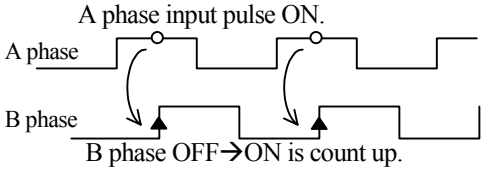
- 1 Phase 1 Input without start/reset (ex. up page)

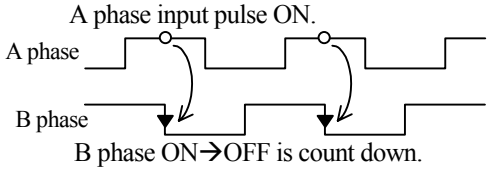


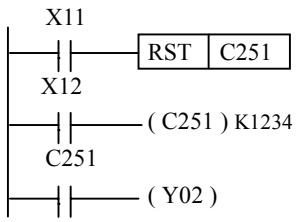
- ♦ C235 use X000 as the interrupt pulse input point. (X12 is not pulse input point. Just as enable signal)
- ♦ Through X00 count input to drive C235 one time, current value of counter will be increment or decrement. When current value from ”-6” increase to “-5” or from “-4” decrease to “-5”, output contact is ON. If from ”-5” decrease to “-6” or from “-5” increase to “-4”, output contact is OFF ;i.e., current value = setting value ON, other OFF.
- ♦ If the current value is +2,147,483,647, when increment by “1” will change to -2,147,483,648. If the current value is -2,147,483,648, when decrement by “1” will change to +2,147,483,647. This counter we called it to circular counter.
- ♦ The counting direction assigned by special auxiliary relay M8235 - M8240.(i.e. X10 control counting direction)
- ♦ When the reset input to X11 is ON, C235 current value reset to ”0”, and contact turned OFF.

◎ 2 (A-B) Phase 2 Input High Speed Counter (This counter operated by interrupt and independent cycle time)

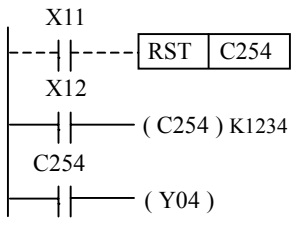
- ♦ EX-serial can use 2 point 2 phase 32 bits up/down counter at the same time.
- ♦ Monitor M8xxx ON/OFF status, then can know the counting direction of Cxxx.
- ♦ The counting direction of this type counter are decided by A phase and B phase, when the input pulse is ON of A phase, and when B phase input pulse is OFF→ON, then is up counter, when B phase ON→OFF then is down counter.







- ♦ This example use X00 as A phase input, X01 as B phase input, without start/rest.
- ♦ When X11 ON, reset C251 current value to ”0”, and the contact turned OFF.
- ♦ When X12 ON, enable the C251 start counting.
- ♦ Whether count up or count down, when current value = setting value, Y02 ON, other OFF.



- ♦ X00 as A phase input, X01 as B phase input, X02 as reset input, X06 as start input.
- ♦ When X11 or X02 ON, reset C254 current value to ”0”, and the contact turned OFF.
- ♦ When X12 ON, enable the C254 start counting.
- ♦ Whether count up or count down, when current value = setting value, Y04 ON, other OFF

◎ Data Register (D)

- Latched Data Register (D000 – D255) 256 Points
- General Data Register (D256 – D3999) ： General data register can be used as same as file register

♦ All of data register are 16 bits (the msb is sign), also can pair of any one to 32 bits data.

- Special Data Register (D8000 – D8255) 256 Points

♦ The special data register is used to control or monitor the programmable logic controller internal status.

♦ When the power OFF→ON, all of the data register are set to initial value.

◎ Index Register (V, Z)

♦ 16 bits operation mode V & Z all is 16 bits register. 32 bits operation mode pair of (V, Z) as 32 bits register V is upper word, Z is lower word.

♦ Sometime the use of multiple index registers V & Z is necessary in larger program or the program need handle large quantities of data.

For Example : MOV D0Z,D100

Just change index Z value, then can move the content of D00~D99 to D100.

♦ Following instruction format can modified by index V,Z

KnXxxZ, KnYxxZ, KnMxxZ, KnSxxZ, TxxZ, CxxZ, DxxZ

♦ Following is error instruction format

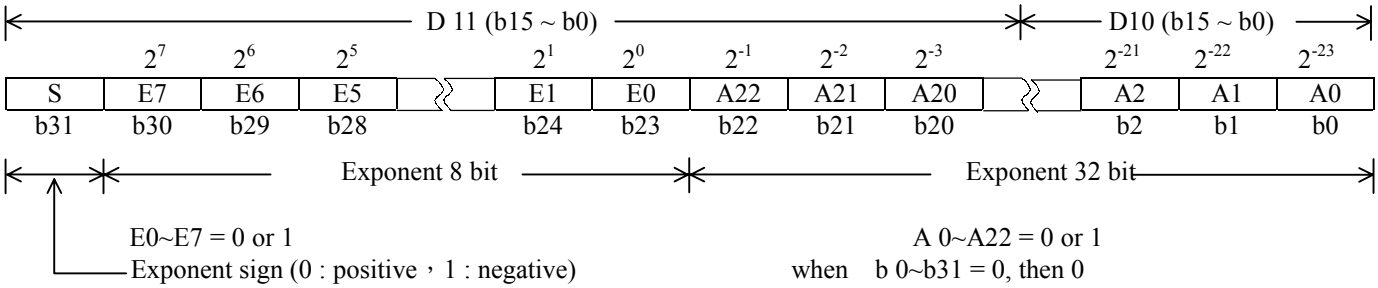
KnZMxx (index register V,Z can't connect to Kn directly)

♦ Example

```
MOV   K10, Z                ; index Z=10
ADD   D0, D2, D100Z         ; D0+D2 → D110
```

◎ Binary Floating Data

Binary floating data is a data register which use an continuous serial number, for example (D11, D10).



Binary Floating data = $\pm (2^0+A22\times2^{-1}+A21\times2^{-2}+\ldots\ldots+A0\times2^{-23}) \times 2^{(E7\times2^7+E6\times2^6+\ldots\ldots+E0\times2^0)} /2^{127}$

(Example) A22=1 ， A21=0 ， A20=1 ， A19~A0=0

E7=1 ， E6~E1=0 ， E0=1

Binary floating data = $\pm (2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+\ldots\ldots+0\times2^{-23}) \times 2^{(1\times2^7+0\times2^6+\ldots\ldots+1\times2^0)} /2^{127}$

$= \pm1.625\times2^{129}/2^{127}=\pm1.625\times2^2$

positive and negative sign is decided by b31, can't use negation.

♦ The using of zero flag(M8020), negative flag(M8021), carry flag(M8022), the flag action of floating operation as follows,

Zero flag : when result is 0, then it is 1.

Negative flag : when result not reach minimum unit, when it is not 0, then it is 1.

Carry flag : when result more than absolute value using range, then it is 1.

5. Applied Instructions

Applied instructions allow the user to perform complex data manipulations, mathematical operations. Each applied instruction has unique mnemonics and special function numbers. Each applied instruction will be expressed using a table similar to that show below. And will be found at the beginning of the description of each new instruction.

◎ COMPARE

FNC(10)			16 bits:CMP & CMP(P) - - - - - - - - - - - - - - - - - - - 7 Steps
D	CMP	P	32 bits:(D)CMP&(D)CMP(P) - - - - - - - - - - - - -13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: [D.]

X	Y	M	S

No modification of the instruction mnemonic is required for 16 bit operation, and it will operate continuously, i.e. on every scan cycle of the user program, the instruction will operation and provide a new result.

However, pulse operation requires a 'P' to be added directly after the mnemonic, while 32 bit operation requires a "D" to be added before the mnemonic. This means that if an instruction was being used with both pulse and 32 bit applied operation it would look like D***P, where *** was the basic mnemonic.

The 'pulse' function allows the associated instruction to be Activated on the rising edge of the control input. The Instruction is driven ON for the duration of one program Scan cycle. Thereafter, even if the control input remains on the associated instruction will not be active.

Following is Symbols list:

[D.]: Destination device

[S.]: Source device

[m,n]: Number of active devices, bits or an operational constant.

Following is instruction modifications:

*** - An instruction operation in 16 bit mode, where *** identifies the instruction mnemonic.

*** P- An instruction modified to use 16 bits pulse operation.

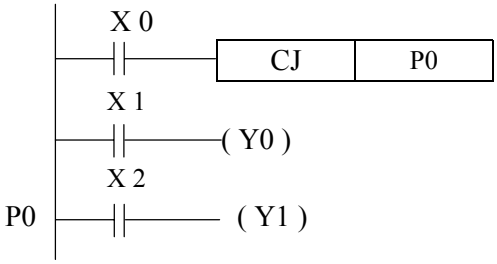
D*** - An instruction modified to use 32 bits operation.

◎ Condition Jump

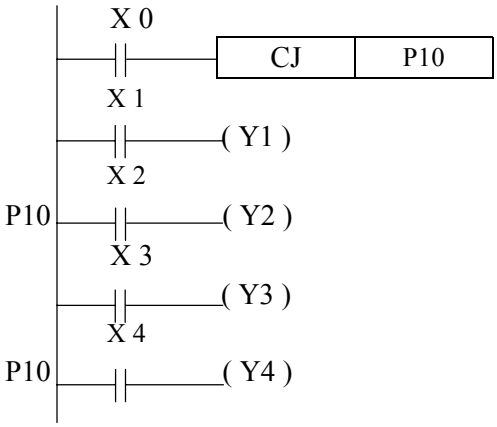
FNC(00)			16 bits: CJ & CJ(P) - 3 Steps
	CJ	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

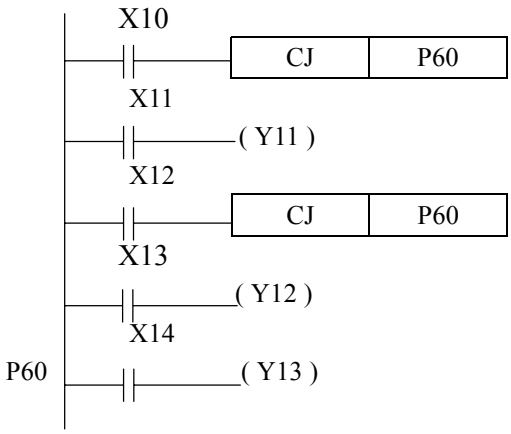
Operand: P00 ~ P63



Example (A)



Example (B)



Example (C)

- ♦ Example (A): If X0 ON forces the program to jump to LAB P0, any program area which is skipped will not update. Output statuses will not change even input the devices.
- ♦ Example (A): If miss LAB P0 pointer, then X0 ON will jump directly to END.
- ♦ If a backwards jump is used, then need to care the watchdog timer overrun.
- ♦ If LAB pointer is duplicated to use, only the last pointer is effective.
- ♦ Example (B): X0 ON forces the program to jump to the second LAB pointer.
- ♦ Example (C): Many CJ statements can be assigned to jump to the same pointer.

◎ Subroutine Call

FNC(01)			16 bits: CALL & CALL(P) - - - - - - - - - - - - - - - - - - 3 Steps
	CALL	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

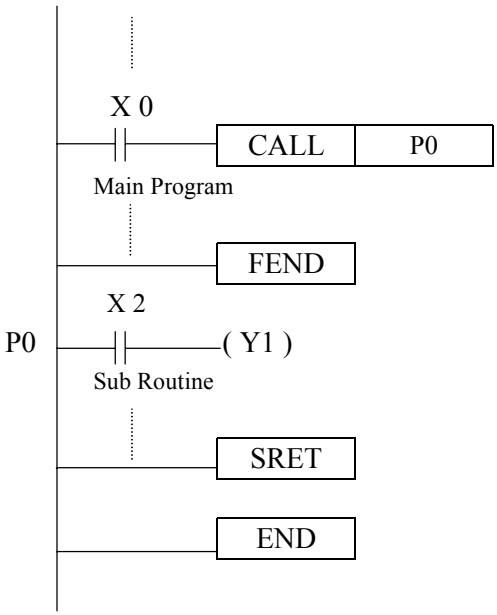
Operand: P00~P63

◎ Subroutine Return

FNC(02)			16 bits: SRET - 1 Steps
	SRET		

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operand: None



- When X0 ON, program will jump to subroutine pointer LAB P0 and execute Subroutine until SRET instruction is executed, then program return to original step and continue processing.
- The LAB assigns beginning of subroutine must be programmed after an FEND.
- The same LAB can only be used once, but many CALL statements can be assigned to a single LAB subroutine.
- Subroutines can be nested for 5 levels including one CALL instruction.

◎ Interrupt Return

FNC(03)	16 bits: IRET - 1 Steps
IRET	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operand: None

◎ Enable Interrupt

FNC(04)	16 bits: EI - 1 Steps
EI	

EX	EX_{1S}	EX_{1N}	EX_{2N}

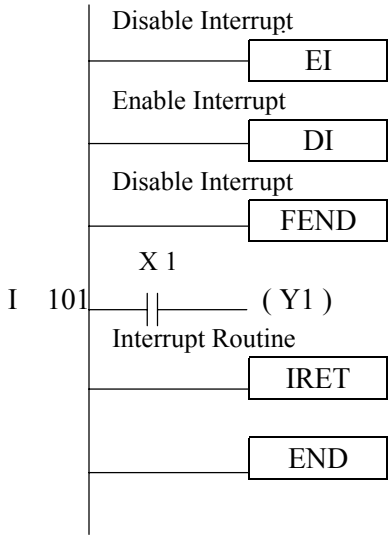
Operand: None

◎ Disable Interrupt

FNC(05)	16 bits: DI - 1 Steps
DI	

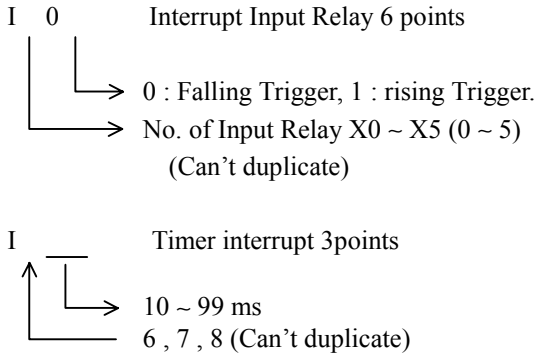
EX	EX_{1S}	EX_{1N}	EX_{2N}

Operand: None



- The programmable controller has a default status of disabling interrupt operation.
- As under normal operation, when an interrupt routine is when an interrupt routine is activate all other interrupt are disabled.
- Interrupt routine are always programmed after a FEND instruction.
- The IRET instruction may only be used within interrupt routines.

Number of Interrupt pointer



<< Note >>

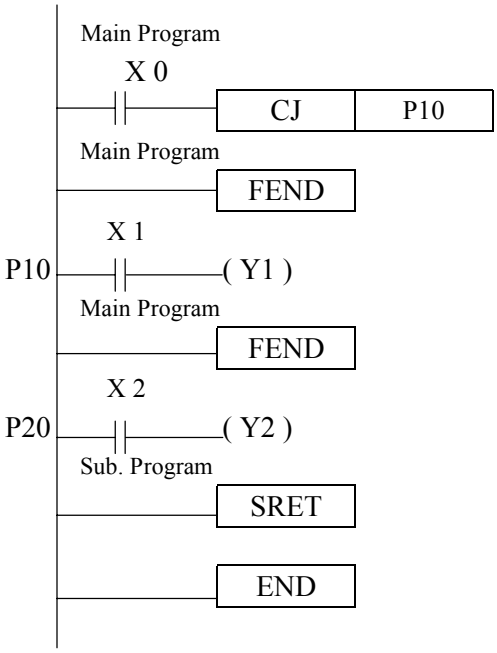
- When an interrupt program execute, other Interrupt Call is ineffective.
- If Interrupt occur within the range of Disable Interrupt (DI~EI), this interrupt request signal is stored temporarily, and execute until within the range of Enable Interrupt (EI~DI).
- When Disable Interrupt flag M805Δ act, the corresponding Interrupt input will not be executed.

◎ First End

FNC(06)	16 bits: FEND - 1 Steps
FEND	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operand: None



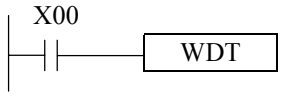
- A FEND instruction indicates the first end of a main program and the start of the subroutine program area to be used.
- Multiple FEND instruction can be use to separate different subroutines.
- When FEND is executed, the program return to Step 0.
- FEND can't be used after an END instruction.

◎ Watch Dog Timer

FNC(07)	16 bits: WDT - 1 Steps
WDT P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operand: None



- This instruction will compare the cycle time with the content of special data register D8000.
- If the watch dog timer > the content of D8000, then error occurred and error code is 6309.
- Can use MOV instruction to change content of special data register D8000.
- If do not write WDT instruction in program, then the watch dog timer is ineffective.

◎ FOR

FNC(08)	16 bits: FOR - 7 Steps
FOR	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S.]

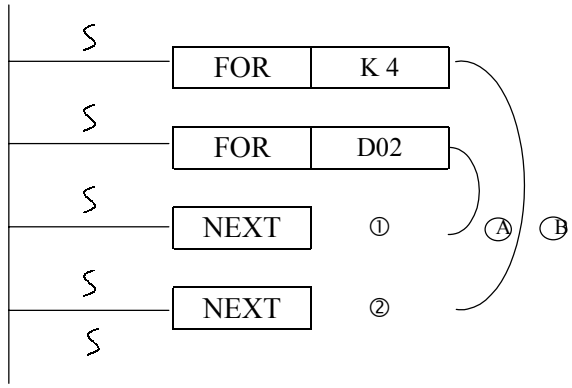
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

◎ NEXT

FNC(09)	16 bits: NEXT - 7 Steps
NEXT	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operand: None



- After program B execute 4 times, then execute the program below ② NEXT.
- If the content of D0Z is 5, then program B is executed 4 times, and program A will be executed 20 times.
- The maximum nest level of FOR –NEXT is 5 levels.

◎ Compare

FNC(10)			16 bits: CMP & CMP(P) - - - - - - - - - - - - - - - - - - - 7 Steps
D	CMP	P	32 bits: (D)CMP&(D)CMP(P) - - - - - - - - - - - -13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

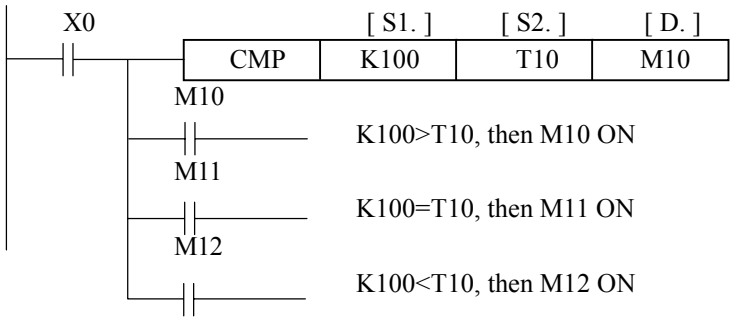
Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: [D.]

X	Y	M	S

Flag: M8020, M8021, M8022



♦ Data of [S1.] is compared with data of [S2.] and [D.] will be changed according to the result. This will automatic occupy 3 bit destination devices from head address of designation M10 ~ M12.

♦ Full algebraic comparisons are used, i.e. -10 smaller than +2.

♦ When X0 OFF, then [D.] bit devices status will not be changed.

◎ Zone Compare

FNC(11)			16 bits: ZCP & ZCP(P) - - - - - - - - - - - - - - - - - - - 9 Steps
D	ZCP	P	32 bits: (D)ZCP&(D)ZCP(P) - - - - - - - - - - - - -17 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

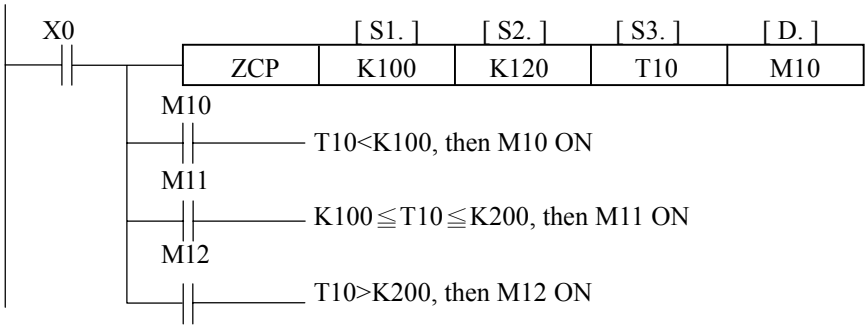
Operands: [S1.] [S2.][S3.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: [D.]

X	Y	M	S

Flag: M8020, M8021, M8022



♦ Content of [S3.] is compared with data range of [S1.] and [S2.] and [D.] will be changed according to the result. This will automatic occupy 3 bit destination devices from head address of designation M10 ~ M12.

♦ Full algebraic comparisons are used, i.e. -10 smaller than +2.

♦ When X0 OFF, then [D.] bit devices status will not be changed.

◎ Move

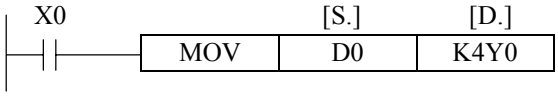
FNC(12)			16 bits: MOV & MOV(P) - 5 Steps
D	MOV	P	32 bits: (D)MOV&(D)MOV(P) - - - - - - - - - - - - -9 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

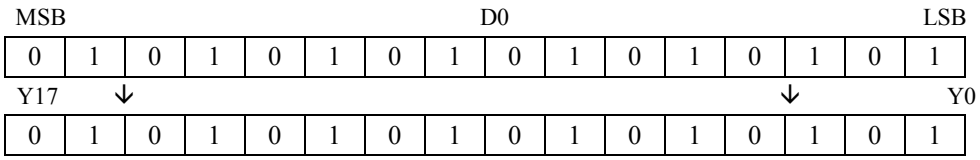
Operands: [S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

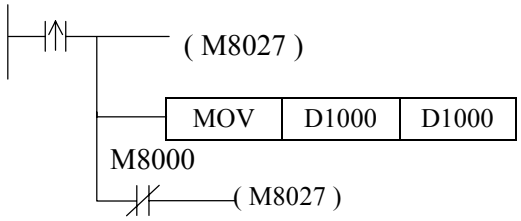
[D.]



♦ When X0 ON, contents of source device [S.] copied to destination device [D.].



♦ When M8027 ON, CPU will write content of [S.] into EEPROM, [D.] only D register can be used.



Note: When M8027 ON, for avoid to damage EEPROM, must be used Pulse Instruction MOV(P).

◎ Shift Move

FNC(13)			16 bits: SMOV & SMOV(P) - - - - - - - - - - - - - - - - - 7 Steps
	SMOV	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]

♦ Reserved

◎ Complement

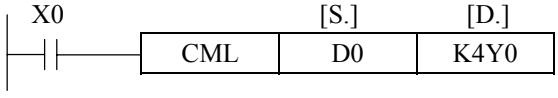
FNC(14)			16 bits: CML & CML(P) - 5 Steps
D	CML	P	32 bits: (D)CML & (D)CML(P) - - - - - - - - -- - - -9 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

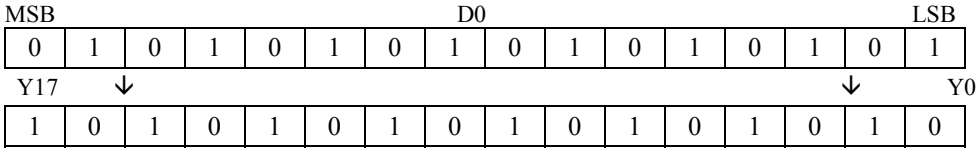
Operands: [S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]



♦ Each data bit within the source device [S.] is inverted and then copied to the designated destination [D.].



◎ Block Move

FNC(15)			16 bits: BMOV & BMOV(P) - - - - - - - - - - - - - - - - - 7 Steps
	BMOV	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

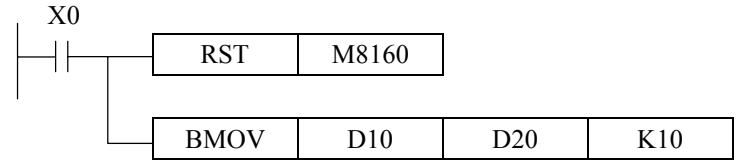
[S.]

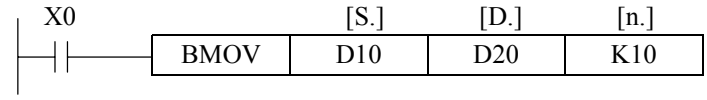
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

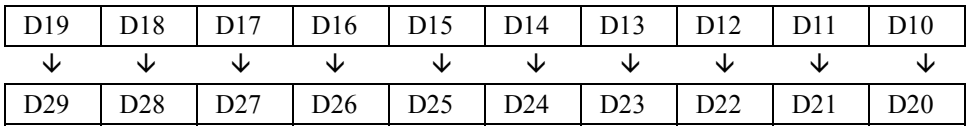
n

[D.]

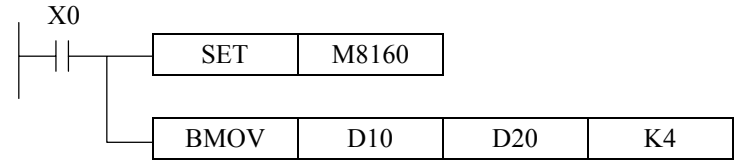
Flag:

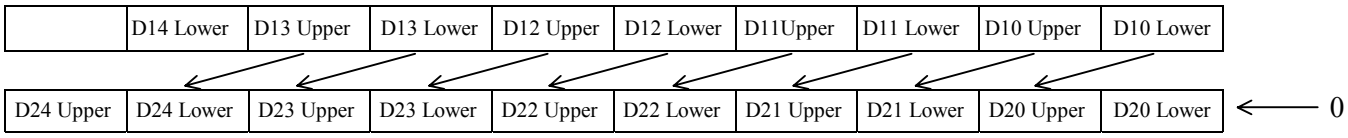




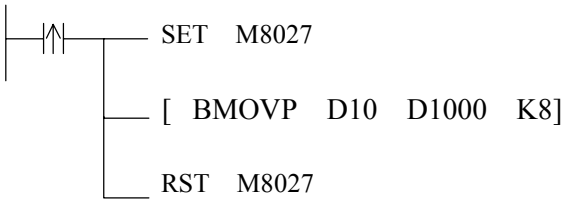


Ex1s, Ex1n V1.16 Edition additional function: [S.] and [D.] only D register is effective.





Ex1s, Ex1n V1.40 Edition additional function: [S.] and [D.] only D register is effective.
When M8027 ON, CPU will write the content of [S.] into EEPROM, [D.] only D register can be used.



Note: When M8027 ON, for avoid to damage EEPROM, must be used Pulse Instruction MOV(P).

◎ Fill Move

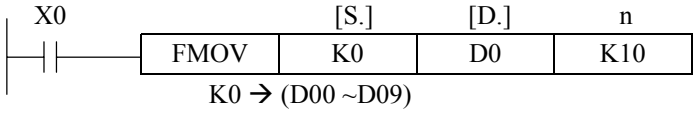
FNC(16)			16 bits: FMOV & FMOV(P) - - - - - - - - - - - - - - - - - 7 Steps
	FMOV	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

n → [D.] →



K0 → (D00 ~D09)

◎ Exchange

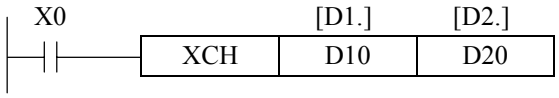
FNC(17)			16 bits: XCH & XCH(P) - - - - - - - - - - - - - - - - - - - 5 Steps
D	XCH	P	32 bits: (D)XCH & (D)XCH(P) - - - - - - - - - - - -9 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [D1.]

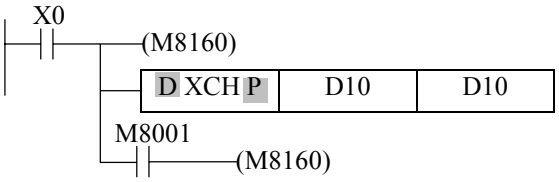
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D2.] →

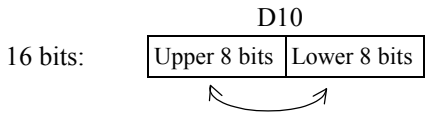


Before : (D10)=100 After : (D10)=200
(D20)=200 (D20)=100

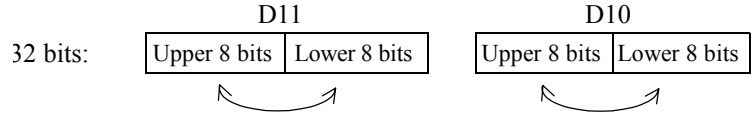
<< Function of Expanded >> SWAP



- If M8160 ON, [D1.] and [D2.] are the same word device, then the upper 8 bits and the lower 8bits will exchange.
- If [D1.] and [D2.] are not the same device, error flag M8067 ON, error code 6705. Error step number is stored to D8069 and not be executed.



Before executing (D10)=0050H=80，After executing (D10)=5000H=20480



Before executing (D11,D10)=87654321H=80，After executing 65872143H

◎ BCD (BINARY CODE TO DECIMAL)

FNC(18)			16 bits: BCD & BCD(P) - 5 Steps
D	BCD	P	32 bits: (D)BCD & (D)BCD(P) - - - - - - - - - - - - -9 Steps

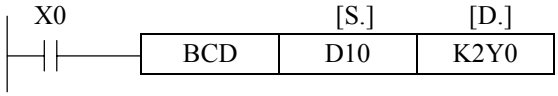
EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]



- The binary source data [S.] is converted into an equivalent BCD number and stored to the destination device [D.].
- If the converted BCD number exceeds the operational ranges of 0 to 9999 (16 bit operation) or 0 to 99999999 (32 bit operation), an error will occur. Error flag M8067 ON, error code 6705 and error step number stored to D8069. Program will be executed continuously, but result will not be stored to [D.]
- This instruction can be used to output data to a seven segment display directly.

◎ BIN (DECIMAL CODE TO BINARY)

FNC(19)			16 bits: BIN & BIN(P) - 5 Steps
D	BIN	P	32 bits: (D)BIN & (D)BIN(P) - - - - - - - - - - - - - -9 Steps

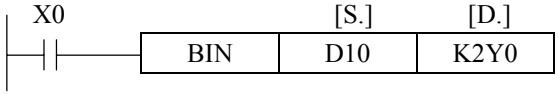
EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]



- The BCD source data [S.] is converted into an equivalent binary number and stored at the destination device [D.].
- If the source data is not provided in a BCD format, an error will occur. Error flag M8067 ON, error code 6705 and error step number stored to D8069.
- The device [S.] can't be used constant K/H.

◎ Addition

FNC(20)			16 bits: ADD & ADD(P) - - - - - - - - - - - - - - - - - 7 Steps
D	ADD	P	32 bits: (D)ADD &(D)ADD(P) - - - - - - - - - - - - - 13 Steps

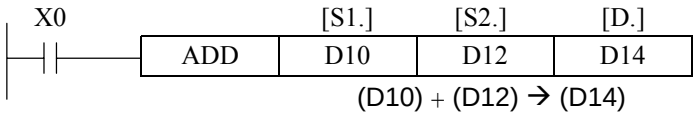
EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]

Flag: M8020, M8021, M8022



(D10) + (D12) → (D14)

- The data contained within the source devices [S1.], [S2.] is added and the result stored to specified destination devices [D.].
- All calculations are algebraically processed, i.e. 5+(-8) = -3.
- If the result of a calculation is "0", then zero flag M8020 ON.
- If the result exceeds 32,767 (16 bit limit) or 3,147,483,647 (32 bit operation), the carry flag M8022 ON.
- If the result exceeds -32,767 (16 bit limit) or -2,147,483,647 (32 bit limit), the borrow flag M8021 ON.

◎ Subtraction

FNC(21)			16 bits: SUB & SUB(P) - - - - - - - - - - - - - - - - - 7 Steps
D	SUB	P	32 bits: (D)SUB &(D)SUB(P) - - - - - - - - - - - - - 13 Steps

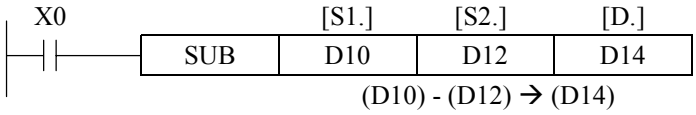
EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

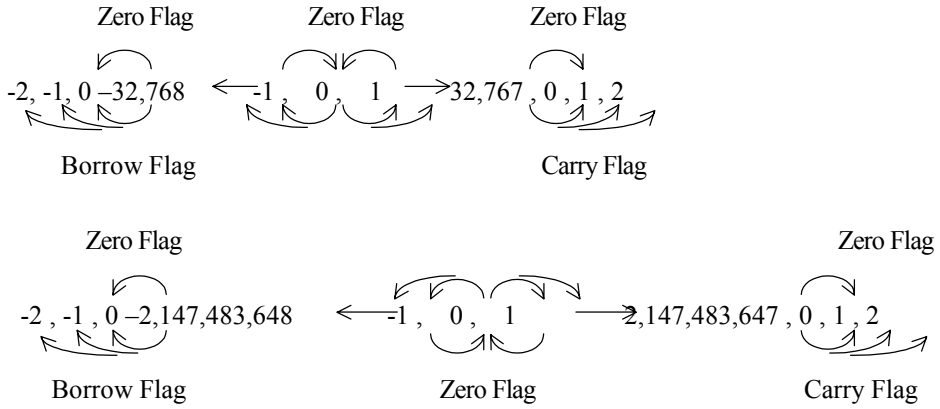
[D.]

Flag: M8020, M8021, M8022



(D10) - (D12) → (D14)

- Content of [S1.] subtract content of [S2.], and the result stored to specified destination devices [D.].
- All calculations are algebraically processed, i.e. 5 - 8 = -3.
- The MSB of devices is sign (0:Positive, 1:Negative).



◎ Multiplication

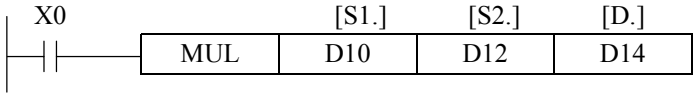
FNC(22)			16 bits: MUL & MUL(P) - - - - - - - - - - - - - - - - - 7 Steps
D	MUL	P	32 bits: (D)MUL &(D)MUL(P) - - - - - - - - - - - - 13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

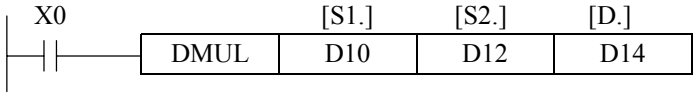
Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]



16 bit: (D10) × (D12) → (D15, D14)



32 bit: (D11,D10) × (D13,D12) → (D17,D16,D15,D14)

♦ The primary source [S1.] is multiplied by the secondary source [S2.]. The result is stored to destination [D.].

◎ Division

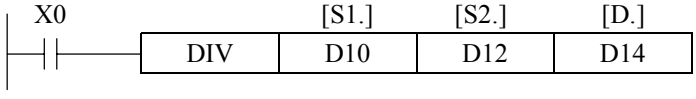
FNC(23)			16 bits: DIV & DIV(P) - - - - - - - - - - - - - - - - - -- 7 Steps
D	DIV	P	32 bits: (D)DIV & (D)DIV(P) - - - - - - - - -- - - - -13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

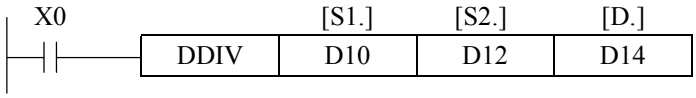
[D.]



Dividend divisor quotient remainder

(D10) ÷ (D12) → (D14) . . . . . (D15)

16 bits 16 bits 16 bits 16 bits



Dividend divisor quotient remainder

(D11,D10) ÷ (D13,D12) → (D15,D14).…….(D17,D16)

32 bits 32 bits 32 bits 32 bits

♦ The primary source [S1.] is divided by the secondary source [S2.]. The result is stored to destination [D.].

♦ If value of source device [S2.] is “0” (zero), then an operation error is executed. Error code 6706 and error step number stored to D8069, the program operation is cancelled.

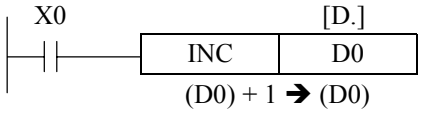
♦ V1.17 edition：If value of source device [S2.] is “0” (zero), then will not execute and directly jump to next instruction.

◎ Increment

FNC(24)			16 bits: INC & INC(P) - - - - - - - - - - - - - - - - - - 3 Steps
D	INC	P	32 bits: (D)INC & (D)INC(P) - - - - - - - - - - - - - -5 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

	K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
Operands:	←					[D.]			→



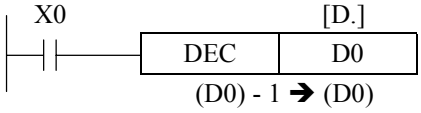
- ♦ On every execution of the instruction, the device specified as the destination [D.] and its current value increased 1.
- ♦ In 16 bit operation, when +32,767 is reached, the next execution will write a value of -32,768 to destination device.
- ♦ In 32 bit operation, when +2,147,483,647 is reached, the next execution will write -2,147,483,648 to destination device.
- ♦ The carry, zero and borrow flag are unaffected in the operation.

◎ Decrement

FNC(25)			16 bits: DEC & DEC(P) - - - - - - - - - - - - - - - - - - 3 Steps
D	DEC	P	32 bits: (D)DEC & (D)DEC(P) - - - - - - - - -- - - - -5 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

	K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
Operands:	←					[D.]			→



- ♦ On every execution of the instruction, the device specified as the destination [D.] and its current value decreased 1.
- ♦ In 16 bit operation, when -32,768 is reached, the next execution will write a value of +32,767 to destination device.
- ♦ In 32 bit operation, when -2,147,483,648 is reached, the next execution will write +2,147,483,647 to destination device.
- ♦ The carry, zero and borrow flag are unaffected in the operation.

◎ Logical AND

FNC(26)			16 bits: WAND & WAND(P) - - - - - - - - - - - - - - 7 Steps
D	WAND	P	32 bits: (D)WAND &(D)WAND(P) - - - - - - - - - 13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎ Logical OR

FNC(27)			16 bits: WOR & WOR(P) - - - - - - - - - - - - - - - - 7 Steps
D	WOR	P	32 bits: (D)WOR & (D)WOR(P) - - - - - - - - - - -13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎ Logical XOR

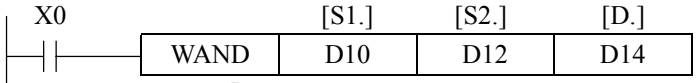
FNC(28)			16 bits: WXOR & WXOR(P) - - - - - - - - - - - - - - 7 Steps
D	WXOR	P	32 bits: (D)WXOR &(D)WXOR(P) - - - - - - - - - 13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

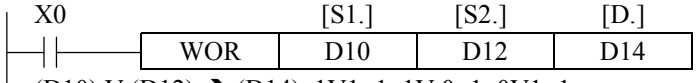
Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

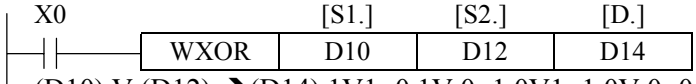
[D.]



(D10) Λ (D12) ➔ (D14), 1 Λ 1=1, 1 Λ 0=0, 0 Λ 1=0,



(D10) V (D12) ➔ (D14), 1V1=1, 1V 0=1, 0V1=1,



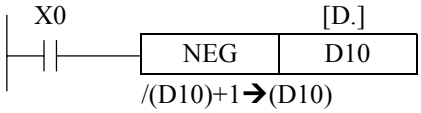
(D10) V (D12) ➔(D14),1V1=0,1V 0=1,0V1=1,0V 0=0.

◎ Negation

FNC(29)			16 bits: NEG & NEG(P) - - - - - - - - - - - - - - - - - - - 3 Steps
D	NEG	P	32 bits: (D)NEG & (D)NEG(P) - - - - - - - - - - - - -- -5 Steps

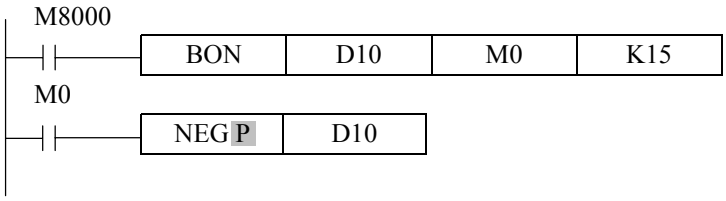
EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:	K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
			←			[D.]			→



- ♦ When X0 ON, the selected device [D.] is inverted. (“1”→”0”,“0”→”1”)
- ♦ When this is complete, a further binary 1 is added to the bit pattern. The result is become a negative number or a negative number will become a positive.

< Example >> Absolute Value of Negative



<< Note of Negation >>

(D 10)=2

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

(D 10)=1

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

(D 10)=0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(D 10)= -1 $\overline{(D\ 10)}$+1=1

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	→	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

(D 10)= -2 $\overline{(D\ 10)}$+1=2

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	→	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

(D 10)= -32,765 $\overline{(D\ 10)}$+1= 32,765

1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	→	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1

(D 10)= -32,766 $\overline{(D\ 10)}$+1= 32,766

1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	→	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0

(D 10)= -32,767 $\overline{(D\ 10)}$+1= 32,767

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	→	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(D 10)= -32,768 $\overline{(D\ 10)}$+1= -32,768

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	→	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

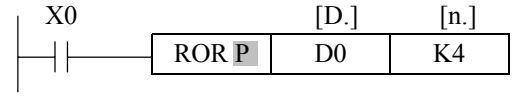
◎ Rotation Right

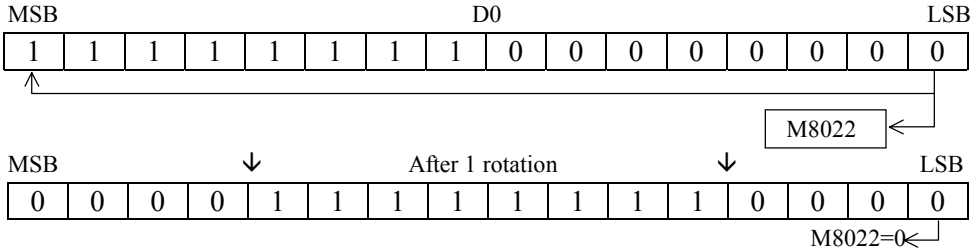
FNC(30)			16 bits: ROR & ROR(P) - - - - - - - - - - - - - - - - - - - 5 Steps
D	ROR	P	32 bits: (D)ROR & (D)ROR(P) - - - - - - - - - - - - -- -9 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

	K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
Operands:	←n→		←———			[D.]			———→

Flag: M8022





♦ After rotation right, the LSB of specified devices is shifted into carry flag M8022.

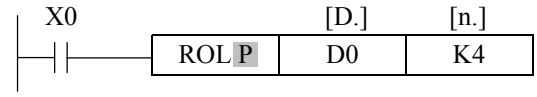
◎ Rotation Left

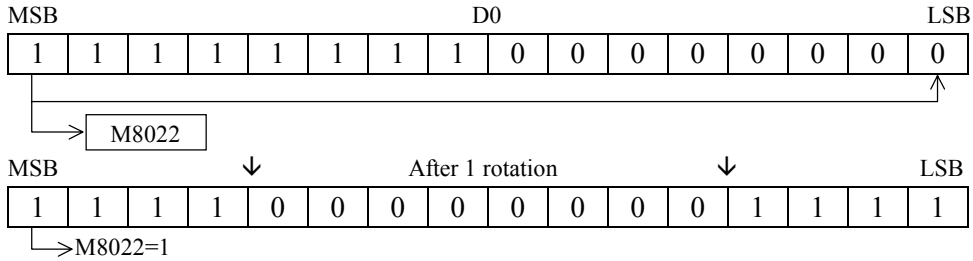
FNC(31)			16 bits: ROL & ROL(P) - - - - - - - - - - - - - - - - - - - 5 Steps
D	ROL	P	32 bits: (D)ROL & (D)ROL(P) - - - - - - - - - - - - - -9 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

	K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
Operands:	←n→		←———			[D.]			———→

Flag:





♦ After rotation left, the MSB of specified devices is shifted into carry flag M8022.

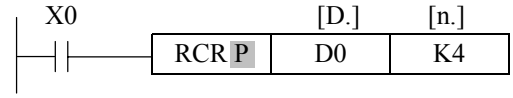
◎ Rotation Right with Carry

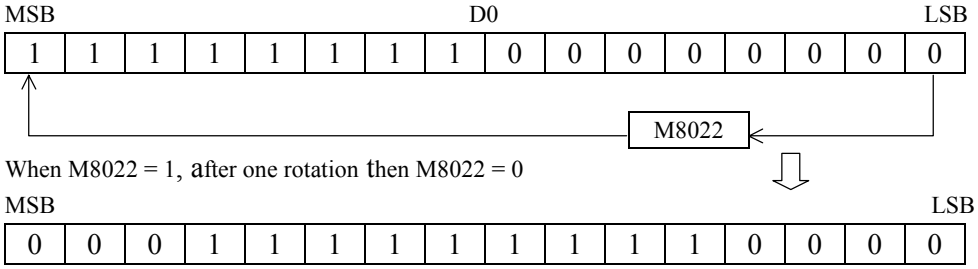
FNC(32)			16 bits: RCR & RCR(P) - - - - - - - - - - - - - - - - - - -5 Steps
D	RCR	P	32 bits: (D)RCR & (D)RCR(P) - - - - - - - - - - - - - -9 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

	K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
Operands:	←n→		←———			[D.]			———→

Flag:





When M8022 = 1, after one rotation then M8022 = 0

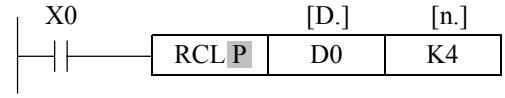
◎ Rotation Left with Carry

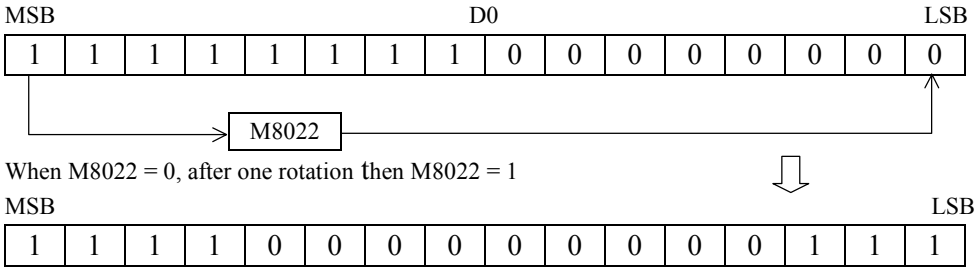
FNC(33)			16 bits: RCL & RCL(P) - - - - - - - - - - - - - - - - - - - 5
D	RCL	P	32 bits: (D)RCL & (D)RCL(P) - - - - - - - - - - - - - - -9

EX	EX_{1S}	EX_{1N}	EX_{2N}

	K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
Operands:	←n→		←———			[D.]			———→

Flag:





When M8022 = 0, after one rotation then M8022 = 1

◎ Shift Right

FNC(34)			16 bits: SFTR & SFTR(P) - - - - - - - - - - - - - - - - - 9 steps
	SFTR	P	

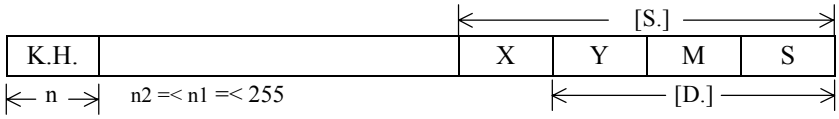
EX	EX_{1S}	EX_{1N}	EX_{2N}

◎ Shift Left

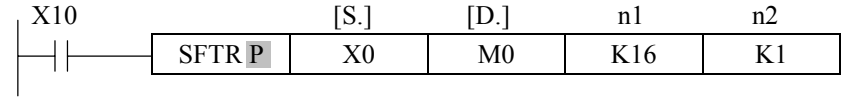
FNC(35)			16 bits: SFTL & SFTL(P) - - - - - - - - - - - - - - - - - 9 steps
	SFTL	P	

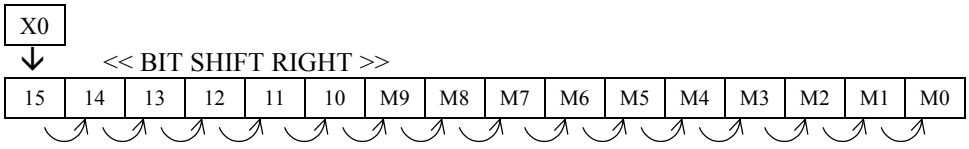
EX	EX_{1S}	EX_{1N}	EX_{2N}

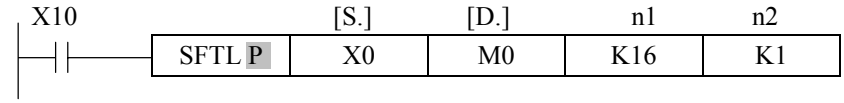
Operands:

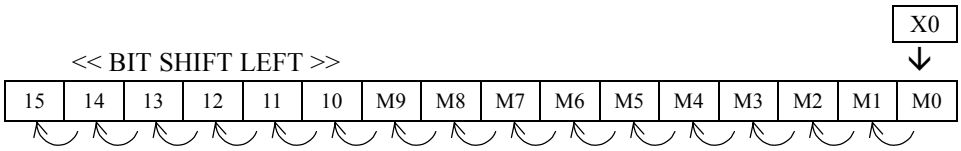


Flag:

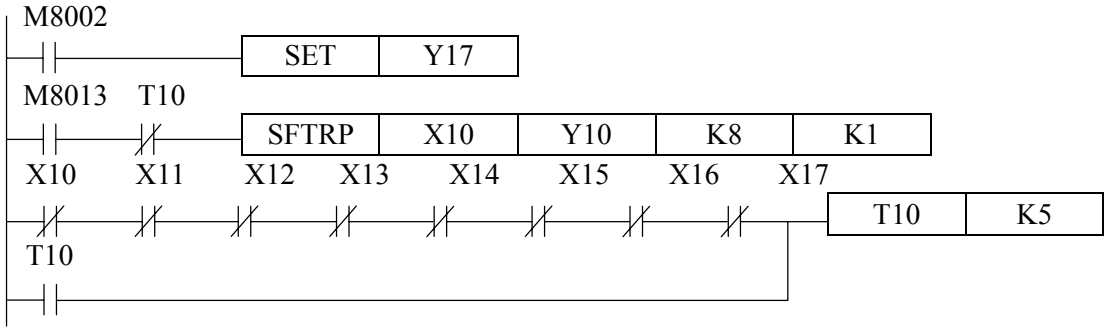








Example I/O Test: Wiring X10 ↔ Y10 … X17 ↔ Y17



◎ Word Shift Right

FNC(36)			16 bits: WSFR & WSFR(P) - - - - - - - - - - - - - - - - 9 steps
	WSFR	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎ Word Shift Left

FNC(37)			16 bits: WSFL & WSFL(P) - - - - - - - - - - - - - - - - 9 steps
	WSFL	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

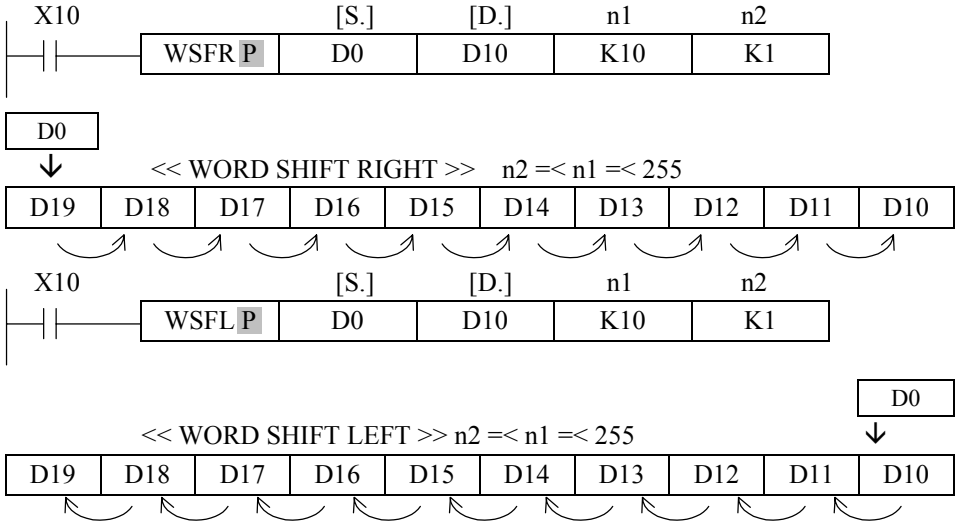
[S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

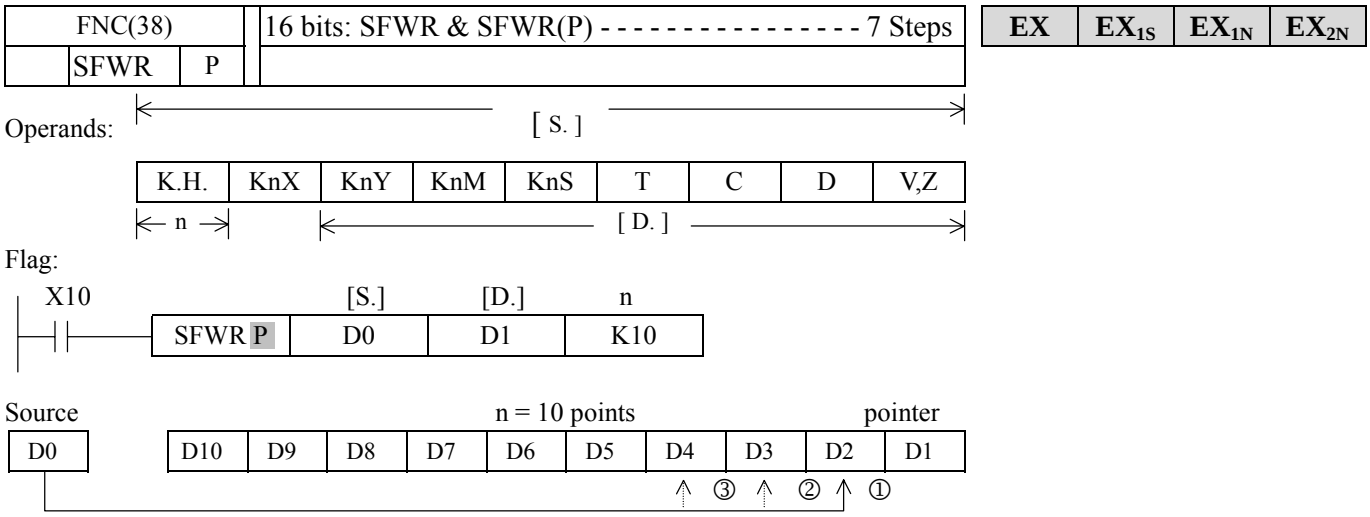
n

[D.]

Flag:



◎ Shift Register Write

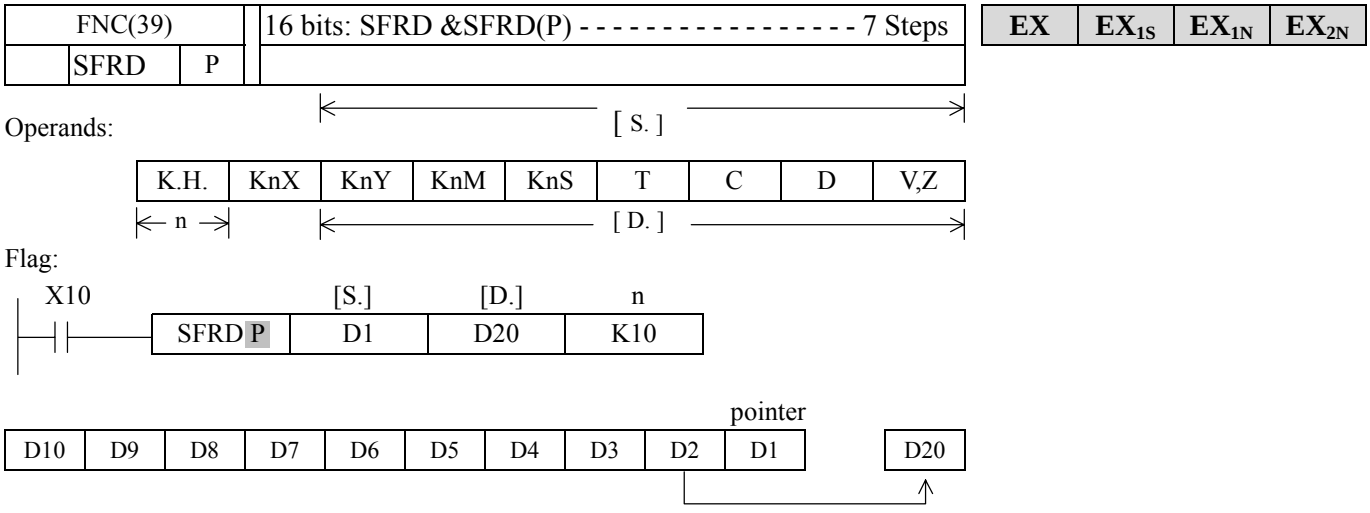


♦ When X10 OFF → ON, content of D0 stored into D2 and D1="1". When next rising pulse, content of D0 stored into D3 and D1="2", the position of insertion into the stack is automatically calculated by controller.

♦ If content of [D.] exceeds the value "n-1" (n is length of the FIFO stack), then insertion into the FIFO stack is stopped. The carry flag M8022 is turned ON.

♦ Before starting to use a FIFO stack, ensure that contents of the head address register [D.] are equal to "0".

◎ Shift Register Read



♦ When X10 OFF → ON, content of D2 stored into D20 and content of D1 decreased 1 (D1=D1-1).

♦ When contents of source device [S.] are equal to "0", i.e. the FIFO stack is empty, zero flag M8020 is turned on.

♦ This instruction will always read the source data from the register [S.]+1.

◎ Zone Reset

FNC(40)			16 bits: ZRST(P) - 5 steps
	ZRST	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

← [D1.][D2.] →								
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands:

← [D1.] [D2.] →			
X	Y	M	S

Flag:

- ♦ The range of specified devices are reset, for data devices, the current value is set to "0", and for bit elements, the bit status are turned OFF.
- ♦ The specified device range cannot contain mixed devices types, i.e. if C00 specified as the first destination devices [D1.], then cannot paired with T99 as the second devices.
- ♦ If [D1.] is bigger than (>) [D2.], then only [D1.] is reset.

◎ Decode

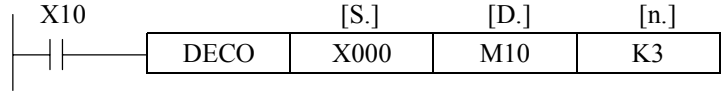
FNC(41)			16 bits: DECO(P) - 7 steps
	DECO	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

← [S.] →								
X	Y	M	S	K.H	T	C	D	V,Z

← [D.] → n = 1 – 8 ← [D.] →



- ♦ If the specified device [D.] is T, C or D, then n ≤ 4.
- ♦ If the sources all are "0", then M10 set to "1".

◎ Encode

FNC(42)			16 bits: ENCO(P) - 7 steps
	ENCO	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

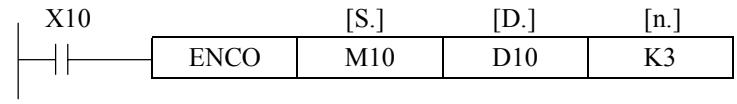
[S.]

X	Y	M	S	K.H	T	C	D	V,Z

[D.] [D.]

n = 1 – 8

Flag:



- If the specified device [S.] is T, C or D, then n ≤ 4.
- The number of active (ON) bits within the source device [S.] is more than one, only the highest bit effective.
- If bits of source device [S.] all are "0", then error occurred.

◎ Sum

FNC(43)			16 bits: SUM(P) - 5 steps
D	SUM	P	32 bits: (D)SUM(P) -9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

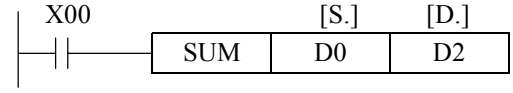
Operands:

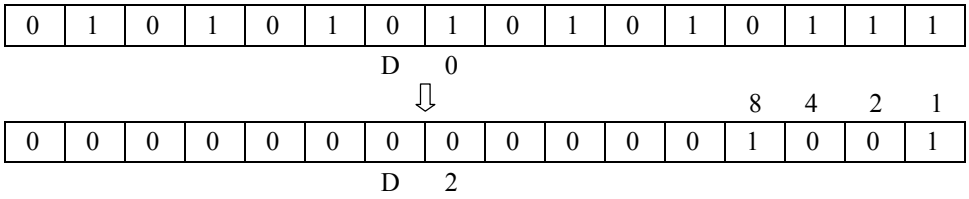
[S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]

Flag:





- The number of active (ON) bits within the source device [S.], i.e. bits which have a value of "1" are counted. The count is stored in the destination device [D.].
- If there is no bit as 0, then zero flag M8020 ON.

◎ Bit On Check

FNC(44)			16 bits: BON(P) - 7 steps
D	BON	P	32 bits: (D)BON(P) - 13 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S.]

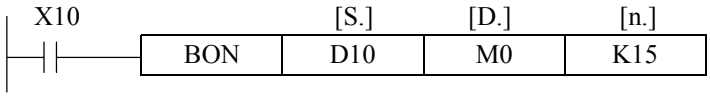
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[n.] = 0~15 or 0~31

Operands: [D.]

X	Y	M	S

Flag:



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

Bit15,D10=0, then M0 = OFF. LSB

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

Bit15,D10=1, then M0 = ON LSB

◎ Mean

FNC(45)			16 bits: MEAN(P) - 7 steps
	MEAN	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

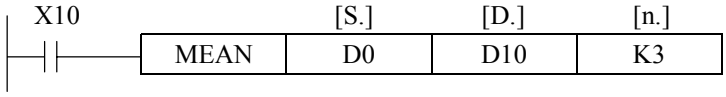
Operands: [S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: n [D.]

[n]=1-64

Flag:



♦ [(D0) + (D1) + (D2)] / 3 → (D10)

◎ Annunciator Set

FNC(46)			16 bits: ANS - 7 steps
	ANS		

EX	EX_{1S}	EX_{1N}	EX_{2N}

Reserved

◎ Annunciator Reset

FNC(47)			16 bits: ANR(P) - 1 steps
	ANR		

EX	EX_{1S}	EX_{1N}	EX_{2N}

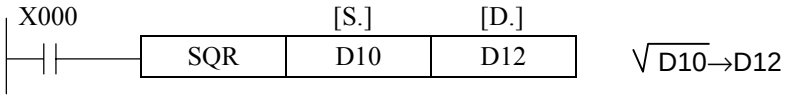
Reserved

◎ Square Root

FNC(48)			16 bits: SQR(P) - 5 steps
D	SQR	P	32 bits: (D)SQR(P) -9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Flag: M8020, M8021, M8022



- [S.] must be positive. When it is negative, error flag M8067 ON, and stop executing.
- When the result with decimal fraction, don't care it; but borrow flag M8021 will ON.
- When result is 0, zero flag M8020 will ON.

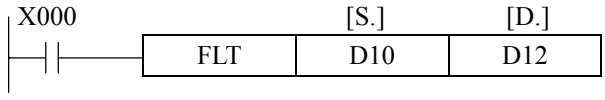
◎ Float

FNC(49)			16 bits: FLT(P) - 5 steps
D	FLT	P	32 bits: (D)FLT(P) -9 steps

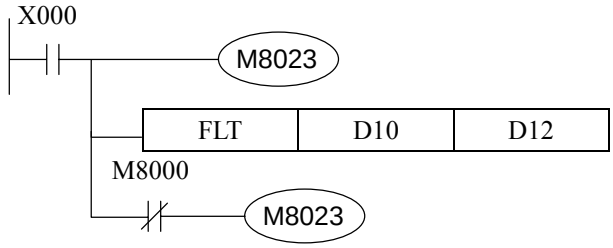
EX	EX_{1S}	EX_{1N}	EX_{2N}

Flag: M8020, M8021, M8022

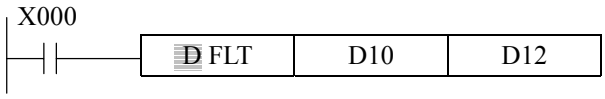
♦ FLT Instruction is converted command between BIN integer and binary floating data. Because constant K, H will automatically convert when floating data operate, then not fit this instruction



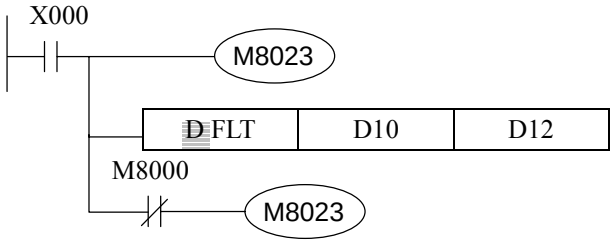
(D10) → (D13、D12)
BIN integer binary floating data



(D11、D10) → (D12)
binary floating data BIN integer
Decimal fraction don't care



(D11、D10) → (D13、D12)
BIN integer binary floating data



(D11、D10) → (D13、D12)
binary floating data BIN integer
Decimal fraction don't care

♦ When M8023 = ON, execute binary floating data → BIN integer。
When M8023 = OFF, then execute BIN integer → binary floating data.

♦ Binary floating data → BIN integer, the operating result is decimal fraction, don't care it, but M8021 / M8022 will ON; when result is 0, M8020 will ON

◎ Output & Input Refresh

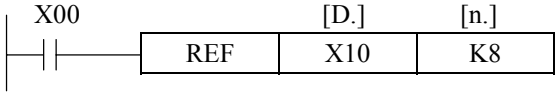
FNC(50)			16 bits: REF(P) - 5 steps
	REF	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

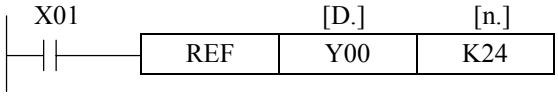
Operand: [D.] should always be a multiple of 10, i.e. 00,10..
[n.] should always be a multiple of 8, i.e. 8,16,24..

♦ PLC input all refresh before program STEP 0 execute; output is executed after END or FEND instruction. It is not changed in performing process. If it needs immediately input data or output performing result in the performing process, then have to use output & input refresh instruction.

<< Input Fresh >> only X10 – X17 to be flashed



<< Output Fresh >> refresh Y00-Y07, Y10-Y17, Y20-Y27.



◎ Refresh and Filter Adjustment

FNC(51)			16 bits: REFF(P) - 3 steps
	REFF	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operand: [n.] = 0 - 60

♦ To avoid noise interference, PLC input relay all designed with hardware RC filter to adjust software filter time.

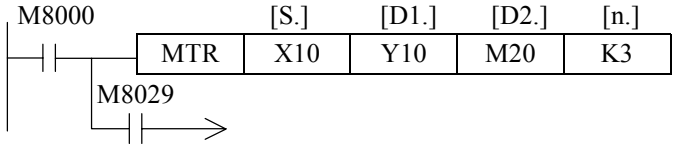
♦ This instruction only change X00-X07 software filter time, i.e., content of D8020. If it has to change other input point filter time, please use MOV instruction.

◎ Matrix

FNC(52)	16 bits: MTR - 9 Steps
MTR	

EX	EX_{1S}	EX_{1N}	EX_{2N}

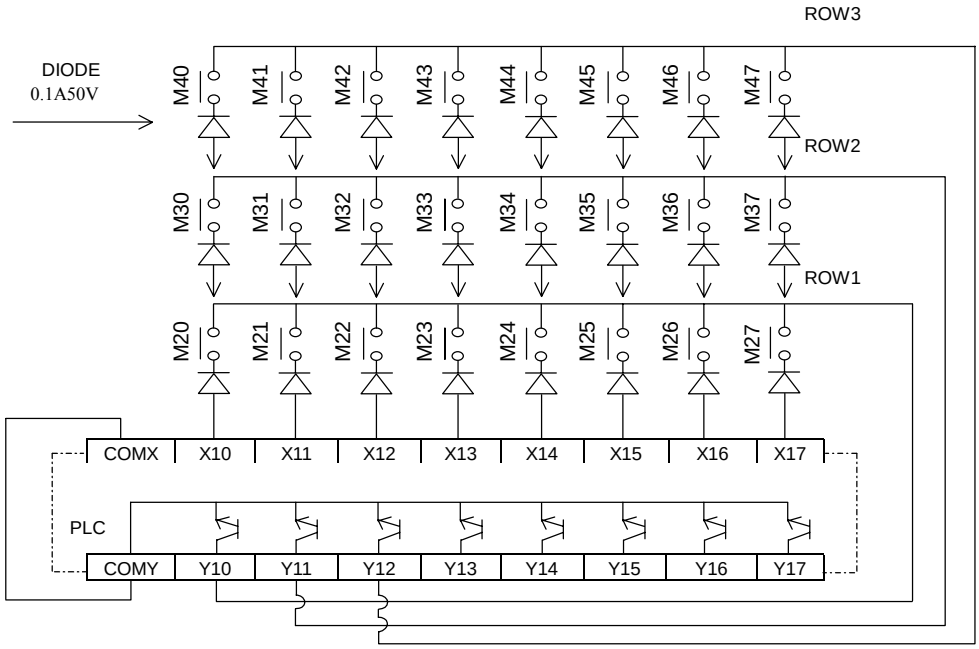
Operand: (S.): X00, X10, X20, X30 - - - - - - - X160, X170.
(D1.): Y00, Y10, Y20, Y30 - - - - - - Y160, Y170.
(D2.): Y, M, S multiple of 10, i.e. 00, 10, 20 etc.
(n.): K, H. n=2 ~ 8.

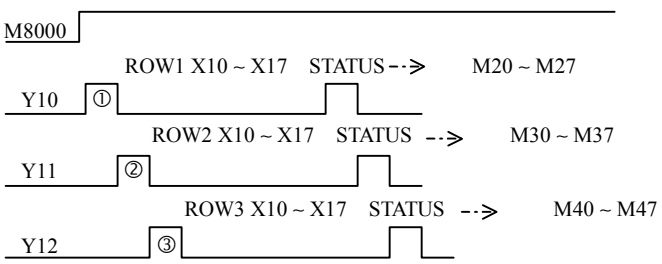


♦ MTR instruction allows 8 consecutive input devices [S.] to be used multiple (n) times. The result was stored in (D2.).
(S.): Head address of the input devices. (n.): row numbers.
(D1.): Head address of the output trigger devices.
(D2.): Head address of the matrix table.

♦ After completion of full reading of the matrix, the complete flag M8029 to be turned ON. This flag will be automatically reset when this instruction is executed.

♦ This instruction can be used once, and only the transistor module can be selected.





◎ Set by High Speed Counter

FNC(53)			
D	HSCS		32 bits: HSCS - 13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

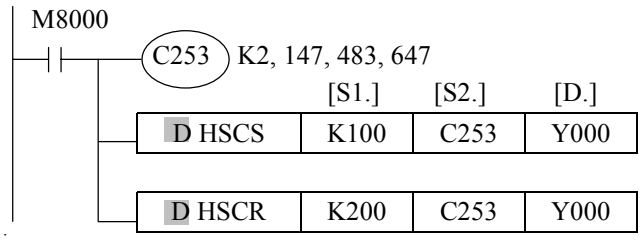
[S2.] : C235~C255

Operands:

X	Y	M	S

[D.] When [D.], can use Index to assign I010~I060 to interrupt.

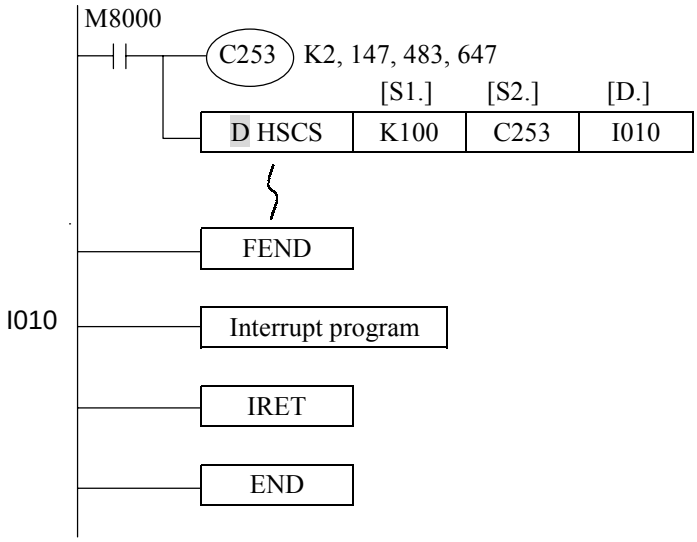
Flag:



♦ When use FNC53, operate external output action by interrupt. When current value of C253 changed from 99 to 100 and from 101 to 100, Y000 will be set. When current value of C253 is changed from 199 to 200 and from 201 to 200, Y000 OFF.

♦ This command is specialized instruction of 32 bits, please input D HSCS command.

♦ Only can use FNC53, FNC54, FNC55 once.



♦ [D.] of D HSCS can assign I0 □ 0 = (□=1~6)(□=1~6 can not be reuse.)

♦ Therefore, when current value of High Speed Counter which is assigned by [S2.] is as same as the value which is assigned by [S1.], interrupt main program and jump to execute I0 □ 0 interrupt program immediately.

♦ When Special auxiliary relay M8059 ON, I010~I060 interrupt are all prohibited.

◎ Reset by High Speed Counter

FNC(54)		
D	HSCR	

32 bits: HSCR - 13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

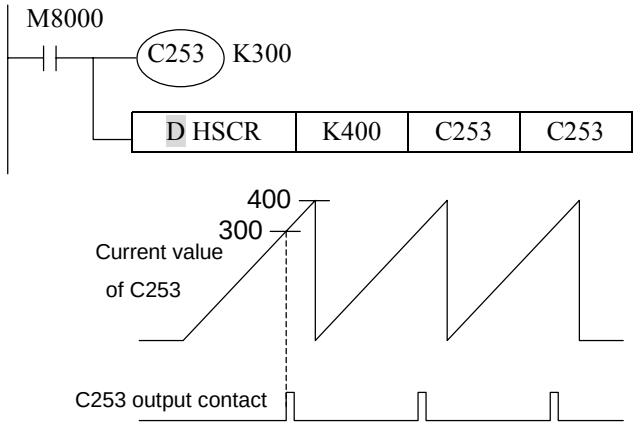
[S2.] : C235~C255

Operands:

X	Y	M	S

[D.] Can assign [D.] and [S2.] are the same High Speed Counter.

Flag:



♦ When current value of C253 is 400, C253 will be cleared immediately. Current value will become 0, and output contact will not act.

♦ This command is specialized instruction of 32 bits, so have to use D HSCR.

◎ Zone Compare For High Speed Counter

FNC(55)		
D	HSZ	

32 bits: HSZ- 17 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

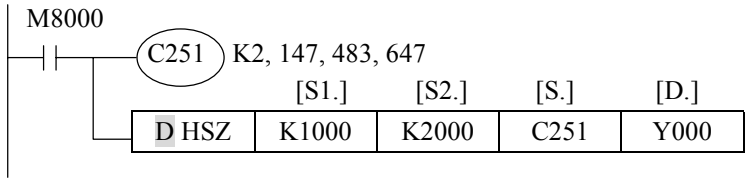
[S2.] : C235~C255

Operands:

X	Y	M	S

[D.]

Flag:



<Compare action of input>

K1000>C251 current value	Y000	ON
K1000≦C251 current value≦K2000	Y001	ON
K2000<C251 current value	Y002	ON

♦ This command is specialized instruction of 32 bits, so have to use D HSZ.

♦ Content of [S1.] and [S2.] is according to [S1.] ≦ [S2.] .

♦ When use FNC55, operate external output by Interrupt. Output will act without effect by scan-cycle.

◎ Speed Detect

FNC(56)	16 bits: SPD - 7 Steps
SPD	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: (S1.): X000~X005. When C251 is used, X02 and X03 can not be used.

Operands: [S2.]

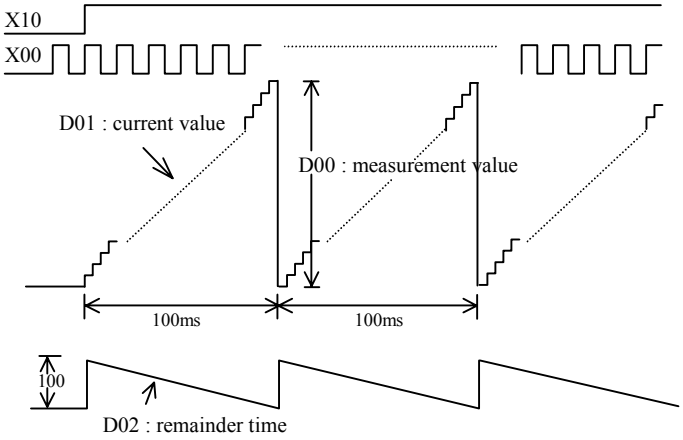
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

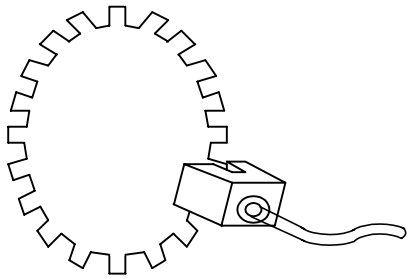
Flag: M8029 [D.]

	[S1.]	[S2.]	[D.]
SPD	X00	K100	D00

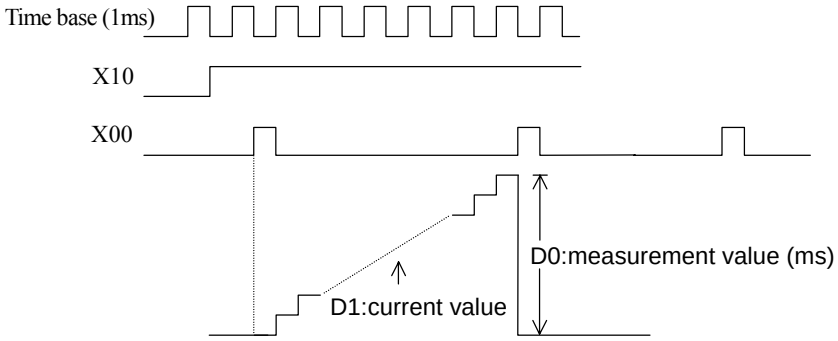
- The input pulse assigned by [S1.], and the [S2.] assign measurement time, the result will be stored at [D.].
- This will automatic occupy 3 word devices from the head address of [D.]. (D00~D02)
- This example D01 count up the pulse number of X00 (OFF→ON), and put the result into D00 at 100msec after. Then reset D01to "0" and start counting again.
- D02 is used to measurement remainder time.
- The counting pulse amount of the assign time can't be more than 65535
- Following formula can calculated RPM
  RPM : N = (D00×60) × 1000 / n × t
  n: (pulse/revolution), t: (measurement time).
- The pulse frequency of (X00-X05) is same with HSC.
- If input relay (X00-X05) is assigned by the SPD, they can't be used to other purpose or interrupt input point.
- If pulse output assign Y00, then X00 can't be used; if assign Y01, then X01 can't be used.
- V1.45 or more, add complete flag M8029, easily reach many data of continuous measurement, then count an average value.

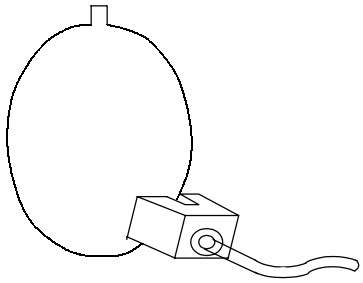
.(i) measure frequency mode





(ii) measure cycle mode





- ♦ When the content of [S2.] is ”0”, then it’s measurement cycle mode.
- ♦ The measurement time (ms) and RPM are inverse ratio, it can get number of turning round by formula thereinafter RPM = N = 60 × 1000/D0

◎ Pulse Output

FNC(57)			16 bits: PLSY(P) - 7 steps
D	PLSY		32 bits: (D)PLSY(P) - 13 steps

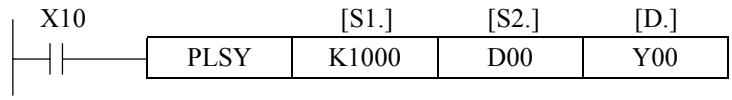
EX	EX_{1S}	EX_{1N}	EX_{2N}

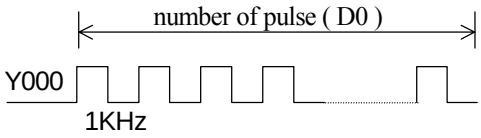
Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.] : Y00 – Y01

Flag: M8029





- ♦ This instruction is pulse output without slope.
- ♦ [S1.] : assign output frequency (1~5000Hz), [S2.]: assign output pulse
  [D.]: assign pulse output point
- ♦ PLSY is used to output a consecutive pulse
  16 bit range: 1 ~ 32,767, 32 bit range: 1 ~ 2,147,483,647.
- ♦ If [S2.] is specified to 0, then it will continue to generate pulse.
- ♦ The pulse duty cycle is 50% ON 50% OFF.
- ♦ Data of [S2.] can be changed during execution, but the new will not be effective until current operation has been completed, and complete flag M8029 set to ON .
- ♦ This instruction can be used once, and only the transistor module can be selected.

◎ Pulse Width Modulation

FNC(58)			16 bits: PWM - 7 steps
	PWM		

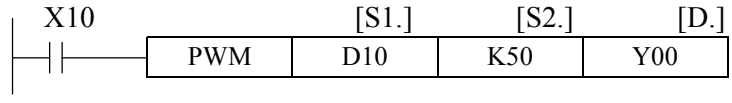
EX	EX_{1S}	EX_{1N}	EX_{2N}

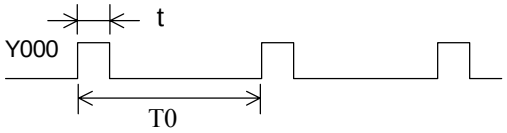
Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.] : Y00 – Y03

Flag: None





- ♦ [S1.]: ON duty width (t), range: (1 ~ 32,767 msec)
- ♦ [S2.]: (T), range: (1 – 32,767 msec)
- ♦ [D.]: The output point (Y). (by interrupt handing)
- ♦ If content of [S1.] is bigger than content of [S2.], then error occurred.
- ♦ This instruction can be used once, and only the transistor module can be selected.

◎ PULSE OUTPUT WITH SLOPE

FNC(59)			16 bits: PLSR - 9 steps
D	PLSR		32 bits: (D)PLSR- 17 steps

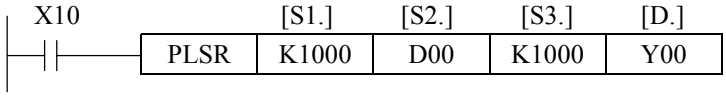
EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.][S2.][S3.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.] : Y00 – Y01

Flag: M8029



♦ [S1.] assign output frequency.(10 ~ 100,000 pps),
[S2.] assign opposite position (M8134,M8135=0) or absolute position (M8134,M8135=1)output pulse.
[S3.] assign acceleration/deceleration time. When set acceleration/deceleration separate flag is just for acceleration time, then D8165 and D8167 are deceleration time.
[D.] assign pulse output point. (Fix Y00 and Y01 to be pulse output point, and Y02, Y03 to be direction output point).

♦ When use this instruction, have to convert increment distance or absolute position to pulse, then stored at [S2.].

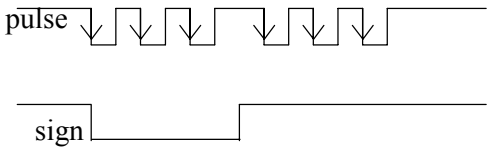
♦ The pulse duty cycle is 50% ON, 50% OFF

♦ When pulse output, X10 OFF, pulse is stopped outputting immediately.

♦ When instruction running, changing content of [S2.] is ineffective.

♦ This instruction for Y00 or Y01 only can be used once (total twice), and have to select transistor output type.

♦ There is only one kind of pulse output type in this instruction (Negative Logic Type, Pulse & Sign) can be controlled step or servo motor.



♦ (X00: Y00 0 signal, X01:Y01 0 signal, X06:Y00 dog point signal, X07:Y01dog point signal)

X00	X01	X02	X03	X04	X05	X06	X07
Zero1	Zero2					Dog1	Dog2

♦ Use this instruction for zero return, dog point must assigned to X06(Y00) or X07(Y01), and servo zero must assigned to X00(Y00) or X01(Y01). Servo End & Servo Ready are assigned by user.

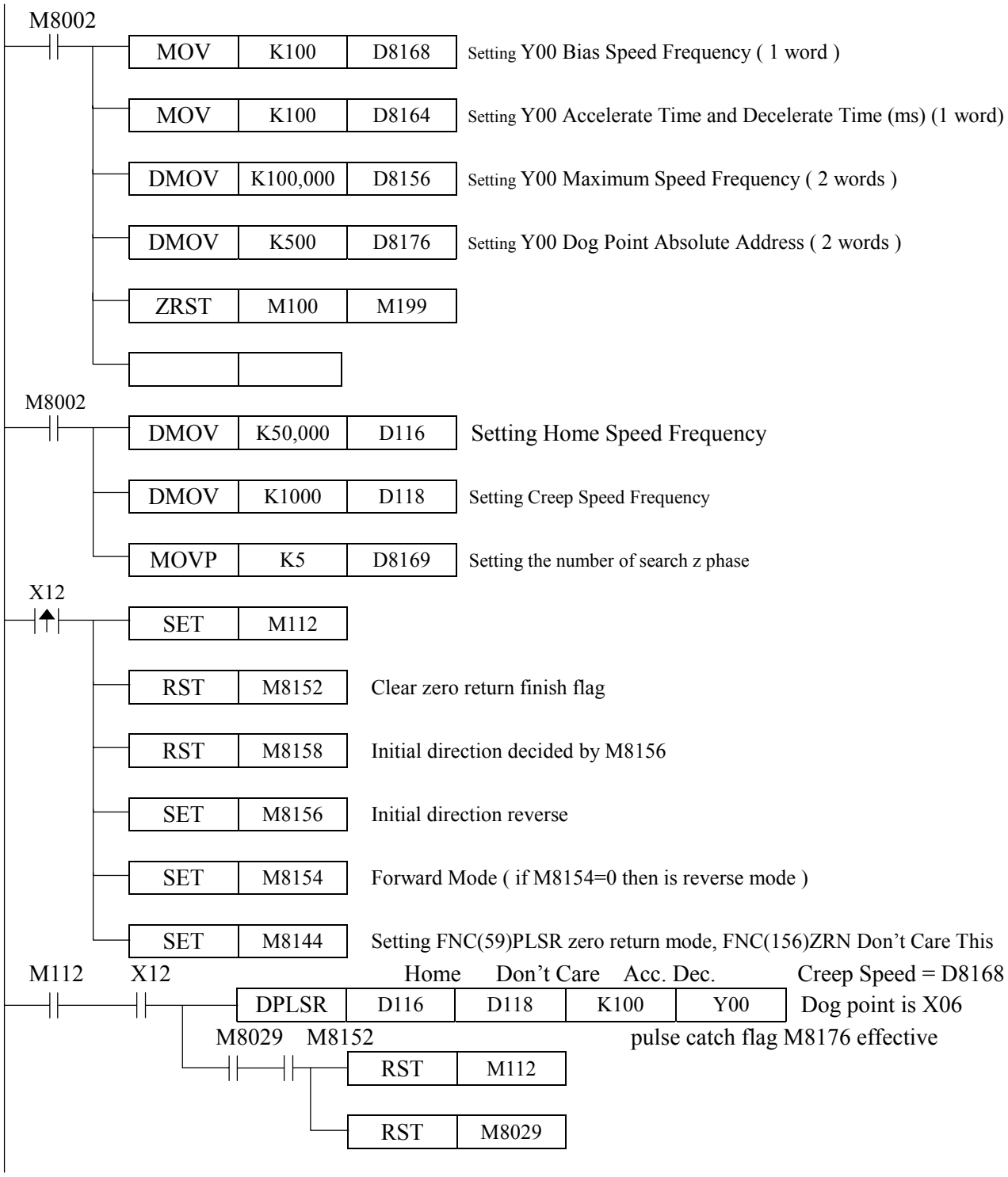
♦ This is a multifunction, so PLSR can execute JOG, zero return, single position drive by setting different flag.

♦ When zero return, if there is no zero signal (stepping motor), set parameter D8169 or D8171 to ”0”.

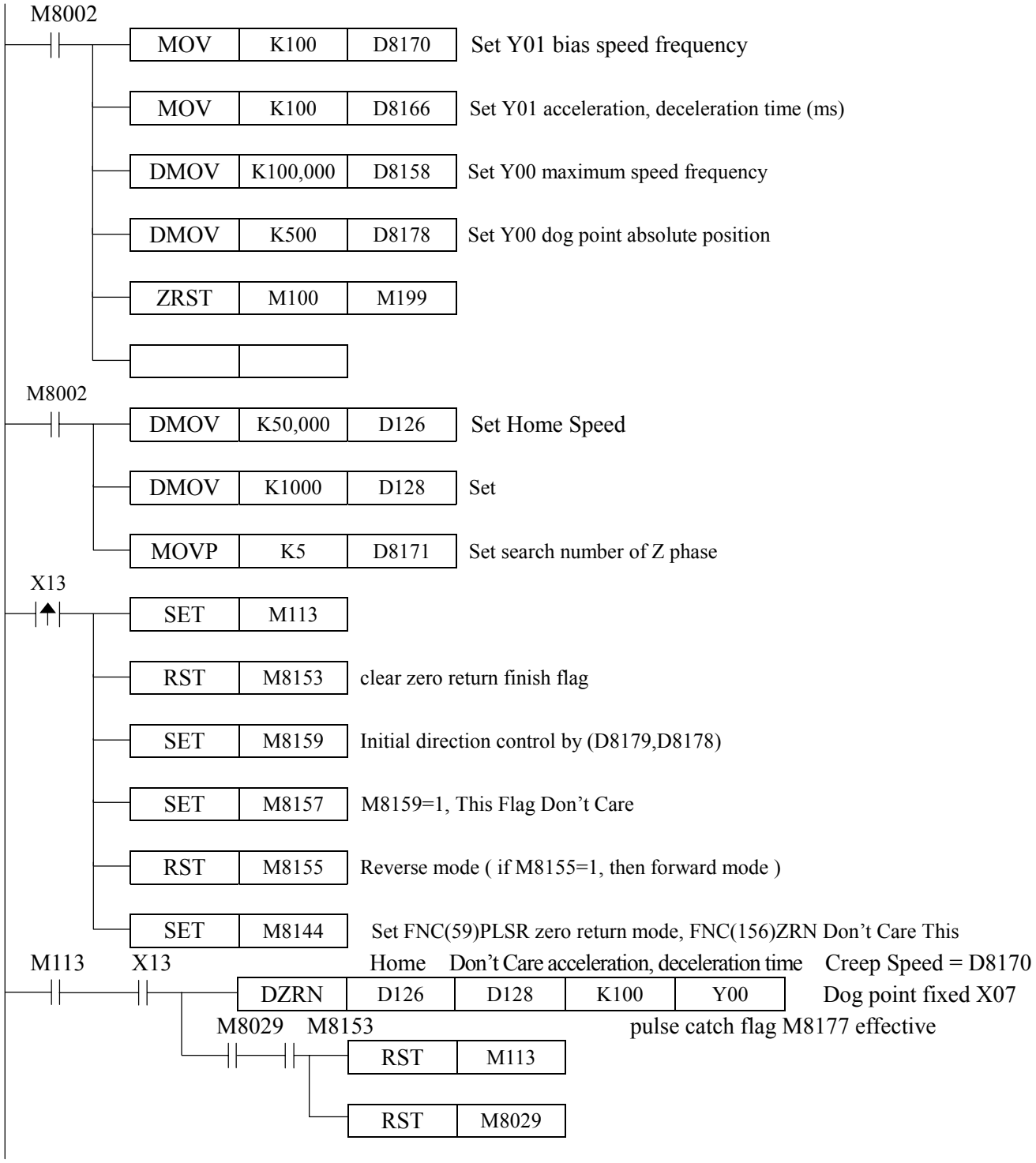
♦ Fixed Y00 and Y01 to Pulse output signal; Y02 and Y03 to Sign output signal.

♦ After this instruction execute, acceleration and deceleration time D8164 and D8166 data will be changed to [S3.]

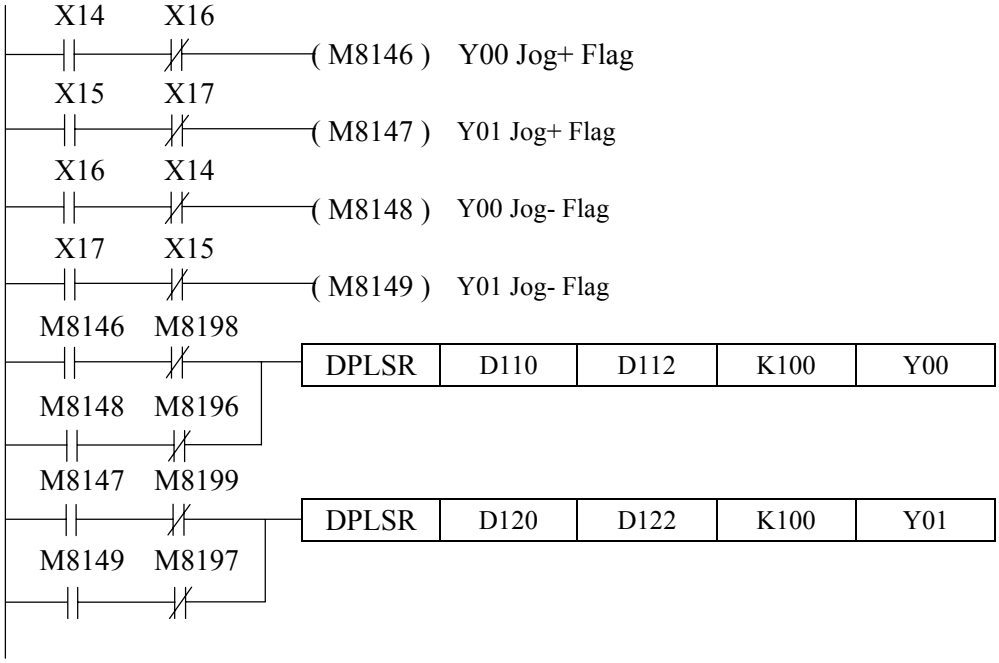
- FNC(59) Zero Return EX(1) (This Example Initial Direction controlled by Flag M8156 or M8157)
  Initial Direction also can be controlled by the absolute address of dog point



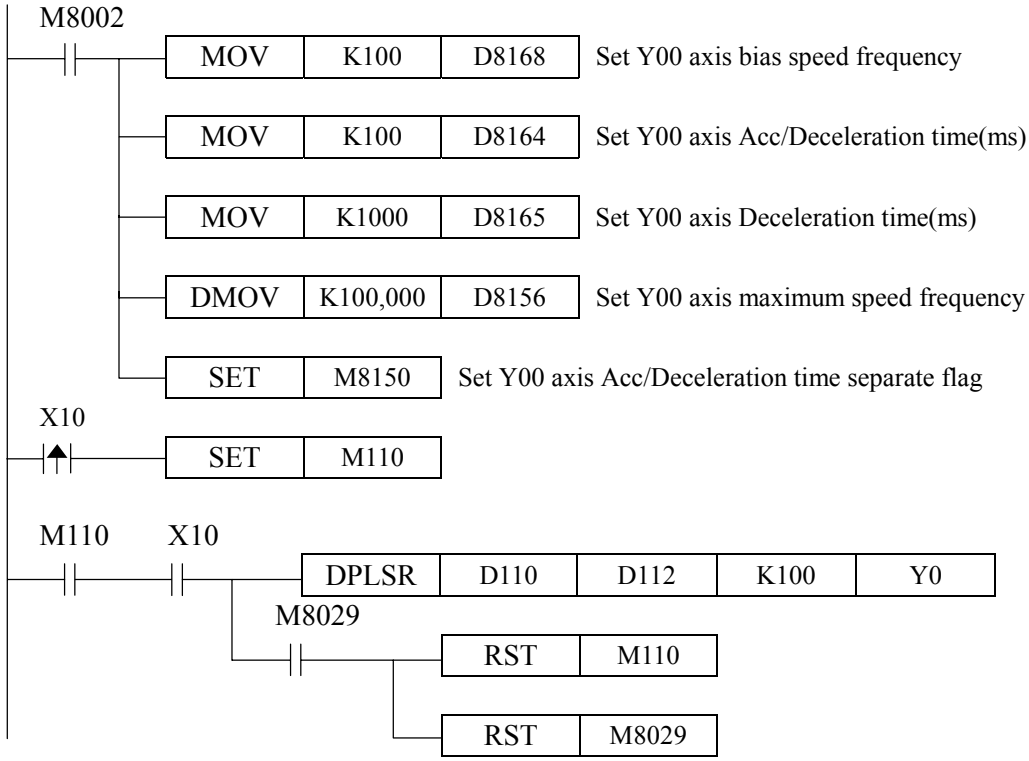
- FNC(59)Zero Return EX(2) (Initial direction controlled by dog point absolute position)
  Also could choose Initial direction to be controlled by flag M8156 or M8157
  If bias absolute position greater than dog point absolute position, then reverse. If bias absolute position less than dog point position, then forward.



- Example FNC(59) Jog+ and Jog- (M8029 ineffective)



- Example first position drive



◎ Initial State

FNC(60)			16 bits: IST - 7 steps
	IST		

EX	EX_{1S}	EX_{1N}	EX_{2N}

Reserved

◎ Data Search

FNC(61)			16 bits: SER(P)- -9 steps
D	SER	P	32 bits: (D)SER(P) - - - - - - - - - - - - - - - - - - - 17 steps

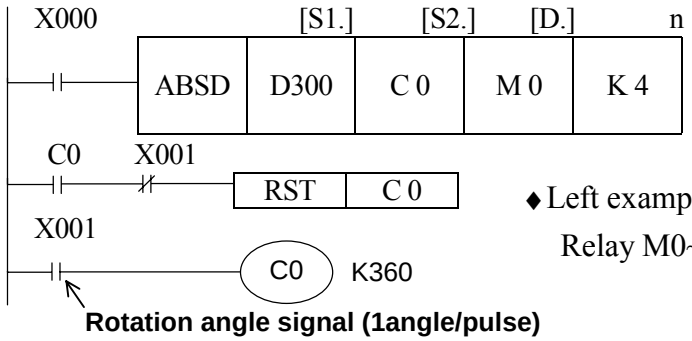
EX	EX_{1S}	EX_{1N}	EX_{2N}

Reserved

◎ Absolute Drum Sequence

FNC(62)	16 bits: ABSD - 9 steps
ABSD	

EX	EX_{1S}	EX_{1N}	EX_{2N}



This instruction is used to bring a varied output type to counter. It can detect the angle of the circle control action.

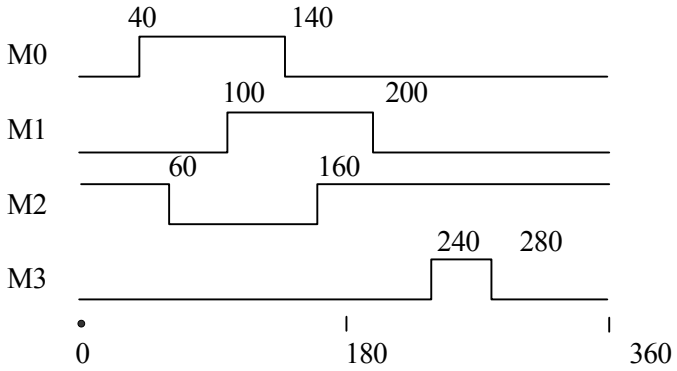
♦Left example is used to control ON/OFF status of Auxiliary Relay M0~M3 when rotation table rotate within a circle.

♦ Using MOVE instruction to write following values into D300~D307

ON setting value	OFF setting value	Output point
D300= 40	D301= 140	M0
D302= 100	D303= 200	M1
D304= 160	D305= 60	M2
D306= 240	D307= 280	M3

Put Turn ON value to even number of D device, and put Turn OFF value to Odd number of D device

♦When X0 ON, change of M0~M3 is mentioned as follows. Turn ON and Turn OFF value can re-change to write into D300~D307



♦Output point number is decided by setting value of [D.]

♦When X0 become OFF, output is not changed.

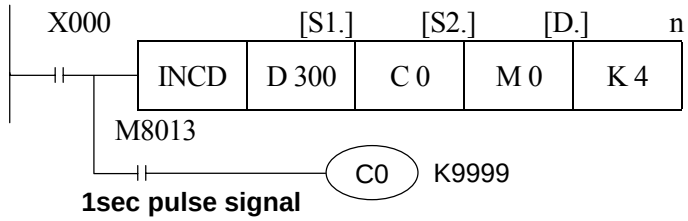
♦ABSD instruction just can be used once in one program.

♦When assign High Speed Counter in [S.], then also can use (D)ABSD instruction.
For current value of counter at this time, the output status will delay because of scan-time, recommend to use Table high-speed compare mode of HSZ instruction.

◎ Incremental Drum Sequence

FNC(63)	16 bits: INCD - 9 steps
INCD	

EX	EX_{1S}	EX_{1N}	EX_{2N}

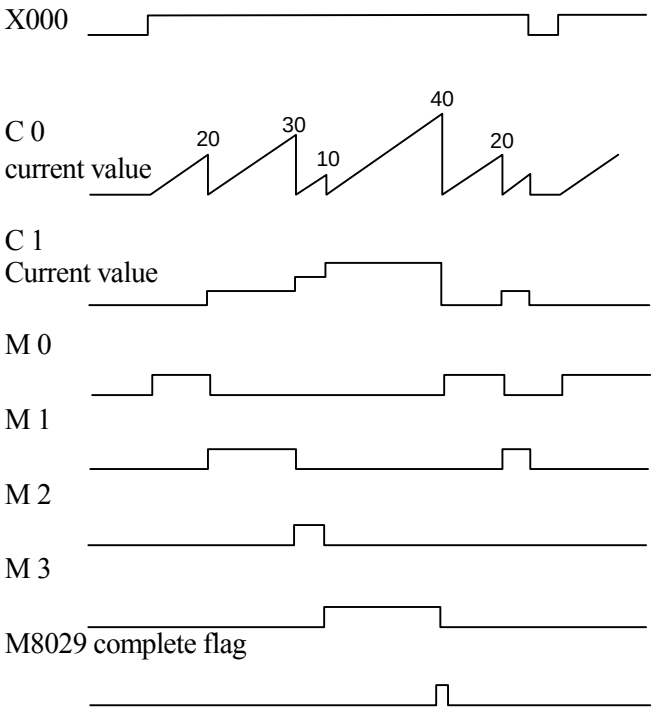


This instruction is used to produce a varied output when using a couple of counter.

Following is the control range of 4 points (M0~M3)

♦ Use MOVE instruction to write following value into [S1.] in advance.

D300 = 20 D302 = 10
D301 = 30 D303 = 40



♦ When counting value of C0 reach to setting value of D300~D303, C0 reset automatically in turn

♦ C1 count occurred number of C0 reset.

♦ M0~M3 act in turn according to counting value of C1.

♦ After complete last operation of setting number by ”n”, flag M8029 become ON. Above mentioned action will be always repeated.

♦ When X0 OFF, C0 and C1 is cleared, M0~M3 become OFF, then operate again when X0 become ON.

♦ INCD instruction only can be used once in one program.

◎ Teaching Timer

FNC(64)	16 bits: TTMR - 5 steps
TTMR	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Reserved

◎ Special Timer

FNC(65)	16 bits: STMR - 7 steps
STMR	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Reserved

◎ Alternate Output

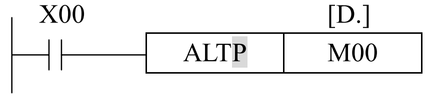
FNC(66)			16 bits: ALT(P) - 3 steps
	ALT	P	

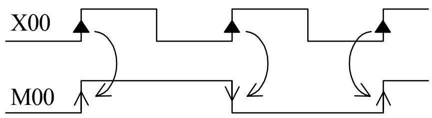
EX	**EX_{1S}**	**EX_{1N}**	**EX_{2N}**

Operands: [D.]

X	Y	M	S

Flag:





◎ Ramp

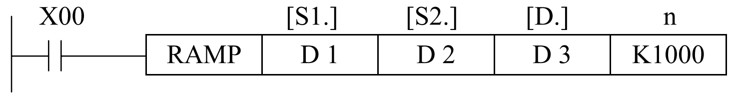
FNC(67)			16 bits: RAMP - 9 steps
	RAMP		

EX	**EX_{1S}**	**EX_{1N}**	**EX_{2N}**

Operands: [S1.][S2.][D.] : D

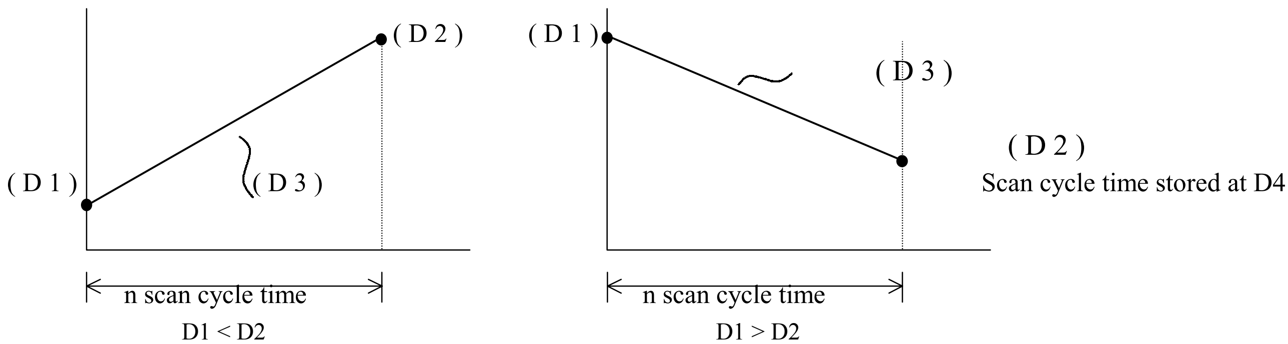
n : K,H n = 1 to 32,767

Flag: M8029



- When X0 ON, content of [S1.] and [S2.] are stored into [D.]. Content of [D.] is increased by "1" each scan cycle.

  n: the number of scan cycle.



- After M8029 is driven, write once scan-time value (longer than actual scan-time) into M8039, and then PLC will enter to fixed scan mode.

  For example, n = K1000 in above example. If scan cycle is set to 20msec, then value in D3 will be changed from setting value of D1 to setting value of D2 within 20sec.
- If X0 become OFF when acting, then act of RAMP signal will stop in midway. If X0 ON again, then D4 will be cleared and D3 will restart by setting value of D1.
- After end of execution, flag M8029 act, and then value of D3 will return to value of D1.
- Control of start / end point can be executed by RAMP instruction and analog output.
- Enter into RUN status when X0 ON.

◎ Rotary Control

FNC(68)			16 bits: ROTC - 9 steps
	ROTC		

EX	EX_{1S}	EX_{1N}	EX_{2N}

Reserved

◎ Sort

FNC(69)			16 bits: SORT - 11 steps
	SORT		

EX	EX_{1S}	EX_{1N}	EX_{2N}

Reserved

◎ Tenkey Input

FNC(70)			16 bits: TKY - 7 Steps
D	TKY		32 bits: (D)TKY -- 13 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [D1.]

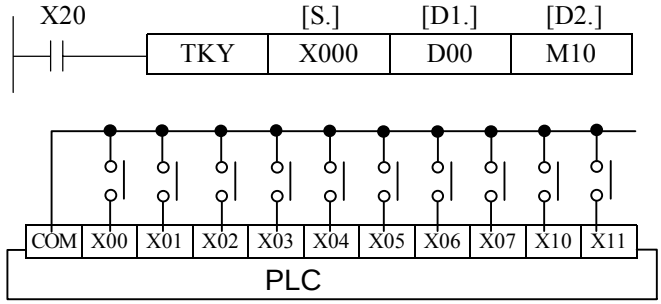
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: [S.]

X	Y	M	S

[D2.]

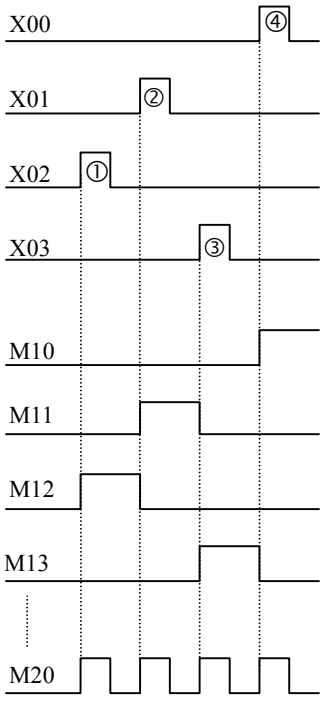
Flag:



♦ This instruction can read 10 consecutive devices and will store an entered numeric string in [D1].

♦ In 16 bits operation, [D1] can store numbers from 0000 to 9999 (max. 4 digits). In 32 bits operation, [D1] value from 00000000 to 99999999 (max. 8 digits). In both cases, if the number exceeds the allowable ranges, the highest digit will overflow, and ignored it.

♦ When X20 OFF, all of the [D2.] devices are reset, but contents of [D1.] keep intact.



♦ This example will stored 2130 into D00.

♦ When two or more keys were pressed, only the First Key is effective.

♦ This example, the status of M10~M19 to be changed by the status of X00~X11.

♦ This example M20 is key release flag.

♦ This instruction may only be used once.

◎ Hexadecimal Key

FNC(71)			16 bits: HKY - 9 Steps
D	HKY		32 bits: (D)HKY - 17 Steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[D2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

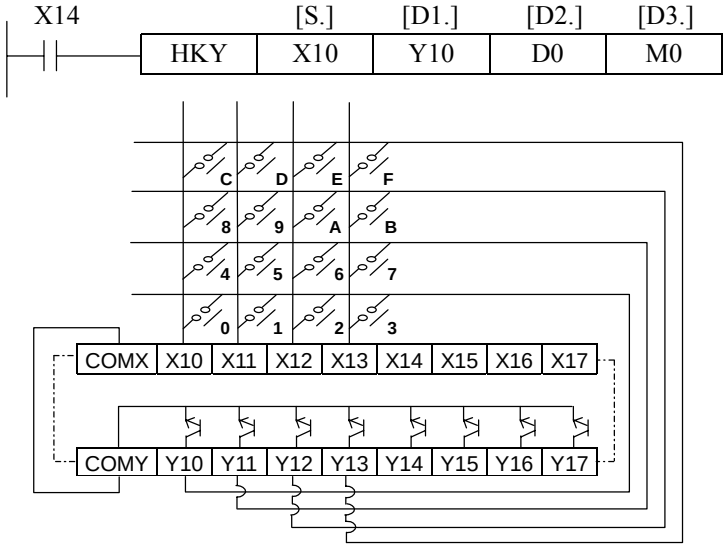
Operands:

[D3.]

X	Y	M	S

[D1.] [S.]

Flag:



- ♦ When the numeric key (0 ~ 9) be pressed, then causes bit device [D3.]+7 turn ON for the duration of key press.
- ♦ When the function key (A ~ F) be pressed, then causes bit device [D3.]+6 turn ON for the duration of key press.
- ♦ When the function key has been pressed, then will set bit devices [D3.]+0 to [D3.]+5 to ON, and remain ON until the next function key has been activated.

F	E	D	C	B	A
↓					↓
M5	M4	M3	M2	M1	M0

- ♦ In 16 bits operation, [D2.] can store numbers from 0000 to 9999 (max. 4 digits). In 32 bits operation, [D2.] value from 00000000 to 99999999 (max. 8 digits). In both cases, if the number exceeds the allowable ranges, the highest digit will overflow, and ignored it.
- ♦ When two or more keys were pressed, only the first key is effective. When X14 OFF, all [D3.] devices are reset, but contents of [D2.] keep intact.
- ♦ This instruction requires 8 scans cycle time to read the key input. After 8 scans, complete flag M8029 to be turned ON. This flag is automatically reset when this instruction execute.
- ♦ This may only be used once, and only the transistor module can be selected.

◎ Digital Switch

FNC(72)	16 bits: DSW - 9 Steps
DSW	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [n] =1~8 [D2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

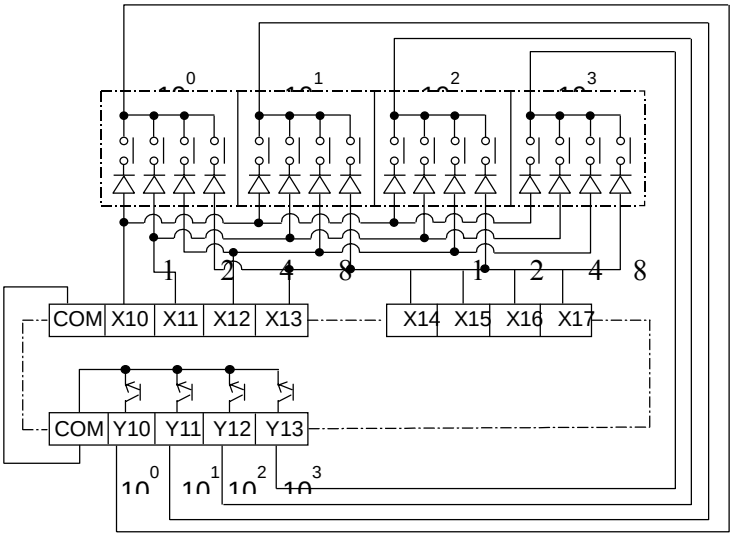
Operands: [S.] [D1.]

X	Y	M	S

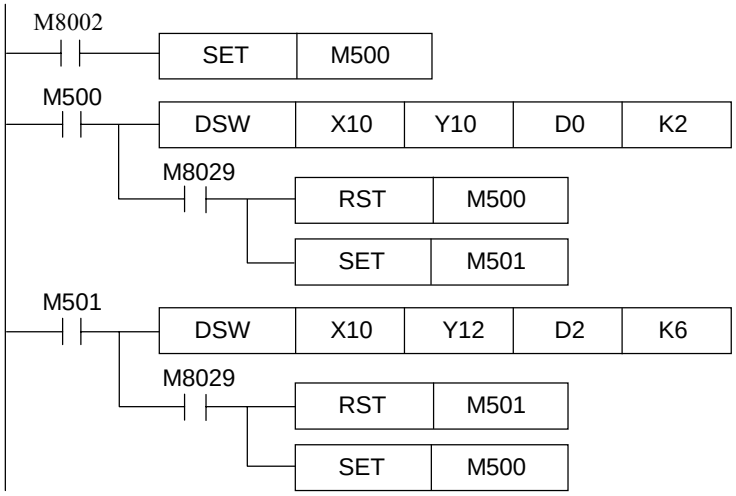
Flag:M8029

X00

DSW	[S.]	[D1.]	[D2.]	[n]
	X10	Y10	D0	K4



- ♦ This instruction used n (1~8) output points and 4 input points to read in n (1~8) thumbwheel switch. If the read data is larger than 32 bits (n≥5), then [D2.] automatically occupy the next word device.
- ♦ This example the BCD 4 digit thumbwheel switch (1,2,4,8) is connected to X10~X13 or X14~X17, the source [S.] needs to be used X10,X14,X20,X24….as the head address.
- ♦ Once DSW execute, then the flag M8029 reset to "0". When execution is completed, M8029 set to "1".
- ♦ Each pin (1,2,4,8) of the thumbwheel switch needs to be connected a diode (0.1A/50V)
- ♦ This may only be used once, and only the transistor module can be selected. If use M8029, then can control two or more DSW .



◎ Seven Segment Decoder

FNC(73)			16 bits: SEGD(P) - 5 steps
	SEGD	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

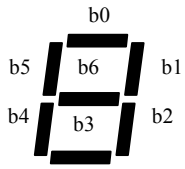
[S.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]

Flag:

- A single hexadecimal digit (0~9, A~F) occupying the lower 4 bits of the source device [S.] is decoded to a data format used to drive a seven segment display.
- The decoded data is stored in the lower 8 bits of destination device [D.]. The upper 8 bits was unchanged.

(S.)		Seven segment display	(D.)								data
Hex	Bit		b7	b6	b5	b4	b3	b2	b1	b0	
0	0000		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	0	0	0	1	F

◎ Seven Segment With Latch

FNC(74)			16 bits: SEGL(P) - 5 steps
	SEGL	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Reserved

◎ Arrow Switch

FNC(75)			16 bits: ARWS(P) -9 steps
	ARWS		

EX	EX_{1S}	EX_{1N}	EX_{2N}

Reserved

◎ Ascii Code Conversion

FNC(76)			16 bits: ASC -11 steps
	ASC		

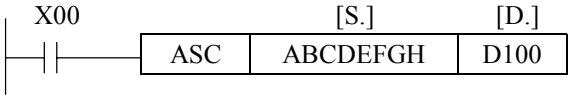
EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S.]: 8 character or alphanumeric data.

Operands:

← [D.] →			
	T	C	D

Flag:



♦ The source data string [S.] consists of up to 8 characters.

♦ The character ”A”~”H” is converted to ASCII codes, then stored into D100~D103.

When M8161 is OFF

M8161=OFF	Upper 8 bits	Lower 8 bits
D100	“B”	“A”
D101	“D”	“C”
D102	“F”	“E”
D103	“H”	“G”

When M8161 is ON

	Upper 8	Lower 8		Upper 8	Lower 8
D100	0	“A”	D104	0	“E”
D101	0	“B”	D105	0	“F”
D102	0	“C”	D106	0	“G”
D103	0	“D”	D107	0	“H”

◎ Print

FNC(77)			16 bits: PR - 5 steps
	PR		

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

← [S.] →			
	T	C	D

Operands: [D.]: Y

Reserved

◎ FROM

FNC(78)			16 bits: FROM(P) - 9 steps
D	FROM	P	32 bits: (D)FROM(P) -17 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

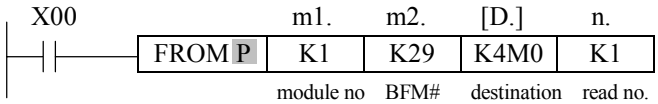
Operands: [D.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: m1 = 0 ~ 7 no. of special module

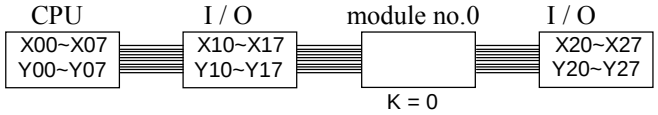
m2.= 0 ~ 31 no. of buffer memory (BFM)

n.= 1 ~ 32 no. of read (when D, n=1~16)



♦ When X00 ON, the buffer memory of special module BFM#29 to be read and stored into M00~M15.

<< Special Device Module Number m1>>



♦ The BFM is the memory address of special module.

♦ The number of special module is address to NO.0~NO.7 and beginning with the one closest to the CPU unit.

♦ The special module can up to 8 maximum, and no occupy I/O points.

◎ TO

FNC(79)			16 bits: TO(P) - 9 steps
D	TO	P	32 bits: (D)TO(P) - 17 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

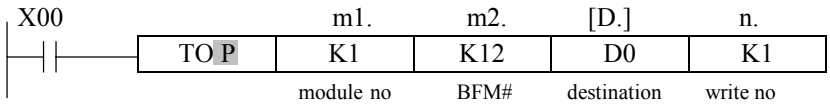
Operands: [D.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: m1 = 0 ~ 7 no. of special module

m2.= 0 ~ 31 no. of buffer memory (BFM)

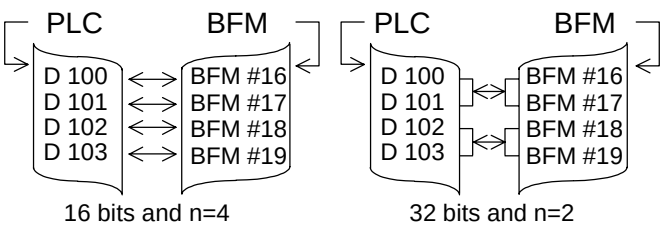
n.= 1 ~ 32 no. of write (when D, n=1~16)



♦ When X00 ON, the content of D0 to be write into the buffer memory BFM#12 of the special module NO.1

♦ If used pulse command can decrement cycle time.

<< Number of Read n >>



◎ Communication

FNC(80)	16 bits: RS - 9 steps
RS	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

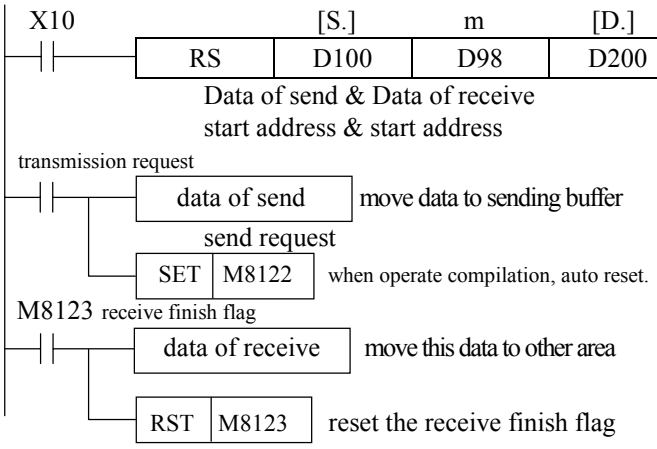
←S, →								
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
←→	m,n=1~128						←→	D. m,n

Flag:

<< Communication Format >> D8120

	Content	0	1
Bit0	Data length	7 bit	8 bit
Bit1	Parity	(00):none, (01):odd, (11):even	
Bit2			
Bit3	Stop Bit	1 bit	2 bit
Bit4	Baud rate (bps)	(0011):300, (0100):600	
Bit5		(0101):1200, (0110):2400	
Bit6		(0111):4800, (1000):9600	
Bit7			
Bit8	Start 1	None	D8124
Bit9	End 1	None	D8125
Bit10	Reserved		
Bit11	Reserved		
Bit12	End 2	None	D8126
Bit13	RS Mode	User define	ModBus
Bit14	ModBus Mode	Ascii Mode	RTU Mode
Bit15	Reserved		

- EXPLC use EXADP232/422/485 communicate board to execute transmitting and receiving data. The protocol is assigned by special auxiliary D8120.
- When RS executing, changing data of D8120 is not affect current operation.
- The communicate port of EXPLC can be as master unit or slaver unit, so once RS execute, then enable. the function of communication and waiting trigger signal.



- The protocol and data frame all defined by user and can be selected different communicate interface board, so EXPLC can communicate with the other kind machine (inverter, bar code machine …).

<< Request of transmission >> M8122

♦ When the transmit request flag M8122 to be driven in the waiting communicate status, then PLC will transmit from the head address of D100 for D98 number of bytes to slaver, and M8122 will auto reset after transmit completed.

<< Receive Finish Flag >> M8123

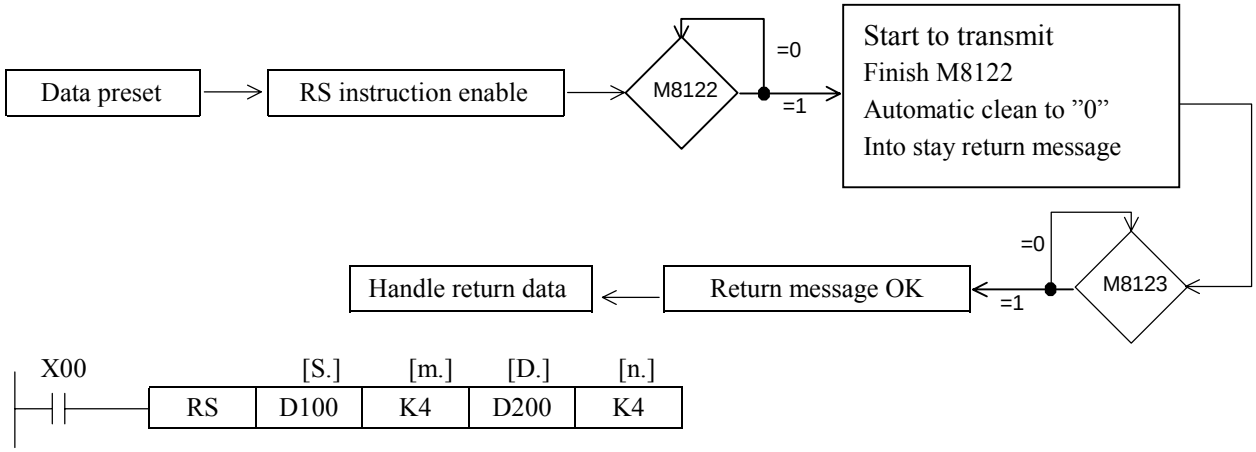
♦ When PLC finish to receive data, receive finish flag M8123 will set to “1”, user can use program to reset it.

<< Carrier Detect Flag >> M8124

Reserved

<< Associate Parameter >>

<< RS instruction formula >>



< 8 Bits Mode > M8161=ON is 8 bits operation

	16 bits	
	ignore	Lower 8 bits

start	data of transmitting (m)				end
STX	D100 down	D101 down	D102 down	D103 down	ETX
start	data of receiving (n)				end
STX	D200 down	D201 down	D202 down	D203 down	ETX

< 16 Bits Mode > M8161=OFF is 16 bits operation

	16 bits	
	Up 8 bits	Down 8bits

start	data of transmitting (m)				end
STX	D100 down	D100 up	D101 down	D101 up	ETX
start	data of receiving (n)				end
STX	D200 down	D200 up	D201 down	D201 up	ETX

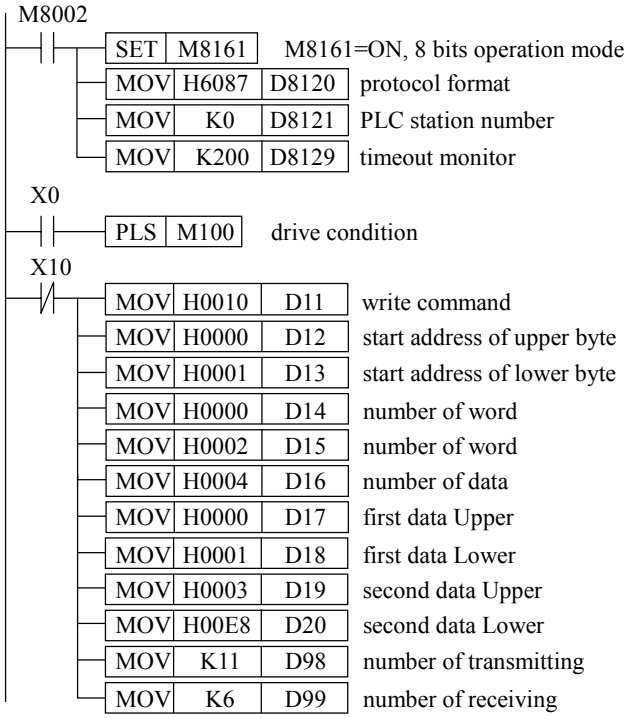
♦ If error occurrence was in the communication, then error flag M8063 to be set and error code in the D8063.

♦ If RS instruction is used, then can’t use PRUN instruction.

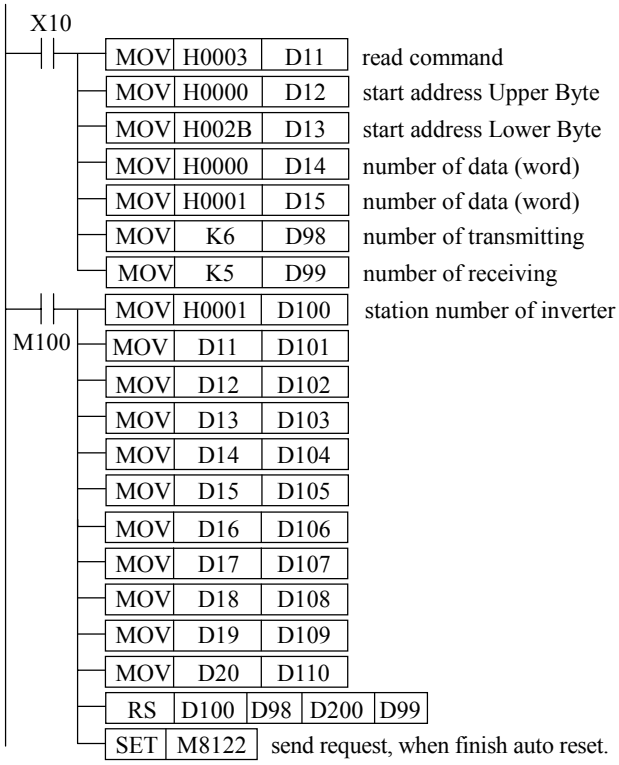
♦ About Example description, refer to Application Note.

5.1. << MODBUS RTU >> CRC Error Check Mode

Ex: application note with type of YASKAWA VS-606V7 (The protocol can reference to YASKAWA's user manual)



♦ The data of CRC, the PLC will auto calculated it.

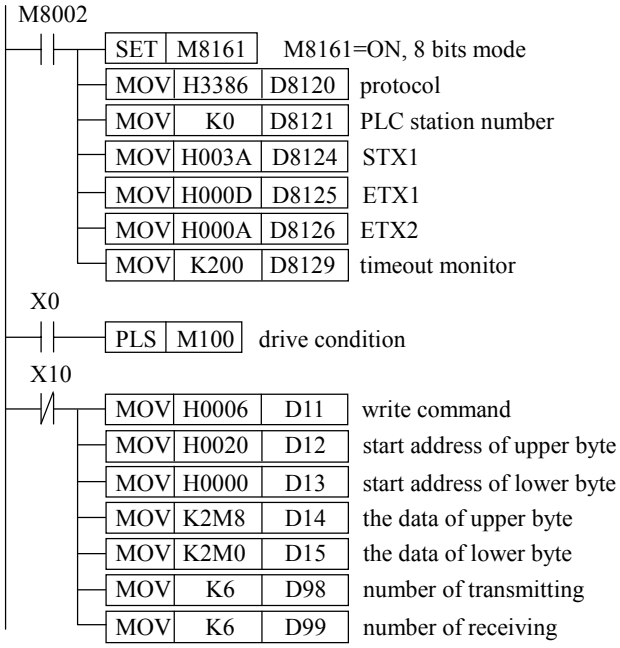


♦ At ModBus RTU mode, the number of communication must be set correct and the format have to be STX/ETX.

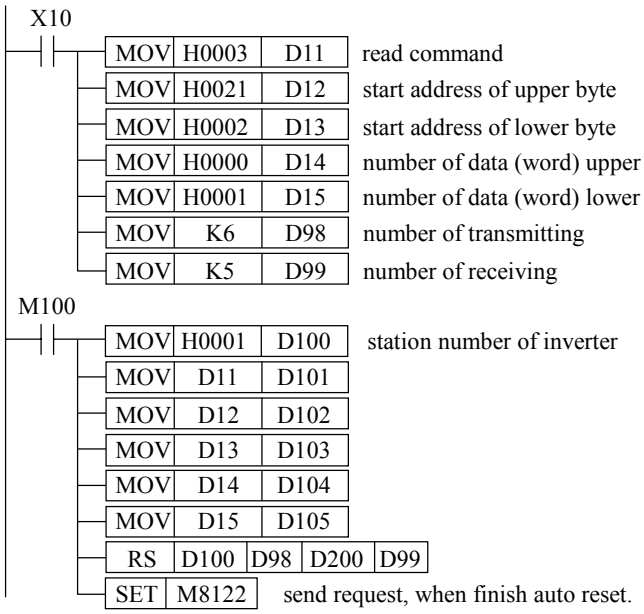
♦ The data of error check is not included in the RS number.

5.2. << MODBUS ASCII >> LRC Error Check Mode

Ex: application note with type of DELTA-S inverter (The protocol can reference to DELTA's user manual)



- ♦ The data of LRC, the PLC will auto calculated it.
- ♦ The data of error check is not included in the RS number.



- ♦ At ModBus Ascii mode, the number of communication must be set correct and the format have to be STX/ETX.
- ♦ When complete to receive, flag M8123 will auto set to "1", so if want to continue monitor the status of inverter, just use M8123 as drive condition.

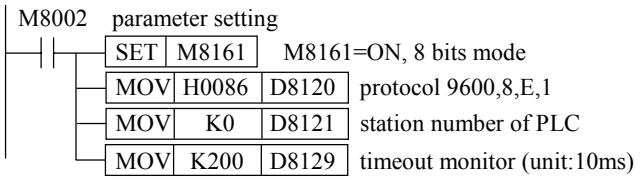
5.3. << Other Mode >> User Defined Error Check

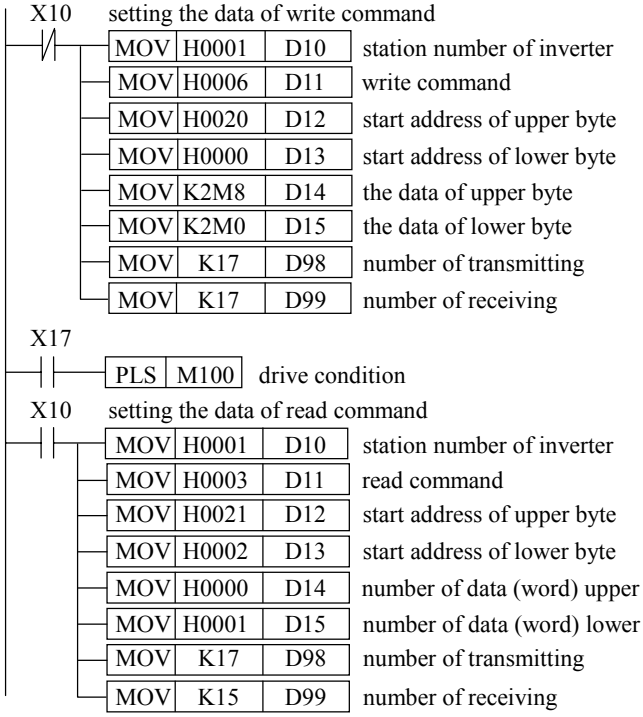
Ex: application note with type of DELTA-S inverter (The protocol can reference to DELTA's user manual)

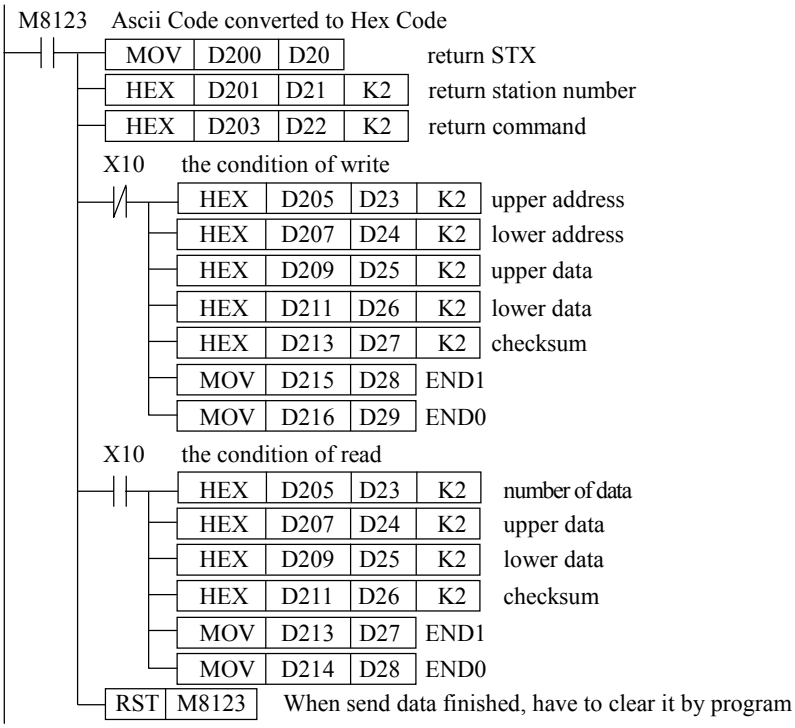
Tx	STX	ADR 1	ADR 0	CMD 1	CMD 0	start address				N.O.W				LRC 1	LRC 0	END 1	END 0
	':'	'0'	'1'	'0'	'3'	'2'	'1'	'0'	'2'	'0'	'0'	'0'	'1'	'D'	'8'	CR	LF

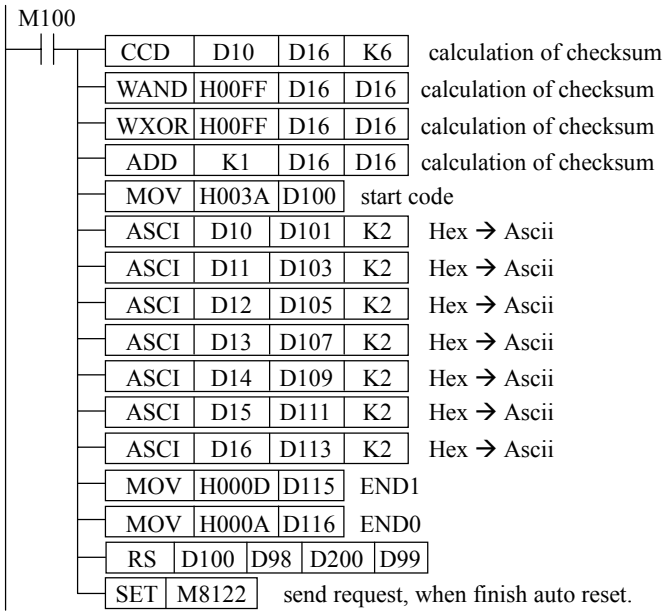
Rx	STX	ADR 1	ADR 0	CMD 1	CMD 0	N.O.B (byte)		data				LRC 1	LRC 0	END 1	END 0	
	':'	'0'	'1'	'0'	'3'	'0'	'2'	'0'	'3'	'E'	'8'	'0'	'F'	CR	LF	

- The number of transmitting, from STX to END0 total 17.
- The number of receiving, from STX to END0 total 15.
- The number of communication is included STX and END.









♦ At this mode, the data value of error check, PLC can't auto calculate.

◎ Parallel Running

FNC(81)			16 bits: PRUN(P) - 5 Steps
D	PRUN	P	32 bits: (D)PRUN(P) - 9 Steps

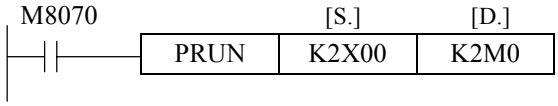
EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S.]: KnX, KnM the lowest bit device is ”0”

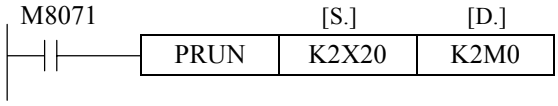
[D.]: KnM, KnY the lowest bit device is ”0”

Flag: M8073, M8129

Master program M8070=1, [S.] [D.] is pseudo operand



Slaver program M8071=1, [S.] [D.] is pseudo operand



- ♦ The content of D490~D497 of the master will transmit to D490~D497 of the slaver (M8070=1).
- ♦ The content of D500~D507 of the slaver will transmit to D500~D507 of the master (M8070=0).
- ♦ This instruction just set the status of M8070 and M8071, don’t need to assign data register (D), then will auto communicate.
- ♦ Because only the data register communicate each other, just used MOV to execute conversion, then input relay of master can control the output relay of slaver, and the input relay of slaver can control the master.
- ♦ Relative parameter

  M8122: start communication transmitted flag.

  M8123: receive finished flag

  M8070: master flag

  M8071: slaver flag

  M8129: sum check error flag

  M8073: overtime flag

  D8070: overtime register(ms)

  D8072: communication taking time(ms)
- ♦ Example program please refer to EXPLC Application Note F081。
- ♦ When PRUN instruction used, then can’t use RS instruction.

◎ Hex To Ascii Conversion

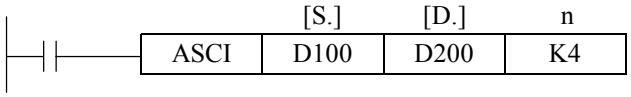
FNC(82)			16 bits: ASCI(P) - 7 Steps
	ASCI	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

					[S.]			
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
n = (1 – 32)					[D.]			

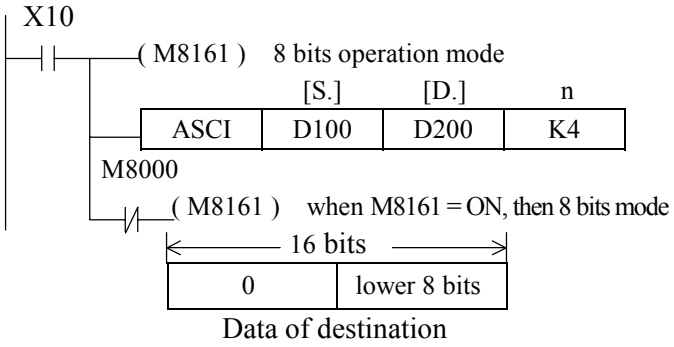
Flag:



- The hexadecimal data of source [S.] to be converted ASCII code and stored into upper/lower byte of destination device [D.] for n number of bytes.
- When M8161=OFF, 16 bits operation mode.

example: (D100)=0ABCH, (D101)=1234H

	K1	K2	K3	K4	K5	K6	K7	K8
D200 down	“C”	“B”	“A”	“0”	“4”	“3”	“2”	“1”
D200 up		“C”	“B”	“A”	“0”	“4”	“3”	“2”
D201 down			“C”	“B”	“A”	“0”	“4”	“3”
D201 up				“C”	“B”	“A”	“0”	“4”
D202 down					“C”	“B”	“A”	“0”
D202 up						“C”	“B”	“A”
D203 down							“C”	“B”
D203 up								“C”



- The hexadecimal data of source [S.] to be converted ASCII code and stored into lower byte of destination device [D.] for n number of bytes.
- When M8161=ON, 8 bits operation mode.

example: (D100)=0ABCH, (D101)=1234H

	K1	K2	K3	K4	K5	K6	K7	K8
D200 down	“C”	“B”	“A”	“0”	“4”	“3”	“2”	“1”
D201 down		“C”	“B”	“A”	“0”	“4”	“3”	“2”
D202 down			“C”	“B”	“A”	“0”	“4”	“3”
D203 down				“C”	“B”	“A”	“0”	“4”
D204 down					“C”	“B”	“A”	“0”
D205 down						“C”	“B”	“A”
D206 down							“C”	“B”
D207 down								“C”

◎ Ascii To Hex Conversion

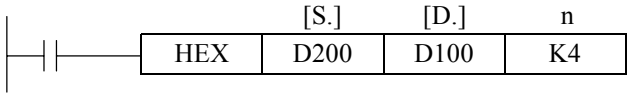
FNC(83)			16 bits: HEX(P) - 7 Steps
	HEX	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S.]								
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
n = (1 – 32)					[D.]			

Flag:

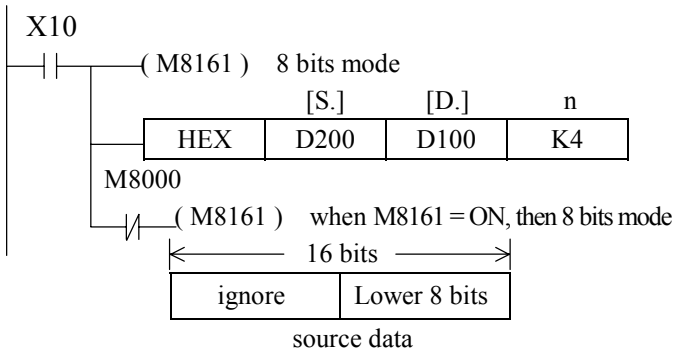


♦ The ASCII code of the upper/lower byte in source [S.] to be converted to the hexadecimal data and stored into the destination device [D.] for n number byte.

♦ When M8161=OFF, 16 bits operation mode.

Ex.: D200 down =”0”, D200 up =”A”, D201 down =”B”, D201 up =”C”
D202 down =”1”, D202 up =”2”, D203 down =”3”, D203 up =”4”

	D102	D101	D100
K1			0H
K2			0AH
K3			0ABH
K4			0ABCH
K5		0H	ABC1H
K6		0AH	BC12H
K7		0ABH	C123H
K8		0ABCH	1234H



♦ The ASCII code of the lower byte in source [S.] to be converted to the hexadecimal data and stored into the destination device [D.] for n number byte.

♦ When M8161=ON, 8 bits operation mode.

Ex: D200=”0”, D201=”A”, D202=”B”, D203=”C”
D204=”1”, D205=”2”, D206=”3”, D207=”4”

	D102	D101	D100
K1			0H
K2			0AH
K3			0ABH
K4			0ABCH
K5		0H	ABC1H
K6		0AH	BC12H
K7		0ABH	C123H
K8		0ABCH	1234H

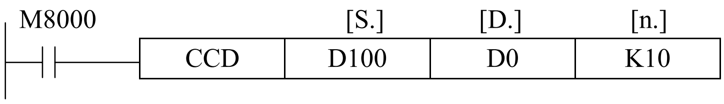
◎ Check Code

FNC(84)		16 bits: CCD(P) - 7 Steps
CCD	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

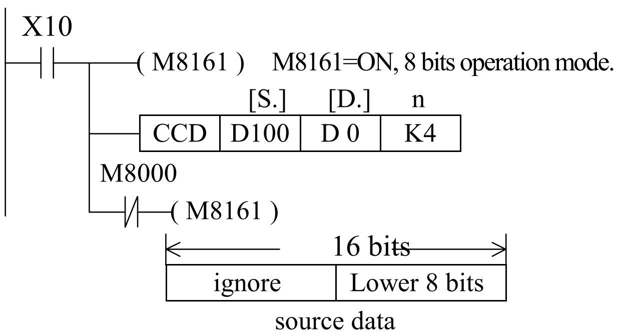
Operands: [S1.][S2.]

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z



♦Calculation the data of n bytes (16 bits) from the head address of source [S.], then put the Sum→D00, Vertical Parity→D01([D.]+1).

M8161=OFF 16 bit mode									
(S.)		Bit Pattern							
D100 L	K100	0	1	1	0	0	1	0	0
D100 H	K111	0	1	1	0	1	1	1	1
D101 L	K100	0	1	1	0	0	1	0	0
D101 H	K98	0	1	1	0	0	0	1	0
D102 L	K123	0	1	1	1	1	0	1	1
D102 H	K66	0	1	0	0	0	0	1	0
D103 L	K100	0	1	1	0	0	1	0	0
D103 H	K95	0	1	0	1	1	1	1	1
D104 L	K210	1	1	0	1	0	0	1	0
D104 H	K88	0	1	0	1	1	0	0	0
Vertical parity		1	0	0	0	0	1	0	1
Sum	K1091								



♦Calculation the data of n bytes (8 bits) from the head address of source [S.], then put the Sum→D00, Vertical Parity→D01([D.]+1).

M8161=ON 8 bit mode									
(S.)		Bit Pattern							
D100	K100	0	1	1	0	0	1	0	0
D101	K111	0	1	1	0	1	1	1	1
D102	K100	0	1	1	0	0	1	0	0
D103	K98	0	1	1	0	0	0	1	0
D104	K123	0	1	1	1	1	0	1	1
D105	K66	0	1	0	0	0	0	1	0
D106	K100	0	1	1	0	0	1	0	0
D107	K95	0	1	0	1	1	1	1	1
D108	K210	1	1	0	1	0	0	1	0
D109	K88	0	1	0	1	1	0	0	0
Vertical parity		1	0	0	0	0	1	0	1
SUM	K1091								

◎ Volume Read

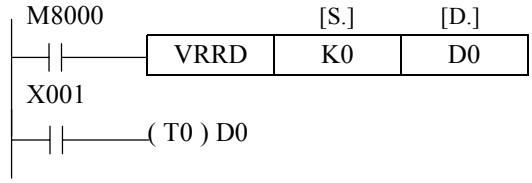
FNC(85)		16 bits: VRRD(P) - 5 Steps
	VRRD P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[D.]								
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[S.] = (0 – 3)



♦ The identified volume [S.] of the master unit is read as an analog input and converted to 8 bits binary code (0-255) stored into the destination device [D.].

♦ The content of [D.] can as Timer data or Counter data.

◎ Volume Scale

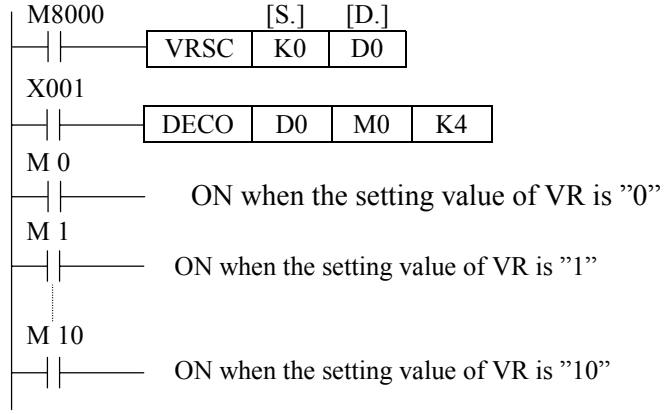
FNC(86)		16 bits: VRSC(P) - 5 Steps
	VRSC P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[D.]								
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[S.] = (0 – 3)



♦ The identified volume [S.] of the master unit is read as an analog input and converted to 8 bits binary code (0-255) then divided 16, the result (0-15) stored into the destination device [D.].

♦ This function the volume can as a 16 (0-15) position rotary switch.

◎ PID

FNC(88)	16 bits: PID - 9 Steps
PID	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S1.][S2.][S3.]

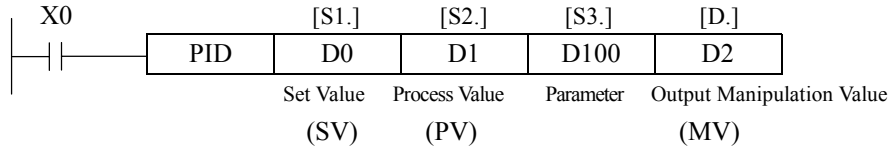
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]

Operands:

X	Y	M	S

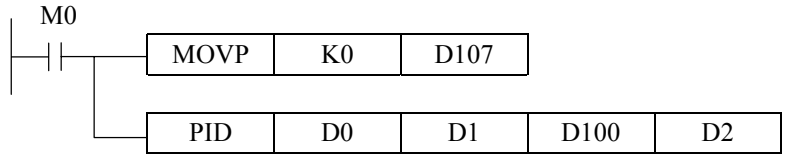
Flag:



[S1.] : Set Value
[S2.] : Process Value
[S3.] ~ [S3.]+6 : Control Parameter
} Use setting execute program as left mentioned, and stored the result (MV) into [D]
[D.] : Output manipulation value data register

♦ It will occupy 25 devices continuous from assigned [S3.]. In this example, it occupy D100 ~ D124.

♦ When execute in first time, have to clear the content of [S3.]+7 to be 0.



♦ Before execute PID operation, have to use MOV command to write the parameter set value for PID control first.

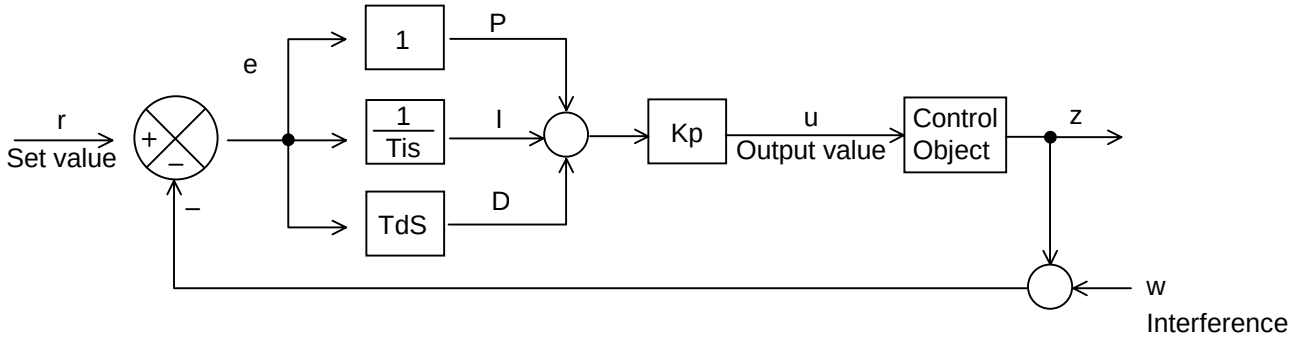
[S3.]	Sampling Time (Ts)	1~32767 (ms) (can't set shorter than scan-time)
[S3.] + 1	Act direction (ACT)	BIT0 : 0 : forward action ; 1 : reverse action BIT1 : 0 : Without input change Alarm ; 1 : With input change Alarm BIT2 : 0 : Without output change Alarm ; 1 : With output change Alarm BIT3 : reserved BIT4 : reserved BIT5 : 0 : Without output limit ; 1 : With output limit BIT6 ~ BIT15 : reserved
[S3.] + 2	Input Filter (α)	0 ~ 99 (%)
[S3.] + 3	Proportion Constant (Kp)	1 ~ 32767 (%)
[S3.] + 4	Integral Time Constant (Ti)	1 ~ 32767 (x 100ms), 0 is without integral action
[S3.] + 5	Derivative Filter Constant (Kd)	0 ~ 100 (%)
[S3.] + 6	Time Derivative Constant (Td)	1 ~ 32767 (x 10ms), 0 is without derivative action
[S3.] + 7 ~ [S3.] + 19	} For internal operation when execute PID	
[S3.] + 20	System reserved	
[S3.] + 21	System reserved	
[S3.] + 22	Output maximum value limitation, it is effective when [S3.]+1, BIT5=1	
[S3.] + 23	Output minimum value limitation, it is effective when [S3.]+1, BIT5=1	
[S3.] + 24	System reserved	

♦ Basic operation of PID instruction :

This instruction is based on speed form, measure Derivative calculation formula to execute PID operation. In PID control, execute operation formula of forward action or reverse action according to the content of "Act direction" which is assigned by [S3.].

PID basic formula:

Output $u(t) = Kp \{ e(t) + \frac{1}{Ti} \int_0^t e(t)dt + Td \frac{de(t)}{dt} \}$ e(t) = error value



◎

FNC(89)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(90)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(91)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(92)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(93)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(94)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(95)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(96)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(97)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(98)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎

FNC(99)				

EX	EX_{1S}	EX_{1N}	EX_{2N}

◎ Floating Point Compare

FNC(110)			
D	ECMP	P	32 bits:(D)ECMP & (D)ECMP(P) - - - - - - - - - - 13 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.]

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

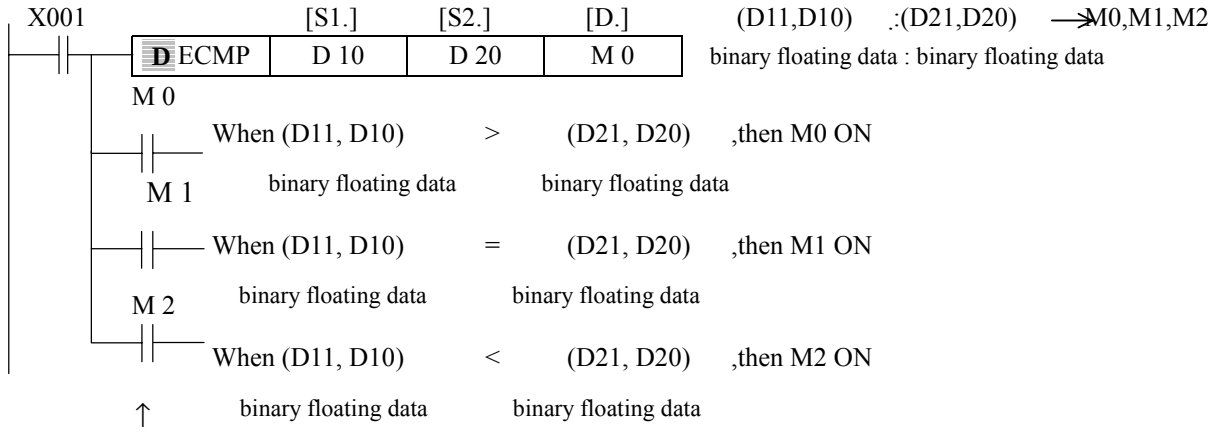
[S2.]

Operands: [D.]

X	Y	M	S

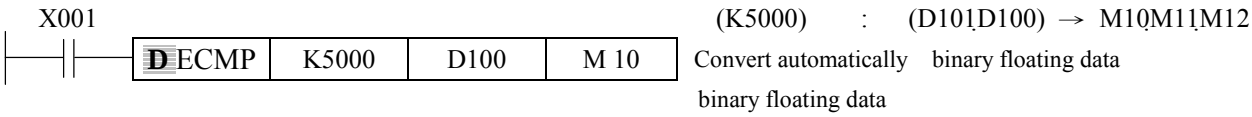
The result is indicated by 3 bit devices specified with the head address entered as D.

Flag:M8020, M8021, M8022



When X001 OFF, then not execute ECMP, M0~M2 status unchanged

- ♦ Compare the binary floating data of the source devices [S1.] and [S2.], this will automatic ON/OFF 3 bit devices from the head address of [D.].
- ♦ When source operand assigned by constant K or H, it will be converted to binary floating data automatically.



◎ Floating Point Zone Compare

FNC(111)			
D	EZCP	P	32 bits:(D)EZCP & (D)EZCP(P) - - - - - - - - - - - 17 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.] [S2.] [S.] [S1.] [S2.] [S.]

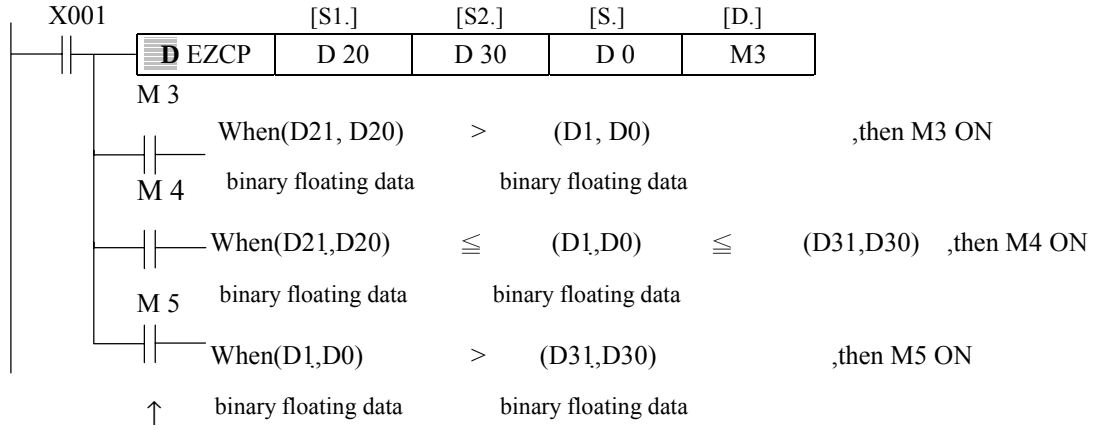
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: [D.]

X	Y	M	S

[D.] occupy 3 bit devices from the head address,[S1.]、[S2.]set [S1.]≦[S2.]

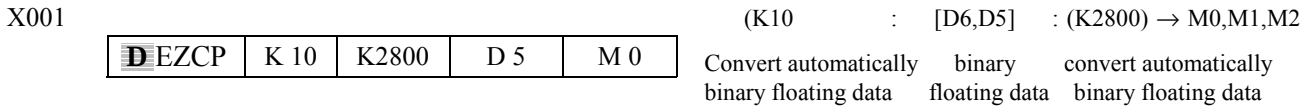
Flag:M8020, M8021, M8022



If X001OFF, then not execute ECMP, M3~M5 status unchanged.

♦ The result will automatically set 3 bit devices from the head address of [D.]

♦ When source operand assigned by constant K or H, it will be converted to binary floating data automatically



♦ Set [S1.] ≦ [S2.], if [S1.] > [S2.], then data of [S2.] is as same as data of [S1.].

◎ Float to Scientific conversion

FNC(118)			
D	EBCD	P	32 bits:(D)EBCD & (D)EBCD(P) - - - - - - - - - - - 9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

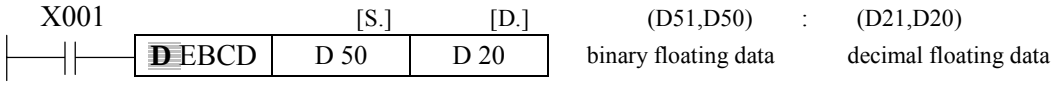
K,H	KnX	KnY	KnM	KnS	T	C	D [S.]	V,Z

D [D.]

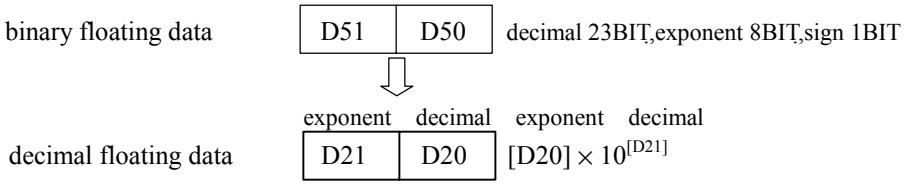
Operands:

X	Y	M	S

Flag:



♦ Convert binary floating data assigned by source device to decimal floating data, and store it to destination device.



♦Reserved

◎ Scientific to Float conversion

FNC(119)			
D	EBIN	P	32 bits:(D)EBIN & (D)EBIN(P) - - - - - - - - - - - 9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

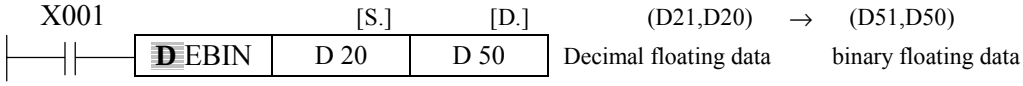
K,H	KnX	KnY	KnM	KnS	T	C	D [S.]	V,Z

D [D.]

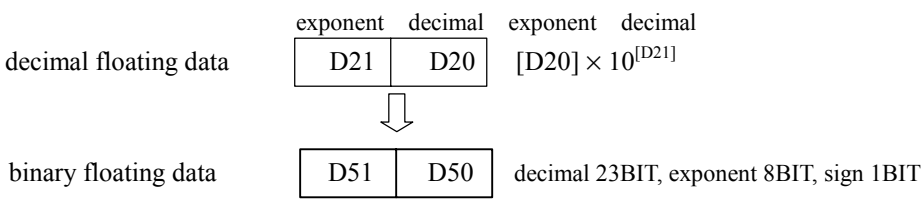
Operands:

X	Y	M	S

Flag:



♦Convert binary floating data assigned by source device to decimal floating data, and store it to destination device.



♦ Reserved

◎ Floating Point Addition

FNC(120)			
D	EADD	P	32 bits:(D)EADD & (D)EADD(P) - - - - - - - - - - 13 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

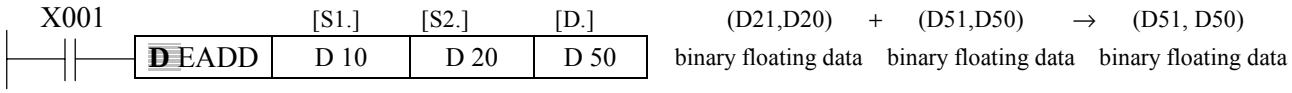
Operands:

[S1.] [S2.]					[S1.] [S2.]		[D.]	
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands:

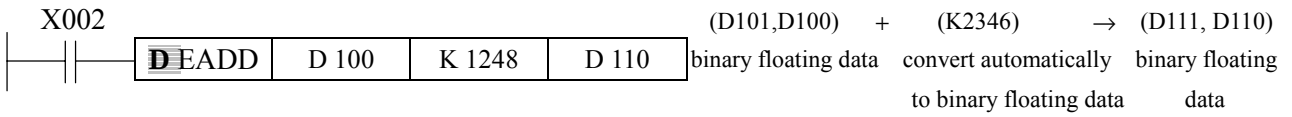
X	Y	M	S

Flag: None



♦ Two devices of binary floating data are added, then the result stored by form of binary floating data at destination device.

♦ When source operand assigned by constant K or H, it will be converted to binary floating data automatically.



♦ Enable assign source operand [S.] and destination operand [D.] to same device number.

◎ Floating Point Subtraction

FNC(121)			
D	ESUB	P	32 bits:(D)ESUB & (D)ESUB(P) - - - - - - - - - - 13 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

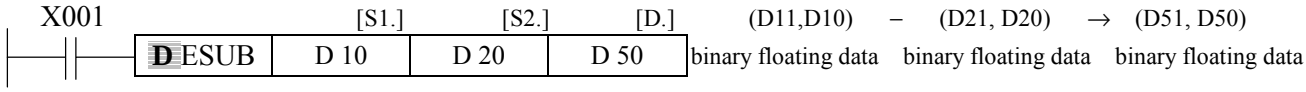
Operands:

[S1.] [S2.]					[S1.] [S2.]		[D.]	
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands:

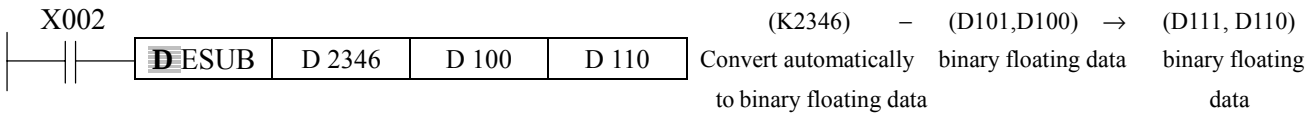
X	Y	M	S

Flag: None



♦ Binary floating data of [S1.] subtract binary floating data of [S2.], then the result stored by form of binary floating data at destination device of [D.].

♦ When source operand assigned by constant K or H, it will be converted to binary floating data automatically.



♦ Enable assign source operand [S.] and destination operand [D.] to same device number.

◎ Floating Point Multiplication

FNC(122)			32 bits:(D)EMUL & (D)EMUL(P) - - - - - - - - - 13 steps
D	EMUL	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S1.]	[S2.]				[S1.]	[S2.]	[D.]	
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands:

X	Y	M	S

Flag: None

X001 [S1.] [S2.] [D.] (D11,D10) × (D21, D20) → (D51, D50)

D EMUL	D 10	D 20	D 50

binary floating data × binary floating data → binary floating data

♦ Two source devices, binary floating data of [S1.] is multiplied by binary floating data of [S2.], then the result stored by form of binary floating data at destination device of [D.].

♦ When source operand assigned by constant K or H, it will be converted to binary floating data automatically.

X002 (K2346) × (D101,D100) → (D111, D110)

D EMUL	K 2346	D 100	D 110

Convert automatically to binary floating data × binary floating data → binary floating data

◎ Floating Point Division

FNC(123)			32 bits:(D)EDIV & (D)EDIV(P) - - - - - - - - - - - 13 steps
D	EDIV	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S1.]	[S2.]				[S1.]	[S2.]	[D.]	
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands:

X	Y	M	S

Flag: None

X001 [S1.] [S2.] [D.] (D11,D10) ÷ (D21, D20) → (D51, D50)

D EDIV	D 10	D 20	D 50

binary floating data ÷ binary floating data → binary floating data

♦ The binary floating data of assignation device [S1.] is divided by binary floating data of assignation device [S2.], then the result stored by form of binary floating data at destination device of [D.].

♦ When source operand assigned by constant K or H, it will be converted to binary floating data automatically.

X002 (D101,D100) ÷ (K100) → (D111, D110)

D EDIV	D 100	K 100	D 110

binary floating data ÷ convert automatically to binary floating data → binary floating data

◎ Floating Point Square Root

FNC(127)			32 bits:(D)ESQR & (D)ESQR(P) - - - - - - - - - - 13 steps
D	ESQR	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S.]							[S.]	
K,H	KnX	KnY	KnM	KnS	T	C	D	Z
							[D.]	

Operands:

X	Y	M	S

The content of [S.] is positive number, then effective

Flag: M8020

X001 [S.] [D.] $\sqrt{(D11,D10)}$ → (D21, D20)

DESQR	D 10	D 20

binary floating data → binary floating data

- To be square root of binary floating data of source device [S.],then the result stored by binary floating data at destination device of [D.].
- When source operand assigned by constant K or H, it will be converted to binary floating data automatically.

X002 $\sqrt{K1024}$ → (D111, D110)

DESQR	k1024	D110

Convert automatically to binary floating data → binary floating data

- If the result is "0", then zero flag M8020 will ON.
- The content of source operand is effective only when it is positive. If the number is negative, then error flag M8067 will ON and stop executing.

◎ Float to Integer

FNC(129)			16 bits:INT & INT - 5 steps
D	INT	P	32 bits:(D)INT & (D)INT(P) - - - - - - - - - - - - - - -9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

K,H	KnX	KnY	KnM	KnS	T	C	[S.] D	V,Z
							[D.]	

Operands:

X	Y	M	S

Flag:

X001 [S.] [D.] (D11, D10) → (D20)

INT	D10	D20

binary floating data → BIN integer, remove the number of decimal fraction

X002 [S.] [D.] (D101, D100) → (D111, D110)

DINT	D100	D200

binary floating data → BIN integer, remove the number of decimal fraction

- Convert binary floating data of assigned device [S.] to BIN integer, then store the result at destination device [D.]

♦ When the result is “0”, then zero flag M8020 will ON.

When it converts and becomes 0 because of less than 1, borrow flag M8021 will ON.

If the calculating result more than following limit, then will overflow and carry flag M8022 will ON.

When 16 bit operation**:** -32,768~32,767

When 32 bit operation**:** -2,147,483,648~2,147,483,647

◎ Sine

FNC(130)			
D	SIN	P	32 bits:(D)SIN & (D)SIN(P) - - - - - - - - - - - - - - 9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

K,H	KnX	KnY	KnM	KnS	T	C	D [S.]	V,Z

D [D.]

Operands:

X	Y	M	S

0°≦angle<360°

Flag

X000

	[S.]	[D.]
D SIN	D50	D60

(D51, D50)RAD → (D61, D60)SIN

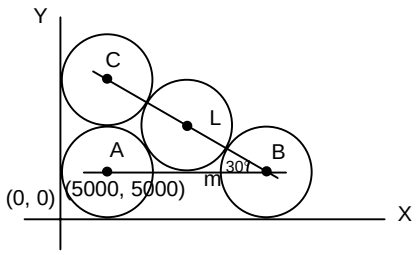
binary floating data → binary floating data

♦ Used assigned angle (RAD) by source [S.] to get SIN value, then store the result at destination device [D.].

[S.]	D 51	D 50
[D.]	D 61	D 60

RAD value (angle x π /180) assign to binary floating data

SIN value binary floating data



M8002

D MOV **P**	K 60	D 0	∠C=60° → (D0) binary integer value

M8000

D FLT	D 0	D 4	Convert ∠C to binary floating value → (D5, D4)
D EDIV	K31415926	K1800000000	D 20 (π /180) → (D21,D20)
D EMUL	D 4	D 20	D 30 (D5, D4)angle × (π /180) → (D31, D30) RAD binary floating value
D SIN	D 30	D 32	(D31, D30)RAD → (D33, D21) SIN binary floating value
D MUL	K 10000	K 2	D 40 The length of Line L is double of diameter
D FLT	D 40	D 42	Convert Line L integer value to binary floating point format
D EMUL	D 42	D 32	D 100 D100 is the binary floating point value of Line m
D INT	D 100	D 200	D200 is the binary integer value of Line m

◎ Cosine

FNC(131)			
D	COS	P	32 bits:(D)COS & (D)COS(P) - - - - - - - - - - - 9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

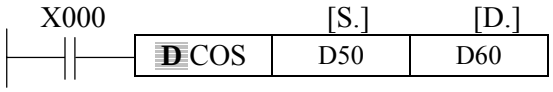
←[S.]→								
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z
							←[D.]→	

Operands:

X	Y	M	S

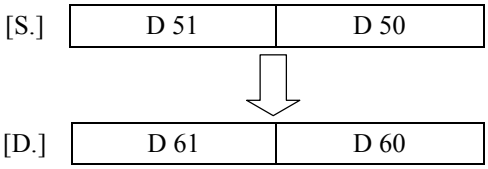
0°≦ angle <360°

Flag:



(D51, D50)RAD → (D61, D60)COS
binary floating data binary floating data

♦ Used assigned angle (RAD) by source device [S.] to get COS value, then store the result at destination device [D.].



◎ Tangent

FNC(132)			
D	TAN	P	32 bits:(D)TAN & (D)TAN(P) - - - - - - - - - - - - - 9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

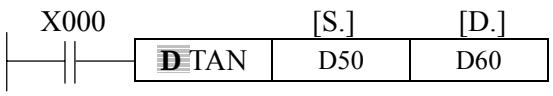
←[S.]→								
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z
							←[D.]→	

Operands:

X	Y	M	S

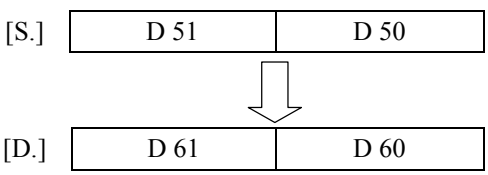
0°≦ angle <360°

Flag:



(D51, D50)RAD → (D61, D60)TAN
binary floating data binary floating data

♦ Used assigned angle (RAD) by source device [S.] to get TAN value, then store the result at destination device [D.].



◎ Byte Swap

FNC(147)			16 bits:SWAP & SWAP(P) - - - - - - - - - - - - - - - - - 5 steps
D	SWAP	P	32 bits:(D)SWAP & (D)SWAP(P) - - - - - - - - -9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S.]

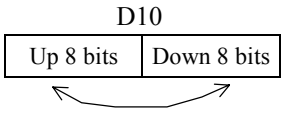
K,H	KnX	KnY	KnM	KnS	T	C	D	Z

Operands:

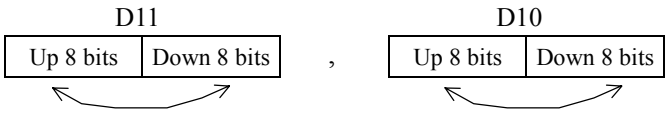
X	Y	M	S

Flag

♦when 16bits, Down 8 bits and Up8 bits exchange.



♦when 32 bits, Up 8 bits and Down 8 bits exchange



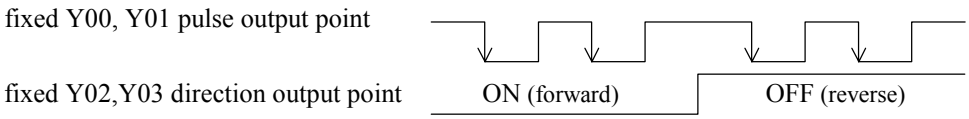
♦ If use continuative executing instruction, each scan cycle will execute to exchange, please pay attention

♦ This instruction is as same as FNC17 (XCH) function of expanded.

◎ FNC150 – 159 Position Control

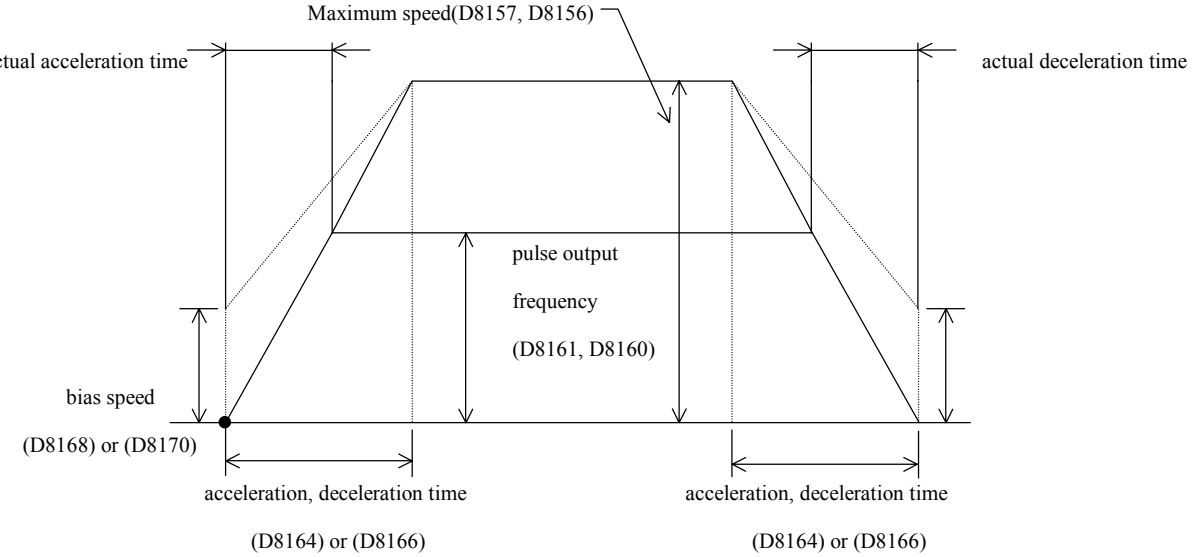
♦ FNC(150~159) with two axes pulse output position control function

♦ The Ex series of controller pulse output signal: pulse (negative logic) + sign, as following drawing



♦ The pulse duty cycle is 50% ON, 50% OFF

♦ Single position control. The curve condition of controller and relative device.



♦ When choose Y00, then special function of X00 input, like high speed counter C235, C241, C244, C246, C247, C249, C254 and interrupt signal I000, I001 can't choose. (except MPG function and zero signal)

♦ When choose Y01, then special function of X01 input, like high speed counter C236, C241, C244, C246, C247, C249, C254 and interrupt signal I100, I101 can't choose again. (except MPG function and zero signal)

◎ Absolute current value read

FNC(155)			16 bits:ABS - 7 steps
D	ABS		32 bits:(D)ABS - 11 steps

EX	**EX_{1S}**	**EX_{1N}**	**EX_{2N}**

Operands: [S.]

K,H	KnX	KnY	KnM	KnS	T	C	D	Z

Operands:

X	Y	M	S

Flag: M8029

Reserved

◎ Zero return

FNC(156)			16 bits:ZRN - 9 steps
D	ZRN		32 bits:(D)ZRN - 17 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.] [S2.]

K,H	KnX	KnY	KnM	KnS	T	C	D	Z

Operands:

X	Y	M	S

Flag: M8029

X10

[S1.]	[S2.]	[S3.]	[D.]
ZRN	K1000	D00	X02	Y00

♦ [S1.] specify zero return speed (Home Speed) 10 ~ 100,000 Hz。
[S2.] specify creep speed 10 ~ 32,767 Hz。
[S3.] specify the Near point signal (A contact), range:X00~X07(pulse catch flag M8170~M8177)。
Servo motor zero signal fixed X00 (Y00), X01(Y01) output point, edge signal be selected, so don't care a, b contact
[D.] specify pulse output point。(Pulse:Y00,Y01, Sign:Y02,Y03)

♦ When execute ZRN, zero-return busy flag M8138 (Y00) or M8139 (Y01) will be set automatically to avoid driving DRVI, DRVA at same time.

♦ Y00 or Y01 only can use once and have to choose transistor output mode.

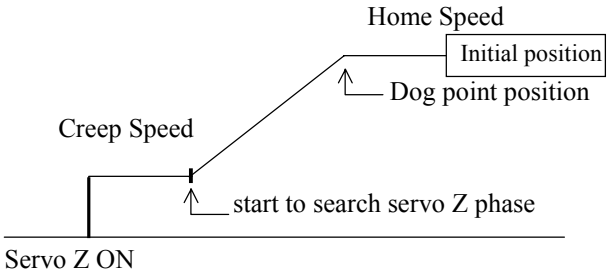
♦ After this instruction executing, acceleration/deceleration time D8164, D8166 data will change to [S2.] data.

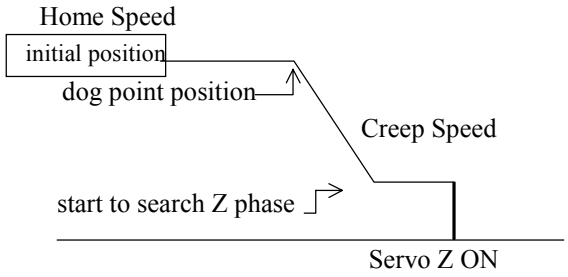
♦ FNC(156)ZRN acceleration/deceleration separate flag M8150 and M8151 ineffective.

♦ Avoid executing zero return initial direction error, Ex1s, Ex1n, Ex2n series offer some relative parameter. User can set it according to machinery characteristics.
When M8158, M8159=0, choose M8156, M8157 to decide zero return initial direction, =0:forward, =1:reverse.
When M8158, M8159=1, choose bias absolute position D8154, D8152 and dog point absolute position D8176, D8178 to compare to decide initial direction.
If (D8155,D8154) > (D8177,D8176), then reverse direction. If (D8155,D8154) < (D8177,D8176), then forward position.

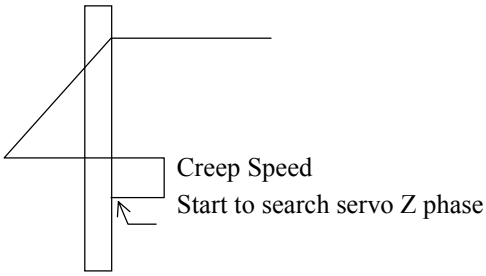
♦ For search dog point speed, system use pulse catch flag (M8170~M8177) to be dog point input point.

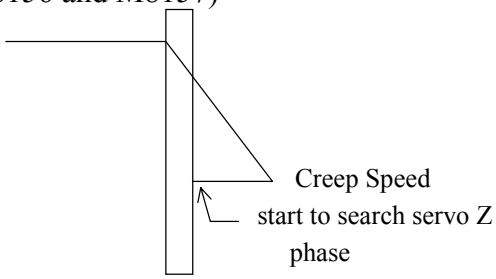
♦ In Ex1s, Ex1n, Ex2n series, there are two modes for zero return (1) forward mode M8154, M8155=1 (2) reverse mode M8154, M8155=0
(1) reverse mode M8156 =1or M8157=1 (When M8158=0 or M8159=0)



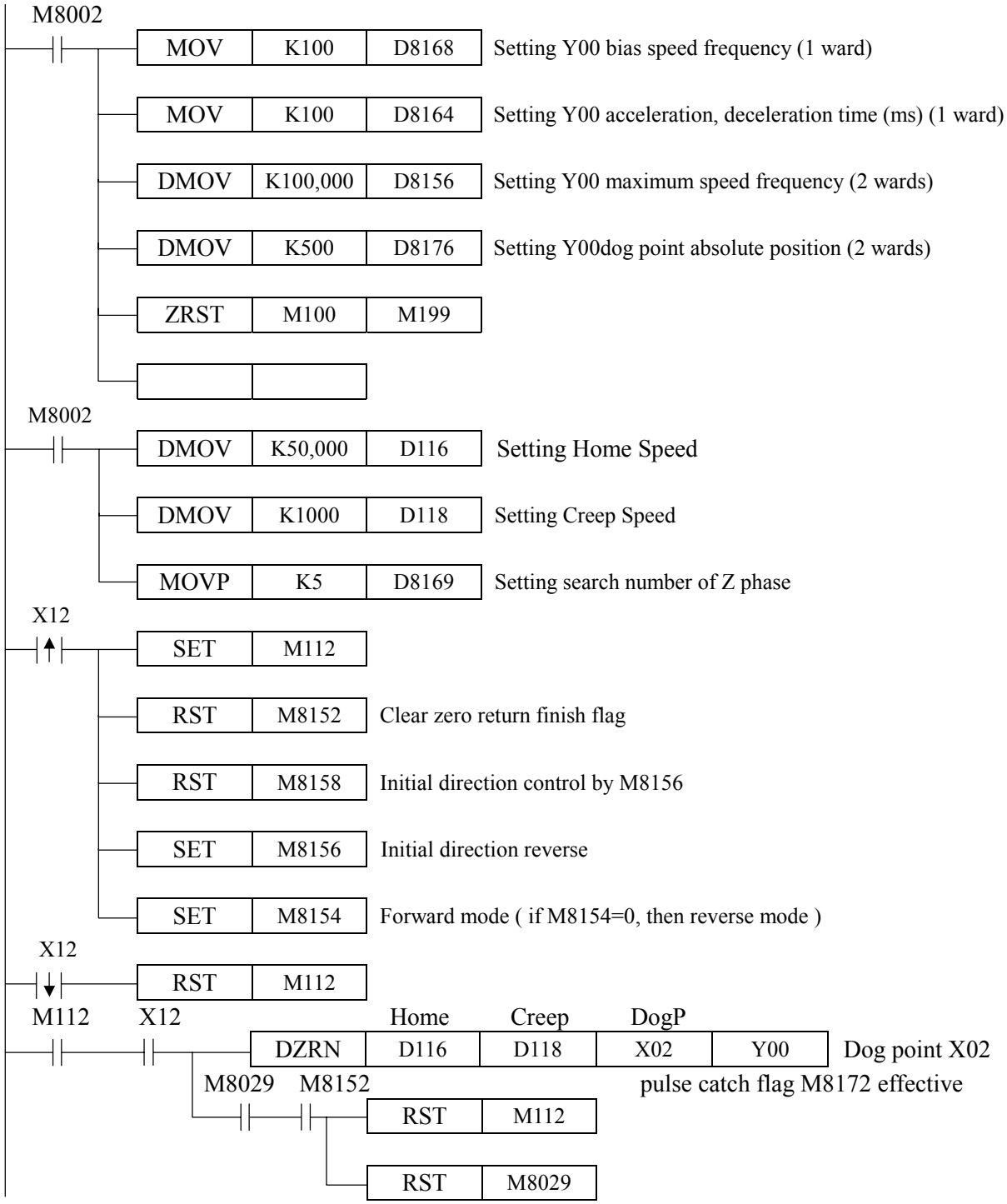


(2) reverse mode (when M8158=1 or M8159=1, Don't Care M8156 and M8157)





◈ Forward mode example (this example initial direction control by flag M8156 or M8157)
initial direction could also choose to control by dog point absolute position.



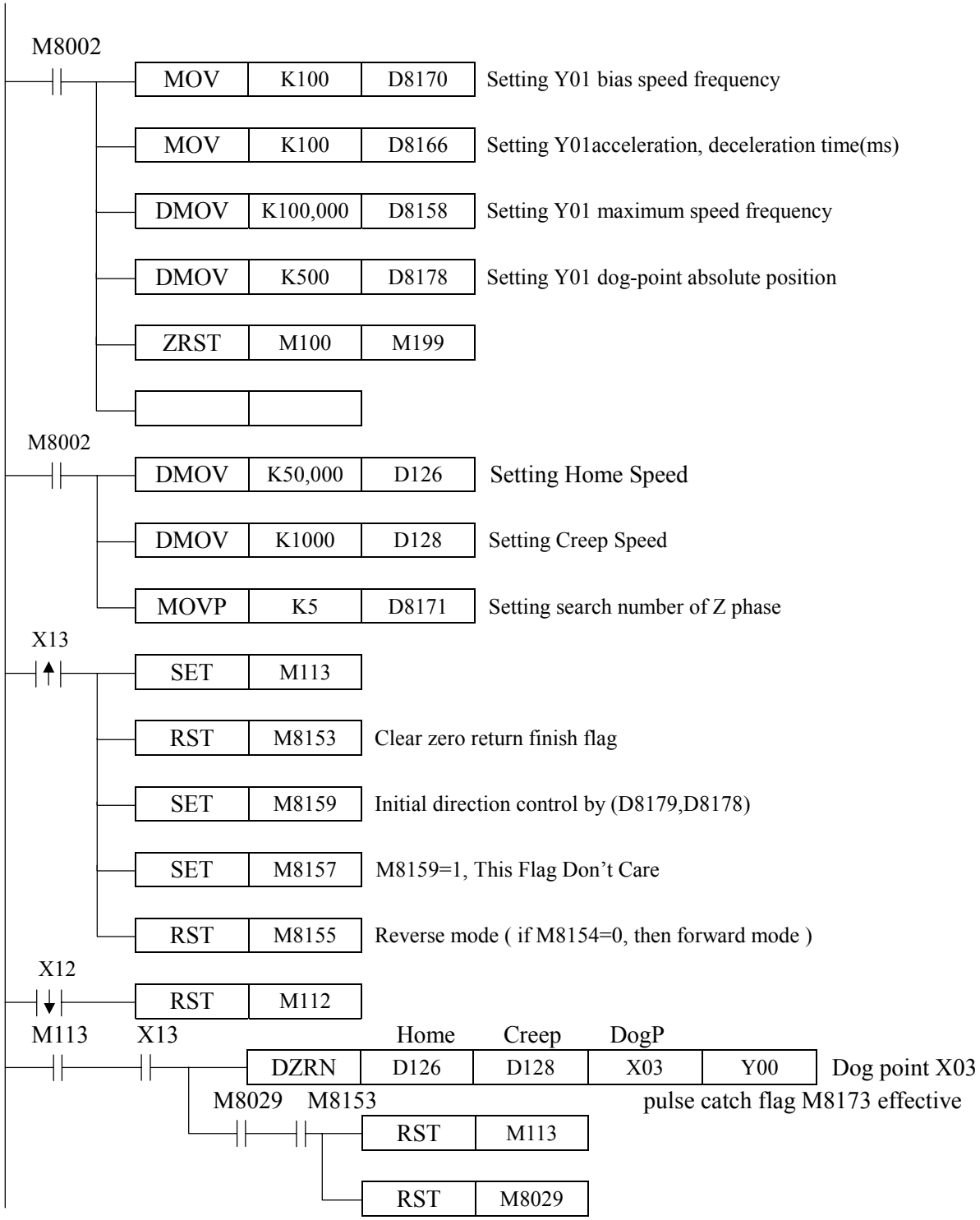
◈ Attention

♦ When execute FNC(156)ZRN, the content of bias speed frequency D8168 or D8170 will change to (Creep) search servo zero speed value.

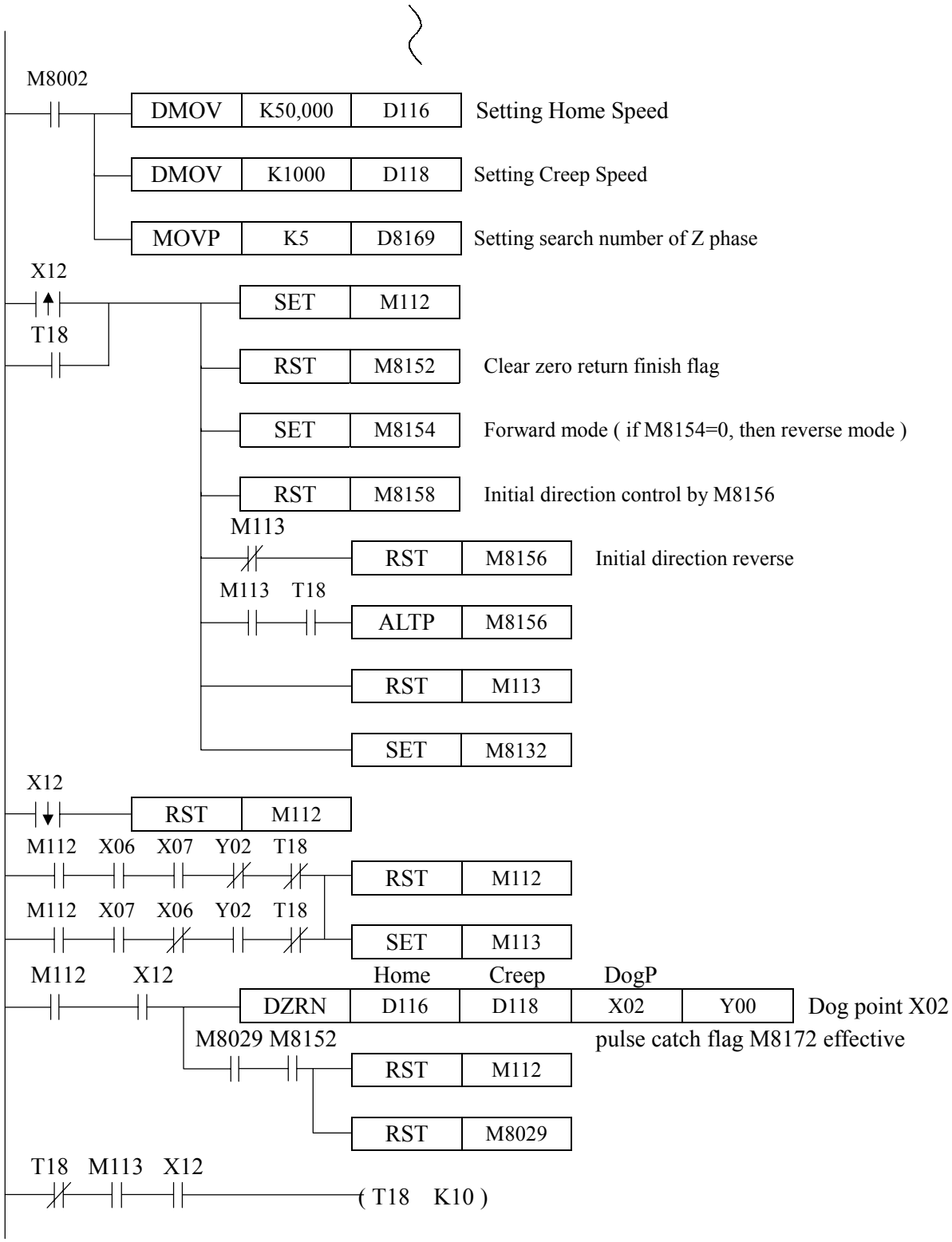
◈ Reverse mode example (initial direction control by dog point absolute position)

initial direction could also choose to control by flag M8156 or M8157

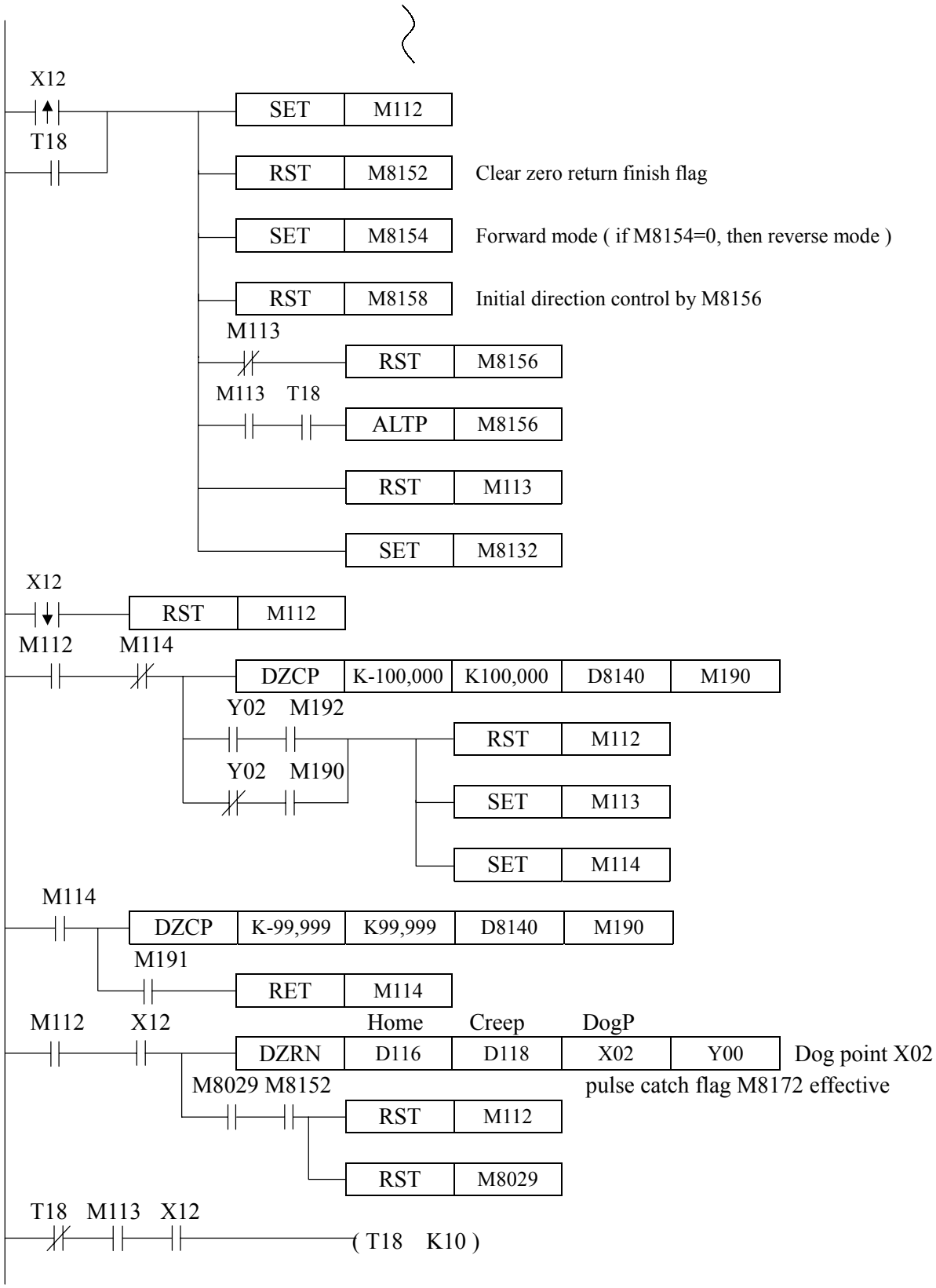
If bias absolute position greater than dog point position, then reverse. If bias absolute position next than dog point position, then forward.



◈ Forward mode touch hardware limited switch, stay 1 second, then reverse example (this example initial direction control by flag M8156 or M8157)



◈ forward mode touch software limited switch, stay 1 second, then reverse (this example initial direction control by flag M8156 or M8157)



◎ Pulse V

FNC(157)			16 bits:PLSV - 7 steps
D	PLSV		32 bits:(D)PLSV - 13 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S.]

K,H	KnX	KnY	KnM	KnS	T	C	D	Z

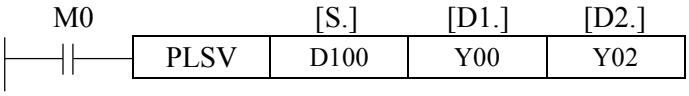
[S.]

Operands: [D2.]

X	Y	M	S

[D1.]

Flag: M8029



- ♦ [S.] assign output frequency and forward(+)reverse(-)direction。[16bits]:10~32,767Hz, [32bits]:10 ~ 100,000 Hz.
  [D1.] assign pulse output point。(fixed Y00,Y01 output point)。
  [D2.] assign direction output point。(fixed Y02,Y03 output point)。
- ♦ About above mentioned example, if D102=0, then it is without target running mode, like fig. 157-1, 157-2; if D102≠0, then it is target running mode.
- ♦ When pulse output, can change content of [S.], but can't change sign(+,-). If drive contact OFF, then decelerate to bias speed stop directly. M8140 and M8141 are mark location change flag.
- ♦ Following mode can reach, refer to Application Note：

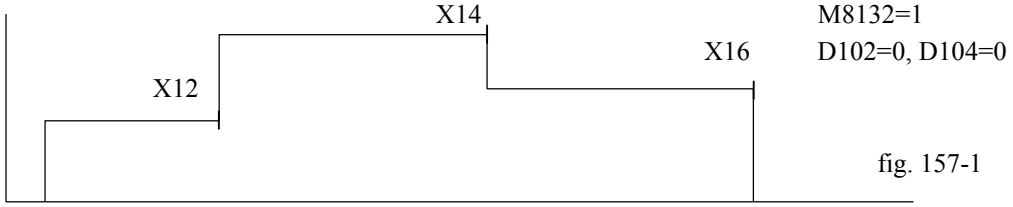


fig. 157-1

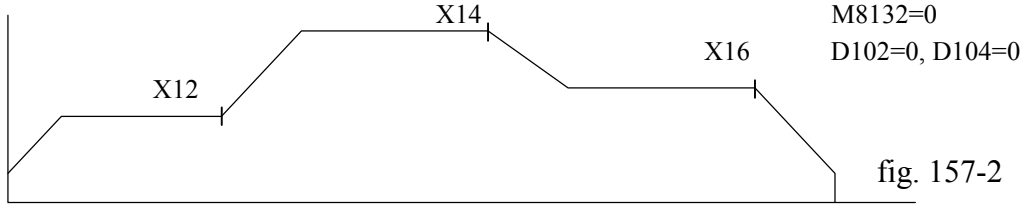
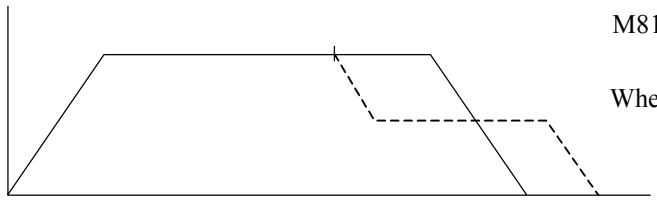
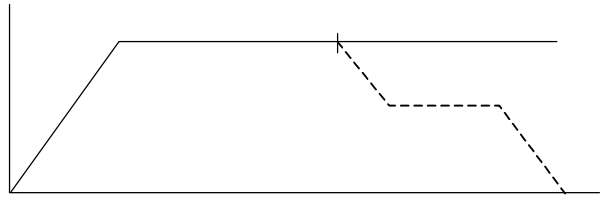


fig. 157-2



M8132=0
D102≠0, D104≠0
When M8140 or M8141=1, target location change

fig. 157-3



M8132=0
D102=0, D104≠0
When M8140 or M8141=1, target location change

fig. 157-4

◎ Drive to increment

FNC(158)			16 bits:DRVI - 9 steps
D	DRVI		32 bits:(D)DRVI - 17 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

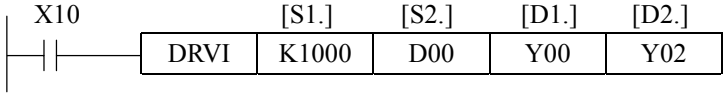
Operands: [S1.] [S2.]

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

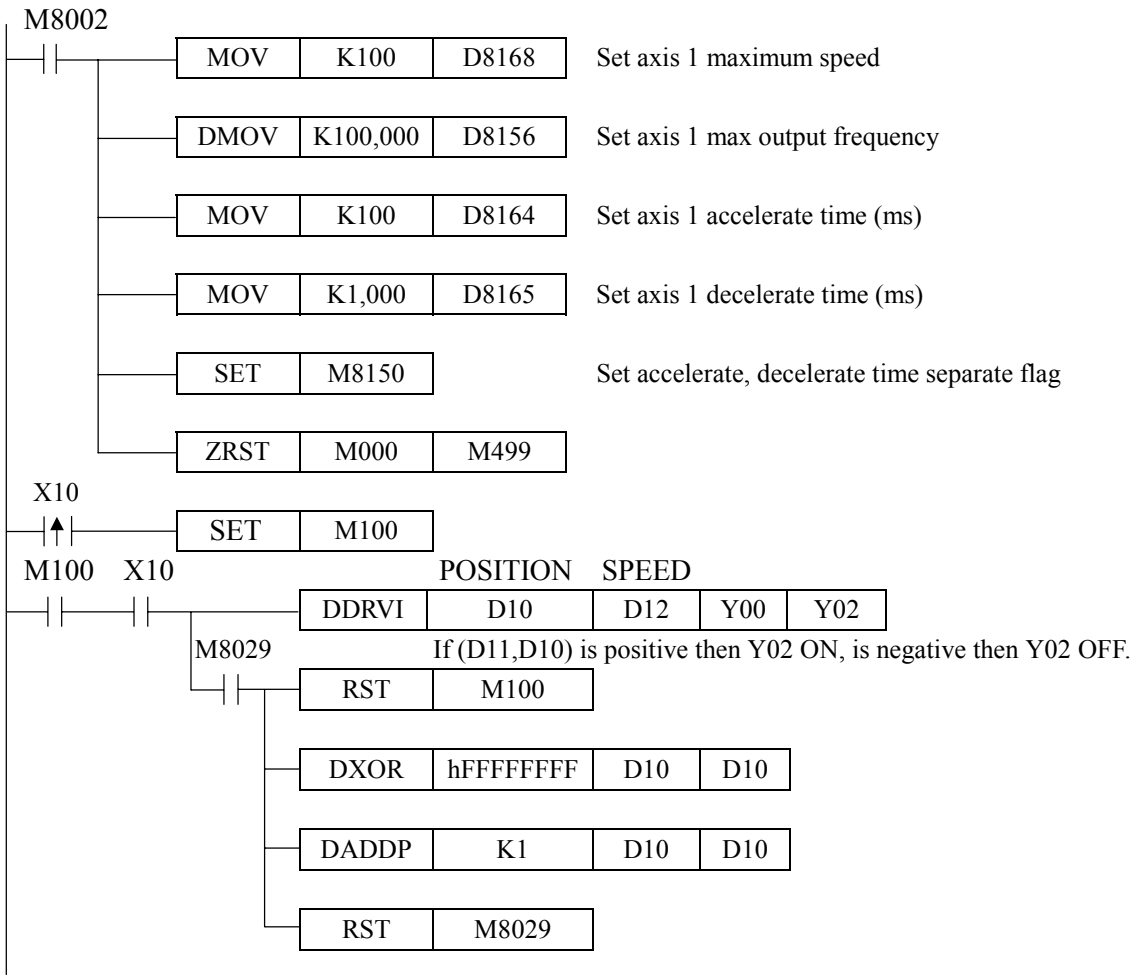
Operands: [D1.] [D2.]

X	Y	M	S

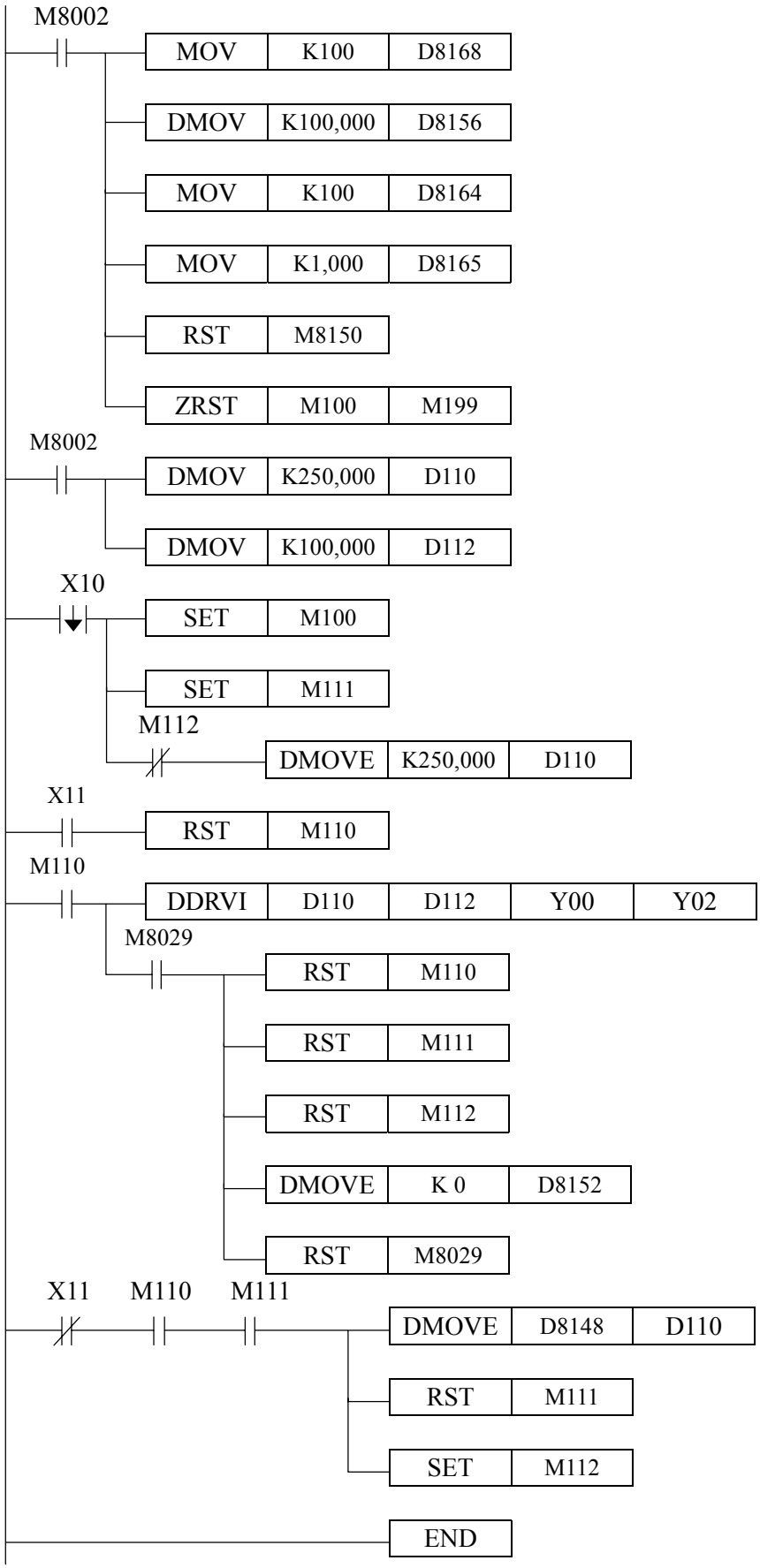
Flag: M8029



- ♦ [S1.] specify output pulse number, Don't care flag M8134,M8135。
  [S2.] specify output frequency。[16bits]:10~32,767Hz, [32bits]:10 ~ 100,000 Hz.
  [D1.] specify pulse output signal point。(only Y00,Y01)
  [D2.] specify direction output signal point。(only Y02,Y03)
- ♦ For Y00 or Y01, this instruction can be used once, and only transistor module can be selected.
- ♦ When executing DRVI, busy flag M8182 (Y00)or M8183 (Y01) will be set automatically by system.
- ♦ When output pulse, modify the content of [S1] [S2] is ineffective。



◈ Example of single position drive condition mode, stop then start again, then will output remain pulse number



◎ Drive to absolute

FNC(159)			16 bits:DRVA - 9 steps
D	DRVA		32 bits:(D)DRVA - 17 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

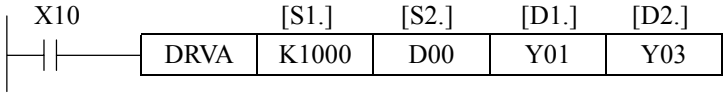
Operands: [S1.] [S2]

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: [D1.] [D2.]

X	Y	M	S

Flag: M8029

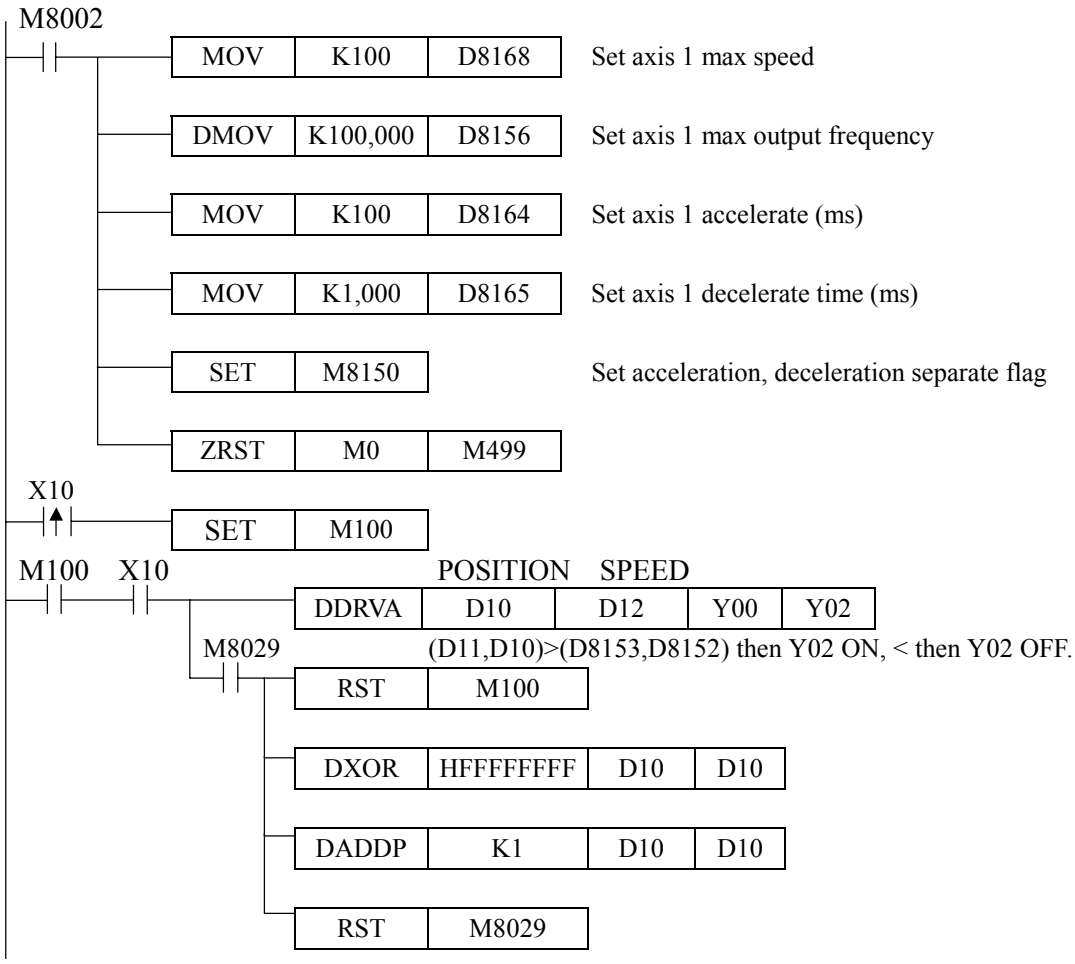


♦ [S1.] specify absolute address。
Don't care Flag M8134,M8135.
[S2.] specify output frequency。[16bits]:10~32,767Hz, [32bits]:10 ~ 100,000 Hz.
[D1.] specify pulse output signal point。(only Y00,Y01)
[D2.] specify direction output signal point。(only Y02,Y03)

♦ For Y00 or Y01, this instruction can be used once, and only transistor module can be selected.

♦ When executing DRVA, busy flag M8182 (Y00) or M8183 (Y01) will be set automatically by system.

♦ When output pulse, modify content of [S1] [S2] is ineffective。



◎ Time Compare

FNC(160)			16 bits:TCMP & TCMP(P) - - - - - - - - - - - - - - - - - 5 steps
D	TCMP	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.][S2.][S3.]

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

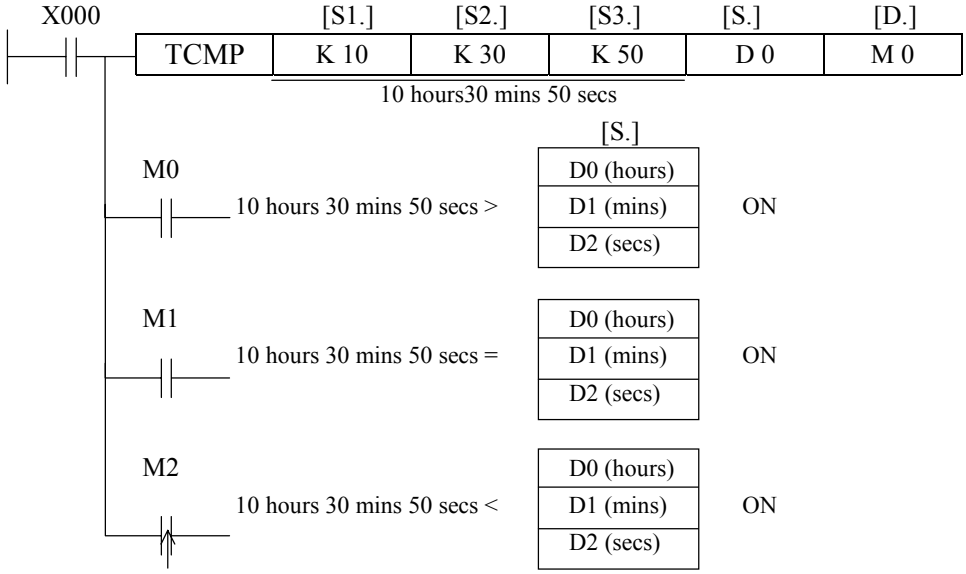
[S.]

Operands: [D.]

X	Y	M	S

[D.] occupy continuative 3 bits

Flag: M8020, M8021, M8022



When X000 OFF, not execute TCMP, M0~M2 status unchanged.

♦ Time of source device「[S1.],[S2.],[S3.]」compare with time value which stored at 3 bits from the head address of [S.]. According the result, the device of 3 bits from the head address of [D.] will be ON/OFF automatically.

[S1.] **:** "hour" assign「0~23」hour.
[S2.] **:** "min" assign「0~59」min.
[S3.] **:** "sec" assign「0~59」sec.

[S.] **:** "hour" assign「0~23」hour.
[S.] + 1 **:** "min" assign「0~59」min.
[S.] + 2 **:** "sec" assign「0~59」sec.

[D.],[D.] +1.,[D.] +2 **:** according the result, device of 3 bits from the head address of [D.] is ON/OFF automatically.

♦ The content of real time clock stored at special register D8015(hour),D8014(min),D8013(sec).

◎ Time Zone Compare

FNC(161)			16 bits:TZCP & TZCP(P) - - - - - - - - - - - - - - - - - - 9 steps
D	TZCP	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands: [S1.][S2.][S3.]

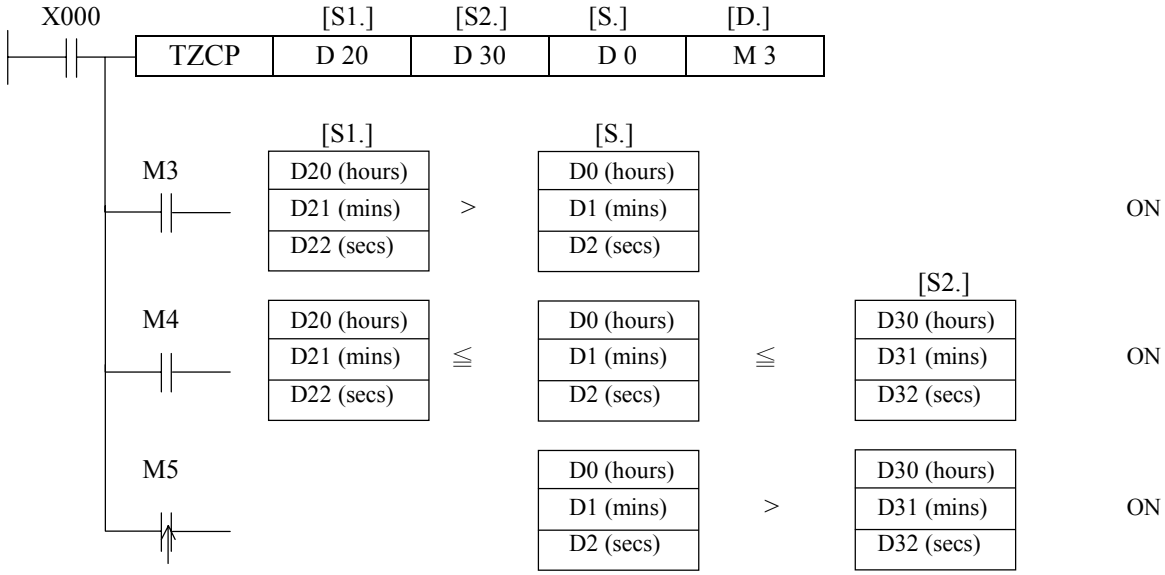
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands: [D.]

X	Y	M	S

Occupy 3 bits from the head address of [D.], set [S1.]≦[S2.]

Flag: M8020, M8021, M8022



When X000 OFF, then not execute TZCP, M3~M5 status unchanged.

♦ Compare it with time value zone of 3 bits from the head address of [S.]. According to the result, then 3 bits from the head address of [D.] will be ON/OFF automatically.

[S1.], [S.] +1, [S.] +2 **:** The lower limit of compare range, assign "hour" , "min" , "sec".
[S2.], [S2.] +1, [S2.] +2 **:** The topper limit of compare range, assign "hour" , "min" , "sec".

[S.], [S.] +1, [S.] +2 **:** real time clock, assign "hour" , "min" , "sec".
[D.], [D.] +1, [D.] +2 **:** According result of comparison, device of 3 bits from the head address of [D.] is ON/OFF automatically.

Setting range of "hour" , "min" , "sec" compare with real time clock, reference to FNC160 (TCMP).

◎ Time Addition

FNC(162)			16 bits: TADD & TADD(P) - - - - - - - - - - - - - - - - - 7 steps
D	TADD	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S1.][S2.]

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]

Operands:

X	Y	M	S

Flag: M8020, M8021, M8022

X000

	[S1.]	[S2.]	[D.]
TADD	D 10	D 20	D 30

(D10,D11,D12)+(D20,D21,D22) →(D30,D31,32)

[S1.]		[S2.]		[D.]
D10 10(hours)		D20 3(hours)		D30 13(hours)
D11 30(mins)	+	D21 10(mins)	→	D31 40(mins)
D12 10(secs)		D22 5(secs)		D32 15(secs)
10hours30mins10secs		3hours10mins5secs		13hours40mins15secs

- The time value stored at 3 bits from the head address of [S1.] add the time value stored at 3 bits from the head address of [S2.], then stored the result at the device of 3 bits from the head address of [D.].
- If the result greater than "24", carry flag M8022 ON, and the value of addend subtract 24, then stored at [D.].

[S1.]		[S2.]		[D.]
18 (hours)		10 (hours)		4 (hours)
10 (mins)	+	20 (mins)	→	30 (mins)
30 (secs)		5 (secs)		35 (secs)
18hours10mins10secs		10hours20mins5secs		4hours30mins35secs

- The result is "0" (0hour 0min 0sec), then zero flag M8020 ON.
- Setting range of "hour" , "min" , "sec" compare with real time clock, reference to FNC160 (TCMP) instruction.

◎ Time Subtraction

FNC(163)			16 bits: TSUB & TSUB(P) - - - - - - - - - - - - - - - - - 7 steps
D	TSUB	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S1.][S2.]

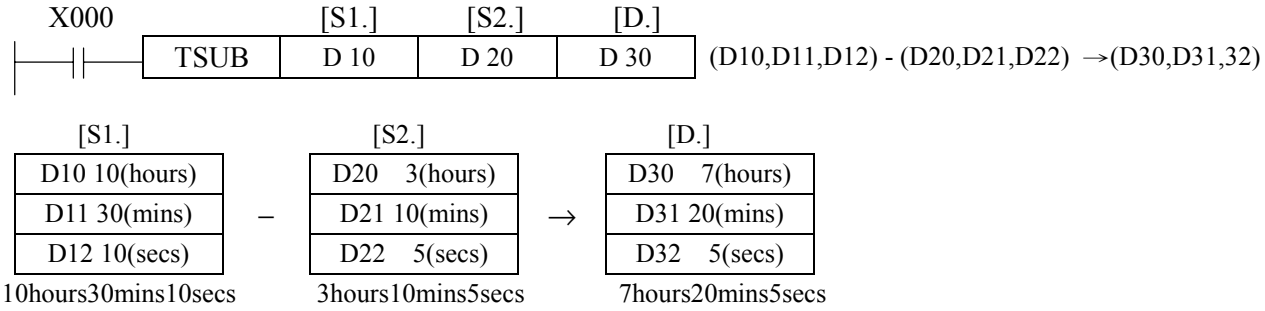
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]

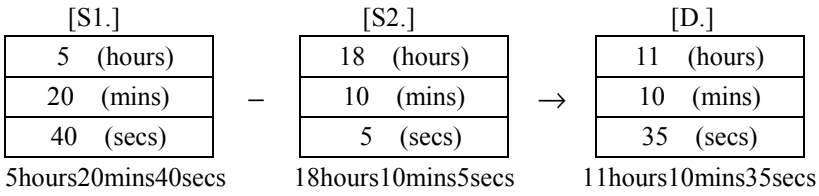
Operands:

X	Y	M	S

Flag: M8020, M8021, M8022



- The time value stored at 3 bits from the head address of [S1.] subtract the time value stored at 3 bits from the head address of [S2.], then stored the result at the device of 3 bits from the head address of [D.].
- The result less than "0", borrow flag ON, and the result of subtraction added 24, then stored at [D.]



- The result is "0" (0hour 0min 0sec), then zero flag M8020 ON.
- Setting range of "hour" , "min" , "sec" compare with real time clock, reference to FNC160(TCMP)instruction.

◎ Time Read

FNC(166)			16 bits: TRD & TRD(P) - 5 steps
D	TRD	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

[D.]

Operands:

X	Y	M	S

Occupy 7 bits from the head address of [D.]

Flag:

The data of real time clock stored at destination device of 7 bits from the head address of [D.]

.

Device	Item	Data		Device	Item
D8018	Year	0~99(last two figure)	→	D0	Year
D8017	Month	1~12	→	D1	Month
D8016	Date	1~31	→	D2	Date
D8015	Hours	0~23	→	D3	Hours
D8014	Minutes	0~59	→	D4	Minutes
D8013	Seconds	0~59	→	D5	Seconds
D8019	Week	0(Sun)~6(Sat)	→	D6	Week

◎ Time Write

FNC(167)			16 bits: TWR & TWR(P) - 5 steps
D	TWR	P	

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S.]

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands:

X	Y	M	S

Flag:

New time data write into special register D8013-D8019

.

Device	Item	Data		Device	Item
D10	Year	0~99(last two figure)	→	D8018	Year
D11	Month	1~12	→	D8017	Month
D12	Date	1~31	→	D8016	Date
D13	Hours	0~23	→	D8015	Hours
D14	Minutes	0~59	→	D8014	Minutes
D15	Seconds	0~59	→	D8013	Seconds
D16	Week	0(Sun)~6(Sat)	→	D8019	Week

◎ GRAY CODE

FNC(170)			16 bits:GRY & GRY(P) - 5 steps
D	GRY	P	32 bits:(D)GRY & (D)GRY(P) - - - - - - - - - - - - -9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S.]

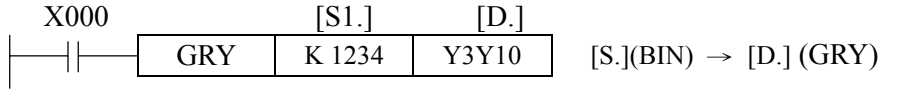
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

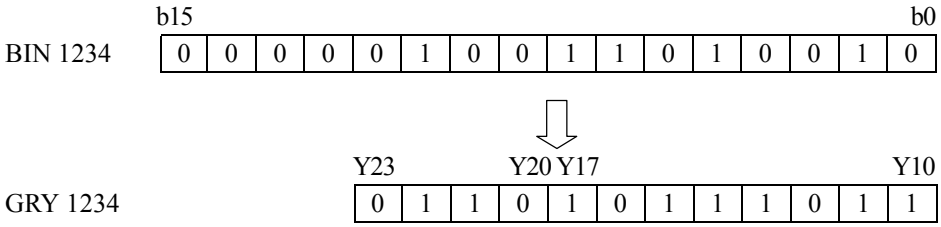
[D.]

Operands:

X	Y	M	S

Flag:





♦ [S.] effective value range

When 16 bits operation **:** 0~32,767

When 32 bits operation **:** 0~2,147,483,647

◎ GRAY CODE

FNC(171)			16 bits:GBIN & GBIN(P) - - - - - - - - - - - - - - - - - - 5 steps
D	GBIN	P	32 bits:(D)GBIN & (D)GBIN(P) - - - - - - - - - - - -9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

Operands:

[S.]

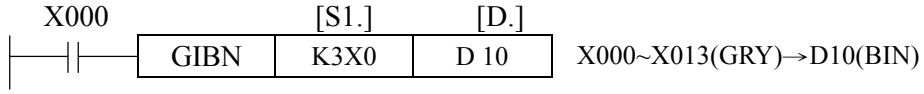
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

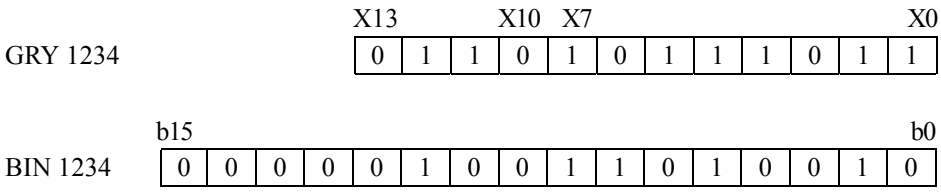
[D.]

Operands:

X	Y	M	S

Flag:





♦ When FNC51 (REFF) be used, need notice filter time (D8020-D8037) will response time.

♦ [S.] effective value range

When 16 bits operation**:** 0~32,767

When 32 bits operation**:** 0~2,147,483,647

◎ LD ※ (LoaD compare)

FNC(224~230)			16 bits: - 5 steps
D	LD ※		32 bits: - 9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

※ : = , > , < , <> , ≦ , ≧

Operands: [S1.][S2.]

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

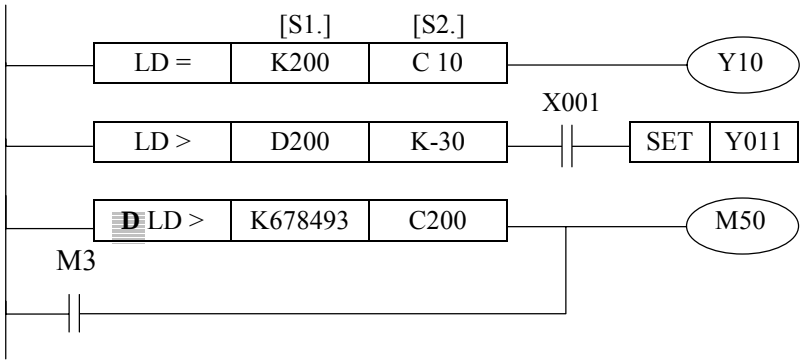
Operands:

X	Y	M	S

Flag:

♦ Comparison of BIN to the content of two source operands, according the result, update operate status

FNC No.	16 bits instruction	32 bits instruction	ON	OFF
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.]



♦ The upper bit of [S1.][S2.] is sign bit, i.e. 0: positive, 1: negative

♦ If use 32 bits counter (C200~) to compare, have to use 32 bits instruction.
If use 16 bits instruction to compare, then error will occur.

◎ AND ※ (AND compare) AND=, AND>, AND<, AND<>, AND<=, AND>=

FNC(232~238)			16 bits: - 5 steps
D	AND ※		32 bits: - 9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

※ : = , > , < , <> , ≦ , ≧

Operands: [S1.][S2.]

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

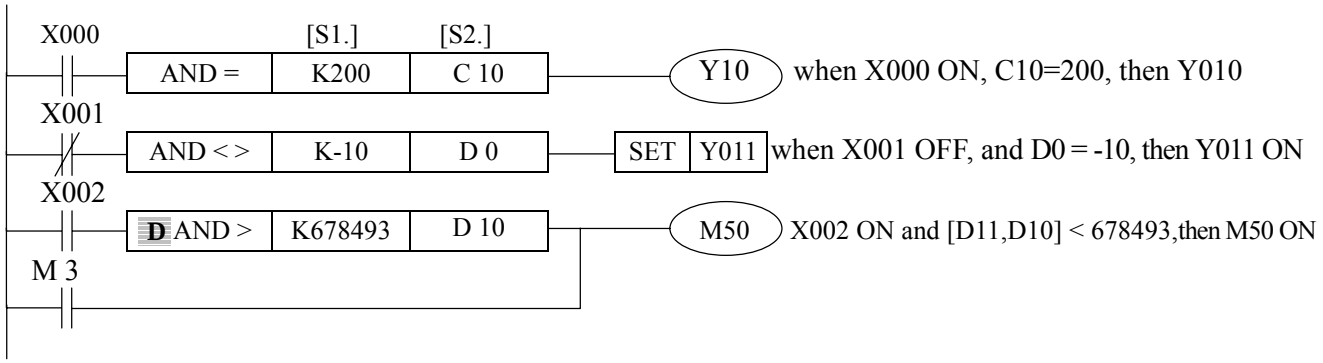
Operands:

X	Y	M	S

Flag:

♦ Comparison of BIN to the content of two source operands, according to the result, update operate status.

FNC No.	16 bits instruction	32 bits instruction	ON	OFF
232	AND =	**D** AND =	[S1.] = [S2.]	[S1.] ≠ [S2.]
233	AND >	**D** AND >	[S1.] > [S2.]	[S1.] ≦ [S2.]
234	AND <	**D** AND <	[S1.] < [S2.]	[S1.] ≧ [S2.]
236	AND < >	**D** AND < >	[S1.] ≠ [S2.]	[S1.] = [S2.]
237	AND ≦	**D** AND ≦	[S1.] ≦ [S2.]	[S1.] > [S2.]
238	AND ≧	**D** AND ≧	[S1.] ≧ [S2.]	[S1.] < [S2.]



♦ The upper bit of [S1.][S2.] is sign bit, i.e. 0: positive, 1: negative

♦ Use 32 bits counter (C200~) to compare, have to use 32 bits instruction.

. If use 16 bits instruction to compare, then error will occur.

◎ OR ※ (OR compare) OR=, OR>, OR<, OR<>, OR<=, OR>=

FNC(240~246)			16 bits: - 5 steps
D	OR ※		32 bits: - 9 steps

EX	EX_{1S}	EX_{1N}	EX_{2N}

※ : = , > , < , <> , ≦ , ≧

Operands: [S1.][S2.]

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

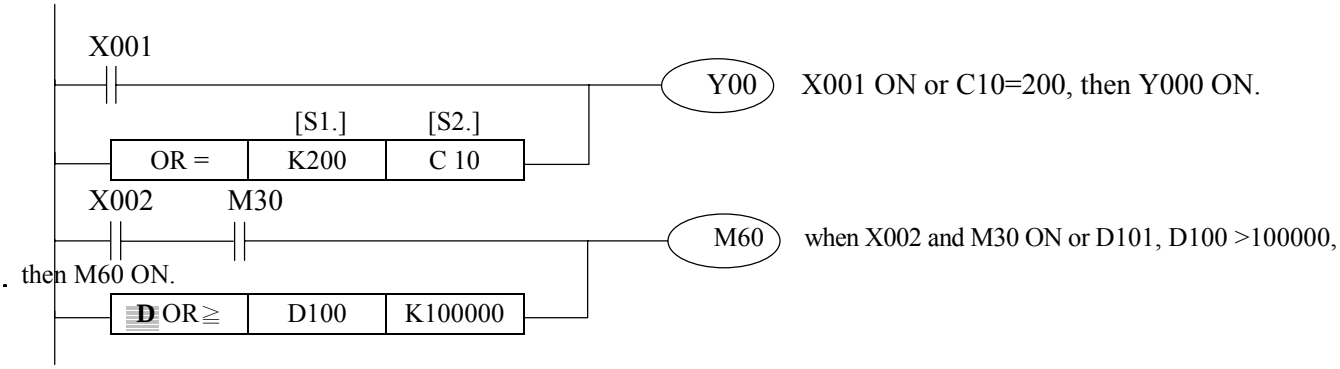
Operands:

X	Y	M	S

Flag:

♦Comparison of BIN to the content of two source operands, according the result, update operate status.

FNC No.	16 bits instruction	32 bits instruction	ON	OFF
240	OR =	**D** OR =	[S1.] = [S2.]	[S1.] ≠ [S2.]
241	OR >	**D** OR >	[S1.] > [S2.]	[S1.] ≦ [S2.]
242	OR <	**D** OR <	[S1.] < [S2.]	[S1.] ≧ [S2.]
244	OR < >	**D** OR < >	[S1.] ≠ [S2.]	[S1.] = [S2.]
245	OR ≦	**D** OR ≦	[S1.] ≦ [S2.]	[S1.] > [S2.]
246	OR ≧	**D** OR ≧	[S1.] ≧ [S2.]	[S1.] < [S2.]



♦ The upper bit of [S1.][S2.] is sign but, i.e. 0:positive, 1:negative

♦ When use 32 bits counter (C200~) to compare, then have to use 32 bits instruction.
If use 16 bits instruction to compare, then error will occur.

6. Special Auxiliary Relay & Data Register

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8000	Run monitor a contact	○	○	○	ON	√	×
M8001	Run monitor b contact	○	○	○	OFF	√	×
M8002	Initial pulse a contact	○	○	○	---	√	×
M8003	Initial pulse b contact	○	○	○	---	√	×
M8004	Error occurrence	○	○	○	OFF	√	×
M8005							
M8006							
M8007							
M8008	24V power failure	○	○	○	OFF	√	×
M8009	24Vdc drop to low	○	○	○	OFF	√	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8010							
M8011	10ms clock pulse 5ms ON 5ms OFF	○	○	○	---	√	×
M8012	100ms clock pulse 50ms ON 50ms OFF	○	○	○	---	√	×
M8013	1.0sec clock pulse 0.5sec ON 0.5sec OFF	○	○	○	---	√	×
M8014	1.0min clock pulse 0.5min ON 0.5min OFF	○	○	○	---	√	×
M8015							
M8016							
M8017							
M8018							
M8019	Real Time Data Error Flag	○	○	○	OFF	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8020	Zero flag	○	○	○	OFF	√	√
M8021	Borrow flag	○	○	○	OFF	√	√
M8022	Carry flag	○	○	○	OFF	√	√
M8023							
M8024							
M8025							
M8026							
M8027							
M8028							
M8029	Instruction execution complete flag	○	○	○	OFF	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8030							
M8031							
M8032							
M8033							
M8034	Output disable when ON	○	○	○	OFF	√	√
M8035	Run/Stop flag	○	○	○	---	√	√
M8036	Force to run mode	○	○	○	---	√	√
M8037	Force stop mode	○	○	○	---	√	√
M8038							
M8039	Fix scantime mode flag	○	○	○	OFF	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8000	Watchdog timer (ms)	○	○	○	100	√	√
D8001	Type & Version	○	○	○	---	√	×
D8002	Memory size	○	○	○	---	√	√
D8003	Memory kind	○	○	○	---	√	×
D8004	Error number	○	○	○	0	√	×
D8005	---						
D8006	---						
D8007							
D8008							
D8009							

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8010	Preset scan time (unit = 0.1ms)	○	○	○	10	√	√
D8011	Min. can time (unit = 0.1ms)	○	○	○	10	√	×
D8012	Max. scan time (unit = 0.1ms)	○	○	○	10	√	×
D8013	Second	○	○	○	0	√	√
D8014	Minute	○	○	○	0	√	√
D8015	Hour	○	○	○	12	√	√
D8016	Day	○	○	○	11	√	√
D8017	Month	○	○	○	08	√	√
D8018	Year	○	○	○	03	√	√
D8019	Week	○	○	○	1	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8020	X000~X007 filter	○	○	○	5	√	√
D8021	X010~X017 filter	○	○	○	5	√	√
D8022	X020~X027 filter	○	○	○	5	√	√
D8023	X030~X037 filter	○	○	○	5	√	√
D8024	X040~X047 filter	○	○	○	5	√	√
D8025	X050~X057 filter	○	○	○	5	√	√
D8026	X060~X067 filter	○	○	○	5	√	√
D8027	X070~X078 filter	○	○	○	5	√	√
D8028	Z index register	○	○	○	0	√	√
D8029	V index register	○	○	○	0	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8030	X100~X107 filter	○	○	○	5	√	√
D8031	X110~X117 filter	○	○	○	5	√	√
D8032	X120~X127 filter	○	○	○	5	√	√
D8033	X130~X137 filter	○	○	○	5	√	√
D8034	X140~X147 filter	○	○	○	5	√	√
D8035	X150~X157 filter	○	○	○	5	√	√
D8036	X160~X167 filter	○	○	○	5	√	√
D8037	X170~X177 filter	○	○	○	5	√	√
D8038	End of User Program Step Number	○	○	○	---	√	×
D8039	Fixed scan time	○	○	○	---	√	√

※ D8001: 22 102

Version 1.02

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8040							
M8041							
M8042							
M8043							
M8044							
M8045							
M8046	Any STL state ON	○	○	○	OFF	√	√
M8047	Monitor STL state	○	○	○	OFF	√	√
M8048							
M8049							

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8050	I0xx int. disable when ON	○	○	○	ON	√	√
M8051	I1xx int. disable when ON	○	○	○	ON	√	√
M8052	I2xx int. disable when ON	○	○	○	ON	√	√
M8053	I3xx int. disable when ON	○	○	○	ON	√	√
M8054	I4xx int. disable when ON	○	○	○	ON	√	√
M8055	I5xx int. disable when ON	○	○	○	ON	√	√
M8056	I6xx int. disable when ON	○	○	○	ON	√	√
M8057	I7xx int. disable when ON	○	○	○	ON	√	√
M8058	I8xx int. disable when ON	○	○	○	ON	√	√
M8059	Don’t used						

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8060	I/O combination error	○	○	○	OFF	√	×
M8061	PLC hardware error	○	○	○	OFF	√	×
M8062	RS232C error	○	○	○	OFF	√	×
M8063	Link/485 error	○	○	○	OFF	√	×
M8064	Parameter error	○	○	○	OFF	√	×
M8065	Syntax error	○	○	○	OFF	√	×
M8066	Program error	○	○	○	OFF	√	×
M8067	Operation error	○	○	○	OFF	√	×
M8068	Operation error	○	○	○	OFF	√	×
M8069	I/O bus error	○	○	○	OFF	√	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8070	=1, master station	○	○	○	OFF	√	√
M8071	=1, slave station	○	○	○	OFF	√	√
M8072	Reserved	○	○	○	OFF	√	×
M8073	Parallel link master station overtime flag	○	○	○	OFF	√	×
M8074							
M8075	Ready to start sampling trace instruction						
M8076	Sampling trace ready instruction					√	√
M8077	Sampling trace executing signal					√	×
M8078	Sampling Trace	○	○	○	OFF	-	-
M8079	Reserved	○	○	○	OFF	-	-

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8040	STL on state no.1	○	○	○	---	√	×
D8041	STL on state no.2	○	○	○	---	√	×
D8042	STL on state no.3	○	○	○	---	√	×
D8043	STL on state no.4	○	○	○	---	√	×
D8044	STL on state no.5	○	○	○	---	√	×
D8045	STL on state no.6	○	○	○	---	√	×
D8046	STL on state no.7	○	○	○	---	√	×
D8047	STL on state no.8	○	○	○	---	√	×
D8048							
D8049							

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8050	I0xx int. vector	○	○	○	---	√	×
D8051	I1xx int. vector.	○	○	○	---	√	×
D8052	I2xx int. vector	○	○	○	---	√	×
D8053	I3xx int. vector	○	○	○	---	√	×
D8054	I4xx int. vector	○	○	○	---	√	×
D8055	I5xx int. vector	○	○	○	---	√	×
D8056	I6xx int. vector	○	○	○	---	√	×
D8057	I7xx int. vector	○	○	○	---	√	×
D8058	I8xx int. vector	○	○	○	---	√	×
D8059							

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8060	I/O combination error	○	○	○	---	√	×
D8061	PLC hardware error	○	○	○	---	√	×
D8062	Communication error	○	○	○	---	√	×
D8063	Communication error	○	○	○	---	√	×
D8064	Parameter error	○	○	○	---	√	×
D8065	Syntax error	○	○	○	---	√	×
D8066	Circuit error	○	○	○	---	√	×
D8067	Operation error	○	○	○	---	√	×
D8068	Error code	○	○	○	---	√	×
D8069	Error step numbers	○	○	○	---	√	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8070	Parallel link overtime register (ms)	○	○	○	---	√	√
D8071							
D8072	Parallel link taking time (ms)	○	○	○	---	√	×
D8073							
D8074	sampling remain times	○	○	○	○	√	×
D8075	Sampling times set (1-256)	○	○	○	---	√	√
D8076	Sampling cycle time set <<0: sample per cycle，1:10ms sample once…	○	○	○	---	√	√
D8077	Sampling trace condition assigned			○	---	√	√
D8078	Set component no. of conditioned sampling trace			○	---	√	√
D8079	Sampling data index			○	---	√	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8080	Don't used				---	-	×
M8081	Don't used				---	-	×
M8082	Don't used				---	-	×
M8083	Don't used				---	-	×
M8084	Don't used				---	-	×
M8085	Don't used				---	-	×
M8086	Don't used				---	-	×
M8087	Don't used				---	-	×
M8088	Don't used				---	-	×
M8089	Don't used				---	-	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8090	Don't used				---	-	×
M8091	Don't used				---	-	×
M8092	Don't used				---	-	×
M8093	Don't used				---	-	×
M8094	Don't used				---	-	×
M8095	Don't used				---	-	×
M8096	Don't used				---	-	×
M8097	Don't used				---	-	×
M8098	Don't used				---	-	×
M8099	High-speed Circular counter enable flag(0.1ms)	○	○	○	---	√	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8100	Don't used				---	-	×
M8101	Don't used				---	-	×
M8102	Don't used				---	-	×
M8103	Don't used				---	-	×
M8104	Don't used				---	-	×
M8105	Don't used				---	-	×
M8106	Don't used				---	-	×
M8107	Don't used				---	-	×
M8108	Don't used				---	-	×
M8109	Don't used				---	-	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8110	2AD-CH1 Voltage (OFF) or Current (ON) Monitor Selective Flag	○	○	○	---	√	√
M8111	2AD-CH2 Voltage (OFF) or Current (ON) Monitor Selective Flag	○	○	○	---	√	√
M8112	2AD-CH1 Enable Flag	○	○	○	---	√	√
M8113	2AD-CH2 Enable Flag	○	○	○	---	√	√
M8114	2TC-CH1 Enable Flag	○	○	○	---	√	√
M8115	2TC-CH2 Enable Flag	○	○	○	---	√	√
M8116	2PT-CH1 Enable Flag	○	○	○	---	√	√
M8117	2PT-CH2 Enable Flag	○	○	○	---	√	√
M8118	2LD-CH1 Enable Flag	○	○	○	---	√	√
M8119	2LD-CH2 Enable Flag	○	○	○	---	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8080	Sampling bit component No.00				---	-	×
D8081	Sampling bit component No.01				---	-	×
D8082	Sampling bit component No.02				---	-	×
D8083	Sampling bit component No.03				---	-	×
D8084	Sampling bit component No.04				---	-	×
D8085	Sampling bit component No.05				---	-	×
D8086	Sampling bit component No.06				---	-	×
D8087	Sampling bit component No.07				---	-	×
D8088	Sampling bit component No.08				---	-	×
D8089	Sampling bit component No.09				---	-	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8090	Sampling bit component No.10				---	-	×
D8091	Sampling bit component No.11				---	-	×
D8092	Sampling bit component No.12				---	-	×
D8093	Sampling bit component No.13				---	-	×
D8094	Sampling bit component No.14				---	-	×
D8095	Sampling bit component No.15				---	-	×
D8096	Sampling character component No.00				---	-	×
D8097	Sampling character component No.01				---	-	×
D8098	Sampling character component No.02				---	-	×
D8099	加算 Circular counter(unit:0.1ms)				---	-	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8100	System reserved				---	-	×
D8101	System reserved				---	-	×
D8102	Size of memory 2:2k, 4:4k, 8:8k steps			○	---	√	×
D8103	System reserved, don't used				---	-	×
D8104	System reserved, don't used				---	-	×
D8105	System reserved, don't used				---	-	×
D8106	System reserved, don't used				---	-	×
D8107	System reserved, don't used				---	-	×
D8108	System reserved				---	-	×
D8109	System reserved				---	-	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8110	2AD, TC, PT, 2LD parameter (refer to user's manual)	○	○	○	---	√	√
D8111	2AD, TC, PT, 2LD parameter (refer to user's manual)	○	○	○	---	√	√
D8112	2AD-CH1 measurement value	○	○	○	0	√	×
D8113	2AD-CH2 measurement value	○	○	○	0	√	×
D8114	2AD, TC, PT, LD parameter (refer to user's manual of 2AD)	○	○	○	---	√	√
D8115	2AD, TC, PT, LD parameter (refer to user's manual of 2AD)	○	○	○	---	√	√
D8116	2AD, TC, PT, LD parameter (refer to user's manual of 2AD)	○	○	○	---	√	√
D8117	2AD, TC, PT, LD parameter (refer to user's manual of 2AD)	○	○	○	---	√	√
D8118	Internal system reserved, don't used				---	-	×
D8119	Internal system reserved, don't used				---	-	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8120	Reserved	○	○	○	OFF	-	-
M8121	Wait Transmit flag	○	○	○	OFF	√	√
M8122	Transmit request flag	○	○	○	OFF	√	√
M8123	Receive finish flag	○	○	○	OFF	√	√
M8124	Carrier detect flag	○	○	○	OFF	√	√
M8125					---		
M8126					---		
M8127					---		
M8128	Modbus CRC checksum error flag	○	○	○	OFF	√	×
M8129	Modbus LRC checksum error flag	○	○	○	OFF	√	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8130	Y00 Without Target Flag	○	○	○	OFF	√	√
M8131	Y01 Without Target Flag	○	○	○	OFF	√	√
M8132	Y00 Emergency Stop Flag FNC(157)PLSV	○	○	○	OFF	√	√
M8133	Y01 Emergency Stop Flag FNC(157)PLSV	○	○	○	OFF	√	√
M8134	FNC(59)PLSR Y00 absolute position drive flag	○	○	○	OFF	√	√
M8135	FNC(59)PLSR Y01 absolute position drive flag	○	○	○	OFF	√	√
M8136	Y00 MPG enable flag FNC(59)			○	OFF	√	√
M8137	Y01 MPG enable flag FNC(59)			○	OFF	√	√
M8138	Y00 MPG busy flag FNC(59)			○	OFF	√	×
M8139	Y01 MPG busy flag FNC(59)			○	OFF	√	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8140	Y00 Mark Flag FNC(157)PLSV	○	○	○	OFF	√	√
M8141	Y01 Mark Flag FNC(157)PLSV	○	○	○	OFF	√	√
M8142	FNC(59)PLSR Linear Interpolation enable flag				---		
M8143	FNC(59)PLSR Circular Interpolation enable flag				---		
M8144	FNC(59)PLSR Y00 Zero Return Flag	○	○	○	OFF	√	√
M8145	FNC(59)PLSR Y01 Zero Return Flag	○	○	○	OFF	√	√
M8146	FNC(59) PLSR Y00 Jog Forward Flag	○	○	○	OFF	√	√
M8147	FNC(59) PLSR Y01 Jog Forward Flag	○	○	○	OFF	√	√
M8148	FNC(59) PLSR Y00 Jog Reverse Flag	○	○	○	OFF	√	√
M8149	FNC(59) PLSR Y01 Jog Reverse Flag	○	○	○	OFF	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8150	DRVI,DRVA Y00 accelerate time and decelerate time separated flag	○	○	○	OFF	√	√
M8151	DRVI,DRVA Y01 accelerate time and decelerate time separated flag	○	○	○	OFF	√	√
M8152	FNC(59)PLSY,FNC(156)ZRN Y00 zero return finish flag	○	○	○	OFF	√	√
M8153	FNC(59)PLSY,FNC(156)ZRN Y01 zero return finish flag	○	○	○	OFF	√	√
M8154	FNC(156) Y00 zero return mode, 0:reverse mode 1:forward mode	○	○	○	OFF	√	√
M8155	FNC(156) Y01 zero return mode, 0:reverse mode 1:forward mode	○	○	○	OFF	√	√
M8156	Y00 zero return direction, when M8158=0 then 0:forward 1:reverse	○	○	○	OFF	√	√
M8157	Y01 zero return direction, when M8159=0 then 0:forward 1:reverse	○	○	○	OFF	√	√
M8158	Y00 ZRN init dir selection, =0 decide by M8156, =1 decide by D8177,6	○	○	○	OFF	√	√
M8159	Y01 ZRN init dir selection, =0 decide by M8156, =1 decide by D8179,8	○	○	○	OFF	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8120	Communication protocol format	○	○	○	0368h	√	√
D8121	Station Number	○	○	○	00h	√	√
D8122	Remain data number when transmit	○	○	○	---	√	√
D8123	Receive data number	○	○	○	---	√	√
D8124	Start (STX)	○	○	○	02h	√	√
D8125	End1 (ETX1)	○	○	○	03h	√	√
D8126	End2 (ETX2)	○	○	○	---	√	√
D8127							
D8128							
D8129	Overtime detect (ms)	○	○	○	200	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8130	Y00 MPG movement (Lower Word)			○	0	√	√
D8131	Y00 MPG movement (Upper Word)			○		√	√
D8132	Y01 MPG movement (Lower Word)			○	0	√	√
D8133	Y01 MPG movement (Upper Word)			○		√	√
D8134	Y00 time of follow MPG movement (ms)			○	10	√	√
D8135	Y01 time of follow MPG movement (ms)			○	10	√	√
D8136	Y00 target relate position (Lower Word)	○	○	○	0	√	√
D8137	Y00 target relate position (Upper Word)	○	○	○		√	√
D8138	Y01 target relate position (Lower Word)	○	○	○	0	√	√
D8139	Y01 target relate position (Upper Word)	○	○	○		√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8140	Y00 current absolute position (Lower Word)	○	○	○	0	√	√
D8141	Y00 current absolute position (Upper Word)	○	○	○			
D8142	Y01 current absolute position (Lower Word)	○	○	○	0	√	√
D8143	Y01 current absolute position (Upper Word)	○	○	○			
D8144	Y00 relate position movement (Lower Word)	○	○	○	0	√	×
D8145	Y00 relate position movement (Upper Word)	○	○	○			
D8146	Y01 relate position movement (Lower Word)	○	○	○	0	√	×
D8147	Y01 relate position movement (Upper Word)	○	○	○			
D8148	Y00 remain pulse (Lower Word)	○	○	○	0	√	×
D8149	Y00 remain pulse (Upper Word)	○	○	○			

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8150	Y01 remain pulse (Lower Word)	○	○	○	0	√	×
D8151	Y01 remain pulse (Upper Word)	○	○	○			
D8152	Y00 starting absolute position (Lower Word)	○	○	○	0	√	×
D8153	Y00 starting absolute position (Upper Word)	○	○	○			
D8154	Y01 starting absolute position (Lower Word)	○	○	○	0	√	√
D8155	Y01 starting absolute position (Upper Word)	○	○	○			
D8156	Y00 maximum output frequency (Lower Word)	○	○	○	100K	√	√
D8157	Y00 maximum output frequency (Upper Word)	○	○	○			
D8158	Y01 maximum output frequency (Lower Word)	○	○	○	100K	√	√
D8159	Y01 maximum output frequency (Upper Word)	○	○	○			

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8160	SWAP mode	○	○	○	OFF	√	√
M8161	8/16bits selection flag	○	○	○	---	√	√
M8162					---	-	×
M8163					---	-	×
M8164					---	-	×
M8165					---	-	×
M8166					---	-	×
M8167					---	-	×
M8168					---	-	×
M8169					---	-	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8170	X00 pulse catch	○	○	○	OFF	√	√
M8171	X01 pulse catch	○	○	○	OFF	√	√
M8172	X02 pulse catch	○	○	○	OFF	√	√
M8173	X03 pulse catch	○	○	○	OFF	√	√
M8174	X04 pulse catch	○	○	○	OFF	√	√
M8175	X05 pulse catch	○	○	○	OFF	√	√
M8176	X06 pulse catch	○	○	○	OFF	√	√
M8177	X07 pulse catch	○	○	○	OFF	√	√
M8178	Reserved				---	-	×
M8179	Reserved				---	-	×

Following Device For Monitor Used, Will Auto set or clear by system

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8180	Y00 FNC(59)PLSR Busy Flag	○	○	○	OFF	√	×
M8181	Y01 FNC(59)PLSR Busy Flag	○	○	○	OFF	√	×
M8182	Y00 FNC(158)DRVI, FNC(159)DRVA Busy Flag	○	○	○	OFF	√	×
M8183	Y01 FNC(158)DRVI, FNC(159)DRVA Busy Flag	○	○	○	OFF	√	×
M8184	Y00 FNC(156)ZRN Busy Flag	○	○	○	OFF	√	×
M8185	Y01 FNC(156)ZRN Busy Flag	○	○	○	OFF	√	×
M8186	Linear Interpolation busy flag						
M8187							
M8188	Circular Interpolation busy flag						
M8189							

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8190	Y00 FNC(59)PLSR Zero Return Busy Flag	○	○	○	OFF	√	×
M8191	Y01 FNC(59)PLSR Zero Return Busy Flag	○	○	○	OFF	√	×
M8192	Y00 FNC(157)PLSV Busy Flag	○	○	○	OFF	√	×
M8193	Y01 FNC(157)PLSV Busy Flag	○	○	○	OFF	√	×
M8194	Y00 FNC(57)PLSY Busy Flag	○	○	○	OFF	√	×
M8195	Y01 FNC(57)PLSY Busy Flag	○	○	○	OFF	√	×
M8196	Y00 FNC(59)PLSR Jog Forward Busy Flag	○	○	○	OFF	√	×
M8197	Y01 FNC(59)PLSR Jog Reverse Busy Flag	○	○	○	OFF	√	×
M8198	Y00 FNC(59)PLSR Jog Forward Busy Flag	○	○	○	OFF	√	×
M8199	Y01 FNC(59)PLSR Jog Reverse Busy Flag	○	○	○	OFF	√	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8160	Y00 current speed (pps) Lower Word	○	○	○	0	√	×
D8161	Y00 current speed (pps) Upper Word	○	○	○			
D8162	Y01 current speed (pps) Lower Word	○	○	○	0	√	×
D8163	Y01 current speed (pps) Upper Word	○	○	○			
D8164	Y00 time of acceleration and deceleration (ms)	○	○	○	100	√	√
D8165	Y00 time of deceleration (ms) , when M8150 ON effective	○	○	○	100	√	√
D8166	Y01 time of accelerate and deceleration (ms)	○	○	○	100	√	√
D8167	Y01 time of deceleration (ms), when M8151 ON effective	○	○	○	100	√	√
D8168	Y00 bias speed (pps)	○	○	○	100	√	√
D8169	Y00 search the number of servo Z phase	○	○	○	1	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8170	Y01 bias speed (pps)	○	○	○	100	√	√
D8171	Y01 the number of search servo Z phase	○	○	○	1	√	√
D8172	Y00 The pulse of acceleration to maximum speed (Lower Word)	○	○	○	0	√	×
D8173	Y00 The pulse of acceleration to maximum speed (Upper Word)	○	○	○			
D8174	Y01 The pulse of acceleration to maximum speed (Lower Word)	○	○	○	0	√	×
D8175	Y01 The pulse of acceleration to maximum speed (Upper Word)	○	○	○			
D8176	Y00 Dog Point Absolute Address (Lower Word)	○	○	○	0	√	√
D8177	Y00 Dog Point Absolute Address (Upper Word)	○	○	○			
D8178	Y01 Dog Point Absolute Address (Lower Word)	○	○	○	0	√	√
D8179	Y01 Dog Point Absolute Address (Upper Word)	○	○	○			

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8180	The content of z0 register	○	○	○	0	√	√
D8181	The content of v0 register	○	○	○	0	√	√
D8182	The content of z1 register	○	○	○	0	√	√
D8183	The content of v1 register	○	○	○	0	√	√
D8184	The content of z2 register	○	○	○	0	√	√
D8185	The content of v2 register	○	○	○	0	√	√
D8186	The content of z3 register	○	○	○	0	√	√
D8187	The content of v3 register	○	○	○	0	√	√
D8188	The content of z4 register	○	○	○	0	√	√
D8189	The content of v4 register	○	○	○	0	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8190	The content of z5 register	○	○	○	0	√	√
D8191	The content of v5 register	○	○	○	0	√	√
D8192	The content of z6 register	○	○	○	0	√	√
D8193	The content of v6 register	○	○	○	0	√	√
D8194	The content of z7 register	○	○	○	0	√	√
D8195	The content of v7 register	○	○	○	0	√	√
D8196	Y00 MPG electronic gear ratio (numerator)			○	1	√	√
D8197	Y00 MPG electronic gear ratio (denominator)			○	1	√	√
D8198	Y01 MPG electronic gear ratio (numerator)			○	1	√	√
D8199	Y01 MPG electronic gear ratio (denominator)			○	1	√	√

Up/Down Counter

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8200 | | | | | M8234	When M8xxx=1, Cxxx down counter When M8xxx=0, Cxxx up counter	○	○	○	---	√	√

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8200 | | | | | D8234	Reserved	○	○	○	---	-	×

High Speed Counter

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
M8235 | | M8245	When M8xxx=1, Cxxx down counter When M8xxx=0, Cxxx up counter	○	○	○	---	√	√
M8246 | | M8255	If Cxxx is down counter Then M8xxx=1 If Cxxx is up counter Then M8xxx=0	○	○	○	---	√	×

Number	Content Of Register	The series of PLC				R	W
		EX_{1S}	EX_{1N}	EX_{2N}	Initial value		
D8235 | | D8245	system reserved, don't used.	○	○	○	---	-	×
D8246 | | D8249	system reserved, don't used.	○	○	○	---	-	×

Error code	Associated Meaning
0000	No error
6001	
6002	
6003	
6004	
6005	
6006	
6007	
6008	
6009	

Error code	Associated Meaning
0000	No error
6101	SRAM hardware error
6102	
6103	Dummy Error
6104	EEPROM hardware error
6105	Led frame error
6106	AC power quality no good
6107	24Vdc power failure
6108	Monitor program overflow
6109	User program overflow

Error code	Associated Meaning
0000	No error
6201	
6202	
6203	
6204	
6205	
6206	
6207	
6208	
6209	

Error code	Associated Meaning
0000	No error
6301	
6302	
6303	
6304	
6305	
6306	
6307	
6308	
6309	Watchdog overflow

Error code	Associated Meaning
0000	No error
6401	
6402	
6403	
6404	
6405	
6406	
6407	
6408	
6409	

Error code	Associated Meaning
0000	No error
6501	
6502	
6503	
6504	
6505	
6506	System program error
6507	System watchdog error
6508	
6509	No this instruction

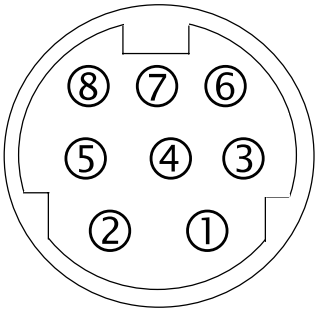
Error code	Associated Meaning
0000	No error
6601	LD & LDI used continuously more than 8
6602	LD, LDI & ANL, ORL incorrect
6603	MPS used continuously more than 11 times
6604	MPP & MPS incorrect
6605	STL & RET error
6606	No SRET or IRET instruction
6607	FOR & NEXT error
6608	MC & MCR incorrect
6609	END missing

Error code	Associated Meaning
0000	No error
6701	
6702	
6703	
6704	
6705	Applied instruction error (program keep run)
6706	Applied instruction error (program stop)
6707	
6708	
6709	

7. Index A

◎ pLocon Ex1s,Ex1n,Ex2n series RS232-C INTERFACE PIN ARRANGEMENT

♦ Up view the connector of pLocon Ex1s, Ex1n, Ex2n series



1	RTS
2	TXD
3	GND
4	CTS
5	5V+
6	GND
7	RXD
8	

♦ If connect to the data access (with power), don't connect 5V+ to each other.

◎ IBM PC RS232-C INTERFACE PIN ARRANGEMENT

9 pin RS - 232	25 pin RS - 232
1 : CD	8 : CD
2 : RXD	3 : RXD
3 : TXD	2 : TXD
4 : DTR	20 : DTR
5 : GND	7 : GND
6 : DSR	6 : DSR
7 : RTS	4 : RTS
8 : CTS	5 : CTS
9 : RI	22 : RI