XC Series programmable controller

User manual

XINJE Electronic Co., Ltd

Catalog

1．Preface

——The characteristic of programmable controller

The program of XC series programmable controller has the following characteristic：

- **Support two types of program language**

  In XC series PLC，besides instruction list format，you can also adopt ladder format on the screen. And，these one format could convert to the other。

- **With rich basic function**

  Base on the thought of“basic function、high-speed disposal、convient to use”，XC series PLC can support not only functions relative to ordinal control，but also basic applied instructions of data move and compare、arithmetic and logic control、loop and shift of data etc.，besides, it can support interruption、compare instructions exclusive used by high-speed counter、high-speed pulse output and other high-speed dispose instructions.

- **For high speed input, both single phase and double phase high-speed counters are available**

  As the common counters in PLC only act in the scan cycle, so their response speed is about 10Hz. However, as the high-speed counter placed in XC series PLC only carry on interrupt disposal with the high-speed pulse from special input relay. So it’s independent with the scan cycle and can count with the speed of up to 60KHz。

2．Summary of XC series PLC

XC series PLC is mini model PLC with powerful function. This series products can satisfy diverse control demand. With compact design、excellent extend capability、competitive price and powerful function, XC series PLC has become perfect resolution of small size control.

2-1. XC series PLC's products summary and program mode

Introduction

XC series programmable controller

- Input/output 14~60 points
- With FlashROM memory inside
- Real time clock：With clock inside，Li battery drop power memory。
- Basic instruction：27 types
  Applied instruction：89 types
- Has two communication ports，can connect with instruments and printers etc.，COM 2 can be RS-232 or RS-422
- With rich instructions, easy to program。

Program format

《List program》

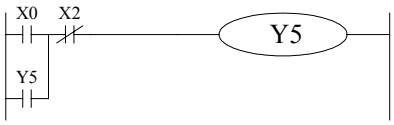
List program is the format which use "LD"、"AND"、"OUT" and other ordinal control instructions to input. This format is the basic input format to compile ordinal control program。

E.g. Step	Instruction	Soft unit's ID
0	LD	X000
1	OR	Y005
2	ANI	X002

《Ladder program》

Use ordinal control signal and soft unit's ID to draw the ordinal circuit's graph on the screen, it is called ladder program. As this method uses trigger point's symbols and coil symbols to denote the ordinal control circuit，so it is easy to understand the program's contents. At the same time, it's also available to monitor the PLC's action via the status displayed in the circuit。

E.g. ：

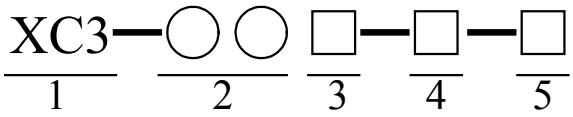


Alternation

The programs compiled with the preceding two methods are both stored in the PLC's program memory in the format of list. So，the denotion and edition of this two program format can convert to one another.

2-2．XC series PLC's model and type

XC series main unit

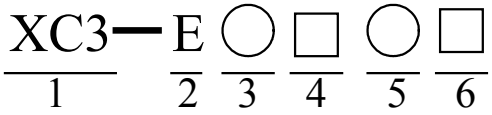


1、 Family name
2、 Total points of input and output
3、 Output format R：Relay output T：Transistor output
RT：Both relay and transistor output
4、 Supply power E：AC power C：DC power
5、 With clock inside S：With clock inside

Model				Input points（DC24V）	Output points（R，T）
AC power		DC power			
Relay output	Transistor output	Relay output	Transistor output		
XC3-14R-E	XC3-14T-E	XC3-14R-C	XC3-14T-C	8 points	6 points
XC3-24R-E	XC3-24T-E	XC3-24R-C	XC3-24T-C	14 points	10 points
XC3-32R-E	XC3-32T-E	XC3-32R-C	XC3-32T-C	18 points	14 points
XC5-48R-E	XC5-48T-E	XC5-48R-C	XC5-48T-C	28 points	20 points
XC5-60R-E	XC5-60T-E	XC5-60R-C	XC5-60T-C	36 points	24 points

NOTE：For the model which has both relay output and transistor output, only Y0 and Y1 are transistor output, the others are relay output.。

Digital extend module



1、 Family name
2、 E：Extend module
3、 Input points
4、 X：Exclusive used by input
5、 Output points
6、 Output format YR：Exclusive used by relay output
YT：Exclusive used by transistor output

Model			Total points	Input points（DC24V）	Output points（R，T）
Input	Relay output	Transistor output			
XC3-E16X	-	-	16 points	16 points	-
-	XC3-E16YR	XC3-E16YT	16 points	-	16 points
-	XC3-E8X8YR	XC3-E8X8YT	16 points	8 points	8 points

2-3．General specification

General specificatinon

Item	Specification
Insulation voltage	Up to DC 500V 2MΩ
Noise immunity	1000V 1uS pulse per minute
Ambient temperature	0℃~60℃
Ambient humidity	5~95%
COM 1	RS-232C，connect with host machine、HMI program or debug
COM 2	RS-232C/RS-485，connect with the net or aptitude instrument、inverter etc.
Installation	Screw fixed or orbit installation
Grounding	The third type ground（can't public ground with strong power system.）

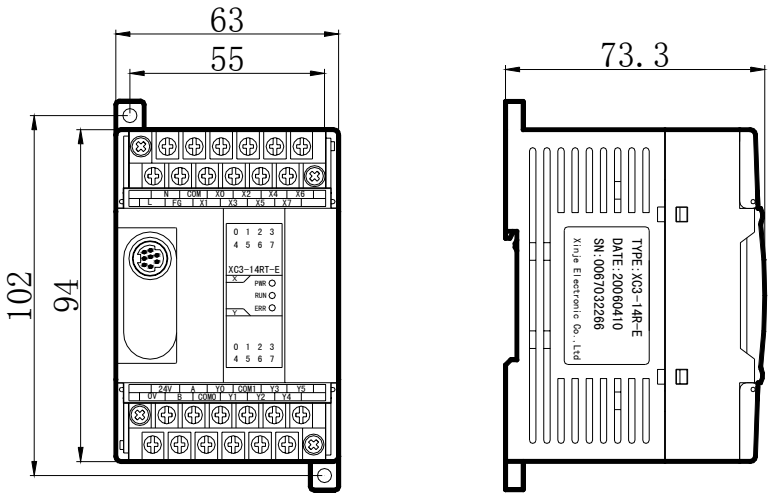
Specification

Item		Specification	
		14 points	**60 points**
Program executing format		Loop scan format、time scan format	
Program format		Use both list and ladder chart	
Instruction types		84 types	
Dispose speed		0.5uS	
Power cut retentive		Use FlashROM and Li battery	
User program's capacity		30000 steps（The program is smaller than 1M）	
I/O points		Input 36 points / output 24 points	
Interior coil's points (M)		6144 points	8000 points
Timer（T）	Points No.	640 points	
	Specification	100mS timer：The set time 0.1~3276.7 sec 10mS timer：The set time 0.01~327.67 sec 1mS timer：The set time 0.01~327.67 sec	
Counter（C）	Points No.	640 points	
	Specification	16 bits counter：The set value K0~32767 32 bits counter：The set value K0~2147483647	
Data register（D）		8000 words	
FlashROM register（D）		1792 words	5000 words
Special coil（M）		512 words	
Special register（D）		256 words	1000 words
High-speed counter/exterior interrupt		7 types format	
Setting of time scan space		0~255mS	
Password protection		8 bits length	
Self diagnose function		Self-diagnose、Monitor timer、grammar check	

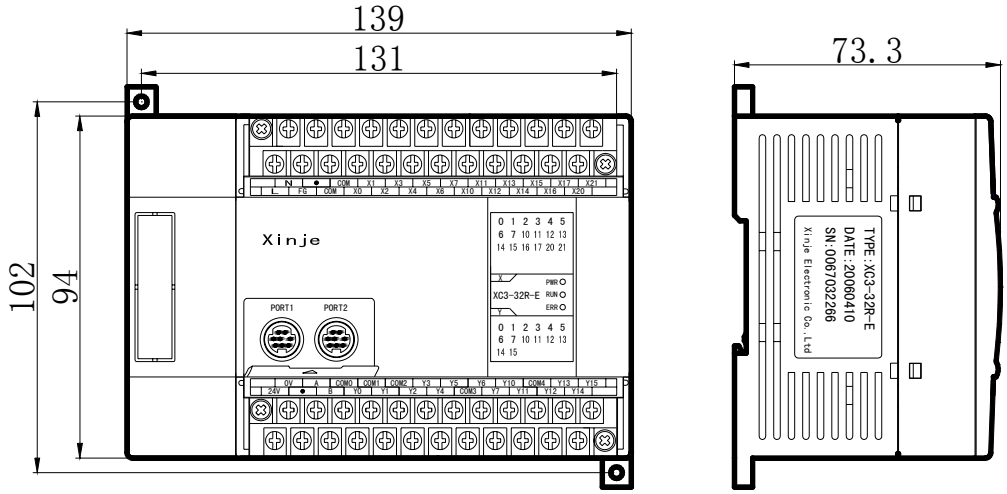
2-4．Size

Size

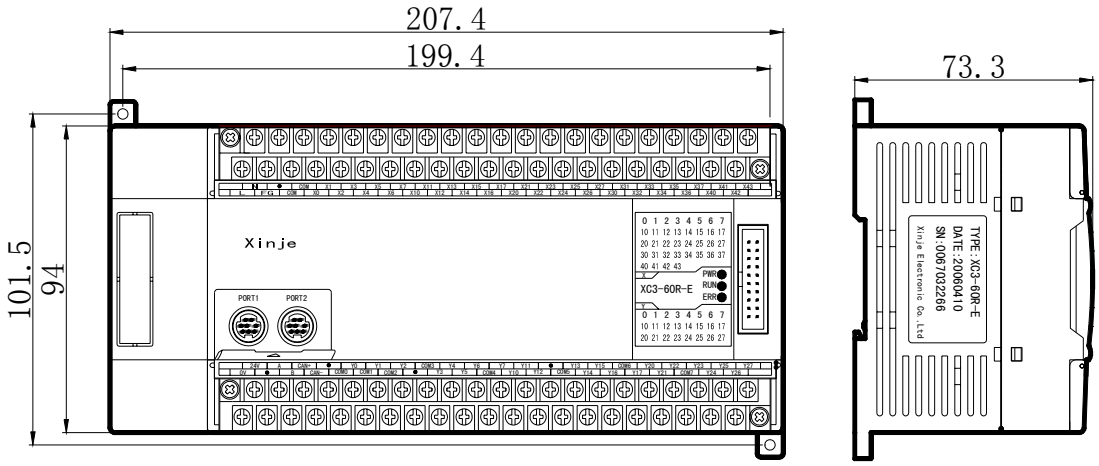
XC3 series 14 points main unit（including extension）



XC3 series 32 points main unit（Including 24 points、16 points main unit）

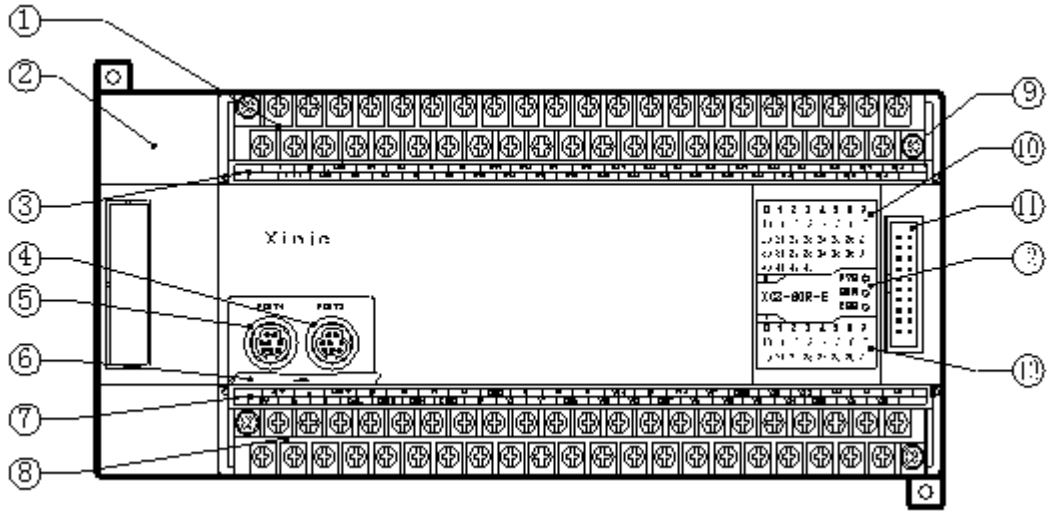


XC5 series 60 points main unit（Including 48 points main unit）



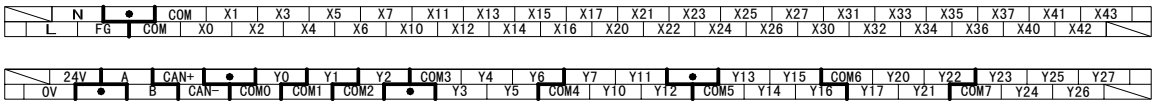
2-5. Terminal arrangement

Main unit's arrangement

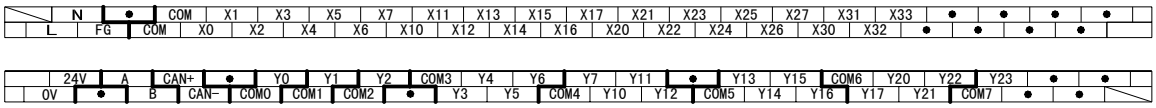


① Input terminals ② BD extension ③ Input label ④ COM2
⑤ COM1 (Program port) ⑥ COM ports' cover board ⑦ Output label
⑧Output terminals ⑨ Screws ⑩ Input indicate LED ⑪ Extension port
⑫Programming status indicate LED ⑬ Output indicate LED

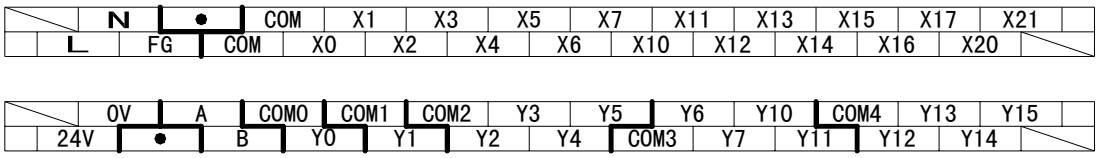
XC3- 60 main unit：36 in/24 out



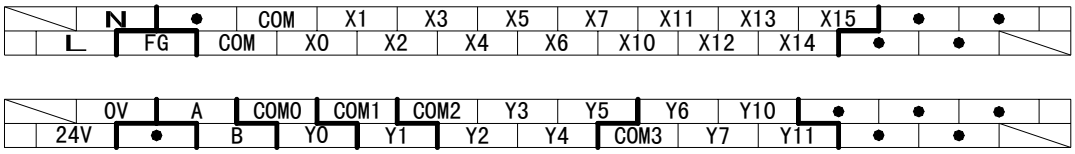
XC3- 48 main unit：36 in/24 out



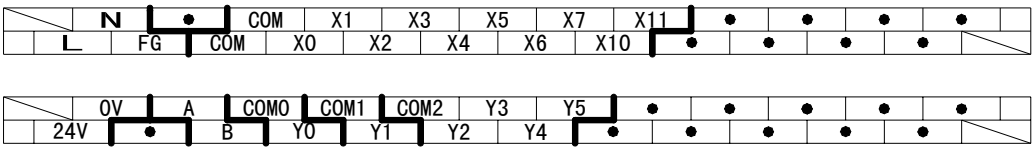
XC3- 32 main unit：18 in/14 out



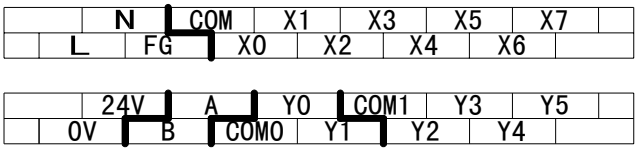
XC3- 24 main unit：14 in /10 out



XC3- 16 main unit：10 in /6 out

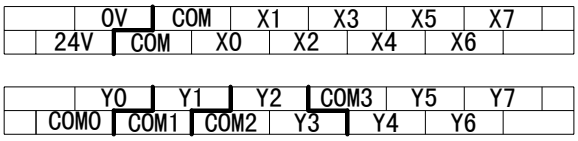


XC3- 14 main unit：8 in /6 out

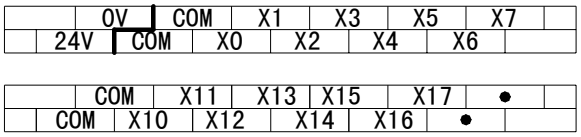


Arrangement of extension

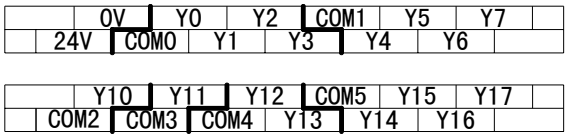
XC-E8X8YR



XC-E16X



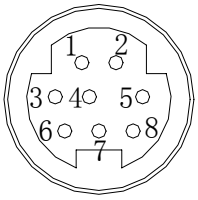
XC-E16YR



2-5. Definition of COM ports

COM 1

See the pin graph of COM 1：

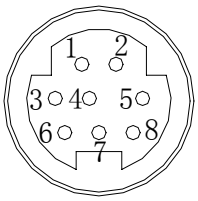


2：PRG
4：RxD
5：TxD
6：VCC
8：GND

Mini Din 8 core jack（hole）

COM 2

See the pin graph of COM 2：



4：RxD
5：TxD
8：GND

Mini Din 8 core jack（hole）

Program cable

See the following graph, which is the connection of program cable：

Mini Din 8 core jack（hole）　　DB9 pin（hole）

3. Power circuit's specification、input/output specification and exterior layout

In this chapter, we'll tell the power constitution, interior signal circuit's composing, output circuit's composing and exterior layout of XC series PLC.

When using the extend modules or special modules at the same time, please connect the power according to the user manual.

3-1. Power specification

For the power specification of XC series programmable controller's basic units, see the following table：

AC power type

Rated voltage	AC100V~240V
Voltage allow bound	AC90V~265V
Rated frequency	50/60Hz
Allow momentary power-cut time	Interrupt time≤0.5 AC cycle，alternation≥1 sec
Impulse current	Max 40A 5mS below/AC100V max 60A 5mS below /AC200V
Max power consumption	12W
Power for sensor use	24VDC±10% max 400mA

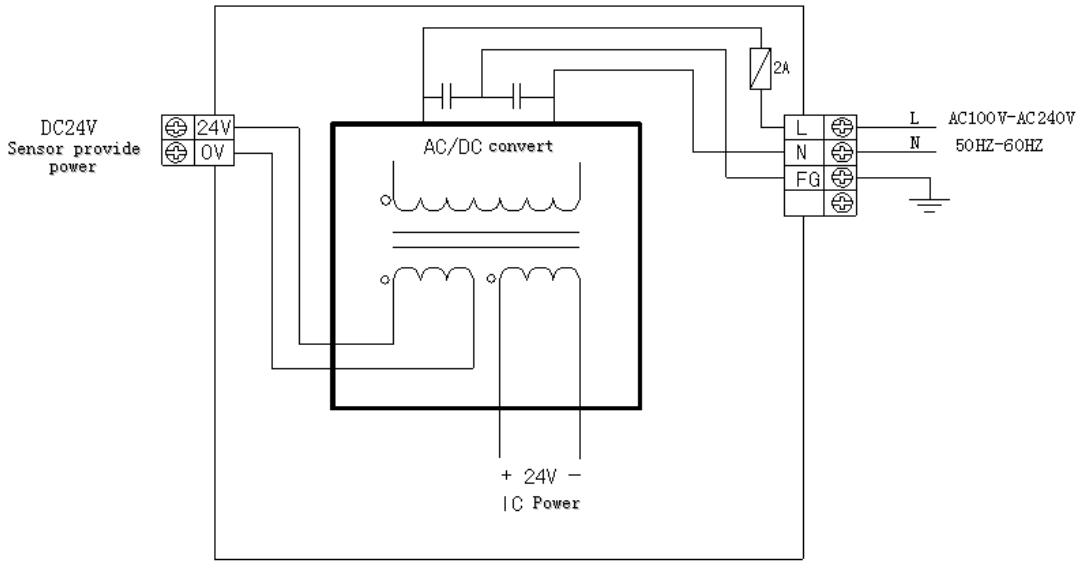
- To avoid voltage decend, please use the power cable thicker than 2mm^2
- Even appear power cut within 10ms，programmable controller can still continue to work. But if long time power cut or abnormal power decend, programmable controller will stop working, output will also appear OFF status，when recover power supply, the programmable controller will automatically start working.
- Connect the grounding terminals of basic unit and of extend module together, then ground

DC power type

Rated voltage	DC24V
Voltage allow bound	DC21.6V~26.4V
Input current (Only basic unit)	120mA DC24V
Allow momentary power-cut time	10mS DC24V
Impulse current	10A DC26.4V
Max power consumption	12W
Power for sensor use	24VDC±10% Max 400mA

3-2．AC power、DC input type

Composing and connection

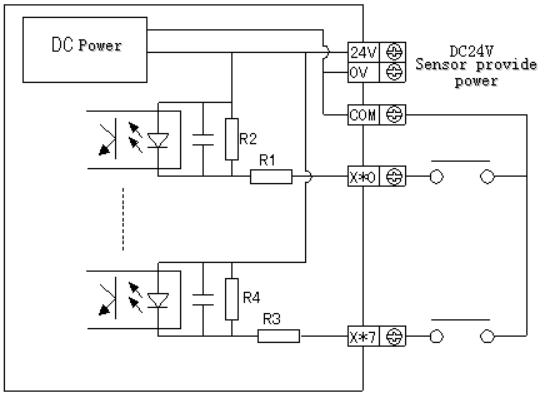


- The power is connected between L and N terminals.
- 24+、COM terminals can be used as power 400mA/DC24V which supply sensor. Besides, this terminal can't be given power from outside.
- [ ] terminal is vacant terminal，please don't go on exterior connection or use it as relay terminal.
- Please connect the basic unit with extend module's [COM] terminal.

3-3. Input specification

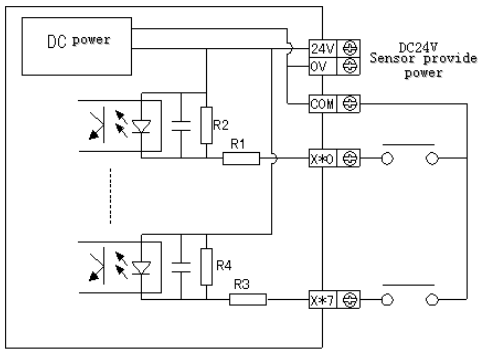
Basic unit

Input signal's voltage	DC24V±10%
Input signal's current	7mA/DC24V
Input ON current	Up to 4.5mA
Input OFF current	Low than 1.5mA
Input response time	About 10ms
Input signal's format	Contact input or NPN open collector transistor
Circuit insulation	Photoelectricity coupling insulation
Input action's display	LED light when input ON

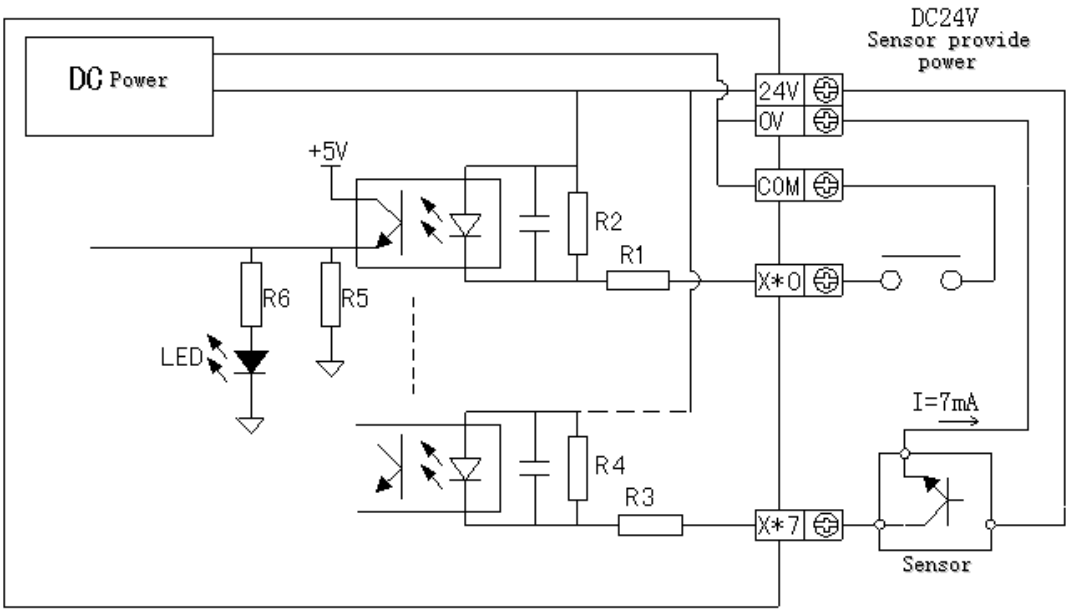


Extend unit

Input signal's voltage	DC24V±10%
Input signal's current	7mA/DC24V
Input ON current	Up to 4.5mA
Input OFF current	Below 1.5mA
Input response time	About 10ms
Input signal's format	Contacts input or NPN open collector transistor
Circuit insulation	Photoelectricity coupling insulation
Input action's display	LED light when input ON.



3-4．DC input signal's disposal（AC power type）



- Input terminals

  When connect input terminals and COM terminals with contacts without voltage or NPN open collector transistor, if input is ON，LED lamp lights, which indicates input. There are many COM terminals to connect in PLC.

- Input circuit

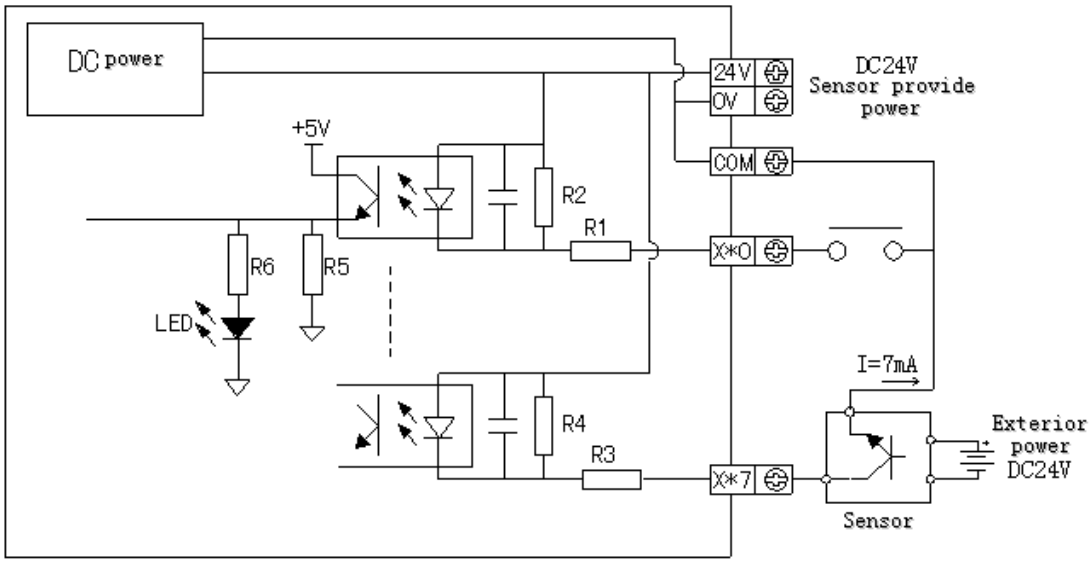
  Use optical coupling instrument to insulate the input once circuit and twice circuit. There's a C-R filter in the twice circuit. It is set to avoid wrong operation caused by vibration of input contacts or noise along with input signal. As the preceding reason, for the changing of input ON→OFF，OFF→ON，in PLC, the response time delays about 10ms. There's a digital filter inside X000~X015. This kind of filter can change from 0~15ms according to the special register (D8020).

- Input sensitive

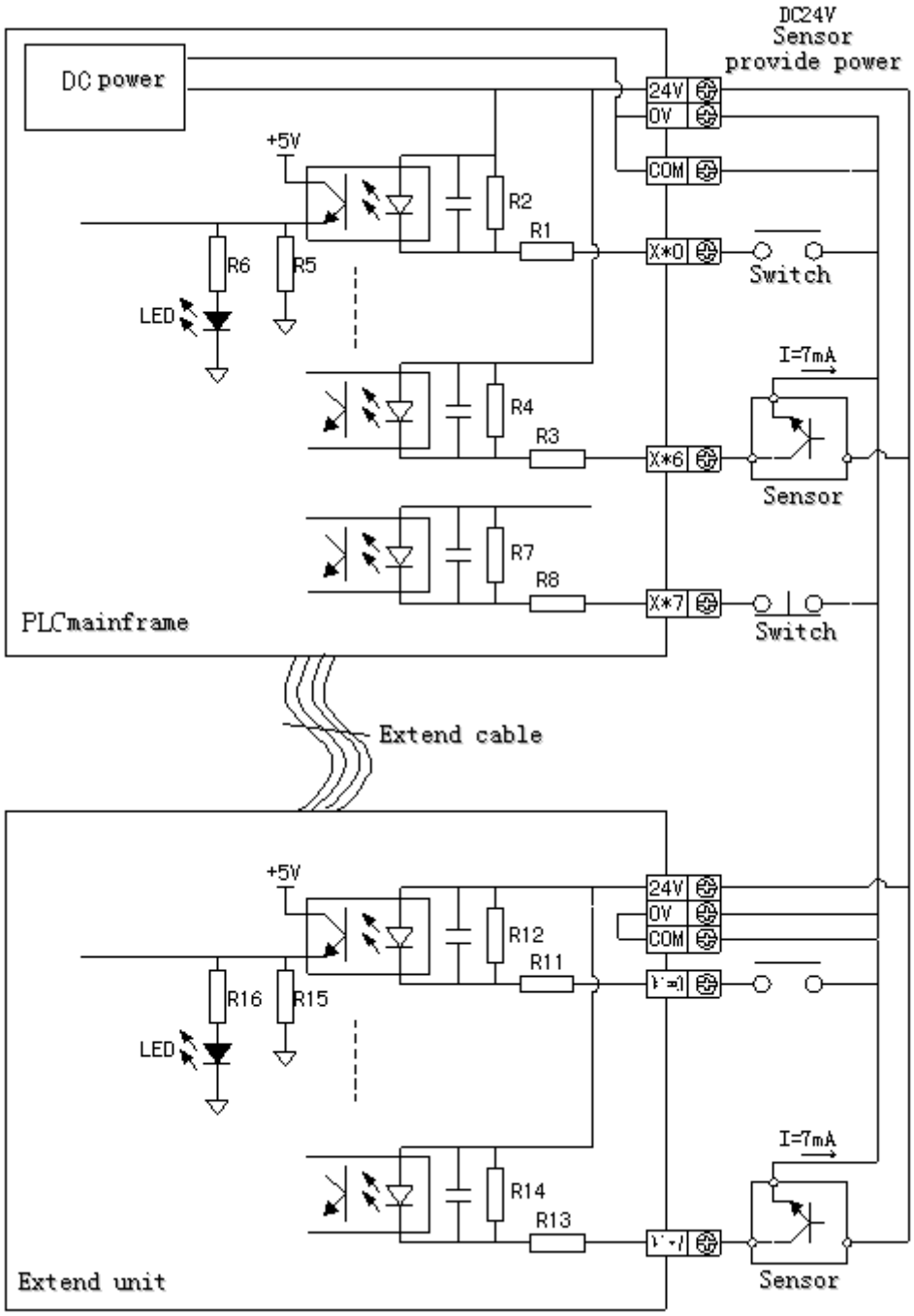
  The PLC's input current is DC24V　7mA，but to be safe，it needs current up to 3.5mA when it's ON，lower than 1.5mA when it's OFF.

Exterior circuit used by sensor

XC series PLC's input power is supplied by its interior 24V power，so if use exterior power to drive photoelectricity sensor etc.,，this exterior power should be DC24V±4V，please use NPN open collector type for sensor's output transistor. But in the sensors whose output transistor has strong pressure endurance ablilty，and there is no diode or resistance (open collector) as the following, it will be no problem even exterior voltage is not the same.



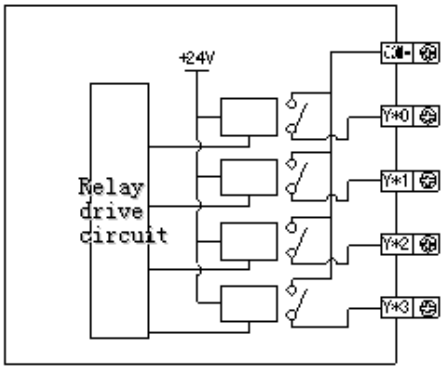
Input connection



3-5. Output specification

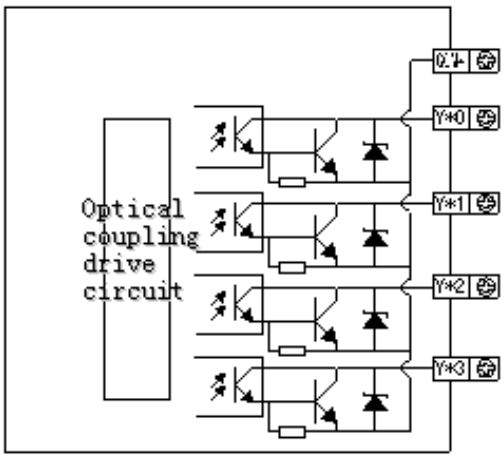
Relay output

Interior power		Below AC250V、DC30V
Circuit insulation		Mechanism insulation
Action denotion		LED indicate lamp
Max load	Restance load	3A
	Induce load	80VA
	Lamp load	100W
Open circuit's leak current		-
Mini load		DC5V 2mA
Response time	OFF→ON	10ms
	ON→OFF	10ms



Transistor output

Interior power		Below DC5~30V
Circuit insulation		Optical coupling insulation
Action denote		Indicate lamp LED
Max load	Restance load	0.8A
	Induce load	12W/DC24V
	Lamp load	1.5W/DC24V
Open circuit's leak current		-
Mini load		DC5V 2mA
Response time	OFF→ON	Below 0.2ms
	ON→OFF	Below 0.2ms

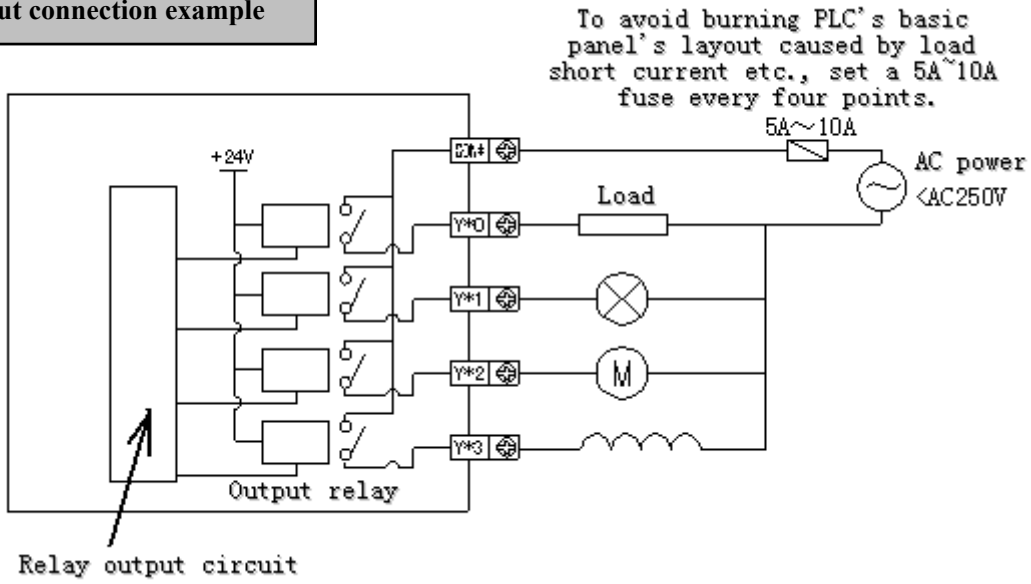


3-6．Disposal of relay output circuit

Relay output circuit

- **Output terminal**
  Relay output type includes 1~4 public-end output type. So each public-end unit can drive different power-voltage system's（E.g.：AC200V，AC100V，DC24V etc.）load.
- **Circuit's insulation**
  Between the relay output coils and contacts，PLC's interior circuits and exterior circuits, load circuits are electric insulation。Besides, each public-end blocks are separate.
- **Action display**
  LED lamp lights when output relay's coils galvanize，output contacts are ON。
- **Response time**
  From the output relay galvanize (or cut) to the output contacts be ON (or OFF), the response time is about 10ms
- **Output current**
  The current-voltage below AC250V can drive the load of pure resistace 2A/1 point、inductance load below 80VA（AC100V or AC200V） and lamp load below 100W（AC100V or AC200V）。
- **Open circuit's leak current**
  When the output contact be OFF and there's no leak current，can directly drive Ne lamp etc.。
- **The life of relay output contacts**
  Standard life of induce AC load such as contacter、electromagnetism valve：5 million times for 20VA load. Cut power device's life according to the company's test: for 80VA load, the action life is up to 2 million times.

Output connection example

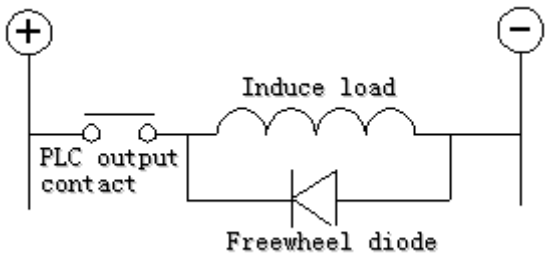


Constitution of output circuit

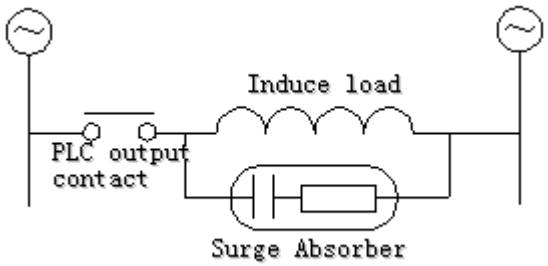
- For DC induce load, please parallel connect with commutate diode. If not connect with the commutate diode, the contact's life will be decreased greatly. Please choose the commutate diode which allow inverse voltage endurance up to 5~10 times of the load's voltage, ordinal current exceeds load current.

DC load

- Parallel connect AC induce load with surge absorber can reduce noise.



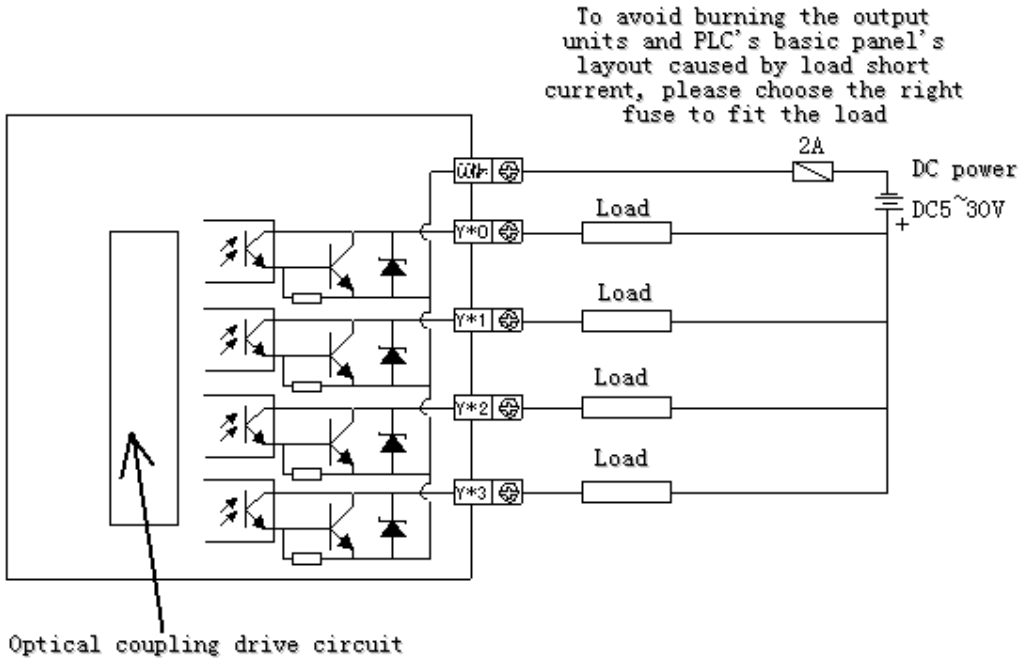
AC load



3-7. Disposal of transistor output circuit

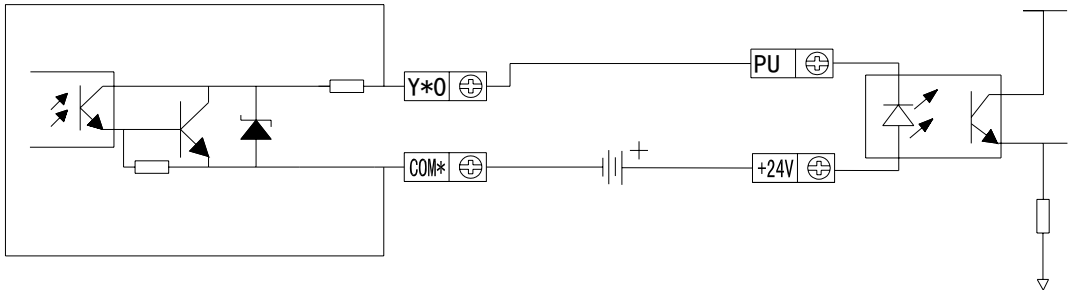
Transistor output circuit

- Output terminal
  Basic unit's transistor output has 1~4 public-end output。
- Exterior power
  Please use DC5~30V steady-voltage power for load drive。
- Circuit insulation
  Use photoelectricity coupling device to insulate PLC's interior circuit and output transistor. Besides, each public block is separate.
- Action denote
  When drive optical coupling，LED lights，output transistor is ON。
- Response time
  From photoelectricity coupling device drive (or cut) to transistor ON (or OFF) , the time PLC uses is below 0.2ms。
- Output current
  The current is 0.5A per point。But as restrict by temperature goes up, the current is 0.8A every four points。
- Open circuit's current
  Below 0.1mA。



E.g. ： The following is the connection graph of –RT series PLC with step motor driver。

PLC side　　Step motor driver side



4．Each soft unit’s usage and function

This chapter, we’ll give some description of the PLC’s data and the function of interior input/output relay, auxiliary relay, status, counter, data register etc. This item is the base to use PLC.

4-1. Every soft unit of programmable controller

In the programmablecontroller, there are many relays、timers and counters，they all have countless a contacts（Normally open contacts）and b contacts（Normally closed contacts），Connect these contacts and cols to constitute sequencial control circuit。The following, we'll briefly introduce each soft unit：

【Input（X）and output（Y）relay】

- In each basic unit，assign the ID of input relay, output relay in the format of X000~X007，X010~X017…,Y000~Y007，Y010~Y017… this octal format. The ID of extension is connected behind basic unit.
- Use digital filter in the special input filter of input relay, so you can use the program to change the sieve value. So in the high-speed receive application, you can assign this type of relay's ID No.。

【Auxiliary relay（M）】

- Auxiiary relay is the relay inside the programmable controller，this type of output relay is different from input/output relay，it can't gain exterior input，it also can't drive exterior load，it can only be used in the program。
- The relay used for retentive can still save its ON/OFF status in the case of PLC power cut.

【Status（S）】

- Relay used as step ladder chart。
- When not used as working procedure No.，it's the same with auxiliary relay and can be used as common contact/coil to carry on programming。Besides, it can also be signal alarm to diagonise exterior trouble.

【Timer（T）】

- Timer could carry on plus operation to 1ms，10ms，100ms etc. time pulse in PLC, When reach certain set value, output contact act.
- T100~T199 are timers with the unit of 100ms clock pulse，their current values are the accumulate values. So, even though timer coil's drive input is cut, they will still hold the current value, go on accumulating the action.

【Counter（C）】

- The counters can be divided to the following sorts according the their usage and purpose：

  [Used for interior count] Common use / power cut retentive use

  16 bits counter：Used for add count，count bound: 1~32，767

  32 bits counter：Used for add / minus count，count bound: -2，147，483，648~+2，147,483,647

  These counters are used for PLC's interior signal，usually their response speed is below 10Hz。

  [Used for high-speed count] For power cut retentive use

  32 bits counter：For add / minus count， count bound: -2，147，483，648~+2，147,483,6487

  (Single phase single count，single phase double count，AB phase count) allocate to the special input relay。High-speed counter can count with the frequency below 60kHz，independent with the PLC's scan time.

【Data register（D）】

- It's the soft unit used by data register to save data。XC series PLC's data registers are all 16 bits (The high bit is the sigh bit), Combine two registers can carry on 32 bits data disposal (The high bit is the sigh bit).

  Just the same with other soft unit, data registers also can be divided to be two types: forcommon use and power cut retentive use。

【Constant（K）】 (H)

- In the diverse value used by PLC, K means decimal integer，H means Hex. Value. They are used to be the set value and current value for the timer and counter, or application instructions' operands.

【Pointer（P）(I)】

- Pointer is used for branch and interrupt。The pointer (P) used by branch is the jump aim used for condition jump or subroutine jump. Pointer used for interrupt is used for the assigned input interrupt, time interrupt.

4-2．Device’s ID list

For the allocate of device’s ID., please see the following list：
Besides, when connect input / output extend device and special extend device on the basic units, for the input / output relay’s No., please refer to the user manual.

<table>
<tr><th rowspan="2">Mnemonic</th><th rowspan="2">Name</th><th colspan="2">Bound</th><th colspan="2">Point</th></tr>
<tr><th>14 points</th><th>60 points</th><th>14 points</th><th>60 points</th></tr>
<tr><td>X</td><td>Input coil</td><td colspan="2">X000~X512</td><td>512</td><td>512</td></tr>
<tr><td>Y</td><td>Output coil</td><td colspan="2">Y000~Y512</td><td>512</td><td>512</td></tr>
<tr><td rowspan="2">M</td><td rowspan="2">Interior relay</td><td>M0~M2999
【M3000~M6144】</td><td>M0~M2999
【M3000~M7999】</td><td>6144</td><td>8000</td></tr>
<tr><td colspan="2">For special usage M8000~M8511</td><td>512</td><td>512</td></tr>
<tr><td>S</td><td>Flow</td><td colspan="2">S0~S511
【S512~S1023】</td><td>1024</td><td>1024</td></tr>
<tr><td rowspan="8">T</td><td rowspan="8">Timer</td><td colspan="2">T0~T99：100ms not accumulation</td><td rowspan="8">640</td><td rowspan="8">640</td></tr>
<tr><td colspan="2">T100~T199：100ms accumulation</td></tr>
<tr><td colspan="2">T200~T299：10ms not accumulation</td></tr>
<tr><td colspan="2">T300~T399：10ms accumulation</td></tr>
<tr><td colspan="2">T400~T499：1ms not accumulation</td></tr>
<tr><td colspan="2">T500~T599：1ms accumulation</td></tr>
<tr><td colspan="2">T600~T619：1ms with interruption precise time</td></tr>
<tr><td rowspan="3">C</td><td rowspan="3">Counter</td><td colspan="2">C0~C299：16 bits forth counter</td><td rowspan="3">640</td><td rowspan="3">640</td></tr>
<tr><td colspan="2">C300~C599：32 bits forth/back counter</td></tr>
<tr><td colspan="2">C600~C635：high-speed counter</td></tr>
<tr><td rowspan="2">D</td><td rowspan="2">Data register</td><td colspan="2">D0~D3999
【D4000~D7999】</td><td>8000</td><td>9000</td></tr>
<tr><td>For special usage D8000~D8255</td><td>For special usage D8000~D9000</td><td>256</td><td>1000</td></tr>
<tr><td>FD</td><td>FlashROM register</td><td>FD0~FD1791</td><td>FD0~FD4999</td><td>1792</td><td>5000</td></tr>
</table>

◆ **NOTE：**

※1. The memorizer area in 【 】 is the defaulted power-cut rententive area；soft element D、M、S、T、C can be set to change the power-cut rententive area。Fot the details, please see the following table

※2. FlashROM register needn't to set power cut rententive, its data won't lose when power is cut (No battery).

※3. The serial No. of input coil、output relay are octal data，other memorizers' No. are all algorism data。

※4. The exterior device which is not connected with I/O can be used ad fast-speed interior relay。

Setting of soft unit's power cut save area

Mnemonic	Set area	Function	System's defaulted value
D	FD8202	Start denotion of D power cut save area	4000
M	FD8203	Start denotion of M power cut save area	3000
T	FD8204	Start denotion of T power cut save area	640
C	FD8205	Start denotion of C power cut save area	320
S	FD8206	Start denotion of S power cut save area	512

4-3．Data disposal of programmable controller

According to different usage and purpose, XC series programmable controllers use 5 types of format. For its usage and function, see the following：

《DEC》（DEC：DECIMAL NUMBER）

- The set value of timer and counter（K constant）
- The number of auxiliary relay（M），timer（T），counter（C），status（S）（Soft unit's number）
- Assign the value in the operands and nstruction's action（K constant）

《HEX》（HEX：HEXADECIMAL NUMBER）

- The same with DEC data，it is used to assign the value in the operands and instruction's action（H constant）

《BIN》（BIN：BINARY NUMBER）

- Just as said before，carry on data allocation to timer, counter or data register in the format of DEC. or Hex., But in the PLC, these data are all be put in the format of binary data。And, when carry on monitor on the periphery device, these soft units will auto switch to be DEC. data as showed in the graph.（they can also switch to be Hex. Data.）。

《OCT》（OCT：OCTAL NUMBER）

- The input relay, output relay's soft units' No. of XC series PLC are allocate in the format of OCT data. So, it can go on carry of [1-7，10-17，. . . 70-77，100-107]。

《BCD code》（：BINARY CODE DECIMAL）

- BCD is the method which use 4 bits binary to denote decimal 0~9. It's easy to despose bit. So, BCD is available to denote digital switch or 7 segments display control.

《Other data（float）

- XC series PLC has the function of high precision floating point operation. Use binary floating point data to execute floating point operation, use decimal floating value to execute monitor.

4-4．Some encode principles of device

- **Data register can be used as offset**

  Format：Dn[Dm]、Xn[Dm] 、Yn[Dm] 、Mn[Dm] etc.。

  E,g,：D0[D1]，value in D1 is 12，so D0[D1] means D12；

  X4[D1]，value in D1 is 12，so X4[D1] means X16

  The detailed meaning of T11[D1] should distinguish according to the instruction:

  Word offset made with bit soft units：DXn[Dm]：DX[n+Dm]；

  The soft unit with offset，the offset can only denote with soft unit D.

- **Bit units compose words**

  Input X、output Y、middle coil M，can compose a 16 bits word. E.g. DX0 means X0~X17 compose a 16 bits data，DX20 means X20~X37 compose a 16 bits data.

  Format：Add D before bit soft units

  Bit units combine to be words：DX、DY、DM、DS、DT、DC

  DXn（The bound of n is the bound of X），adopt 16 points in the back, add 0 if not enough

  E.g.：DX1，means the word of X1~X20

  ※Bit units with offset combine to be words

  Format：Add D before bit soft units with offset

  E.g.：DX128[D123]，means a word starts with X128[D123]

  Note：The word combined with bit soft units can not carry on bit seek address。

- **The bit of word soft unit**

  Format：Dn.m

  Register can carry on bit seek address，e.g.: Dn.m means the No.m bit of data register（0≤m≤15）。

  E.g.：D123.11，means the No.11 bit of D123

  ※The bit of word soft units with offset

  Format：Dn[Dm].x

  E.g.：if D123's value is 5，then "D12[D123].11" means the No.11 bit of D（12+5）。

  NOTE：For the word soft unit's bits，they can't be combined to be word soft units

- **T/C means the difference of register's word and bit**

  For T and C register，Tn/Cn means it needs to distinguish if it's a bit register or a word register according to the instruction.

  T、C can be used to denote timer、counter's status，or used to denote timer、counter's current value，this will be distinguished by the instruction。

  E.g.：MOV T11 D12，T11 means word register；LD T11，T11 means bit register

- **Tag type：P，I**

  E.g.：P7，means tag 7，use CJ command or CALL command, the program will jump to the tag place to execute，but the jump bound is limited in the step state of the most inner layer.

- **Attentaion items**

（1）After M8000 、D8000、FD8000 are all used by the system.

（2）The constant which the user inputs will auto convert according to the user's command （16bits integer—>32 bits integer and 32 bits float、32 bits integer—>32 bits float）

（3）DW: D0 is combined by D0、D1，and D1 is the high word、D0 is the low word
DM1 is a 16 bits data composed by M1~M16，and M16 is the highest bit（MSB）、M1 is the lowest bit（LSB）。

4-5.Timer's number and function [T]

Timer's ID

For the number of timer [T], please see the following table。（Allocate the No. in the format of DEC.）

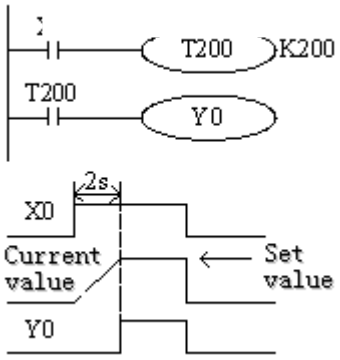
100ms not accumulate（16 bits）	T0~T99
100ms accumulate（16 bits）	T100~T199
10ms not accumulate（16 bits）	T200~T299
10ms accumulate（16 bits）	T300~T399
1ms not accumulate（16 bits）	T400~T499
1ms accumulate（16 bits）	T500~T599
1ms accurt time with interrupt（32 bits）	T600~T619 (T600,T602....T618) (Each one engross two timer No.) The number must be even.

Function

The timer accumulates the clock pulse of 1ms，10ms，10ms in PLC. Output the contact's action when reach the set value.

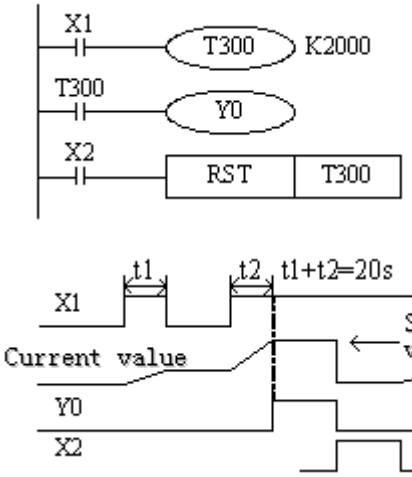
For the common timer, don't set the exclusive instructions, use OUT instruction to time; Use constant K in the program memory to be the set value, or use content in data register (D) to indirect assign.

Normal



If the drive input X000 of timer's coil T200 is ON, T200 accumulates 10ms clock pulse with the current value counter. When the value equals the set value K200, timer's output contact acts. I.e. output contact acts 2 seconds after coil drive. Drive input X000 cut or drop power, timer reset, output contact reset.

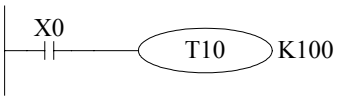
Accumulation



If the drive input X001 of timer's coil T300 is ON, T300 accumulates 10ms clock pulse with the current value counter. When the value reach the set value K2000, counter's output contact acts. In the count process, even the input X001 cut or drop power, when start again, go on counting, its accumulation time is 20 seconds. When reset input X002 is ON, timer reset, output contact reset.

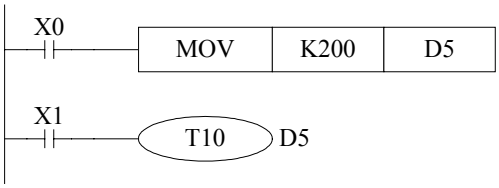
Assign method of the set value

《Constant assignment (K)》



T10 is a timer with the unit of 100ms. Assign 100 as a constant, then 0.1s×100=10s timer work.

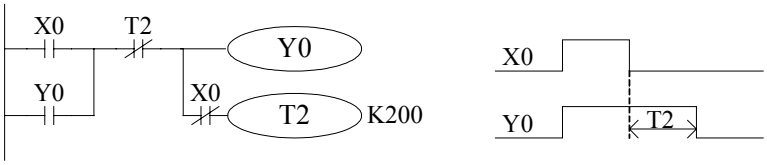
《Indirect assignment (K)》



Write content in indirect data register to program or input via data switch.
When assigned as power cut rententive register, please note that voltage low will cause the set value instable.

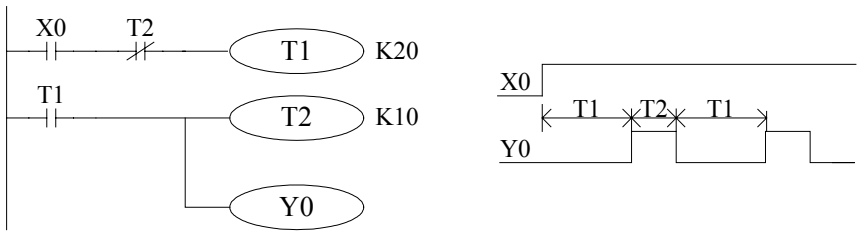
Action

《Output delay on-off timer》



When X000is ON, output Y000；
When X000 changes from ON to OFF, delay T2（20 seconds） time， cut when output Y000.

《Flicker》



After X000 closed, Y000 starts to output in flicker.

T1 control the cut time of Y000, T2 control the close time of Y000.

4-6.Counter's ID and function [C]

Counter's ID

For the counter's number（C）, please see the following table。

16 bits sequencial counter	C0~C299
32 bits sequencial counter	C300~C599 (C300,C302...C598)(Each one engrosses 2 counter No.) The number must be even
High speed counter	C600~C635 (C600,C602...C634) (Each one engrosses 2 counter No.) The number must be even

Counter's characteristic

The characters of 16 bits counter and 32 bits counter are the following. They could be used separately according to the switch of the count direction and the use condition of count bound.

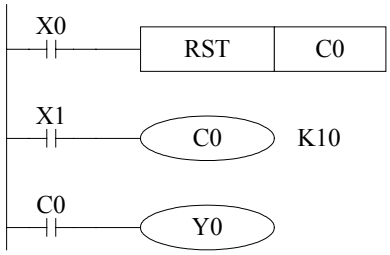
Item	16 bits counter	32 bits counter
Count direction	Count forward	Count forward
The set value	1~32，767	0~+2,147,483,647
The assigned set value	Constant K or data register	Same as the left but data register must be in a couple
The change of the current value	Count forward and then change	Count forward and then change（Loop counter）
Output contacts	Count forward and then rententive	Count forward and then rententive, reset when count backward
Reset action	When executing RST command，counter's current value is 0，output contacts recover	
Current value's register	16 bits	32 bits

Function

The assignment of normally use counter and power cut rententive counter could be changed via modifting the peripheral equipment's parametre

16 bits binary increase counter, its valid setting value is K1~K32,767(Decimal constant). The set value K0 and K1 have the same meaning, i.e. act when output contacts at the beginning of first time count.

16 bits counter For normally use or power cout rententive

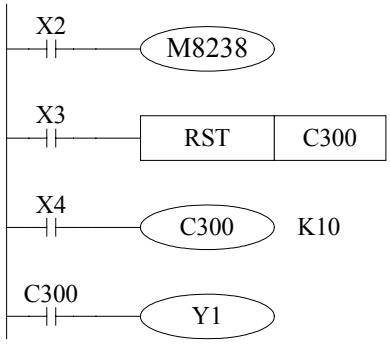


If cut the PLC's power, then the value of the normally use counter will be cleared. However, counter used by power cut rententive could save the count value after power cut, and the counter will go on counting from the value.

- Every time when X001 drives coil C0, the counter's current value will increase. When execute the coil instruction the tenth time, output contact acts. Later, even X001 acts, counter's current value will not change.
- If reset input X000 is ON, execute RST instruction, counter's current value is 0, output contact acts.
- For the counter's set value, it could not only set by constant K, but also be assigned by data register's ID. E.g. assign D10, if the content of D10 is 123, it's the same with setting K123.
- When write the set value to the current value register via MOV instruction etc. When input next time, output coil gets, current value register turns to the set value.

32 bits counter For normally use or power cout rententive

For 32 bits binary increase counter, its valid bound is K1~K2，147，483，647 (Decimal constant).



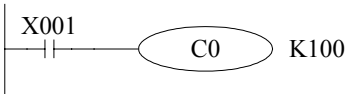
- Its function and action is the same with 16 bits increase counter. Set the value positive according to constant K or content in data register D. See contents in consecutive data register as a couple, and dispose it as 32 bits data. So, when assign D0, dispse D0 and D1 as a 32 bits set data. Use counte input X004 to drive coil C300 and execute increase count.

- When reset input X3 is ON, execute RST instruction, counter's current value turns to be 0, output contact resets.
- When use counter as power cut rententive, counter's current value, output contact's action and reset status cut power rententive.
- 32 bits counter can also be used as 32 bits data register. But 32 bits data register can't be used as device in 16 bits applied instructions.

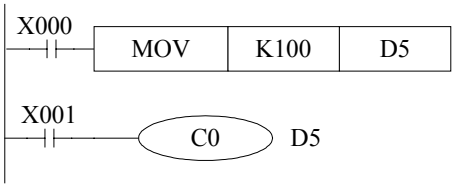
Assign method of the set value

◆ **16 bits counter**

《Constant assignment（K）》



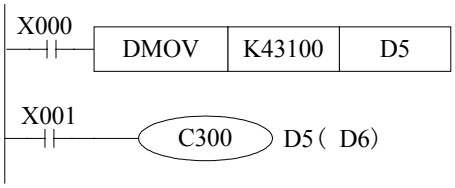
《Indicate assignment（K）》



◆ **32 bits counter**

《Constant assignment（K）》

《Indicate assignment（K）》



4-7. Some points to note

《Action order of input/output relay and response delay》

- Input disposal

  Before PLC executing the program, read all the input terminal's ON/OFF status of PLC to the image area. In the process of executing the program, even the input changed, the content in the input image area will not change. However, in the input disposal of next scan cycle, read out the change.

- Output disposal

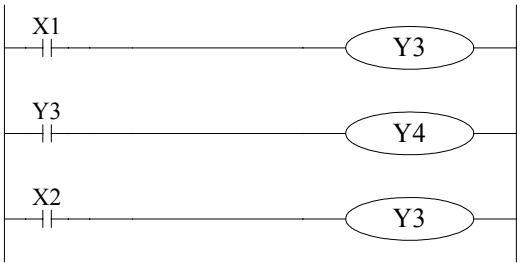
  Once finish executing all the instructions, transfer the ON/OFF status of output Y image area to the output lock memory area. This will be the actual output of the PLC. The contacts used for the PLC's exterior output will act according to the device's response delay time.

  When use this input/output format in a batch, the drive time and operation cycle of input filter and output device will also appear response delay.

《Not accept narrow input pulse signal》

PLC's input ON/OFF time should be longer than its loop time. If consider input filter's response delay 10ms, loop time is 10ms，then ON/OFF time needs 20 ms separately. So, up to 1，000/（20+20）=25Hz input pulse can't be disposed. But, this condition could be improved when use PLC's special function and applied instructions.

《Dual output（Dual coils）action》



When executing dual output (use dual coil),the back side act in prior

As showed in the left map, please consider the things of using the same coil Y003 at many position:
E.g. X001=ON，X002=OFF

At first, X001 is ON,its image area is ON, output Y004 is also ON.

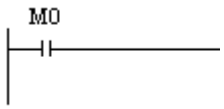
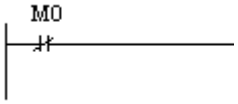
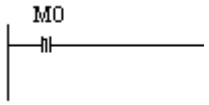
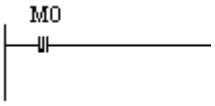
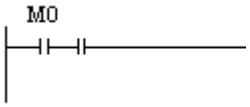
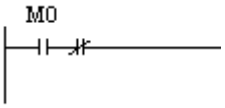
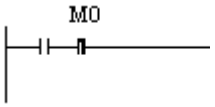
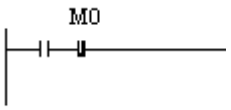
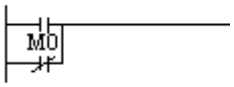
But, as input X002 is OFF, the image area of Y003 is OFF.

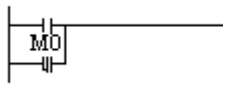
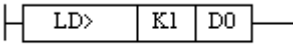
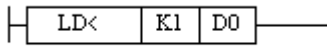
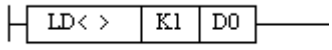
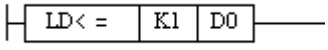
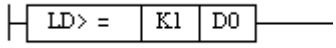
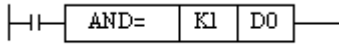
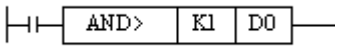
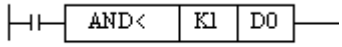
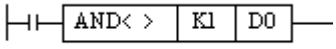
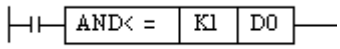
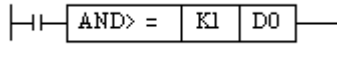
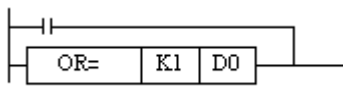
So, the actual output is : Y003=OFF, Y004= ON.

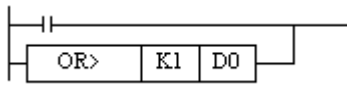
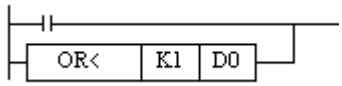
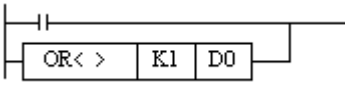
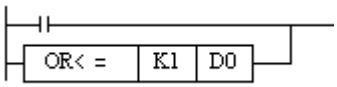
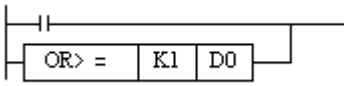
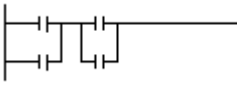
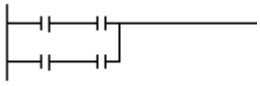
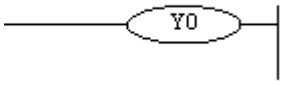
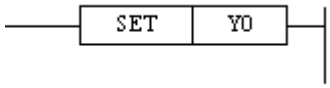
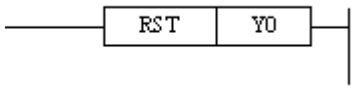
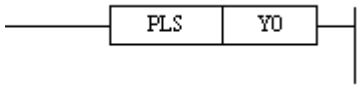
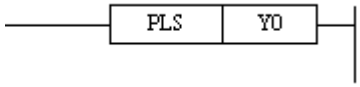
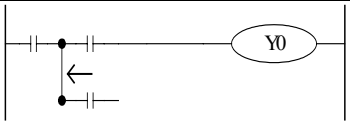
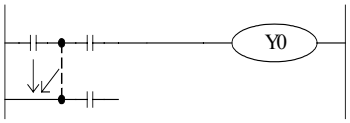
5. Basic program instructions

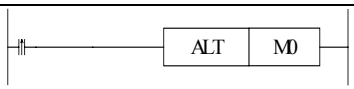
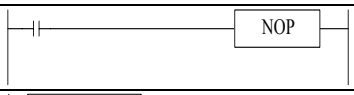
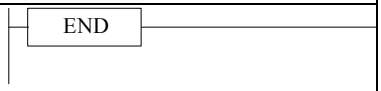
In this chapter, we give some basic instructions and their functions.

5-1. List of basic instructions

Mnemonic	Function	Format and device
LD (LoaD)	Initial logical operation contact type NO (normally open)	
LDI (LoaD Inverse)	Initial logical operation contact type NC (normally closed)	
LDP (LoaD Pulse)	Initial logical operation-Rising edge pulse	
LDF (LoaD Falling Pulse)	Initial logical operation-Falling /trailing edge pulse	
AND (AND)	Serial connection of NO (normally open) contacts	
ANI (AND Inverse)	Serial connection of NC (normally closed) contacts	
ANDP (AND Pulse)	Serial connection of rising edge pulse	
ANDF (AND Falling pulse)	Serial connection of falling/trailing edge pulse	
OR (OR)	Parallel connection of NO (normally open) contacts	
ORI (OR Inverse)	Parallel connection of NC (normally closed) contacts	
ORP (OR Pulse)	Paralllel connection of rising edge pulse	

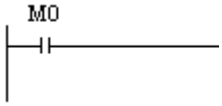
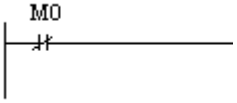
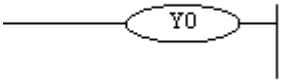
ORF (OR Falling pulse)	Parallel connection of falling/trailing edge pulse	
LD＝	Initial comparision contact. Active when the comparison （S1）＝（S2） is true.	
LD＞	Initial comparision contact. Active when the comparison （S1）>（S2） is true	
LD＜	Initial comparision contact. Active when the comparison （S1）<（S2） is true	
LD＜＞	Initial comparision contact. Active when the comparison （S1）≠（S2） is true	
LD＜＝	Initial comparision contact. Active when the comparison （S1）≤（S2） is true	
LD＞＝	Initial comparision contact. Active when the comparison （S1）≥（S2） is true	
AND＝	Serial comparison contact. Active when the comparison （S1）＝（S2） is true.	
AND＞	Serial comparison contact. Active when the comparison （S1）>（S2） is true.	
AND＜	Serial comparison contact. Active when the comparison （S1）<（S2） is true.	
AND＜＞	Serial comparison contact. Active when the comparison （S1）≠（S2） is true.	
AND＜＝	Serial comparison contact. Active when the comparison （S1）≤（S2） is true.	
AND＞＝	Serial comparison contact. Active when the comparison （S1）≥（S2） is true.	
OR＝	Parallel comparison contact. Active when the comparison （S1）＝（S2） is true.	

OR＞	Parallel comparison contact. Active when the comparison （S1）＞（S2）is true.	
OR＜	Parallel comparison contact. Active when the comparison （S1）＜（S2）is true.	
OR＜＞	Parallel comparison contact. Active when the comparison （S1）≠（S2）is true.	
OR＜＝	Parallel comparison contact. Active when the comparison （S1）≤（S2）is true.	
OR＞＝	Parallel comparison contact. Active when the comparison （S1）≥（S2）is true.	
ANB (ANd Block)	Serial connection of multiply parallel circuits	
ORB (OR Block)	Parallel connection of multiply parallel circuits	
OUT (OUT)	Final logic operation type coil drive	
SET (SET)	Set a bit device permanently ON	
RST (ReSeT)	Reset a bit device permanently OFF	
PLS (PuLSe)	Rising edge pulse	
PLF (PuLse Falling)	Falling/trailing edge pulse	
MCS (New bus line start)	Connect the public serial contacts	
MCR (Bus line return)	Clear the public serial contacts	

ALT (Alternate state)	The status of the assigned deviceis inverted on every operation of the instruction	
NOP (No Operation)	No operation or null step	
END (END)	Force the current program scan to end	

5-2.【LD】,【LDI】,【OUT】 instructions

Mnemonic and function

Mnemonic	Function	Format and device X,Y,M,S,T,C
LD (LoaD)	Initial logic operation contact type NO (Normally Open)	
LDI (LoaD Inverse)	Initial logic operation contact type NC (Normally Closed)	
OUT (OUT)	Final logic operation type drive coil	

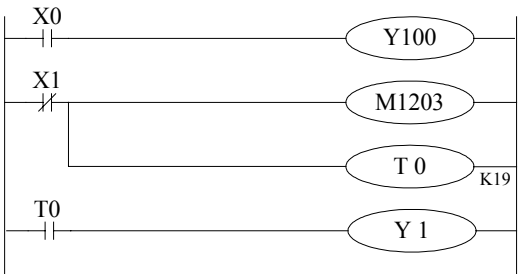
Instruction description

- Connect the LD and LDI instructions directly to the left bus bar. Or use them to define a new block of program when using ANB instruction.
- OUT instruction is the coil drive instruction for the output relay、auxiliary relay、status、timer、counter. For the input relay, cannot use.
- Can not sequentially use parallel OUT command for many times.
- For the timer's time coil or counter's count coil, after using OUT instruction, set constant K is necessary.
- For the constant K's set bound、actual timer constant、program's step relative to OUT instruction（include the set value）

See the following table：

Timer/Counter	Setting constant K	Actual setting	Program steps
1ms timer	1～32，767	0.001～32.767 sec	3
10ms timer		0.01～32.767 sec	3
100ms timer		0.1～32.767 sec	3
16 bits counter	1～32，767	Same as the left	3
32 bits counter	1～＋2，147，483，647	Same as the left	5

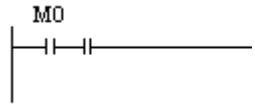
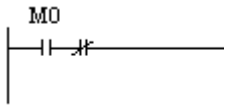
Program




```
0   LD     X0
1   OUT    Y100
2   LDI    X1
3   OUT    M1203
4   OUT    T0
    SP     K19
7   LD     T0
8   OUT    Y1
```

5-3【AND】,【ANI】 instructions

Mnemonic and function

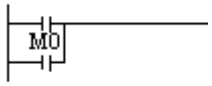
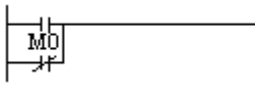
Mnemonic	Function	Format and device X,Y,M,S,T,C
AND (AND)	Serial connection of NO (Normally Open) contacts	
ANI (AND Inverse)	Serial connection of NC (Normally Closed) contacts	

5-4.【OR】,【ORI】 instructions

Instruction description

- Use the AND and ANI instructions for serial connection of contacts. As many contacts as required can be connected in series. They can be used for many times.
- The output processing to a coil, through writing the initial OUT instruction is called a "follow-on" output（For an example see the program below: OUT M10 and OUT Y005）. Follow-on outputs are permitted repeatedly as long as the output order is correct. There's no limit for the serial connected contacts's No. and follow-on outputs' No. .

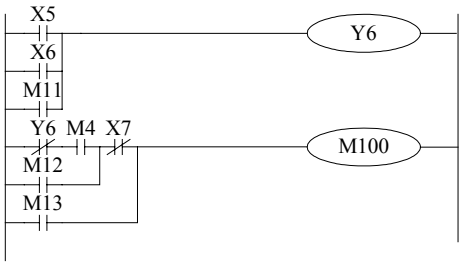
Mnemonic and function

Mnemonic	Function	Format and device X,Y,M,S,T,C
OR (OR)	Parallel connection of NO (Normally Open) contacts	
ORI (OR Inverse)	Parallel connection of NC (Normally Closed) contacts	

Instruction description

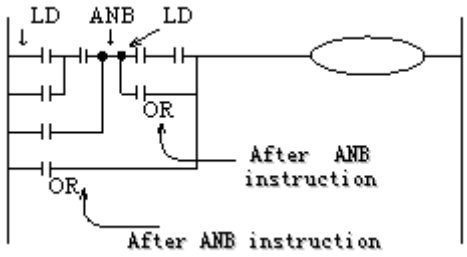
- Use the OR and ORI instructions for parallel connection of contacts. To connect a block that contains more than one contact connected in series to another circuit block in parallel, use an ORB instruction.
- OR and ORI start from the instruction's step, parallel connect with the LD and LDI instruction's step said before. There is no limit for the parallel connect times.

Program



0	LD	X5
1	OR	X6
2	OR	M11
3	OUT	Y6
4	LDI	Y6
5	AND	M4
6	OR	M12
7	ANI	X7
8	OR	M13
9	OUT	M100

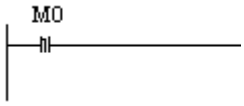
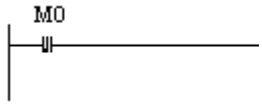
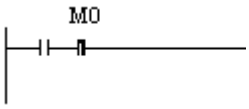
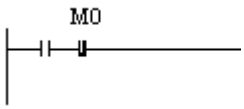
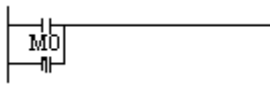
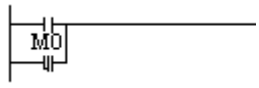
The relationship with ANB instruction



The parallel connection with OR,ORI instructions should connect with LD, LDI instructions in principle. But after the ANB instruction, it's available to add a LD or LDI instruction.

5-5.【LDP】,【LDF】,【ANDP】,【ANDF】,【ORP】,【ORF】

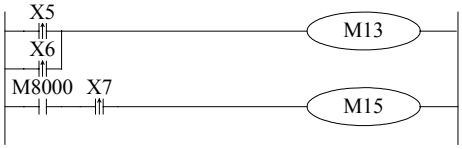
Mnemonic and function

Mnemonic	Function	Format and device X,Y,M,S,T,C
LDP (LoaDPulse)	Initial logical operation-Rising edge pulse	
LDF (LoaD Falling pulse)	Initial logical operation Falling/trailing edge pulse	
ANDP (AND Pulse)	Serial connection of Rising edge pulse	
ANDF (AND Falling pulse)	Serial connection of Falling/trailing edge pulse	
ORP (OR Pulse)	Parallel connection of Rising edge pulse	
ORF (OR Falling pulse)	Parallel connection of Falling/trailing edge pulse	

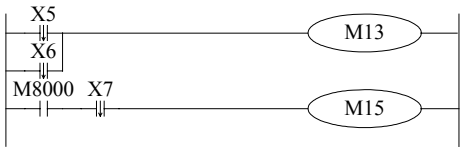
Instruction description

- LDP、ANDP、ORP are active for one program scan after the associated device switches from OFF to ON.
- LDF、ANDF、ORF are active for one program scan after the associated device switches from ON to OFF.

Program



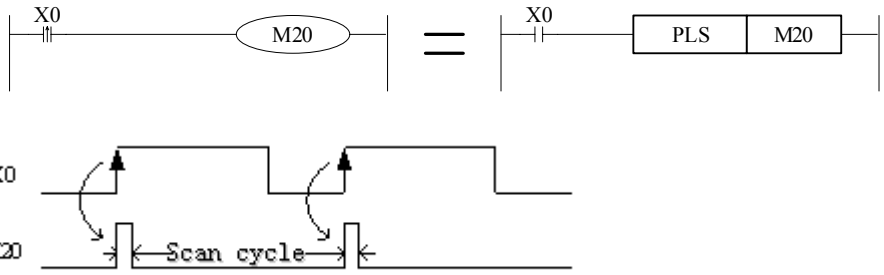

```
0   LDP    X5
1   ORP    X6
2   OUT    M13
3   LD     M8000
4   ANDP   X2
5   OUT    M15
```



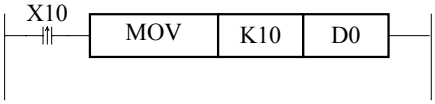

```
0  LDF   X5
1  ORF   X6
2  OUT   M13
3  LD    M8000
4  ANDF  X2
5  OUT   M15
```

In the up chart, when X000～X002 turns from ON to OFF or OFF to ON, M0 or M1 is active.

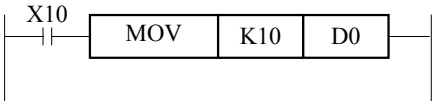
Output drive



In two conditions, when X0 turns from OFF to ON, M20 gets a scan cycle.



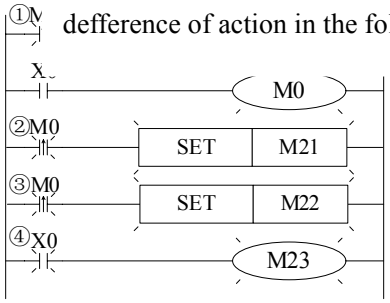
When X10 turns from OFF to ON, only execute once MOV instruction.



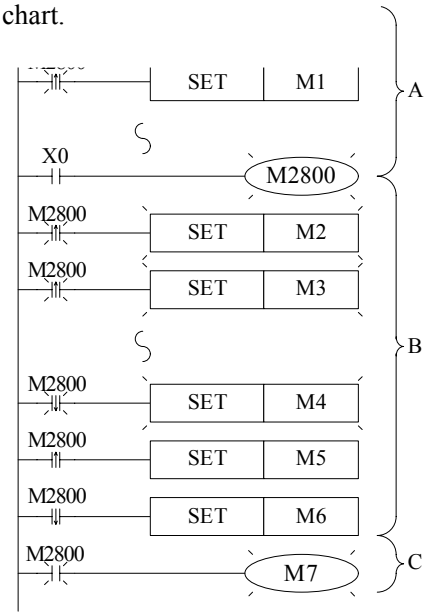
When X10 turns from OFF to ON, each scan cycle execute once MOV instruction.

The defference caused by the defference of auxiliary relay's ID

When assign the auxiliary relay (M) as LDP, LDF, ANDP, ORP, ORF devices, the defference of device's ID bound will cause the defference of action in the following chart.



After X000 drive M0, M0's correspond （1）～（4）contacts act.
In which:（1）～（3） execute M0 rising pulse check.（4）is LD instruction.



With the center of M2800 which is driven by X000, it’s deivded to be A,B these two area. In the contacts which are checked out by rising edge pulse and falling edge pulse in A,B these two areas, only one contact activates. The contact in area C is LD instruction. Hence it ges when M2800 activates. With this charatcterstc, program effectivey to the【carry on status transferring with the same signal】in the step ladder.

5-6. Contact compare instruction

Mnemonic and function

Mnemonic	Function
LD＝	Initial comparision contact. Active when the comparison （S1）＝（S2） is true.
LD＞	Initial comparision contact. Active when the comparison （S1）>（S2） is true
LD＜	Initial comparision contact. Active when the comparison （S1）<（S2） is true
LD＜＞	Initial comparision contact. Active when the comparison （S1）≠（S2） is true
LD＜＝	Initial comparision contact. Active when the comparison （S1）≤（S2） is true
LD＞＝	Initial comparision contact. Active when the comparison （S1）≥（S2） is true
AND＝	Serial comparison contact. Active when the comparison （S1）＝（S2） is true.
AND＞	Serial comparison contact. Active when the comparison （S1）>（S2） is true.
AND＜	Serial comparison contact. Active when the comparison （S1）<（S2） is true.
AND＜＞	Serial comparison contact. Active when the comparison （S1）≠（S2） is true
AND＜＝	Serial comparison contact. Active when the comparison （S1）≤（S2） is true.
AND＞＝	Serial comparison contact. Active when the comparison （S1）≥（S2） is true.
OR＝	Parallel comparison contact. Active when the comparison （S1）＝（S2） is true.
OR＞	Parallel comparison contact. Active when the comparison （S1）＞（S2）is true.
OR＜	Parallel comparison contact. Active when the comparison （S1）＜（S2）is true.
OR＜＞	Parallel comparison contact. Active when the comparison （S1）≠（S2）is true.
OR＜＝	Parallel comparison contact. Active when the comparison （S1）≤（S2）is true.
OR＞＝	Parallel comparison contact. Active when the comparison （S1）≥（S2）is true.

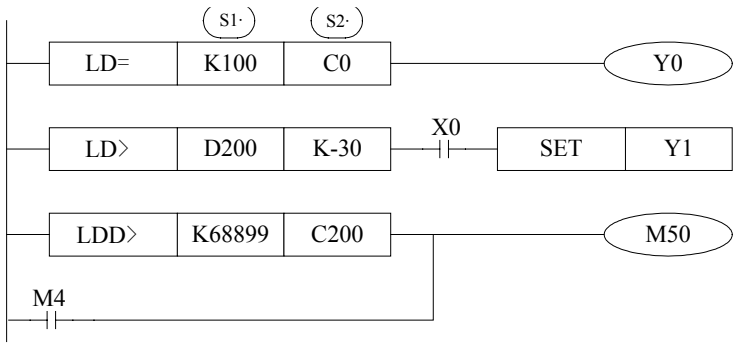
LD □

Format and function

The value of S1 and S2 are tested according to the comparison of the instruction. If the comparison is ture then the LD contact is active. If the comparison is false then the LD contact is not active.

16 bits	32 bits	Active condition	Not active condition
LD＝	DLD＝	（S1）＝（S2）	（S1）≠（S2）
LD＞	DLD＞	（S1）＞（S2）	（S1）≤（S2）
LD＜	DLD＜	（S1）＜（S2）	（S1）≥（S2）
LD＜＞	DLD＜＞	（S1）≠（S2）	（S1）＝（S2）
LD＜＝	DLD＜＝	（S1）≤（S2）	（S1）＞（S2）
LD＞＝	DLD＞＝	（S1）≥（S2）	（S1）＜（S2）

Program



Note items

- When the source data's highest bit（16 bits：b15，32 bits：b31）is 1，use the data as a negative.
- The comparison of 32 bits counter（C200~） must use 32 bits instruction. If assigned as 16 bits instruction, it will lead the program error or operation error.

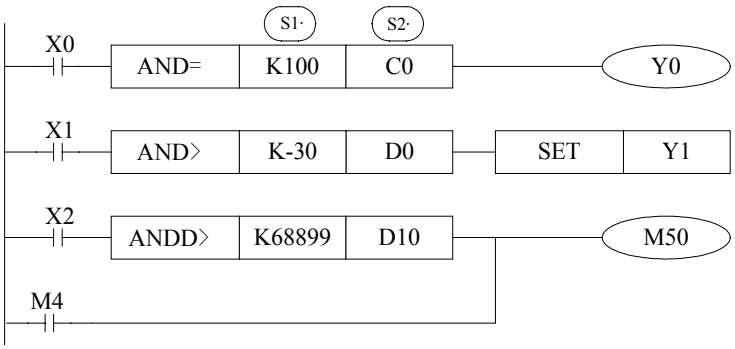
AND □

Format and function

The value of S1 and S2 are tested according to the instruction. If the comparison is ture then the AND contact is active. If the comparison is false then the AND contact is not active.

16 bits	32 bits	Active condition	Not active condition
AND＝	DAND＝	（S1）＝（S2）	（S1）≠（S2）
AND＞	DAND＞	（S1）＞（S2）	（S1）≤（S2）
AND＜	DAND＜	（S1）＜（S2）	（S1）≥（S2）
AND＜＞	DAND＜＞	（S1）≠（S2）	（S1）＝（S2）
AND＜＝	DAND＜＝	（S1）≤（S2）	（S1）＞（S2）
AND＞＝	DAND＞＝	（S1）≥（S2）	（S1）＜（S2）

Program



Note items

- When the source data's highest bit（16 bits：b15，32 bits：b31）is 1，use the data as a negative.
- The comparison of 32 bits counter（C200~） must be 32 bits instruction.If assigned as a 16 bits instruction, it will lead the ptogram error or operation error.

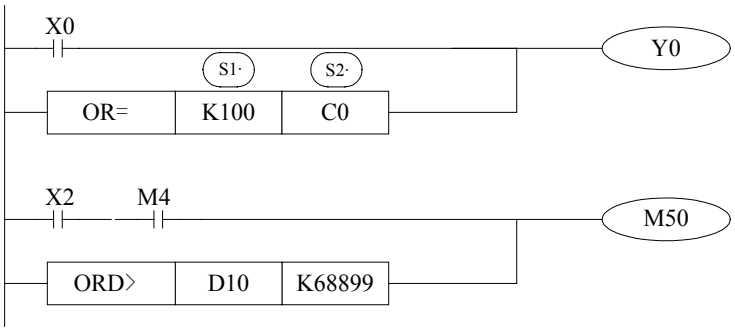
OR □

Foramt and function

The value of S1 and S2 are tested according to the instruction. If the comparison is ture then the OR contact is active. If the comparison is false then the OR contact is not active.

16 bits	32 bits	Active condition	Not active condition
OR＝	DOR＝	（S1）＝（S2）	（S1）≠（S2）
OR＞	DOR＞	（S1）＞（S2）	（S1）≤（S2）
OR＜	DOR＜	（S1）＜（S2）	（S1）≥（S2）
OR＜＞	DOR＜＞	（S1）≠（S2）	（S1）＝（S2）
OR＜＝	DOR＜＝	（S1）≤（S2）	（S1）＞（S2）
OR＞＝	DOR＞＝	（S1）≥（S2）	（S1）＜（S2）

Program

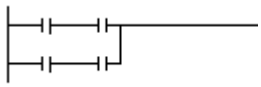


Note items

- When the source data's highest bit（16 bits：b15，32 bits：b31）is 1，use the data as a negative.
- The comparison of 32 bits counter（C200~） must be 32 bits instruction.If assigned as a 16 bits instruction, it will lead the ptogram error or operation error.

5-7.【ORB】

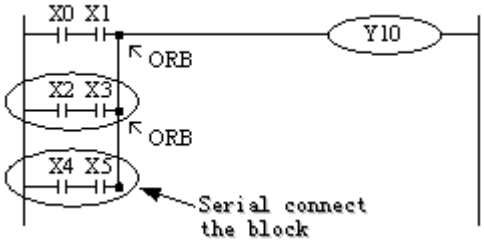
Mnemonic and function

Mnemonic	Function	Format and device
ORB (OR Block)	Parallel connection of multiply parallel circuits	

Description

- To declare the starting point of the circuit (usually serial circuit blocks) to the preceding circuit in parallel. Serial circuit blocks are those in which more than one contacts in series or the ANB instruction is used.
- An ORB instruction is an independent instruction and is not associated with any device number.
- There are no limitations to the number of parallel circuits when using an ORB instruction in the sequential processing configuration.
- When using ORB instructions in a batch, use no more than 8 LD and LDI instructions in the definition of the program blocks (to be connected parallel).

Program



Recommended sequencial programming method：

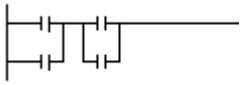
```
0  LD    X0
1  AND   X1
2  LD    X2
3  AND   X3
4  ORB
5  LDI   X4
6  AND   X5
7  ORB
8  OUT   Y10
```

Non-preferred batchprogramming method：

```
0  LD    X0
1  AND   X1
2  LD    X2
3  AND   X3
4  LDI   X4
5  AND   X5
6  ORB
7  ORB
```

5-8.【ANB】

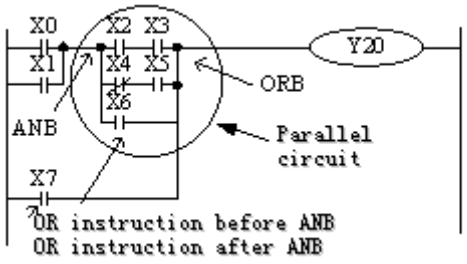
Mnemonic and function

Mnemonic	Function	Format and device
ANB (ANd Block)	Serial connection of multiply parallel circuits	

Description

- To declare the starting point of the circuit block, use a LD or LDI instruction. After completing the parallel circuit block, connect it to the preceding block in series using the ANB instruction. It is possible to use as many ANB instructions as necessary to connect a number of parallel circuit blocks to the preceding block in series. When using ANB instructions in a batch, use no more than 8 LD and LDI instructions in the definition of the program blocks (to be connected in parallel)

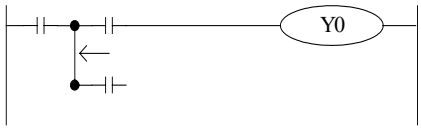
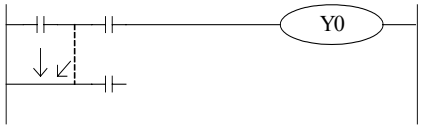
Program




```
0    LD     X0
1    OR     X1
                2      Start of a branch
3    AND    X3
       I       4      Start of a branch
5    AND    X5
                       End of a parallel circuit block
       OR              End of a parallel circuit block
       ANB             Serial connect with the preceding circuit
9    OR     X7
10   OUT    Y20
```

5-9.【MCS】,【MCR】

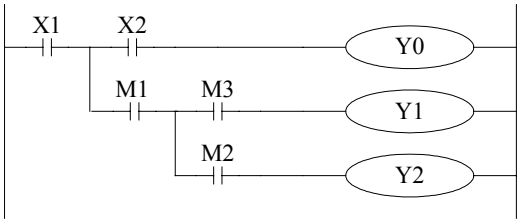
Mnemonic and function

Mnemonic	Function	Format and device
MCS (Master control)	Denotes the start of a master control block	
MCR (Master control Reset)	Denotes the end of a master control block	

Mnemonic and function

- After the execution of an MCS instruction, the bus line（LD、LDI）shifts to a point after the MCS instruction. An MCR instruction returns this to the original bus line.
- MCS、MCR instructions should use in pair.
- The bus line could be used nesting. Between the matched MCS、MCR instructions use matched MCS、MCR instructions. The nest level increase with the using of MCS instruction. The max nest level is 10. When executing MCR instruction, go back to the upper bus line.
- When use flow program, bus line management could only be used in the same flow. When end some flow, it must go back to the main bus line.

Description



5-10.【ALT】

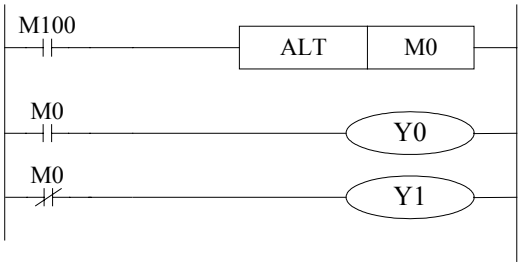
Mnemonic and function

Mnemonic	Function	Format and device X,Y,M,S,T,C
ALT (Alternate state)	The status of the assigned deviceis inverted on every operation of the instruction	

Description

The status of the destination device is alternated on every operation of theALT instruction.

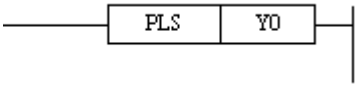
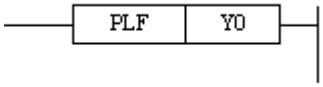
Program




```
0   LD     M100
1   ALT    M0
2   LD     M0
3   OUT    Y0
4   LDI    M0
5   OUT    Y1
```

5-11.【PLS】,【PLF】

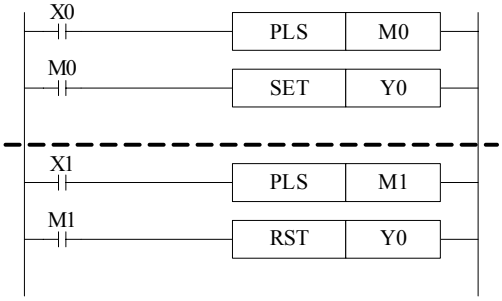
Mnemonic and function

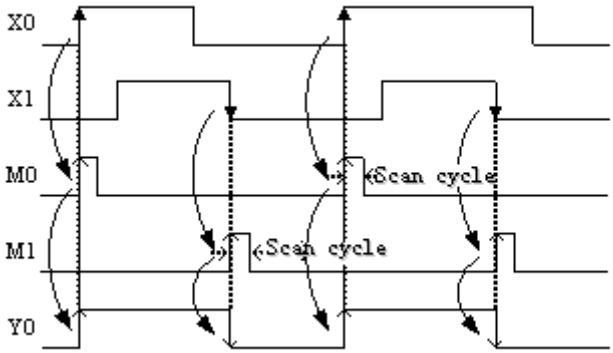
Mnemonic	Function	Format and device (all but special M)
PLS (PuLSe)	Rising edge pulse	
PLF (PuLse Falling)	Falling/trailing edge pulse	

Description

- When a PLS instruction is executed, object devices Y and M operate for one operation cycle after the drive input signal has turned ON.
- When a PLF instruction is executed, object devices Y and M operate for one operation cycle after the drive input signal has turned OFF.
- When the PLC status changed from RUN to STOP and back to RUN with the input signals still ON, PLS M0 is operated again. However, if an M coil which is battery backed (latched) was used instead of M0 it would not re-activate. For the battery backed device to be re-pulsed, its driving input (ex.X0) must be switched OFF during the RUN/STOP/RUN sequence before it will be pulsed once more.

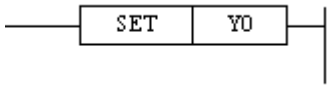
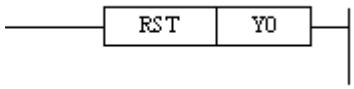
Program





5-12.【SET】,【RST】

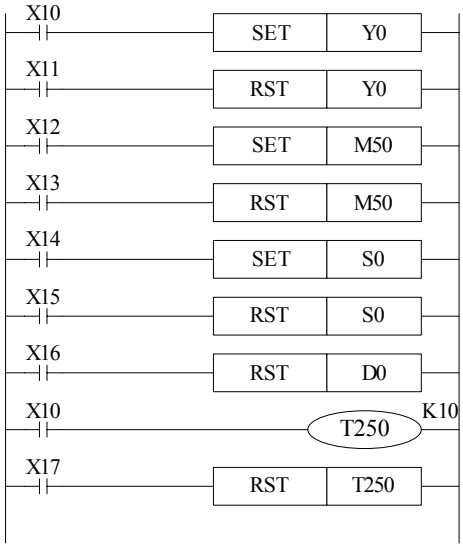
Mnemonic and function

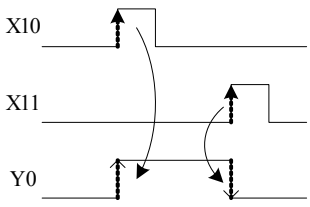
Mnemonic	Function	Format and device X,Y,M,S,T,C
SET (SET)	Set a bit device permanently ON	
RST (ReSeT)	Reset a bit device permanently OFF	

Description

- Turning ON X010 causes Y000 to turn ON. Y000 remains ON even after X010 turns OFF. Turning ON X011 causes Y000 to turn OFF. Y000 remains OFF even after X011 turns OFF. It's the same with M、S.
- SET and RST instructions can be used for the same device as many times as necessary. However, the last instruction activated determines the current status.
- After assign the start definition ID and end definition ID, operate the operands in one bound at the same time is available.
- Besides, it's also possible to use RST instruction to reset the current contents of timer, counter and contacts.
- When use SET、RST instruction, please try to avoid using the same definition ID with OUT instruction.

Program

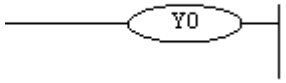
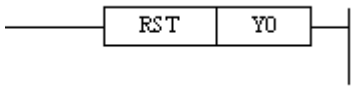




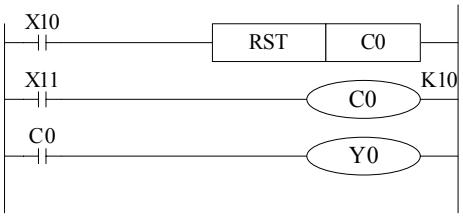

```
0   LD    X10
1   SET   Y0
2   LD    X11
3   RST   Y0
4   LD    X12
5   SET   M50
6   LD    X13
7   RST   M50
8   LD    X14
9   SET   S0
11  LD    X15
12  RST   S0
14  LD    X16
15  RST   D0
16  LD    X10
17  OUT   T250
      SP  K10
20  LD    X17
21  RST   T250
```

5-13.【OUT】,【RST】

Mnemonic and function

Mnemonic	Function	Format and device
OUT (OUT)	Final logic operation type coil drive	
RST (ReSeT)	Reset a bit device permanently OFF	

Program of interior counter

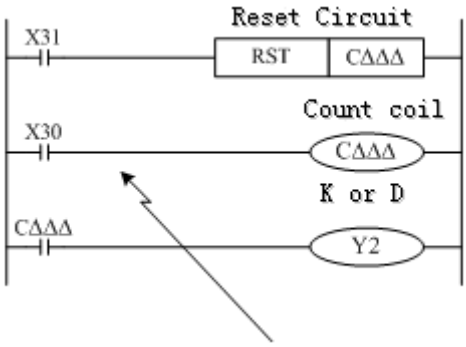


C0 carries on increase count for the OFF→ON of X011. When reach the set value K10, output contact Co activates. Afterwards, even X011 turns from OFF to ON, counter's current value will not change, output contact keep on activating.

To clear this, let X010 be the get status and reset the output contact. It's necessary to assign constant K or indirect data register's ID behind OUT instruction.

Counter used for power cut rententive. Even when power is cut, hold the current value and output contact's action status and reset status.

Program of high speed counter



- In the single phase single input counters among C600－C618, when X031 is ON, the output contact of counter C△△△ reset. Counter's current value turns to be 0.
- When X030 is ON, count the ON/OFF status of the count input X000-X021 which are decided by counter's ID.
- Counter's current value increase, when reach the set value (content of K or D), output contact is reset.

To drive high speed counter's count coil's contacts, please use the contacts which is always ON in the executing process of high speed counter. When driving dount coils, if use input realy（X000－X021）which are used for high speed counter's input ID（X000－X021），correct count will not be carries out.

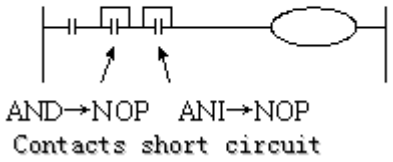
5-14.【NOP】,【END】

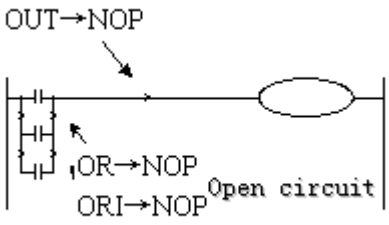
Mnemonic and function

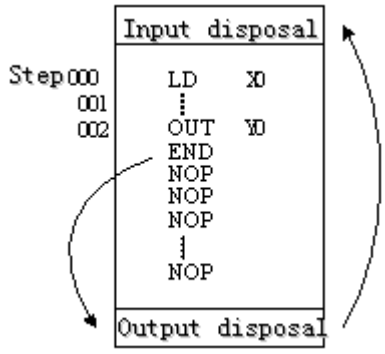
Mnemonic	Function	Format and device：None
NOP (No Operation)	No operation or null step	
END (END)	Force the current program scan to end	

Description

- When clear the whole program, all the instructions become NOP. If add NOP instructions between the common instructions, they have no effect and PLC will keep on working. If add NOP instructions in the program, then when modify or add programs, the step vary will be decreased. But the program should have rest quantity.
- If replace the program's instructions with NOP instructions, then the circuit will be changed, please note this.







PLC repeatedly carry on input disposal, program executing and output disposal. If write END instruction at the end of the program, then the instructions behind END instruction won't be executed. If there's no END instruction in the program, the PLC executes the end step and then repeat executing the program from step 0.

When debug, insert END in each program segment to check out each program's action.

Then, after comfirm the correction of preceding block'saction, delete END instruction.

Besides, the first execution of RUN begins with END instruction.

When executing END instruction, refresh monitor timer. (Check if scan cycle is a long timer.)

5-15. Items to note when programming

1、Contacts' structure and step number

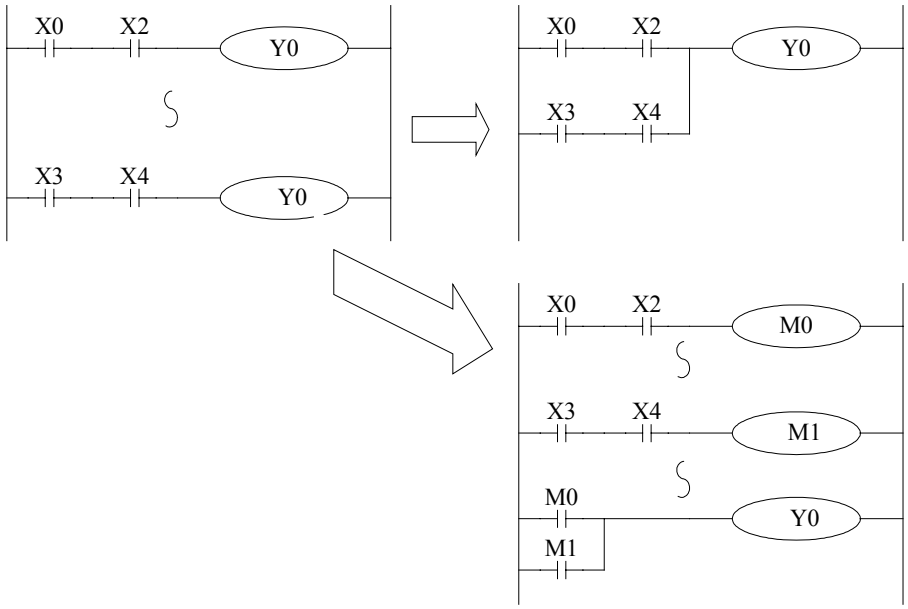
Even in the sequencial control circuit with the same action, it's also available to simple the program and save program's steps according to the contacts' structure. General program principle is：a）write the circuit with many serial contacts on the top；b）write the circuit with many parallel contacts in the left.

2、Program's executing sequence

Handle the sequencial control program by【From top to bottom】and【From left to right】 Sequencial control instructions also encode following this flow.

3、Dual output dual coil's activation and the solution

- If carry on coil's dual output (dual coil) in the sequencial control program, then the backward action is prior.
- Dual output (dual coil) doesn't go against the input rule at the program side. But as the preceding avtion is very complicate, please modify the program as in the following example.



- There are other methods. E.g. jump instructions or step ladder. However, when use step ladder, if the main program's output coil is programmed, then the disposal method is the same with dual coil, please note this.

6. Applied instructions

This chapter tells the applied instructions' function of XC series PLC.

6-1.Applied instruction list

The applied instructions' sort and their correspond instructions are listed in the following table:

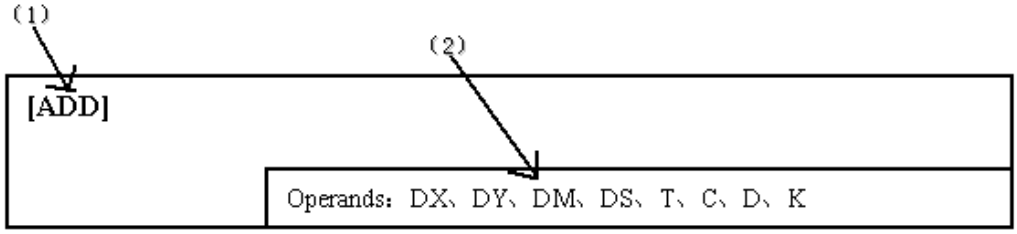
Sort	Mnemonic	Function
Program Flow	CJ	Condition jump
	CALL	Call subroutine
	SRET	Subroutine return
	STL	Flow start
	STLE	Flow end
	SET	Open the assigned flow, close the current flow
	ST	Open the assigned flow, not close the current flow
	FOR	Start of a FOR-NEXT loop
	NEXT	End of a FOR-NEXT loop
	EI	Enable interrupts
	DI	Disable interrupts
	IRET	Interrupt return
	FEND	First end
	END	Subroutine end
Data Move	MOV	Move
	BMOV	Block move
	FMOV	Fill move
	ZRST	Zone reset
	SWAP	The high and low byte of the destinated devices are exchanged
	XCH	Exchange
Data Operation	ADD	Addition
	SUB	Subtraction
	MUL	Multiplication
	DIV	Division
	INC	Increment
	DEC	Decrement
	MEAN	Mean
	WAND	Word And
	WOR	Word OR
	WORX	Word exclusive OR
	CML	Compliment
	NEG	Negative

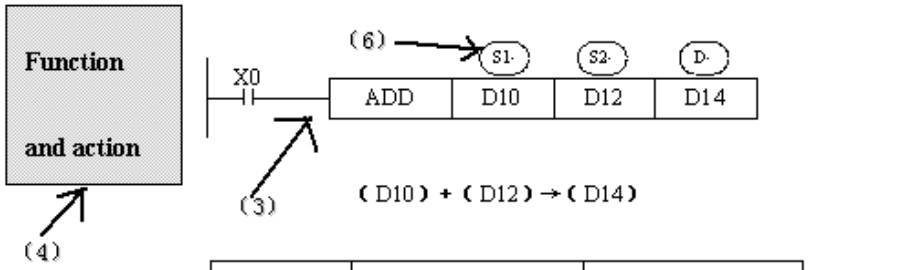
Data Shift	SHL	Arithmetic Shift Left
	SHR	Arithmetic Shift Right
	LSL	Logic shift left
	LSR	Logic shift right
	ROL	Rotation shift left
	ROR	Ritation shift right
	SFTL	Bit shift left
	SFTR	Bit shift right
	WSFL	Word shift left
	WSFR	Word shift right
Data Convert	WTD	Single word integer converts to double word integer
	FLT	32 bits integer converts to float point
	FLTD	64 bits integer converts to float point
	INT	Float point converts to binary
	BIN	BCD converts to binary
	BCD	Binary converts to BCD
	ASC	Hex. converts to ASCII
	HEX	ASCII converts to Hex.
Float Point Operation	ECMP	Float compare
	EZCP	Float Zone compare
	EADD	Float Add
	ESUB	Float Subtract
	EMUL	Float Multiplication
	EDIV	Float division
	ESOR	Float Square Root
	SIN	Sine
	COS	Cosine
	TAN	Tangent
Clock Operation	TCMP	Time Compare
	TZCP	Time Zone Compare
	TADD	Time Add
	TSUB	Time Subtract
	TRD	Read RTC data
	TWR	Set RTC data

6-2. Reading method of the applied instruction's description

The understanding method of instruction's description

In this manual, instructions are described with the following format.





Flag		
	Zero	M8020
	Borrow bit	M8021
	Carrier	M8022

(5)

- The data contained within the two source devices are combined and the total is stored in the specified destination device. Each data's highest bit is the sign bit, 0 stands for positive,、 1 stands for negative. All calculations are algebraic processed. (5+ (-8) =-3)
- If the result of a calculation is "0", the "0" flag acts. If the result exceeds 323, 767 (16 bits limit) or 2, 147, 483, 647 (32 bits limit), the carry flag acts. (refer to the next page). If the result exceeds -323, 768 (16 bits limit) or -2, 147, 483, 648 (32 bits limit), the borrow flag acts (Refer to the next page)
- When carry on 32 bits operation, word device's low 16 bits are assigned, the device following closely the preceding device's ID will be the high bits. To avoid ID repetition, we recommend you assign device's ID to be even ID.
- The same device may be used as a source and a destination. If this is the case then the result changes after every scan cycle. Please note this point.

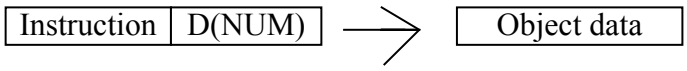
① Instruction's name
② Device which can be used
③ Ladder example
④ Tell the instruction's basic action, using way, applied example, extend function, note items etc.
⑤ Flag after executing the instruction. Instructions without the direct flag will not display.
⑥ (S·) : Source operand, its content won't change after executing the instruction
(D·) : Destinate operand, its content changes with the execution of the instruction

The related description

- The assignment of the data

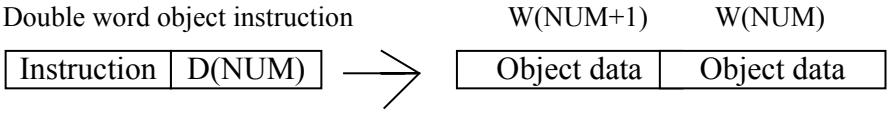
  The data register of XC series PLC is a single word (16 bit) data register,single word data only engross one data register which is assigned by single word object instruction. The disposal bound is: Dec. -327，68~327，67, Hex. 0000~FFFF。

Single word object instruction



Double word（32 bit）engrosses two data register, it's composed by two consecutive data registers, the first one is assigned by double word object instruction. The dispose bound is: Dec. -214,748,364,8~214,748,364,7, Hex. 00000000~FFFFFFFF。

Double word object instruction



- The denote way of 32 bits instruction

  If an instruction can not only be 16 bits but also be 32 bits, then the denote method for 32 bits instruction is to add a "D" before 16 bits instruction.

  E.g：ADD D0 D2 D4 denotes two 16 bits data adds；

  DADD D10 D12 D14 denotes two 32 bits data adds

6-3. Program flow instructions

Mnemonic	Instruction's name
CJ	Condition Jump
CALL	Call subroutine
SRET	Subroutine return
FOR	Start of a FOR-NEXT loop
NEXT	End of a FOR-NEXT loop
EI	Enable interrupts
DI	Disable interrupts
IRET	Interrupt return
FEND	First end
END	Subroutine ends

[CJ]

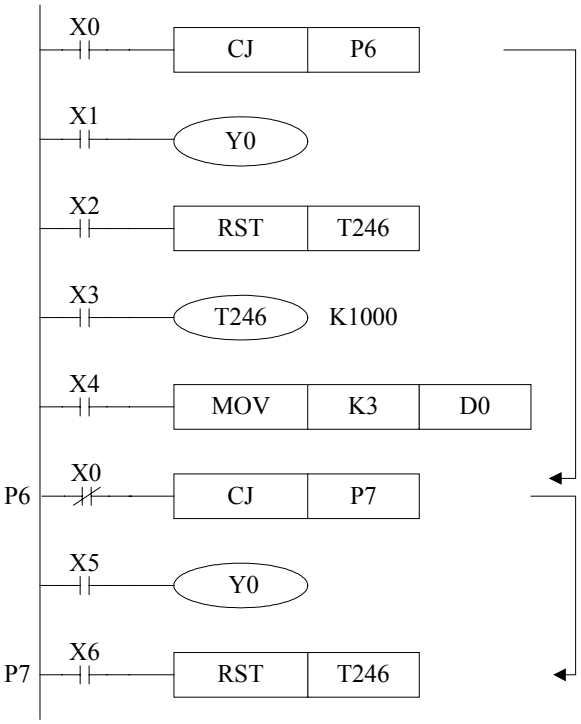
Operand：P

Function and action

With CJ instruction, it will shorten the operate cycle and use double coil is available. In the following chart, if X000 “ON”, then jump to the next step marked P6 from the first step. When X000 “OFF”, do not execute jump instruction.

chart：program P6 program P7

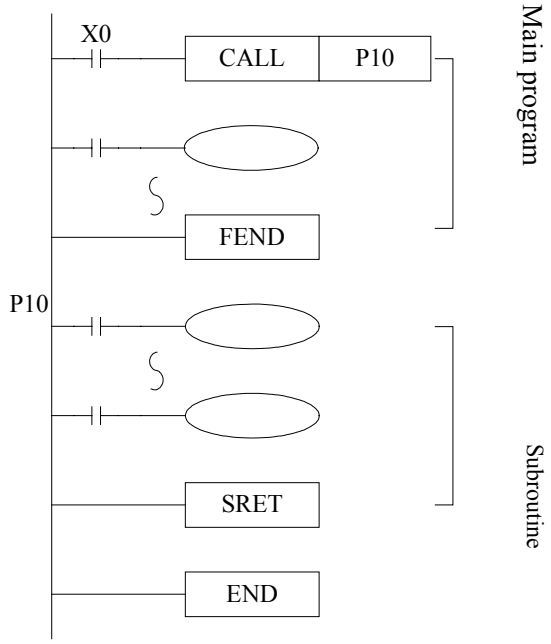
Program’s timer T0~T599 and high speed counter C600~C619, if jump after driving, then go on working, the output contacts are also activated.



- Y000 turns to be the dual coil, X001 activates when X000=OFF, X005 activates when X000=ON. With condition jump, even it’s a subsection program, when convert a same coil to be up to 2 programs inside the JUMP or outside the JUMP, treat it as a normal coil.
- When accumulate the reset instructions of timer and counter beyond the JUMP, time coil and the jump count coil reset (contact recover and clear the current value) is valid.

[CALL] and [SRET]	
	Usable device：P

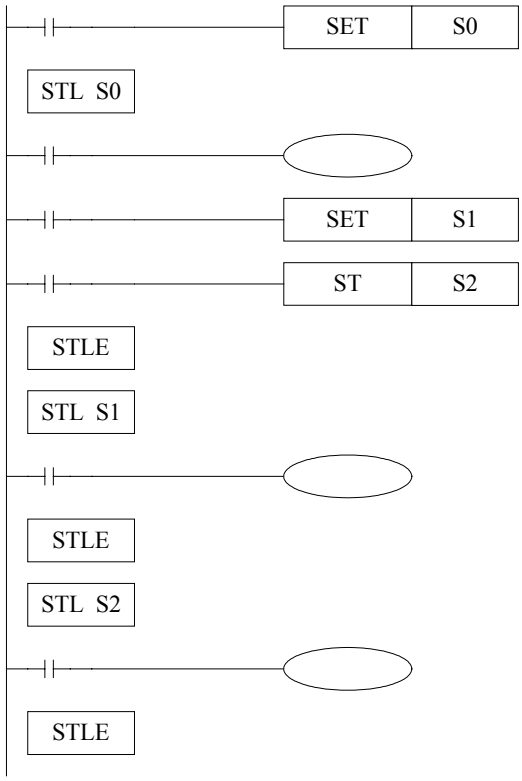
Function and action



- When X000 is “ON”, then execute “CALL” instruction and jump to the step denoted with P10. After finish executing the subroutine, go back via executing SRET instruction. Program after FEND instruction which will be said later.
- Please don’t use CJ instruction’s denotion and repeat number. However, using repeat number of CALL instruction’s operate object is allowed.
- There could be 9 times CALL instruction in the subroutine, and for the whole, the nesting level is 10.

[STL], [SET][ST] and [STLE]	
	Usable device：S

Function and action

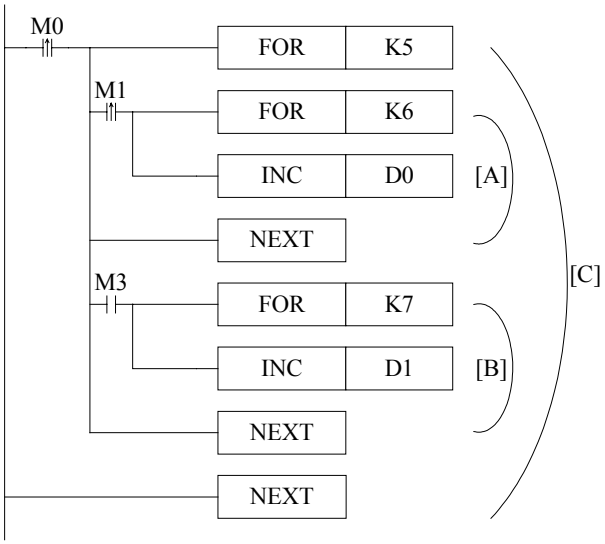


- STL and STLE should be used in pairs. It's available to nesting, and the current nesting level is 1.
- After executing SET Sxxx instruction, the flow assigned by this instruction is ON.
- After executing RST Sxxx instruction, the assigned flow is OFF.
- In flow S0, SET S1 close the current flow S0, open the flow S1.
- In flow S0, ST S2 open flow S2, but not close flow S0.
- When the flow turns from ON to OFF, set the flow's OUT、PLS、PLF、not accumulate timer etc. OFF or reset.

[FOR] and [NEXT]	
	sable device：DX、DY、DM、DS、T、C、D、K

Function and action

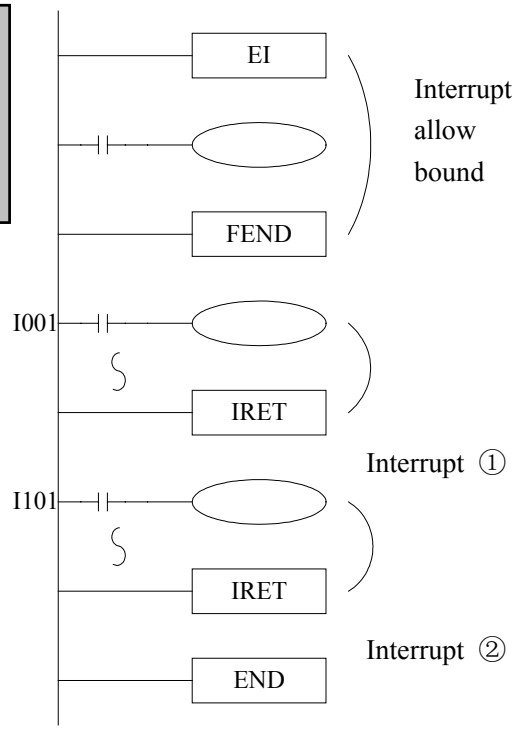
A FOR~NEXT loop operates for its set number of times before the main program is allowed to finish the current program scan.

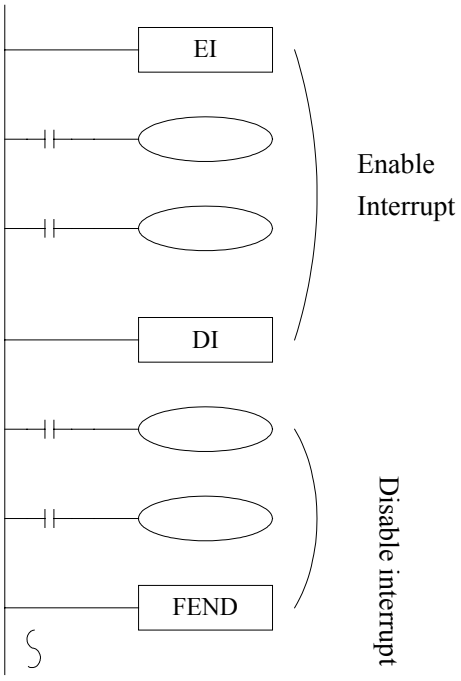


- FOR、NEXT instructions must be programmed as a pair. Nesting is allowed, and the nesting level is 8.
- Between FOR/NEXT, LDP、LDF instructions are effective for one time. Everytime when M0 turns from OFF to ON, and M1 turns from OFF to ON, [A] loop is executed 6 times.
- Everytime if M0 turns from OFF to ON and M3 is ON, [B] loop is executed 5×7=35 times.
- If there are many loop times, the scan cycle will be prolonged. Monitor timer error may occur, please note this.
- If NEXT is before FOR, or no NEXT, or NEXT is behind FENG，END, or FOR and NEXT number is not equal, an error will occur.

[EI]、 [DI] and [IRET]	
	Operands：None

Function and action



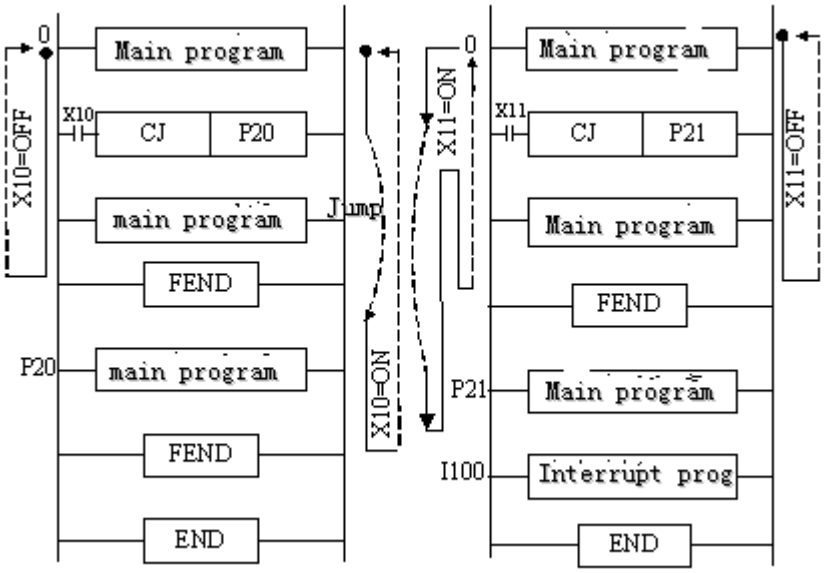


- Usually PLC is in the status of interruption. If use EI instruction to enable interruption, then in the process of scan the program, if X000 or X001 is "ON", then execute the interrupt routine ①、②, then return to the main program.
- Interrupt pointer（I***）should be behind FEND instruction and program as label.
- I001、I101 is X000 pulse rising edge check

- Program via DI instruction, disable interruption area could be set.
- Even occur interruption between DI~EI instructions（0 interruption forbidden area）, it could also be momeried and be executed after EI instruction.（Special auxiliary relay M8050~M8059 used for disable interruption ） If the interruption area is long, accept interruption will delay.
- When needn't interruption forbidden, please just program with EI instruction, it's not necessary to program with DI instruction.

[FEND] and [END]	
	Operands：None

Function and action	An FEND instruction indicates the first frrst end of a main program and the start of the frogram area to be used for subroutines. Under normal operating circumstances the FEND instruction performs a similar action to the END instruction, i.e.output processing, input processing and watchdog timer refresh are all carried out on execution.



- If program the tag of CALL instruction behind FEND instruction, there must be SRET instruction. If the interrupt pointer program behind FEND instruction, there must be SRET instruction.
- After executing CALL instruction and before executing SRET instruction, if execute FEND instruction; or execute FEND instruction after executing FOR instruction and before executing NEXT, then an error will occur.
- In the condition of using many FEND instruction, please compile routine or subroutine between the last FEND instruction and END instruction.

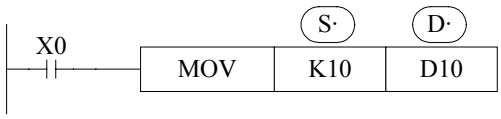
6-4. Data Move

Mnemonic	Function
MOV	Move
BMOV	Block Move
FMOV	Fill Move
ZRST	Zone Reset
SWAP	Float To Scientific
XCH	Exchange

[MOV]

Operands：DX、DY、DM、DS、T、C、D、K

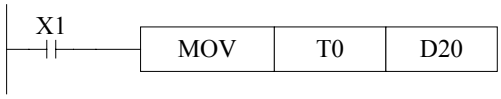
Function and action



Move data from one storage area to a new storage area

- When move contents from source to destination, if X000 is OFF, data will not change.
- Constant K10 will automatically convert to be BIN code.

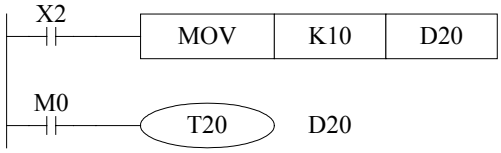
《Read out the current value of timer、counter》



（T0 current value）→（D20）

It's the same with the counter.

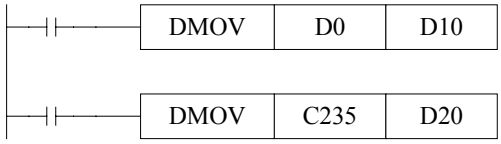
《Indirect assign the set value of timer、counter》



（K10）（D10）

D20=K10

《Move of 32 bits data》



（D1，D0）→（D11，D10）

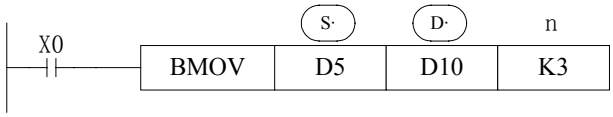
（C235 current value）→（D21，D20）

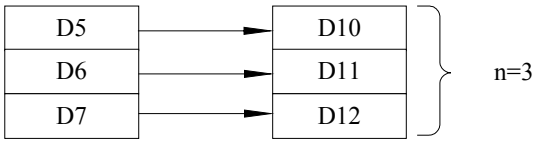
[BMOV]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action

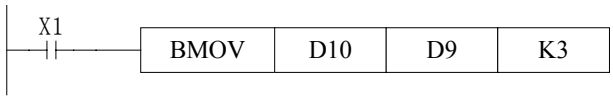
- A quantity of consecutively occurring data elements can be copied to a new destination. The source data is identified as a device head address(S) and a quantity of consecutive data elements (n). This is moved to the destination device (D) for the same number of elements (n). (If the quantity of source device (n) exceeds the actual number of available source devices, then only those devices which fall in the available range will be used. If the number of source devices exceeds the available space at the destination location, then only the available destination devices will be written to.)

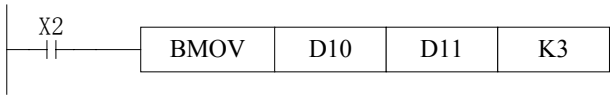


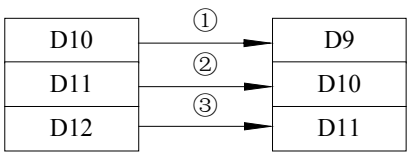


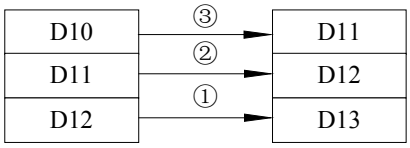
- The BMOV instruction has a built in automatic feature to prevent overwriting errors from occurring when the source (S-n) and destination (D-n) data ranges coincide. This is clearly identified in the following diagram:

(NOTE: The numbered arrows indicate the order in which the BMOV is processed).





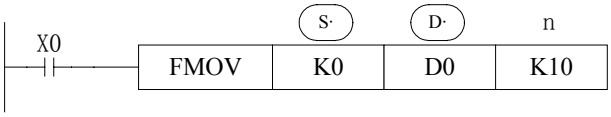




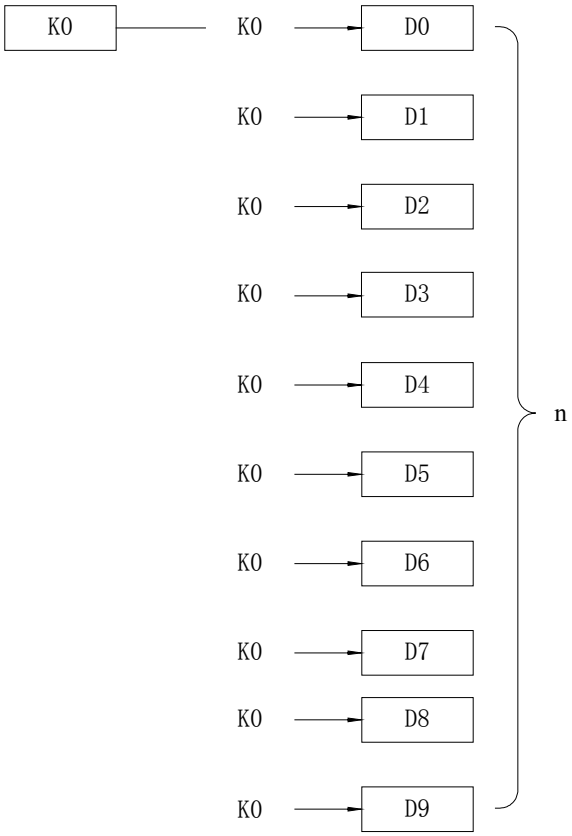
[FMOV]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action



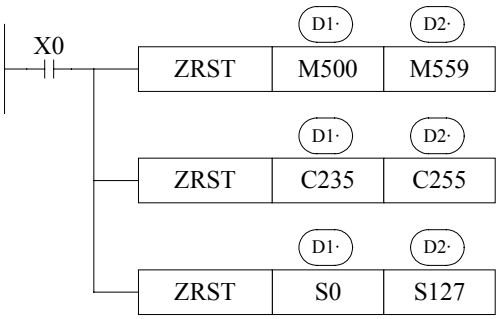
- Move K0 to D0~D9. Copy a single data device to a range of destination devices.
- The data stored in the source device (S) is copied to every device within the destination range, The range is specified by a device head address (D) and a quantity of consecutive elements (n).
- If the specified number of destination devices (n) exceeds the available space at the destination location, then only the available destination devices will be written to.



[ZRST]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action



Zone reset bit device M500~M599。

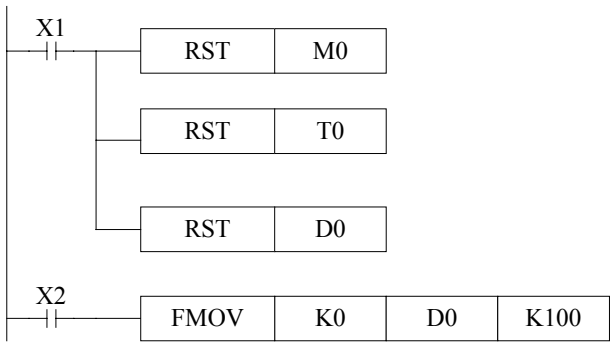
Zone reset word device C235~C255

Zone reset status S0~S127

- D1 and D2 are assigned to be the same device, and D1 < = D2. When D1>D2, only reset device in D1.
- The instruction is 16 bits, but it's available to use D1, D2 to assign 32 bits counter. But mix assignment is not allowed. I.e. D1 is a 16 bits counter, D2 is a 32 bits counter, this condition is not allowed.

Other reset instructions

- As single reset instruction of device, RST instruction is available of bit device Y，M，S and word device T，C，D.
- As Fill Move instruction of K0, you could write 0 into device DX，DY，DM，DS，T，C，D.



Reset M0

Reset the current value of T0

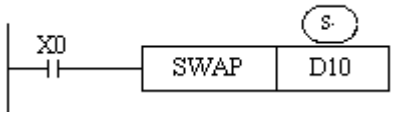
Reset D0

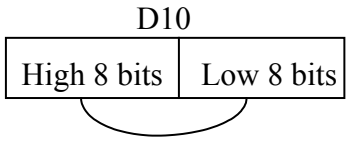
Write K0 into D0~D9

[SWAP]

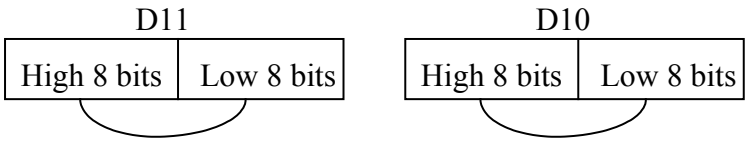
Operands：DX、DY、DM、DS、T、C、D

Function and action





- Low 8 bits and high 8 bits change when it is 16 bits instruction.



- Each 8 bits and high 8 bits change when it is 32 bits instruction.

- If the instruction is a consecutive executing instruction, each operation cycle should change.
- The usage of this instruction is the same with the extend function of XCH instruction.

[XCH]

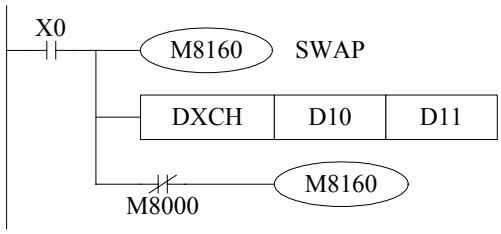
Operands：DX、DY、DM、DS、T、C、D、K

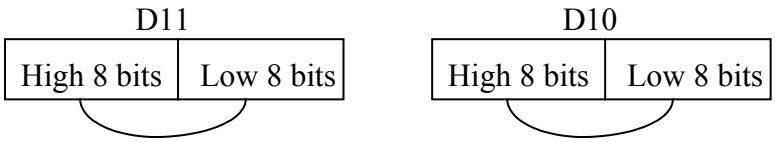
Function and action

The contents of the two destination devices D1 and D2 are swapped, i.e. the complete word devices are exchanged.

Before XCH （D10）=100 →After XCH （D10）=101
（D11）=101 （D11）=100

Extend function





- When M8160 is ON and D1 and D2 is the same device, the low 8 bits and high 8 bits can be exchanged.
- The things of 32 bits instruction is the same.
- When using the byte XCH (i.e.M8160 is ON) D1 and D2 must be the same device otherwise a program error will occure and M8067 will be turned ON.
- The extend function is the same with the action of SWAP instruction, usually use SWAP instruction.

6-5. Data operation instructions

Mnemonic	Function
ADD	Addation
SUB	Subtraction
MUL	Multiplication
DIV	Division
INC	Increment
DEC	Decrement
MEAN	Mean
WAND	Logic Word And
WOR	Logic Word Or
WORX	Logic Exclusive Or
CML	Compliment
NEG	Negation

[ADD]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action

（D10）＋（D12）→（D14）

Flag	Zero	M8020
	Borrow bit	M8021
	Carrier	M8022

- The data contained within the two source devices are combined and the total is stored in the specified destination device. Each data's highest bit is the sign bit, 0 stands for positive,、1 stands for negative. All calculations are algebraic processed.（5+（-8）=-3）
- If the result of a calculation is "0", the "0" flag acts. If the result exceeds 323，767（16 bits limit）or 2，147，483，647（32 bits limit）, the carry flag acts.（refer to the next page）. If the result exceeds -323，768（16 bits limit）or -2，147，483，648（32 bits limit）, the borrow flag acts（Refer to the next page）
- When carry on 32 bits operation, word device's low 16 bits are assigned, the device following closely the preceding device's ID will be the high bits. To avoid ID repetition, we recommend you assign device's ID to be even ID.
- The same device may be used as a source and a destination. If this is the case then the result changes after every scan cycle. Please note this point.

[SUB]

Operands：DX、DY、DM、DS、T、C、D、K

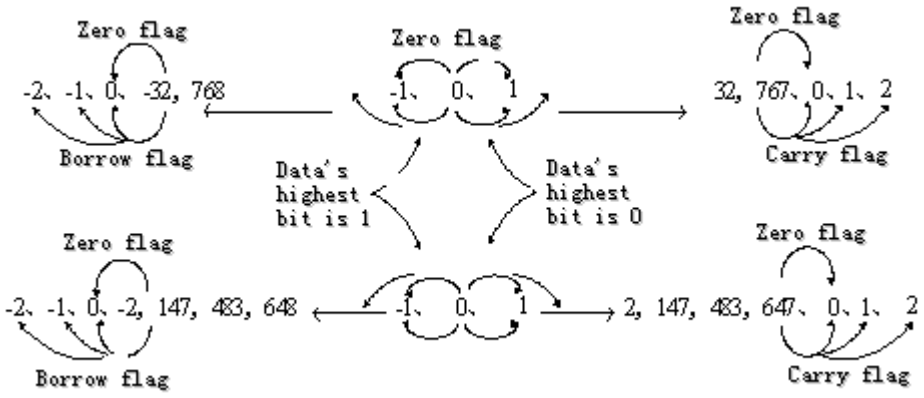
Function and action

（D10）—（D12）→（D14）

Flag	Zero	M8020
	Borrow bit	M8021
	Carrier	M8022

- (S1·) appoint the soft unit's content, subtract the soft unit's content appointed by (S2·) in the format of algebra. The result will be stored in the soft unit appointed by (D·). (5-(-8)=13)
- The action of each flag, the appointment method of 32 bits operation's soft units are both the same with the preceding ADD instruction.

The relationship of the flag's action and positive/nagetive data is the following chart.

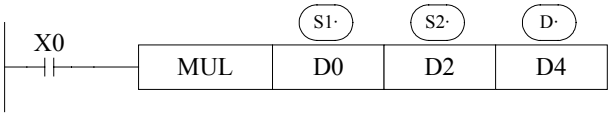


[MUL]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action

《16 bits operation》



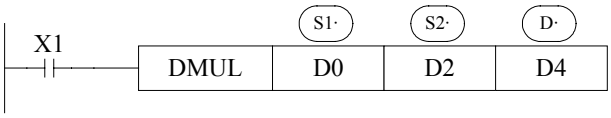
BIN　　BIN　　BIN

（D0）×（D2）→（D5，D4）

16 bits　　16 bits →　32 bits

- The contents of the two source devices are multiplied together and the result is stored at the destination device in the format of 32 bits. As in the upward chart： when （D0）=8、（D2）=9，（D5，D4）=72。
- The result's highest bit is the symbol bit: positive（0）、negative（1）.
- When be bit unit, it can carry on the bit appointment of K1~K8. When appoint K4, only the result's low 16 bits can be obtained.

《32 bits operation》



BIN　　BIN　　BIN

D1，D0）×（D3，D2）→（D7，D6，D5，D4）

32 bits　　32 bits →　64 bits

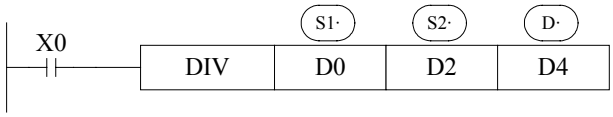
- In 32 bits operation, when use bit device as the destination address, only low 32 bits result can be obtained. The high 32 bits result can not be obtained, so please operate again after transfer one time to the word device
- Even use word device, 64 bits results can't be monitored at once.
- In this situation, float point data operation is recommended.

[DIV]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action

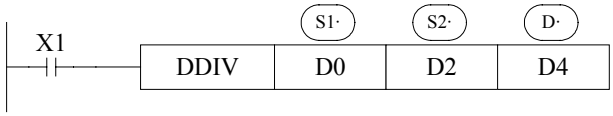
《16 bits operation》



Dividend		Divisor		Result		Remainder
BIN		BIN		BIN		BIN
(D0)	÷	(D2)	→	D4）	---	(D5)
16 bits		16 bits		16 bits		6 bits

- (S1·) appoints the device's content be the dividend, (S2·) appoints the device's content be the divisor, (D·) appoints the device and the next one to store the result and the remainder.

《32 bits operation》



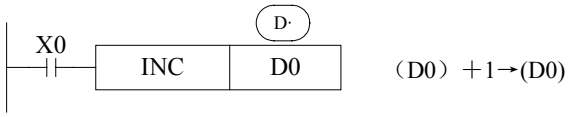
Dividend		Divisor		Result		Result
BIN		BIN		BIN		BIN
(D1,D0)	÷	(D3,D2)		（D5,D4）	---	(D7,D6)
32 bits		32 bits		32 bits		32 bits

- The dividend is composed by the device appointed by (S1·) and the next one. The divisor is composed by the device appointed by (S2·) and the next one. The result and the remainder are stored in the four sequencial devices, the first one is appointed by (D·).
- If the value of the divisor is 0, then an operation error is executed and the operation of the DIV instruction is cancelled.
- When appoint the bit device as (D·), the remainder will not obtained.
- The highest bit of the result and remainder is the symbol bit (positive:0, negative: 1). When any of the dividend or the divisor is negative, then the result will be negative. When the dividend is negative, then the remainder will be negative.

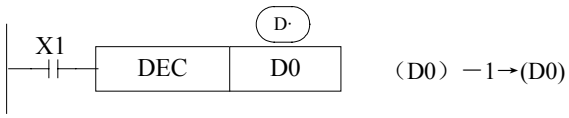
[INC] and [DEC]

Operands：DX、DY、DM、DS、T、C、D

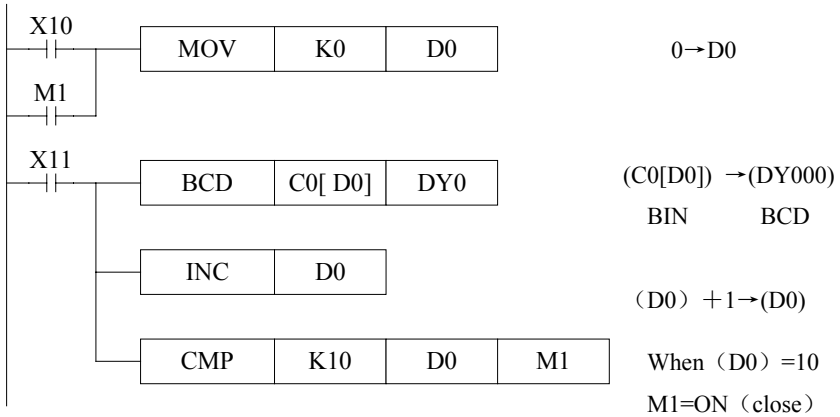
Function and action



- On every execution of the instruction the device specified as the destination (D·) has its current value incremented (increased) by a value of 1.
- In 16 bits operation, when +32，767 is reached, the next increment will write -32，767 to the destination device. In this case, there's no additional flag to identifiy this change in the counted value.



- On every execution of the instruction the device specified as the destination (D·) has its current value decremented (decreased) by a value of 1.
- When -32，768 or -2，147，483，648 is reached, the next decrement will write +32，767 or +2，147，483，647 to the destination device.

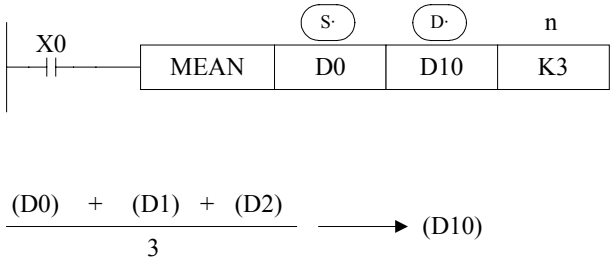


- Change the current value of counter C0~C9 to be BCD code and output to DY000，
- In advance, via reset and input X010 to clear D0.
- Everytime when X011 gets, output the current value of C0, C1…C9 sequencially.

[MEAN]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action



- The value of all the devices within the source range is summed and then divided by the number of devices summed, i.e. n.. This generates an integer mean value which is stored in the destination device (D) The remainder of of the caculated mean is ignored.
- If the value of n is specified outside the stated range (1 to 64) an error is generated.

[WAND]、 [WOR] and [WXOR]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action

- Execute logic AND operation with each bit

0&0=0 0&1=0
1&0=0 1&1=1

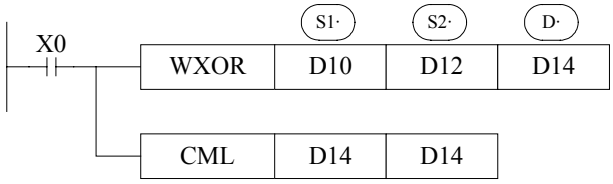
- Execute logic OR operation with each bit

0 or 0=0 0 or 1=1
1 or 0=1 1 or 1=1

- Execute logic Exclusive OR operation with each bit.

0 xor 0=0 0 xor 1=1
1 xor 0=1 1 xor 1=0

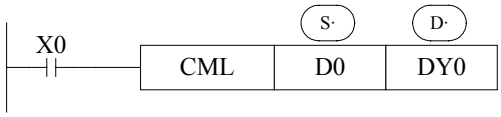
If use this instruction along with CML instruction , XOR NOT operation could also be executed .



[CML]

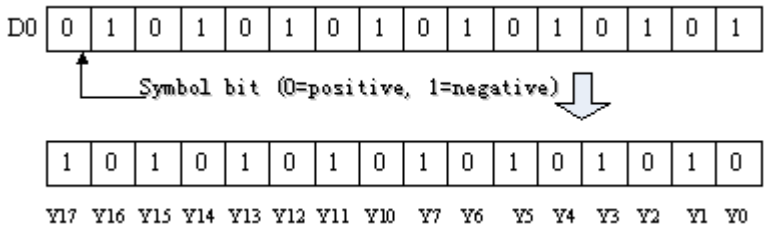
Operands：DX、DY、DM、DS、T、C、D

Function and action

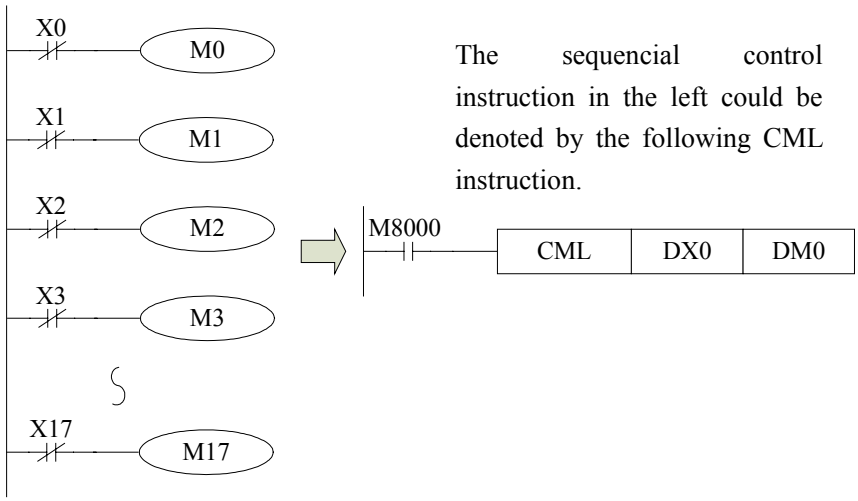


A copy of each data bit within the source device (S·) is inverted and then moved to the designated destination (D·).

- Each data bit in the source device is inverted and sent to the destination device. If use constant K in the source device, it can be auto convert to be binary.
- It's available when you want to inverted output the PLC's output



《Reading of inverted input》

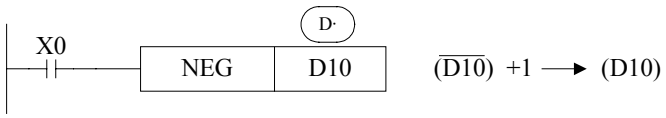


The sequencial control instruction in the left could be denoted by the following CML instruction.

(NEG)

Operands：DX、DY、DM、DS、T、C、D

Function and action



- The bit patten of the selected device is inverted, I.e. any occurrence of a “1’ becomes a “0” and any occurrence of “0” becomes “1”, when this is complete, a further binary 1 ia added to the bit patten. The result is the total logic sigh change of the selected devices contents.
- When using continually executing instructions, then this instruction will be executed in every scan cycle.

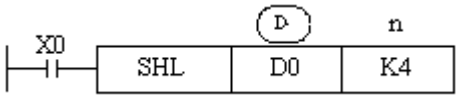
6-6. Shift instructions

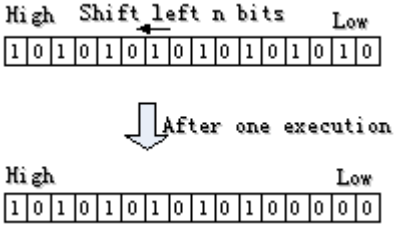
Mnemonic	Function
SHL	Arithmetic shift left
SHR	Arithmetic shift right
LSL	Logic shift left
LSR	Logic shift right
ROL	Rotation left
ROR	Rotation right
SFTL	Bit shift left
SFTR	Bit shift right
WSFL	Word shift left
WSFR	Word shift right

[SHL] and [SHR]	
	Operands：DX、DY、DM、DS、T、C、D

Function and action

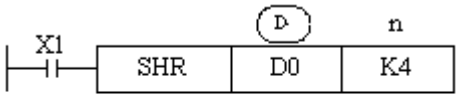
《Arithmetic shift left》

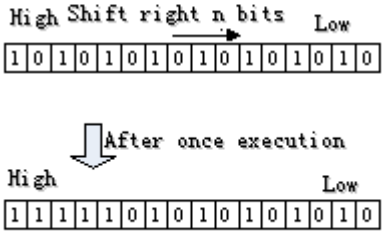




- After one execution, fill 0 in the low bit

《Arithmetic shift left》





- After once execution, the highest bit remains.

NOTE:

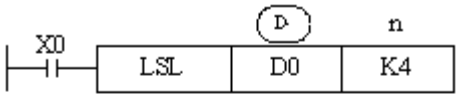
- In every scan cycle, loop shift left/right action will be executed
- The things of 32 bits is the same.

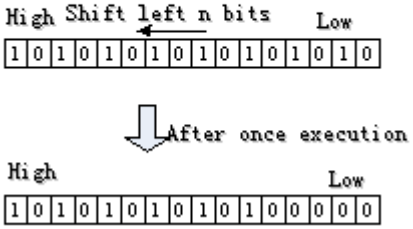
[LSL] and [LSR]

Operands：DX、DY、DM、DS、T、C、D

Function and action

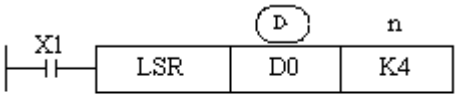
《Logic shift left》

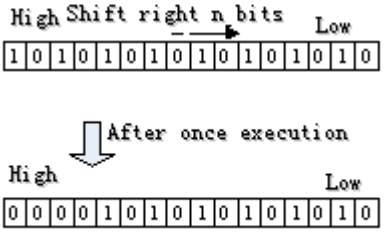




- After once execution, fill 0 in the low bit.

《Logic shift right》





- After once execution, fill 0 in the highest bit.

NOTE:

- In every scan cycle, loop shift left/right action will be executed
- The things of 32 bits is the same.

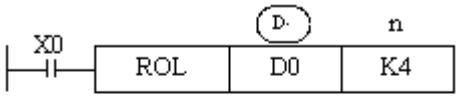
[ROL] and [ROR]

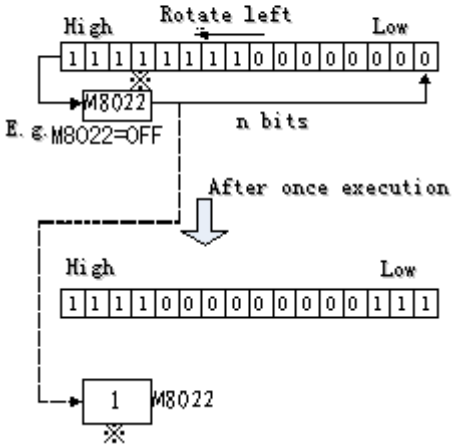
Operands：DX、DY、DM、DS、T、C、D

Function and action

The bit patten of the destination device is rotated n bit places to the left on every operation of the instruction

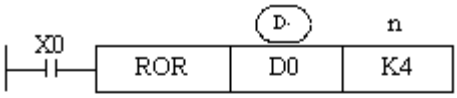
《Rotation shift left》

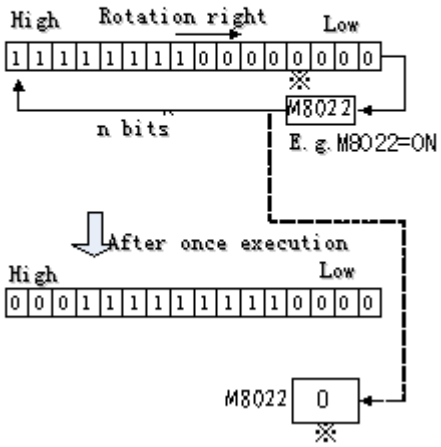




- Every time when X000 turns from OFF to ON, executes n bits left rotation.

《Rotation shift right》





- Every time when X000 turns from OFF to ON, executes n bits right rotation.

- As there is a carry flag in the rotation circuit, so if drive M8022 before executing the rotation instruction, it could be sent to the destination address.
- Please note that rotation left/right action is executed in every scan cycle.
- The situation of 32 bits is the same.

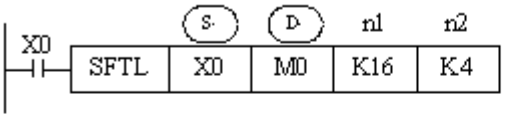
[SFTL] and [SFTR]

Operands：DX、DY、DM、DS、T、C、D

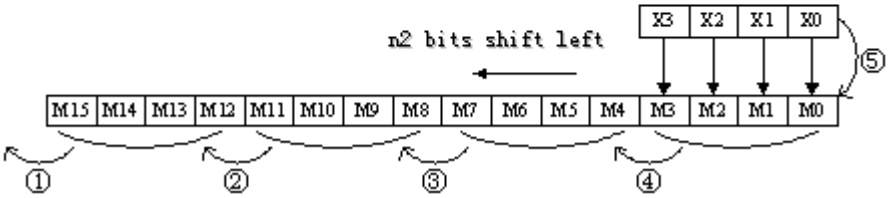
Function and action

- The instruction copies n2 source devices to a bit stack of length n1. For every new addition of n2 bits, the existing data within the bit stack is shifted n2 bits to the left/right. Any bit data moving to the position exceeding the n1 limit is diverted to an overflow area. The bit shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

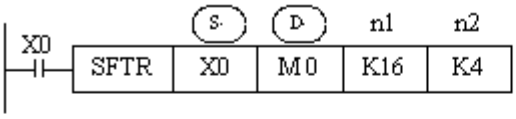
《Bit shift left》



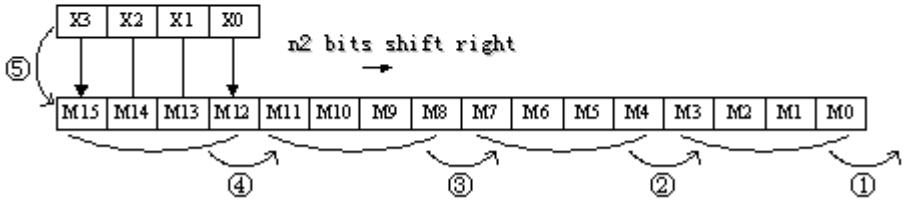
① M15~M12→overflow
② M11~M 8→M15~M 12
③ M 7~M 4→M11~M8
④ M 3~M 0→M7~M4
⑤ X 3~X 0→M3~M0



《Bit shift right》



① M 3~M 0→overflow
② M 7~M 4→M3~M0
③ M11~M 8→M7~M4
④ M15~M12→M11~M8
⑤ X 3~X 0→M15~M12



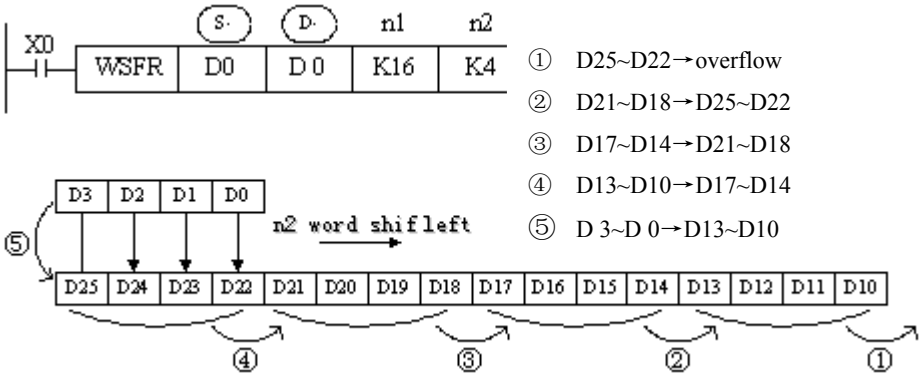
[WSFL] and [WSFR]

Operands：DX、DY、DM、DS、T、C、D

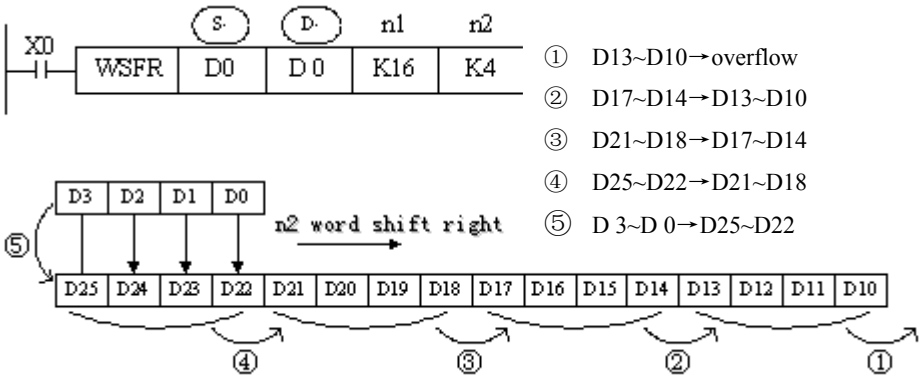
Function and action

- The instruction copies n2 source devices to a word stack of length n1. For each addition of n2 words, the existing data within the word stack is shifted n2 words to the left/right. Any word data moving to a position exceeding the n1 limit is diverted to an overflow area. The word shifting operation will occure everytime the instruction is processed unless it is modified with either the pulse suffix or a controller interlock.

《Word shift left》



《Word shift right》



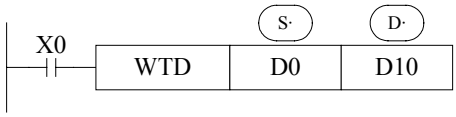
6-7. Data convertion

Mnemonic	Function
WTD	Single word integer converts to double word integer
FLT	32 bits integer converts to float point
FLTD	64 bits integer converts to float point
INT	Float point converts to integer
BIN	BCD convert to binary
BCD	Binary converts to BCD
ASC	Hex. converts to ASCII
HEX	ASCII converts to Hex.

[WTD]

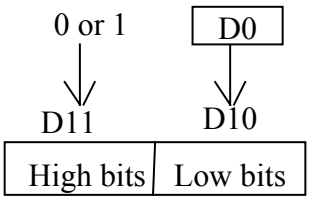
Operands：DX、DY、DM、DS、T、C、D

Function and action



（D0） → （D11，D10）

Single word Double word



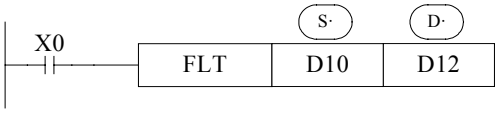
- When single word D0 is positive integer, after executing this instruction, the high bit of double word D10 is 0.
- When single word D0 is negative integer, after executing this instruction, the high bit of double word D10 is 1.

[FLT] and [FLTD]

Operands：DX、DY、DM、DS、T、C、D

Function and action

《16 bits》



(D10) → (D13,D12)

BIN integer Binary float point

《32 bits》

（D11,D10）→ （D13,D12）

BIN integer Binary float point

《64 bits》

（D13,D12,D11,D10）→ （D17,D16,D15,D14）

BIN integer Binary float point

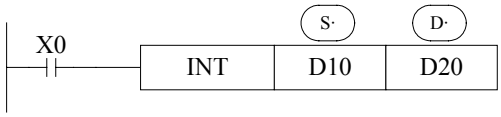
- Convert BIN integer to binary float point. As the constant K、H will auto convert by the float operation instruction, so this FLT instruction can't be used.
- The instruction is contrary to INT instruction.

[INT]

Operands：DX、DY、DM、DS、T、C、D

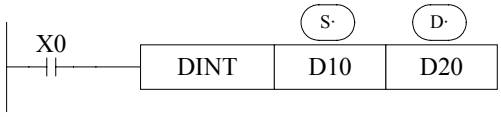
Function and action

《16 bits》



（D11,D10） → (D20)

《32 bits》



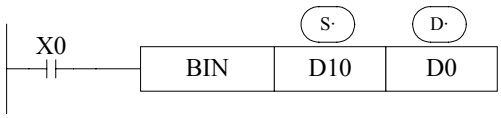
（D11,D10） → (D20)

- The binary source number is converted into an BIN integer and stored at the destination device. Abandon the value behind the decimal point.
- This instruction is contrary to FLT instruction.
- When the result is 0, the flag bit is ON。
  When converting, less than 1 and abandon it, zero flag is ON.
  16 bits operation：-32，768~32，767
  32 bits operation：-2，147，483，648~2，147，483，647

BCD converts to [BIN]

Operands：DX、DY、DM、DS、T、C、D

Function and action



Data's bound：0~9,999 or 0~99，999，999 is valid.

Convert and move instruction of Source（BCD）→ destination（BIN）

- When source data is not BCD code, M8067（Operation error）， M8068（Operation error lock）will not work.
- As constant K automatically converts to binary, so it's not suitable for this instruction.

Binary converts to BCD [BCD]	
	Operands：DX、DY、DM、DS、T、C、D

Function and action

Convert and move instruction of source (BIN)→destination (BCD).

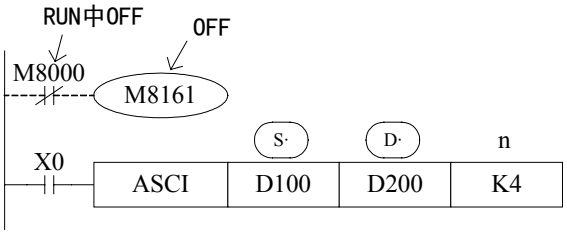
- When use BCD instruction, if the converted BCD number exceeds the operational ranges of 0 to 9999(16 bits operation) and 0 to 99999999 (32 bit operation) an error will occur.
- This instruction can be used to output data directly to a seven segment display.

Hex. converts to ASCII [ASCI]

Operands: DX、DY、DM、DS、T、C、D

Function and action

《16 bits convert mode》 When M8161=OFF



Convert each bit of source's (S) Hex. format data to be ASCII code, move separately to the high 8 bits and low 8 bits of destination (D). The convert alphanumeric number is assigned with n.

(D) is low 8 bits, high 8 bits, store ASCII data.

The convert result is the following：

Assign start device：

(D100)=0ABCH

(D101)=1234H

(D102)=5678H

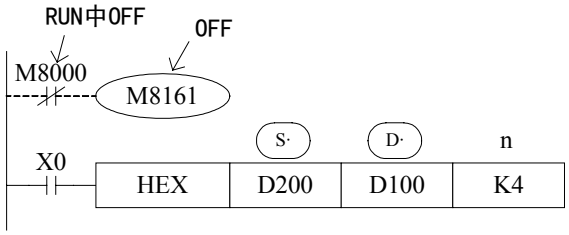
D \ n	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200 down	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D200 down		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D201 down			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D201 up				[C]	[B]	[A]	[0]	[4]	[3]
D202 down					[C]	[B]	[A]	[0]	[4]
D202 up						[C]	[B]	[A]	[0]
D203 down							[C]	[B]	[A]
D203 up								[C]	[B]
D204 down									[C]

ASCII converts to [HEX]

Operands：DX、DY、DM、DS、T、C、D

Function and action

《16 bits switch mode》 When M8161=OFF

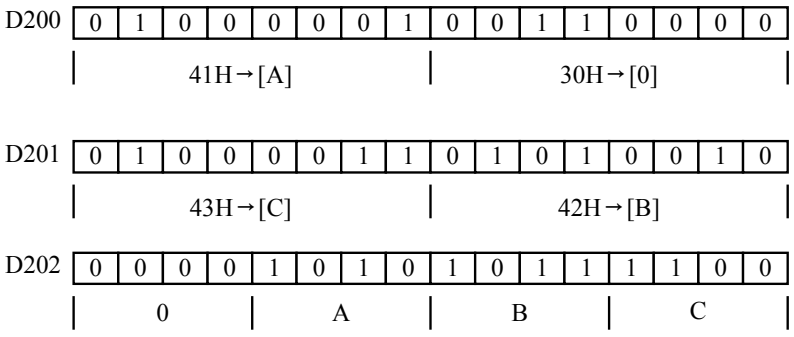


Convert the high and low 8 bits in source to HEX data. Move 4 bits every time to destination. The convert alphanumeric number is assigned by n.

The convertion of the upward program is the following：

(S・)	ASCII	HEX Conversion
D200 up	30H	0
D200 up	41H	A
D201 down	42H	B
D201 up	43H	C
D202 down	31H	1
D202 up	32H	2
D203 down	33H	3
D203 up	34H	4
D204 down	35H	5

n \ (D・)	D102	D101	D100
1	Not change to be 0		••• 0H
2			•• 0AH
3			• 0ABH
4			0ABCH
5		••• 0H	ABC1H
6		•• 0AH	BC12H
7		• 0ABH	C123H
8		0ABCH	1234H
9	••• 0H	ABC1H	2345H



6-8. Floating Operation

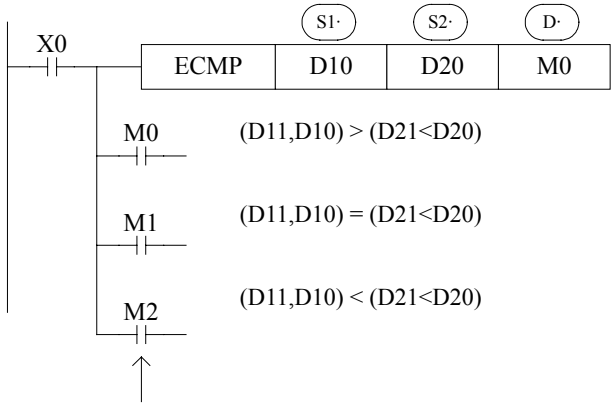
Mnemonic	Function
ECMP	Float Compare
EZCP	Float Zone Compare
EADD	Float Add
ESUB	Float Subtract
EMUL	Float Multiplication
EDIV	Float Division
ESOR	Float Square Root
SIN	Sine
COS	Cosine
TAN	Tangent

[ECMP]

Operands：DX、DY、DM、DS、T、C、D、K

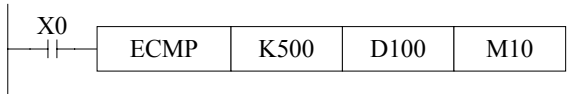
Function and action

（D11,D10） ： （D21,D20）→M0,M1,M2



The status of the destination devicse will be kept even if the ECMP instruction is deactivated.

- The binary float data of S1 is compared to S2. The result is indicated by 3 bit devices specified with the head address entered as D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



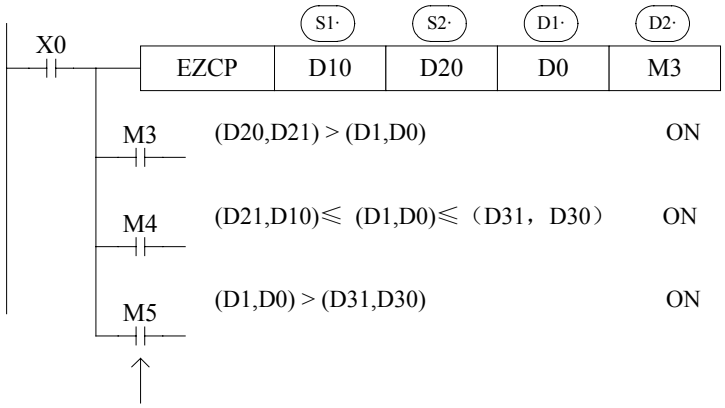
（K500）：（D101，D100）→M10,M11,M12

[EZCP]

Operands：DX、DY、DM、DS、T、C、D、K

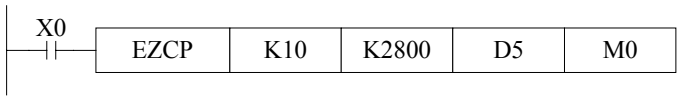
Function and action

Compare a float range with a float value



The status of the destination devicse will be kept even if the EZCP instruction is deactivated.

- The data of S1 is compared to the data of S2. The result is indicated by 3 bit devices specified with the head address entered as D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



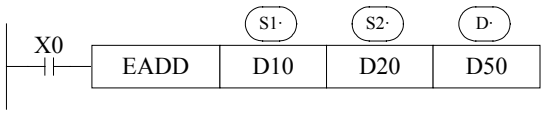
（K10）：[D6,D5]：（K2800）→M0，M1，M2

- Please set S1<S2, when S2>S1, see S2 as the same with S1 and compare them.

[EADD]

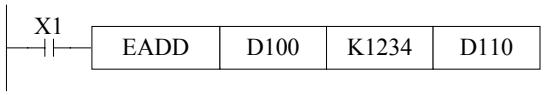
Operands：DX、DY、DM、DS、T、C、D、K

Function and action



（D11,D10） + (D21,D20) → (D51,D50)

- The floating point values stored in the source devices S1 and S2 are algebraically added and the result stored in the destination device D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



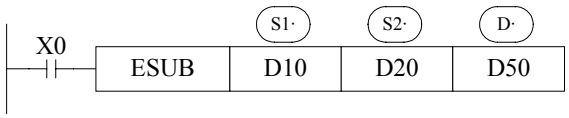
（K1234） + (D101,D100) → (D111,D110)

- The same device may be used as a source and as the destination. If this is the case then,on continuous operation of the EADD instruction, the result of the prevous operation will be used as a new source value and a new result calculated. This will happen every program scan unless the pulse modifier or an interlock program is used.

[ESUB]

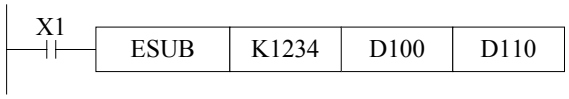
Operands：DX、DY、DM、DS、T、C、D、K

Function and action



(D11,D10) － (D21,D20) → (D51,D50)

- The floating point value of S2 is subtracted from the floating point value of S1 and the result stored in destination device D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



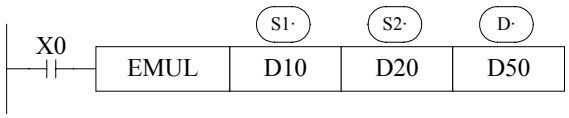
（K1234） － (D101,D100) → (D111,D110)

- The same device may be used as a source and as the destination. If this is the case then,on continuous operation of the EADD instruction, the result of the prevous operation will be used as a new source value and a new result calculated. This will happen every program scan unless the pulse modifier or an interlock program is used.

[EMUL]

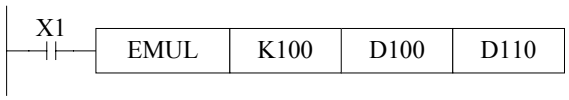
Operands：DX、DY、DM、DS、T、C、D、K

Function and action



（D11，D10）×（D21,D20）→(D51,D50)

- The floating point value of S1 is multiplied with the floating point value point value of S2. The result of the multiplication is stored at D as a floating point value.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.

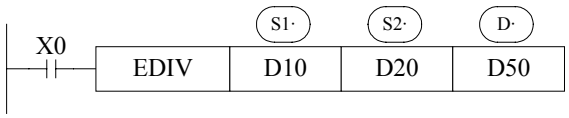


(K2346) × (D101,D100) → (111,D110)

[EDIV]

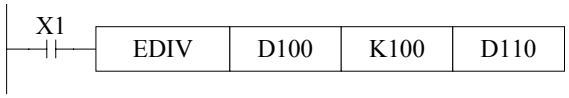
Operands：DX、DY、DM、DS、T、C、D、K

Function and action



（D11,D10）÷（D21,D20）→（D51,D50）

- The floating point value of S1 is divided by the floating point value of S2. The result of the division is stored in D as a floating point value. No remainer is calculated.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



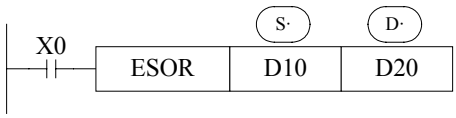
(D101,D100) ÷ (K2346) →(D111,D110)

- If S2 is zero then a divide by zero error occurs and the operation fails.

[ESOR]

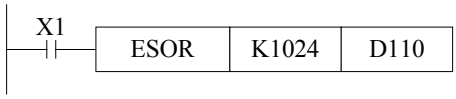
Operands：DX、DY、DM、DS、T、C、D、K

Function and action



(D11,D10) →（D21,D20）

- A square root is performed on the floating point value in S the result is stored in D.
- If a constant K or H used as source data, the value is converted to floating point before the addition operation.



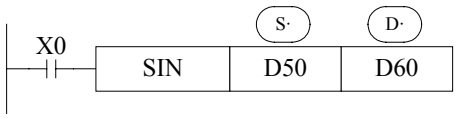
（K1024）→（D111，D110）

- When the result is zero, zero flag activates
- Only when the source data is positive will the operation be effective. If S is negative then an error occurs and error flag M8067 is set ON, the instruction can't be executed.

[SIN]

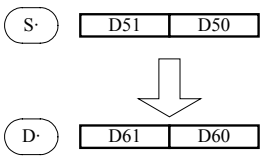
Operands：DX、DY、DM、DS、T、C、D、K

Function and action



(D51,D50) → (D61,D60)SIN

- This instruction performs the mathematical SIN operation on the floating point value in S (angle RAD). The result is stored in D.



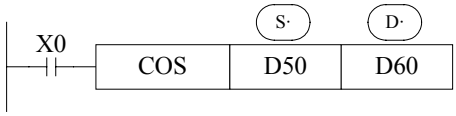
RAD value（angle×Π/180）
Assign binary float point value

SIN value
Binary float point

[COS]

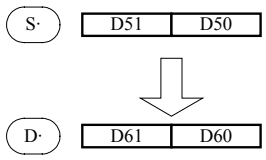
Operands：DX、DY、DM、DS、T、C、D、K

Function and action



(D51,D50)RAD → (D61,D60)COS

- This instruction performs the mathematical COS operation on the floating point value in S (angle RAD). The result is stored in D.



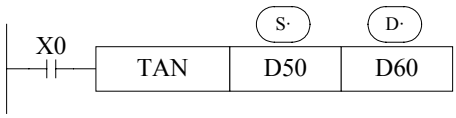
RAD value（angle×Π/180）
Assign binary float point value

SIN value
Binary float point

[TAN]

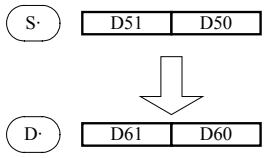
Operands：DX、DY、DM、DS、T、C、D、K

Function and action



(D51,D50)RAD → (D61,D60)TAN

- This instruction performs the mathematical TAN operation on the floating point value in S. The result is stored in D.



RAD value（Angle×Π/180）
Assign binary float point value

SIN value
Binary float point

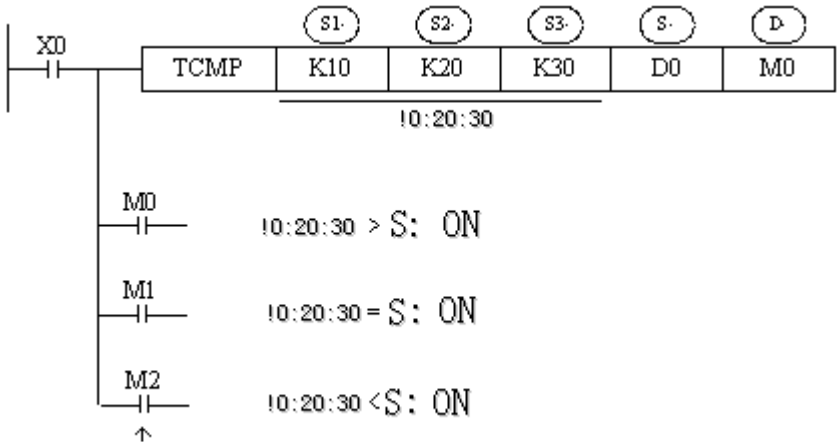
6-9. Clock operation

Mnemonic	Function
TCMP	Time Compare
TZCP	Time Zone Compare
TADD	Time Add
TSUB	Time Subtract
TRD	Read RTC data
TWR	Set RTC data

[TCMP]	
	Operands：DX、DY、DM、DS、T、C、D、K

Function and action

Compare the assigned time with time data.



The status of the destination devices is kept, even if the TCMP instruction is deactivated.

- (S1·), (S2·) and (S3·) represent hours, minutes and seconds respectively. This time is compared to the time value in the 3 data devices specified by the head address (S·). The result is indicated in the 3 bit devices specified by the head address (D·).

(S1·) ：Hour
(S2·) ：Minute
(S3·) ：Second

(S·) ：Hour
(S·) ＋1 ：Minute
(S·) ＋2 ：Second

(D·)，(D·)＋1，(D·)＋2 ：According to the compare result, the 3 devices output ON/OFF.

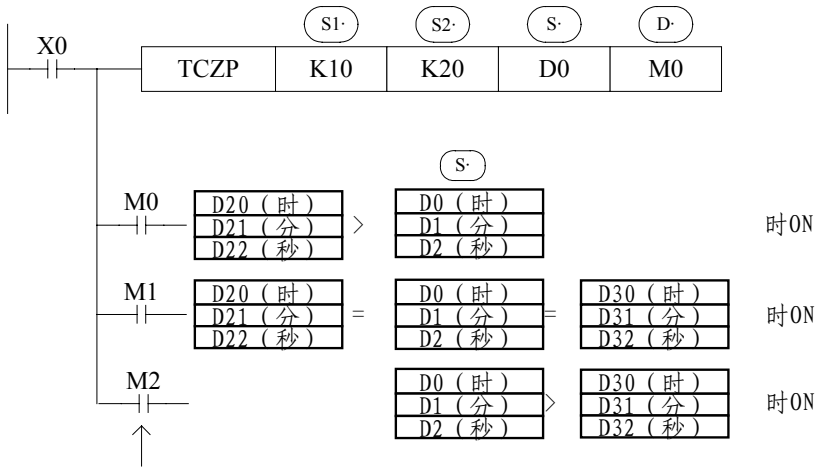
The valid range of “Hour” is「0~23」。
The valid range of “MInute” is「0~59」。
The valid range of “Second” is「0~59」。

[TZCP]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action

Compare the two assigned time with time data



The status of the destination devices is kept, even if the TZCP instruction is deactivated.

- (S1·), (S2·) and (S·) represent time values. Each specifying the head address of 3 data devices. (S·) is compared to the time period defined by (S1·) and (S2·). The result is indicated in the 3 bit devices specified by the head address (D·).

(S1·), (S1·)+1, (S1·)+2 : Assign the compare time's lower limit with the format of "Hour", "Minute" and "Second".

(S2·), (S2·)+1, (S2·)+2 : Assign the compare time's lower limit with the format of "Hour", "Minute" and "Second".

(S·), (S·)+1, (S·)+2 : Assign the time data with the format of "Hour", "Minute" and "Second".

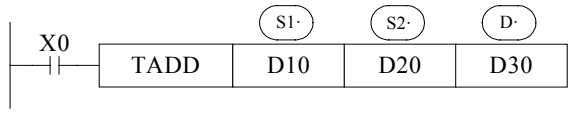
(D·), (D·)+1, (D·)+2 : According to the compare result, the 3 devices output ON/OFF.

The valid range of "Hour"is「0~23」。
The valid range of "Minute"is「0~59」。
The valid range of "Second"is「0~59」。

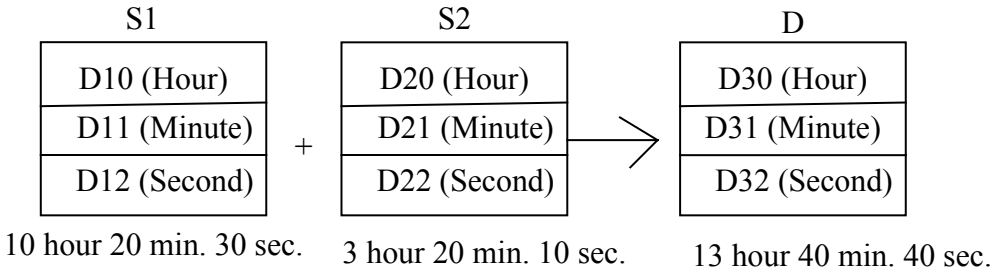
[TADD]

Operands：DX、DY、DM、DS、T、C、D、K

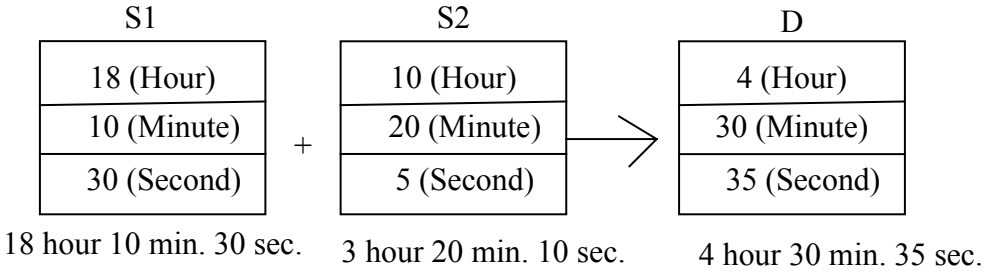
Fnction and action



(D10，D11，D12)＋（D20，D21，D22）→（D30，D31，D32）



- Each of S1, S2 and D specify the head address of 3 data devices to be used a time value. The time value in S1 is added to the value in S2, the result is stored to D as a new time value.
- If the addition of the two times results in a value greater than 24 hours, the value of the result is the time remaining above 24 hours. When this happens the carry flag M8022 is set ON.



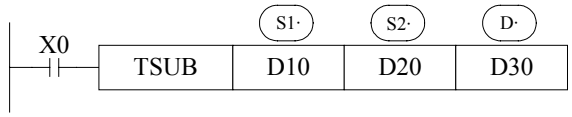
- When the result is 0（0 Hour 0 Minute 0 Second），Set zero flag ON.

The valid range of “Hour”is「0~23」。
The valid range of “Minute”is「0~59」。
The valid range of “Second”is「0~59」。

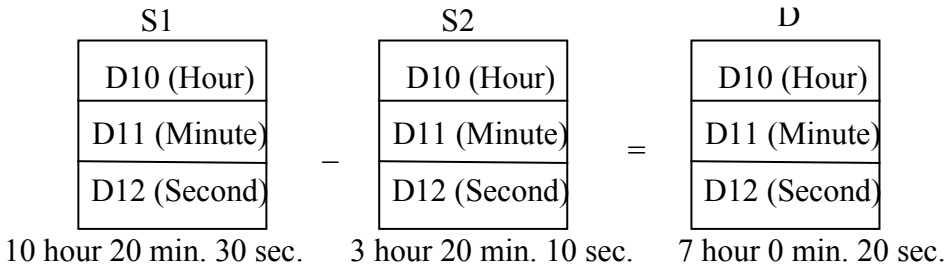
[TSUB]

Operands：DX、DY、DM、DS、T、C、D、K

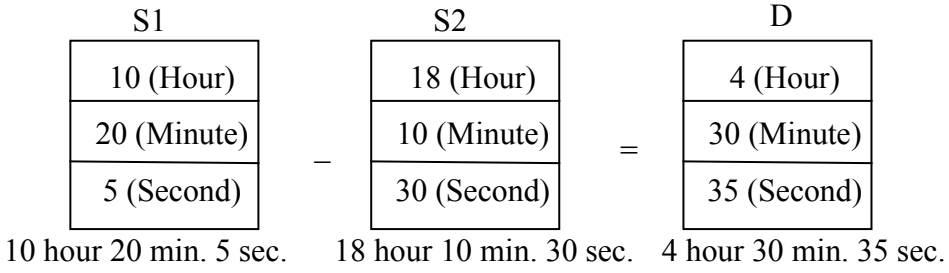
Function and action



(D10，D11，D12)－（D20，D21，D22）→（D30， 31，D32）



- Each of S1, S2 and D specify the head address of 3 data devices to be used a time value. The time value in S1 is subtracted from the time value in S2, the result is stored to D as a new time.
- If the subtraction of the two times results in a value less than 00:00:00 hours, the value of the result is the time remaining below 00:00:00 hours. When this happens the borrow flag M8021 is set ON.



- When the result is 0（0 hour 0 min. 0 sec.）, zero flag set ON.

The valid range of “Hour”is「0~23」。
The valid range of “Minute”is「0~59」。
The valid range of “Second”is「0~59」。

[TRD]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action

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

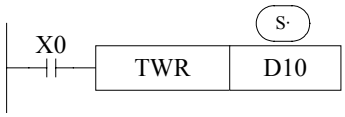
- Read PLC's real time clock according to the following format.
  The reading source is the special data register （D8013~D8019）which save clock data.

Device	Meaning	Values	Device	Meaning
D8018	Year	1~99	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	Day	0 (Sat.)~6 (Sun.)	D6	Day

[TWR]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action



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

- Write the set clock data into PLC's real time clock.
  In order to wirte real time clock, the 7 data devices specified with the head address S should be set.

Device	Meaning	Values	Device	Meaning
D0	Year	1~99	D8018	Year
D1	Month	1~12	D8017	Month
D2	Date	1~31	D8016	Date
D3	Hours	0~23	D8015	Hours
D4	Minutes	0~59	D8014	Minutes
D5	Seconds	0~59	D8013	Seconds
D6	Day	0 (Sat.)~6 (Sun.)	D8019	Day

This instruction removes the need to use M8015 during real time clock setting. When setting the time it is a good idea to set the source data to a time a number of minutes ahead and then drive the instruction when the real time reaches this value.

7. High speed count and pulse output

This chapter, we'll tell XC series PLC's high speed count and pulse output function .

7-1.Interior high speed counter's No. and function

High speed counter's No.

Interior high speed counter's No. is in the following table. They're allocatedin the input X000~X021 according to the counter's No. thay cannot be used repeatedly.

The input No.s which are not used as high speed counter can be used as ordinal input relay in the sequencial control

[U]：Increase count input；[D]：decrease count input；[A]：A phase input；[B]：B phase input

	Single phase count										Single phase double input					AB phase count		
	C600	C602	C604	C606	C608	C610	C612	C614	C616	C618	C620	C622	C624	C626	C628	C630	C632	C634
X000	U										U					A		
X001											D					B		
X002		U										U					A	
X003												D					B	
X004			U										U					
X005				U									D					
X006					U													
X012						U									U			A
X013															D			B
X014							U											
X017								U										
X020									U									
X021										U								

Function

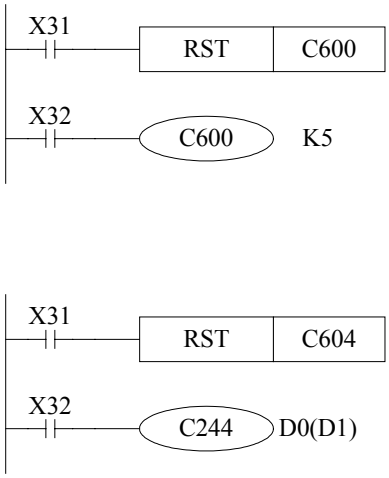
High speed counter executes according to the format in the upward table and to the special inputs. Go on high speed action according to the interrupt disposal. It's independent with the PLC's scan cycle.

Item	Single phase single count input	Single phase double count input	Double phases double count input
The appoint method of count direction	Can only execute increase count	Correspond with the action of increase count input and minus count input, auto increase/decrease count.	When A phase input is ON, if B phase is OFF → ON, increase count acts, If ON → OFF, decrease count acts.

7-2.Using method of single direction high speed counter

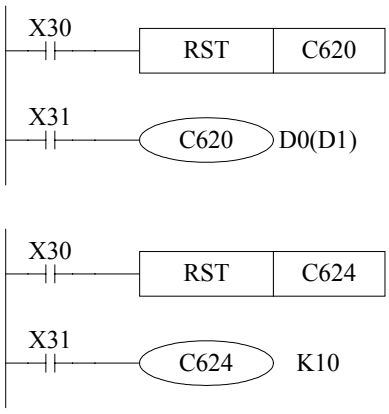
Action Example

Single direction single input



- When X032 activates, C600 is ON and count the ON/OFF status of input X000
- When X032 activates, execute RST instruction.
- When X032 activates, C604 starts to count. The count input is X004，In this example the set value is the content in data register whichis indirectly appointed.
- It's also available to execute reset via X031 in the sequencial control program
- Single phase high speed count has only increase count, but no decrease count

Single direction double input

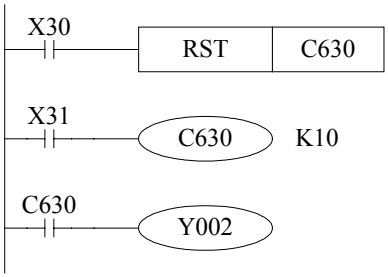


- When X031 activates, C620 is ON, execute increase count via input OFF → ON of X001, execute decrease count via input OFF→ON of X001.
- When X031 and X004 are both ON, C624 starts to count
  The input for increase count is X004, input for decrease count is X005
- Execute reset via X031 in sequencial control program.

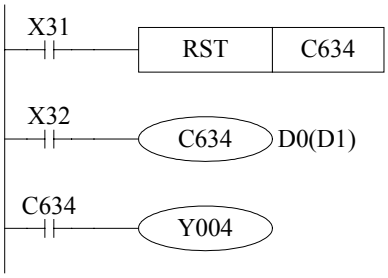
7-3.Using method of double high speed counter

Double directions double input

Double directions double input counter is a 32 bits increase/decrease binary counter. The action corresponds with the current value's output contact is same with the single direction high speed counter said before.



- When X032 is ON, C630 count the action of the input X000(A phase) 、 X001(B phase) via interruption.
  If X030 is ON, then execute RST instruction reset
- If the current value exceeds the set value, then Y002 is ON；If the current value is less than the set value，then Y002 is OFF.

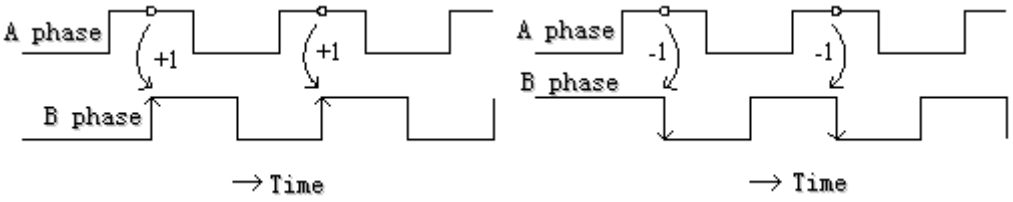


- When X032 is ON, C624 starts to count immediately. The count input is X012(A phase)、X013(B phase)
- Reset in the sequencial control program via X031
- When the current value is larger than the set value, Y004 is active, when smaller than the set value, then Y004 is not active.

Double directions double input

- When counter's A phase gets, if B phase input is OFF→ON, then it is increase count, if B phase is ON→OFF, then it is decrease count.

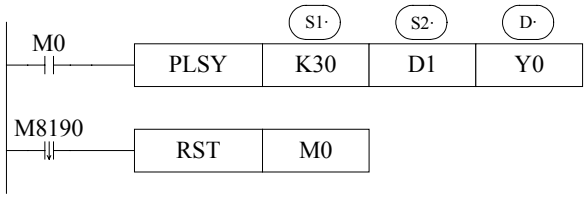
- The output of double phase encoder is A phase and B phase with 90 degrees phase difference. So, the high speed counter will automatic increase/decrease count as in the following chart:
- This type of counter acts as a counter with increment by one time.



The up line acts when roll forward

The down line acts when roll backward

Function and action



- A specified quantity of pulses S2 is output through device D at a specified frequency S 1.

  (S1·) assign the frequency. Range: 0~200Hz

  (S2·) assign the pulse's quantity

  The max pulse number of pulse: 16 bit operation→1~32,767

  32 bit operation→1~2,147,483,647

  If set the value to be zero, then the generate pulse number is not limit.

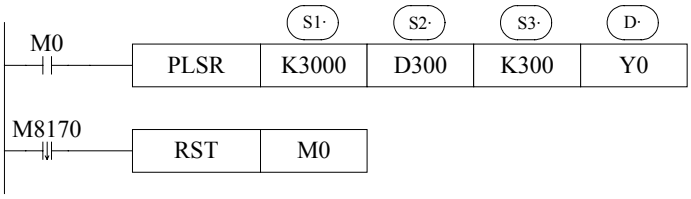
  (D·) assign the output pulse's Y number, it can output freely at Y port.

  - When M000 is ON, PLSY instruction output 30Hz pulse at Y000, the number is assigned by D001. If set the pulse number to be 0, it means send unlimit number of pulses. Then set M8190 ON. Set coil M8190 OFF when the specified number of pulses has been completed. Also reset M000.

7-5. [PLSR]

Operands：DX、DY、DM、DS、T、C、D、K

Function and action



- A specified quantity of pulses S2 is output through device D at a specified frequency S 1.

  (S1·) Assign the highest frequency. Range: 200~100,000Hz

  (S2) Assign total the pulse's quantity

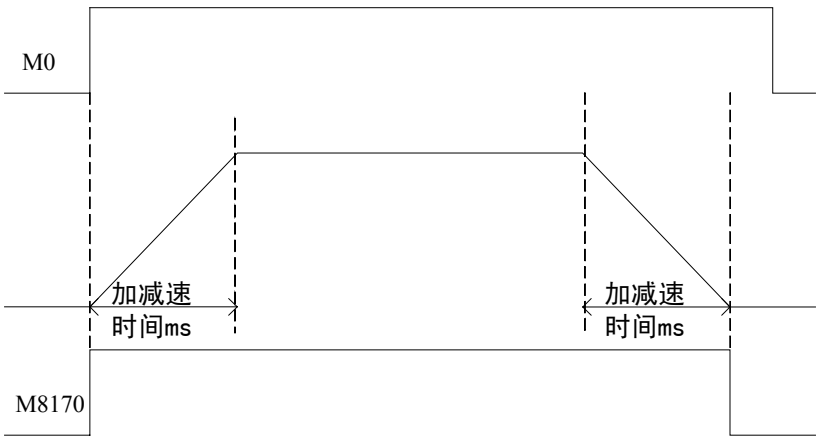
  The max pulse number of pulse: 16 bit operation→1~32,767
  32 bit operation→1~2,147,483,647

  If set the value to be zero, then the generate pulse number is not limit.

  (S3) Speedup/speed-down time. Range: below 5000ms

  (D·) Assign the output pulse's Y number, only Y000 and Y001 is available.

- When M000 is ON, PLSR instruction starts to output pulse. Generate the assigned pulse number according to the assigned acceleration/deceleration time, highest frequency. If want to output with a certain frequency, you can set acceleration/deceleration time to be 0. If set the pulse number to be 0, it can send out unlimit number of pulses. Then set coil M8170 ON.
- When the pulse number reach the set value, stop pulse outputting. Coil M8170 set OFF, reset M000, see the following chart:



8. Application program example

In this chapter, we give some application examples of XC series PLC.
XC series PLC is miniature with high speed, powerful capacity. Besides the independent application of input/output, they also can be used to pulse output etc. So they can satisfy diverse control.

8-1．Simply application of pulse output

E.g.：The following is the program which send alternate high pulse and low pulse.

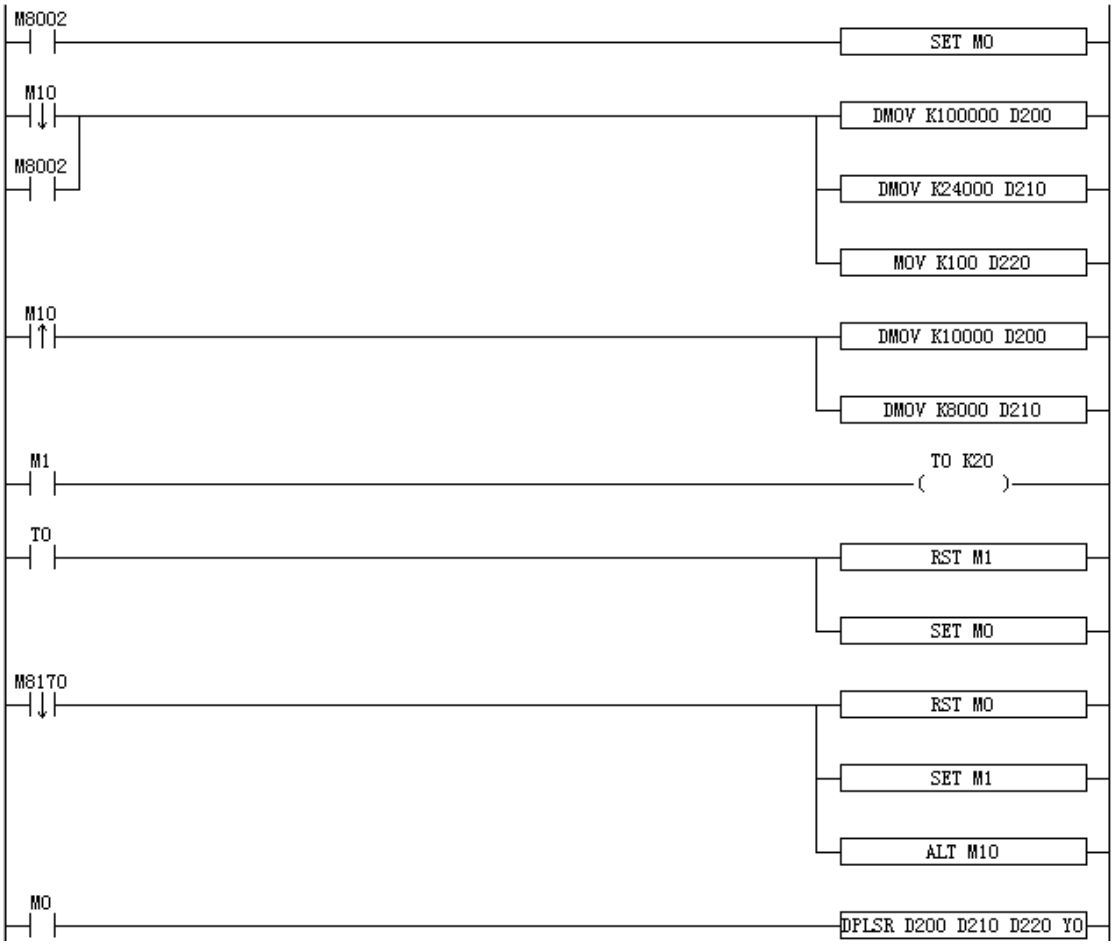
Each parameter：

Step motor's parameter：step square angle =1.8 degree / step，subsection number =40，the totalpulse number after rotate a round is 8000.

High frequency pulse：Highest frequency 100KHz，the total pulse number 24000（3 rounds）

Low frequency pulse：The lowest frequency 10KHz，the total pulse number 8000（1 round）

Ladder program：



Program description：

When PLC changes from STOP to RUN，M8002 gets once scan，set the high frequency M0.

peedup rotate 3 rounds with high frequency, set coil M8170 at the same time. The motor rotates 3 rounds and speed-down till stop. Coil M8170 reset, then reset M0, set M1, inverse the state of M10, set the low frequency pulse's parameter into D200、D210. The counter

with low frequency, after finish running, start to run with high frequency. Loop to run with a

```
LD      M0
DPLSR   D200   D210   D220   Y0
LDF     M8170
RST     M0
SET     M1
ALT     M10
LDF     M8170
OR      M8002
DMOV    K100000   D200
DMOV    K24000    D210
MOV     K100      D220
LD      M8002
SET     M0
LDP     M10
DMOV    K10000   D200
DMOV    K8000    D210
LD      M1
OUT     T0   K20
LD      T0
RST     M1
SET     M0
```

9. Appendix

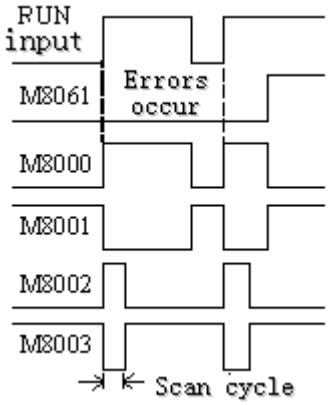
This chapter gives some auxiliary information of XC series PLC.

9-1. List of special auxiliary relay, special data register

9-1．List of special auxiliary relay、special data register

For the special soft units' kind and their function, please see the following description.

PC status（M）

Address ID	Function	Description
M8000	Run and monitor a contact	
M8001	Run and monitor b contact	
M8002	Initial pulse a contact	
M8003	Initial pulse b contact	
M8004	Errors occur	Take action when any of M8060~M8067 is ON（Except M8062）
M8005	Battery low	Take action when the battery is abnormal low
M8006	Battery too low, lock and save	Lock the status when battery abnormal low
M8007	Momentary stop and check	Even though M8007 acts，if in the time bound of D8008, then PC goes on running
M8008	Power cut and checking	When M8008 ON→OFF，M8000 turns to be OFF
M8009	DC24V drop power	Take action when the extend unit，extend module appeare DC24V drop power

PC status（D）

Address ID	Function	Description
D8000	Monitor timer	Initial value 200ms，when use program to modify,itwill be effective after END，WDT instructions
D8001	PC type and system's version No.	BCD code：D4- serial No.；D3- model；D2~0- version No.
D8002	Register's capacity	2…2K step；4…4Kstep；8…8Kstep
D8003	Register's type	Save different RAM/EEPROM/inside EPROM/ Save memory protection switch's ON/OFF status
D8004	Wrong M address ID	
D8005	Battery	0.1V unit
D8006	Battery too low Check bound	Initial 3.0v（0.1v unit）（When power is ON，transferby system ROM）
D8007	Momentary stop and check	Save the action times of M8007。When the power is cut, the value will be cleared.
D8008	Power cut check time	AC power type，the detail things of initial value 10ms is said later
D8009	DC24V drop power	DC24V power drop basic unit、the minimum input device's ID in the extension

Clock（M）

Address ID	Function	Description
M8010		
M8011	Tosc= 10ms	
M8012	Tosc= 100ms	
M8013	Tosc= 1 second	
M8014	Tosc= 1 minute	
M8015	Clock stop and lay previously	
M8016	Time read displays stop	
M8017	±30 seconds amend	
M8018	Year’s bit	Be OFF when defaulted
M8019	Clock error	

Sy

Address ID	Function	Description
M8020	Zero	When plus or minus result is 0
M8021	Borrow a bit	Appear borrow in minus operation
M8022	Carry	When appear carry in plus operation or overflow in shift operation
M8023		
M8024	BMOV direction assignment	
M8025	HSC mode	
M8026	RAMP mode	
M8027	PR mode	
M8028	100ms/10ms timer switch	
M8029		

Clock（D）

ID	Function	Description
D8010	Current scan cycle	0.1s as the unit
D8011	Minimum value of scan time	0.1s as the unit
D8012	Maximum value of scan time（0.1s as unit）	
D8013	Second（Hour）	0~59
D8014	Minute（Clock）	0~59
D8015	Hour（Clock）	0~59
D8016	Day（Clock）	0~31
D8017	Month（Clock）	0~12
D8018	Year（Clock）	0~99
D8019	Week（Clock）	0（Sun.）~6（Sat.）

Symbol（D）

ID	Function	Description
D8020	Enter filter time	Enter filter value 0~50(Initial value is 10ms)
D8021		
D8022		
D8023		
D8024		
D8025		
D8026		
D8027		
D8028		
D8029		

PC mode（M）

ID	Function	Description
M8030	Battery LED lamp display	After drive M8030, even battery too low, the indicate lamp on PC panel won't light
M8031	Not rententive register clear	When drive this M, the current value of Y,M,S,TC's ON/OFF image memory and T,C,D will be cleared.
M8032	Rententive register clear	
M8033	Memory keep stop	When PLC turns from RUN to STOP, keep the content is the image register and data register
M8034	All output forbidden	Set all PC's exterior contacts OFF
M8035	Forceto be be run mode	
M8036	Force to run the instruction	
M8037	Force to stop instruction	
M8038	Parameter setting	Flag of communication parameter
M8039	Invariable scan mode	When M8039 turns to be ON, PC executes loop operation till D8039 reachs the assigned scan cycle time

PC mode（D）

ID	Function	Description
D8030		
D8031		
D8032		
D8033		
D8034		
D8035		
D8036		
D8037		
D8038		
D8039	Eternal scan time	Initial value 0ms（Take 1ms as unit）（When power is ON, transferred by system ROM）can be modified via the program.

Step ladder（M）

ID	Function	Description
M8040	Forbid transfer	When M8040 is driven, transfer among the forbidden status
M8041		
M8042	Initial pulse	Pulse output corresponds with start input
M8043	Return finished	Be active when reach the finish status of origin back mode
M8044	Origin condition	Be active when test out the machine's origin
M8045	All outputs reset forbidden	All output reset forbidden when mode switch
M8046	STL status active	In the action of M8047, act when any of S900~S999turns to be ON.
M8047	STL monitor effective	When drive this M, D8040~D8047 are valid
M8048	Signal alarm is active	In the action of M8049, act when any of S900~S999turns to be ON.
M8049	Signal alarm is active	When drive this M, the action of D8049 is valid

Interrupt（M）

ID	Function	Description
M8050 I00□		After EI operation, even allow interruption, but when this M takes action, the correspond input interruption can't take action singlely. E.g.：When M8050 is ON, forbid to interrupt I00 port
M8051 I10□		
M8052 I20□		
M8053 I30□		
M8054 I40□		
M8055 I50□		
M8056 I60□		After EI operation, even allow interruption, but when this M takes action, the correspond input interruption can't take action singlely.
M8057 I70□		
M8058 I80□		
M8059	Counter interrupt forbidden	Forbid the interruption from I010~I060

Step ladder（D）

ID	Function	Description
D8040	ON status ID 1	Save the status's minimum ID among S0~S899 into D8040, save the next minimum ID into D8041, then save 8 points device like this, save the maximum device into D8047
D8041	ON status ID 2	
D8042	ON status ID 3	
D8043	ON status ID 4	
D8044	ON status ID 5	
D8045	ON status ID 6	
D8046	ON status ID 7	
D8047	ON status ID 8	
D8048		Symbol of current S
D8049	ON status minimum ID	Save the minimum ID of alarm relay S900~S999 which is in the status of ON

Error check（M）

ID	Function	Description
M8060	I/O constitution error	
M8061	PC hardware error	Give power and STOP->RUN check
M8062	PC/PP communication error	
M8063	RS232 communication error	
M8064	Parameter error	Give power and STOP->RUN check
M8065	Grammer error	Give power and STOP->RUN check
M8066	Circuit error	Give power and STOP->RUN check
M8067	Operation error	Give power and STOP->RUN check
M8068	Operation error lock	
M8069	I/O bus check	
M8070	Scan time out	
M8071	No user program	Interior code check error
M8072	User program error	Executing code or list check error
M8099	High speed circuit counter	
M8109	Output refresh error	

Error check（D）

ID	Function	Description
D8060	The start ID of I/O error	
D8061	Number of PC hardware error code	
D8062	Number of PC/PP communication error code	
D8063	Number of parameter connection communication error code	
D8064	Number of parameter error code	
D8065	Number of grammer error code	
D8066	Number of circuit error code	
D8067	Number of operation error code	Divide 0 error
D8068	Number of lock save error code	
D8069		
D8070	The scan time of time out	1ms unit
D8071		
D8072		
D8099	High speed circuit counter	
D8109	Y ID which appears output refresh error	

Communication（M）

ID	Function	Description	
M8120			Serial Port 1
M8121	RS232 send waiting		
M8122	RS232 sending flag		
M8123	RS232 finish receiving flag		
M8124	RS232 receiving flag		
M8125	Receive not integrate flag	Reception end in gear, but the received data's number is less than the needed.	
M8126	Global signal		
M8127	Receive error flag		
M8128	Receive correct flag		
M8129	Time out judgement flag		

ID	Function	Description	
M8130			Serial Port 2
M8131	RS232 send waiting		
M8132	RS232 sending flag		
M8133	RS232 finish receiving flag		
M8134	RS232 receiving flag		
M8135	Receive not integrate flag	Reception end in gear, but the received data's number is less than the needed.	
M8136	Global signal		
M8137	Receive error flag		
M8138	Receive correct flag		
M8139	Time out judgement flag		

ID	Function	Description	
M8140			Serial Port 3
M8141	RS232 send waiting		
M8142	RS232 sending flag		
M8143	RS232 finish receiving flag		
M8144	RS232 receiving flag		
M8145	Receive not integrate flag	Reception end in gear, but the received data's number is less than the needed.	
M8146	Global signal		
M8147	Receive error flag		
M8148	Receive correct flag		
M8149	Time out judgement flag		

Communication（D）

ID	Function	Description	
D8120			Serial Port 1
D8121			
D8122	RS232 transfers data's left number		
D8123	RS232 receives data's number		
D8124			
D8125			
D8126			
D8127	Communication error codes	0: Hardware error 10:No start symbol 8: CRC check error 11:No end symbol 9: Bureau ID error	
D8128			
D8129			

ID	Function	Description	
D8130			Serial Port 2
D8131			
D8132	RS232 transfers data's left number		
D8133	RS232 receives data's number		
D8134			
D8135			
D8136			
D8137	Communication error codes	0: Hardware error 10:No start symbol 8: CRC check error 11:No end symbol 9: Bureau ID error	
D8138			
D8139			

ID	Function	Description	
D8140			Serial Port 3
D8141			
D8142	RS232 transfers data's left number		
D8143	RS232 receives data's number		
D8144			
D8145			
D8146			
D8147	Communication error codes	0: Hardware error 10:No start symbol 8: CRC check error 11:No end symbol 9: Bureau ID error	
D8148			
D8149			

High speed count（M）

ID	Counter's No.	Function	Description
M8150	C600	Count finish flag	24 segments count finished, flag is 1
M8151	C602	Count finish flag	24 segments count finished, flag is 1
M8152	C604	Count finish flag	24 segments count finished, flag is 1
M8153	C606	Count finish flag	24 segments count finished, flag is 1
M8154	C608	Count finish flag	24 segments count finished, flag is 1
M8155	C610	Count finish flag	24 segments count finished, flag is 1
M8156	C612	Count finish flag	24 segments count finished, flag is 1
M8157	C614	Count finish flag	24 segments count finished, flag is 1
M8158	C616	Count finish flag	24 segments count finished, flag is 1
M8159	C618	Count finish flag	24 segments count finished, flag is 1

ID	Counter's No.	Function	Description
M8160	C620	Count finish flag	24 segments count finished, flag is 1
M8161	C622	Count finish flag	24 segments count finished, flag is 1
M8162	C624	Count finish flag	24 segments count finished, flag is 1
M8163	C626	Count finish flag	24 segments count finished, flag is 1
M8164	C628	Count finish flag	24 segments count finished, flag is 1
M8165	C630	Count finish flag	24 segments count finished, flag is 1
M8166	C632	Count finish flag	24 segments count finished, flag is 1
M8167	C634	Count finish flag	24 segments count finished, flag is 1
M8168	C636	Count finish flag	24 segments count finished, flag is 1
M8169	C638	Count finish flag	24 segments count finished, flag is 1

Pulse output（M）

ID	High frequency pulse No.	Function	Discription
M8170	PULSE_1	Flag of ending out pulse	Be 1 in pulse output
M8171		32 bits pulse sending out overflow flag	Overflow is 1
M8172			
M8173	PULSE_2	Flag of ending out pulse	Be 1 in pulse output
M8174		32 bits pulse sending out overflow flag	Overflow is 1
M8175			
M8176	PULSE_3	Flag of ending out pulse	Be 1 in pulse output
M8177		32 bits pulse sending out overflow flag	Overflow is 1
M8178			
M8179	PULSE_4	Flag of ending out pulse	Be 1 in pulse output
M8180		32 bits pulse sending out overflow flag	Overflow is 1
M8181			
M8182	PULSE_5	Flag of ending out pulse	Be 1 in pulse output
M8183		32 bits pulse sending out overflow flag	Overflow is 1
M8184			

ID	Low frequency pulse No.	Function	Discription
M8190	PULSE_1	Sending out pulse sign	Be 1 in pulse output
M8191			
M8192	PULSE_2	Sending out pulse sign	Be 1 in pulse output
M8193			
M8194	PULSE_3	Sending out pulse sign	Be 1 in pulse output
M8195			
M8196	PULSE_4	Sending out pulse sign	Be 1 in pulse output
M8197			
M8198	PULSE_5	Sending out pulse sign	Be 1 in pulse output
M8199			
……			
M8236	PULSE_24	Sending out pulse sign	Be 1 in pulse output
M8237			

Sequencial / inverse count

Address No.	Counter's No.	Function	Discription
M8238	C300	Sequencial / inverse count control	1 is plus count，0 is minus count
……			

ID	Counter's No.	Function	Discription
D8150	C600	The current segment (meansthe No.n segment)	
D8151	C602	The current segment	
D8152	C604	The current segment	
D8153	C606	The current segment	
D8154	C608	The current segment	
D8155	C610	The current segment	
D8156	C612	The current segment	
D8157	C614	The current segment	
D8158	C616	The current segment	
D8159	C618	The current segment	

ID	Counter's No.	Function	Discription
D8160	C620	The current segment (means the No.n segment)	
D8161	C622	The current segment	
D8162	C624	The current segment	
D8163	C626	The current segment	
D8164	C628	The current segment	
D8165	C630	The current segment	
D8166	C632	The current segment	
D8167	C634	The current segment	
D8168	C636	The current segment	
D8169	C638	The current segment	

Pulse output（D）

ID	High frequency pulse No.	Function	Discription
D8170	PULSE_1	Accumulate low 16 bits pulse number	
D8171		Accumulate high 16 bits pulse number	
D8172		The current segment (means the No.n segment)	
D8173	PULSE_2	Accumulate low 16 bits pulse number	
D8174		Accumulate high 16 bits pulse number	
D8175		The current segment (means the No.n segment)	
D8176	PULSE_3	Accumulate low 16 bits pulse number	
D8177		Accumulate high 16 bits pulse number	
D8178		The current segment (means the No.n segment)	
D8179	PULSE_4	Accumulate low 16 bits pulse number	
D8180		Accumulate high 16 bits pulse number	
D8181		The current segment (means the No.n segment)	
D8182	PULSE_5	Accumulate low 16 bits pulse number	
D8183		Accumulate high 16 bits pulse number	
D8184		The current segment (means the No.n segment)	

ID	Low frequency pulse No.	Function	Discription
D8190	PULSE_1	Accumulate low 16 bits pulse number	
D8191		Accumulate high 16 bits pulse number	
D8192	PULSE_2	Accumulate low 16 bits pulse number	
D8193		Accumulate high 16 bits pulse number	
D8194	PULSE_3	Accumulate low 16 bits pulse number	
D8195		Accumulate high 16 bits pulse number	
D8196	PULSE_4	Accumulate low 16 bits pulse number	
D8197		Accumulate high 16 bits pulse number	
D8198	PULSE_5	Accumulate low 16 bits pulse number	
D8199		Accumulate high 16 bits pulse number	
……			
D8236	PULSE_24	Accumulate low 16 bits pulse number	
D8237		Accumulate high 16 bits pulse number	

Any problems please contact with:
XINJE Electronic Co.,Ltd.
Website: www.xinje.com
E-mail: sale@xinje.com
MSN: loving-fiona@hotmail.com