Programmable Controllers

series user manual

Preface

Thank you for purchasing programmable controllers, this manual is the basic instructions of the programming manual, and the programming of the note matters, please fully understand the content of the correct use of this product.

Explanation

- Only operators with certain electrical knowledge can perform other operations such as wiring the product. If there is any unclear usage, please consult our company's technicians.
- When using this product, please confirm whether it meets the requirements and safety. If this product malfunctions and may cause machine failure or loss, please set up backup and safety functions by yourself.
- The contents described in the manual are subject to specification changes without notice.

safety matters

- When using under the following conditions and environments, please consult our technical staff and confirm the specifications. At the same time, you must leave room for rated functions and other use and take safety insurance measures into consideration. Control the security measures to a minimum.
- When used outdoors, where there is potential chemical pollution, electrical radiation, and conditions and environments that are not recorded in product samples or instructions.
- Used in nuclear energy control, railways, aviation, vehicle equipment, combustion equipment, medical equipment, safety machinery, administrative agencies and special industries, etc.
- Systems, machinery, devices, etc. that are expected to have a great impact on people and property.
- Used for high-reliability equipment such as gas, water pipes, electricity supply systems and 24-hour non-stop operation systems.

Responsibility statement

- Corresponding to the content of this manual, although carefully edited and checked, if you have any questions or find errors, please contact our company.
- The examples listed in the manual and other technical materials are for user understanding and reference only, and actions are not guaranteed.
- Due to changes in specifications and products, the content described in the document is
- for reference only, and the actual product shall prevail. Our company reserves the right of final interpretation.

Directory

1 PLC introduction

1.1 PLC host configuration

According to different hardware structure functions, PLC is divided into standard, customized, bus,compact PLC and expansion modules.
The main unit of the programmable controller is the basic unit; in order to expand its input and output points, expansion modules are provided; in addition, special expansion modules for special control can also be connected to meet a variety of applications.

1.2 Expansion module composition

PLC provides digital modules, analog modules, temperature modules, weighing modules, functional modules, etc. The expansion module can only be connected to any host of our company. In order to ensure the correct installation and operation of this product, please read the relevant manual carefully before use. A host can match up to 16 extensions.

2 Device device function

2.1 Introduction of soft components

A device with a certain function inside a PLC, and a soft element is a device with a certain function inside a PLC. These devices are composed of electronic circuits, registers, and memory units. In the ladder diagram, such as buttons, switches, relays (Relay), timers (Timer) and counters (Counter) and so on.
The basic internal devices of PLC are as follows:

Device name	Description
Input relay (X)	● The input relay is the basic unit of internal memory storage corresponding to PLC and external input points (terminals used to connect with external input switches and receive external input signals). It is driven by an input signal sent from the outside, making it 0 or 1. The state of the input relay cannot be changed by the method of programming, that is, the basic unit corresponding to the input relay cannot be rewritten. The input relay corresponding to no input signal can only be left empty and cannot be used for other purposes. ● Device representation: X0, X1,...X7, X10, X11..., the device symbol is represented by X, and the sequence is numbered in octal. ● The address number of the I/0 expansion module: It is the same as the number of the host, and it is numbered in octal based on the last point of the host.
Output relay (Y)	● The output relay is the basic unit of internal memory storage corresponding to PLC and external output points (used to connect with external loads). It can be driven by input relay contacts, other internal device contacts and its own contacts. It uses a normally open contact to connect to an external load, and other contacts, like input contacts, can be used multiple times without limitation. There is no output relay corresponding to the output, it is empty, if necessary, it can be used as an internal relay. ● Device representation: Y0, Y1,...Y7, Y10, Y11..., the device symbol is represented by Y, and the sequence is numbered in octal. ● The address number of the I/0 expansion module: It is the same as the number of the host, and it is numbered in octal based on the last point of the host.

Auxiliary relay (M)	● Auxiliary relay is a kind of auxiliary relay inside the PLC. Its function is the same as the auxiliary (middle) relay in the electrical control circuit. Each auxiliary relay also corresponds to a basic unit of the memory. It can be input relay contacts, output relay contacts and other internal The device's contact is driven, and its own contact can also be used for unlimited times. The auxiliary relay cannot directly drive the external output, and it needs to pass the output point to output. ● Device representation: M0, M1,...M7, M8..., the device symbol is represented by M, and the order is numbered in decimal.
Status relay (S)	● The status relay is a stepping action control program input method, and the control program can be written by using the instruction STL to control the transfer of the status relay S. If the step program is not used at all in the program, the status relay S can also be used as an auxiliary relay M, or as an alarm point for external fault diagnosis. ● Device representation: S0, S1,...S1023, the device symbol is represented by S, and the order is numbered in decimal.
Timer (T)	● The timer is used to complete the timing control. The timer contains coils, contacts and timing value registers. When the coil is energized and the predetermined time is reached, its contacts will act. The timer's timing value is given by the set value. Each timer has a specified clock cycle (timing unit: 1ms/10ms/100ms). ● Device representation: T0, T1,...T255, the device symbol is represented by T, and the order is numbered in decimal. Different number ranges correspond to different clock cycles.
Counter (C)	● The counter is used to realize counting operation. To use the counter, the set value of counting (that is, the number of pulses to be counted) should be given in advance. The counter contains coils, contacts and counting memory. When the coil turns from Off→On, it is regarded as the counter has a pulse input, and its count value is increased by one. There are 16-bit and 32-bit and high-speed counters for users to choose. ● Device representation: C0, C1,...C255, the device symbol is represented by C, and the order is numbered in decimal.
Data register (D)	● When PLC performs various sequence control and timing value and count value related control, it often needs to do data processing and numerical calculation, and the data register is specially used to store data or various parameters. Each data register has a 16-bit binary value, that is, a word is stored, and two data registers with adjacent numbers are used to process double words. ● Device representation: D0, D1,...D11999, the device symbol is represented by C, and the sequence is numbered in decimal.
Index register (E、F)	● E, F, and general data registers are 16-bit data registers, which can be written and read freely, and can be used for word devices, bit devices and constants for indirect addressing. ● Device representation: E0~E7, F0~F7, the device symbol is represented by E, F, and the order is numbered in decimal.
Constant (K、H)	● K represents a decimal integer value, and H represents a hexadecimal value. They are used as the set value and current value of timers and counters, or the operands of application instructions.

2.2 Numerical value, constant (K, H)

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

There are 5 types of numerical values that can be used for PLC numerical values to perform calculation tasks. The tasks and functions of various numerical values are described below.

1. Binary
   The numerical calculation or storage in the PLC adopts binary system. The binary value and related terms are as follows:

Types of	Description
Bit	Bit is the most basic unit of binary value, and its state is either 1 or 0
Nibble	It is composed of 4 consecutive digits (such as b3～b0) which can be used to represent a decimal number 0～9 or hexadecimal 0～F
Byte	It is composed of two consecutive nibbles (that is, 8 bits, b7～b0), which can represent 00～FF in hexadecimal
word	It is composed of two consecutive bytes (that is, 16 bits, b15 ~ b0), which can represent the hexadecimal 4-digit value 0000 ~ FFFF
Double word	It is composed of two consecutive words (that is, 32 bits, b31 ~ b0), which can represent 8-bit hexadecimal values 00000000 ~ FFFFFFFF

2. Octal
   The PLC's external input and output terminal numbers adopt octal coding
   Example: External input: X0～X7, X10～X17...(device number)
   External output: Y0～Y7, Y10～Y17...(device number)

3. Decimal
   As the setting value of timer T, counter C, etc., for example: TMR C0 K50. (K constant)
   S, M, T, C, D, E, F, P, I and other device numbers, for example: M10, T30. (Device number)
   Used as an operand in application instructions, for example: MOV K123 D0. (K constant)

4. BCD
   A decimal data is represented by half a byte or 4 digits, so consecutive 16 digits can represent 4-digit decimal numerical data.

5. Hexadecimal
   Used as an operand in application instructions, for example: MOV H1A2B D0. (H constant)

Constant K:

In the PLC system, the decimal value is usually represented by the word "K" in front of the value.
Example: K100, expressed as a decimal system, and its value is 100.

- When K is used with bit devices X, Y, M, and S, it can be combined into data in the form of nibble, byte, word or double word.

Example: K2Y10, K4M100. Here K1 represents a combination of 4 bits, and K2~K4 represent combinations of 8, 12 and 16 bits respectively.

Constant H:

The hexadecimal value in the PLC is usually represented by the "H" character in front of the value, for example: H100, which is expressed in hexadecimal and the value is 100.

The numerical comparison table is as follows:

Binary		Octal	Decimal	BCD		Hexadecimal
For PLC internal calculation		Device X, Y number	Constant K, device M, S, T, C, D, E, F, P, I number	For DIP switch and 7-segment display		Constant H
0 0 0 0	0 0 0 0	0	0	0 0 0 0	0 0 0 0	0
0 0 0 0	0 0 0 1	1	1	0 0 0 0	0 0 0 1	1
0 0 0 0	0 0 1 0	2	2	0 0 0 0	0 0 1 0	2
0 0 0 0	0 0 1 1	3	3	0 0 0 0	0 0 1 1	3
0 0 0 0	0 1 0 0	4	4	0 0 0 0	0 1 0 0	4
0 0 0 0	0 1 0 1	5	5	0 0 0 0	0 1 0 1	5
0 0 0 0	0 1 1 0	6	6	0 0 0 0	0 1 1 0	6
0 0 0 0	0 1 1 1	7	7	0 0 0 0	0 1 1 1	7
0 0 0 0	1 0 0 0	10	8	0 0 0 0	1 0 0 0	8
0 0 0 0	1 0 0 1	11	9	0 0 0 0	1 0 0 1	9
0 0 0 0	1 0 1 0	12	10	0 0 0 1	0 0 0 0	A
0 0 0 0	1 0 1 1	13	11	0 0 0 1	0 0 0 1	B
0 0 0 0	1 1 0 0	14	12	0 0 0 1	0 0 1 0	C
0 0 0 0	1 1 0 1	15	13	0 0 0 1	0 0 1 1	D
0 0 0 0	1 1 1 0	16	14	0 0 0 1	0 1 0 0	E
0 0 0 0	1 1 1 1	17	15	0 0 0 1	0 1 0 1	F
0 0 0 1	0 0 0 0	20	16	0 0 0 1	0 1 1 0	10
0 0 0 1	0 0 0 1	21	17	0 0 0 1	0 1 1 1	11
: : :	: : :	: : :	: : :	: : :	: : :	: : :
0 1 1 0	0 0 1 1	143	99	1 0 0 1	1 0 0 1	63

2.3 Input and output relay (X, Y)

Input and output relays are all numbered in octal

Host number: The number of input and output terminals is fixed from X0 and Y0, and the number of numbers varies with the number of points of the host.

I/O expansion: The number of input and output terminals is calculated according to the connection sequence of the host.

name	range								
	14 point	16 point	24 point	32 point	40 point	48 point	60 point	68 point	Expansion I/0 (Note 1)
Input (X)	X0~X7 (8 point)	X0~X7 (8 point)	X0~X13 (8 point)	X0~X17 (16 point)	X0~X27 (24 point)	X0~X27 (24 point)	X0~X43 (36 point)	X0~X43 (36 point)	X※～X377
Output (Y)	Y0~Y5 (6 point)	Y0~Y7 (8 point)	Y0~Y13 (8 point)	Y0~Y17 (16 point)	Y0~Y17 (16 point)	Y0~Y27 (24 point)	Y0~Y27 (24 point)	Y0~Y37 (32 point)	Y※~Y377

Description:

Note 1: Expansion I/O input and output starting number starts with the last number of connecting host input/output points. The numbers of the extended I/O are arranged in sequential order. If the last point of the host is X n□ (the number range in □ is 0-7), the start number of the digital extended input is X (n+1)0 . The same is true for the extended output start number. The maximum input number can reach X377, and the maximum output number can reach Y377.

Example: The last point of the host is X27, and the start number of the extended input is X30. The last point of the host is X43, and the start number of the extended input is X50.

1. Input relay: X0～X377

The number of the input relay (or input terminal) is coded in octal, the maximum number of points can reach 256 points, and the range is as follows: X0～X7, X10～X17,..., X370～X377.

Function of input contact X:

The input contact X is connected with the input device, and the input signal is read into the PLC. There is no limit to the number of times the A or B contact of each input contact X can be used in the program. The On/Off of the input contact X will only change with the On/Off of the input device. You cannot use the programming software to force the On/Off of the input contact X.

2. Output relay: Y0～Y377

The number of the output relay (or output terminal) is coded in octal, the maximum number of points can reach 256 points, and the range is as follows: Y0～Y7, Y10～Y17,..., Y370～Y377.

Function of output contact Y:

The output contact Y sends out On/Off signals to drive the load connected to the output contact Y. There are two types of output contacts, one is a relay and the other is a transistor. There is no limit to the number of times that the A or B contact of each output contact Y can be used in the program, but the number of the output coil Y is only recommended in the program. It can be used once, otherwise, according to the PLC's program scanning principle, the power to determine the output state will fall on the last output Y circuit in the program.

- Input processing

1. The PLC will read the On/Off status of the external input signal into the input image area once before executing the program.
2. If the input signal changes on/off during program execution, the state in the input image area will not change, and the new On/Off state of the input signal will be read until the next scan starts.
3. There is a delay of about 10ms from the time the external signal On→Off or Off→On changes to the time when the contact in the program is recognized as On/Off (but it may be affected by the program scan cycle).

- Program processing

After the PLC reads the On/Off status of each input signal in the input image area, it starts to execute each instruction in the program sequentially from address 0, and the processing result, namely the On/Off of each output coil, is also successively stored in each device image area. Inside.

- Output processing

1. When the END instruction is executed, the On/Off status of Y in the device image area is sent to the output image area for latch, and this image area is actually the coil of the output relay.
2. There is about 10ms delay between the relay coil On→Off or Off→On changing to the contact On/Off.
3. Using a transistor module, there will be a delay of about 10~20us from the On→Off or Off→On change to the contact On/Off.

2.4 Auxiliary relay (M)

All auxiliary relays are numbered in decimal system, please refer to the corresponding table for the serial number of each series:

Auxiliary relay (M)	General use	M0~M499, 500 points. Can use parameter settings to change to the power failure retention area	4096 points in total
	For power failure	M500~M999, M2000~M4095, 2,596 points. Can use parameter settings to change to non-latched area	
	Special use	M1000~M1999, 1,000 points. Part of it is maintained	

Function of auxiliary relay:

Auxiliary relay M and output relay Y have output coils and A, B contacts, and there is no limit to the number of times they can be used in the program. Users can use auxiliary relay M to combine control loops, but they cannot directly drive external loads. According to its nature, it can be divided into the following three types:

1. General auxiliary relay: If the general auxiliary relay encounters a power failure when the PLC is running, its status will all be reset to Off, and its status will remain Off when it is re-powered.
2. Auxiliary relay for power failure retention: If the auxiliary relay for power failure retention encounters a power failure when the PLC is running, its state will all be maintained, and its state will be the state before the power failure when the power is turned on again.
3. Special auxiliary relay: each special auxiliary relay has its specific function, please don't use the undefined special auxiliary relay. Special auxiliary relays cannot be used as ordinary relay M.

2.5 Status relay (S)

The status relays are all numbered in decimal system, please refer to the corresponding table for the serial number of each series:

Status relay (S)	Initial	S0 ~ S9, 10 points. Can be modified to be latched by setting up parameters.	Total 1024 points
	Zero return	S10 ~ S19, 10 points, used with IST instruction. Can be modified to be latched by setting up parameters.	
	General purpose	S20 ~ S499, 480 points. Can be modified to be latched by setting up parameters.	
	Latched	S500 ~ S899, 400 points. Can be modified to be non-latched by setting up parameters.	
	Alarm	S900 ~ S1023, 124 points. Can be modified to be latched by setting up parameters.	

Function of status relay:

The state relay S can be easily set up in the engineering automation control program. It is the most basic device of the step ladder diagram. STL, RET, etc. must be included in the step ladder diagram (or Sequential Function Chart, SFC) Use with instructions.

The device number of stepping relay S is S0～S1023 with 1,024 points. Each stepping relay S and output relay Y have output coils and A, B contacts, and there is no limit to the number of times they can be used in the program, but they cannot directly drive external loads. When stepping relay (S) is not used for stepping ladder diagram, it can be used as a general auxiliary relay. Its nature can be divided into the following four types:

Initial step relay	S0~S9, a total of 10 points. The step point used as the initial state in the Sequential Function Chart (SFC).
Zero return step relay:	S10~S19, 10 points. When the ZL 60 IST instruction is used in the program, S10~S19 are planned for home return. If the IST instruction is not used, it will be used as a general stepping relay.
General purpose step relay	S20~S499, 480 points. In the sequence function chart (SFC) as a general purpose step point, if there is a power failure when the PLC is running, its status will be cleared.
Latched step relay	S20～S127, 108 points. In the sequence function diagram (SFC), if the stepping relay for power failure retention encounters a power failure when the PLC is running, its state will all be maintained, and its state will be the state before the power failure when the power is retransmitted.
Alarm step relay	S900~S1023, 124 points. The step relay for alarm and the alarm point drive command ZL 46 ANS are used as alarm contacts to record relevant warning information and to eliminate external faults.

2.6 Timer (T)

The timers are all numbered in decimal, please refer to the corresponding table for the serial number:

Timer T	100ms general purpose	T0 ~ T199, 200 points. When M1028 is OFF, T64 to T126 is 100ms When M1028 is ON, T64 to T126 is 10ms	Total 256 points
	10ms general purpose	T200~T245，T250~T255, 40 points. When M1038 is OFF, For T200 to T245 and T250 to T255 is 10ms When M1038 is ON, For T200 to T245 and T250 to T255 is 1ms	
	1ms accumulative	T246 ~ T249, 4 points.	

Timer function:

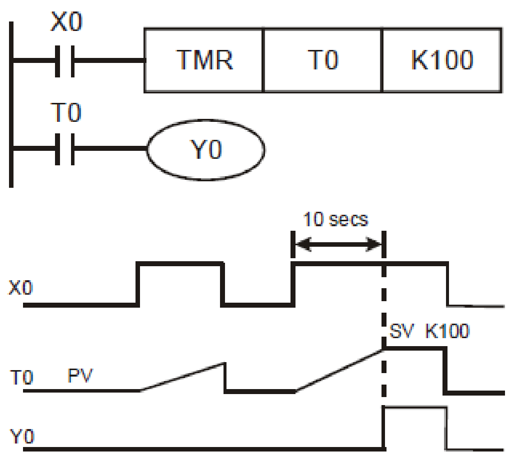
The timer uses 1ms, 10ms, and 100ms as a timing unit. The timing method adopts counting up. When the current value of the timer = the set value, the output coil is turned on. The set value is a decimal K value. Data register D can also be used As a set value.

The actual setting time of the timer = timer unit * setting value.

According to its nature, it can be divided into the following three types:

1. General purpose timer:

The timer is generally used to time one time when the TMR instruction is executed. When the TMR instruction is executed, if the timing reaches, the output coil is turned on.

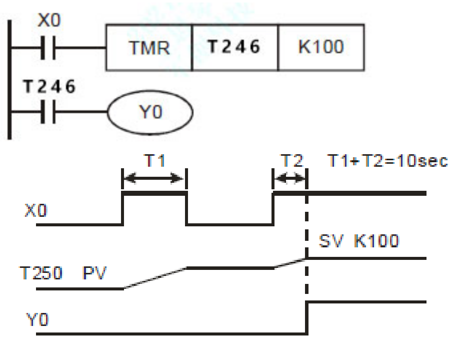


1、 When X0 = On, The PV in timer T0 will count up by 100ms. When the PV = SV K100, the output coil T0 will be On.

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

2、Accumulative type timer:

The timer executes once when the program reaches TMR instruction. When TMR instruction is executed, the output coil will be On when the timing reaches its target.



1、 When X0 = On, The PV in timer T246 will count up by 100ms. When the PV = SV K100, the output coil T246 will be On.

2、 When X0 = Off or the power is off, timer T246 will temporarily stop the timing and the PV remain unchanged. When X0 is On again, the timing will resume and the PV will count up and when the PV = SV K100, the output coil T246 will be On.

2.7 Counter (C)

All counters are numbered in decimal system, please refer to the corresponding table for the serial number of each series:

Counter C	16-bit counting up, for general purpose	C0 ~ C99, 100 points.	Total 253 points
	16-bit counting up, for latched	C100 ~ C199, 100 points.	
	32-bit counting up/down, for general purpose	C200 ~ C234, 20 points.	
32-bit counting up/down high-speed counter C	software 1-phase 1 input	C235~C238, 4 points	
	Hardware 1-phase 2 inputs	C241~C243, 3 points	
	Hardware 2-phase 2 inputs	C251 ~ C254, 4 points	

1、 Features of counter:

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

2、 Functions of counters

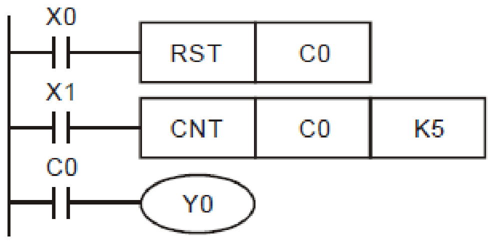
When the pulse input signals of the counter go from Off to On and the present value in the counter equals the set value, the output coil will be On. The set value should be a K value in decimal and the data register D can also be a set value.

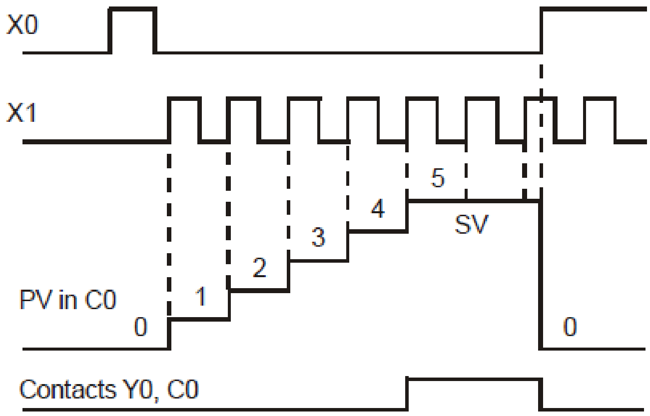
16-bit counters C0 ~ C199:

- The setup range of 16-bit counter: K0 ~ K32,767. K0 is the same as K1. The output contact will be On immediately when the first counting starts.
- PV in the general purpose counter will be cleared when the power of the PLC is switched off. If the counter is a latched type, the counter will retain the PV and contact status before the power is off and resume the counting after the power is on again.
- If you use MOV instruction, send a value bigger than the SV to the present value register of C0, next time when X1 goes from Off to On, the contact of counter C0 will be On and its PV will equal SV.
- The SV in the counter can be constant K (set up directly) or the values in register D (set up indirectly, excluding special data registers D1000~ D1999).
- If you set up a constant K as the SV, it should be a positive value. Data register D as SV can be positive or negative. When the PV reaches up to 32,767, the next PV will turn to -32,768.

  Example:

```
LD      X0
RST     C0
LD      X1
CNT     C0   K5
LD      C0
OUT     Y0
```

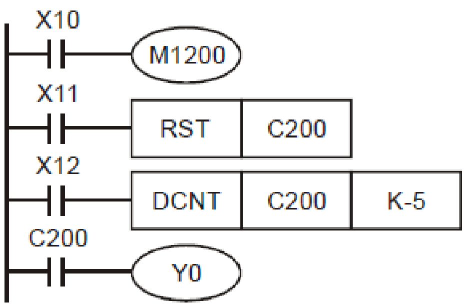




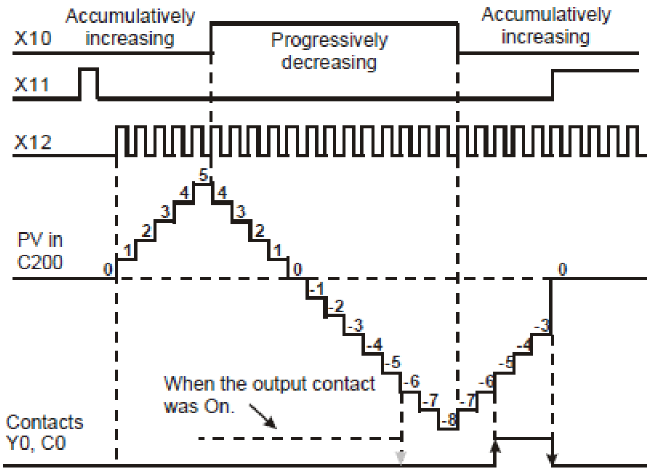
a) When X0 = On, RST instruction will be executed, PV in C0 will be "0" and the output contact will be reset to Off.
b) When X1 goes from Off to On, the PV in the counter will count up (plus 1).
c) When the counting of C0 reaches SV K5, the contact of C0 will be On and PV of C0 = SV = K5. The X1 trigger signal comes afterwards will not be accepted by C0 and the PV of C0 will stay at K5.

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

- The setup range of 32-bit counter: K-2,147,483,648 ~ K2,147,483,647.
- Addition or subtraction of the counters is designated by On/Off status of special auxiliary relays M1200 ~ M1234. For example, when M1200 = Off, C200 will be an addition counter; when M1200 = On, C200 will be a subtraction counter.
- The SV can be constant K or data register D (excluding special data registers D1000 ~ D1999). Data register D as SV can be a positive or negative value and an SV will occupy two consecutive data registers.
- PV in the general purpose counter will be cleared when the power of the PLC is switched off. If the counter is a latched type, the counter will retain the PV and contact status before the power is off and resume the counting after the power is on again.
- When the PV reaches up to 2,147,483,647, the next PV will turn to -2,147,483,648. When the PV reaches down to -2,147,483,648, the next PV will turn to 2,147,483,647.



1. X10 drives M1200 to determine whether C200 is an addition or subtraction counter.
2. When X11 goes from Off to On, RST instruction will be executed and the PV in C200 will be cleared to "0" and the contact will be Off.
3. When X12 goes from Off to On, the PV in the counter will count up (plus 1) or count down (minus 1).



4. When the PV in C200 changes from K-6 to K-5,the contact of C200 will go from Off to On. When the PV in C200 changes from K-5 to K-6, the contact of C200 will go from On to Off.
5. If you use MOV instruction, HPP to send a value bigger than the SV to the present value register of C0, next time when X1 goes from Off to On, the contact of counter C0 will be On and its PV will equal SV.

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

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

PLC models support high-speed counters. C235~C240 are program interrupted one-phase high-speed counters, with counting frequencies up to 10KHz. C241~C254 are hardware high speed counters (Hardware High Speed Counter hereinafter referred to as HHSC). There are four HHSCs (HHSC0~3). The pulse input frequency of HHSC0~3 and HHSC1 can reach 200 kHz. among them:

Numbering	Abbreviation
C251	HHSC0
C252	HHSC1
C253	HHSC2
C254	HHSC3

- Each HHSC can only be assigned to one number at a time. Use the DCNT command as the assignment.
- Each HHSC has three counting modes:
  (1) 1 phase 1 input, also known as pulse/direction (Pulse/Direction) mode
  (2) 1 phase 2 input, also called forward/reverse (FWD/REV) mode
  (3) 2 phase 2 input, also known as AB-phase (AB-phase) mode
  (4) Please refer to the table below for distinguishing by serial number.

Type / Input	software high-speed counter						Hardware high-speed counter							
	1 phase 1 input						1 phase 1 input				2 phase 2 input			
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C251	C252	C253	C254
X0	U/D						U/D				A			
X1		U/D									B			
X2			U/D					U/D				A		
X3				U/D								B		
X4					U/D				U/D				A	
X5						U/D							B	
X6										U/D				A
X7														B

Description:

U	Count up
D	Count down
A	Phase A input
B	Phase B input

3、 Counting modes

The counting modes of the hardware high-speed counters in CPU can be set in D1225 ~ D1228:

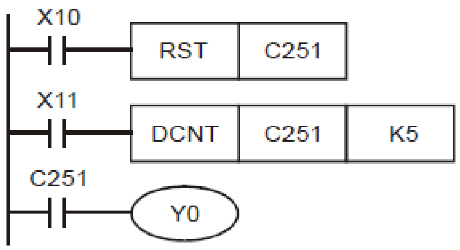
Counting modes		Wave pattern	
Type	Set value in special D	Counting up(+1)	Counting down(-1)
1-phase 1 input	1 (Normal frequency)		
	2 (Double frequency)		
1-phase 2 inputs	1 (Normal frequency)		
	2 (Double frequency)		

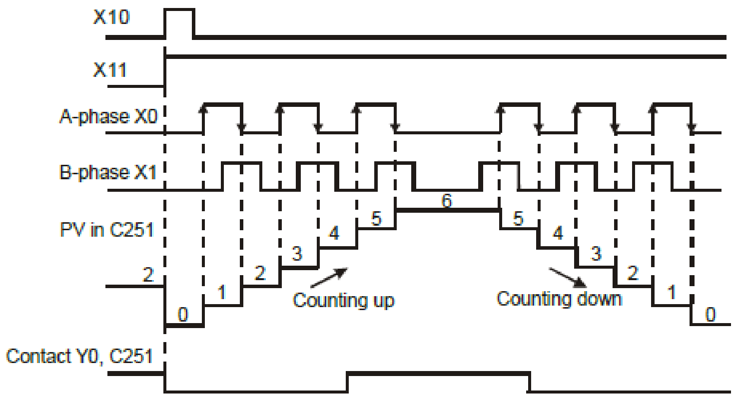
Counting modes		Wave pattern	
Type	Set value in special D	Counting up(+1)	Counting down(-1)
2-phase 2 inputs	1 (Normal frequency)		
	2 (Double frequency)		
	3 (Triple frequency)		
	4 (4 times frequency)		

4、High-speed counter related flag signals and special registers for related settings:

Flag	Function
M1235 ~ M1244	C235 ~ C244 High speed counter counting direction specified. When M12□□=Off ， C2□□： Count on。 When M12□□=On ， C2□□:Count off。
D1225	The counting mode of the 1st group counters (C251)
D1226	The counting mode of the 2nd group counters (C252)
D1227	The counting mode of the 3rd group counters (C253)
D1228	The counting mode of the 4th group counters (C254)
D1225~D1228	PLC hardware high speed counter HHSC0~ HHSC3 counting mode setting, not the following setting values are preset for the double frequency counting mode. 2： for the double frequency counting mode, (factory value). 3： it is the triple frequency counting mode. 4： it is the quadruple frequency counting mode.（desired value）

2-phase AB input high-speed counter:





1、 When X10 is On, RST instruction will be executed. The PV in C251 will be cleared to "0" and the output contact will be reset to be Off.

2、 In C251, when X11 is On and C251 receives the A-phase signals from X0 and B-phase signals from X1, the PV in the counter will count up (plus 1) or count down (minus 1). You can select different counting modes .

3、 When the counting of C251 reaches SV K5, the contact of C251 will be On. If there are still input signals coming in, the counting will continue.

4、 The counting modes (normal frequency, double frequency, triple frequency or 4 times frequency) of C251 (HHSC0) can be set up by D1225.

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

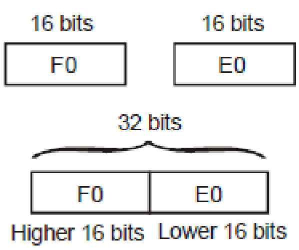
A data register is for storing a 16-bit datum of values between -32,768 to +32,767. The highest bit is "+" or "-" sign. Two 16-bit registers can be combined into a 32-bit register (D + 1; D of smaller No. is for lower 16 bits). The highest b it is "+" or "-" sign and it can store a 32-bit datum of values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

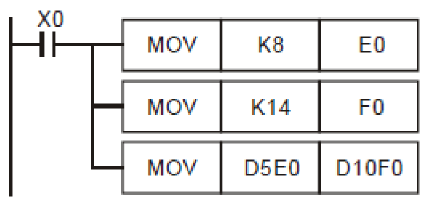
Data register D	General purpose	JS/T 24 points and below points host	D0 ~ D499, 500 points.Fixed as general purpose. Cannot be modified to be latched by setting up parameters.
		JS 32 points and above points host JM host	D0 ~ D199, 200 points. Can be modified to be latched by setting up parameters.
		JH2 series host	D0 ~ D199， D12000~D30000, 28200 points.
	Latched	JS/T 24 points and below points host	D500 ~ D499, 500 points
		JS 32 points and above points host JM,JH2 host	D200~D999、D2000~D11999，10,800 points.
	Special purpose	D1000~D1999， 1,000 pints. Some are latched.	
	Index register E, F	E0~E7， F0~F7， 16 points.	

1、 Registers can be divided into the following four types according to their nature：

- General purpose register： When the PLC is powered off, the value data in the register will be cleared to 0.
- Latched register： When the power of PLC is switched off, the data in the register will not be cleared but will retain at the value before the power is off. You can use RST or ZRST instruction to clear the data in the latched register.
- Special purpose register： Every register of this kind has its special definition and purpose, mainly for storing the system status, error messages and monitored status.
- Index register E, F： The index register is a 16-bit register, E0~E7, F0~F7 total 16 points.

2、Index Register [E], [F]





Index registers E, F are 16-bit data registers and can be written and read.
If you need to use a 32-bit register, you have to designate E. In this case, F will be covered by E and cannot be used anymore; otherwise, the content in E (32-bit) will be incorrect. We suggest you use DMOVP K0 E instruction, the content in E (including F) will be cleared to "0" when the power of PLC is switched on.
The combination of E, F when you use a 32-bit index register:
(F0, E0), (F1, E1), (F2, E2), …(F7, E7)

- The index register is the same as normal operands, can be used for moving or comparison on word devices (KnX, KnY, KnM, KnS, T, C, D) and bit devices (X, Y, M, S). supports constant (K, H) index register. has 16 points of index registers E0 ~ E7, F0 ~ F7.
- When you use the instruction mode to generate constant (K, H) index register function, please use symbol "@". For example, "MOV K10@E0 D0F0".

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

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

Description：

1、Input point X as a high-speed counter cannot be used as an external interruption signal. For example, if C251 occupies X0, X1, X2 and X3, the external input interruption No. I00□(X0), I10□(X1), I20□(X2), and I30□(X3) cannot be used.

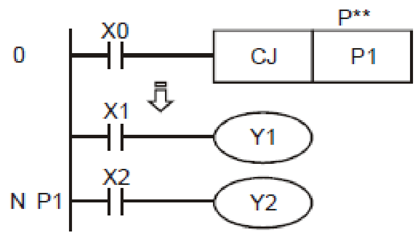
2、If an interrupt subroutine is executed, the next interrupt subroutine will not be executed until the execution of the interrupt is complete.

3、The time it takes for an interrupt subroutine in a PLC to be executed affects the efficiency of the PLC. It is suggested that the size of an interrupt subroutine not be large.

4、Pointer N: Used with MC and MCR instructions. MC is the master control start instruction. When MC instruction is executer, the instructions between MC and MCR will still be executed normally.

5、Pointer P: Used with application commands ZL 00 CJ, ZL 01 CALL, ZL 02 SRET.

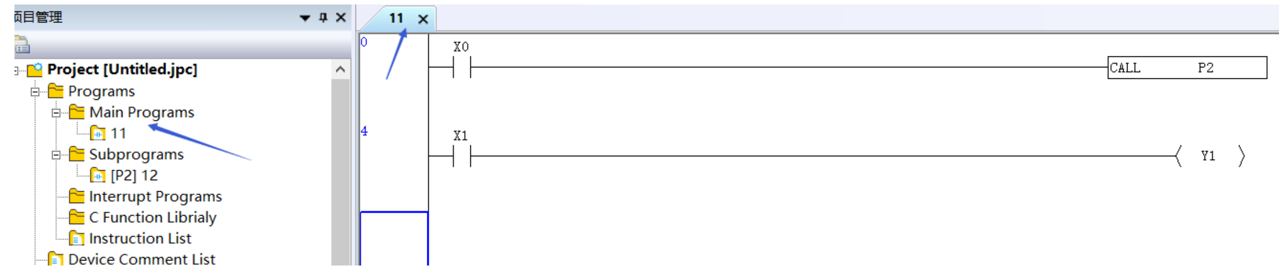
- CJ Conditional Jump:

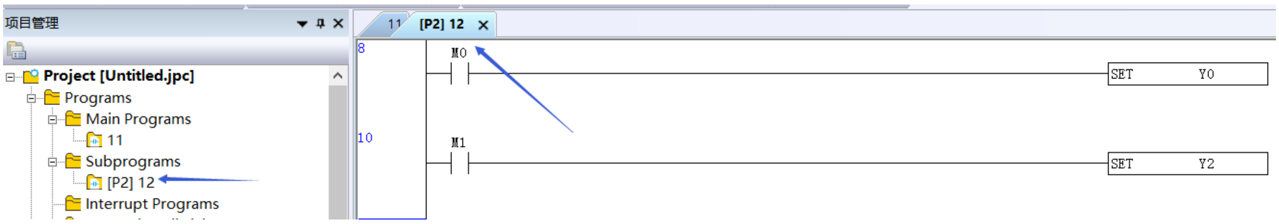


1、When X0 = On, the program will jump from address 0 to N (designated label P1) and keep on the execution. The addresses in the middle will be ignored.

2、When X0 = Off, the program will execute from address 0 and keep on executing. At this time, CJ instruction will not be executed.

- CALL Call Subroutine, SRET Subroutine Return：





instruction list

```
000000    LD     X0
000001    CALL   P2
000004    LD     X1
000005    OUT    Y1
000006    FEND
000007    P2
000008    LD     M0
000009    SET    Y0
000010    LD     M1
000011    SET    Y2
000012    SRET
000013    END
```

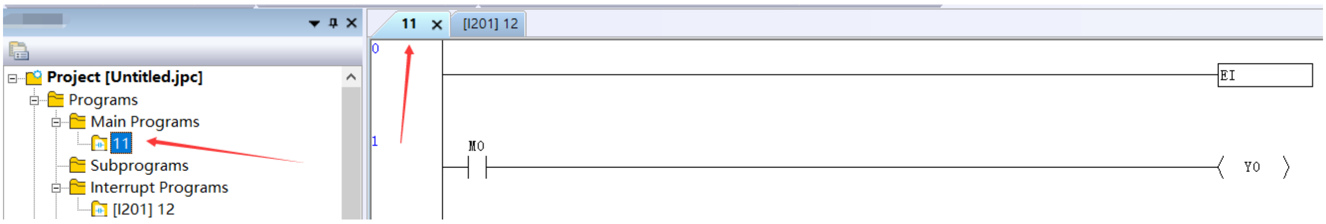
1、When X0 = On, CALL instruction will be executed and the program will jump to P2 and executed the designated subroutine. When SRET instruction is executed, the program will return to address 24 and keep on the execution.

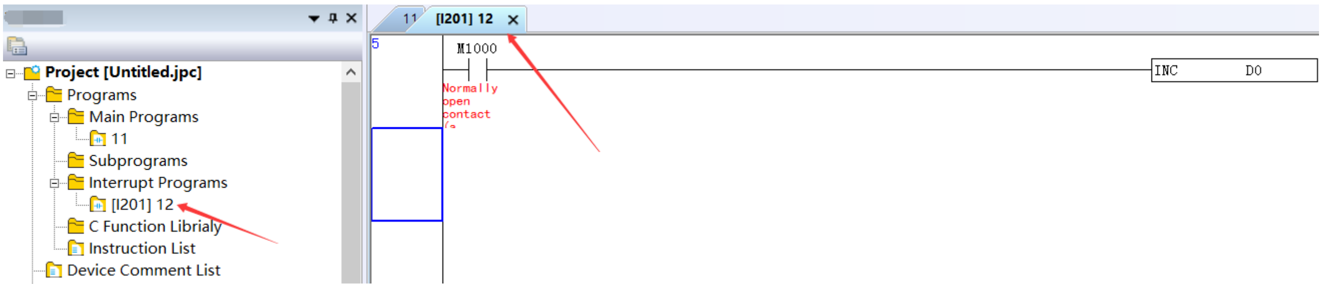
2、There is no need to edit the FEND and SRET codes in the ladder diagram. After the compilation is passed, the instruction list will be automatically generated.

Interruption Pointer I: Used with application commands ZL 04 EI, ZL 05DI, and ZL 03 IRET, the purpose can be divided into the following six types. The interrupt insertion action must be combined with commands such as EI interrupt insertion enable, DI interrupt insertion prohibition, and IRET interrupt insertion return.

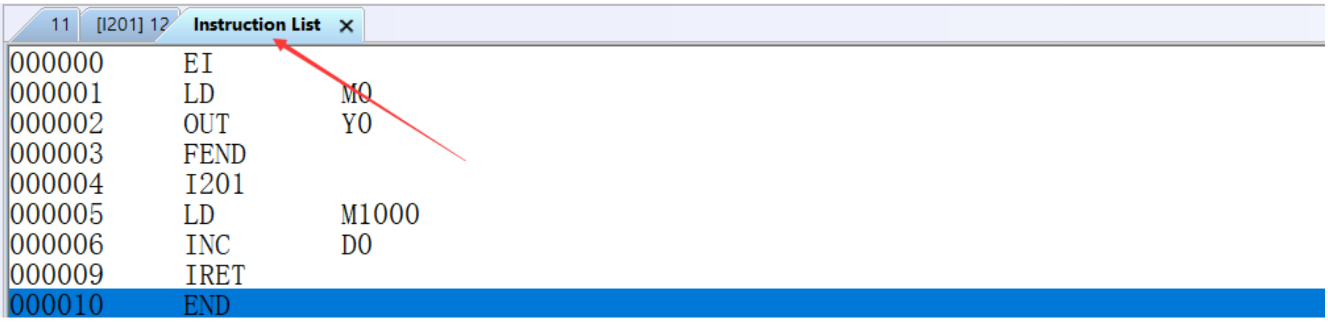
1、External interruption: Due to the special hardware design inside the CPU, the input signals coming in at input terminals X0 ~ X5 when rising-edge or falling-edge triggers will not be affected by the scan cycle. The currently executed program will be interrupted immediately and the execution will jump to the designated interruption subroutine pointer I00□(X0), I10□(X1), I20□(X2), I30□(X3), I40□(X4), I50□(X5). Till the execution reaches IRET instruction, the program will return to the original position and keep on its execution.

Example: X2 rising edge interrupt.





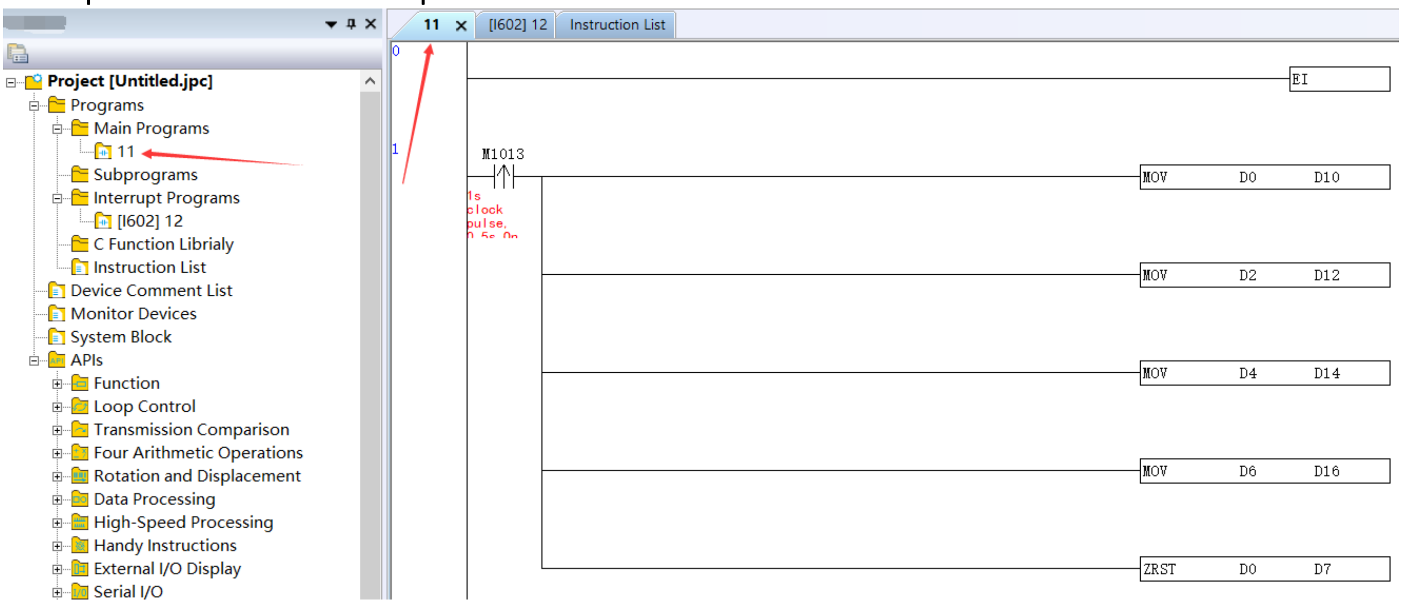
instruction list

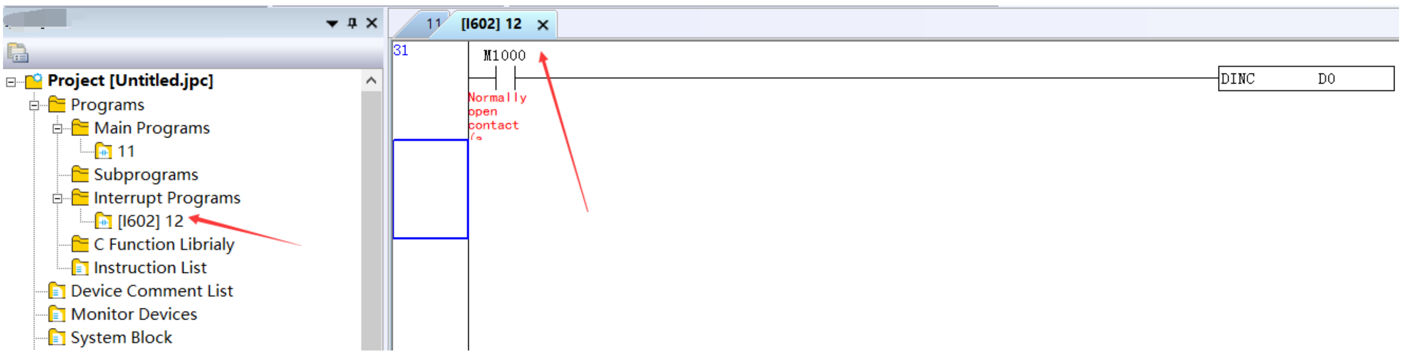



```
000000    EI
000001    LD      M0
000002    OUT     Y0
000003    FEND
000004    I201
000005    LD      M1000
000006    INC     D0
000009    IRET
000010    END
```

2、Timed interruption: PLC automatically interrupts the currently executed program every a fixed period of time and jumps to the execution of a designated interruption subroutine.

Example: I602 timer interrupt.





11 | [I602] 12 | **Instruction List**

```
000000    EI
000001    LDP     M1013
000004    MOV     D0      D10
000009    MOV     D2      D12
000014    MOV     D4      D14
000019    MOV     D6      D16
000024    ZRST    D0      D7
000029    FEND
000030    I602
000031    LD      M1000
000032    DINC    D0
000037    IRET
000038    END
```

3、Interruption when the counting reaches the target: The high-speed counter comparison instruction ZL 53 DHSCS can designates that when the comparison reaches the target, the currently executed program will be interrupted and jump to the designated interruption subroutine executing the interruption pointers I010, I020, I030, I040, I050 and I060.

4、Pulse interruption: The pulse output instruction ZL 57 PLSY can be set up that the interruption signal is sent out synchronously when the first pulse is sent out by enabling flags M1342 and M1343. The corresponding interruptions are I130 and I140. You can also set up that the interruption signal is sent out after the last pulse is sent out by enabling flags M1340 and M1341. The corresponding interruptions are I110 and I120.

2. 10 Functions of Special Auxiliary Relays and Special Registers

The types and functions of special auxiliary relay (special M) and special data register (special D) are as follows. In the following tables, there is a "*" mark in the upper right corner of the number. You can refer to the function description in the next section. If the attribute column is marked as "R", it means that it can only be read. If it is marked as "R/ W" means that it can be read and written. If it is marked as "-", it means no change. Marked as "#", it means that the system will be set according to the PLC status, and the user can read the setting value and compare the description of the manual to further understand the system information.

Special M	Function Description
M1000	Normally open contact (a contact). This contact is ON when running and it is ON when the status is set to RUN.
M1001	Normally OFF contact (b contact). This contact is OFF in running and it is OFF when the status is set to RUN.
M1002	ON only for 1 scan after RUN. Initial pulse is contact a. It will get positive pulse in the RUN moment. Pulse width=scan period.
M1003	OFF only for 1 scan after RUN. Initial pulse is contact a. It will get negative pulse in the RUN moment. Pulse width=scan period.
M1004	On when error occurs
M1005	Password of data backup memory card and MPU password don't match
M1006	Data backup memory card isn't initial
M1007	Data doesnˇt exist in the program area of data backup memory card
M1008	Monitor timer flag (ON: PLC WDT time out)
M1009	When 24VDC is not sufficient that causes LV signal, M1009 will be ON
M1011	10ms clock pulse, 5ms On/5ms Off
M1012	100ms clock pulse, 50ms On / 50ms Off
M1013	1s clock pulse, 0.5s On / 0.5s Off
M1014	1min clock pulse, 30s On / 30s Off
M1015	High-speed timer activates
M1016	When it is Off, it will display two right-most bits. When it is On, it will display (two right-most bits + 2000).
M1017	30 seconds adjustment in RTC
M1018	Flag for Radian/Degree, On for degree
M1019	Start-up flag of frequency measurement card
M1020	Borrow flag
M1021	Borrow flag
M1022	Carry flag
M1024	COM1 Monitor request
M1025	If PLC receive illegal communication request when HPP, PC or HMI connects to PLC, M1025 will be set and save the error code in D1025.
M1027	PLSV instruction (Y0, Y1) acceleration and deceleration time enable flag
M1028	10ms time switch flag, when M1028=Off, the time base of T64~T126 is 100ms, when On, the time base is changed to 10ms

M1029	Y0, Y1 pulse sending completion flag
M1030	Y2, Y3 pulse sending completion flag
M1031	Clear all latched memory
M1032	Clear all latched memory
M1033	Memory latched at STOP
M1036	Y4, Y5 pulse sending completion flag
M1037	Y6, Y7 pulse sending completion flag
M1038	Switching T200~T245 ,T250~T255 timer resolution (10ms/1ms). ON = 1ms.
M1039	Constant scan mode
M1040	Step transition starts
M1041	Step transition starts
M1044	Zero point condition
M1045	All outputs clear inhibit
M1046	STL state setting (On)
M1047	STL monitor enable
M1048	Flag for alarm point state
M1049	Monitor flag for alarm point
M1050	I001 masked
M1051	I101 masked
M1052	I201 masked
M1053	I301 masked
M1054	I401 masked
M1055	I501 masked
M1056	I6 masked
M1057	I7 masked
M1059	I010~I060 masked
M1060	System error message 1
M1061	System error message 2
M1062	System error message 3
M1063	System error message 4
M1064	Operator error
M1065	Syntax error
M1066	Program error
M1067	Program execution error
M1068	Execution error locked (D1068)
M1070	PWM command Y0 output frequency unit switching, 100us when On, 1ms when Off
M1071	PWM command Y0 output frequency unit switching, 100us when On, 1ms when Off
M1076	Battery voltage is too low or malfunction
M1077	Battery voltage is too low or malfunction
M1080	COM2 Monitor request
M1081	FLT command change direction flag
M1082	Flag changed for RTC
M1102	Y10, Y11 pulse sending completion flag
M1103	Y12, Y13 pulse sending completion flag

M1104	Y14, Y15 pulse sending completion flag
M1105	Y16, Y17 pulse sending completion flag
M1106	Y20, Y21 pulse sending completion flag
M1107	Y22, Y23 pulse sending completion flag
M1108	Y24, Y25 pulse sending completion flag
M1109	Y26, Y27 pulse sending completion flag
M1110	Y30, Y31 pulse sending completion flag
M1111	Y32, Y33 pulse sending completion flag
M1112	Y34, Y35 pulse sending completion flag
M1113	Y36, Y37 pulse sending completion flag
M1114	Y40, Y41 pulse sending completion flag
M1115	Y42, Y43 pulse sending completion flag
M1116	Y44, Y45 pulse sending completion flag
M1117	Y46, Y47 pulse sending completion flag
M1118	Y50, Y51 pulse sending completion flag
M1119	Y52, Y53 pulse sending completion flag
M1120	COM2(RS-485) Communication protocol holding (communication protocol will be the original setting even if D1120 is changed)
M1121	RS-485 Communication data transmission ready
M1122	Sending request
M1123	Receiving completed
M1124	Receiving wait
M1125	Communication reset
M1127	MODRD/RDST/MODRW instructions. Data receiving is completed, RS instruction doesn include.
M1128	Sending / Receiving
M1129	Receiving time out
M1131	MODRD/RDST/MODRW, M1131=On when data convert to HEX
M1132	When it is On, it denotes that there is no relative communication instruction in PLC program.
M1138	COM1 (RS-232) communication protocol holding. D1036 modification invalid after setting.
M1139	When SLAVE mode, ASCII/RTU selection for COM1 (RS-232). OFF for ASCII mode and ON for RTU mode.
M1140	MODRD/MODWR/MODRW data received error
M1141	MODRD/MODWR/MODRW command error
M1142	VFD-A command data received error
M1143	1.COM2(RS-485) ASCII/RTU mode selections when PLC is SLAVE (it is Off when in ASCII mode and it is On when in RTU) 2.COM2(RS-485) ASCII/RTU mode selections when PLC is MASTER (used with MODRD/ MODWR/MODRW instructions) (it is Off when in ASCII mode and it is On when in RTU mode)
M1144	Ouput start switch of accel/decel pulse output function of adjustable slope
M1145	Acceleration flag of accel/decel pulse output function of adjustable slope
M1146	Target attained frequency flag of accel/decel pulse output function of adjustable slope
M1147	Deceleration flag of accel/decel pulse output function of adjustable slope
M1148	Complete function flag of accel/decel pulse output function of adjustable slope
M1149	Stop counting temporality flag of accel/decel pulse output function of adjustable slope

M1150	Declare DHSZ command used for multi-group settings comparison mode
M1151	Finish executing multi-group settings comparison mode
M1152	Declare DHSZ command used to be frequency control mode
M1153	Finish executing frequency control mode
M1154	Start designated deceleration function flag of accel/decel pulse output function of adjustable slope
M1161	8/16 bits mode (it is On when in 8 bits mode)
M1162	Using flag for the integral of decimal system and the floating point of binary systems. ON for the floating point of binary.
M1167	HKY input is 16 bits mode
M1168	SMOV working mode indication
M1169	PWD mode selection
M1170	Start executing single step
M1171	Execute single step
M1172	2-phase pulse output switch (on is start)
M1173	On is continuous output switch
M1174	Output pulse number attained flag
M1178	VR0 potentiometer starts
M1179	VR1 potentiometer starts
M1184	Startup MODEM
M1185	Start to initiate MODEM
M1186	Fail to initiate MODEM
M1187	Finish initiating MODEM
M1188	Display if current MODEM is on-line or not
M1200	C200 counting mode (on: count down
M1201	C202 counting mode (on: count down)
M1202	C202 counting mode (on: count down)
M1203	C203 counting mode (on: count down)
M1204	C204 counting mode (on: count down)
M1205	Y54, Y55 pulse sending completion flag
M1206	Y52, Y53 pulse sending completion flag
M1207	PLSV instruction acceleration and deceleration time enable
M1226	C226 counting mode (on: count down)
M1227	C227 counting mode (on: count down)
M1228	C228 counting mode (on: count down)
M1229	C229 counting mode (on: count down)
M1230	C230 counting mode (on: count down)
M1231	C231 counting mode (on: count down)
M1232	C232 counting mode (on: count down)
M1233	C233 counting mode (on: count down)
M1234	C234 counting mode (on: count down)
M1235	C235 counting mode (on: count down)
M1236	C236 counting mode (on: count down)
M1237	C237 counting mode (on: count down)

M1238	C238 counting mode (on: count down)
M1239	C239 counter mode setting (on: count down)
M1240	C240 counter mode setting (on: count down)
M1241	C241 counter mode setting (on: count down)
M1242	C242 counter mode setting (on: count down)
M1243	C243 counter mode setting (on: count down)
M1244	C244 counter mode setting (on: count down)
M1245	C245 counter mode setting (on: count down)
M1246	C246 counter monitor (on: count down)
M1247	C247 counter monitor (on: count down)
M1248	C248 counter monitor (on: count down)
M1249	C249 counter monitor (on: count down)
M1250	C250 counter monitor (on: count down)
M1251	C251 counter monitor (on: count down)
M1252	C252 counter monitor (on: count down)
M1253	C253 counter monitor (on: count down)
M1254	C254 counter monitor (on: count down)
M1256	System used
M1258	Swap Y0 and Y1 pulse output signal
M1259	Swap Y2 and Y3 pulse output signal
M1260	Let X5 be the reset input signal of all high-speed counter
M1261	DHSCR command High-speed comparison flag
M1264	HHSC0 Reset function enable
M1265	HHSC0 Start function enable
M1266	HHSC1 Reset function enable
M1267	HHSC1 Start function enable
M1268	HHSC2 Reset function enable
M1269	HHSC2 Start function enable
M1270	HHSC3 Reset function enable
M1271	HHSC3 Start function enable
M1272	HHSC0 Reset control
M1273	HHSC0 Start control
M1274	HHSC1 Reset control
M1275	HHSC1 Start control
M1276	HHSC2 Reset control
M1277	HHSC2 Start control
M1278	HHSC3 Reset control
M1279	HHSC3 Start control
M1280	I00 masked
M1281	I10 masked
M1282	I20 masked
M1283	I30 masked
M1284	I40 masked
M1285	I50 masked

M1286	I6 masked
M1287	I7 masked
M1288	I8 masked
M1289	I010 masked
M1290	I020 masked
M1291	I030 masked
M1292	I040 masked
M1293	I050 masked
M1294	I060 masked
M1295	I110 masked
M1296	I120 masked
M1297	I130 masked
M1298	I140 masked
M1299	I150 masked
M1300	I160 masked
M1301	I170 masked
M1302	I180 masked
M1303	Swap high and low byte
M1308	Y0, Y1 emergency stop without deceleration
M1309	Y2, Y3 emergency stop without deceleration
M1310	Y4, Y5 emergency stop without deceleration
M1311	Y6, Y7 emergency stop without deceleration
M1312	Y10, Y11 emergency stop without deceleration
M1313	Y12, Y13 emergency stop without deceleration
M1314	Y14, Y15 emergency stop without deceleration
M1315	Y16, Y17 emergency stop without deceleration
M1316	Y20, Y21 emergency stop without deceleration
M1317	Y22, Y23 emergency stop without deceleration
M1318	Y24, Y25 emergency stop without deceleration
M1319	Y26, Y27 emergency stop without deceleration
M1320	Y30, Y31 emergency stop without deceleration
M1321	Y32, Y33emergency stop without deceleration
M1322	Y34, Y35 emergency stop without deceleration
M1323	Y36, Y37 emergency stop without deceleration
M1324	Y40, Y41 emergency stop without deceleration
M1325	Y42, Y43 emergency stop without deceleration
M1326	Y44, Y45 emergency stop without deceleration
M1327	Y46, Y47 emergency stop without deceleration
M1328	Y50, Y51 emergency stop without deceleration
M1329	Y52, Y53 emergency stop without deceleration
M1330	Y54, Y55 emergency stop without deceleration
M1331	C239 Start/Reset function enable
M1332	C239 Start/Reset function enable
M1333	C240 Start/Reset function enable

M1340	After the CH0 (Y0, Y1) pulse is sent out, an interrupt I110 is generated
M1341	After the CH0 (Y0, Y1) pulse is sent out, an interrupt I110 is generated
M1342	CH0 (Y0, Y1) pulse is sent out at the same time, interrupt I130 is generated
M1343	CH0 (Y0, Y1) pulse is sent out at the same time, interrupt I130 is generated
M1344	Y0, Y1 pulse sending flag
M1345	Y2, Y3 pulse sending flag
M1346	Y4, Y5 pulse sending flag
M1347	Y6, Y7 pulse sending flag
M1348	Y10, Y11 pulse sending flag
M1349	Y12, Y13 pulse sending flag
M1350	Y14, Y15 pulse sending flag
M1351	Y16, Y17 pulse sending flag
M1352	Y20, Y21 pulse sending flag
M1353	Y22, Y23 pulse sending flag
M1354	Y24, Y25 pulse sending flag
M1355	Y26, Y27 pulse sending flag
M1356	Y30, Y31 pulse sending flag
M1357	Y32, Y33 pulse sending flag
M1358	Y34, Y35 pulse sending flag
M1359	Y36, Y37 pulse sending flag
M1360	Y40, Y41 pulse sending flag
M1361	Y42, Y43 pulse sending flag
M1362	Y44, Y45 pulse sending flag
M1363	Y46, Y47 pulse sending flag
M1364	Y50, Y51 pulse sending flag
M1365	Y52, Y53 pulse sending flag
M1366	Y54, Y55 pulse sending flag
M1367	Y56, Y57 pulse sending flag
M1415	Indicating reading from Salve ID#8 is completed

Special D	Function Description
D1000	Watchdog timer (WDT) value (Unit: 1ms)
D1002	Program capacity
D1003	Sum of program memory (sum of the PLC internal program memory. User can identify the content of PLC control program by this register)
D1004	Check code for grammar
D1005	System use
D1008	STEP address when WDT timer is ON
D1009	The occuring history of LV signals will be stored in D1009
D1010	Present scan time (Unit: 0.1ms)
D1011	Minimum scan time (Unit: 0.1ms)
D1012	Maximum scan time (Unit: 0.1ms)
D1015	0~32,767(unit: 0.1ms) addition type of high-speed connection timer
D1018	Geocities PI (Low byte)
D1019	Geocities PI(High byte)
D1020	X0 ~ X7 input filter (Unit: ms)
D1021	ES/EH/EH2/SV: X10 ~ X17 input filter (Unit: ms)SC: X10 ~ X17 input filter (time base: scan cycle), range: 0 ~ 1,000 (Unit: times)
D1024	System use flag
D1025	Communication error code
D1028	Index register E0
D1029	Index register F0
D1034	Frequency measurement card working mode
D1035	Set the number of X input point of RUN/STOP
D1036	COM1 (RS-485) Communication protocol
D1037	HKY key repeat time (ms)
D1038	When PLC MPU is slave, the setting of data response delay time. The setting range is 0~10,000, and the time unit is 0.1ms.
D1039	Constant scan time (ms)
D1040	On state number 1 of STEP point S
D1041	On state number 2 of STEP point S
D1042	On state number 3 of STEP point S
D1043	On state number 4 of STEP point S
D1044	On state number 5 of STEP point S
D1045	On state number 6 of STEP point S
D1046	On state number 7 of STEP point S
D1047	On state number 8 of STEP point S
D1049	On number of alarm point
D1050	Modbus communication data conversion. PLC will automatically convert the ASCII data saved in D1070~D1085 to HEX.
D1051	Modbus communication data conversion. PLC will automatically convert the ASCII data saved in D1070~D1085 to HEX.
D1052	Modbus communication data conversion. PLC will automatically convert the ASCII data saved in D1070~D1085 to HEX.
D1053	Modbus communication data conversion. PLC will automatically convert the ASCII data saved in D1070~D1085 to HEX.
D1054	Modbus communication data conversion. PLC will automatically convert the ASCII data saved in D1070~D1085 to HEX.

D1055	Modbus communication data conversion. PLC will automatically convert the ASCII data saved in D1070~D1085 to HEX.
D1056	Present value of SX/EX MPU analog input channel 0 (CH0) and EH MPU AD card channel 0 (CH0)
D1057	Present value of SX/EX MPU analog input channel 1 (CH1) and EH MPU AD card channel 1 (CH1)
D1058	Present value of EX MPU analog input channel 2 (CH2)
D1059	Present value of EX MPU analog input channel 3 (CH3)
D1061	System error message: Record of the number of errors in the power failure retention area
D1064	System used flag
D1065	System used flag
D1066	Algorithm error code
D1067	Algorithm error code
D1068	Lock the algorithm error address
D1069	M1065~M1067 error address
D1070	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1071	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1072	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1073	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1074	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1075	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1076	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1077	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1078	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1079	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1080	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.

D1081	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1082	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1083	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1084	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1085	When the PLC built-in RS-485 communication command receives feedback signals from receiver. The signals will be saved in the registers D1070~D1085. User can use the contents saved in the registers to check the feedback data.
D1089	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1090	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1091	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1092	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1093	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1094	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1095	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1096	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1097	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1098	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1099	When the PLC built-in RS-485 communication command is executed, the transmitting signals will be stored in the registers D1089~D1099. User can use the contents saved in registers to check the feedback data.
D1100	Corresponding state after LV signal action
D1101	File register start address

D1102	File register copy number
D1103	Set as the starting D number of the file register (must be greater than 2000)
D1104	Acceleration and deceleration pulse output use control register (D) start number
D1119	System use
D1120	COM2(RS-485) Communication protocol
D1121	PLC communication address (the address that save PLC communication address, it is latched)
D1122	Residual words of transmitting data
D1123	Residual words of receiving data
D1124	Start character definition (STX)
D1125	First ending character definition (EXT1)
D1126	Second ending character definition (EXT2)
D1129	RS-485 time-out setting (ms)
D1130	MODBUS return error code record
D1133	System program version
D1135	System hardware version
D1137	Address of operator error occurs
D1139	Connection number of BCD module expansion unit (the maximum is two units)
D1141	System used
D1144	Adjustable slope acceleration and deceleration pulse output Y0 control register (D) start number
D1145	Number of left-side high-speed special extension modules; maximum 8 modules extendable (only supports (for SV)
D1146	Connection number of DISP module expansion unit
D1147	Memory card type (Memory state) b0=0:NO CARD (H0000) b0=1:FLASH CARD exists b8=0:The switch of FLASH CARD is Off (H0001) b8=1:The switch of FLASH CARD is On (H0101)
D1148	System use flag signal
D1149	Function card type: 0:NO CARD 1:RS-232(DVP-F232),DU01(DVPDU01) 2:RS-422(DVP-F422) 3:COM3(DVP-F232S,DVP-F485S) 4:Potentiometer switch(DVP-F6VR) 5:DIP switch(DVP-F8ID) 6:Transitor output card(DVP-F2OT) 7:Digital input(DVP-F4IP) 8:2AD card (DVP-F2AD) 9:2DA card(DVP-F2DA)
D1150	DHSZ instruction multiple set value comparison mode table count register
D1151	DHSZ instruction frequency control mode table count register
D1152	DHSZ D value changed High word
D1153	DHSZ D value changed low word
D1154	Adjustable slope acceleration and deceleration pulse Y0 output function deceleration interval time (10~32,767 ms) recommended value
D1155	Adjustable slope acceleration and deceleration pulse Y0 output function deceleration interval time (-1~32,700 HZ) recommended value
D1156	Special D specified by RTMU instruction (No. K0~K9)
D1157	Special D that indicated by RTMU command (K0~K9)
D1158	Special D that indicated by RTMU command (K0~K9)
D1159	Special D that indicated by RTMU command (K0~K9)
D1160	Special D that indicated by RTMU command (K0~K9)
D1161	Special D that indicated by RTMU command (K0~K9)
D1162	Special D that indicated by RTMU command (K0~K9)
D1163	Special D that indicated by RTMU command (K0~K9)
D1164	Special D that indicated by RTMU command (K0~K9)

D1165	Special D that indicated by RTMU command (K0~K9)
D1166	Rising-edge or falling-edge switch of X10 (SCV1.4 or over models)
D1168	RS instruction, interrupt request when receiving specified data character (I150)
D1169	RS instruction, interrupt request when receiving specified data legnth (I160)
D1170	Single step (Single step) PC value during execution
D1172	Two-phase pulse output frequency (12Hz~20KHz)
D1173	Two-phase pulse output mode selection (K1 and K2)
D1174	Target number of two-phase output pulse (low word)
D1175	Target number of two-phase output pulse (high word)
D1176	Current output number of two-phase pulse (low word)
D1177	Current output number of two-phase pulse (high word)
D1178	VR0 value
D1179	VR1 value
D1180	Interrupt I401 to capture high-speed count value(low word)
D1181	Interrupt I401 to capture high-speed count value(high word)
D1182	Pointer register E1
D1183	Pointer register F1
D1184	Pointer register E2
D1185	Pointer register F2
D1186	Pointer register E3
D1187	Pointer register F3
D1188	Pointer register E4
D1189	Pointer register F4
D1190	Pointer register E5
D1191	Pointer register F5
D1192	Pointer register E6
D1193	Pointer register F6
D1194	Pointer register E7
D1195	Pointer register F7
D1196	Number system stting (provided for DVP-SX series only)
D1197	System use flag signal
D1198	Interrupt I501 to capture high-speed count value(low word)
D1199	Interrupt I501 to capture high-speed count value(high word)
D1200	M0~M999 auxiliary relay start address when power failure
D1201	M0~M999 auxiliary relay power failure retention end address
D1202	M2000~M4095 auxiliary relay power failure start address
D1203	M2000~M4095 auxiliary relay power failure retention end address
D1204	T0~T199, the start address of the 100ms timer to keep the power off
D1205	T0~T199, 100ms timer power failure retention end address
D1206	T200~T239, the start address of the 10ms timer to keep the power off
D1207	T200~T239, 10ms timer power failure retention end address
D1208	C0~C199, 16-bit counter power failure retention start address
D1209	C0~C199, 16-bit counter power failure retention end address
D1210	C200~C234, 32-bit counter start address when power failure
D1211	TCP/IP Port
D1212	IP0.1
D1213	IP2.3
D1214	S0~S899 step point power failure retention start address
D1215	S0~S899 step point power failure retention end address

D1216	D0~D999 register power failure retention start address
D1217	D0~D999 register power failure retention end address
D1218	D2000~D9999 register start address for power failure retention
D1219	D2000~D9999 register power failure retention end address
D1220	mask0.1
D1221	mask2.3
D1222	gateway0.1
D1223	gateway2.3
D1225	The first group of counter (HHSC0) counting mode setting
D1226	The second group counter (HHSC1) counting mode setting
D1227	The third group counter (HHSC2) counting mode setting
D1228	The fourth group of counter (HHSC3) counting mode setting
D1256	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1257	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1258	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1259	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1260	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1261	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1262	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1263	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1264	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1265	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1266	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1267	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1268	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1269	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1270	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1271	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1272	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1273	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1274	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.

D1275	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1276	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1277	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1278	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1279	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1280	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1281	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1282	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1283	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1284	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1285	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1286	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1287	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1288	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1289	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1290	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1291	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1292	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1293	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1294	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1295	MODRW command of RS-485 is built-in. The characters that sent during executing is saved in D1256~D1295. User can check according to the content of these registers.
D1296	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1297	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1298	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1299	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.

D1300	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1301	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1302	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1303	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1304	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1305	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1306	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1307	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1308	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1309	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1310	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1311	MODRW command of RS-485 is built-in. PLC system will convert ASCII in the content of the register that user indicates to HEX and save it in D1296 ?D1311.
D1313	Real time clock (RTC) second 00~59
D1314	Real time clock (RTC) minute 00~59
D1315	Real time clock (RTC) hour 00~23
D1316	Real time clock (RTC) day 01~31
D1317	Real time clock (RTC) month 01~12
D1318	Real time clock (RTC) week 1~7
D1319	Real time clock (RTC) year 00-99
D1340	The 1st step start frequency and the last step end frequency of CH0 pulse
D1341	Maximum output frequency (Low word) (it is fixed to 200KHz)
D1342	Maximum output frequency (High word) (it is fixed to 200KHz)
D1343	Acceleration /Deceleration time of CH0 pulse
D1352	SC: the 1st step starting frequency and last step ending frequency of Y11.EH: the 2nd step starting frequency and last step ending frequency of CH1 (Y2, Y3) output
D1353	SC: Y11 acceleration/deceleration time setting.EH: acceleration/deceleration time setting of 2nd group CH1 (Y2, Y3) output.
D1355	Starting reference for Master to read from Salve ID#1
D1356	Starting reference for Master to read from Salve ID#2
D1357	Starting reference for Master to read from Salve ID#3
D1358	Starting reference for Master to read from Salve ID#4
D1359	Starting reference for Master to read from Salve ID#5
D1360	Starting reference for Master to read from Salve ID#6
D1361	Starting reference for Master to read from Salve ID#7
D1362	Starting reference for Master to read from Salve ID#8
D1363	Starting reference for Master to read from Salve ID#9
D1364	Starting reference for Master to read from Salve ID#10

D1365	Starting reference for Master to read from Salve ID#11
D1366	Starting reference for Master to read from Salve ID#12
D1367	Starting reference for Master to read from Salve ID#13
D1368	Starting reference for Master to read from Salve ID#14
D1369	Starting reference for Master to read from Salve ID#15
D1370	Starting reference for Master to read from Salve ID#16
D1371	When M1070 is on, it determines the time unit of CH0 PWM pulse output
D1372	When M1070 is on, it determines the time unit of CH1 PWM pulse output
D1379	Y4, Y5 start frequency
D1380	Y6, Y7 start frequency
D1381	Y4, Y5 acceleration and deceleration time
D1382	Y6, Y7 acceleration and deceleration time
D1383	Y10, Y11 acceleration and deceleration time
D1384	Y12, Y13 acceleration and deceleration time
D1385	Y14, Y15 acceleration and deceleration time
D1386	Y16, Y17 acceleration and deceleration time
D1387	Y20, Y21 acceleration and deceleration time
D1388	Y22, Y23 acceleration and deceleration time
D1389	Y24, Y25 acceleration and deceleration time
D1390	Y26, Y27 acceleration and deceleration time
D1391	Y30, Y31 acceleration and deceleration time
D1392	Y32, Y33 acceleration and deceleration time
D1393	Y34, Y35 acceleration and deceleration time
D1394	Y36, Y37 acceleration and deceleration time
D1395	Y40, Y41 acceleration and deceleration time
D1396	Y42, Y43 acceleration and deceleration time
D1397	Y44, Y45 acceleration and deceleration time
D1398	Y46, Y47 acceleration and deceleration time
D1399	Y50, Y51 acceleration and deceleration time
D1400	Y10, Y11 start frequency
D1401	Y12, Y13 start frequency
D1402	Y14, Y15 start frequency
D1403	Y16, Y17 start frequency
D1404	Y20, Y21 start frequency
D1405	Y22, Y23 start frequency
D1406	Y24, Y25 start frequency
D1407	Y26, Y27 start frequency
D1408	Y30, Y31 start frequency
D1409	Y32, Y33 start frequency
D1410	Y34, Y35 start frequency
D1411	Y34, Y35 start frequency
D1412	Y40, Y41 start frequency
D1413	Y42, Y43 start frequency
D1414	Y44, Y45 start frequency
D1415	Y50, Y51 start frequency
D1416	Y50, Y51 start frequency
D1417	Y52, Y53 start frequency
D1418	Y54, Y55 start frequency
D1419	Y56, Y57 start frequency

D1420	Y52, Y53 acceleration and deceleration time
D1421	Y54, Y55 acceleration and deceleration time
D1422	Y56, Y57 acceleration and deceleration time
D1423	Write the start communication address setting of slave ID# 9
D1424	Write the start communication address setting of slave station ID# 10 D1425:: Write the start communication address setting of slave station ID# 11
D1426	Y0, Y1 maximum speed (LOW WORD)
D1427	Y0, Y1 maximum speed (HIGH WORD)
D1428	Y2, Y3 maximum speed (LOW WORD)
D1429	Y2, Y3 maximum speed (HIGH WORD)
D1430	Y4, Y5 maximum speed (LOW WORD)
D1431	Y4, Y5 maximum speed (HIGH WORD)
D1432	Y6, Y7 maximum speed (LOW WORD)
D1433	Y6, Y7 maximum speed (HIGH WORD)
D1434	Y10, Y11 maximum speed (LOW WORD)
D1435	Y10, Y11 maximum speed (HIGH WORD)
D1436	Y12, Y13 maximum speed (LOW WORD)
D1437	Y12, Y13 maximum speed (HIGH WORD)
D1438	Y14, Y15 maximum speed (LOW WORD)
D1439	Y14, Y15 maximum speed (HIGH WORD)
D1440	Y16, Y17 maximum speed (LOW WORD)
D1441	Y16, Y17 maximum speed (HIGH WORD)
D1442	Y20, Y21 maximum speed (LOW WORD)
D1443	Y20, Y21 maximum speed (HIGH WORD)
D1444	Y22, Y23 maximum speed (LOW WORD)
D1445	Y22, Y23 maximum speed (HIGH WORD)
D1446	Y24, Y25 maximum speed (LOW WORD)
D1447	Y24, Y25 maximum speed (HIGH WORD)
D1448	Y26, Y27 maximum speed (LOW WORD)
D1449	Y26, Y27 maximum speed (HIGH WORD)
D1450	Setting the data write length for slave ID#1
D1451	Set the data write length for slave ID#3
D1452	Set the data write length for slave ID#3
D1453	Set the data write length for slave ID#4
D1454	Setting the data write length for slave ID#5
D1455	Setting the data write length for slave ID#6
D1456	Setting the data write length for slave ID#7
D1457	Set the data write length for slave ID#8
D1458	Setting the data write length for slave ID#9
D1459	Set the data write length for slave ID#10
D1460	Setting the data write length for slave ID#11
D1461	Set the data write length for slave ID#12
D1462	Set the data write length for slave ID#13
D1463	Setting the data write length for slave ID#14
D1464	Set the data write length for slave ID#15
D1465	Setting the data write length for slave ID#16
D1472	Y44, Y45 sent pulses (low word)
D1473	Y44, Y45 sent pulse number (high word)
D1474	Y44, Y45 target position (low word)

D1475	Y44, Y45 target position (high word)
D1478	Y44, Y45 acceleration (floating point number low word)
D1479	Y44, Y45 acceleration (floating point number high word)
D1480	Y44, Y45 current speed (floating point low word)
D1481	Y44, Y45 current speed (floating point high word)
D1482	Y44, Y45 target speed (floating point low word)
D1483	Y44, Y45 target speed (floating point high word)
D1488	Y46, Y47 sent pulse number (low word)
D1489	Y46, Y47 sent pulse number (high word)
D1490	Y46, Y47 target position (low word)
D1491	Y46, Y47 target position (high word)
D1494	Y46, Y47 acceleration (floating point number low word)
D1495	Y46, Y47 acceleration (floating point number low word)
D1496	Y46, Y47 current speed (floating point low word)
D1497	Y46, Y47 current speed (floating point high word)
D1498	Y46, Y47 target speed (floating point low word)
D1499	Y46, Y47 target speed (floating point high word)
D1504	Y50, Y51 sent pulse number (low word)
D1505	Y50, Y51 sent pulse number (high word)
D1506	Y50, Y51 target position (low word)
D1507	Y50, Y51 target position (high word)
D1510	Y50, Y51 acceleration (floating point number low word)
D1511	Y50, Y51 acceleration (floating point number high word)
D1512	Y50, Y51 current speed (floating point low word)
D1513	Y50, Y51 current speed (floating point high word)
D1514	Y50, Y51 target speed (floating point low word)
D1515	Y50, Y51 target speed (floating point high word)
D1520	Y52, Y53 sent pulse number (low word)
D1521	Y52, Y53 sent pulse number (high word)
D1522	Y52, Y53 target position (low word)
D1523	Y52, Y53 target position (high word)
D1526	Y52, Y53 acceleration (floating point number low word)
D1527	Y52, Y53 acceleration (floating point number high word)
D1528	Y52, Y53 current speed (floating point low word)
D1529	Y52, Y53 current speed (floating point high word)
D1530	Y52, Y53 target speed (floating point low word)
D1531	Y52, Y53 target speed (floating point high word)
D1536	Y54, Y55 sent pulse number (low word)
D1537	Y54, Y55 sent pulse number (high word)
D1538	Y54, Y55 target position (low word)
D1539	Y54, Y55 target position (high word)
D1542	Y54, Y55 acceleration (floating point number low word)
D1543	Y54, Y55 acceleration (floating point high word)
D1544	Y54, Y55 current speed (floating point low word)
D1545	Y54, Y55 current speed (floating point high word)
D1546	Y54, Y55 target speed (floating point low word)
D1547	Y54, Y55 target speed (floating point high word)
D1552	Y56, Y57 sent pulse number (low word)
D1553	Y56, Y57 sent pulse number (high word)

D1554	Y56, Y57 target position (low word)
D1555	Y56, Y57 target position (high word)
D1558	Y56, Y57 acceleration (floating point number low word)
D1559	Y56, Y57 acceleration (floating point number high word)
D1560	Y56, Y57 current speed (floating point low word)
D1561	Y56, Y57 current speed (floating point high word)
D1562	Y56, Y57 target speed (floating point low word)
D1563	Y56, Y57 target speed (floating point high word)
D1568	Y0, Y1 absolute position after ZRN instruction
D1569	Y2, Y3 Absolute position after ZRN instruction
D1570	Y4, Y5 absolute position after ZRN instruction
D1571	Y6, Y7 absolute position after ZRN instruction
D1572	Y10, Y11 Absolute position after executing ZRN instruction
D1573	Y12, Y13 Absolute position after ZRN instruction
D1574	Y14, Y15 Absolute position after ZRN instruction
D1575	Y16, Y17 Absolute position after ZRN instruction is executed
D1576	Y20, Y21 Absolute position after executing ZRN instruction
D1577	Y22, Y23 Absolute position after executing ZRN instruction
D1578	Y24, Y25 Absolute position after ZRN instruction is executed
D1579	Y26, Y27 Absolute position after executing ZRN instruction
D1580	Y30, Y31 Absolute position after executing ZRN instruction
D1581	Y32, Y33 Absolute position after ZRN instruction
D1582	Y34, Y35 absolute position after ZRN instruction
D1583	Y36, Y37 Absolute position after executing ZRN instruction
D1584	Y40, Y41 Absolute position after executing ZRN instruction
D1585	Y42, Y43 Absolute position after executing ZRN instruction
D1586	Y44, Y45 absolute position after ZRN instruction
D1587	Y46, Y47 Absolute position after ZRN instruction
D1588	Y50, Y51 Absolute position after executing ZRN instruction
D1589	Y52, Y53 absolute position after ZRN instruction
D1590	Y54, Y55 Absolute position after ZRN instruction
D1591	Y56, Y57 absolute position after ZRN instruction
D1648	Y0, Y1 sent pulse number (low word)
D1649	Y0, Y1 sent pulse number (high word)
D1650	Y0, Y1 target position (low word)
D1651	Y0, Y1 target position (high word)
D1654	Y0, Y1 acceleration (floating point number low word)
D1655	Y0, Y1 acceleration (floating point number high word)
D1656	Y0, Y1 current speed (floating point low word)
D1657	Y0, Y1 current speed (floating point high word)
D1658	Y0, Y1 target speed (floating point low word)
D1659	Y0, Y1 target speed (floating point high word)
D1664	Y2, Y3 sent pulse number (low word)
D1665	Y2, Y3 sent pulse number (high word)
D1666	Y2, Y3 target position (low word)
D1667	Y2, Y3 target position (high word)
D1670	Y2, Y3 acceleration (floating point number low word)
D1671	Y2, Y3 acceleration (floating point high word)
D1672	Y2, Y3 current speed (floating point low word)

D1673	Y2, Y3 current speed (floating point high word)
D1674	Y2, Y3 target speed (floating point low word)
D1675	Y2, Y3 target speed (floating point high word)
D1680	Y4, Y5 sent pulse number (low word)
D1681	Y4, Y5 sent pulse number (high word)
D1682	Y4, Y5 target position (low word)
D1683	Y4, Y5 target position (high word)
D1686	Y4, Y5 acceleration (floating point number low word)
D1687	Y4, Y5 acceleration (floating point high word)
D1688	Y4, Y5 current speed (floating point low word)
D1689	Y4, Y5 current speed (floating point high word)
D1690	Y4, Y5 target speed (floating point low word)
D1691	Y4, Y5 target speed (floating point high word)
D1696	Y6, Y7 sent pulse number (low word)
D1697	Y6, Y7 sent pulse number (high word)
D1698	Y6, Y7 target position (low word)
D1699	Y6, Y7 target position (high word)
D1702	Y6, Y7 acceleration (floating point number low word)
D1703	Y6, Y7 acceleration (floating point number high word)
D1704	Y6, Y7 current speed (floating point low word)
D1705	Y6, Y7 current speed (floating point high word)
D1706	Y6, Y7 target speed (floating point low word)
D1707	Y6, Y7 target speed (points high word)
D1712	Y10, Y11 sent pulse number (low word)
D1713	Y10, Y11 sent pulse number (high word)
D1714	Y10, Y11 target position (low word)
D1715	Y10, Y11 target position (high word)
D1718	Y10, Y11 acceleration (floating point number low word)
D1719	Y10, Y11 acceleration (floating point number high word)
D1720	Y10, Y11 current speed (floating point low word)
D1721	Y10, Y11 current speed (floating point high word)
D1722	Y10, Y11 target speed (floating point low word)
D1723	Y10, Y11 target speed (floating point high word)
D1728	Y12, Y13 sent pulse number (low word)
D1729	Y12, Y13 sent pulse number (high word)
D1730	Y12, Y13 target position (low word)
D1731	Y12, Y13 target position (high word)
D1734	Y12, Y13 acceleration (floating point number low word)
D1735	Y12, Y13 acceleration (floating point number high word)
D1736	Y12, Y13 current speed (floating point low word)
D1737	Y12, Y13 current speed (floating point high word)
D1738	Y12, Y13 target speed (floating point low word)
D1739	Y12, Y13 target speed (floating point high word)
D1744	Y14, Y15 sent pulse number (low word)
D1745	Y14, Y15 sent pulse number (high word)
D1746	Y14, Y15 target position (low word)
D1747	Y14, Y15 target position (high word)
D1750	Y14, Y15 acceleration (floating point number low word)
D1751	Y14, Y15 acceleration (floating point number high word)

D1752	Y14, Y15 current speed (floating point low word)
D1753	Y14, Y15 current speed (floating point high word)
D1754	Y14, Y15 target speed (floating point low word)
D1755	Y14, Y15 target speed (floating point high word)
D1760	Y16, Y17 sent pulse number (low word)
D1761	Y16, Y17 sent pulse number (high word)
D1762	Y16, Y17 target position (low word)
D1763	Y16, Y17 target position (high word)
D1766	Y16, Y17 acceleration (floating point number low word)
D1767	Y16, Y17 acceleration (floating point number high word)
D1768	Y16, Y17 current speed (floating point low word)
D1769	Y16, Y17 current speed (floating point high word)
D1770	Y16, Y17 target speed (floating point low word)
D1771	Y16, Y17 target speed (floating point high word)
D1776	Y20, Y21 sent pulse number (low word)
D1777	Y20, Y21 sent pulse number (high word)
D1778	Y20, Y21 target position (low word)
D1779	Y20, Y21 target position (high word)
D1782	Y20, Y21 acceleration (floating point number low word)
D1783	Y20, Y21 acceleration (floating point number high word)
D1784	Y20, Y21 current speed (floating point low word)
D1785	Y20, Y21 current speed (floating point high word)
D1786	Y20, Y21 target speed (floating point low word)
D1787	Y20, Y21 target speed (floating point high word)
D1792	Y22, Y23 sent pulse number (low word)
D1793	Y22, Y23 sent pulse number (high word)
D1794	Y22, Y23 target position (low word)
D1795	Y22, Y23 target position (high word)
D1798	Y22, Y23 acceleration (floating point number low word)
D1799	Y22, Y23 acceleration (floating point high word)
D1800	Y22, Y23 current speed (floating point low word)
D1801	Y22, Y23 current speed (floating point high word)
D1802	Y22, Y23 target speed (floating point low word)
D1803	Y22, Y23 target speed (floating point high word)
D1808	Y24, Y25 sent pulse number (low word)
D1809	Y24, Y25 sent pulse number (high word)
D1810	Y24, Y25 target position (low word)
D1811	Y24, Y25 target position (high word)
D1814	Y24, Y25 acceleration (floating point low word)
D1815	Y24, Y25 acceleration (floating point number high word)
D1816	Y24, Y25 current speed (floating point low word)
D1817	Y24, Y25 current speed (floating point high word)
D1818	Y24, Y25 target speed (floating point low word)
D1819	Y24, Y25 target speed (floating point high word)
D1824	Y26, Y27 sent pulse number (low word)
D1825	Y26, Y27 sent pulse number (high word)
D1826	Y26, Y27 target position (low word)
D1827	Y26, Y27 target position (high word)
D1830	Y26, Y27 acceleration (floating point number low word)

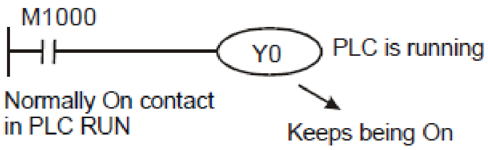
D1831	Y26, Y27 acceleration (floating point number high word)
D1832	Y26, Y27 current speed (floating point low word)
D1833	Y26, Y27 current speed (floating point high word)
D1834	Y26, Y27 target speed (floating point low word)
D1835	Y26, Y27 target speed (floating point high word)
D1840	Y30, Y31 sent pulse number (low word)
D1841	Y30, Y31 sent pulse number (high word)
D1842	Y30, Y31 target position (low word)
D1843	Y30, Y31 target position (high word)
D1846	Y30, Y31 acceleration (floating point number low word)
D1847	Y30, Y31 acceleration (floating point number high word)
D1848	Y30, Y31 current speed (floating point low word)
D1849	Y30, Y31 current speed (floating point high word)
D1850	Y30, Y31 target speed (floating point low word)
D1851	Y30, Y31 target speed (floating point high word)
D1856	Y32, Y33 sent pulse number (low word)
D1857	Y32, Y33 sent pulse number (high word)
D1858	Y32, Y33 target position (low word)
D1859	Y32, Y33 target position (high word)
D1862	Y32, Y33 acceleration (floating point low word)
D1863	Y32, Y33 acceleration setting (floating point number high word)
D1864	Y32, Y33 current speed (floating point low word)
D1865	Y32, Y33 current speed (floating point high word)
D1866	Y32, Y33 target speed (floating point low word)
D1867	Y32, Y33 target speed (floating point high word)
D1872	Y34, Y35 sent pulse number (low word)
D1873	Y34, Y35 sent pulse number (high word)
D1874	Y34, Y35 target position (low word)
D1875	Y34, Y35 target position (high word)
D1878	Y34, Y35 acceleration (floating point number low word)
D1879	Y34, Y35 acceleration (floating point number high word)
D1880	Y34, Y35 current speed (floating point low word)
D1881	Y34, Y335 current speed (floating point high word)
D1882	Y34, Y35 target speed (floating point low word)
D1883	Y34, Y35 target speed (floating point high word)
D1888	Y36, Y37 sent pulse number (low word)
D1889	Y36, Y37 sent pulse number (high word)
D1890	Y36, Y37 target position (low word)
D1891	Y36, Y37 target position (high word)
D1894	Y36, Y37 acceleration (floating point number low word)
D1895	Y36, Y37 acceleration (floating point number high word)
D1896	Y36, Y37 current speed (floating point low word)
D1897	Y36, Y37 current speed (floating point high word)
D1898	Y36, Y37 target speed (floating point low word)
D1899	Y36, Y37 target speed (floating point high word)
D1904	Y40, Y41 sent pulse number (low word)
D1905	Y40, Y41 sent pulse number (high word)
D1906	Y40, Y41 target position (low word)
D1907	Y40, Y41 target position (high word)

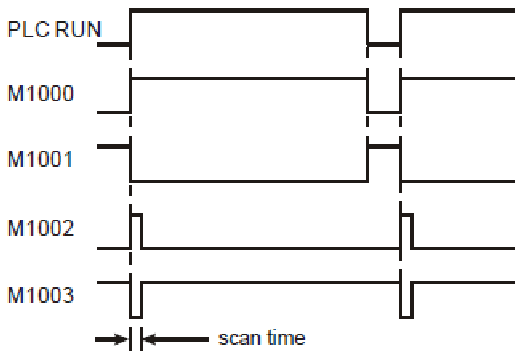
D1910	Y40, Y41 acceleration (floating point number low word)
D1911	Y40, Y41 acceleration (floating point number high word)
D1912	Y40, Y41 current speed (floating point low word)
D1913	Y40, Y41 current speed (floating point high word)
D1914	Y40, Y41 target speed (floating point low word)
D1915	Y40, Y41 target speed (floating point high word)
D1920	Y42, Y43 sent pulse number (low word)
D1921	Y42, Y43 sent pulse number (high word)
D1922	Y42, Y43 target position (low word)
D1923	Y42, Y43 target position (high word)
D1926	Y42, Y43 acceleration (floating point number low word)
D1927	Y42, Y43 acceleration (floating point number high word)
D1928	Y42, Y43 current speed (floating point low word)
D1929	Y42, Y43 current speed (floating point high word)
D1930	Y42, Y43 target speed (floating point low word)
D1931	Y42, Y43 target speed (floating point high word)
D1936	Y0, Y1 deceleration time
D1937	Y2, Y3 deceleration time
D1938	Y4, Y5 deceleration time
D1939	Y6, Y7 deceleration time
D1940	Y10, Y11 deceleration time
D1941	Y12, Y13 deceleration time
D1942	Y14, Y15 deceleration time
D1943	Y16, Y17 deceleration time
D1944	Y20, Y21 deceleration time
D1945	Y22, Y23 deceleration time
D1946	Y24, Y25 deceleration time
D1947	Y26, Y27 deceleration time
D1948	Y30, Y31 deceleration time
D1949	Y32, Y33 deceleration time
D1950	Y34, Y35 deceleration time
D1951	Y36, Y37 deceleration time
D1952	Y40, Y41 deceleration time
D1953	Y42, Y43 deceleration time
D1954	Y44, Y45 deceleration time
D1955	Y46, Y47 deceleration time
D1956	Y50, Y51 deceleration time
D1957	Y52, Y53 deceleration time
D1958	Y54, Y55 deceleration time
D1959	Y56, Y57 deceleration time
D1966	Number of modules
D1968	Expansion module 1 read address
D1969	Expansion module 2 read address
D1970	Expansion module 3 read address
D1971	Expansion module 4 read address
D1972	Expansion module 5 read address
D1973	Expansion module 6 read address
D1974	Expansion module 7 read address
D1975	Expansion module 8 read address

D1976	Expansion module 9 read address
D1977	Expansion module 10 read address
D1978	Expansion module 11 read address
D1979	Expansion module 12 read address
D1980	Expansion module 13 read address
D1981	Expansion module 14 read address
D1982	Expansion module 15 read address
D1983	Expansion module 16 read address
D1984	Expansion module 1 write address
D1985	Expansion module 2 write address
D1986	Expansion module 3 write address
D1987	Expansion module 4 write address
D1988	Expansion module 5 write address
D1989	Expansion module 6 write address
D1990	Expansion module 7 write address
D1991	Expansion module 8 write address
D1992	Expansion module 9 write address
D1993	Expansion module 10 write address
D1994	Expansion module 11 write address
D1995	Expansion module 12 write address
D1996	Expansion module 13 write address
D1997	Expansion module 14 write address
D1998	Expansion module 15 write address
D1999	Expansion module 16 write address

2.11 Functions of Special Auxiliary Relays and Special Registers

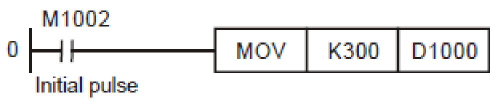
◆ PLC Operation Flag (M1000-M1003)





1、 M1001: M1001 (B contact) is constantly "On" during operation and detection. When PLC is in RUN status, M1001 remains "On" .
2、 M1002: M1002 is to enable single positive pulse at the moment when RUN is activated (Normally OFF). The pulse width = 1 scan time. Use this contact for all kinds of initial settings.
3、 M1003: M1003 is to enable single negative pulse at the moment when RUN is activated (Normally ON).

◆ Monitor Timer D1000



1、 Monitor timer is used for monitoring PLC scan time. When the scan time exceeds the set time in the monitor timer, the red ERROR LED indicator remains beaconing and all outputs will be "Off" .
2、 The initial set value of the time in the monitor timer is 200ms. If the program is long or the operation is too complicated, MOV instruction can be used for changing the set value. See the example below for SV = 300ms.
3、 The maximum set value in the monitor timer is 32,767ms. Please be noted that if the SV is too big, the timing of detecting operational errors will be delayed. Therefore, it is suggested that you remain the scan time of shorter than 200ms.
4、 Complicated instruction operations or too many extension modules being connected to the CPU will result in the scan time being too long. Check D1010 ~ D1012 to see if the scan time exceeds the SV in D1000. In this case, besides modifying the SV in D1000, you can also add WDT instruction (ZL 07) into the PLC program. When the CPU execution progresses to WDT instruction, the internal monitor timer will be cleared as "0" and the scan time will not exceed the set value in the monitor timer.。

◆ Program Capacity D1002

Program capacity of PLC model: 30K steps

M1004 D1004、D1137

1、 When errors occur in syntax check, ERROR LED indicator will flash and special relay M1004 = On.

2、 Timings for PLC syntax check:

a) When the power goes from "Off" to "On".

3、 The syntax check may start due to illegal use of instruction operands (devices) or incorrect program syntax loop. The error can be detected by the error code in D1004 and error table. The address where the error exists will be stored in D1137. (The address value in D1137 will be invalid if the error is a general loop error.)

◆ Scan Time-out Timer
M1008、D1008

1、 M1008 = On: Scan time-out occurs during the execution of the program, and PLC ERROR LED indicator remains beaconing.

2、 Monitor the content (STEP address when WDT timer is "On").。

◆ Scan Time Monitor
D1010~D1012

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

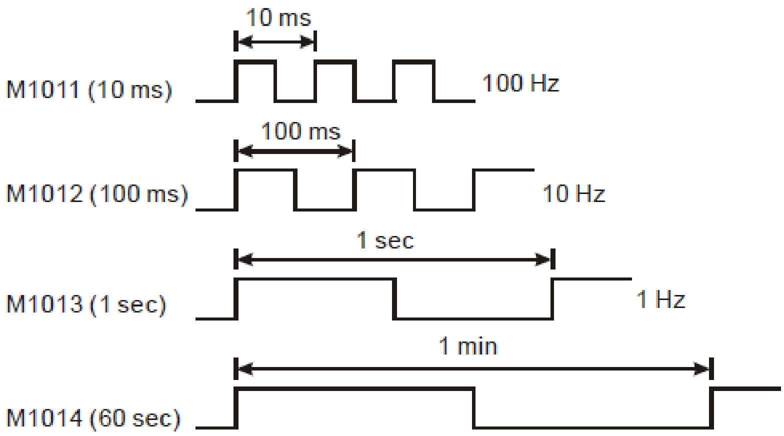
1、 D1010: Present scan time value.

2、 D1011: Minimum scan time value.

3、 D1012: Maximum scan time value.

◆ Internal Clock Pulse
M1011~M1014

The PLC mainframe has the following 4 kinds of clock pulses. As long as the PLC is powered on, these 4 kinds of clock pulses will act automatically.



◆ Real Time Clock
M1016、M1017
M1076 D1313~D1319

1、Special M and special D relevant to RTC

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

2、If the set value in RTC is incorrect, the time will be recovered as "Saturday, 00:00 Jan. 1, 2000" when PLC is powered and restarted.

3、D1313 ~ D1319 will immediately update the RTC only when in TRD instruction or monitoring mode.

◆ π (PI)
D1018、D1019

1、D1018 and D1019 are combined as 32-bit data register for storing the floating point value of π(PI).

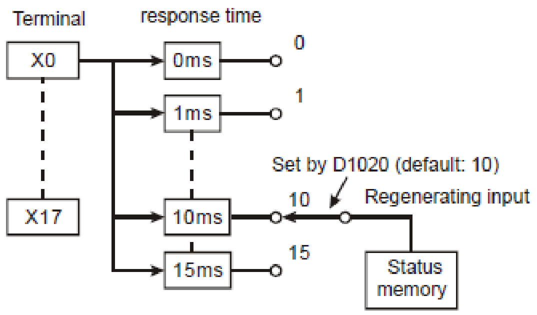
2、 Floating point value = H 40490FDB

◆ Adjustment on Input Terminal Response Time D1020、D1021

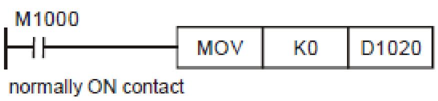
1、D1020 can be used for setting up the response time of receiving pulses at X0 ~X7 .(Setup range: 0 ~ 60; Unit: ms).

2、D1021 can be used for setting up the response time of receiving pulses at X10 ~X17 .(Setup range: 0 ~ 60; Unit: ms).

3、When the power of PLC goes from "Off" to "On" , the content of D1020 and D1021 turn to 10 automatically.



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



5、There is no need to make adjustment on response time when using high-speed counters and interruptions during the program.

◆ Execution completed flag: After the pulse output is completed, M will be set to ON

Applicable instructions: ZL 155 DABSR、 ZL 156 ZRN、ZL 158 DRVI、ZL 159 DRVA、PLSY、PLSR			
Output device	Pulse output complete flag	Output device	Pulse output complete flag
Y0、Y1	M1029	Y30、Y31	M1110
Y2、Y3	M1030	Y32、Y33	M1111
Y4、Y5	M1036	Y34、Y35	M1112
Y6、Y7	M1037	Y36、Y37	M1113
Y10、Y11	M1102	Y40、Y41	M1114
Y12、Y13	M1103	Y42、Y43	M1115
Y14、Y15	M1104	Y44、Y45	M1116

Y16、Y17	M1105	Y46、Y47	M1117
Y20、Y21	M1106	Y50、Y51	M1118
Y22、Y23	M1107	Y52、Y53	M1119
Y24、Y25	M1108	Y54、Y55	M1205
Y26、Y27	M1109	Y56、Y57	M1206

1、 After the pulse output is completed, the corresponding pulse completion flag M point will be set to On, and when the pulse sending command is Off, the corresponding pulse completion flag M point will turn Off. When the instruction is restarted next time, the corresponding pulse completion flag bit M becomes Off again, and then becomes On again after completion.

2、 ZL 63 INCD: When the comparison of the specified number of groups is completed, M1029 will be On for one scan cycle.

3、 ZL 67 RAMP, ZL 69 SORT:

- When the command is executed, M1029 = On, and M1029 must be cleared by the user.
- When this command is Off, M1029 becomes Off.

2.12 Communication Addresses of Devices in PLC

<table>
<tr><th>Device</th><th colspan="2">Range</th><th>Type</th><th>PLC Com. Address (hex)</th><th>Modbus Com. Address (dec)</th></tr>
<tr><td>S</td><td colspan="2">000~255</td><td>Bit</td><td>0000~00FF</td><td>000001~000256</td></tr>
<tr><td>S</td><td colspan="2">246~511</td><td>Bit</td><td>0100~01FF</td><td>000257~000512</td></tr>
<tr><td>S</td><td colspan="2">512~767</td><td>Bit</td><td>0200~02FF</td><td>000513~000768</td></tr>
<tr><td>S</td><td colspan="2">768~1023</td><td>Bit</td><td>0300~03FF</td><td>000769~001024</td></tr>
<tr><td>X</td><td colspan="2">000~377 (Octal)</td><td>Bit</td><td>0400~04FF</td><td>101025~101280</td></tr>
<tr><td>Y</td><td colspan="2">000~377 (Octal)</td><td>Bit</td><td>0500~05FF</td><td>001281~001536</td></tr>
<tr><td rowspan="2">T</td><td colspan="2" rowspan="2">000~255</td><td>Bit</td><td>0600~06FF</td><td>001537~001792</td></tr>
<tr><td>Word</td><td>0600~06FF</td><td>401537~401792</td></tr>
<tr><td>M</td><td colspan="2">000~255</td><td>Bit</td><td>0800~08FF</td><td>002049~002304</td></tr>
<tr><td>M</td><td colspan="2">256~511</td><td>Bit</td><td>0900~09FF</td><td>002305~002560</td></tr>
<tr><td>M</td><td colspan="2">512~767</td><td>Bit</td><td>0A00~0AFF</td><td>002561~002816</td></tr>
<tr><td>M</td><td colspan="2">768~1023</td><td>Bit</td><td>0B00~0BFF</td><td>002817~003072</td></tr>
<tr><td>M</td><td colspan="2">1024~1279</td><td>Bit</td><td>0C00~0CFF</td><td>003073~003328</td></tr>
<tr><td>M</td><td colspan="2">1280~1535</td><td>Bit</td><td>0D00~0DFF</td><td>003329~003584</td></tr>
<tr><td>M</td><td colspan="2">1536~1791</td><td>Bit</td><td>B000~B0FF</td><td>045057~045312</td></tr>
<tr><td>M</td><td colspan="2">1792~2047</td><td>Bit</td><td>B100~B1FF</td><td>045313~045568</td></tr>
<tr><td>M</td><td colspan="2">2048~2303</td><td>Bit</td><td>B200~B2FF</td><td>045569~045824</td></tr>
<tr><td>M</td><td colspan="2">2304~2559</td><td>Bit</td><td>B300~B3FF</td><td>045825~046080</td></tr>
<tr><td>M</td><td colspan="2">2560~2815</td><td>Bit</td><td>B400~B4FF</td><td>046081~046336</td></tr>
<tr><td>M</td><td colspan="2">2816~3071</td><td>Bit</td><td>B500~B5FF</td><td>046337~046592</td></tr>
<tr><td>M</td><td colspan="2">3072~3327</td><td>Bit</td><td>B600~B6FF</td><td>046593~046848</td></tr>
<tr><td>M</td><td colspan="2">3328~3583</td><td>Bit</td><td>B700~B7FF</td><td>046849~047104</td></tr>
<tr><td>M</td><td colspan="2">3584~3839</td><td>Bit</td><td>B800~B8FF</td><td>047105~047360</td></tr>
<tr><td>M</td><td colspan="2">3840~4095</td><td>Bit</td><td>B900~B9FF</td><td>047361~047616</td></tr>
<tr><td rowspan="4">C</td><td rowspan="2">0~199</td><td rowspan="2">16-bit</td><td>Bit</td><td>0E00~0EC7</td><td>003585~003784</td></tr>
<tr><td>Word</td><td>0E00~0EC7</td><td>403585~403784</td></tr>
<tr><td rowspan="2">200~255</td><td rowspan="2">32-bit</td><td>Bit</td><td>0EC8~0EFF</td><td>003785~003840</td></tr>
<tr><td>Word</td><td>0700~076F</td><td>403785~403840</td></tr>
<tr><td>D</td><td colspan="2">000~256</td><td>Word</td><td>1000~10FF</td><td>404097~404352</td></tr>
<tr><td>D</td><td colspan="2">256~511</td><td>Word</td><td>1100~11FF</td><td>404353~404608</td></tr>
<tr><td>D</td><td colspan="2">512~767</td><td>Word</td><td>1200~12FF</td><td>404609~404864</td></tr>
<tr><td>D</td><td colspan="2">768~1023</td><td>Word</td><td>1300~13FF</td><td>404865~405120</td></tr>
<tr><td>D</td><td colspan="2">1024~1279</td><td>Word</td><td>1400~14FF</td><td>405121~405376</td></tr>
<tr><td>D</td><td colspan="2">1280~1535</td><td>Word</td><td>1500~15FF</td><td>405377~405632</td></tr>
<tr><td>D</td><td colspan="2">1536~1791</td><td>Word</td><td>1600~16FF</td><td>405633~405888</td></tr>
<tr><td>D</td><td colspan="2">1792~2047</td><td>Word</td><td>1700~17FF</td><td>405889~406144</td></tr>
<tr><td>D</td><td colspan="2">2048~2303</td><td>Word</td><td>1800~18FF</td><td>406145~406400</td></tr>
<tr><td>D</td><td colspan="2">2304~2559</td><td>Word</td><td>1900~19FF</td><td>406401~406656</td></tr>
<tr><td>D</td><td colspan="2">2560~2815</td><td>Word</td><td>1A00~1AFF</td><td>406657~406912</td></tr>
</table>

Device	Range	Type	PLC Com. Address (hex)	Modbus Com. Address (dec)
D	2816~3071	Word	1B00~1BFF	406913~407168
D	3072~3327	Word	1C00~1CFF	407169~407424
D	3328~3583	Word	1D00~1DFF	407425~407680
D	3584~3839	Word	1E00~1EFF	407681~407936
D	3840~4095	Word	1F00~1FFF	407937~408192
D	4096~4351	Word	9000~90FF	436865~437120
D	4352~4607	Word	9100~91FF	437121~437376
D	4608~4863	Word	9200~92FF	437377~437632
D	4864~5119	Word	9300~93FF	437633~437888
D	5120~5375	Word	9400~94FF	437889~438144
D	5376~5631	Word	9500~95FF	438145~438400
D	5632~5887	Word	9600~96FF	438401~438656
D	5888~6143	Word	9700~97FF	438657~438912
D	6144~6399	Word	9800~98FF	438913~439168
D	6400~6655	Word	9900~99FF	439169~439424
D	6656~6911	Word	9A00~9AFF	439425~439680
D	6912~7167	Word	9B00~9BFF	439681~439936
D	7168~7423	Word	9C00~9CFF	439937~440192
D	7424~7679	Word	9D00~9DFF	440193~440448
D	7680~7935	Word	9E00~9EFF	440449~440704
D	7936~8191	Word	9F00~9FFF	440705~440960
D	8192~8447	Word	A000~A0FF	440961~441216
D	8448~8703	Word	A100~A1FF	441217~441472
D	8704~8959	Word	A200~A2FF	441473~441728
D	8960~9215	Word	A300~A3FF	441729~441984
D	9216~9471	Word	A400~A4FF	441985~442240
D	9472~9727	Word	A500~A5FF	442241~442496
D	9728~9983	Word	A600~A6FF	442497~442752
D	9984~10239	Word	A700~A7FF	442753~443008
D	10234~10495	Word	A800~A8FF	443009~443246
D	10496~10751	Word	A900~A9FF	443247~443502
D	10752~11007	Word	AA00~AAFF	443503~443758
D	11008~11263	Word	AB00~ABFF	443759~444014
D	11264~11519	Word	AC00~ACFF	444015~444270
D	11520~11775	Word	AD00~ADFF	444271~444526
D	11776~11999	Word	AE00~AEDF	444527~444750

3 Basic Instructions

3.1Basic Instructions and Step Ladder Instructions

1、Basic Instructions

Instruction Code	Function	Operands
LD	Loading in A contact	X、Y、M、S、T、C
LDI	Loading in B contact	X、Y、M、S、T、C
AND	Series connection- A contact	X、Y、M、S、T、C
ANI	Series connection- B contact	X、Y、M、S、T、C
OR	Parallel connection- A contact	X、Y、M、S、T、C
ORI	Parallel connection- B contact	X、Y、M、S、T、C
ANB	Series connection- loop blocks	N/A
ORB	Parallel connection- loop blocks	N/A
MPS	Store the current result of the internal PLC operations	N/A
MRD	Reads the current result of the internal PLC operations	N/A
MPP	Pops (recalls and removes) the currently stored result	N/A

2、Output instructions

Instruction Code	Function	Operands
OUT	Output coil	Y、S、M
SET	Latched（On)	Y、S、M
RST	Clear the contacts or the registers	Y、 M、S、T、C、D、E、F

3、Timers, Counters

Instruction Code	Function	Operands
TMR	16-bit timer	T-K or T-D
CNT	16-bit counter	C-K or C-D（16 bits)
DCNT	32-bit counter	Z、 C-K or C-D（32 bits))

4、Main control instructions

Instruction Code	Function	Operands
MC	Master control start	N0~N7
MCR	Master control reset	N0~N7

1、 nstructions for detecting the contacts of rising-/falling-edge

Instruction Code	Function	Operands
LDP	Rising-edge detection operation	S、X、Y、M、T、C
LDF	Falling-edge detection operation	S、X、Y、M、T、C
ANDP	Rising-edge series connection	S、X、Y、M、T、C
ANDF	Falling-edge series connection	S、X、Y、M、T、C
ORP	Rising-edge parallel connection	S、X、Y、M、T、C

ORF	Falling-edge parallel connection	S、X、Y、M、T、C

2、Rising-/falling-edge output instructions

Instruction Code	Function	Operands
PLS	Rising-edge output	Y、M
PLF	Falling-edge output	Y、M

3、End instruction

Instruction Code	Function	Operands
END	Program ends	N/A

4、Other instructions

Instruction Code	Function	Operands
NOP	No operation	N/A
INV	Inverting operation	N/A
P	Pointer	P0~P255
I	Interruption program marker	I□□□

5、Step ladder instructions

Instruction Code	Function	Operands
STL	Step transition ladder start instruction	S
RET	Step transition ladder return instruction	N/A

3.2 [LD] , [LDI] , [AND]

Mnemonic	Function						
LD	Loading in A contact						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Instruction description:** LD instruction is used to connect the contacts to the bus. Support X, Y, M, S components can be modified, for example: LD X0E1

2、**Program Example:**

Mnemonic	Function						
LDI	Loading in B contact						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

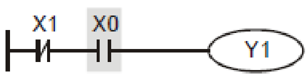
1、**Instruction description:** LD instruction is used to connect the contacts to the bus. Support X, Y, M, S components can be modified, for example: LDI X0E1

2、**Program Example:**

Mnemonic	Function						
AND	Series connection- A contact						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Instruction description:** Use the AND instruction to connect a contact in series. The number of serial contacts is not limited, and this instruction can be used multiple times. Support X, Y, M, S components can be modified, for example: AND X0E1

2、**Program Example**:



3.3 [ANI] , [OR] , [ORI]

Mnemonic	Function						
ANI	Series connection- B contact						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

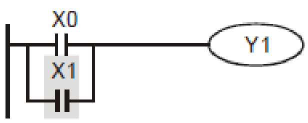
1、**Instruction description:** One contact can be connected in series with ANI instruction. The number of serial contacts is not limited, and this instruction can be used multiple times. Support X, Y, M, S components can be modified, for example: ANI X0E1

2、**Program Example**

Mnemonic	Function						
OR	Parallel connection- A contact						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Instruction description**： OR is used as a parallel connection command for a contact. Support X, Y, M, S components can be modified, for example: OR X1E1

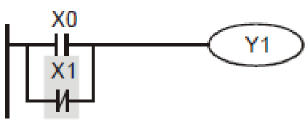
2、**Program Example:**



Mnemonic	Function						
ORI	Parallel connection- B contact						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Instruction description:** ORI is used as a parallel connection command for a contact. Support X, Y, M, S components can be modified, for example: ORI X1E1

2、**Program Example:**



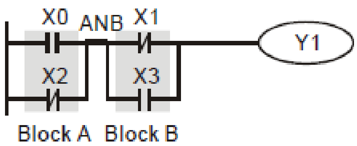
3.4 [ANB] , [ORB]

Mnemonic	Function
ANB	Series connection- loop blocks
Operand	N/A

1、**Instruction description**: To perform the "AND" operation of the preserved logic results and content in the accumulative register.

2、**Program Example**:

Ladder diagram:



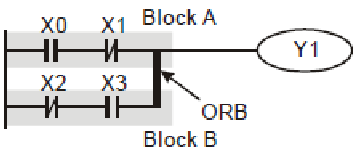
Instruction code:		Operation:
LD	X0	Loading in contact A of X0
ORI	X2	Connecting to contact B of X2 in parallel
LDI	X1	Loading in contact B of X1
OR	X3	Connecting to contact A of X3 in parallel
ANB		Connecting circuit block in series
OUT	Y1	Driving Y1 coil

Mnemonic	Function
ORB	Parallel connection- loop blocks
Operand	N/A

1、**Instruction description**: To perform the "OR" operation of the preserved logic results and content in the accumulative register.

2、**Program Example**:

Ladder diagram:



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

3.5 [MPS] , [MRD], [MPP]

Mnemonic	Function
MPS	Store the current result of the internal PLC operations
Operand	N/A

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

Mnemonic	Function
MRD	Reads the current result of the internal PLC operations
Operand	N/A

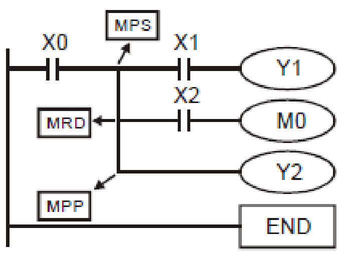
1、**Explanations:** To read the operational result and store it into the accumulative register (the pointer of operational result stays intact).

Mnemonic	Function
MPP	Pops (recalls and removes) the currently stored result
Operand	N/A

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

2、**Program Example:**

Ladder diagram:



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

3.6 [OUT] , [SET], [RST]

Mnemonic	Function						
OUT	Output coil						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	×	√	√	√	×	×	×

1、**Explanations**: Output the result of the logic operation before the OUT instruction to the specified component. Support Y, M, S components can be modified, for example: OUT Y1E2

2、**Actions of coil contact:**

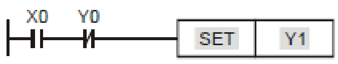
Operational result	OUT instruction		
	Coil	Contact	
		A contact (normally open)	B contact (normally closed)
FALSE	Off	Off	On
TRUE	On	On	Off

Mnemonic	Function						
SET	Latched (On)						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	×	√	√	√	×	×	×

1、**Explanations:** When the SET instruction is driven, its designated device will be "On" and keep being On both when SET instruction is still being driven or not driven. Use RST instruction to set "Off" the device. Support Y, M, S components can be modified, for example: SET Y1E2

2、**Program Example:**

Ladder diagram:



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

Mnemonic	Function							
RST	Clear the contacts or the registers							
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	E0~E7/F0~F7
	×	√	√	√	√	√	√	√

1、**Explanations**: When the RST instruction is driven, the actions of the designated devices are:

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

2、**Program Example**:

Ladder diagram:

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

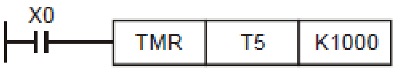
3.7 [TMR] , [ATMR], [CNT] , [DCNT]

Mnemonic	Function	
TMR	16-bit timer	
Operand	T-K	T0~T255，K0~K32,767
	T-D	T0~T255，D0~D11999

1、**Explanations**： When TMR instruction is executed, the designated coil of the timer will be On and the timer will start to time. When the set value in the timer is reached (present ⩾ set value).

2、**Program Example**

Ladder diagram:



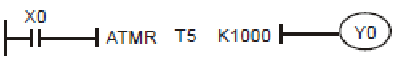
Instruction code:		Operation:
LD	X0	Loading in contact A of X0 T5 timer
TMR	**T5 K1000**	Set value in timer T5 as K1,000

Mnemonic	Function	
ATMR	16-bit contact type timer counter	
Operand	T-K	T0~T255，K0~K32,767
	T-D	T0~T255，D0~D11999

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

2、**Program Example:**

Ladder diagram:



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

Mnemonic	Function	
CNT	16-bit counter	
Operand	C-K	C0~C199，K0~K32,767
	C-D	C0~C199，D0~D11999

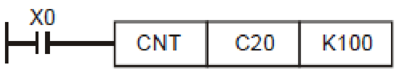
1、**Explanations**： When the CNT instruction goes from Off to On, the designated counter coil will be driven, and the present value,in the counter will plus 1. When the counting reaches the set value (present value = set value), the contact will be:

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

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

2、**Program Example：**

Ladder diagram:



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

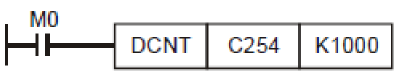
Mnemonic	Function	
DCNT	32-bit counter	
Operand	C-K	C200~C255，K-2,147,483,648~K2,147,483,647
	C-D	C200~C255，D0~D11999

1、**Explanations**：

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

2、**Program Example：**

Ladder diagram:



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

3.8 [MC/MCR] , [LDP], [LDF]

Mnemonic	Function
MC/MCR	Master control Start/Reset
Operand	N0~N7

1、 **Explanations:**

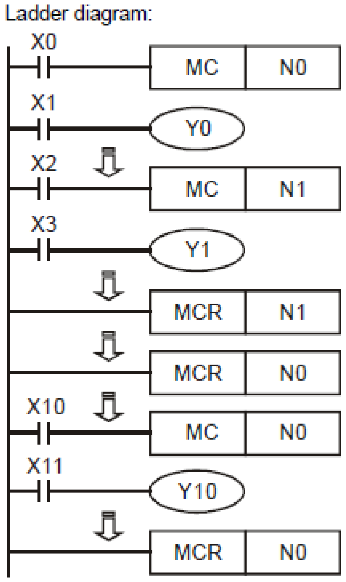
- MC is the main-control start instruction. When MC instruction is executed, the execution of instructions between MC and MCR will not be interrupted. When MC instruction is Off, the actions of the instructions between MC and MCR are:

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

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

2、 **Program Example**

Ladder diagram:



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

Mnemonic	Function						
LDP	Rising-edge detection operation						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Explanations:**

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

2、**Program Example**

Ladder diagram:

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

Mnemonic	Function						
LDF	Falling-edge detection operation						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Explanations:**

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

2、**Program Example**

Ladder diagram:

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

3.9 [ANDP] , [ANDF] , [ORP] , [ORF]

Mnemonic	Function						
ANDP	Rising-edge series connection						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Explanations:**

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

2、**Program Example**

Ladder diagram:

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

Mnemonic	Function						
ANDF	Falling-edge series connection						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Explanations:**

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

2、**Program Example:**

Ladder diagram:

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

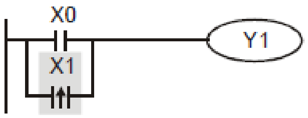
Mnemonic	Function						
ORP	Rising-edge parallel connection						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Explanations:**

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

2、**Program Example**

Ladder diagram:



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

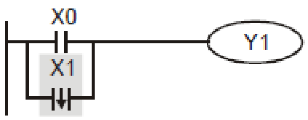
Mnemonic	Function						
ORF	Falling-edge parallel connection						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	√	√	√	√	√	√	×

1、**Explanations:**

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

2、**Program Example**

Ladder diagram:



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

3.10 [PLS] , [PLF]

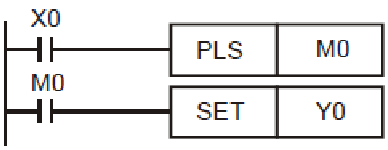
Mnemonic	Function						
PLS	Rising-edge output						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	×	√	√	×	×	×	×

1、**Explanations:**

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

2、**Program Example**

Ladder diagram: Instruction code: Operation:



LD	X0	Loading in A contact of X0
PLS	**M0**	M0 rising-edge output
LD	M0	Loading in contact A of M0

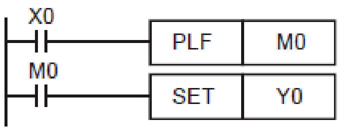
Mnemonic	Function						
PLF	Falling-edge output						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D11999
	×	√	√	×	×	×	×

1、**Explanations:**

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

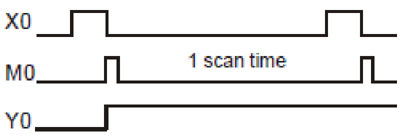
2、**Program Example**

Ladder diagram:



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

Timing Diagram:



3.11 [END] , [NOP] , [INV]

Mnemonic	Function
END	Program End
Operand	N/A

1、**Explanations:**

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

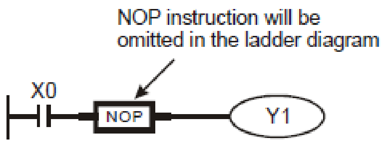
Mnemonic	Function
NOP	No operation
Operand	N/A

1、**Explanations:**

NOP instruction does not conduct any operations in the program; therefore, after the execution of NOP, the existing logical operation result will be kept. If you want to delete a certain instruction without altering the length of the program, you can use NOP instruction.

2、**Program Example:**

Ladder diagram:



Instruction code:		Operation:
LD	X0	Loading in B contact of X0
NOP		No operation
OUT	Y1	Driving Y1 coil

Mnemonic	Function
INV	Inverting Operation
Operand	N/A

1、**Explanations:**
The logical operation result before INV instruction will be inverted and stored in the accumulative register.

2、**Program Example:**

Ladder diagram:

Instruction code:		Operation:
LD	X0	Loading in A contact of X0
INV		Inverting the operation result
OUT	Y1	Driving Y1 coil

3.1 Pointer [P], Interruption Pointer [I]

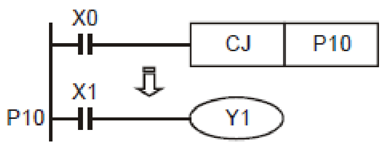
Mnemonic	Function
P	Pointer
Operand	P0~P255

1、**Explanations:**

Pointer P is used inZL 00 CJ and ZL01 CALL instructions. The use of P does not need to start from No. 0, and the No. of P cannot be repeated; otherwise, unexpected errors may occur.。

2、**Program Example:**

Ladder diagram:



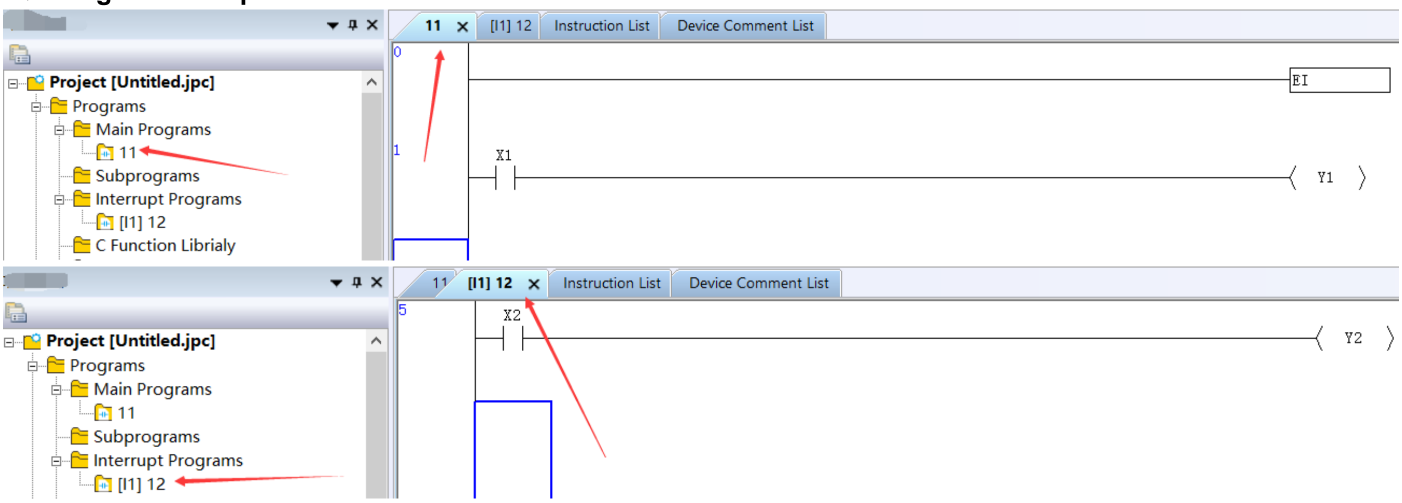
Instruction code:		Operation:
LD	X0	Loading in A contact of X0
CJ	P10	From instruction CJ to P10
:		
P10		Pointer P10
LD	X1	Loading in A contact of X1
OUT	Y1	Driving Y1 coil

Mnemonic	Function
I	Interruption program marker (I)
Operand	I00□, I10□, I20□, I30□, I40□, I50□, I6□□, I7□□, I8□□ I010, I020, I030, I040, I050, I060, I110, I120, I130, I140, I150, I160, I170, I180

1、**Explanations:**

A interruption program has to start with a interruption pointer (I□□□) and ends with 03 IRET. I instruction has to be used with 03 IRET, 04 EI, and 05 DI.

2、**Program Example:**



Instruction code:		Operation:
EI		Enabling interruption
LD	X1	Loading A contact of X1
OUT	Y1	Driving Y1 coil
:		
DI		Disabling interruption
:		
FEND		Main program ends
I001		Interruption pointer
LD	X2	Loading in A contact of X2
OUT	Y2	Driving Y2 coil
:		
IRET		Interruption return

4 Step Ladder Instructions

4.1 Step Ladder Instructions [STL], [RET]

Mnemonic	Function	Operand
STL	Step Transition Ladder Start	S0~S1023

1、 **Explanations:**

2、 STL Sn constructs a step. When STL instruction appears in the program, the program will enter a step ladder diagram status controlled by steps. The initial status has to start from S0 ~ S9. RET instruction indicates the end of a step ladder diagram starting from S0 ~ S9 and the bus returns to a normal ladder diagram instruction. SFC uses the step ladder diagram composed of STL/RET to complete the action of a circuit. The No. of S cannot be repeated.

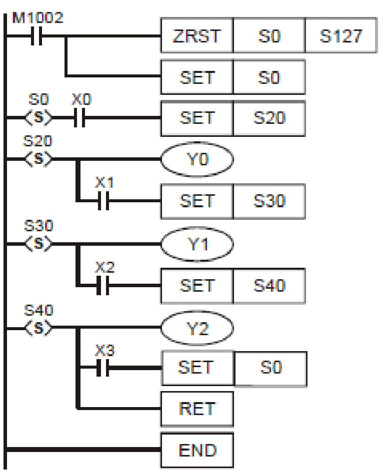
Mnemonic	Function	Operand
RET	Step Transition Ladder Return	N/A

1、 **Explanations:**

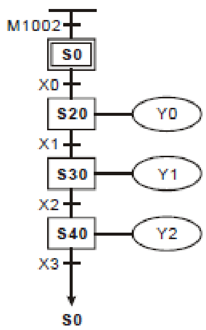
RET indicates the end of a step. There has to be a RET instruction in the end of a series of steps. One PLC program can be written in maximum 10 steps (S0 ~ S9) and every step should end with a RET.

2、 **Program Example:**

Ladder diagram:



SFC:



4.2 Step ladder instruction action description

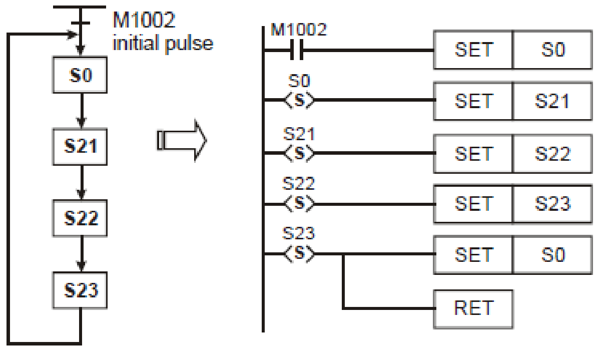
1、STL instruction:

STL instruction is used for designing the syntax of a sequential function chart (SFC), making the program designing similar to drawing a flow chart and allowing a more explicit and readable program. From the figure in the left hand side below, we can see very clearly the sequence to be designed, and we can convert the sequence into the step ladder diagram in the right hand side.

2、RET instruction:

RET instruction has to be written at the end of every step sequence, representing the end of a sequence. There can be more than one step sequence in a program. Therefore, we have to write in RET at the end of every step sequence. There is no limitation on the times of using RET which is used together with S0 ~ S9.

3、If there is no RET instruction at the end of a step sequence, errors will be detected.



4、Actions of Step Ladder:

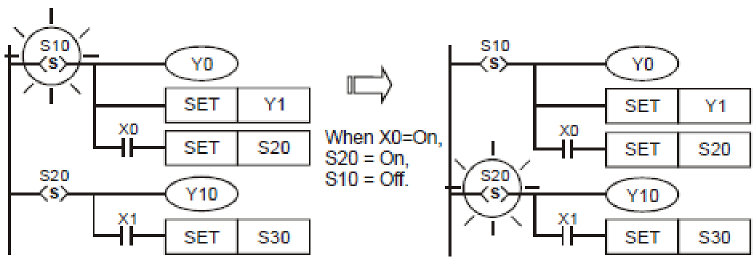
A step ladder is composed of many steps and every step controls an action in the sequence. The step ladder has to:

a) Drive the output coil

b) Designate the transition condition

c) Designate which step will take over the control from the current step

Example:



Explanation:

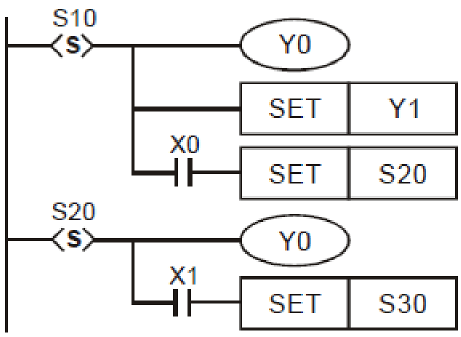
When S10 = On, Y0 and Y1 will be On. When X0 = On, S20 will be On and Y10 will be On. When S10=Off, Y0 will be Off and Y1 will be On.

5、Repeated Use of Output Coil:

- You can use output coils of the same No. in different steps.
- See the diagram in the right. There can be the same output device (Y0) among different statuses. Y0 will be On when S10 or S20 is On. Such as right diagram, there is the same output device Y0 in the different state. No matter S10 or S20 is On, Y0 will be On.
- Y0 will be Off when S10 is transferring to S20. After S20 is On, Y0 will output again. Therefore in this case, Y0

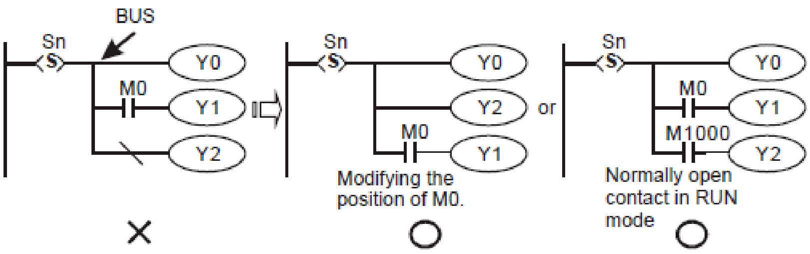
will be On when S10 or S20 is On.

- Normally in a ladder diagram, avoid repeated use of an output coil. The No. of output coil used by a step should also avoid being used when the step ladder diagram returns to a general ladder diagram.

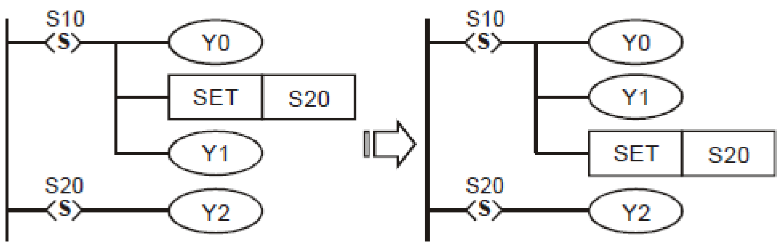


6、Cautions for Driving Output Point:

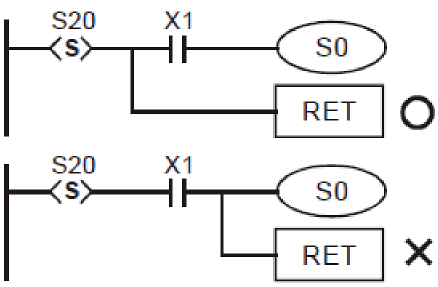
a. See the figure below. After the step point and once LD or LDI instructions are written into the second line, the bus will not be able to connect directly to the output coil, and errors will occur in the compilation of the ladder diagram. You have to correct the diagram into the diagram in the right hand side for a correct compilation.



b. The instruction used for transferring the step (SET S□ or OUT S□) can only be executed after all the relevant outputs and actions in the current status are completed. See the figure below. The executed results by the PLC are the same, but if there are many conditions or actions in S10, it is recommended that you modify the diagram in the left hand side into the diagram in the right hand side. SET S20 is only executed after all relevant outputs and actions are completed, which is a more explicit sequence.



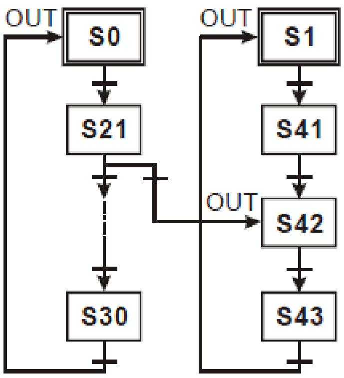
c. The RET instruction must be added after the step ladder program is completed, and RET must also be added after the STL, as shown in the figure below:



4.3 Step ladder programming

1、The first step in the SFC is called the "initial step", S0 ~ S9. Use the initial step as the start of a sequence and end a complete sequence with RET instruction.
2、If STL instruction is not in use, step S can be a general-purpose auxiliary relay.
3、When STL instruction is in use, the No. of step S cannot be repeated.
4、Types of sequences:
- Single sequence: There is only one sequence without alternative divergence, alternative convergence, simultaneous divergence and simultaneous convergence in a program.
- Complicated single sequence: There is only one sequence with alternative divergence, alternative convergence, simultaneous divergence and simultaneous convergence in a program.
- Multiple sequences: There are more than one sequence in a program, maximum 10 sequences, S0 ~ S9.

5、Separation of sequence: Multiple sequences are allowed to be written into the step ladder diagram.
- There are two sequences S0 and S1. The program writes in S0 ~ S30 first and S1 ~ S43 next.
- b) You can designate a step in the sequence to jump to any step in another sequence.
- When the condition below S21 is true, the sequence will jump to step S42 in sequence S1, which is called "separating the step".



6、Restrictions on diverging sequence:
- You can use maximum 8 diverged steps in a divergence sequence.
- You can use maximum 16 loops in multiple divergence sequences or in simultaneous sequences combined into one sequence.
- You can designate a step in the sequence to jump to any step in another sequence.

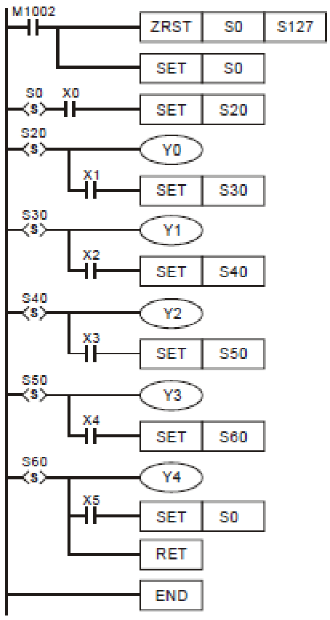
7、Reset of the step and the inhibiting output:
- Use ZRST instruction to reset a step to be Off

8、Latched step:
The On/Off status of the latched step will be memorized when the power of the PLC is switched off. When the PLC is re-powered, the status before the power-off will be recovered and the execution will resume. Please be aware of the area for the latched steps.

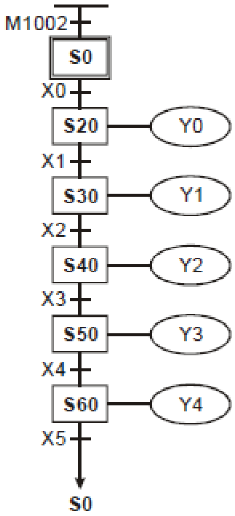
10、Types of Sequences
- Single Sequence: The basic type of sequences
  The first step in a step ladder diagram is called the initial step, which can be S0 ~ S9. The steps following the initial step are general steps, which can be S10 ~ S1023. If you are using IST instruction, S10 ~ S19 will become the steps for zero return.

- Single sequence without divergence and convergence: After a sequence is completed, the control power on the steps will be given to the initial step.

Step ladder diagram

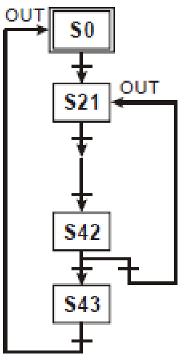


SFC:

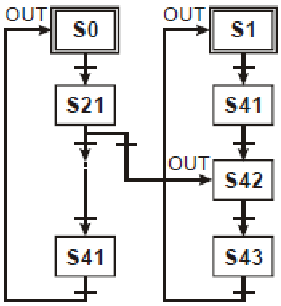


11、Jumping Sequence

1. The control power over the step is transferred to a certain step on top.

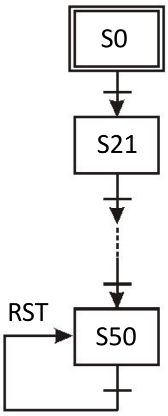


2. The control power over the step is transferred to the step in another sequence.



12、Reset Sequence

When the condition at S50 is true, S50 will be reset and the sequence will be completed at this time.

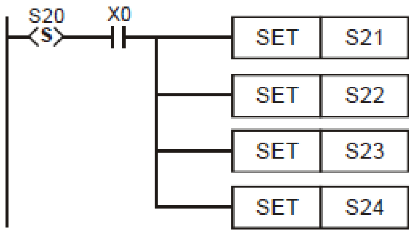


13、Complicated Single Sequence: Including simultaneous divergence, alternative divergence, simultaneous convergence and alternative convergence.

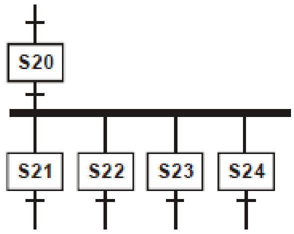
- Structure of simultaneous divergence:

  When the condition at the current step is true, the step can be transferred to many steps. See the diagrams below. When X0 = On, S20 will be simultaneously transferred to S21, S22, S23 and S24.

Ladder diagram:



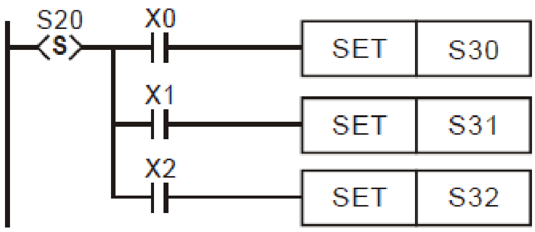
SFC:



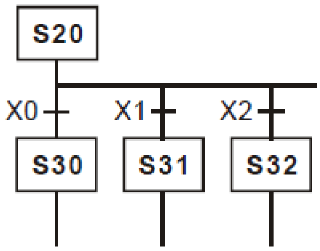
- Structure of alternative divergence:

  When the individual condition at the current status is true, the step will be transferred to another individual step. See the diagrams below. When X0 = On, S20 will be transferred to S30; when X1 = On, S20 will be transferred to S31; when X2 = On, S20 will be transferred to S32.

Ladder diagram:



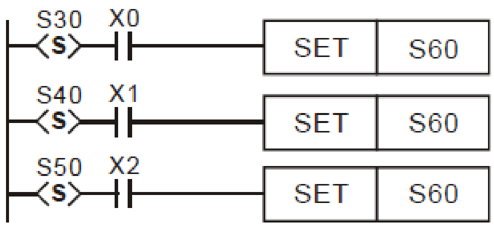
SFC:



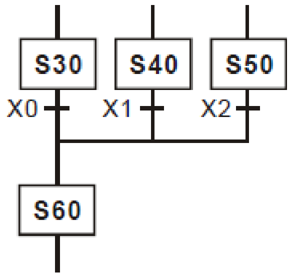
- Structure of alternative convergence:

  See the diagrams below. Depending on the condition of the input signal of which of S30, S40 and S50 becomes true first, the first one will be first transferred to S60.

Ladder diagram:

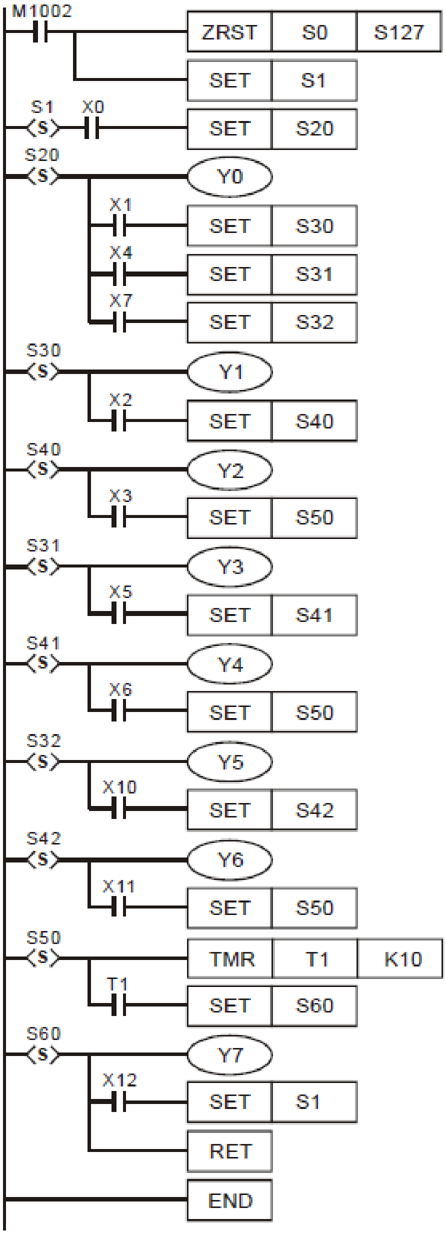


SFC:

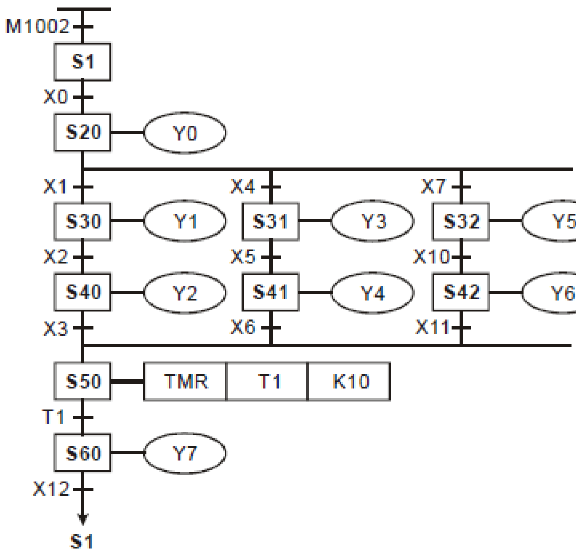


- Examples of alternative divergence & alternative convergence:

Ladder diagram:

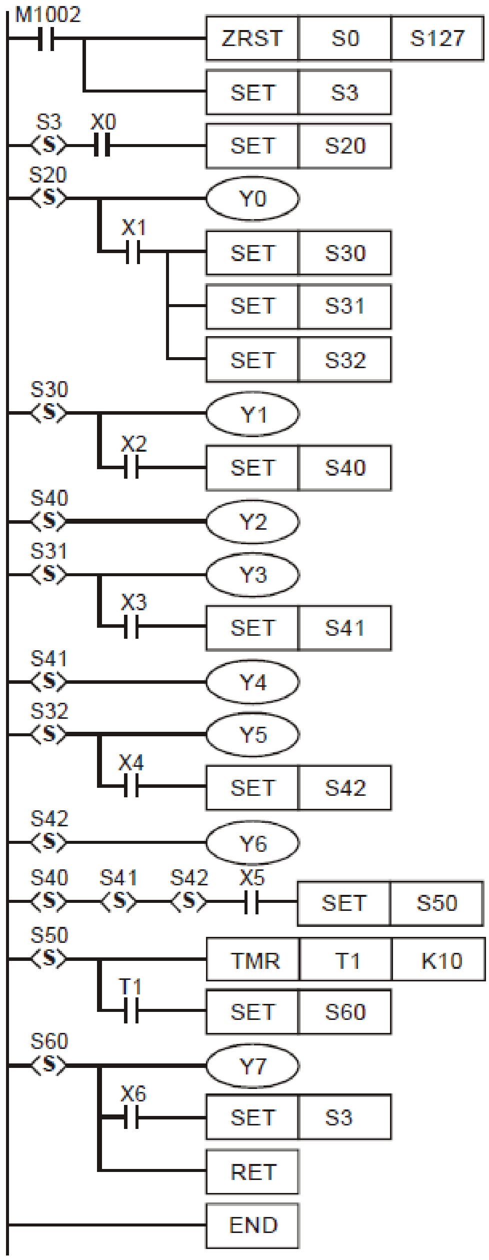


SFC:

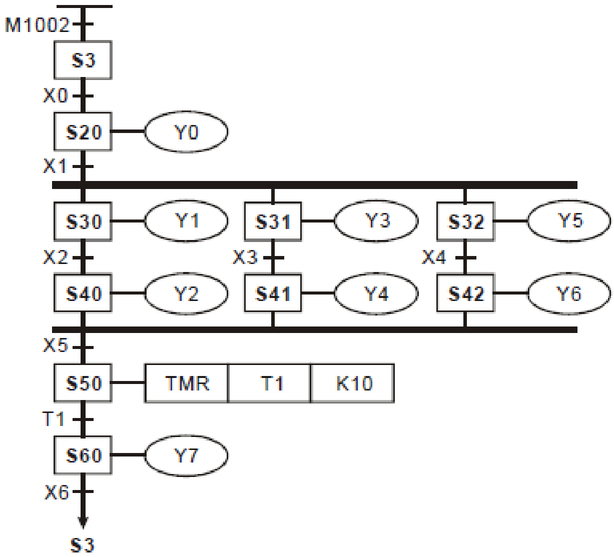


- Examples of simultaneous divergence & simultaneous convergence： X5

Ladder diagram:



SFC:

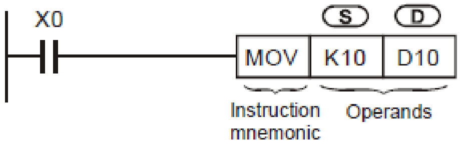


5 Categories & Use of Application Instructions

5.1 Composition of Application Instruction

1、An application instruction has two parts: the instruction and operands. The instruction part of the application instruction usually occupies 1 address (Step)，The instruction part of an application instruction usually occupies 1 step, and one operand occupies 2 or 4 steps depending on the instruction is a 16-bit or 32-bit one.
Instruction: The function of the instruction
Operands: Devices for processing the operations of the instruction

2、Input of application instruction:
Some application instructions are only composed of the instruction part (mnemonic), e.g. EI, DI, WDT…. Most application instructions are composed of the instruction part and many operands. Different application instructions designate different operands. Take MOV instruction for example:

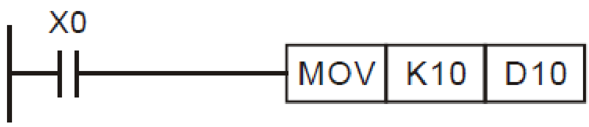


MOV instruction is to move the operand designated in S to the operand designated in D.

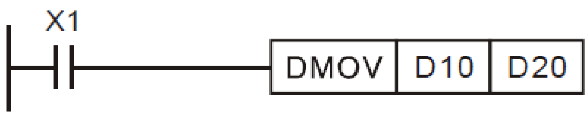
S	Source operand: If there are more than 1 source operands, they will be represented as S1, S2, ….
D	Destination operand: If there are more than 1 destination operands, they will be represented as D1, D2, ….
If the operand can only be constant K/H or a register, it will be represented as m, m1, m2, n, n1, n2, ….	

3、Length of operand (16-bit instruction or 32-bit instruction)
Depending on the contents in the operand, the length of an operand can be 16-bit or 32-bit. Therefore, a 16-bit instruction is for processing 16-bit operands, and 32-bit instruction is for processing 32-bit operands. The 32-bit instruction is indicated by adding a “D” before the 16-bit instruction.

16-bit MOV instruction：When X0 = On, K10 will be sent to D10.



32-bit DMOV instruction：When X1 = On, the content in (D11, D10) will be sent to (D21, D20).

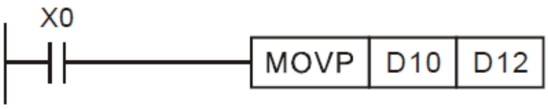


4、Continuous execution type / pulse execution type
In terms of command execution mode, it can be divided into two types: "continuous execution type" and "pulse execution type". Since the execution time required when the instruction is not executed is relatively short, the use of pulse execution instructions in the program as much as possible can reduce the scan cycle. The instruction marked with "P" after the instruction is the pulse execution type instruction. Some

instructions use pulse execution in most applications, such as INC, DEC and shift-related instructions, so the ↑ mark is added to the upper right of the mark of each instruction to indicate that the instruction usually uses pulse execution.

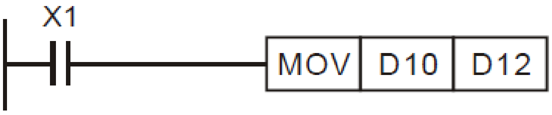
Pulse execution type:

When X0 changes from Off→On, the MOVP instruction is executed once, and the scan instruction is no longer executed, so it is called pulse execution instruction.



Continuous execution:

When X1=On each scan cycle, the MOV instruction is executed once, so it is called a continuous execution instruction.



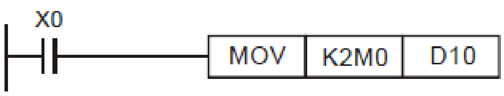
When the two conditional contacts X0 and X1=Off, the instruction will not be executed, and the content of the destination operand D will not change.

5、Designation of operands

- Bit devices X, Y, M, and S can be combined into word device, storing values and data for operaions in the form of KnX, KnY, KnM and KnS in an application instruction.
- Data register D, timer T, counter C and index register E, F are designated by general operands.
- A data register is usually in 16 bits, i.e. of the length of 1 register D. A designated 32-bit data register refers to 2 consecutive register Ds.
- If an operand of a 32-bit instruction designates D0, the 32-bit data register composed of (D1, D0) will be occupied. D1 is the higher 16 bits; D0 is the lower 16 bits. The same rule also apply to timer T, 16-bit timers and C0 ~ C199.
- When the 32-bit counters C200 ~ C254 are used as data registers, they can only be designataed by the operands of 32-bit instructions.

6、Format of operand

- X, Y, M, and S can only On/Off a single point and are defined as bit devices.
- 16-bit (or 32-bit) devices T, C, D, and registers E, F are defined as word devices.
- You can place Kn (n = 1 refers to 4 bits. For 16-bit instruction, n = K1 ~ K4; for 32-bit instruction, n = K1 ~ K8) before bit devices X, Y, M and S to make it a word device for performing word-device operations. For example, K1M0 refers to 8 bits, M0 ~ M7.
- When X0 = On, the contents in M0 ~ M7 will be moved to bit0 ~ 7 in D10 and bit8 ~ 15 will be set to "0".



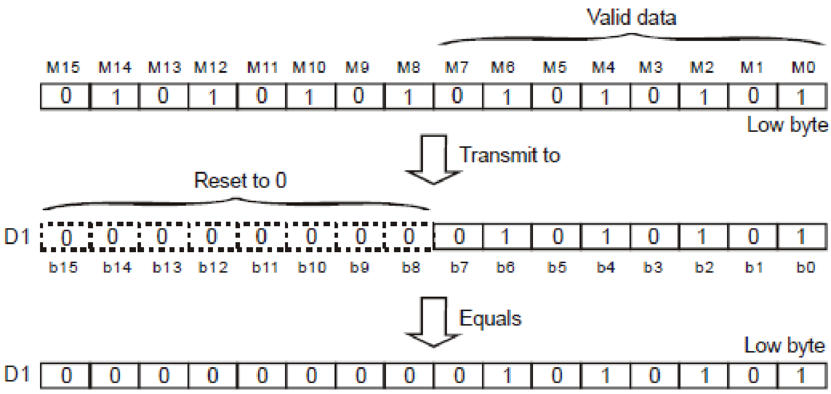
7、Data processing of word devices combined from bit devices

16-bit instruction	
Designated value： K-32,768~K32,767	
Values for designated K1 ~ K4：	
K1 （4 bits))	0~15
K2 （8 bits))	0~255
K3 （12 bits))	0~4,095
K4 （16 bits))	-32,768~+32,767
32-bit instruction	
Designated value： K-2,147,483,648~K2,147,483,647	
Values for designated K1 ~ K8	
K1 （4 bits))	0~15
K2 （8 bits))	0~255
K3 （12 bits))	0~4,095
K4 （16 bits))	0~65,535
K5 （20 bits))	0~1,048,575
K6 （24 bits))	0~167,772,165
K7 （28 bits))	0~268,435,455
K8 （32 bits))	-2,147,483,648~+2,147,483,647

5.2 Handling of Numeric Values

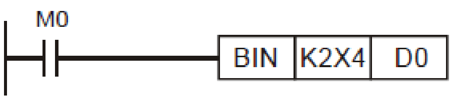
1、Devices only with On/Off status are called bit devices, e.g. X, Y, M and S. Devices used exclusively for storing numeric values are called word devices, e.g. T, C, D, E and F. Bit device plus a specific bit device (place a digit before the bit device in Kn) can be used in the operand of an application instruction in the form of numeric value.

2、n = K1 ~ K4 for a 16-bit value; n = K1 ~ K8 for a 32-bit value. For example, K2M0 refers to an 8-bit value composed of M0 ~ M7.



3、K1M0, K2M0, and K3M0 are transmitted to 16-bit registers and the vacant high bits will be filled in "0". The same rule applied to when K1M0, K2M0, K3M0, K4M0, K5M0, K6M0, and K7M0 are transmitted to 32-bit registers and the vacant high bits will be filled in "0".

4、In the 16-bit (or 32-bit) operation, if the contents of the operand are designated as bit devices K1 ~ K3 (or K4 ~ K7), the vacant high bits will be regarded as "0". Therefore, the operation is a positive-value one. The BCD value composed of X4 ~ X13 will be converted to BIN value and sent to D0.



5、You can choose any No. for bit devices, but please make the 1s digit of X and Y "0", e.g. X0, X10, X20, …Y0, Y10…, and the 1s digit of M and S "8's multiple" ("0" is still the best choice), e.g. M0, M10, M20….

6、Designating continuous device No.

Take data register D for example, continuous D refers to D0, D1, D2, D3, D4….

For bit devices with specifically designated digit, continuous No. refers to:

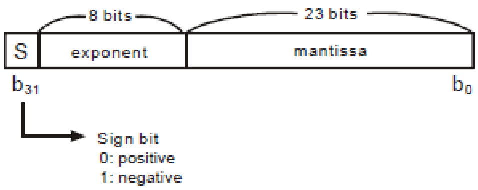
K1X0	K1X4	K1X10	K1X14…
K2Y0	K2Y10	K2Y20	K2X30…
K3M0	K3M12	K3M24	K3M36…
K4S0	K4S16	K4S32	K4S48…

Please follow the No. in the table and do not skip No. in case confusion may occur. In addition, if you use K4Y0 in the 32-bit operation, the higher 16 bits will be regarded as "0". For 32-bit data, please use K8Y0.

The operations in PLC are conducted in BIN integers. When the integer performs division, e.g. 40 ÷ 3 = 13 and the remainder is 1. When the integer performs square root operations, the decimal point will be left out. Use decimal point operation instructions to obtain the decimal point.

7、Binary Floating Point

PLC represents floating points in 32 bits, following the IEEE754 standard:



The expressible size is:

$$(-1)^S \times 2^{E-B} \times 1.M \text{, in which B = 127}$$

Therefore, the range for the 32-bit floating point is $\pm 2^{-126} \sim \pm 2^{+128}$, i.e. $\pm 1.1755 \times 10^{-38} \sim \pm 3.4028 \times 10^{+38}$

Example 1: Representing "23" in 32-bit floating point

Step 1: Convert "23" into a binary value: 23.0=10111

Step 2: Normalize the binary value: 10111=1.0111 × 24, in which 0111 is mantissa and 4 is exponent.

Step 3: Obtain the exponent:

∵ E-B=4 →E-127=4

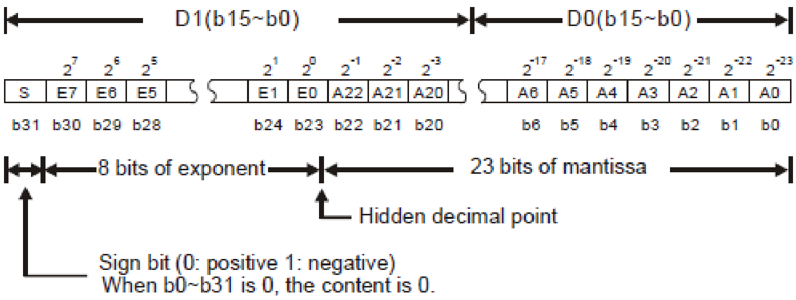
∴ E=131=100000112

Step 4: Combine the sign bit, exponent and mantissa into a floating point

0 10000011 01110000000000000000000$_2$=41B80000$_{16}$

Example 2: Representing "-23.0" in 32-bit floating point

The steps required are the same as those in Example 1. The only difference is you have to alter the sign bit into "1" . PLC uses registers of 2 continuous No. to combine into a 32-bit floating point. For example, we use registers (D1, D0) for storing a binary floating point as below:



8、Decimal Floating Point

- Since the binary floating point are not very user-friendly, we can convert it into a decimal floating point for use. Please be noted that the decimal point operation in PLC is still in binary floating point.
- The decimal floating point is represented by 2 continuous registers. The register of smaller No. is for the constant。

while the register of bigger No. is for the exponent.

Example: Storing a decimal floating point in registers (D1, D0)

Decimal floating point = [constant D0] × 10 $^{[\text{exponent D1}]}$

Constant D0 = ±1,000 ~ ±9,999

Exponent D1 = -41 ~ +35

The constant 100 does not exist in D0 due to 100 is represented as 1,000 × 10^{-1}. The range of decimal floating point is ±1175 × 10^{-41} ~ ±3402×10^{+35}.

9、The decimal floating point can be used in the following instructions:

D EBCD: Converting binary floating point to decimal floating point

D EBIN: Converting decimal floating point to binary floating point

5.3 E, F Index Register Modification

- The index registers are 16-it registers. There are 16 points E0 ~ E7 and F0 ~ F7

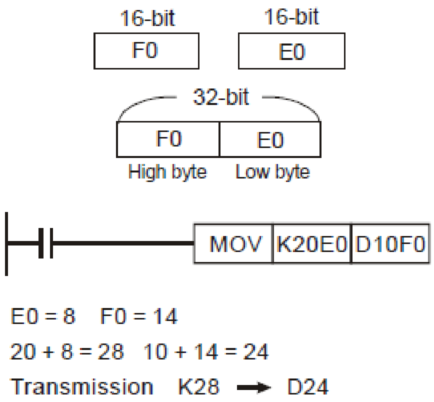
1、E and F index registers are 16-bit data registers. They can be read and written.

2、If you need a 32-bit register, you have to designate E. In this case, F will be covered up by E and cannot be used; otherwise, the contents in E may become incorrect. (We recommend you use MOVP instruction to reset the contents in D to 0 when the PLC is switched on.)

3、Combination of E and F when you designate a 32-bit index register: (E0, F0), (E1, F1), (E2, F2), ... (E7, F7)

See the diagram in the left hand side. E, F index register modification refers to the content in the operand changes with the contents in E and F.

For example, E0 = 8 and K20E0 represents constant K28 (20 + 8). When the condition is true, constant K28 will be transmitted to register D24.



3、 Devices modifiable MPU: P, I, X, Y, M, S, K, H, KnX, KnY, KnM, KnS, T, C, D

If you need to modify device P, I, X, Y, M, S, KnX, KnY, KnM, KnS, T, C and D by E, F, you have to select a 16-bit register, i.e. you can designate E or F. To modify constant K and H in a 32-bit instruction, you have to select a 32-bit register, i.e. you have to designate E.

6 Application Instructions ZL 00-49

6.1 (ZL 00-09) Program flow instructions

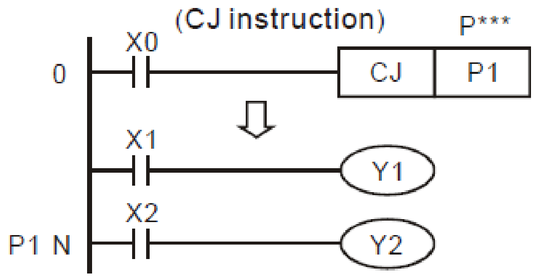
<table>
<tr><td rowspan="2">ZL 00</td><td></td><td rowspan="2">CJ</td><td></td><td rowspan="2">S</td><td colspan="11" rowspan="2">Conditional Jump 16-bit</td></tr>
<tr><td></td><td>p</td></tr>
<tr><td rowspan="3"></td><td colspan="3">Bit device</td><td colspan="11">Word device</td><td rowspan="3">CJ, CJP: 3 steps</td></tr>
<tr><td>X</td><td>Y</td><td>M</td><td>S</td><td>K</td><td>H</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>T</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td colspan="15">● Operands: S: The destination pointer of conditional jump
P can be modified by index register E, F.</td></tr>
</table>

1、Explanations:

- S: The destination pointer of conditional jump.
- When the user does not wish a particular part of PLC program in order to shorten the scan time and execute dual outputs, CJ instruction or CJP instruction can be adopted.。
- When the program designated by pointer P is prior to CJ instruction, WDT timeout will occur and PLC will stop running. Please use it carefully.
- CJ instruction can designate the same pointer P repeatedly. However, CJ and CALL cannot designate the same pointer P; otherwise an error will occur.
- Actions of all devices while conditional jumping is being executed.
  a. Y, M and S remain their previous status before the conditional jump takes place.
  b. Timer 10ms and 100ms that is executing stops.
  c. Timer T192 ~ T199 that execute the subroutine program will continue and the output contact executes normally.
  d. The high-speed counter that is executing the counting continues counting and the output contact executes normally.
  e. The ordinary counters stop executing.
  f. If the "reset instruction" of the timer is executed before the conditional jump, the device will still be in the reset status while conditional jumping is being executed.
  g. Ordinary application instructions are not executed.
  h. The application instructions that are being executed, i.e. ZL 53 DHSCS, ZL 54 DHSCR, ZL 55 DHSZ, ZL 56 SPD, ZL 57 PLSY, ZL 58 PWM, ZL 59 PLSR, ZL 157 PLSV, ZL 158 DRVI, ZL 159 DRVA, continue being executed.

2、Program Example 1:

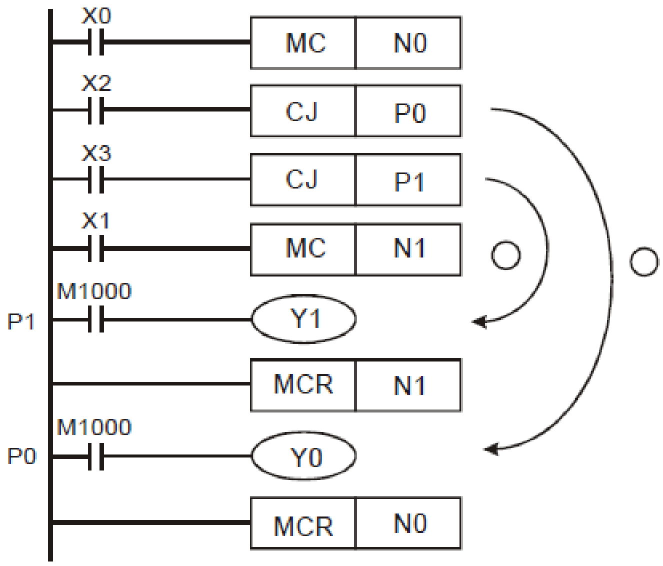
- When X0 = On, the program automatically jumps from address 0 to N (the designated label P1) and keeps its execution. The addresses between 0 and N will not be executed.
- When X0 = Off, as an ordinary program, the program keeps on executing from address 0. CJ instruction will not be executed at this time.



3、Program Example 2:

- CJ instruction can be used in the following 5 conditions between MC and MCR instructions:
  - a) Without MC ~ MCR.
  - b) From without MC to within MC. Valid in the loop P1 as shown in the figure below.
  - c) In the same level N, inside of MC~MCR.
  - d) From within MC to without MCR.
  - e) Jumping from this MC ~ MCR to another MC ~ MCR

(f) When MC instruction is executed, PLC will push the status of the switch contact into the self-defined stack in PLC. The stack will be controlled by the PLC, and the user cannot change it. When MCR instruction is executed, PLC will obtain the previous status of the switch contact from the top layer of the stack. Under the conditions as stated in b), d) and e), the times of pushing-in and obtaining stack may be different. In this case, the maximum stack available to be pushed in is 8 and the obtaining of stacks cannot resume once the stack becomes empty. Thus, when using CALL or CJ instructions, the user has to be aware of the pushing-in and obtaining of stacks.



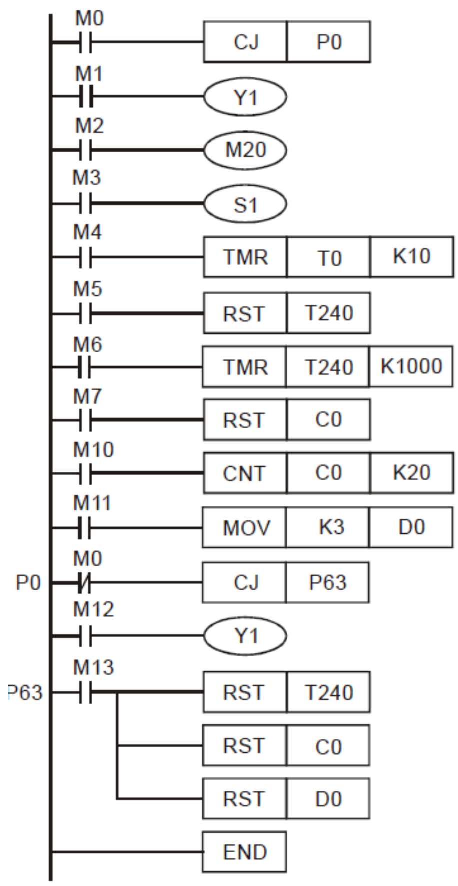
4、Program Example 3:

The states of each device:

Device	Contact state before CJ is executed	Contact state when CJ is being executed	Output coil state when CJ is being executed
Y、M、S	M1、M2、M3 Off	M1, M2, M3 Off→On	Y1 *1、M20、S1 Off
	M1、M2、M3 On	M1, M2, M3 Onf→Off	Y1 *1、M20、S1 On
10、100ms Timer	M4 Off	M4 Off →On	Timer T0 is not enabled.
	M4 On	M4 On →Off	Timer T0 immediately stops and is latched. M0 On→

			Off, T0 is reset as 0.
10、100ms Timer (accumulative)	M6 Off	M6 Off →On	Timer T240 is not enabled.
	M6 On	M6 On →Off	Once the timer function is enabled and when met with CJ instruction, all accumulative timers will stop timing and stay latched. M0 On →Off. T240 remains unchanged.
C0~C234	M7 、M10 Off	M10 On/Off trigger	Counter does not count.
	M7 Off、M10 On/Off trigger	M10 On/Off trigger	Counter C0 stops counting and stays latched. After M0 goes Off, C0 resumes its counting.
Application instruction	M11 Off	M11 Off →On	Application instructions are not executed.
	M11 On	M11 On →Off	The skipped application instructions are not executed, butZL59、ZL157~159 keep being executed.

*1: Y1 is a dual output. When M0 is Off, M1 will control Y1. When M0 is On, M12 will control Y1.
*2:When the high-speed counters (C235 ~ C255) are driven and encounter the execution of CJ instruction, thecounting will resume, as well as the action of the output points.。



ZL 01		CALL	P	S	Call Subroutine

Bit device				Word device											
X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CALL, CALLP: 3 steps 16 bits

- Operand S can designate P.
- P can be modified by index register E, F.

1、Explanations:

- S: pointer to call subroutine.
- Operand S can designate P0 ~ P255.
- Edit the subroutine designated by the pointer after FEND instruction.
- The number of pointer P, when used by CALL, cannot be the same as the number designated by CJ instruction.
- If only CALL instruction is in use, it can call subroutines of the same pointer number with no limit on times.
- Subroutine can be nested for 5 levels including the initial CALL instruction. (If entering the sixth level, the subroutine won' t be executed.)

ZL 02		SRET			Subroutine Return

Bit device				Word device											
X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SRET: 1 steps 16-bit

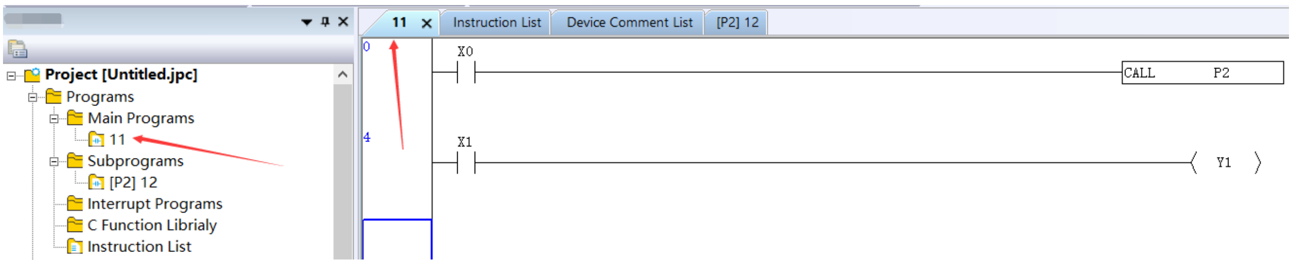
- No operand. No contact to drive the instruction is required.

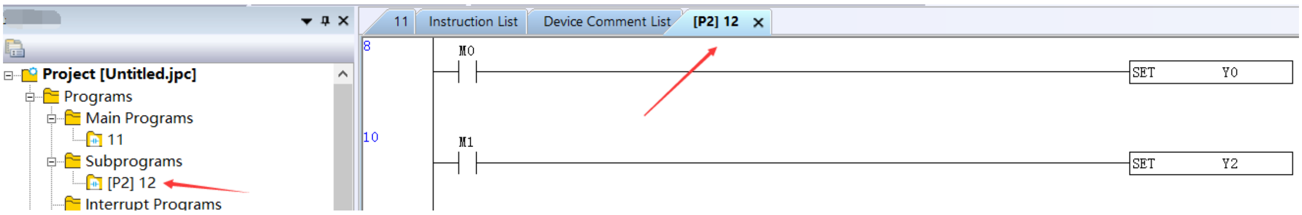
1、Explanations:

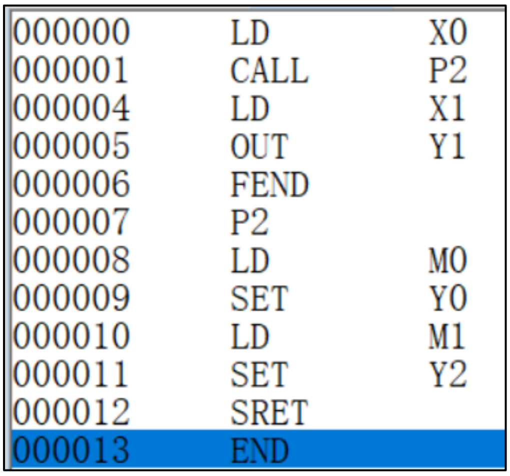
- The subroutine will return to main program by SRET after the termination of subroutine and execute the sequence program located at the next step to the CALL instruction.

2、Program Example 1:

- When X0 = On, CALL instruction is executed and the program jumps to the subroutine designated by P2. When SRET instruction is executed, the program returns to address4 and continues its execution.
- There is no need to edit the FEND and SRET codes in the ladder diagram. After the compilation is passed, the instruction list will be automatically generated.







000000	LD	X0
000001	CALL	P2
000004	LD	X1
000005	OUT	Y1
000006	FEND	
000007	P2	
000008	LD	M0
000009	SET	Y0
000010	LD	M1
000011	SET	Y2
000012	SRET	
000013	END	

ZL 03		IRET			Interrupt Return

Bit device				Word device											
X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	IRET: 1 steps 16-bit
● No operand. No contact to drive the instruction is required.															

1、Explanations:

- Interruption return refers to interrupt the subroutine.
- After the interruption is over, returning to the main program from IRET to execute the next instruction where the program was interrupted.

ZL 04		EI			Enable Interrupts

Bit device				Word device											
X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	EI: 1 steps 16-bit
● No operand. No contact to drive the instruction is required. ● The pulse width of the interruption signal should be >200us.															

ZL 05		DI			Disable Interrupts

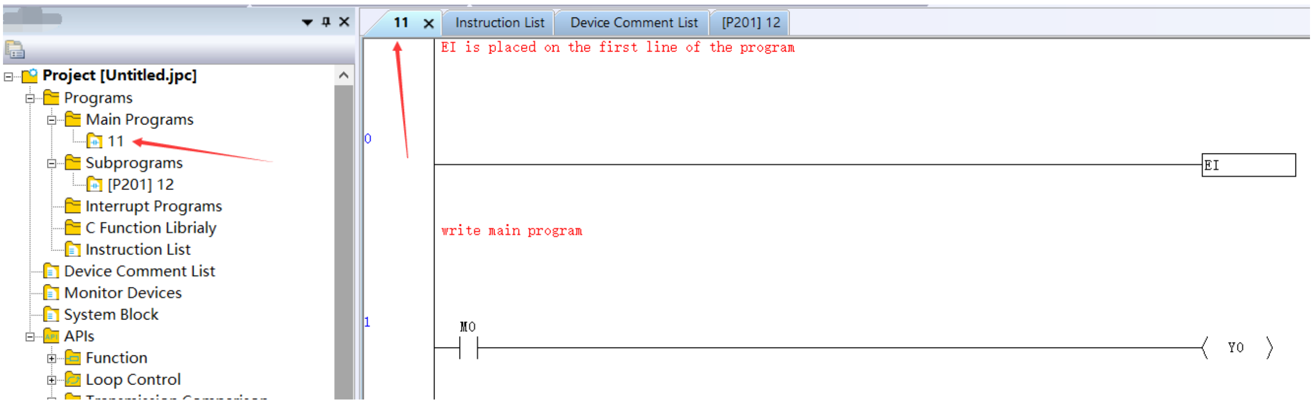
Bit device				Word device											
X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DI: 1 step 16-bit
● No operand. No contact to drive the instruction is required.															

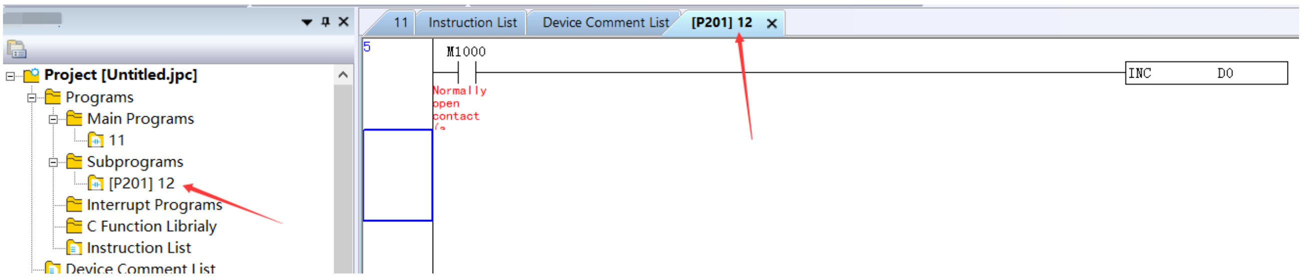
1、Explanations:

- EI instruction allows interrupting subroutine in the program, e.g. external interruption, timed interruption, and high-speed counter interruption.
- In the program, using interruption subroutine between EI and DI instruction is allowed. However, you can choose not to use DI instruction if there is no interruption-disabling section in the program.
- Pointer for interruption (I) must be placed after FEND instruction.
- Other interruptions are not allowed during the execution of interruption subroutine.
- When many interruptions occur, the priority is given to the firstly executed interruption. If several interruptions occur simultaneously, the priority is given to the interruption with the smaller pointer No.
- The interruption request occurring between DI and EI instructions that cannot be executed immediately will be memorized and will be executed in the area allowed for interruption.
- When using the interruption pointer, DO NOT repeatedly use the high-speed counter driven by the same X input contact.
- When immediate I/O is required during the interruption, write REF instruction in the program to update the status of I/O.

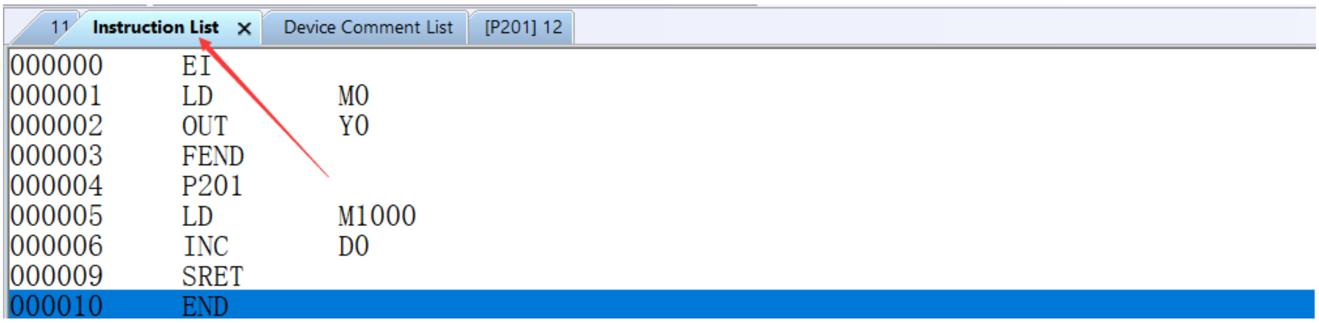
2、Program Example:

During the operation of PLC, when the program scans to the area between EI and DI instructions and X2 = Off→On, interruption subroutine A or B will be executed. When the subroutine executes to IRET, the program will return to the main program and resumes its execution.





The corresponding command list is as follows:


```
11   Instruction List   Device Comment List   [P201] 12
000000   EI
000001   LD     M0
000002   OUT    Y0
000003   FEND
000004   P201
000005   LD     M1000
000006   INC    D0
000009   SRET
000010   END
```

3、No. of interruption pointer I :

a. External interruptions: (I00□, X0), (I10□, X1), (I20□, X2), (I30□, X3), (I40□, X4), (I50□, X5), (I60□, X6), (I70□, X7), (I90□, X10), (I91□, X11), (I92□, X12), (I93□, X13), (I94□, X14), (I95□, X15), (I96□, X16), (I97□, X17) 16 points. (□ = 0 designates interruption in falling-edge, □ = 1 designates interruption in rising-edge).

b. Time interruptions: I6□□, I7□□, 2 points. (□□ = 2~99ms, time base = 1ms)

I8□□ 1 point. (□□ = 5~ 99ms, time base = 0.1ms)

c. High-speed counter interruptions: I010, I020, I030, I040, 1050, 1060 6 points. (used with ZL 53 DHSCS instruction to generate interruption signals)

d. When pulse output interruptions I110, I120 (triggered when pulse output is finished), I130, I140 (triggered when the first pulse output starts) are executed, the currently executed program is interrupted and jumps to the designated interruption subroutine.

e. Communication interruption: I150, I151, I153、I160, I161

f. The order for execution of interruption pointer I: external interruption, time interruption, high-speed counter interruption, pulse interruption, and communication interruption.

<table>
<tr><td>ZL
06</td><td></td><td colspan="2">FEND</td><td></td><td></td><td colspan="11">The End of The Main Program</td></tr>
<tr><td rowspan="3"></td><td colspan="4">Bit device</td><td colspan="11">Word device</td><td rowspan="3">FEND: 1 steps 16-bit</td></tr>
<tr><td>X</td><td>Y</td><td>M</td><td>S</td><td>K</td><td>H</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>T</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td colspan="15">● No contact to drive the instruction is required.</td></tr>
</table>

1、Explanations:

- This instruction denotes the end of the main program. It has the same function as that of END instruction when being executed by PLC.
- CALL must be written after FEND instruction and add SRET instruction in the end of its subroutine. Interruption program has to be written after FEND instruction and IRET must be added in the end of the service program.
- If several FEND instructions are in use, place the subroutine and interruption service programs between the final FEND and END instruction
- After CALL instruction is executed, executing FEND before SRET will result in errors in the program.
- After FOR instruction is executed, executing FEND before NEXT will result in errors in the program.

ZL 08	FOR	S	Start of a FOR-NEXT Loop

	Bit device				Word device											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FOR: 3 steps 16-bit
S					*	*	*	*	*	*	*	*	*	*	*	

- No contact to drive the instruction is required.

1、 Operands:

S: The number of repeated nested loops

ZL 09	NEXT		End of a FOR-NEXT Loop

	Bit device				Word device											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	NEXT: 1 steps 16-bit

- No operand. No contact to drive the instruction is required

1、 Explanations:

- FOR instruction indicates FOR ~ NEXT loops executing back and forth N times before escZLng for the next execution.
- N = K1 ~ K32,767. N is regarded as K1 when N ⩽ 1.
- When FOR~NEXT loops are not executed, the user can use the CJ instruction to escape the loops.
- Error will occur when:

1) NEXT instruction is before FOR instruction.

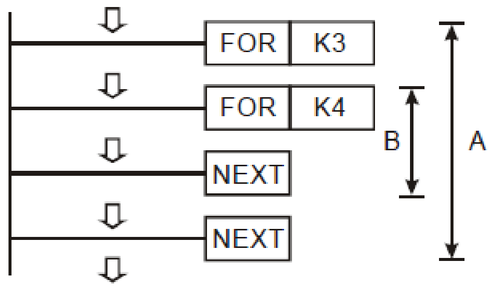
2) FOR instruction exists but NEXT instruction does not exist.

3) There is NEXT instruction after FEND or END instruction.

2、 The number of instructions between FOR ~ NEXT differs.

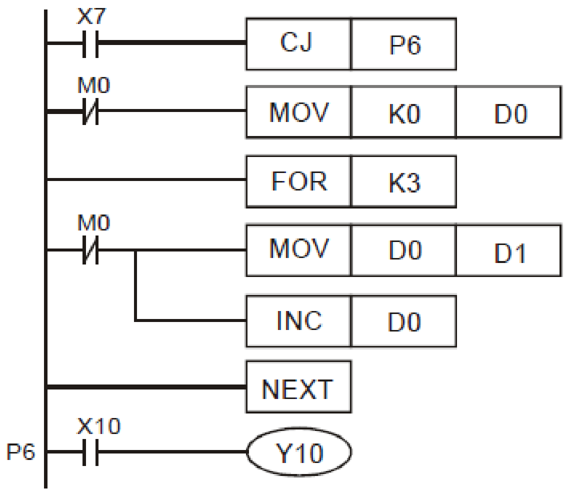
Program Example 1:

After program A has been executed for 3 times, it will resume its execution after NEXT instruction. Program B will be executed for 4 times whenever program A is executed once. Therefore, program B will be executed 3 × 4 = 12 times in total.



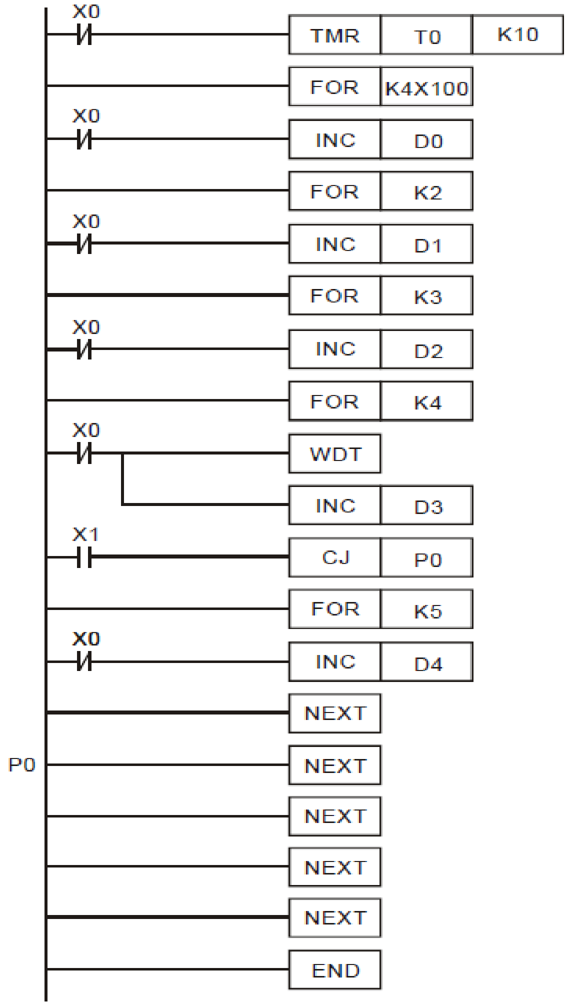
3、 Program Example 2:

When X7 = Off, PLC will execute the program between FOR ~ NEXT. When X7 = On, CJ instruction jumps to P6 and avoids executing the programs between FOR ~ NEXT.



4、Program Example 3:

When the programs between FOR ~ NEXT are not to be executed, the user can adopt CJ instruction for a jumping. When the most inner FOR ~ NEXT loop is in the status of X1 = On, CJ instruction executes jumping to P0 and skips the execution on P0.



6.2 (ZL 10-19) Transmission comparison

ZL 10	D	CMP		S1 S2 D												Compare
	Bit device				Word device											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*	*	CMP, CMPP: 7 steps 16-bit DCMP, DCMPP: 13 steps 32-bit
S2					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												

- If S1 and S2 are used in device F, only 16-bit instruction is applicable.
- Operand D occupies 3 consecutive devices

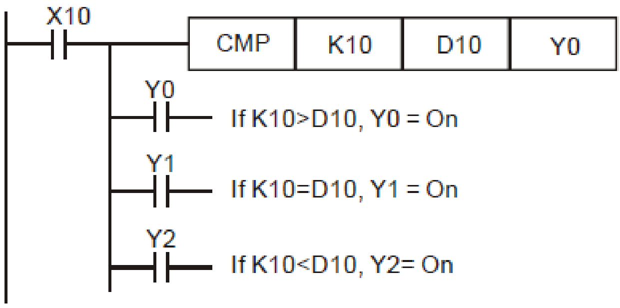
1、Explanations:

- S1: Comparison Value 1 S2: Comparison Value 2 D: Comparison result.
- The contents in S1 and S2 are compared and the result will be stored in D.
- The two comparison values are compared algebraically and the two values are signed binary values. When b15 = 1 in 16-bit instruction or b31 = 1 in 32-bit instruction, the comparison will regard the value as negative binary values.
- The designated device is Y0, then Y0, Y1 and Y2 are automatically occupied.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

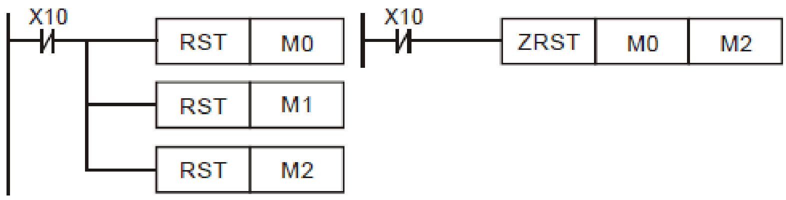
2、Program Example:

When X10 = On, CMP instruction will be executed and one of Y0, Y1, and Y2 will be On. When X10 = Off, CMP instruction will not be executed and Y0, Y1, and Y2 remain their status before X10 = Off.

- If the user need to obtain a comparison result with ≥ ≤, and ≠, make a series parallel connection between Y0 ~ Y2.



- To clear the comparison result, use RST or ZRST instruction.



ZL 11		ZCP		S1 S2 S D	Zone Compare
	D				

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
S					*	*	*	*	*	*	*	*	*	*	*
D		*	*	*											

- If S1, S2 and S are used in device F, only 16-bit instruction is applicable
- The content in S1 should be smaller than the content in S2
- Operand D occupies 3 consecutive devices

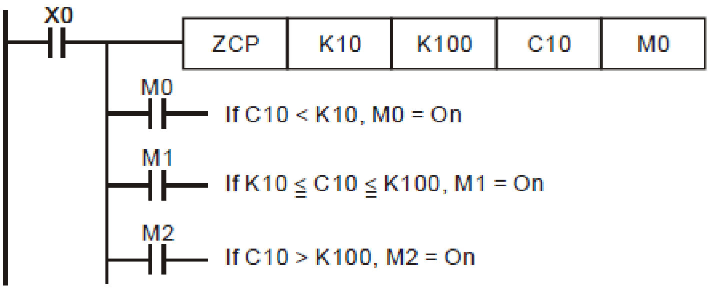
ZCP, ZCPP: 9 steps 16-bit
DZCP, DZCPP: 17 steps 32-bit

1、 Explanations

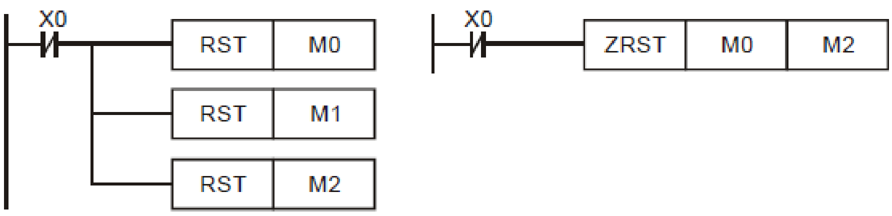
- S1: Lower bound of zone comparison S2: Upper bound of zone comparison S: Comparison value
- S is compared with its S1, S2 and the result is stored in D.
- When S1 > S2, the instruction performs comparison by using S1 as the lower/upper bound.
- The two comparison values are compared algebraically and the two values are signed binary values. When b15 = 1 in 16-bit instruction or b31 = 1 in 32-bit instruction, the comparison will regard the value as negative binary values.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、 Program Example:

- Designate device M0, and operand D automatically occupies M0, M1 and M.
- When X0 = On, ZCP instruction will be executed and one of M0, M1, and M2 will be On. When X0 = Off, ZCP instruction will not be executed and M0, M1, and M2 remain their status before X0 = Off.



- To clear the comparison result, use RST or ZRST instruction



ZL 12	D	MOV	P	S D											Move
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

MOV, MOVP: 5 steps 16-bit
DMOV, DMOVP: 9 steps 32-bit

1、Explanations:

- If S and D are used in device F, only 16-bit instruction is applicable.
- See the specifications of each model for their range of use.
- When this instruction is executed, the content of S will be moved directly to D. When this instruction is not executed, the content of D remains unchanged.
- If the operation result refers to a 32-bit output, (i.e. application instruction MUL and so on), and the user needs to move the present value in the 32-bit high-speed counter, DMOV instruction has to be adopted.

2、Program Example:

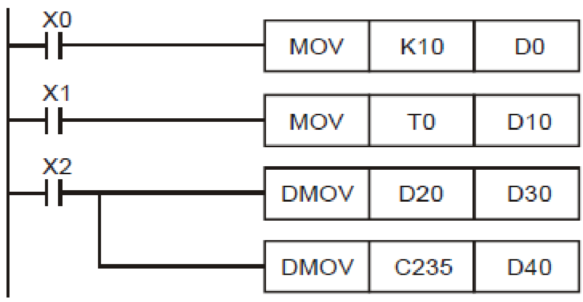
- MOV instruction has to be adopted in the moving of 16-bit data.

a) When X0 = Off, the content in D10 will remain unchanged. If X0 = On, the value K10 will be moved to D10 data register.

b) When X1 = Off, the content in D10 will remain unchanged. If X1 = On, the present value T0 will be moved to D10 data register.

- DMOV instruction has to be adopted in the moving of 32-bit data.

When X2 = Off, the content in (D31, D30) and (D41, D40) will remain unchanged. If X2 = On, the present value of (D21, D20) will be sent to (D31, D30) data register. Meanwhile, the present value of C235 will be moved to (D41, D40) data register.



ZL 13	SMOV					S m1 m2 D n										Shift Move
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	SMOV: 11 steps 16-bit
m1					*	*										
m2					*	*										
D								*	*	*	*	*	*	*	*	
n					*	*										

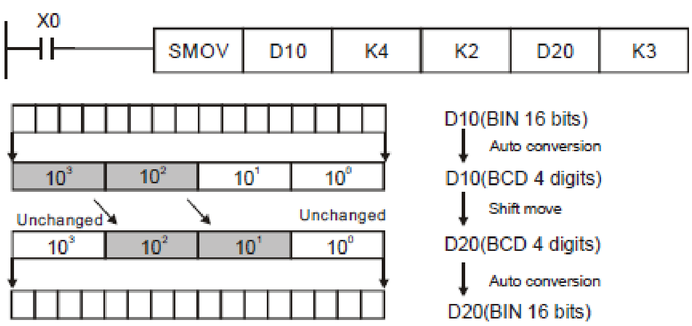
1、 Explanations:

- S: Source of data m1: Start digit to be moved of the source data m2: Number of digits (nibbles) to be moved of the source data D: Destination device n: Start digit of the destination position for the moved digits
- This instruction is able to re-allocate or combine data. When the instruction is executed, m2 digits of contents starting from digit m1 (from high digit to low digit) of S will be sent to m2 digits starting from digit n (from high digit to low digit) of D.
- Range: m1 = 1 ~ 4; m2 = 1 ~ m1; n = m2 ~ 4.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example 1:

- X0=On, specify the 4th digit of the decimal value of D10 (also known as the thousands digit) and start to transfer the contents of the 2 digits calculated from the low digit to the 3rd digit of the decimal value of D20 (that is, the hundreds digit) Digits) from the bottom 2 digits. The contents of 103 and 100 of D20 remain unchanged after this instruction is executed.
- When the BCD value exceeds the range of 0-9,999, the PLC judges it as an operation error and the instruction is not executed.

If D10=K1,234, D20=K5,678 before execution, after execution, D10 remains unchanged, D20=K5,128.

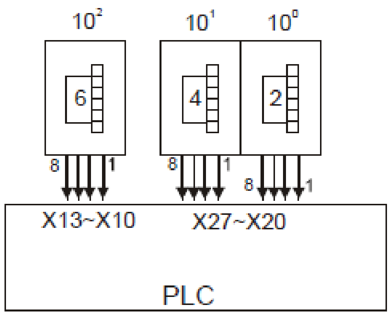


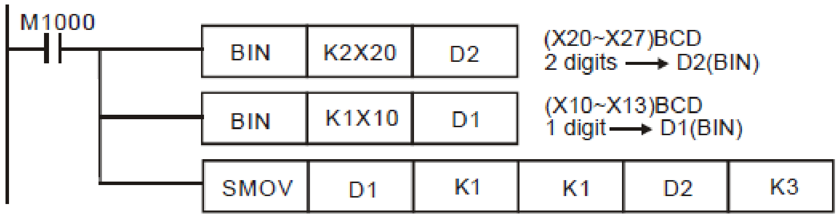
4、 Program Example 2

1) Dip switches connected to the input terminal of non-sequential numbers can be synthesized using this command.

2) Transfer the right 2 digits of the DIP switch to the right 2 digits of D2, and the left 1 digit of the DIP switch to the right 1 digit of D1.

3) Use the SMOV instruction to transfer the first digit of D1 to the third digit of D2 to combine the two sets of DIP switches into one group.





ZL 14		CML		S D	Compliment
	D				

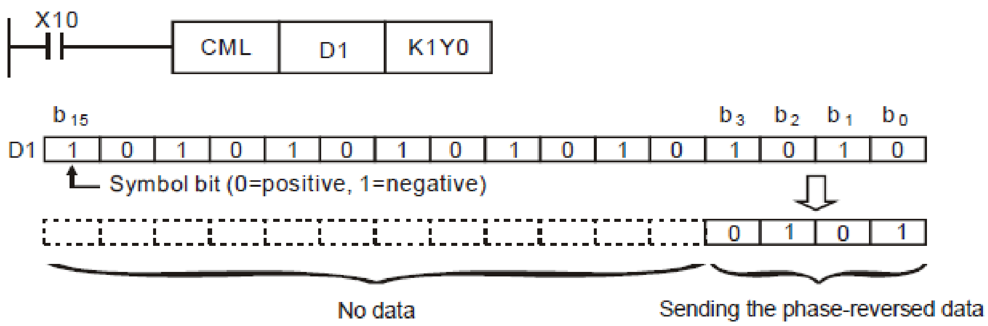
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CML: 5 steps 16-bit
S					*	*	*	*	*	*	*	*	*	*	*	DCML: 9 steps 32-bit
D								*	*	*	*	*	*	*	*	

1、 Explanations:

- S: Source of data D: Destination device.
- This instruction can be used for phase-reversed output.
- Reverse the phase (0→1, 1→0) of all the contents in S and send the contents to D. Given that the content is a constant K, K will be automatically converted into a BIN value.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

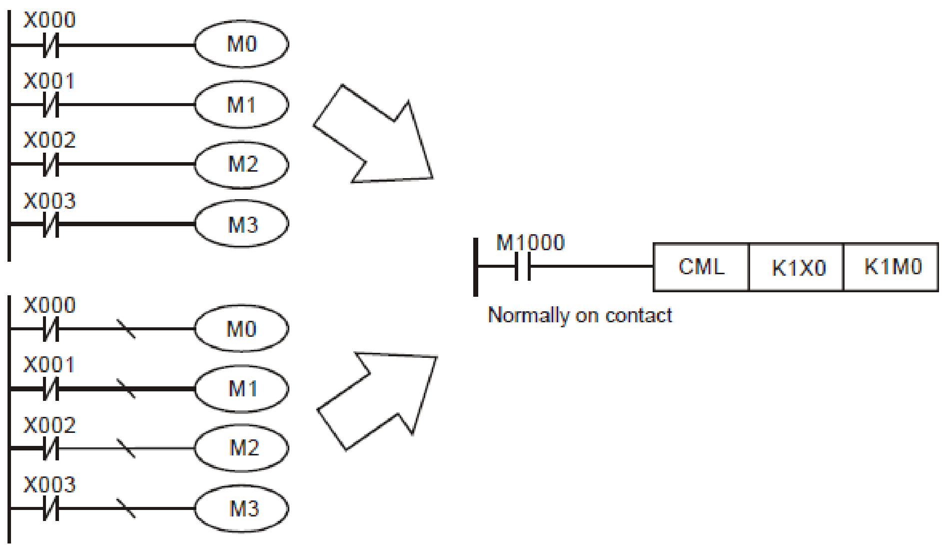
2、Program Example 1:

When X10 = On, b0 ~ b3 in D1 will be phase-reversed and send to Y0 ~ Y3.



3、 Program Example 2:

The loop below can also adopt CML instruction (see right below).



ZL 15	BMOV					S D n									Block Move
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S							*	*	*	*	*	*	*		
D								*	*	*	*	*	*		
n					*	*					*	*	*		

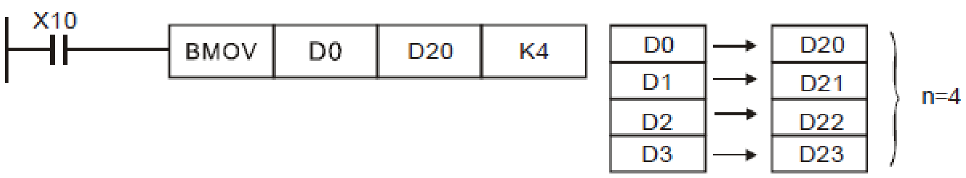
BMOV: 7 steps 16-bit

1、Explanations:

- S: Start of source devices D: Start of destination devices n: Number of data to be moved
- Range of n: 1 ~ 512
- The contents in n registers starting from the device designated by S will be moved to n registers starting from the device designated by D. If n exceeds the actual number of available source devices, only the devices that fall within the valid range will be used.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

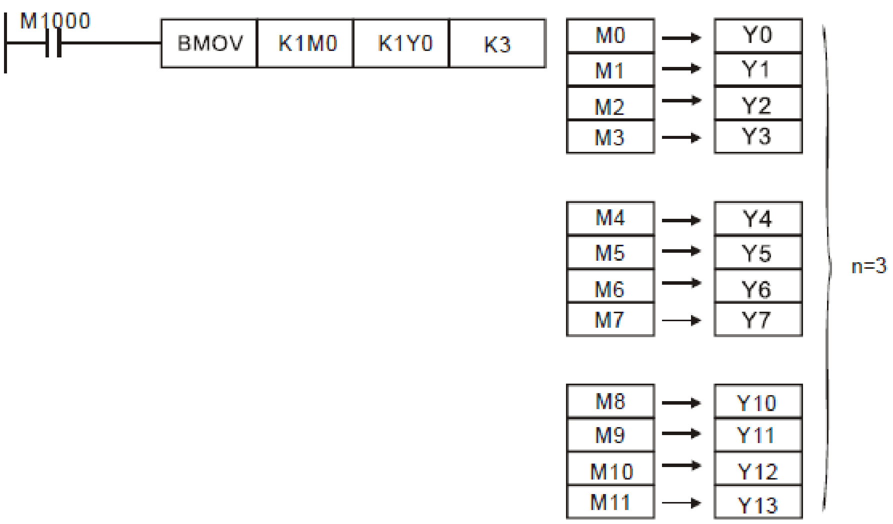
2、Program Example 1:

When X10 = On, the contents in registers D0 ~ D3 will be moved to the 4 registers D20 ~ D23.



3、Program Example 2:

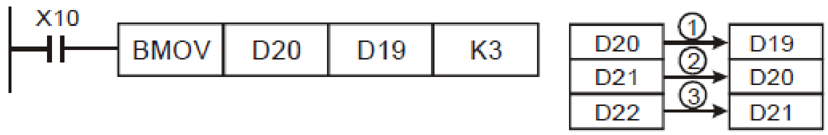
Assume the bit devices KnX, KnY, KnM and KnS are designated for moving, the number of digits of S and D has to be the same, i.e. their n has to be the same.



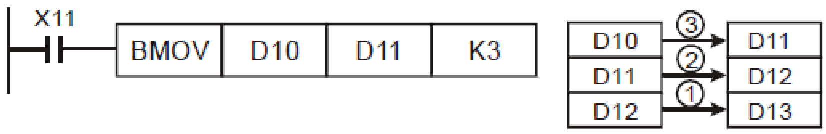
4、Program Example 3:

To avoid coincidence of the device numbers to be moved designated by the two operands and cause confusion, please be aware of the arrangement on the designated device numbers:

a. When S > D, the instruction is processed following the order:1→2→3



b. When S < D, the instruction is processed following the order: 3→2→1



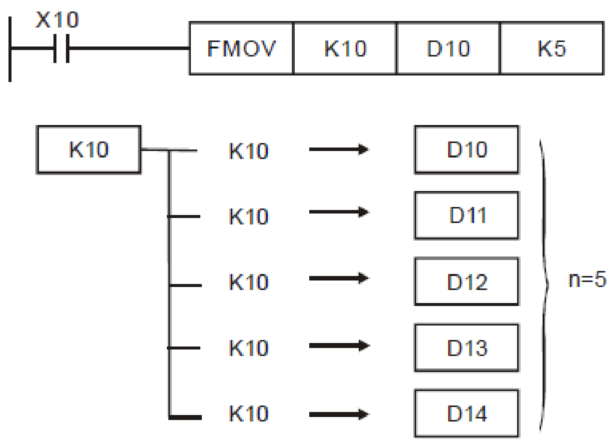
ZL 16	D	FMOV					S D n									Fill Move
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FMOV: 7 steps 16-bit DFMOV : 13 steps 32-bit
S					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*			
n					*	*					*	*	*			

1、Explanations

- S: Source of data D: Destination of data n: Number of data to be moved
- If S is used in device F, only 16-bit instruction is applicable
- The contents in n registers starting from the device designated by S will be moved to n registers starting from the device designated by D. If n exceeds the actual number of available source devices, only the devices that fall within the valid range will be used
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

When X10 = On, K10 will be moved to the 5 consecutive registers starting from D10.



ZL 17	D	XCH		D_1 D_2											Exchange
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
D_1								*	*	*	*	*	*	*	*
D_2								*	*	*	*	*	*	*	*

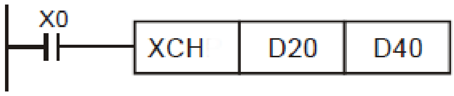
XCH: 5 steps 16-bit
DXCH: 9 steps 32-bit

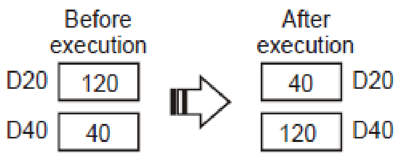
1、 Explanations:

- D1: Data to be exchanged 1. D2: Data to be exchanged 2
- If D1 and D2 are used in device F, only 16-bit instruction is applicable.
- The contents in the devices designated by D1 and D2 will exchange.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example 1:

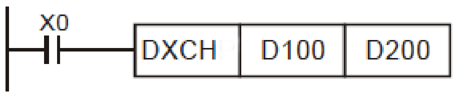
When X0 = Off→On, the contents in D20 and D40 exchange with each other.

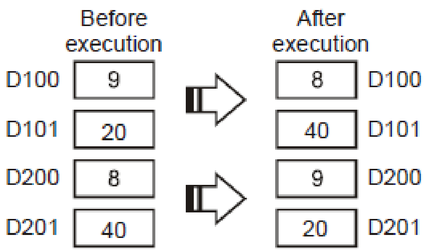




3、Program Example 2:

When X0 = Off → On, the contents in D100 and D200 exchange with each other.





ZL 18	D	BCD				S			D							Binary Coded Decimal
	Bit Devices				Word Devices											BCD,: 5 steps 16-bit DBCD: 9 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

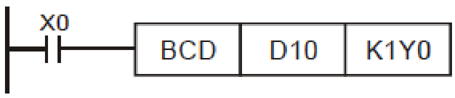
1、 Explanations:

- S: Source of data D: Conversion result.
- If S and D are used in device F, only 16-bit instruction is applicable.
- The four arithmetic operations and applications in PLC and the execution of INC and DEC instructions are performed in BIN format. Therefore, if the user needs to see the decimal value display, simply use this instruction to convert the BIN value into BCD value.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、 Program Example:

When X0 = On, the binary value of D10 will be converted into BCD value, and the 1s digit of the conversionresult will be stored in K1Y0 (Y0 ~ Y3, the 4 bit devices).

When D10 = 001E (hex) = 0030 (decimal), the execution result will be: Y0 ~ Y3 = 0000(BIN).



ZL 19	D	BIN		S D	BCD→BIN conversion

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S							*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

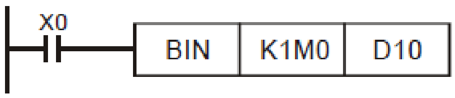
BIN: 5 steps 16bit
DBIN: 9 steps 32bit

1、 Explanations:

- S: Source of data D: Conversion result.
- If S and D are used in device F, only 16-bit instruction is applicable.
- The four arithmetic operations and applications in PLC and the execution of INC and DEC instructions are performed in BIN format. Therefore, if the user needs to see the decimal value display, simply use this instruction to convert the BIN value into BCD value.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

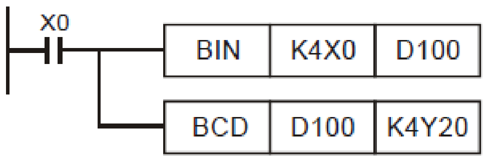
2、 Program Example:

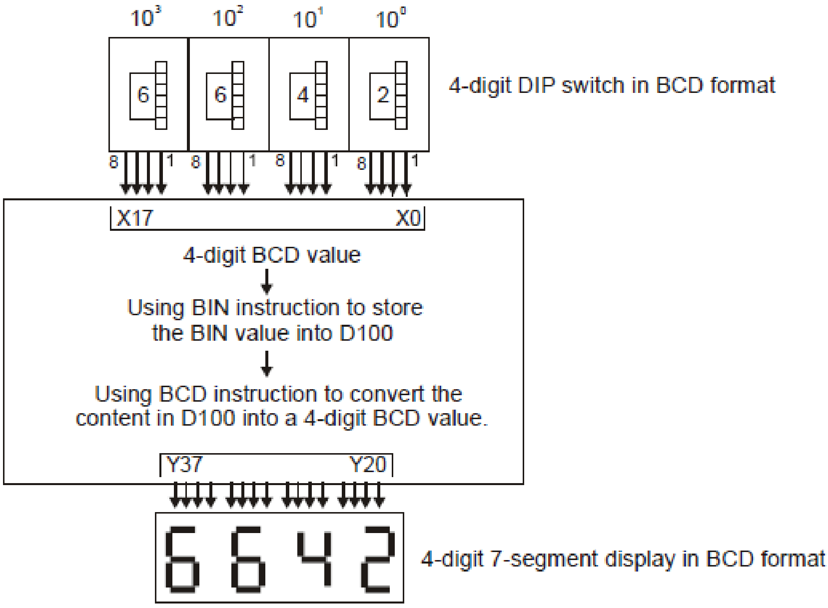
When X0 = On, the binary value of D10 will be converted into BCD value, and the 1s digit of the conversion result will be stored in K1Y0 (Y0 ~ Y3, the 4 bit devices).
When D10 = 001E (hex) = 0030 (decimal), the execution result will be: Y0 ~ Y3 = 0000(BIN).



Note: a. BCD and BIN instruction application instructions:

1) When the PLC wants to read a BCD type DIP switch from the outside, it must use the BIN command to convert the read data into a BIN value before storing it in the PLC.
2) When the PLC wants to display the internally stored data through an external BCD-type 7-segment display, it must use the BCD command to first convert the internal data to be displayed into BCD values and then send it to the 7-segment display.
3) When X0=On, convert the K4X0 BCD value to BIN value and transfer it to D100, then convert the BIN value of D100 to BCD value and transfer it to K4Y20.




10³
10²
10¹
10⁰
6
6
4
2
4-digit DIP switch in BCD format
8
1
X17
X0
4-digit BCD value
Using BIN instruction to store the BIN value into D100
Using BCD instruction to convert the content in D100 into a 4-digit BCD value.
Y37
Y20
6642
4-digit 7-segment display in BCD format

6.3 （ZL 20-29） Four logical operations

ZL 20	D	ADD	S1 S2 D	BIN Addition

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

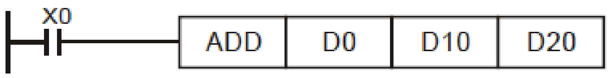
ADD: 7 steps 16-bit
DADD: 13 steps 32-bit

1、Explanations:

- S1: Summand S2: Addend D: Sum
- If S1, S2 and D are used in device F, only 16-bit instruction is applicable.
- This instruction adds S1 and S2 in BIN format and store the result in D.
- The highest bit is symbolic bit 0 (+) and 1 (-), which is suitable for algebraic addition, e.g. 3 + (-9) =-6
- To execute the pulse type, add the NP rising edge "↑" command before the command.

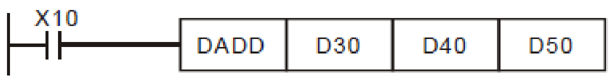
2、Program Example 1:

In 16-bit BIN addition：When X0 = On, the content in D0 will plus the content in D10 and the sum will be stored in D20.



3、Program Example 2:

In 32-bit BIN addition：When X10 = On, the content in (D31, D30) will plus the content in (D41, D40) and the sum will be stored in (D51, D50). D30, D40 and D50 are low 16-bit data; D31, D41 and D51 are high 16-bit data.



(D31, D30) + (D41, D40) = (D51, D50)

ZL 21	D	SUB		S1 S2 D											BIN Subtraction
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

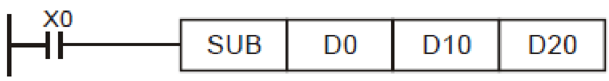
SUB: 7 steps 16-bit
DSUB: 13 steps 32-bit

1、Explanations:

- S1: Minuend S2: Subtrahend D: Remainder
- If S1, S2 and D are used in device F, only 16-bit instruction is applicable.
- This instruction subtracts S1 and S2 in BIN format and stores the result in D.
- The highest bit is symbolic bit 0 (+) and 1 (-), which is suitable for algebraic subtraction
- For flag operations of SUB instruction and the positive/negative sign of the value, see the explanations in ADD instruction on the previous page.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

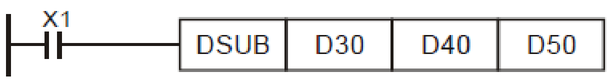
2、Program Example 1:

In 16-bit BIN subtraction: When X0 = On, the content in D0 will minus the content in D10 and the remainder will be stored in D20.



3、Program Example 2:

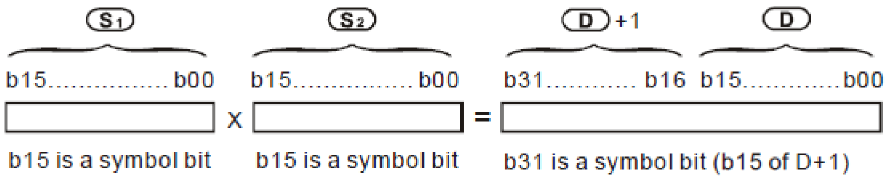
In 32-bit BIN subtraction: When X1 = On, the content in (D31, D30) will minus the content in (D41, D40) and the remainder will be stored in (D51, D50). D30, D40 and D50 are low 16-bit data; D31, D41 and D51 are high 16-bit data.



ZL 22	D	MUL		S1 S2 D											BIN Multiplication
	Bit Devices				Word Devices										MUL: 7 steps 16-bit DMUL: 13 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	

1、 Explanations:

- S1: Multiplicand S2: Multiplicator D: Product
- To execute the pulse type, add the NP rising edge " ↑ " command before the command.
- If S1 and S2 are used in device F, only 16-bit instruction is applicable.
- If D is used in device E, only 16-bit instruction is applicable
- In 16-bit instruction, D occupies 2 consecutive devices.
- In 32-bit instruction, D occupies 4 consecutive devices.
- This instruction multiplies S1 by S2 in BIN format and stores the result in D. Be careful with the positive/negative signs of S1, S2 and D when doing 16-bit and 32-bit operations.
- In 16-bit BIN multiplication:



Symbol bit = 0 refers to a positive value.

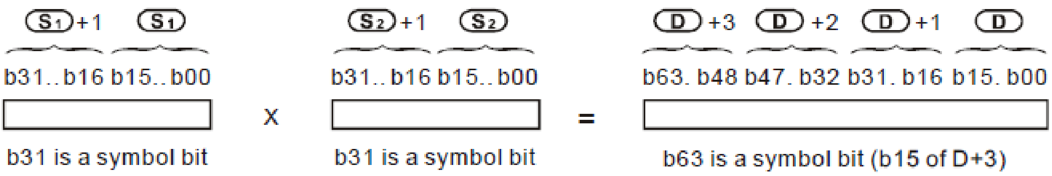
Symbol bit = 1 refers to a negative value.

16-bit value x 16-bit value = 32-bit value

When D serves as a bit device, it can designate K1 ~ K4 and construct a 16-bit result, occupying consecutive 2 groups of 16-bit data.

If the product of a 16-bit multiplication must be a 16-bit value (16-bit value x 16-bit value = 16-bit value), users have to use ZL 114 MUL16/MUL16P. Please refer to the explanation of ZL 114 MUL16/MUL16P for more information.

- 32-bit BIN multiplication:

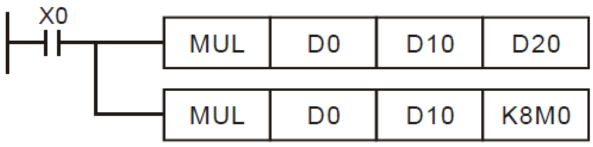


Symbol bit = 0 refers to a positive value. Symbol bit = 1 refers to a negative value.

When D serves as a bit device, it can designate K1 ~ K8 and construct a 32-bit result, occupying consecutive 2 groups of 32-bit data.

2、Program Example:

The 16-bit D0 is multiplied by the 16-bit D10 and brings forth a 32-bit product. The higher 16 bits are stored in D21 and the lower 16-bit are stored in D20. On/Off of the most left bit indicates the positive/negative status of the result value.

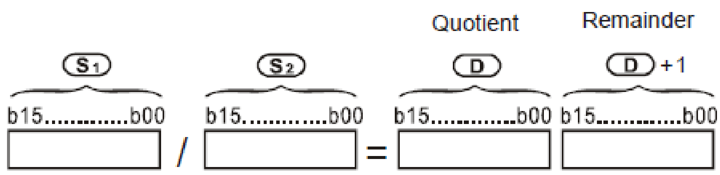


ZL 23		DIV		S1 S2 D	BIN Division
	D				

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*		DIV: 7 steps 16-bit DDIV: 13 steps 32-bit
S2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

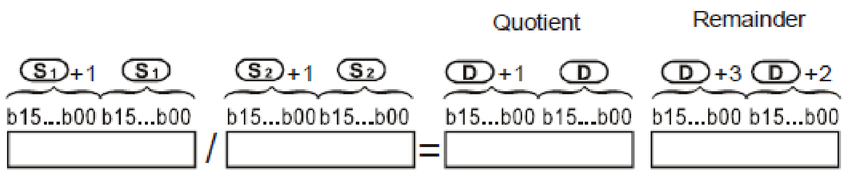
1、Explanations:

- S1: Dividend S2: Divisor D: Quotient and remainder
- To execute the pulse type, add the NP rising edge "↑" command before the command.
- If S1 and S2 are used in device F, only 16-bit instruction is applicable.
- If D is used in device E, only 16-bit instruction is applicable.
- In 16-bit instruction, D occupies 2 consecutive devices.
- In 32-bit instruction, D occupies 4 consecutive devices.
- This instruction divides S1 and S2 in BIN format and stores the result in D. Be careful with the positive/negative signs of S1, S2 and D when doing 16-bit and 32-bit operations.
- This instruction will not be executed when the divisor is 0.
- In 16-bit BIN division:



When D serves as a bit device, it can designate K1 ~ K4 and construct a 16-bit result, occupying consecutive 2 groups of 16-bit data and bringing forth the quotient and remainder.

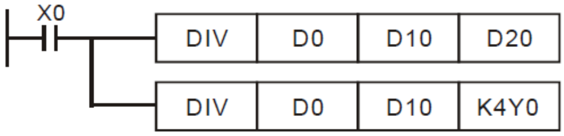
- In 32-bit BIN division:



When D serves as a bit device, it can designate K1 ~ K8 and construct a 32-bit result, occupying consecutive 2 groups of 32-bit data and bringing forth the quotient and remainder.

2、Program Example:

When X0 = On, D0 will be divided by D10 and the quotient will be stored in D20 and remainder in D21. On/Off of the highest bit indicates the positive/negative status of the result value.


X0
DIV
D0
D10
D20
DIV
D0
D10
K4Y0

ZL 24	D	INC	D	Increment

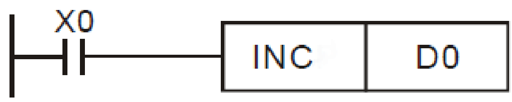
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	INC: 3 steps 16-bit DINC: 5 steps 32-bit
D								*	*	*	*	*	*	*	*	

1、Explanations:

- D: Destination device
- To execute the pulse type, add the NP rising edge "↑" command before the command.
- If D is used in device F, only 16-bit instruction is applicable.
- If the instruction is not a pulse execution one, the content in the designated device D will plus "1" in every scan period whenever the instruction is executed.
- This instruction adopts pulse execution instructions (INCP, DINCP).
- In 16-bit operation, 32,767 pluses 1 and obtains -32,768. In 32-bit operation, 2,147,483,647 pluses 1 and obtains -2,147,483,648.

2、Program Example:

When X0 = Off→On, the content in D0 pluses 1 automatically



ZL 25	D	DEC		D												Decrement
	Bit Devices				Word Devices											DEC: 3 steps 16-bit DDEC : 5 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	

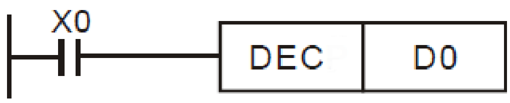
1、Explanations:

- D: Destination device
- If D is used in device F, only 16-bit instruction is applicable.
- If the instruction is not a pulse execution one, the content in the designated device D will minus "1" in every scan period whenever the instruction is executed.
- This instruction adopts pulse execution instructions (DECP, DDECP).
- In 16-bit operation, -32,768 minuses 1 and obtains 32,767. In 32-bit operation, -2,147,483,648 minuses 1 and obtains 2,147,483,647.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

When X0 = Off→On, the content in D0 minuses 1 automatically.

If the DEC instruction needs to be executed once, add the rising edge "↑" of NP before DEC.



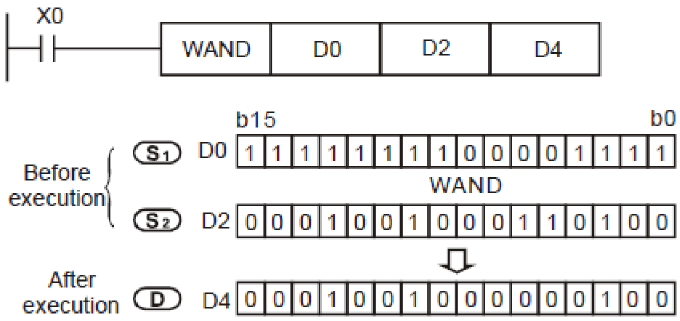
ZL 26	W / D	AND		S1 S2 D												Logical Word AND
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*	*	WAND: 7 steps 16-bit
S2					*	*	*	*	*	*	*	*	*	*	*	DAND: 13 steps 32-bit
D								*	*	*	*	*	*	*	*	

1、Explanations:

- S1: Source data device 1 S2: Source data device 2 D: Operation result
- If S1, S2 and D are used in device F, only 16-bit instruction is applicable.
- This instruction conducts logical AND operation of S1 and S2 and stores the result in D.
- Operation rule: The corresponding bit of the operation result in D will be "0" if any of the bits in S1 or S2 is "0".
- To execute the pulse type, add the NP rising edge "↑" command before the command.

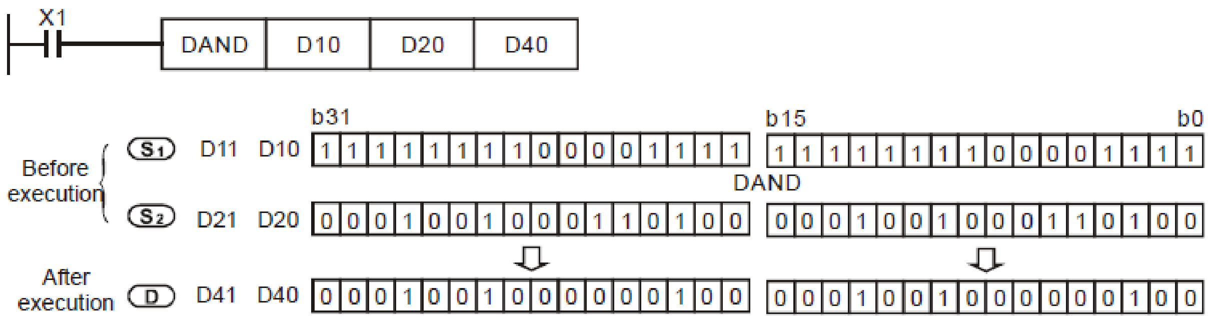
2、Program Example 1:

When X0 = On, the 16-bit D0 and D2 will perform WAND, logical AND operation, and the result will be stored in D4.



3、Program Example 2:

When X1 = On, the 32-bit (D11, D10) and (D21, D20) will perform DAND, logical AND operation, and the result will be stored in (D41, D40).



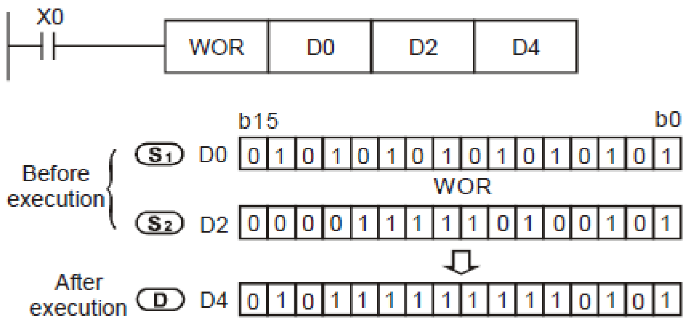
ZL 27	W / D	OR		S1 S2 D												Logical Word OR
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WOR: 7 steps 16-bit DOR: 13 steps 32-bit
S1					*	*	*	*	*	*	*	*	*	*	*	
S2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

1、Explanations:

- S1: Source data device 1 S2: Source data device 2 D: Operation result
- If S1, S2 and D are used in device F, only 16-bit instruction is applicable.
- This instruction conducts logical OR operation of S1 and S2 and stores the result in D.
- Operation rule: The corresponding bit of the operation result in D will be "1" if any of the bits in S1 or S2 is "1".
- To execute the pulse type, add the NP rising edge "↑" command before the command.

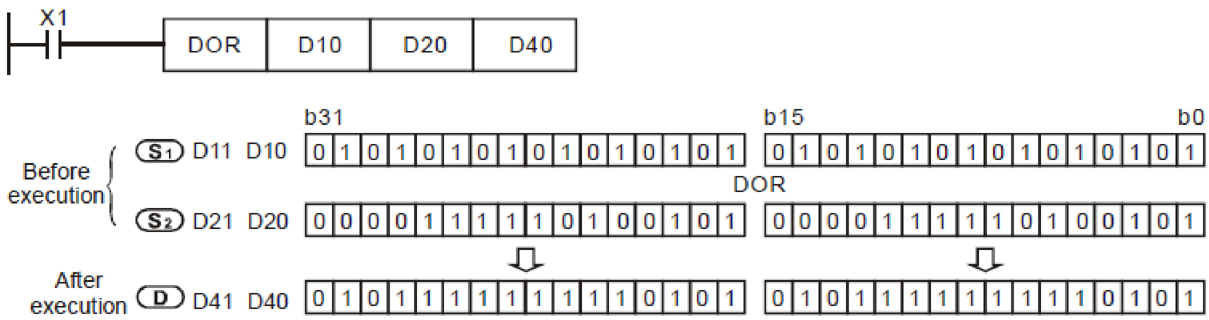
2、Program Example 1:

When X0 = On, the 16-bit D0 and D2 will perform WOR, logical OR operation, and the result will be stored in D4.



3、Program Example 2:

When X1 = On, the 32-bit (D11, D10) and (D21, D20) will perform DOR, logical OR operation, and the result will be stored in (D41, D40).



ZL 28	W / D	XOR		S1 S2 D											Logical Exclusive OR
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

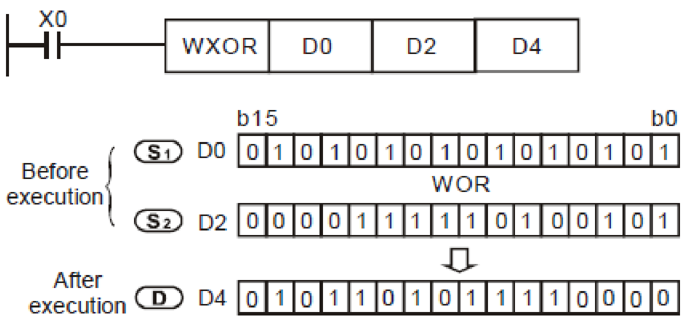
WXOR: 7 steps 16-bit
DXOR: 13 steps 32-bit

1、Explanations:

- S1: Source data device 1 S2: Source data device 2 D: Operation result
- If S1, S2 and D are used in device F, only 16-bit instruction is applicable.
- This instruction conducts logical XOR operation of S1 and S2 and stores the result in D.
- Operation rule: If the bits in S1 and S2 are the same, the corresponding bit of the operation result in D will be "0" ; if the bits in S1 and S2 are different, the corresponding bit of the operation result in D will be "1" .
- To execute the pulse type, add the NP rising edge "↑" command before the command.

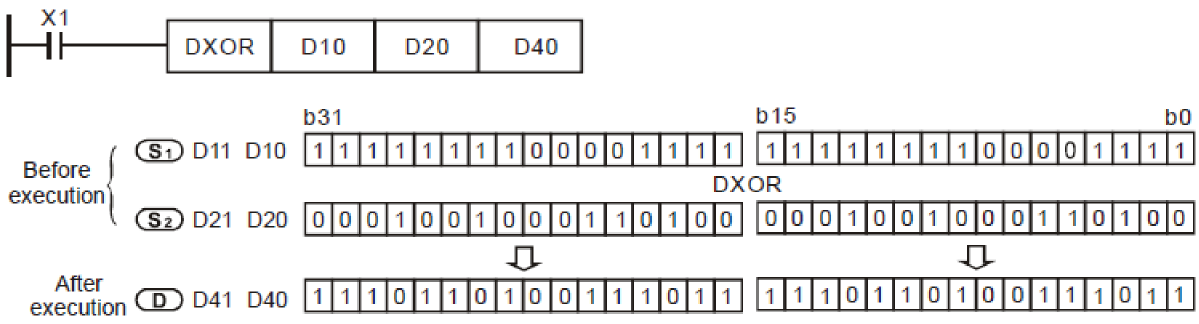
2、Program Example 1:

When X0 = On, the 16-bit D0 and D2 will perform WXOR, logical XOR operation, and the result will be stored in D4.



3、Program Example 2:

When X1 = On, the 32-bit (D11, D10) and (D21, D20) will perform DXOR, logical XOR operation, and the result will be stored in (D41, D40).


X1
DXOR
D10
D20
D40
b31
b15
b0
Before execution
S1 D11 D10
1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1
DXOR
S2 D21 D20
0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0
0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0
After execution
D D41 D40
1 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1
1 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1

ZL 29	D	NEG		D	2' s Complement (Negative)

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	NEG: 3 steps 16-bit DNEG: 5 steps 32-bit
D								*	*	*	*	*	*	*	*	

1、Explanations:

- D: Device to store 2' s complement.
- If D is used in device F, only 16-bit instruction is applicable
- This instruction converts a negative BIN value into an absolute value
- This instruction adopts pulse execution instructions (NEGP, DNEGP).
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example 1:

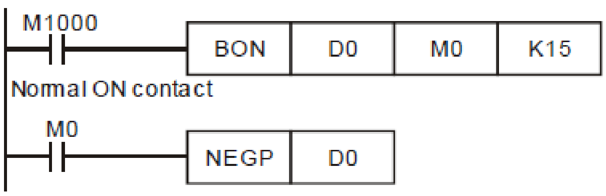
When X0 = Off→On, the phase of every bit of the content in D10 will be reversed (0→1, 1→0) and pluses 1. The result will then be stored in D10.

3、Program Example 2:

- Obtaining the absolute value of a negative value:

  a) When the 15th bit of D0 is "1", M0 = On. (D0 is a negative value).

  b) When M0 = Off→On, NEG instruction will obtain 2's complement of D0 and further its absolute value.



4、Program Example 3:

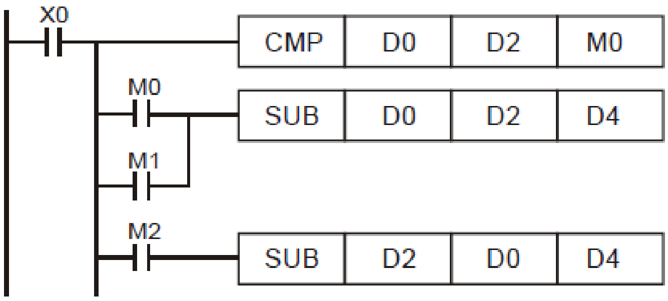
- Obtaining the absolute value by the remainder of the subtraction. When X0 = On:

  a) If D0 > D2, M0 = On.

  b) If D0 = D2, M1 = On.

  c) If D0 < D2, M2 = On.

  d) D4 is then able to remain positive

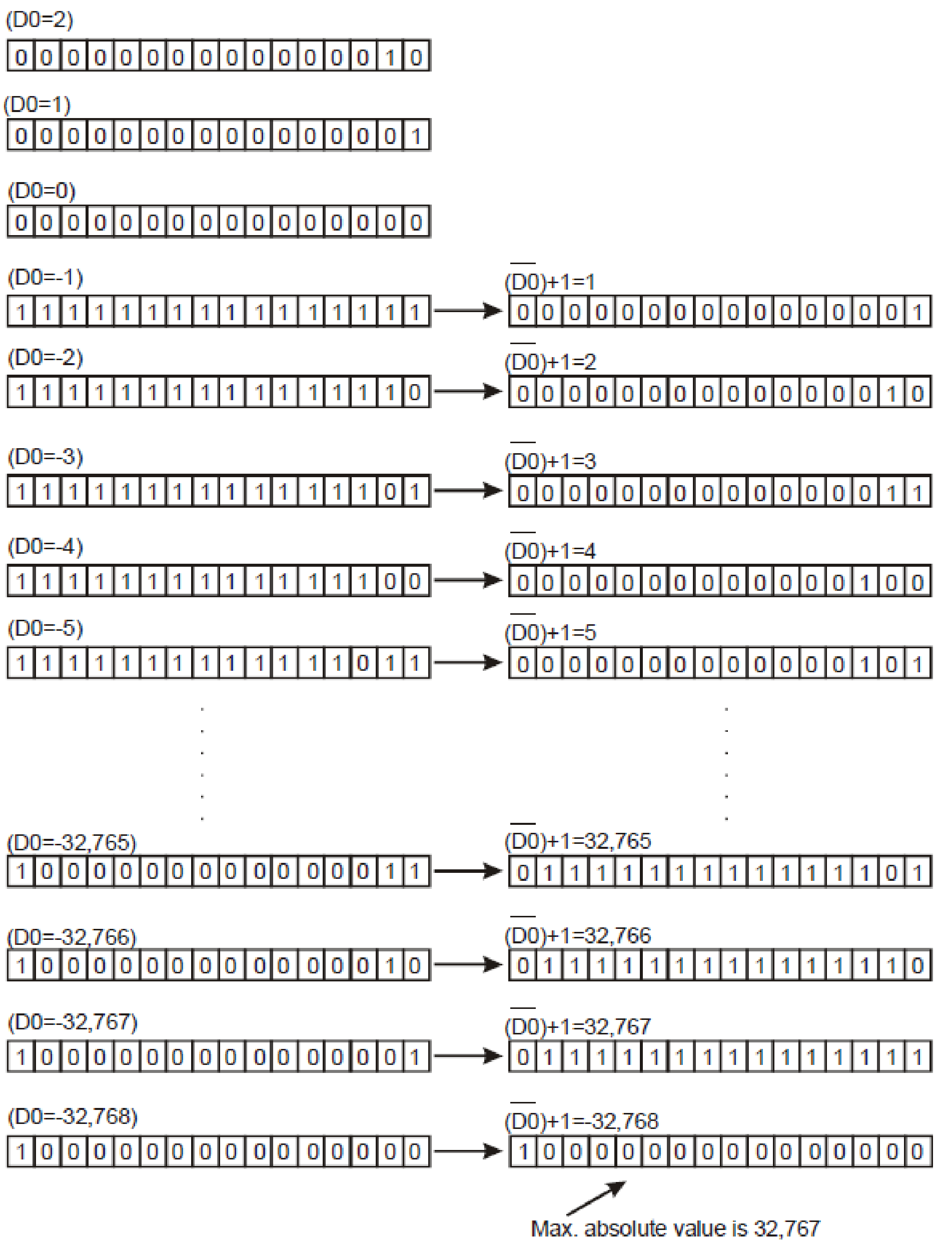


Remarks:

- Negative value and its absolute value

a. The sign of a value is indicated by the highest (most left) bit in the register. 0 indicates that the value is a positive one and 1 indicates that the value is a negative one。

b. NEG instruction is able to convert a negative value into its absolute value。



6.4（ZL 30-39）Cyclic shift

ZL 30			
D	ROR	D n	Rotation Right

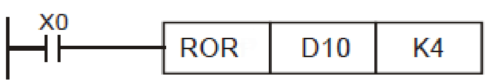
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ROR: 5 steps 16-bit
D								*	*	*	*	*	*	*	*	DROR: 9 steps 32-bit
n					*	*										

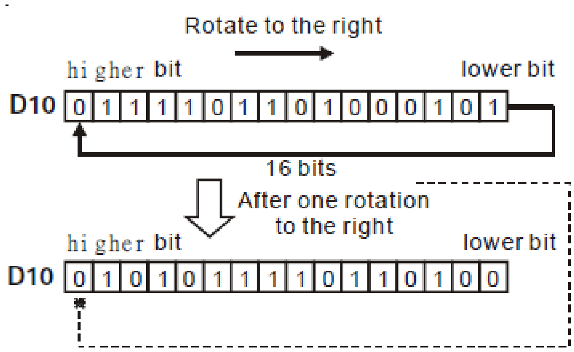
1、Explanations:

- D: Device to be rotated　　n: Number of bits to be rotated in 1 rotation
- If D is used in device F, only 16-bit instruction is applicable.
- If D is designated as KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid.
- Range of n: K1 ~ K16 (16-bit); K1 ~ K32 (32-bit)
- This instruction rotates the device content designated by D to the right for n bits.
- This instruction adopts pulse execution instructions (RORP, DRORP)
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example

When X0 = Off→On, the 16 bits (4 bits as a group) in D10 will rotate to the right, as shown in the figure below.





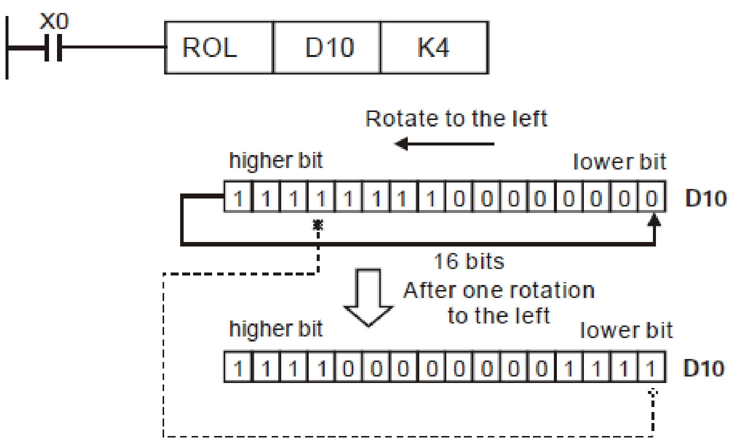
ZL 31	D	ROL		D n												Rotation Left
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ROL: 5 steps 16-bit
D								*	*	*	*	*	*	*	*	DROL: 9 steps 32-bit
n					*	*										

1、Explanations:

- D: Device to be rotated n: Number of bits to be rotated in 1 rotation
- If D is used in device F, only 16-bit instruction is applicable.
- If D is designated as KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid.
- Range of n: K1 ~ K16 (16-bit); K1 ~ K32 (32-bit).
- This instruction rotates the device content designated by D to the left for n bits.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

When X0 = Off→On, the 16 bits (4 bits as a group) in D10 will rotate to the left, as shown in the figure below



ZL 32	RCR	D n	Rotation Right with Carry
	D		

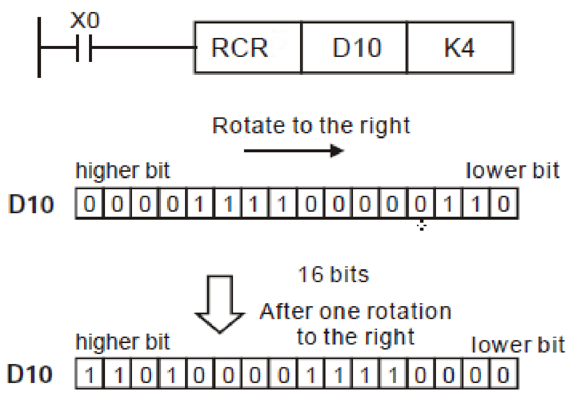
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	RCR: 5 steps 16-bit
D								*	*	*	*	*	*	*	*	DRCR: 9 steps 32-bit
n					*	*										

1、Explanations:

- D: Device to be rotated n: Number of bits to be rotated in 1 rotation
- If D is used in device F, only 16-bit instruction is applicable.
- If D is designated as KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid.
- Range of n: K1 ~ K16 (16-bit); K1 ~ K32 (32-bit).
- This instruction rotates the device content designated by D to the right for n bits.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

When X0 = Off→On, the 16 bits (4 bits as a group) in D10 will rotate to the right, as shown in the figure below。



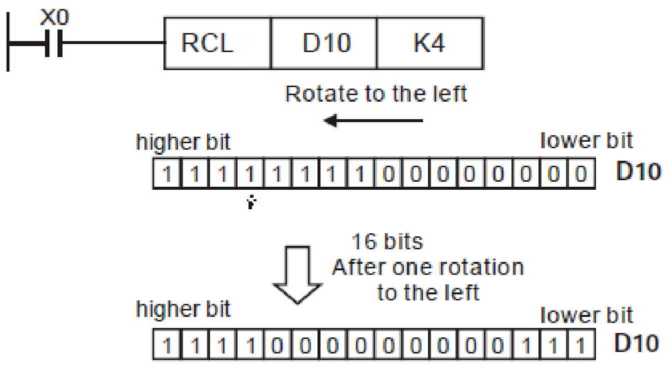
ZL 33	RCL					D n										Rotation Left with Carry
	D															
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	RCL: 5 steps 16-bit
D								*	*	*	*	*	*	*	*	DRCL: 9 steps 32-bit
n					*	*										

1、Explanations:

- D: Device to be rotated n: Number of bits to be rotated in 1 rotation
- If D is used in device F, only 16-bit instruction is applicable
- If D is designated as KnY, KnM, and KnS, only K4 (16-bit) and K8 (32-bit) are valid
- Range of n: K1 ~ K16 (16-bit); K1 ~ K32 (32-bit)
- This instruction rotates the device content designated by D to the left for n bits
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

When X0 = Off→On, the 16 bits (4 bits as a group) in D10 will rotate to the left, as shown in the figure below



ZL 34	SFTR	S D n_1 n_2	Bit Shift Right

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SFTR: 9 steps 16-bit
S	*	*	*	*												
D		*	*	*												
n_1					*	*										
n_2					*	*										

1、Explanations:

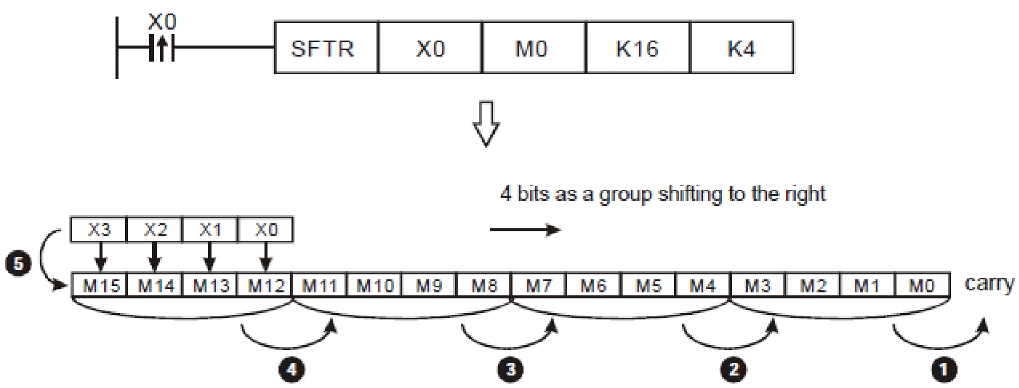
- S: Start No. of the shifted device D: Start No. of the device to be shifted
- n1: Length of data to be shifted n2: Number of bits to be shifted in 1 shift
- Range of n1: 1~ 1,024
- Range of n2: 1 ~ n1
- This instruction shifts the bit device of n1 bits (desired length for shifted register) starting from D to the right for n2 bits. S is shifted into D for n2 bits to supplement empty bits.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

When X0 = Off→On, M0 ~M15 will form 16 bits and shifts to the right (4 bits as a group).

The figure below illustrates the right shift of the bits in one scan.

❶ M3 ~ M0 → carry
❷ M7 ~ M4 → M3 ~ M0
❸ M11 ~ M8 → M7 ~ M4
❹ M15 ~ M12 → M11 ~ M8
❺ X3 ~ X0 → M15 ~ M12 completed



ZL 35	SFTL	S D n_1 n_2	Bit Shift Left

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												SFTL: 9 steps 16-bit
D		*	*	*												
n_1					*	*										
n_2					*	*										

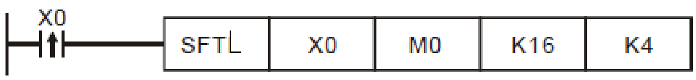
1、Explanations:

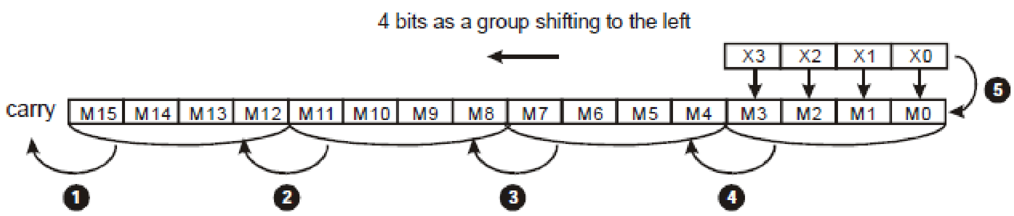
- S: Start No. of the shifted device D: Start No. of the device to be shifted
- n1: Length of data to be shifted n2: Number of bits to be shifted in 1 shift
- Range of n1: 1~ 1,024
- Range of n2: 1 ~ n1
- This instruction shifts the bit device of n1 bits (desired length for shifted register) starting from D to the left for n2 bits. S is shifted into D for n2 bits to supplement empty bits
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- When X0 = Off→On, M0 ~M15 will form 16 bits and shifts to the left (4 bits as a group).
- The figure below illustrates the left shift of the bits in one scan.

❶ M15 ~ M12 → carry
❷ M11 ~ M8 → M15 ~ M12
❸ M7 ~ M4 → M11 ~ M8
❹ M3 ~ M0 → M7 ~ M4
❺ X3 ~ X0 → M3 ~ M0 completed





ZL 36	WSFR	S D n_1 n_2	Word Shift Left

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WSFR: 9 steps 16-bit
S							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n_1					*	*										
n_2					*	*										

1、Explanations:

- S: Start No. of the shifted device D: Start No. of the device to be shifted
- n1: Length of data to be shifted n2: Number of words to be shifted in 1 shift
- The type of devices designated by S and D has to be the same, e.g. KnX, KnY, KnM, and KnS as a category and T, C, and D as another category
- Provided the devices designated by S and D belong to Kn type, the number of digits of Kn has to be the same
- Range of n1: 1~ 512
- Range of n2: 1 ~ n1
- This instruction shifts the stack data of n1 words starting from D to the right for n2 words. S is shifted into D for n2 words to supplement empty words.
- To execute the pulse type, add the NP rising edge " ↑ " command before the command.

2、Program Example 1:

- When X0 = Off→On, the 16 register stack data composed of D20 ~ D35 will shift to the right for 4 registers.
- The figure below illustrates the right shift of the words in one scan.

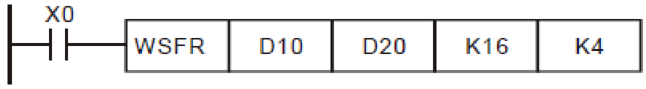
❶ D23 ~ D20 → carry

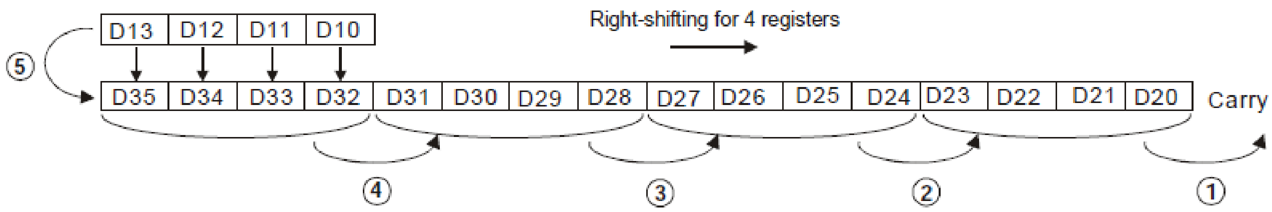
❷ D27 ~ D24 → D23 ~ D20

❸ D31 ~ D28 → D27 ~ D24

❹ D35 ~ D32 → D31 ~ D28

❺ D13 ~ D10 → D35 ~ D32 completed





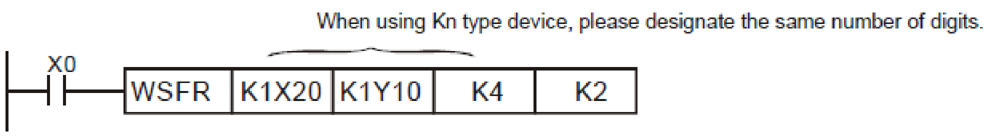
3、Program Example 2:

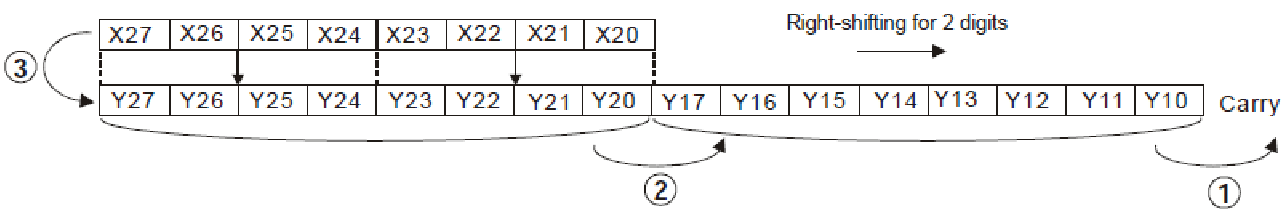
- When X0 = Off→On, the bit register stack data composed of Y10 ~ Y27 will shift to the right for 2 digits.
- The figure below illustrates the right shift of the words in one scan.

❶ Y17 ~ Y10 → carry

❷ Y27 ~ Y20 → Y17 ~ Y10

❸ X27 ~ X20 → Y27 ~ Y20 completed





ZL 37	WSFL	S D n_1 n_2	Word Shift Left

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			WSFL: 9 steps 16-bit
D								*	*	*	*	*	*			
n_1					*	*										
n_2					*	*										

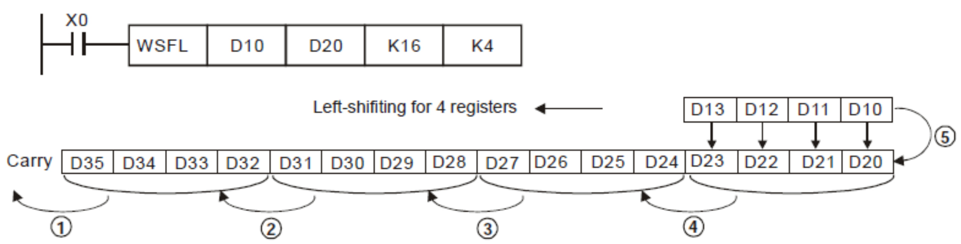
1、Explanations:

- S: Start No. of the shifted device D: Start No. of the device to be shifted
- n1: Length of data to be shifted n2: Number of words to be shifted in 1 shift
- The type of devices designated by S and D has to be the same, e.g. KnX, KnY, KnM, and KnS as a category and T, C, and D as another category
- Provided the devices designated by S and D belong to Kn type, the number of digits of Kn has to be the same.
- Range of n1: 1~ 512
- Range of n2: 1 ~ n1
- This instruction shifts the stack data of n1 words starting from D to the left for n2 words. S is shifted into D for n2 words to supplement empty words.
- To execute the pulse type, add the NP rising edge " ↑ " command before the command.

2、Program Example:

- When X0 = Off→On, the 16 register stack data composed of D20 ~ D35 will shift to the left for 4 registers.
- The figure below illustrates the left shift of the words in one scan.

❶ D35 ~ D32 → carry
❷ D31 ~ D28 → D35 ~ D32
❸ D27 ~ D24 → D31 ~ D28
❹ D23 ~ D20 → D27 ~ D24
❺ D13 ~ D10 → D23 ~ D20 completed



ZL 38	SFWR	S D n	Shift Register Write

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SFWR: 7 steps 16-bit
D								*	*	*	*	*	*			
n					*	*										

1、Explanations:

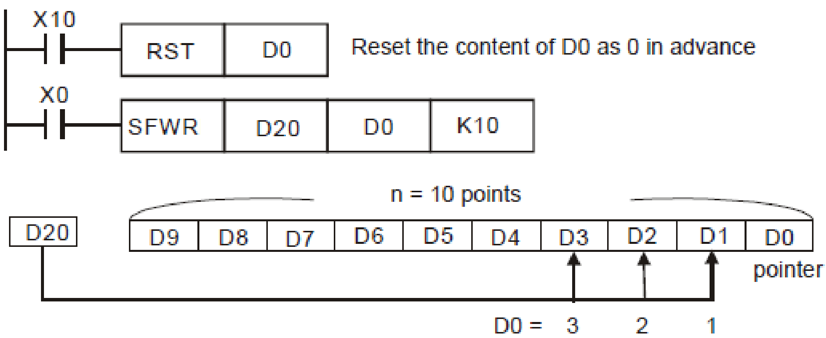
- S: Device of stack data written in D: Start No. of stack data n: Length of stack data
- Range of n: 2 ~ 512
- The stack data of n words starting from D are defined as “first-in, first-out” stack data and designate the first device as the pointer. When the instruction is executed, the content in the pointer pluses 1, and the content in the device designated by S will be written into the designated location in the “first-in, first-out” stack data designated by the pointer. When the content in the pointer exceeds n - 1, this instruction will not process any new value written .
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- Pointer D0 is reset as 0. When X0 = Off→On, the content in D20 will be sent to D1 and the content in pointer D0 becomes 1. After the content in D20 is changed, make X0 = Off→On again, and the content in D2 will be sent to D2 and the content in D0 becomes 2.
- The figure below illustrates the shift and writing in 1~2 execution of the instruction.

❶ The content in D20 is sent to D1.

❷ The content in pointer D0 becomes 1.



ZL 39	SFRD															
	S D n															Shift Register Read
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			SFRD: 7 steps 16-bit
D								*	*	*	*	*	*	*	*	
n					*	*										

1、Explanations:

- S: Start No. of stack data D: Device of stack data read out n: Length of stack data
- Range of n: 2 ~ 512
- The stack data of n words starting from S are defined as "first-in, first-out" stack data. After the content in S minuses 1, the content in the device designated by (S + 1) will be written into the location designated by D, and (S + n-1) ~ (S + 2) will all right shift for one register while the content in (S + n-1) remains the same. When the content in S equals 0, this instruction will not process any new value read out.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

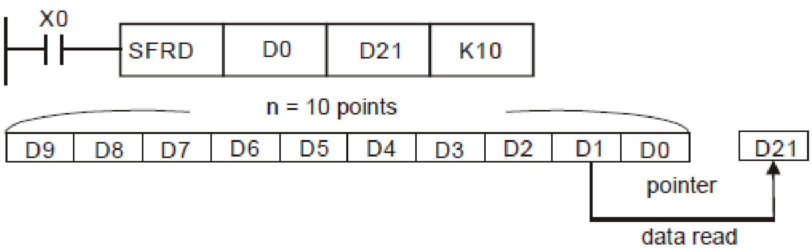
2、Program Example:

- When X0 = Off→On, the content in D1 will be sent to D21 and D9~D2 will shift to the right for 1 register (content in D9 remains unchanged) and the content in D0 minus 1.
- The figure below illustrates the shift and reading in 1~3 execution of the instruction
- The instruction executes a shift read operation according to the following numbers 1~3.

❶ The content in D1 is sent to D21.

❷ D9 ~ D2 shift to the right for 1 register.

❸ The content in D0 minuses 1.



6.5 (ZL40-49) Data processing

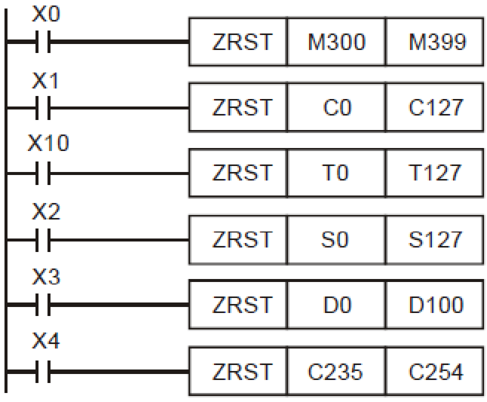
ZL 40	ZRST				D_1				D_2							Zero Reset
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ZRST: 5 steps 16-bit
D_1		*	*	*							*	*	*			
D_2		*	*	*							*	*	*			

1、Explanations:

- D1: Start device of the range to be reset D2: End device of the range to be reset
- No. of operand D1 ⩽ No. of operand D2.
- D1 and D2 have to designate devices of the same type.
- When the instruction is executed, area from D1 to D2 will be cleared.
- I6-bit counter and 32-bit counter can use ZRST instruction together.
- When D1 > D2, only operands designated by D2 will be reset.

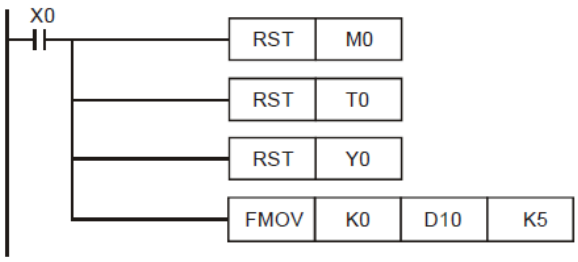
2、Program Example:

- When X0 = On, auxiliary relays M300 ~ M399 will be reset to Off.
- When X1 = On, 16 counters C0 ~ C127 will all be reset (writing in 0; contact and coil being reset to Off).
- When X10 = On, timers T0 ~ T127 will all be reset (writing in 0; contact and coil being reset to Off).
- When X2 = On, steps S0 ~ S127 will be reset to Off.
- When X3 = On, data registers D0 ~ D100 will be reset to 0.
- When X4 = On, 32-bit counters C235 ~ C254 will all be reset. (writing in 0; contact and coil being reset to Off



Remarks:

1) Devices, e.g. bit devices Y, M, S and word devices T, C, D, can use RST instruction.
2) ZL 16 FMOV instruction is also to send K0 to word devices T, C, D or bit registers KnY, KnM, KnS for reset.



ZL 41	DECO	S D n	Decode

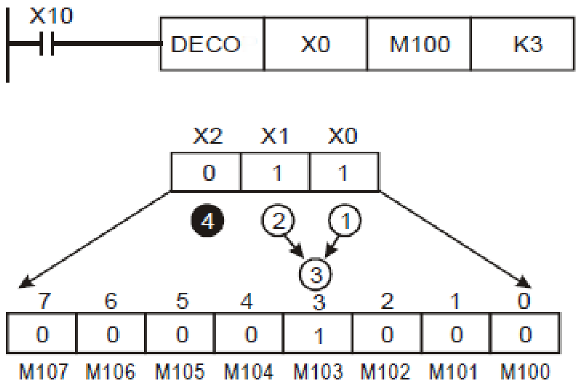
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*	*	*					*	*	*	*	*	DECO: 7 steps 16-bit
D		*	*	*							*	*	*	*	*	
n					*	*										

1、Explanations:

- S: Source device to be decoded D: Device for storing the decoded result n: Length of decoded bits
- Range of n when D is a bit device: 1 ~ 8
- Range of n when D is a word device: 1 ~ 4
- See the specifications of each model for their range of use.
- The lower "n" bits of S are decoded and the results of "2^n" bits are stored in D.
- This instruction adopts pulse execution instructions (DECOP)
- To execute the pulse type, add the NP rising edge "↑" command before the command.

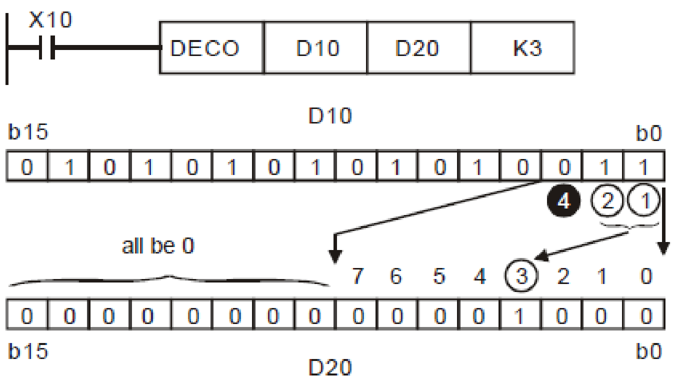
2、Program Example 1:

- When D is used as a bit device, n = 1 ~ 8. Errors will occur if n = 0 or n > 8.
- When n = 8, the maximum points to decode is 28 = 256 points. (Please be aware of the storage range of the devices after the decoding and do not use the devices repeatedly.)
- When X10 = Off→On, this instruction will decode the content in X0 ~ X2 to M100 ~ M107.
- When the source of data is 1 + 2 = 3, set M103, the 3rd bit starting from M100, as 1.
- After the execution of this instruction is completed and X10 turns to Off, the content that has been decoded and output keeps acting.



3、Program Example 2:

- When D is used as a word device, n = 1 ~ 4. Errors will occur if n = 0 or n > 4.
- When n = 4, the maximum points to decode is 2^4 = 16 points.
- When X10 = Off→On, this instruction will decode b2 ~ b0 in D10 to b7 ~ b0 in D20. b15 ~ b8 that have not been used in D20 will all become 0.
- The lower 3 bits of D10 are decoded and stored in the lower 8 bits of D20. The higher 8 bits of D20 are all 0.
- After the execution of this instruction is completed and X10 turns to Off, the content that has been decoded and output keeps acting.



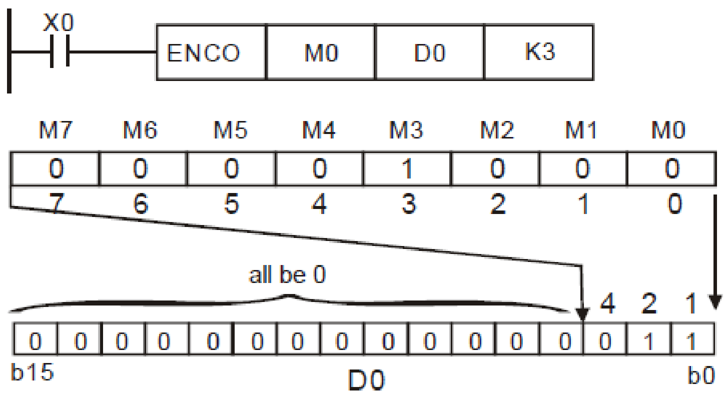
ZL 42	ENCO					S D n										Encode
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*	*	*					*	*	*	*	*	ENCO: 7 steps 16-bit
D											*	*	*	*	*	
n					*	*										

1、Explanations:

- S: Source device to be encoded D: Device for storing the encoded result n: Length of encoded bits
- Range of n when S is a bit device: 1 ~ 8
- Range of n when S is a word device: 1 ~ 4
- The lower “2^n” bits of S are encoded and the result is stored in D.
- If several bits of S are 1, the first bit that is 1 will be processed orderly from high bit to low bit.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example 1:

- When S is used as a bit device, n = 1 ~ 8. Errors will occur if n = 0 or n > 8.
- When n = 8, the maximum points to encode is 28 = 256 points.
- When X10 = Off→On, this instruction will encode the 23 bits data (M0 ~ M7) and store the result in the lower 3 bits (b2 ~ b0) of D0. b15 ~ b3 that have not been used in D0 will all become 0.
- After the execution of this instruction is completed and X10 turns to Off, the content in D remains unchanged.



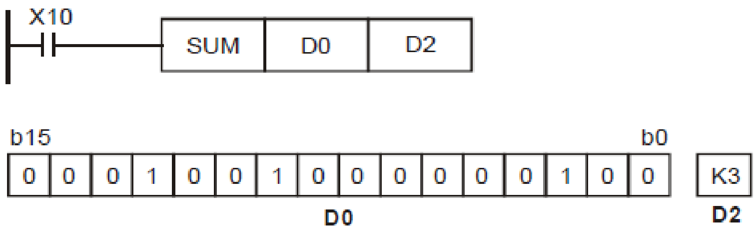
ZL 43		SUM		S D												Sum of Active Bits
	D															
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SUM: 5 steps 16-bit
S					*	*	*	*	*	*	*	*	*	*	*	DSUM: 9 steps 32-bit
D								*	*	*	*	*	*	*	*	

1、Explanations:

- S: Source device D: Destination device for storing counted value
- If S and D are used in device F, only 16-bit instruction is applicable.
- Among the bits of S, the total of bits whose content is “1” will be stored in D.
- When 32- instruction is in use, D will occupy 2 registers。
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- When X10 = On, among the 16 bits of D0, the total of bits whose content is “1” will be stored in D2。



ZL 44	D	BON		S D n	Check Specified Bit Status

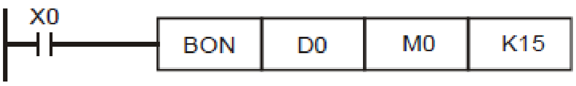
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BON: 7 steps 16-bit DBON: 13 steps 32-bit
S					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												
n					*	*					*	*	*	*	*	

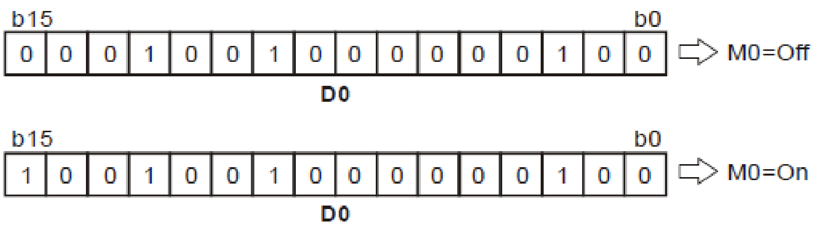
1、Explanations:

- S: Source device D: Device for storing check result n: Bits specified for check
- If S is used in device F, only 16-bit instruction is applicable
- Range of n: 0 ~ 15 (16-bit instruction); 0 ~ 31 (32-bit instruction)
- When the nth bit of S is "1", D = On; when the nth bit of S is "0", D = Off
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- When X0 = On, assume the 15th bit of D0 is "1", and M0 = On. Assume the 15th bit of D0 is "0", and M0 = Off.
- When X0 goes Off, M0 will remains in its previous status.





ZL 45	D	MEAN	S D n	Mean

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S							*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*	*
n					*	*	*	*	*	*	*	*	*	*	*

MEAN: 7 steps 16-bit

DMEAN: 13 steps 32-bit

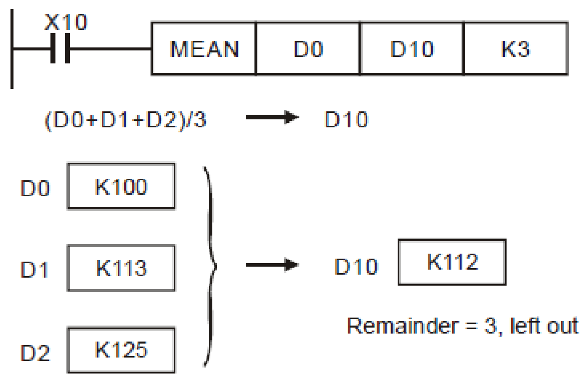
1、Explanations:

- S: Start device to obtain mean value D: Destination device for storing mean value

  n: The number of consecutive source devices used
- If D is used in device F, only 16-bit instruction is applicable.
- Range of n: 1 ~64
- After the content of n devices starting from S are added up, the mean value of the result will be stored in D.
- Remainders in the operation will be left out.
- Provided the No. of designated device exceeds its normal range, only the No. within the normal range can be processed.
- If n falls without the range of 1 ~ 64, PLC will determine it as an “instruction operation error” .
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- When X10 = On, the contents in 3 (n = 3) registers starting from D0 will be summed and then divided by 3. The obtained mean value will be stored in D10 and the remainder will be left out.



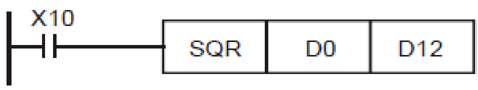
ZL 48	D	SQR				S						D				Square Root
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SQR: 5 steps 16-bit
S					*	*							*			DSQR: 9 steps 32-bit
D													*			

1、Explanations:

- S: Source device D: Device for storing the result
- This instruction performs a square root operation on S and stores the result in D.
- S can only be a positive value. If S is negative, PLC will regard it as an "instruction operation error" and will not execute this instruction.
- The operation result D should be integer only, and the decimal will be left out.

2、Program Example:

- When X10 = On, the instruction performs a square root on D0 and stores the result in D12.



$\sqrt{D0} \rightarrow D12$

<table>
<tr><td>ZL 49</td><td></td><td rowspan="2">FLT</td><td></td><td rowspan="2">S D</td><td rowspan="2" colspan="2">Convert BIN integer to binary floating point</td></tr>
<tr><td></td><td>D</td><td></td></tr>
</table>

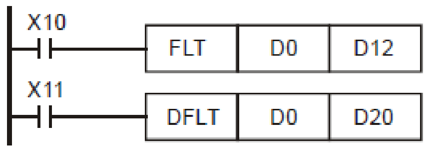
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FLT: 5 steps 16-bit
S													*			DFLT: 9 steps 32-bit
D													*			

1、Explanations:

- S: Source device for conversion D: Device for storing the conversion result.
- BIN integer is converted into binary floating point value. At this time, S of the 16-bit instruction, FLT, occupies 1 register and D occupies 2 registers.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

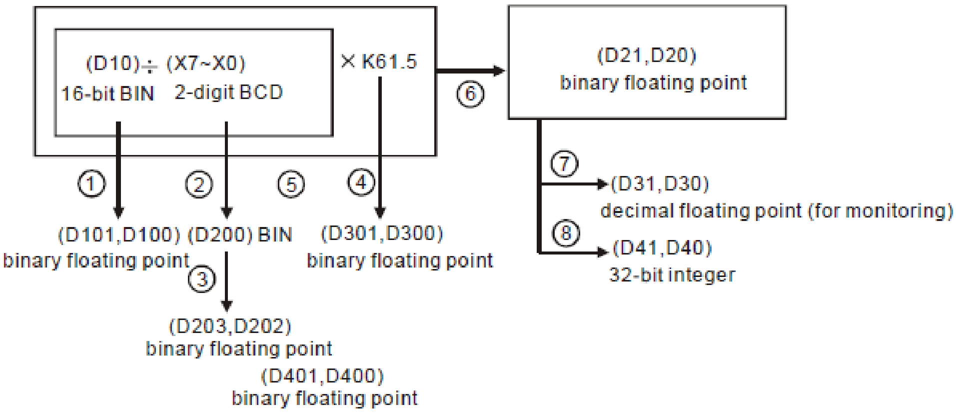
2、Program Example 1:

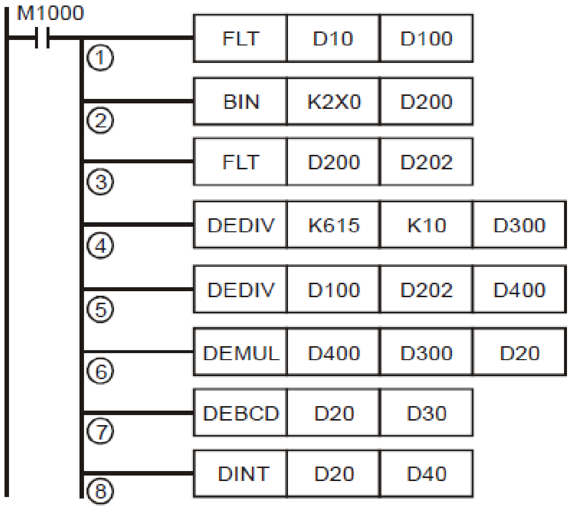
- the BIN integer is converted into binary floating point value.
- When X10 = On, D0 (BIN integer) is converted into D13 and D12 (binary floating point value).
- When X11 = On, D1 and D0 (BIN integer) are converted into D21 and D20 (binary floating point value).
- If D0 = K10, X10 will be On. The 32-bit value of the converted floating point will be H41200000 and stored in 32- bit register D12 (D13).
- If 32-bit register D0 (D1) = K100,000, X11 will be On. The 32-bit value of the converted floating point will be H47C35000 and stored in 32-bit register D20 (D21).



3、Program Example 2:

- Please use this instruction to complete the following operation.





note： 1) D10 (BIN integer) is converted to D101 and D102 (binary floating point value).

2) X7 ~ X0 (BCD value) are converted to D200 (BIN value).

3) D200 (BIN integer) is converted to D203 and D202 (binary floating point value). 4The result of K615 ÷ K10 is stored in D301 and D300 (binary floating point value).

4) The result of binary decimal division (D101, D100) ÷ (D203, D202) is stored in D401 and D400 (binary floating point value).

5) The result of binary decimal multiplication (D401, D400) × (D301, D300) is stored in D21 and D20 (binary floating point value).

6) D21 and D20 (binary floating point value) are converted to D31 and D30 (decimal floating point value).

7) D21 and D20 (binary floating point value) are converted to D41 and D40 (BIN integer).

7 Application Instructions ZL50~ZL99

7.1 (ZL 50-59) High-speed processing

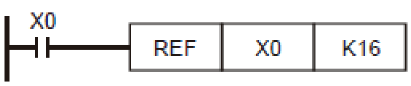
ZL 50	REF					D n										Refresh
	Bit Devices				Word Devices											REF: 5 steps 16-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D	*	*														
n					*	*										

1、Explanations:

- D: Start device to be I/O refreshed n: Number of items to be I/O refreshed
- D must designate X0, X10, Y0, Y10…the points whose 1s digit is "0". See remarks for more details
- Range of n: 8 ~ 256 (has to be the multiple of 8).
- The status of all PLC input/output terminals will be updated after the program scans to END. When the program starts to scan, the status of the external input terminal is read and stored into the memory of the input point. The output terminal will send the content in the output memory to the output device after END instruction is executed. Therefore, this instruction is applicable when the latest input/output data are needed for the operation.
- D has to be designated to be X0, X10, Y0, Y10…such forms whose 1st digit is "0". Range of n: 8 ~ 256 (must be 8's multiple); otherwise it will be regarded as an error. The range varies in different models. See Remarks for more details.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

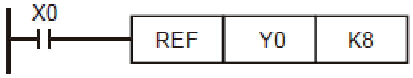
2、Program Example 1:

- When X0 = On, PLC will read the status of input points X0 ~ X17 immediately and refresh the input signals without any input delay.



3、Program Example 2:

- When X0 = On, the 8 output signal from Y0 ~ Y7 will be sent to output terminals and refreshed without having to wait for the END instruction for output.


X0
REF
Y0
K8

ZL 51		REFF				D n										Refresh and Filter Adjust
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	REFF: 3 steps 16-bit
n					*	*										

1、Explanation:

- n: Response time (unit: ms)
- Range of n: n = K0 ~ K60
- To avoid interferences, X0 ~ X17 are equipped with digital filters on output terminals. Digital filters adjust the response time by REFF instruction.
- Rules for adjusting the reponse time of the filter at X0 ~ X17:

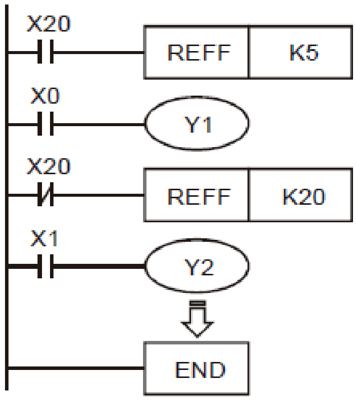
  a) When the power of PLC turns from Off to On or the END instruction is being executed, the response time will be determined upon the contents in D1020 and D1021.

  b) You can use MOV instruction in the program to move the time values to D1020 and D1021 and make adjustments in the next scan.

  c) You can use REFF instruction to change the response time during the execution of the program. The changed response time will be move to D1020 and D1021 and you can make adjustments in the next scan.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- When the power of PLC turns from Off to On, the response time of X0 ~ X17 will be determined by the contents in D1020 and D1021.
- When X20 = On, REFF K5 will be executed and the response time will be changed to 5ms for the adjustment in the next scan.
- When X20 = Off, the REFF K20 will be executed and the response time will be changed to 20ms for the adjustment in the next scan.


X20
REFF
K5
X0
Y1
X20
REFF
K20
X1
Y2
END

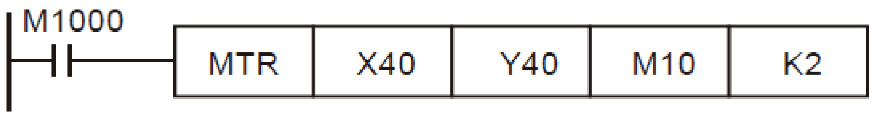
ZL 52		MTR		S D1 D2 n											Input Matrix
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MTR: 9 steps 16-bit
S	*															
D_1		*														
D_2		*	*	*												
n					*	*										

1、Explanations:

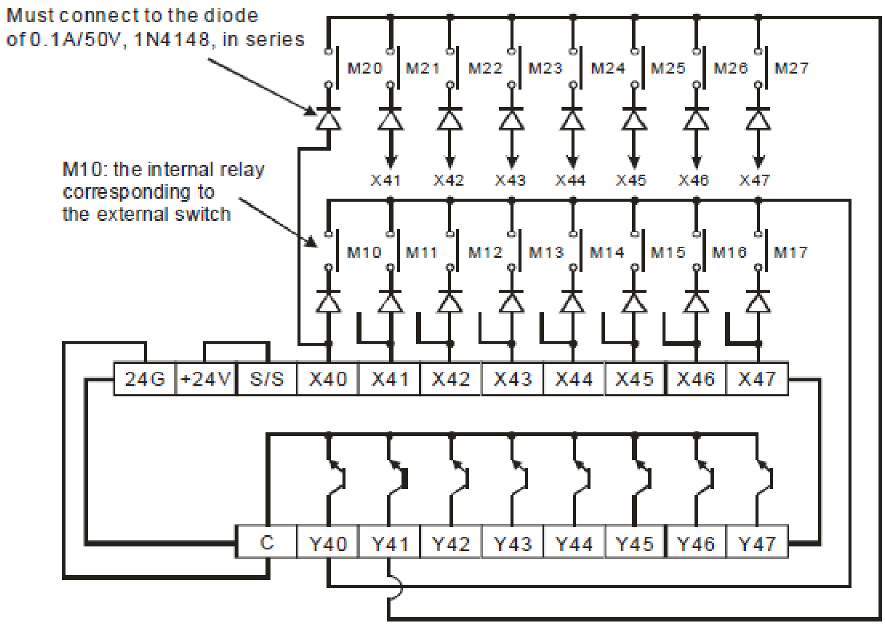
- S: Start device of matrix input　　D1: Start device of matrix output
  D2: Corresponding start device for matrix scan　　n: Number of arrays in matrix scan
- S must designate X0, X10…the X points whose 1st digit is "0" and occupies 8 consecutive points.
- D1 must designate Y0, Y10…the Y points whose 1st digit is "0" and occupies n consecutive points.
- D2 must designate Y0, M0. S0…the Y, M, S points whose 1st digit is "0".
- Range of n: 2 ~ 8.
- S is the start device No. of all input terminals connected to the matrix. Once S is designated, the 8 points following the No. will be the input terminals in the matrix.
- D1 designate the start device No. of transistor output Y in the matrix scan.
- This instruction occupies continuous 8 input devices starting from S. n external output terminals starting from D1 read the 8 switches of n arrays by matrix scan, obtaining 8 × n multiple-matrix input points. The status of scanned switches will be stored in the devices starting from D2.
- Maximum 8 input switches can be parallelly connected in 8 arrays and obtaining 64 input points (8 × 8 = 64).
- When the 8-point 8-array matrix inputs are in use, the reading time of each array is approximately 25ms, totaling the reading of 8 arrays 200ms, i.e. the input signals with On/Off speed of over 200ms are not applicable in a matrix input.
- Whenever this instruction finishes a matrix scan, M1029 will be On for one scan period.
- There is no limitation on the number of times using the instruction, but only one instruction can be executed in one scan cycle.

2、Program Example:

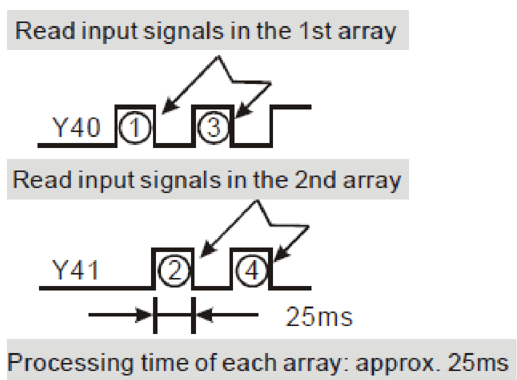
- When PLC RUN, MRT instruction will start to be executed. The statuses of the external 2 arrays of 16 switches, will be read in order and stored in the internal relays M10 ~ M17, M20 ~ M27。



- The figure below illustrates the external wiring of the 2-array matrix input loop constructed by X40 ~ X47 andY40 ~ Y41. The 16 switches correponds to the internal relays M10 ~ M17, M20 ~ M27. Should be used with MTR instruction.



- See the figure above. The 8 points starting from X40 start to perform a matrix scan from Y40 ~ Y41 (n = 2). D_2 designates that the start device No. of the read results is M10, indicating that the first array is read to M10 ~ M17 and the second array is read to M20 ~ M27.



ZL 53	D	HSCS	S1 S2 D	High Speed Counter Set

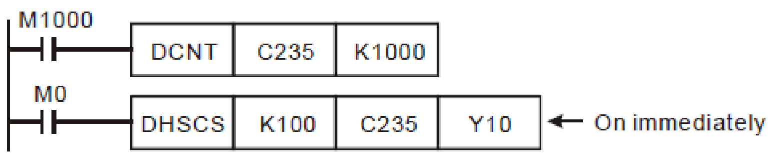
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*		DHSCS: 13 steps 32-bit
S2												*				
D		*	*	*												

1、Explanations:

- S1: Comparative value S2: No. of high speed counter D: Comparison result
- D can designate I0□0; □ = 1 ~ 6
- The high speed counter inputs counting pulses from the corresponding external input terminals X0 ~ X17 by inserting an interruption. When the high speed counter designated in S2 pluses 1 or minuses 1, DHSCS instruction will perform a comparison immediately. When the present value in the high speed counter equals the comparative value designated in S1, device designated in D will turn On. Even the afterward comparison results are unequal, the device will still be On.
- If the devices specified as the device D are Y0 ~ Y17, when the compare value and the present value of the high-speed counter are equal, the comparison result will immediately output to the external inputs Y0 ~ Y17, and other Y devices will be affected by the scan cycle. However, M, S devices are immediate output and will not be affected by the scan cycle.

2、Program Example 1:

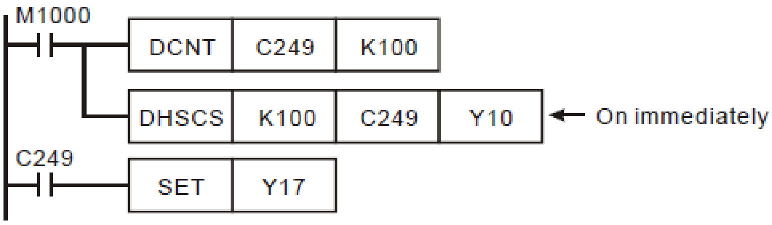
- After PLC RUN and M0 = On, DHSCS instruction will be executed. When the present value in C235 changes from 99 to 100 or 101 to 100, Y10 will be On constantly



3、Program Example 2:

- Differences between Y output of DHSCS instruction and general Y output:
  1) When the present value in C249 changes from 99 to 100 or 101 to 100, Y10 outputs immediately to the external output point by interruption and has nothing to do with the PLC scan time. However, the time will still be delayed by the relay (10ms) or transistor (10us) of the output module.

2) When the present value in C249 changes from 99 to 100, the drive contact of C249 will be On immediately. When the execution arrives at SET Y17, Y17 will still be affected by the scan time and will output after END instruction.

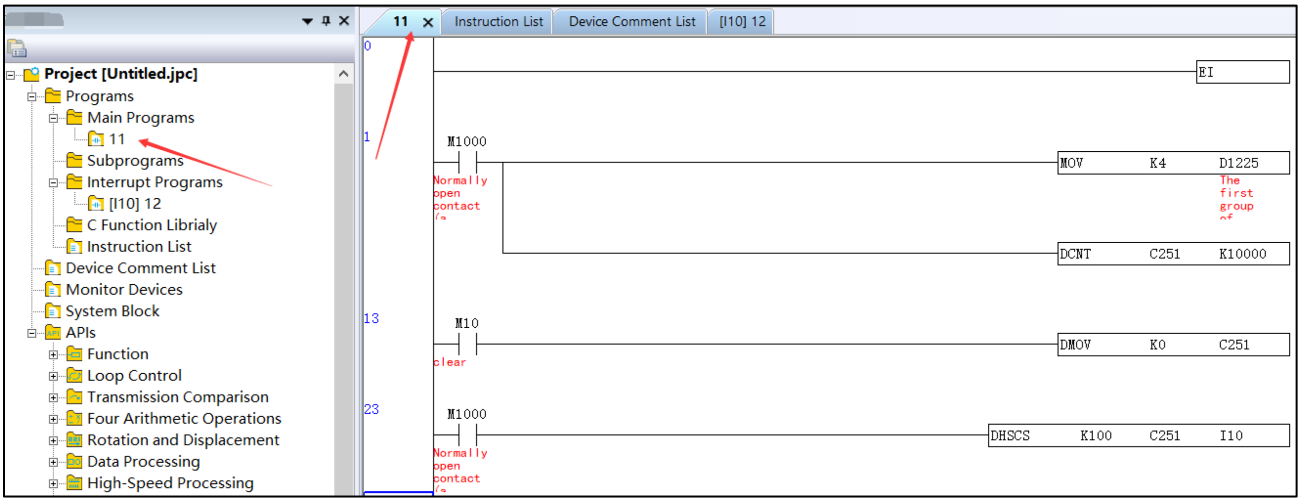


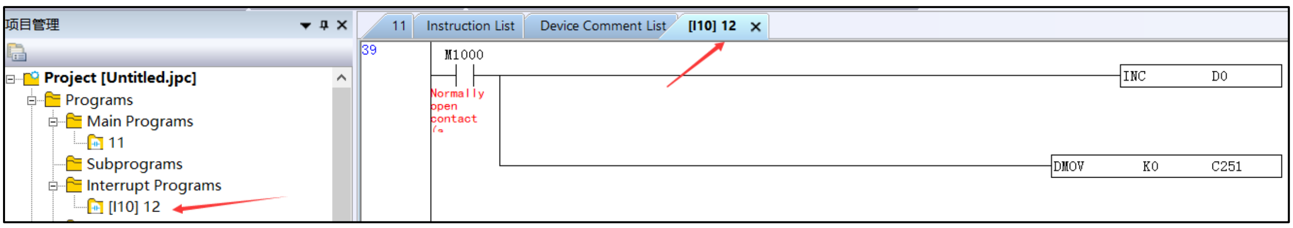
4、Program Example 3:

- High speed counter interruption:

1) Operand D of DHSCS instruction can designate I0□0, □ = 1 ~ 6, as the timing of interruption when the counting reaches its target.

2) When the present value in C251 changes from 99 to 100 or 101 to 100, the program will jump to I010 and execute the interruption service subroutine.





Remarks:

1) The output contact of the high speed counter and the comparative outputs of ZL 53 DHSCS, ZL 34 DHSCR and ZL 55 DHSZ instructions only perform comparison and contact outputs when there is a counting input. When using data operation instructions, e.g. DADD, DMOV, for changing the present value in the high speed

counter or making the present value equals the set value, there will not be comparisons or comparative outputs because there is no counting inputs.

2) Supports high speed counters. C235 ~ C240 are program-interruption 1-phase high speed counter with a total bandwidth of 20kHz, can be used alone with a counting frequency of up to 10kHz. C241 ~ C254 are hardware high speed counter (HHSC). There are four HHSC of HHSC0 ~ 3. The pulse input frequency of HHSC0~4 can reach 200kHz (1 phase or A-B phase).

3) Every HHSC can only be designated to one counter by DCNT instruction.

4) There are three counting modes in every HHSC (see the table below):

- 1-phase 1 input refers to "pulse/direction" mode
- 2-phase 2 inputs refers to "A-B phase" mode.

Type / Input	software high-speed counter						Hardware high-speed counter							
	1 phase 1 input						1 phase 1 input				2 phase 2 input			
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C251	C252	C253	C254
X0	U/D						U/D				A			
X1		U/D									B			
X2			U/D					U/D				A		
X3				U/D								B		
X4					U/D				U/D				A	
X5						U/D							B	
X6										U/D				A
X7														B

Description:

U	Progressively increasing input
D	Progressively decreasing input
A	A phase input
B	B phase input

5) Counting modes:

Special D1225 ~ D1228 are for setting up different counting modes of the hardware high speed counters (HHSC0 ~ 3) . There are normal ~ 4 times frequency for the counting and the default setting is double frequency.

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

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

Flag	Function
M1235 ~ M1244	C235 ~ C244 High speed counter counting direction specified. When M12□□=Off ， C2□□： Count on。 When M12□□=On ， C2□□:Count off。
D1225	The counting mode of the 1st group counters (C251)
D1226	The counting mode of the 2nd group counters (C252)
D1227	The counting mode of the 3rd group counters (C253)
D1228	The counting mode of the 4th group counters (C254)
D1225~D1228	PLC hardware high speed counter HHSC0~ HHSC3 counting mode setting, not the following setting values are preset for the double frequency counting mode. 2： for the double frequency counting mode, (factory value). 3： it is the triple frequency counting mode. 4： it is the quadruple frequency counting mode.（desired value）

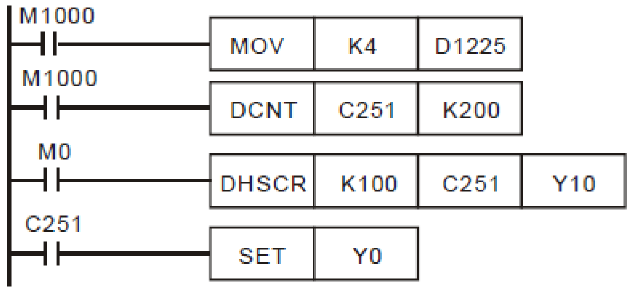
ZL 54	D	HSCR				S1 S2 D										High Speed Counter Reset
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*		DHSCR: 13 steps 32-bit
S2												*				
D		*	*	*								*				

1、Explanations:

- S1: Comparative value S2: No. of high speed counter D: Comparison result
- S2 has to designate the No. of high speed counters C235 ~ C255. See remarks of ZL 53 DHSCS for more details.
- D of high speed counters C241 ~ C254 that are the same as the counters designated by S2
- The high speed counter inputs counting pulses from the corresponding external input terminals X0 ~ X7 by inserting an interruption. When the No. of high-speed counter designated in S2 "+1 " or "-1", DHSCR will perform a comparison immediately. When the present value in the high speed counter equals the comparative value designated in S1, the device designated in D will turn Off and even the afterward comparison results are unequal, the device will still be Off.
- If the devices designated in D are Y0 ~ Y17, when the comparative value equals the present value in the high speed counter, the comparison result will immediately output to the external output terminals Y0 ~ Y17 (and clear the designated Y output) and the rest of Y devices will be affected by the scan cycle. Devices M and S act immediately without being affected by the scan cycle.

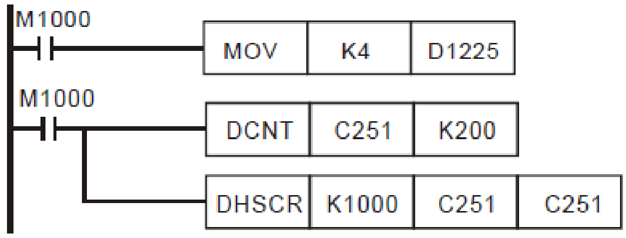
2、Program Example 1:

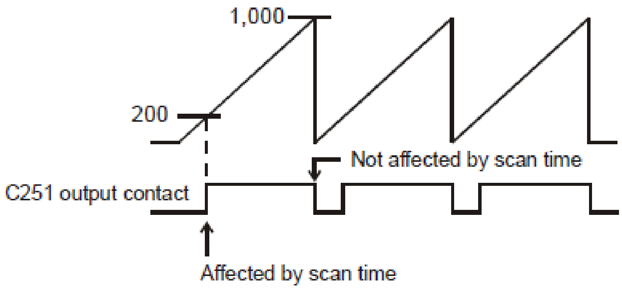
- When M0 = On and the present value in the high speed counter C251 changes from 99 to 100 or 101 to 100, Y10 will be cleared and Off.
- When the present value in the high speed counter C251 changes from 199 to 200, the contact of C251 will be On and make Y0 = On. However, the program scan time will delay the output.
- Y10 will immediately reset the status when the counting reaches its target. D is also able to designate high speed counters of the same No. See Program Example 2.



3、Program Example 2:

- When DHSCR instruction designates the same high speed counter, and the present value in the high speed counter C251 changes from 999 to 1,000 or 1,001 to 1,000, C251 will be reset to Off.





ZL 55	D	HSZ	S1 S2 S D	High Speed Zone Compare

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*		DHSZ: 17 steps 32-bit
S2					*	*	*	*	*	*	*	*	*	*		
S																
D		*	*	*												

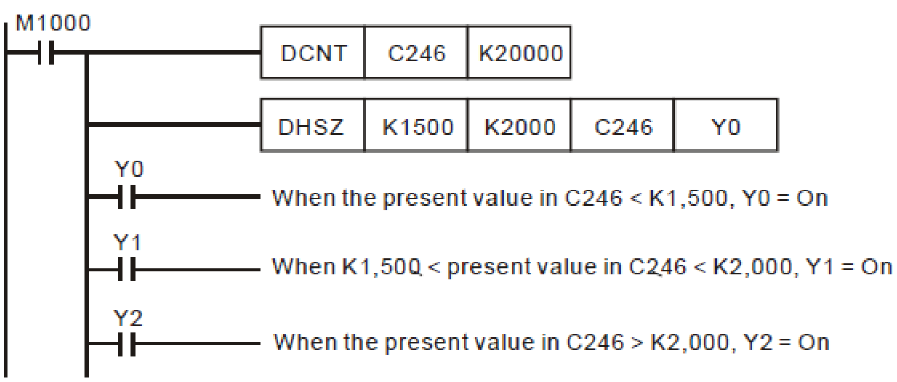
1、Explanations:

- S1: Lower bound of the comparison zone S2: Upper bound of the comparison zone

  S: No. of high speed counter D: Comparison result
- S1 has to be euqal to or smaller than S2. (S1 ⩽ S2)
- When S1 > S2, the instruction will perform a comparison by using S1 as the upper bound and S2 as the lower bound.
- S has to designate high speed counters C235 ~ C255, See remarks of ZL 53 DHSCS for more details.
- D will occupy 3 consecutive devices.
- The output will not be affected by the scan time.
- The zone comparisons and outputs are all processed by inserting interruptions.

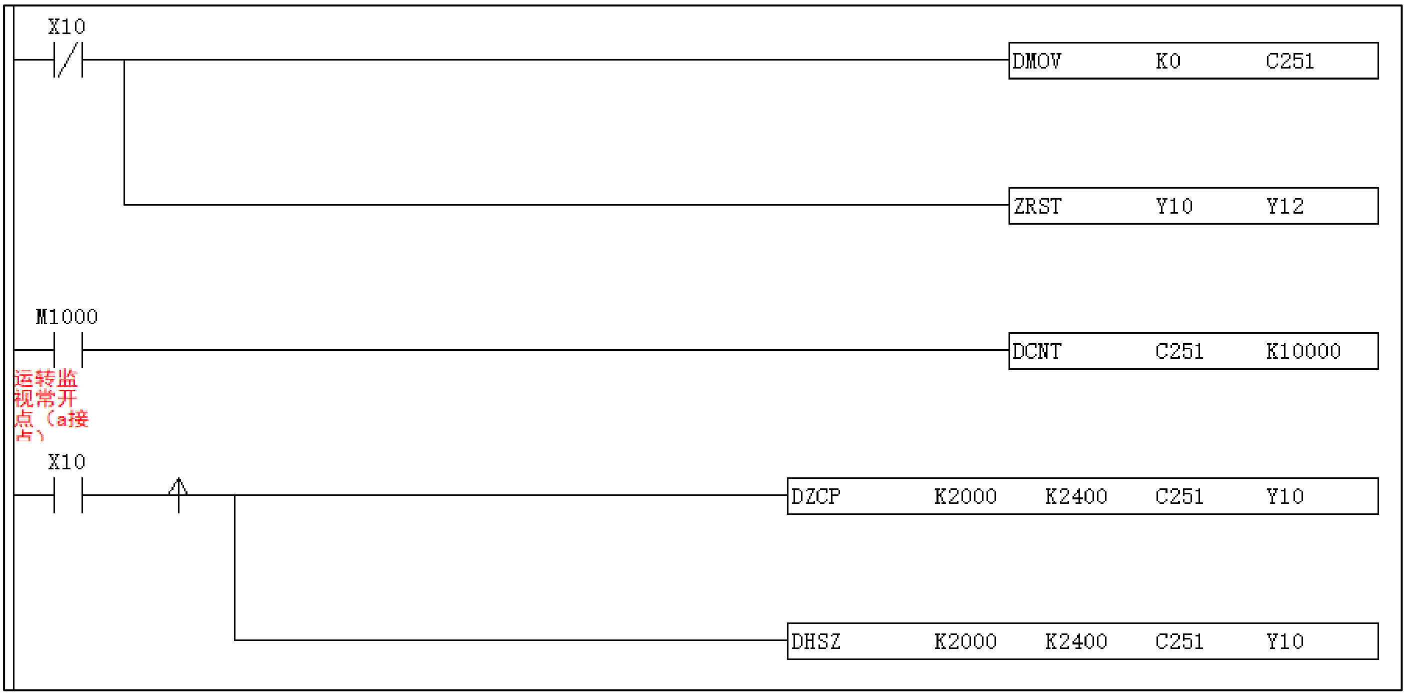
2、Program Example 1:

- Designate device Y0 and Y0 ~ Y2 will be automatically occupied.
- When DHSZ instruction is being executed and the counting of the high speed counter C246 reaches upper and lower bounds, one of Y0 ~ Y2 will be On.

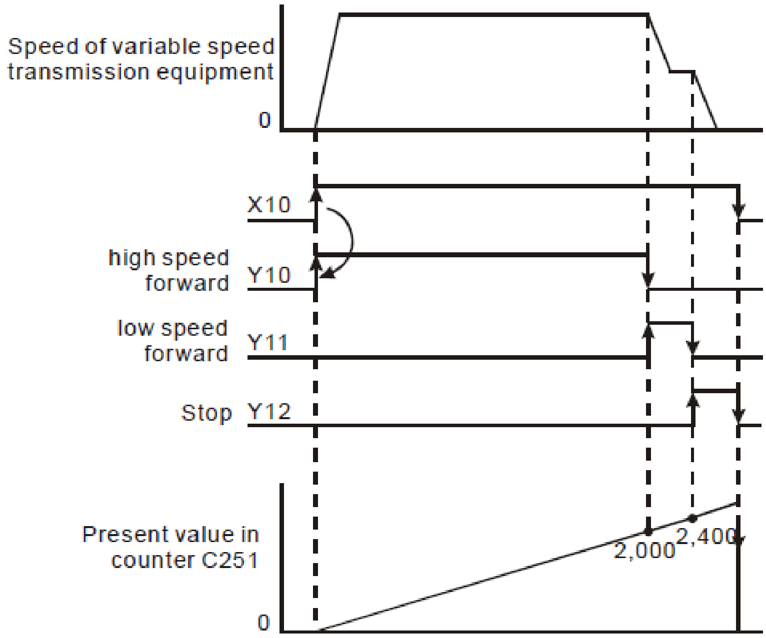


3、Program Example 2:

- Use DHSZ instruction for high/low speed stop control. C251 is an A-B phase high speed counter and DHSZ only performs comparison output when there is a C251 counting pulse input. Therefore, even when the present value in the counter is 0, Y10 will not be On.
- When X10 = On, DHSZ will require that Y10 has to be On when the present value in the counter ≤ K2,000. To solve this requirement, you can execute DZCPP instruction when the program was first RUN and compare C251 with K2,000. When the present value in the counter ≤ K2,000, Y10 will be On. DZCPP instruction is a pulse execution instruction and will only be executed once with Y10 being kept On.
- When the drive contact X10 = Off, Y10 ~ Y12 will be reset to Off.



- The timing diagram



4、Program Example 3:

- The multiple set values comparison mode: If D of DHSZ instruction designates a special auxiliary relay M1150, the instruction will be able to compare (output) the present value in the high speed counter with many set values.
- In this mode: S1: start device in the comparison table. S1 can only designate data register D and can be modified by E and F. Once this mode is enabled, S1 will not be changed even the E and F has been changed.

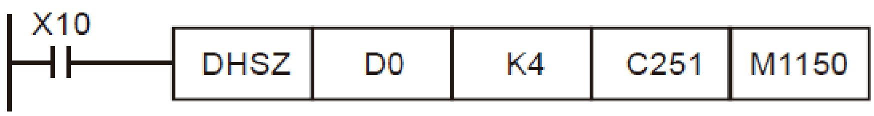
  S2: number of group data to be compared. S2 can only designate K1 ~ K255 or H1 ~ HFF and can be modified by E and F. Once this mode is enabled, S2 cannot be changed. If S2 is not within its range, error code 01EA (hex) will display and the instruction will not be executed.

  S: No. of high speed counter (designated as C241 ~ C254).

  D: Designated mode (can only be M1150)
- The No. of start register designated in S1 and the number of rows (groups) designated in S2 construct a comparison table. Please enter the set values in every register in the table before executing the instruction.
- When the present value in the counter C251 designated in S equals the set values in D1 and D0, the Y output designated by D2 will be reset to Off (D3 = K0) or On (D3 = K1) and be kept. Output Y will be processed as an interruption. No. of Y output pointss are in decimal (range: 0 ~ 255). If the No. falls without the range,SET/RESET will not be enabled when the comparison reaches its target.
- When this mode is enabled, PLC will first acquire the set values in D0 and D1 as the target value for the first comparison section. At the same time, the index value displayed in D1150 will be 0, indicating that PLC

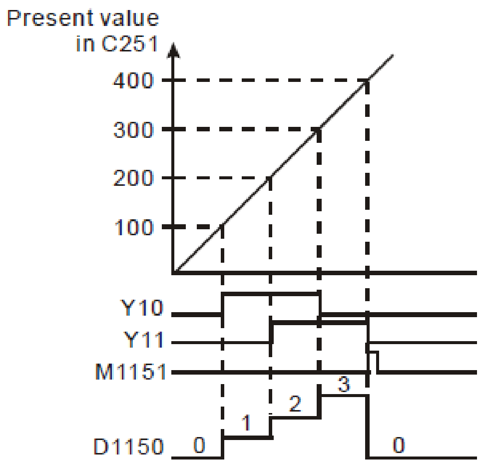
performs the comparison based on the group 0 data.

- When the group 0 data in the table have been compared, PLC will first execute the Y output set in group 0 data and determine if the comparison reaches the target number of groups. If the comparison reaches the target, M1151 will be On; if the comparison has not reached the final group, the content in D1150 will plus 1 and continue the comprison for the next group
- M1151 is the flag for the completion of one execution of the table, can be Off by the user. Or when the next comparion cycle takes place and the group 0 data has been compared, PLC will automatically reset the flag.
- When the drive contact of the instruction X10 goes Off, the execution of the instruction will be interrupted and the content in D1150 (table counting register) will be reset to 0. However, the On/Off status of all outputs will be remained.
- When the instruction is being executed, all set values in the comparison table will be regarded as valid values only when the scan arrives at END instruction for the first time.
- This mode can only be used once in the program.
- This mode can only be used on the hardware high speed counters C241 ~ C254.
- When in this mode, the frequency of the input counting pulses cannot exceed 50kHz or the neighboring two groups of comparative values cannot differ by 1; otherwise there will not be enough time for the PLC to react and result in errors.



- The comparison table:

32-bit data for comparison		No. of Y output	On/Off indication	Table counting register D1150
High word	Low word			
D1 (K0)	D0 (K100)	D2 (K10)	D3 (K1)	0
D5 (K0)	D4 (K200)	D6 (K11)	D7 (K1)	1
D9 (K0)	D8 (K300)	D10 (K10)	D11 (K0)	2
D13 (K0)	D12 (K400)	D14 (K11)	D15 (K0)	3
		K10: Y10 K11: Y11	K0: Off K1: On	0→1→2→3→0 Cyclic scan



- Special registers for flags and relevant settings:

Flag	Function
M1150	DHSZ instruction in multiple set values comparison mode
M1151	The execution of DHSZ multiple set values comparison mode is completed

Special D	Function
D1150	Table counting register for DHSZ multiple set values comparison mode

5、Program Example 4:

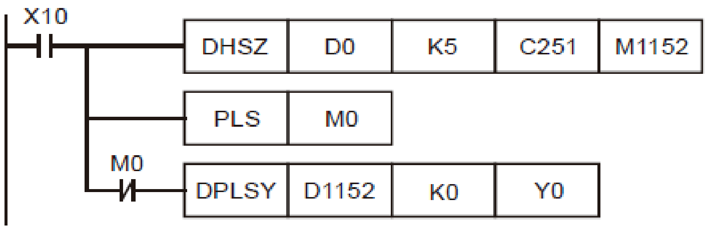
- DHSZ and DPLSY instructions are combined for frequency control. If D of DHSZ instruction is a special auxiliary relay M1152, the present value in the counter will be able to control the pulse output frequency of DPLSY instruction.
- In this mode: S1: start device in the comparison table. S1 can only designate data register D and can be modified by E and F. Once this mode is enabled, S1 will not be changed even the E and F has been changed

  S2: number of group data to be compared. S2 can only designate K1 ~ K255 or H1 ~ HFF and can be modified by E and F. Once this mode is enabled, S2 cannot be changed. If S2 is not within its range, error code 01EA (hex) will display and the instruction will not be executed.

  S: No. of high speed counter (designated as C241 ~ C254).

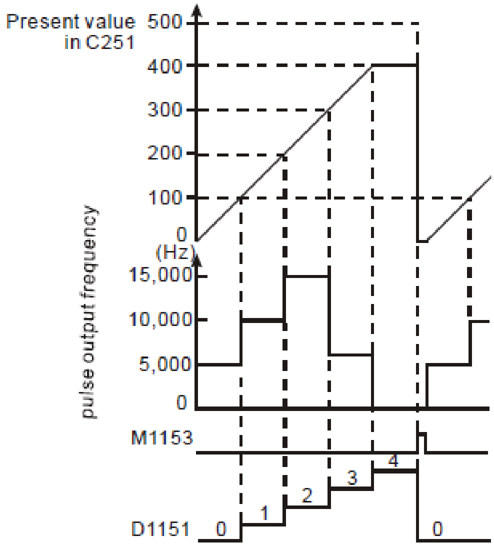
  D: Designated mode (can only be M1152)
- This mode can only be used once. this mode can only be used in the hardware high speed counter C241 ~ C254. Please enter the set values in every register in the table before executing the instruction.

- When this mode is enabled, PLC will first acquire the set values in D0 and D1 as the target value for the first comparison section. At the same time, the index value displayed in D1152 will be 0, indicating that PLC performs the comparison based on the group 0 data.
- When the group 0 data in the table have been compared, PLC will first execute at the frequency set in group 0 data (D2, D3) and copy the data to D1152 and D1153, determining if the comparison reaches the target number of groups. If the comparison reaches the target, M1153 will be On; if the comparison has not reached the final group, the content in D1151 will plus 1 and continue the comprison for the next group.
- M1153 is the flag for the completion of one execution of the table, can be Off by the user. Or when the next comparion cycle takes place and the group 0 data has been compared, PLC will automatically reset the flag.
- If you wish to use this mode with PLSY instruction, please preset the value in D1152.
- If you wish to stop the execution at the last row, please set the value in the last row K0.
- When the drive contact of the instruction X10 goes Off, the execution of the instruction will be interrupted and the content in D1151 (table counting register) will be reset to 0.
- When in this mode, the frequency of the input counting pulses cannot exceed 50kHz or the neighboring two groups of comparative values cannot differ by 1; otherwise there will not be enough time for the PLC to react and result in errors.



- The comparison table:

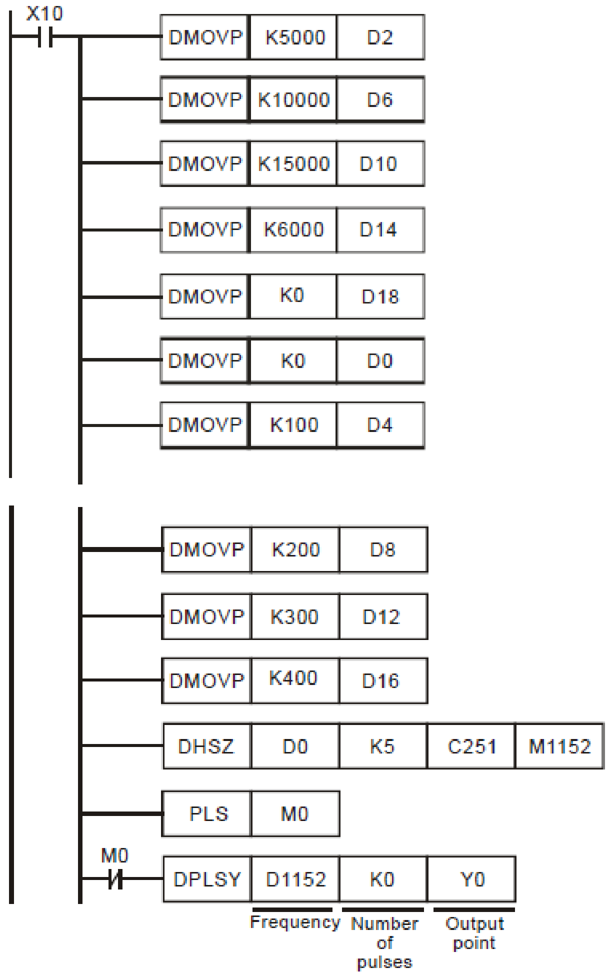
32-bit data for comparison		Pulse output frequency 0 ~ 200kHz	Table counting register D1151
High word	Low word		
D1 (K0)	D0 (K0)	D3, D2 (K5,000)	0
D5 (K0)	D4 (K100)	D7, D6 (K10,000)	1
D9 (K0)	D8 (K200)	D11, D10 (K15,000)	2
D13 (K0)	D12 (K300)	D15, D14 (K6,000)	3
D17 (K0)	D16 (K400)	D19, D18 (K0)	4
			0→1→2→3→4 Cyclic scan



- Special registers for flags and relevant settings:

Flag	Function
M1152	DHSZ instruction in frequency control mode
M1153	The execution of DHSZ frequency control mode is completed

Special D	Function
D1151	Table counting register for DHSZ multiple set values comparison mode
D1152 (low word) D1153 (high word)	In frequency control mode, DHSZ reads the upper and lower limits in the table counting register D1153 and D1152.
D1648 (low word) D1649 (high word)	Current number of pulses output by DPLSY instruction



- During the execution of DHSZ instruction, do not modify the set values in the comparison table.
- The designated data will be arranged into the the above program diagram when the program executes to END instruction. Therefore, PLSY instruction has to be executed after DHSZ instruction has been executed once.

ZL 56	SPD	S1 S2 D	Speed Detection

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1	*															PD: 7 steps 16-bit
S2					*	*	*	*	*	*	*	*	*	*	*	
D											*	*	*			

1、Explanations:

- S1: External pulse input terminal S2: Pulse receiving time (ms) D: Detected result
- The received number of pulses of the input terminal designated in S1 is calculated within the time (in ms) designated in S2. The result is stored in the register designated in D.
- D will occupy 5 consecutive devices. D + 1 and D are the detected value obtained from the previous pulses; D+3 and D + 2 are the current accumulated number of values; D + 4 is the counting time remaining (max. 32,767ms).
- This instruction is mainly used for obtaining a proportional value of rotation speed. The result D and rotation speed will be in proportion. The following equation is for obtaining the rotation speed of motor.

$$N = \frac{60(D0)}{nt} \times 10^3 (rpm)$$

N: Rotation speed

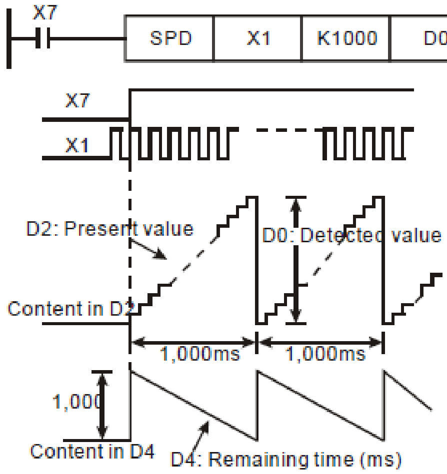
n: The number of pulses produced per rotation

t: Detecting time designated in **S2** (ms)

- The X input point designated by this instruction cannot be used again as the pulse input terminal of the high speed counter or as an external interruption signal.
- There is no limitation on the times of using this instruction in the program, but only one instruction will be executed at a time.

2、Program Example:

- When X7 = On, D2 will calculate the high-speed pulses input by X1 and stop the calculation automatically after 1,000ms. The result will be stored in D0.
- When the 1,000ms counting is completed, D2 will be cleared to 0. When X7 is On again, D2 will start the calculation again.


X7
SPD
X1
K1000
D0
X7
X1
D2: Present value
D0: Detected value
Content in D2
1,000ms
1,000ms
1,000
Content in D4
D4: Remaining time (ms)

<table>
<tr><td>ZL 57</td><td></td><td colspan="4" rowspan="2">PLSY</td><td></td><td colspan="9" rowspan="2">S1 S2 D</td><td rowspan="2">Pulse Y Output</td></tr>
<tr><td></td><td>D</td><td></td></tr>
<tr><td></td><td colspan="4">Bit Devices</td><td colspan="11">Word Devices</td><td rowspan="5">PLSY: 7 steps 16-bit
DPLSY: 13 steps 32-bit</td></tr>
<tr><td></td><td>X</td><td>Y</td><td>M</td><td>S</td><td>K</td><td>H</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>T</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td>S1</td><td>*</td><td></td><td></td><td></td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td></tr>
<tr><td>S2</td><td></td><td></td><td></td><td></td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td></tr>
<tr><td>D</td><td></td><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

1、Explanations:

- S1: Pulse output frequency module) S2: Number of output pulses
  D: Pulse output device (please use transistor output
- S2 designates the number of output pulses. The 16-bit instruction can designate 1 ~ 32,767 pulses and the 32- bit instruction can designate 1 ~ 2,147,483,647 pulses。
- When the PLSY instruction is used in the program, the output cannot be duplicated with the output of the ZL 58 PWM instruction and the ZL 59 PLSR instruction.
- Number of continuous pulses for all series:

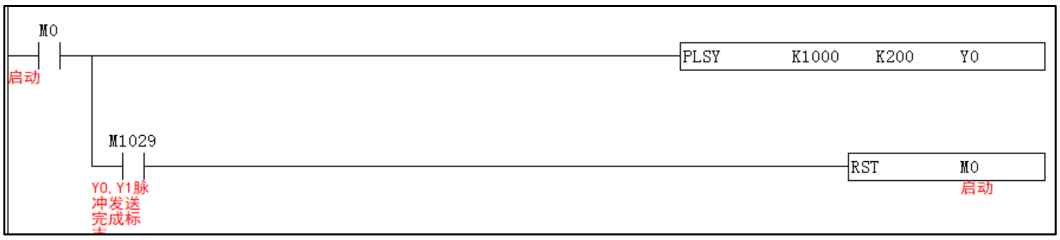
Group No	PUL	DIR	current number of output pulses (32-bit integer)	Pulse complete flag
CH0（Y0,Y1）	Y0	Y1	D1648	M1029
CH1（Y2,Y3）	Y2	Y3	D1664	M1030
CH2（Y4,Y5）	Y4	Y5	D1680	M1036
CH3（Y6,Y7）	Y6	Y7	D1696	M1037
CH4（Y10,Y11）	Y10	Y11	D1712	M1102
CH5（Y12,Y13）	Y12	Y13	D1728	M1103
CH6（Y14,Y15）	Y14	Y15	D1744	M1104
CH7（Y16,Y17）	Y16	Y17	D1760	M1105
CH8（Y20,Y21）	Y20	Y21	D1776	M1106
CH9（Y22,Y23）	Y22	Y23	D1792	M1107
CH10（Y24,Y25）	Y24	Y25	D1808	M1108
CH11（Y26,Y27）	Y26	Y27	D1824	M1109
CH12（Y30,Y31）	Y30	Y31	D1840	M1110
CH13（Y32,Y33）	Y32	Y33	D1856	M1111
CH14（Y34,Y35）	Y34	Y35	D1872	M1112
CH15（Y36,Y37）	Y36	Y37	D1888	M1113
CH16（Y40,Y41）	Y40	Y41	D1904	M1114
CH17（Y42,Y43）	Y42	Y43	D1920	M1115
CH18（Y44,Y45）	Y44	Y45	D1472	M1116
CH19（Y46,Y47）	Y46	Y47	D1488	M1117
CH20（Y50,Y51）	Y50	Y51	D1504	M1118

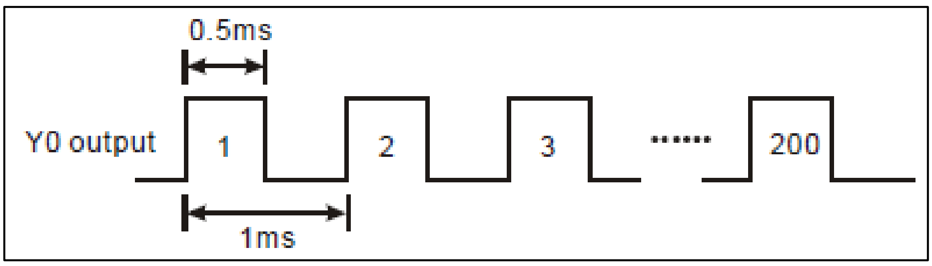
CH21（Y52,Y53)	Y52	Y53	D1520	M1119
CH22（Y54,Y55)	Y54	Y55	D1536	M1205
CH23（Y56,Y57)	Y56	Y57	D1552	M1206

- When PLSY instruction is executed, it will designate the number of output pulses (S2) output from the output device (D) at a pulse output frequency (S1).
- When PLSY instruction is used in the program, its outputs cannot be the same as those in ZL 58 PWM and ZL 59 PLSR.
- when PLSY and DPLSY instruction is disabled, the user will have to reset the pulse output completed flags.
- The user has to reset the pulse output completed flags after the pulse output is completed.
- After PLSY instruction starts to be executed, Y will start a pulse output. Modifying S2 at this moment will not affect the current output. If you wish to modify the number of output pulses, you have to first stop the execution of PLSY instruction and modify the number.
- S1 can be modified when the program executes to PLSY instruction.
- Off time : On time of the pulse output = 1 : 1.
- When the program executes to PLSY instruction, the current number of output pulses will be stored in the special data registers. See remarks for more details.

2、Program Example:

- When X0 = On, there will be 200 pulses output from Y0 at 1kHz. When the pulse output is completed, M1029 will be On and Y10 will be On.
- When X0 = Off, the pulse output from Y0 will stop immediately. When X0 is On again, the output will start again ffrom the first pulse.
- If the frequency needs to be sent all the time, write 0 to S2.





Remarks:

Group No	PUL	DIR	current number of output pulses (32-bit integer)	Pulse complete flag	Pulse sending	Emergency stop without slowing down
CH0（Y0,Y1)	Y0	Y1	D1648	M1029	M1344	M1308
CH1（Y2,Y3)	Y2	Y3	D1664	M1030	M1345	M1309
CH2（Y4,Y5)	Y4	Y5	D1680	M1036	M1346	M1310
CH3（Y6,Y7)	Y6	Y7	D1696	M1037	M1347	M1311
CH4（Y10,Y11)	Y10	Y11	D1712	M1102	M1348	M1312
CH5（Y12,Y13)	Y12	Y13	D1728	M1103	M1349	M1313
CH6（Y14,Y15)	Y14	Y15	D1744	M1104	M1350	M1314
CH7（Y16,Y17)	Y16	Y17	D1760	M1105	M1351	M1315
CH8（Y20,Y21)	Y20	Y21	D1776	M1106	M1352	
CH9（Y22,Y23)	Y22	Y23	D1792	M1107	M1353	
CH10（Y24,Y25)	Y24	Y25	D1808	M1108	M1354	
CH11（Y26,Y27)	Y26	Y27	D1824	M1109	M1355	
CH12（Y30,Y31)	Y30	Y31	D1840	M1110	M1356	
CH13（Y32,Y33)	Y32	Y33	D1856	M1111	M1357	
CH14（Y34,Y35)	Y34	Y35	D1872	M1112	M1358	
CH15（Y36,Y37)	Y36	Y37	D1888	M1113	M1359	
CH16（Y40,Y41)	Y40	Y41	D1904	M1114	M1360	
CH17（Y42,Y43)	Y42	Y43	D1920	M1115	M1361	
CH18（Y44,Y45)	Y44	Y45	D1472	M1116	M1362	
CH19（Y46,Y47)	Y46	Y47	D1488	M1117	M1363	
CH20（Y50,Y51)	Y50	Y51	D1504	M1118	M1364	
CH21（Y52,Y53)	Y52	Y53	D1520	M1119	M1365	
CH22（Y54,Y55)	Y54	Y55	D1536	M1205	M1366	
CH23（Y56,Y57)	Y56	Y57	D1552	M1206	M1367	
Remarks			D1648:Low word of the current number of output pulses from CH0. D1649:High word of the current number of output pulses from CH0.	After CH0-CH23 pulse output is completed, the corresponding flag bit is ON	Only when the pulse is being sent, the flag bit corresponding to CH0-CH23 is ON	Off->On: The high-speed pulse output pauses immediately. On->Off: Continuing to output the pulses which have not been output

ZL 58		PWM		S1 S2 D											Pulse Width Modulation
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*	*	PWM: 7 steps 16-bit
S2					*	*	*	*	*	*	*	*	*	*	*	
D		*														

1、Explanations:

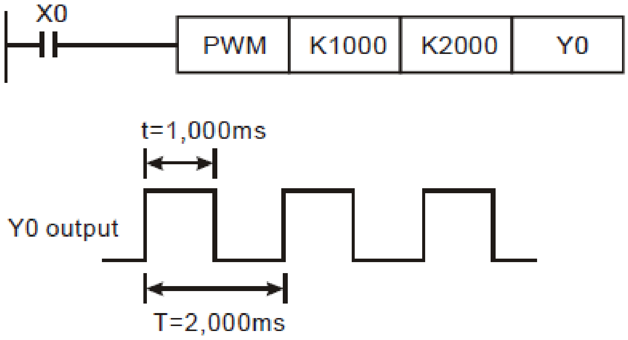
- S1: Pulse output width S2: Pulse output period D: Pulse output device (please use transistor output module)
- S1 ⩽ S2
- Range of S1: (t) 0 ~ 32,767ms. (refer to the remarks for more information about the time unit settings.)
- Range of S2: (T) 1 ~ 32,767ms (but S1 ⩽ S2).
- Pulse output device

Output point	Y0、Y2、Y4、Y6……Y24、Y26

- When PWM instruction is used in the program, its outputs cannot be the same as those of API 57 PLSY, API 59 PLSR or other positioning instructions.
- PWM instruction designates the pulse output width in S1 and pulse output period in S2 and outputs from output device D.
- When, S1 < 0 or S2 ⩽ 0 or S1 > S2, there will be operational errors , and there will be no output from the pulse output device. When S1 = 0, there will be no output from the pulse output device. When S1 = S2, the the pulse output device will keep being On.
- S1 and S2 can be changed when PWM instruction is being executed.

2、Program Example:

- When X0 = On, Y0 will output the pulses as below. When X0 = Off, Y0 output will also be Off


X0
PWM
K1000
K2000
Y0
t=1,000ms
Y0 output
T=2,000ms

ZL 59	D	PLSR			S1 S2 S3 D											Pulse Ramp
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*	*	PLSR: 9 steps 16-bit
S2					*	*	*	*	*	*	*	*	*	*	*	DPLSR: 17 steps 32-bit
S3					*	*	*	*	*	*	*	*	*	*	*	
D		*														

1、Explanations:

- S1: Maximum speed of pulse output S2: Total number of output pulses
  S3: Acceleration/deceleration time (ms) D: Pulse output device (please use transistor output module PLC)
- Range of S1: 10 ~ 32,767Hz (16-bit); 10 ~ 200,000Hz (32-bit). The maximum speed has to be 10’ s multiple; if not, the 1s digit will be left out. 1/10 of the maximum speed is the variation of one acceleration or deleration. Please be aware if the variation reponds to the acceleration/deceleration demand from the step motor, in case the step motor may crash.
- Range of S2: 110 ~ 32,767 (16-bit); 110 ~ 2,147,483,647 (32-bit). If S2 is less than 110, the pulet output will be abnormal.
- Range of S3: below 5,000ms. The acceleration time and deceleration time have to be the same.
- Refer to the related section in explanation of PLSY instruction for D devices and maximum frequency.
- PLSR instruction is a pulse output instruction with acclerating and decelerating functions. The pulses accelerate from the static status to target speed and decelerates when the target distance is nearly reached. The pulse output will stop when the target distance is reached. S2 and S3 can be changed when PLSR instruction is being executed.

Group No	PUL	DIR	current number of output pulses (32-bit integer)	Pulse complete flag	Pulse sending	Emergency stop without slowing down
CH0（Y0,Y1)	Y0	Y1	D1648	M1029	M1344	M1308
CH1（Y2,Y3)	Y2	Y3	D1664	M1030	M1345	M1309
CH2（Y4,Y5)	Y4	Y5	D1680	M1036	M1346	M1310
CH3（Y6,Y7)	Y6	Y7	D1696	M1037	M1347	M1311
CH4（Y10,Y11)	Y10	Y11	D1712	M1102	M1348	M1312
CH5（Y12,Y13)	Y12	Y13	D1728	M1103	M1349	M1313
CH6（Y14,Y15)	Y14	Y15	D1744	M1104	M1350	M1314
CH7（Y16,Y17)	Y16	Y17	D1760	M1105	M1351	M1315
CH8（Y20,Y21)	Y20	Y21	D1776	M1106	M1352	M1316
CH9（Y22,Y23)	Y22	Y23	D1792	M1107	M1353	M1317
CH10（Y24,Y25)	Y24	Y25	D1808	M1108	M1354	M1318
CH11（Y26,Y27)	Y26	Y27	D1824	M1109	M1355	M1319
CH12（Y30,Y31)	Y30	Y31	D1840	M1110	M1356	M1320
CH13（Y32,Y33)	Y32	Y33	D1856	M1111	M1357	M1321
CH14（Y34,Y35)	Y34	Y35	D1872	M1112	M1358	M1322
CH15（Y36,Y37)	Y36	Y37	D1888	M1113	M1359	M1323
CH16（Y40,Y41)	Y40	Y41	D1904	M1114	M1360	M1324
CH17（Y42,Y43)	Y42	Y43	D1920	M1115	M1361	M1325
CH18（Y44,Y45)	Y44	Y45	D1472	M1116	M1362	M1326
CH19（Y46,Y47)	Y46	Y47	D1488	M1117	M1363	M1327
CH20（Y50,Y51)	Y50	Y51	D1504	M1118	M1364	M1328
CH21（Y52,Y53)	Y52	Y53	D1520	M1119	M1365	M1329
CH22（Y54,Y55)	Y54	Y55	D1536	M1205	M1366	M1330
CH23（Y56,Y57)	Y56	Y57	D1552	M1206	M1367	M1331
Remarks			D1648:Low word of the current number of output pulses from CH0. D1649:High word of the current number of output pulses from CH0.	After CH0-CH23 pulse output is completed, the corresponding flag bit is ON	Only when the pulse is being sent, the flag bit corresponding to CH0-CH23 is ON	Off->On：The high-speed pulse output pauses immediately. On->Off: Continuing to output the pulses which have not been output

- when all the CH0 (Y0, Y1) pulses have been sent, M1029 will be On; when all the CH1 (Y2, Y3) pulses have been sent, M1030 will be On; when CH2 (Y4, Y5) pulses have been sent, M1036 will be On; when CH3 (Y6, Y7) pulses have been sent, M1037 will be On. When all the CH4 (Y10, Y11) pulses have been sent, M1102 will be On. When all the CH5 (Y12, Y13) pulses have been sent, M1103 will be On. Next time when PLSR instruction is enabled, M1029, M1030, M1036, M1037, M1102 and M1103 will be 0 again and after the pulse output is completed, they will become 1 again.Other pulse output can be deduced by analogy, the flag bit is detailed in the table above.

- During every acceleration section, the number of pulses (frequency × time) may not all be integers. PLC will round up the number to an integer before the output. Therefore, the acceleration time of every section may not be exactly the same. The offset is determined upon the frequency and the decimal after rounding up. In order to ensure the correct number of output pulses, PLC will supplement insufficient pulses in the last section.
- For the limitation on the times using this instruction in the program, refer to PLSY instruction for more information.
- Range of **S_3**: below 5,000ms. The acceleration time and deceleration time have to be the same.

  a. The acceleration and deceleration time must be more than 10 times the maximum scan period (the contents of D1012), and if the value is set to less than 10x, the slope of the acceleration and deceleration will be incorrect.

  b. The minimum setting value for the acceleration and deceleration time can be determined by the following formula.

$$\boxed{S_3 \geq \frac{90,000}{S_1}}$$

If the setting value is less than the result of the above calculation formula, the acceleration and deceleration time will become larger, and if the setting value is less than 90000/S1, the result value of 90000/S1 will be used as the setting value.

c. The maximum setting value for the acceleration and deceleration time can be determined by the following formula.

$$\boxed{S_3 \leq \frac{S_2}{S_1} \times 818}$$

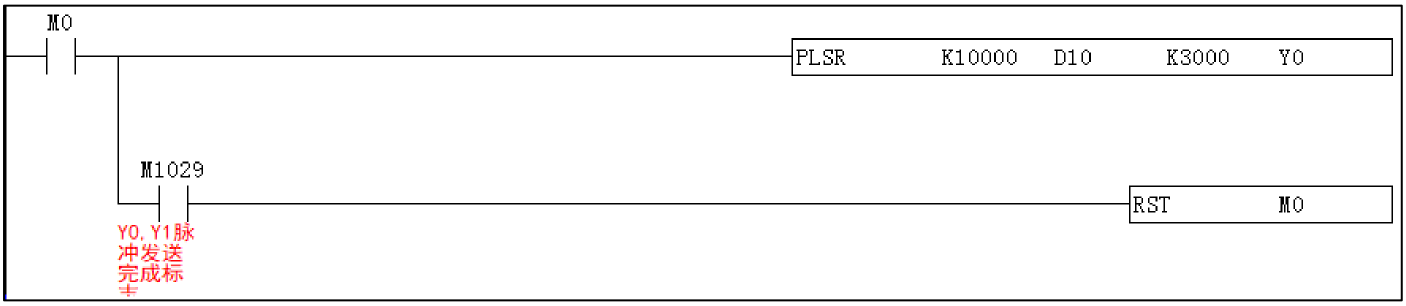
d. The number of variable speed segments for acceleration and deceleration is fixed at 10 segments. If the input acceleration and deceleration time is greater than the maximum setting value, the maximum setting time will prevail, and if it is less than the minimum setting value, the minimum setting value will be the main value.

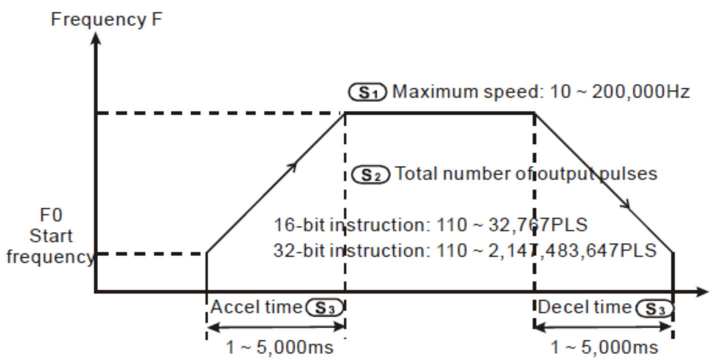
D pulse output device, additional deceleration pulse output device

Refer to the output of the modulation pulse table	
PLSR output	Y0、Y2、Y4、Y6……Y54、Y56

2、Example:

- When M0=On, the PLSR instruction is executed with the maximum frequency value of pulse output 1,000Hz, the total pulse number of all pulse output D10, and the acceleration/deceleration time 3,000ms, then Y0 outputs pulses. Start outputting pulses at a frequency of 1,000/10 Hz each time. The time of each frequency output pulse is fixed 3,000/9 (ms).
- When M0 turns Off, the output is interrupted. When X0 turns On again, the pulse count starts from 0.

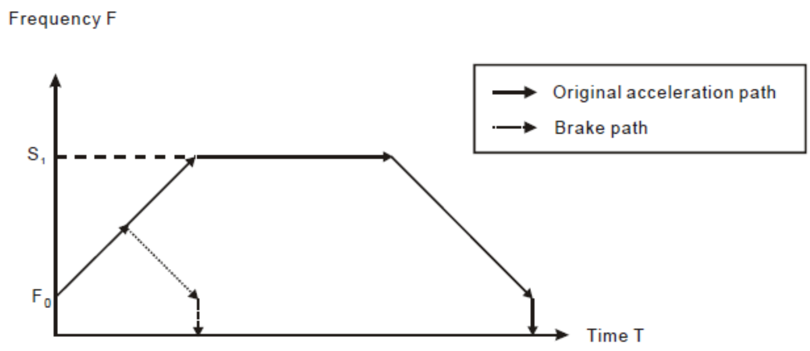




Remarks:

- Based on the number of pulses. If the output cannot reach the maximum acceleration frequency within the acceleration/deceleration time offered, the instruction will automatically adjust the acceleration/deceleration time and the maximum frequency.The operands have to be set before the execution of the instruction PLSR. You cannot change the acceleration/deceleration during the instruciton execution.

- All acceleration/deceleration instructions are included with the brake function. The brake function will be enabled when PLC is performing acceleration and the switch contact is suddenly Off. The deceleration will operate at the slope of the acceleration。



7.2（ZL 60-69）Convenience instructions

ZL 61	D	SER		S1 S2 D n											Search a Data Stack
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1							*	*	*	*	*	*	*		
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*		
n					*	*							*		

SER: 9 steps 16-bit

DSER: 17 steps 32-bit

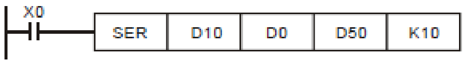
1、Explanations:

- S1: Start device for data stack comparison S2: Data to be compared

  D: Start device for storing comparison result n: Length of data to be compared
- When S2 are used in device F, only 16-bit instruction is applicable.
- D will occupy 5 consecutive points.
- Range of n: for 16-bit instruction 1 ~ 256; for 32-bit instruction 1 ~ 128.
- The n data in the registers starting from S1 are compared with S2 and the results are stored in the registers starting from D.
- In the 32-bit instruction, S1, S2, D and n will designate 32-bit registers.
- For D, the 16-bit counters and 32-bit counters cannot be mixed when being used

2、Program Example:

- When X0 = On, the data stack consist of D10 ~ D19 will be compared against D0 and the result will be stored in D50 ~ D52. If there are equivalent values appearing during the comparison, D50 ~ D52 will all be 0.
- The data are compared algebraically. (-10 < 2).
- The No. of the register with the smallest value among the compared data will be recorded in D53; the biggest will be recorded in D54. When there are more than one smallest value or biggest value, device D will record the No. of the register with bigger value.



n

S_1	Content	Data to be compared	Data No.	Result
D10	88	S_2	0	
D11	100		1	Equal
D12	110		2	
D13	150		3	
D14	100	D0 = K100	4	Equal
D15	300		5	
D16	100		6	Equal
D17	5		7	Smallest
D18	100		8	Equal
D19	500		9	Biggest

D	Content	Description
D50	4	Total number of data with equivalent values
D51	1	No. of the first equivalent value
D52	8	No. of the last equivalent value
D53	7	No. of the smallest value
D54	9	No. of the biggest value

ZL 62	D	ABSD	S1 S2 D n	Absolute Drum Sequencer

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1							*	*	*	*	*	*	*			ABSD: 9 steps 16-bit
S2											*	*	*			DABSD: 17 steps 32-bit
D		*	*	*												
n					*	*										

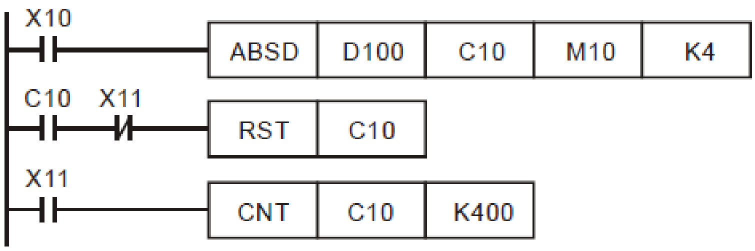
1、Explanations:

- S1: Start device in the data table S2: No. of counter

  D: Start No. of the devices for the comparison results n: Number of data for comparison
- When S1 designates KnX, KnY, KnM and KnS, the 16-bit instruction has to designate K4 and 32-bit instruction has to designate K8.
- Range of n: 1 ~ 64
- ABSD instruction is for the absolute control of the multiple output pulses generated by the present value in the counter.
- S2 of DABSD instruction can designate high speed counters. However, when the present value in the high speed counter is compared with the target value, the result cannot output immediately owing to the scan time. If an immediate output is required, please use DHSZ instruction that is exclusively for high speed counters.

2、Program Example:

- Before the execution of ABSD instruction, use MOV instruction to write all the set values into D100 ~ D107 in advance. The even-number D is for lower bound value and the odd-number D is for upper bound value.
- When X10 = On, the present value in counter C10 will be compared with the four groups of lower and upper bound values in D100 ~ D107. The comprison results will be stored in M10 ~ M13.
- When X10 = Off, the original On/Off status of M10 ~ M13 will be remained.

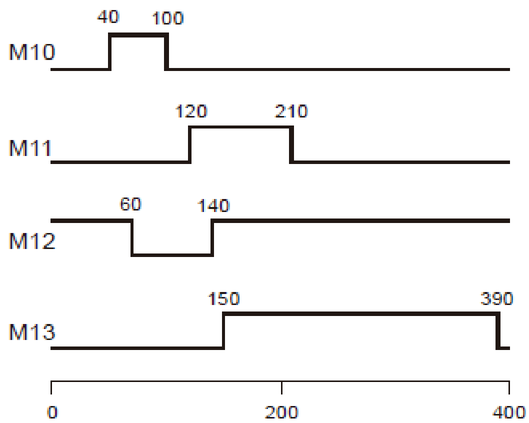


- M10~ M13 will be On when the present value in C10 ≦ upper bound value or ≧ lower bound value.

Lower bound value	Upper bound value	Present value in C10	Output
D100 = 40	D101 = 100	40 ≦ C10 ≦ 100	M10 = On
D102 = 120	D103 = 210	120 ≦ C10 ≦ 210	M11 = On
Lower bound value	Upper bound value	Present value in C10	Output
D104 = 140	D105 = 170	140 ≦ C10 ≦ 170	M12 = On
D106 = 150	D107 = 390	150 ≦ C10 ≦ 390	M13 = On

- If the lower bound value > upper bound value, when C10 < upper bound value (60) or > upper bound value (140), M12 will be On.

Lower bound value	Upper bound value	Present value in C10	Output
D100 = 40	D101 = 100	40 ≦ C10 ≦ 100	M10 = On
D102 = 120	D103 = 210	120 ≦ C10 ≦ 210	M11 = On
D104 = 140	D105 = 60	60 ≦ C10 ≦ 140	M12 = On
D106 = 150	D107 = 390	150 ≦ C10 ≦ 390	M13 = On



ZL 63	INCD	S1 S2 D n														Incremental Drum Sequencer
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1							*	*	*	*	*	*	*			INCD: 9 steps 16-bit
S2												*				
D		*	*	*												
n					*	*										

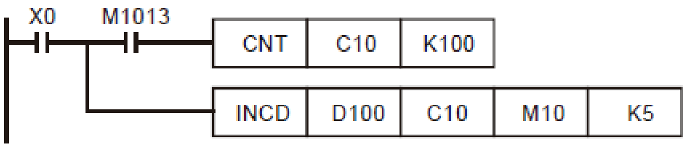
1、Explanations:

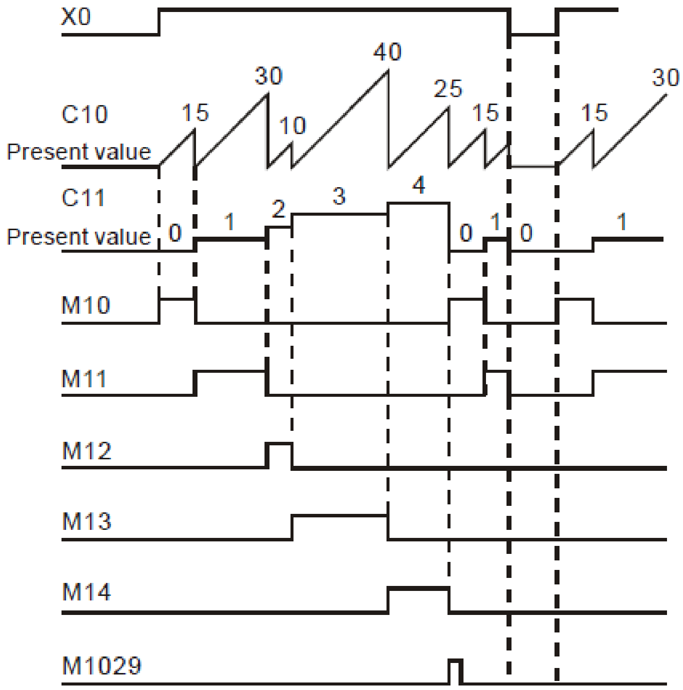
- S1: Start device in the data table S2: No. of counter

  D: Start No. of the devices for the comparison results n: Number of data for comparison
- When S1 designates KnX, KnY, KnM and KnS, it has to designate K4.
- In the 16-bit instruction, S2 has to designate C0 ~ C198 and will occupy 2 consecutive No. of counters.
- Range of n: 1 ~ 64
- INCD instruction is for the relative control of the multiple output pulses generated by the present value in the counter.
- The present value in S2 is compared with S1. S2 will be reset to 0 whenever a comparison is completed. The current number of data processed in temporarily stored in S2 + 1.

2、Program Example:

- Before the execution of INCD instruction, use MOV instruction to write all the set values into D100 ~ D104 in advance. D100 = 15, D101 = 30, D102 = 10, D103 = 40, D104 = 25.
- The present value in C10 is compared against the set values in D100 ~ D104. The present value will be reset to 0 whenever a comparison is completed.
- The current number of data having been processed is temporarily stored in C11.
- The number of times of reset is temporarily stored in C11.
- Whenever the content in C11 pluses 1, M10 ~ M14 will also correspondingly change. See the timing diagram below.
- When X0 goes from On to Off, C10 and C11 will both be reset to 0 and M10 ~ M14 will all be Off. When X0 is On again, the instruction will start its execution again from the beginning.


X0
M1013
CNT
C10
K100
INCD
D100
C10
M10
K5


X0
C10
Present value
C11
Present value
15
30
10
40
25
15
15
30
0
1
2
3
4
0
1
0
1
M10
M11
M12
M13
M14
M1029

ZL 64	TTMR	D n	Teaching Timer

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	TTMR: 5 steps 16-bit
D													*			
n					*	*										

1、Explanations:

- D: Device No. for storing the “On” time of button switch n: Multiple setting
- D will occupy 2 consecutive devices
- Range of n: 0 ~ 2
- TTMR instruction can be used 8 times in the program.
- The “On” time (unit: 100ms) of the external button switch is stored in device No. D + 1. The “On” time (unit: second) of the switch is multiplied by n and stored in D.
- Multiple setting:

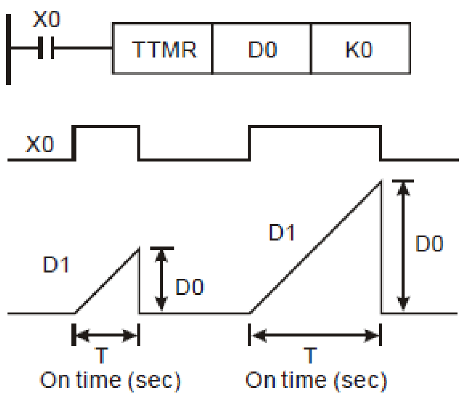
  When n = 0, unit of D = second

  When n = 1, unit of D = 100ms (D × 10)

  When n = 2, unit of D = 10ms (D × 100

2、Program Example 1:

- The “On” (being pressed) time of button switch X0 is stored in D1. The setting of n is stored in D0. Therefore, the button switch will be able to adjust the set value in the timer.
- When X0 goes Off, the content in D1 will be cleared to 0, but the content in D0 will remain.

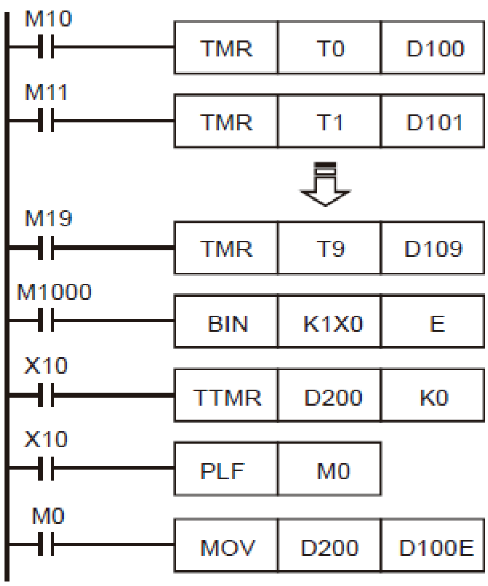


- Assume the “On” time of X0 is T (sec.), see the relation between D0, D1 and n in the table below.

n	D0	D1(unit: 100ms)
K0 (unit: s)	1×T	D1=D0×10
K1 (unit: 100 ms)	10×T	D1=D0
K2 (unit: 10 ms)	100×T	D1=D0/10

3、Program Example 2:

- Use TMR instruction to write in 10 groups of set time.
- Write the set values into D100 ~ D109 in advance.
- The timing unit for timer T0 ~ T9 is 0.1 sec. The timing unit for the teaching timer is 1 sec.
- Connect the 1-bit DIP switch to X0 ~ X3 and use BIN instruction to convert the set value of the switch into a bin value and store it in E.
- Store the "On" time (sec.) of X10 in D200.
- M0 refers to the pulses generated from one scan period after the button switch of the teaching timer X10 is released.
- Use the set number of the DIP switch as the indirectly designated pointer and send the content in D200 to D100E (D100 ~ D109).



Remarks:

- There is no limitation on the times using this instruction in the program and 8 instructions can be executed at the same time.

ZL 65	STMR	S m D	Special Timer

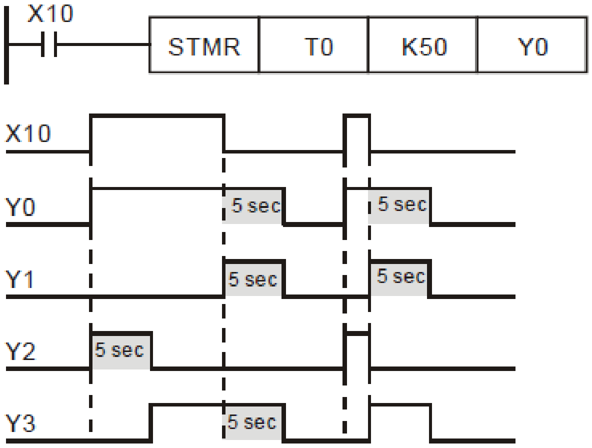
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*					STMR: 7 steps 16-bit
m					*	*										
D		*	*	*												

1、Explanations:

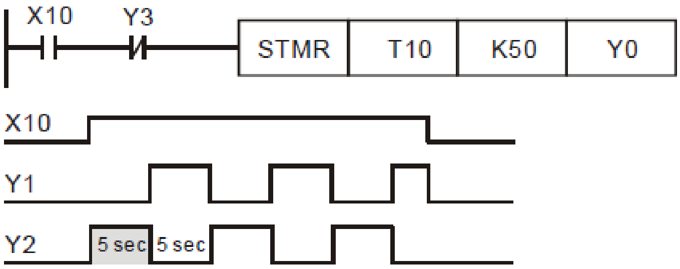
- S: No. of timer m: Set value in timer (unit: 100ms) D: No. of start output device
- Range of S: for T0 ~ T183.
- Range of m: 1 ~ 32,767
- D will occupy 4 consecutive devices.
- STMR instruction is used for Off-delay, one shot timer and flashing sequence.
- The No. of timers designated by STMR instructions can be used only once.

2、Program Example:

- When X10 = On, STMR instruction will designate timer T0 and set the set value in T0 as 5 seconds.
- Y0 is the contact of Off-delay. When X10 goes from Off to On, Y0 will be On. When X10 goes from On to Off, Y0 will be Off after a five seconds of delay.
- When X10 goes from On to Off, there will be a five seconds of Y1 = On output.
- When X10 goes from Off to On, there will be a five seconds of Y2 = On output.
- When X10 goes from Off to On, Y3 will be On after a five seconds of delay. When X10 goes from On to Off, Y3 will be Off after a five seconds of delay.



- Add a b contact of Y3 after X10, and Y1 and Y2 can operate for flashing sequence output. When X10 goes Off, Y0, Y1 and Y3 will be Off and the content in T10 will be reset to 0.



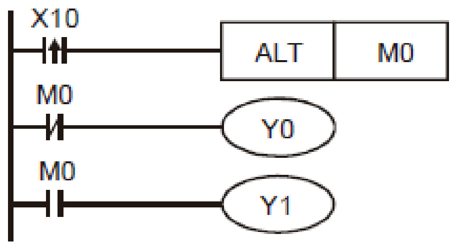
ZL 66	ALT		D													Alternate State
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ALT: 3 steps 16-bit
D		*	*	*												

1、Explanations:

- D: Destination device
- When ALT instruction is executed, "On" and "Off" of D will switch.
- This instruction adopts pulse execution instructions (ATLP).
- To execute the pulse type, add the NP rising edge "↑" command before the command.

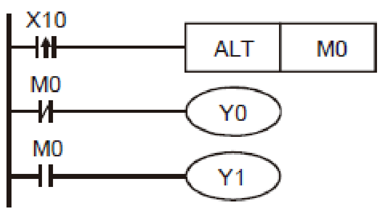
2、Program Example 1

- When X0 goes from Off to On, Y0 will be On. When X0 goes from Off to On for the second time, Y0 will be Off.



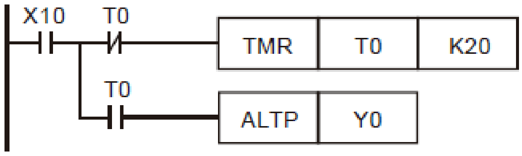
3、Program Example 2:

- Using a single switch to enable and disable control. At the beginning, M0 = Off, so Y0 = On and Y1 = Off. When X10 switches between On/Off for the first time, M0 will be On, so Y1 = On and Y0 = Off. For the second time of On/Off switching, M0 will be Off, so Y0 = On and Y1 = Off.



4、Program Example 3:

- Generate flashing. When X10 = On, T0 will generate a pulse every 2 seconds and Y0 output will switch between On and Off following the T0 pulses.


X10
T0
TMR
T0
K20
T0
ALTP
Y0

ZL 67	RAMP	S1 S2 D n	Ramp Variable Value

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1													*		
S2													*		
D													*		
n					*	*									

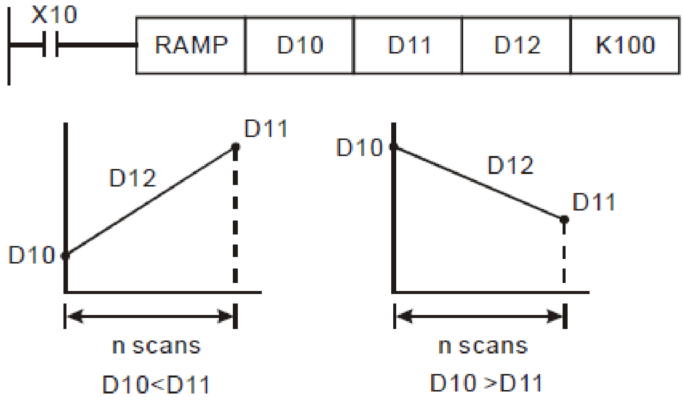
RAMP: 9 steps 16-bit

1、Explanations:

- S1: Start of ramp signal S2: End of ramp signal

  D: Duration of ramp signal n: Scan times
- Range of n: 1 ~ 32,767
- D will occupy 2 consecutive points.
- This instruction is for obtaining slope (the relation between linearity and scan time). Before using this instruction, you have to preset the scan time.
- The set value of start ramp signal is pre-written in D10 and set value of end ramp signal in D11. When X10 = On, D10 increases towards D11 through n (= 100) scans (the duration is stored in D12). The times of scans are stored in D13.
- In the program, first drive M1039 = On to fix the scan time. Use MOV instruction to write the fixed scan time to the special data register D1039. Assume the scan time is 30ms and take the above program for example, n = K100, the time for D10 to increase to D11 will be 3 seconds (30ms × 100).
- When X10 goes Off, the instruction will stop its execution. When X10 goes On again, the content in D12 will be reset to 0 for recalculation.
- When M1026 = Off, M1029 will be On and the content in D12 will be reset to the set value in D10.
- When this instruction is used with analog signal outputs, it will be able to buffer START and STOP.

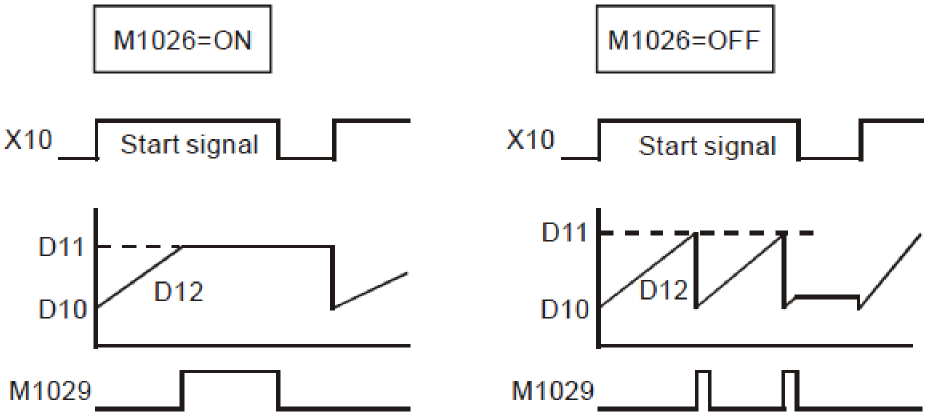
2、Program Example:



The scan times is stored in D13

- Remarks:

  D12 for enabling On/Off of M1026:



ZL 69				
	D	SORT	S_1 m_1 m_2 D n	Sort Tabulated Data

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1													*		
m1					*	*									
m2					*	*									
D													*		
n					*	*							*		

SORT: 11 steps 16-bit

DSORT: 21 steps 32-bit

1、Explanations:

- S: Start device for the original data m1: Groups of data to be sorted

  m2: Number of columns of data D: Start device for the sorted data

  n: Reference value for data sortin
- Range of m1: 1 ~ 32
- Range of m2: 1 ~ 6
- Range of n: 1 ~ m2
- The sorted result is stored in m1 × m2 registers starting from the device designated in D. Therefore, if S and D designate the same register, the sorted result will be the same as the data designated in S.
- It is better that the start No. designated in S is 0.
- There is no limitation on the times of using this instruction. However, only one instruction can be executed at a time.
- The function of sorting one-dimensional data is added. If users set m1 and m2 to 1, the function will be enabled. The operand n represents the number of data. It must be in the range of 1 to 32. The data in the n devices starting from S is sorted. The sorting result is stored in the devices starting from D. This function only needs one scan time. After data is sorted. M1029 will be ON.

2、Program Example:

- When X0 = On, the sorting will start. When the sorting is completed, M1029 will be On. DO NOT change the datato be sorted during the execution of the instruction. If you wish to change the data, please make X0 go from Off to On again.

X0	SORT	D0	K5	K5	D50	D100

- Example table of data sorting

Columns of data: m_2

Groups of data: m_1

Row \ Column	Data Column 1 Students No.	2 Physics	3 English	4 Math	5 Chemistry
1	(D0) 1	(D5) 90	(D10) 75	(D15) 66	(D20) 79
2	(D1) 2	(D6) 55	(D11) 65	(D16) 54	(D21) 63
3	(D2) 3	(D7) 80	(D12) 98	(D17) 89	(D22) 90
4	(D3) 4	(D8) 70	(D13) 60	(D18) 99	(D23) 50
5	(D4) 5	(D9) 95	(D14) 79	(D19) 75	(D24) 69

- Sorted data when D100 = K3

Columns of data: m_2

Groups of data: m_1

Row \ Column	Data Column 1 Students No.	2 Physics	3 English	4 Math	5 Chemistry
1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
3	(D52) 1	(D57) 90	(D62) 75	(D67) 66	(D72) 79
4	(D53) 5	(D58) 95	(D63) 79	(D68) 75	(D73) 69
5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

- Sorted data when D100 = K5

Columns of data: m_2

Groups of data: m_1

Row \ Column	Data Column 1 Students No.	2 Physics	3 English	4 Math	5 Chemistry
1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
3	(D52) 5	(D57) 95	(D62) 79	(D67) 75	(D72) 69
4	(D53) 1	(D58) 90	(D63) 75	(D68) 66	(D73) 79
5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

7.3 (ZL 70-79) External I/O device

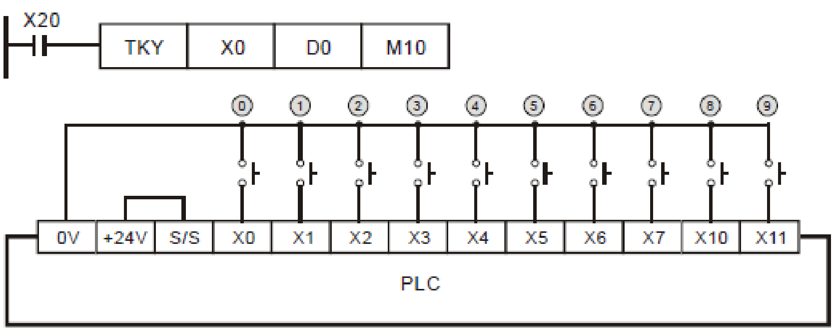
ZL 70		TKY		S D1 D2 n											Ten Key Input
	D														
	Bit Devices				Word Devices										TKY: 7 steps 16-bit DTKY: 13 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S	*	*	*	*											
D_1								*	*	*	*	*	*	*	*
D_2		*	*	*											

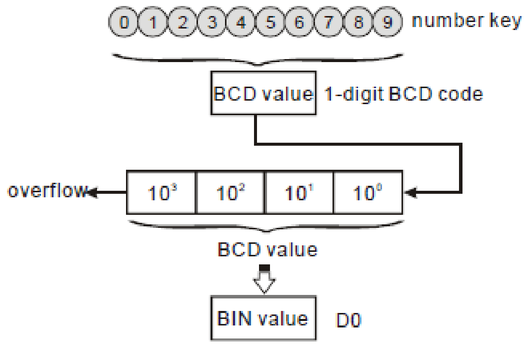
1、Explanations:

- S: Start device for key input D1: Device for storing keyed-in value D2: Key output signal
- S will occupy 10 consecutive points; D2 will occupy 11 consecutive points.
- This instruction designates 10 external input points (representing decimal numbers 0 ~ 9) starting from S. The 10 points are respectively connected to 10 keys. By pressing the keys, you can enter a 4-digit decimal figure 0 ~ 9,999 (16-bit instruction) or a 8-digit figure 0 ~ 99,999,999 (32-bit instruction) and store the figure in D1. D2 is used for storing key status.
- There is no limitation on the times of using this instruction. However, only one instruction can be executed at a time.

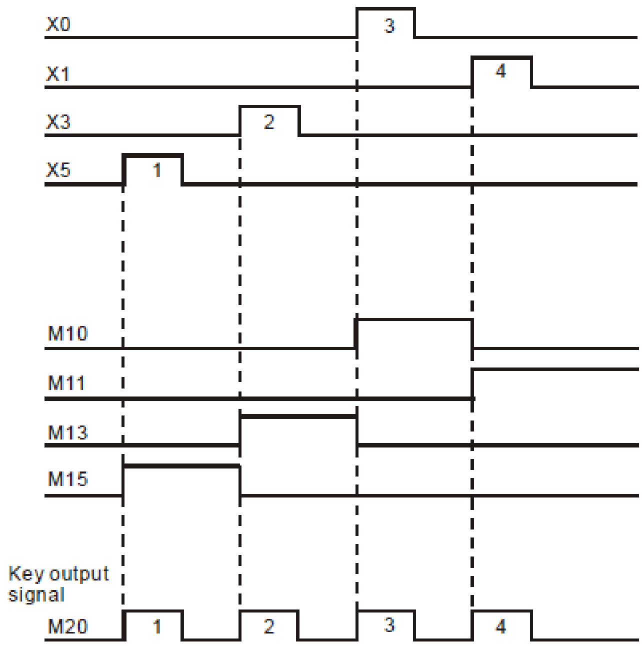
2、Program Example:

- Connect the 10 input points starting from X0 to the 10 keys (0 ~ 9). When X20 = On, the instruction will be executed and the keyed-in values will be stored in D0 in bin form. The key status will be stored in M10 ~ M19.





- As shown in the timing chart below, the 4 points X5, X3, X0, and X1 connected to the keys are entered in order and you can obtain the result 5,301. Store the result in D0. 9,999 is the maximum value allowed to stored in D0. Once the value exceeds 4 digits, the highest digit will overflow.
- M12 = On when from X2 is pressed to the other key is pressed. Same to other keys.
- When any of the keys in X0 ~ X11 is pressed, one of M10 ~ 19 will be On correspondingly.
- M20 = On when any of the keys is pressed.
- When X20 goes Off, the keyed-in value prior to D0 will remain unchanged, but M10 ~ M20 will all be Off.



ZL 71	D	HKY	S D_1 D_2 D_3	Hexadecimal Key Input

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S	*														
D_1		*													
D_2											*	*	*	*	*
D_3		*	*	*											

HKY: 9 steps 16-bit

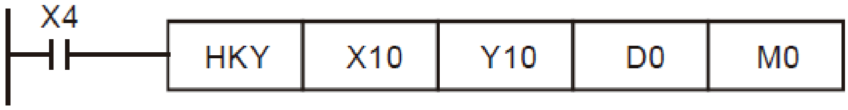
DHKY: 17 steps 32-bit

1、Explanations:

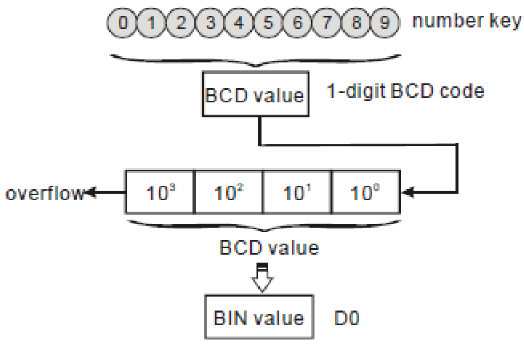
- S: Start device for key scan input D1: Start device for key scan output
  D2: Device for storing keyed-in value D3: Key output signal
- S will occupy 4 consecutive points.
- D1 will occupy 4 consecutive points.
- D3 will occupy 8 consecutive points.
- This instruction designates 4 continuous external input points starting from S and 4 continuous external input points starting from D1 to construct a 16-key keyboard by a matrix scan. The keyed-in value will be stored in D2 and D3 is used for storing key status. If several keys are pressed at the same time, the first key pressed has the priority.
- The keyed-in value is termporarily stored in D0. When the 16-bit instruction HKY is in use, 9,999 is the maximum value D0 is able to store. When the value exceeds 4 digits, the highest digit will overflow. When the 32-bit instruction DHKY is in use, 99,999,999 is the maximum value D0 is able to store. When the value exceeds 8 digits, the highest digit will overflow.
- There is no limitation on the times of using this instruction. However, only one instruction can be executed at a time.

2、Program Example:

- Designate 4 input points X10 ~ X13 and the other 4 input points Y10 ~ Y13 to construct a 16-key keyboard. When X4 = On, the instruction will be executed and the keyed-in value will be stored in D0 in bin form. The key status will be stored in M0 ~ M7



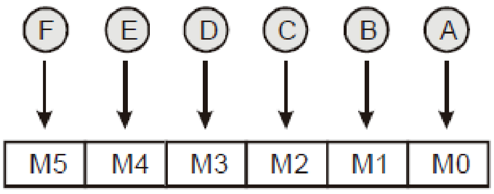
- Key in numbers:



- Function keys input:

  1) When A is pressed, M0 will be On and retained. When D is pressed next, M0 will be Off, M3 will be On and retained.

  2) When many keys are pressed at the same time, the first key pressed has the priority.

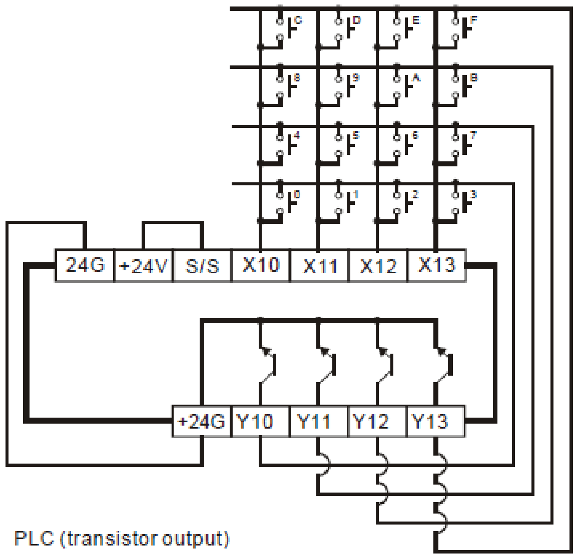


- Key output signal:

  1) When any of A ~ F is pressed, M6 will be On for once.

  2) When any of 0 ~ 9 is pressed, M7 will be On for once.

- When X4 goes Off, the keyed-in value prior to D0 will remain unchanged, but M0 ~ M7 will all be Off.
- External wiring:



Remarks:

- When this instruction is being executed, it will require 8 scans to obtain one valid keyed-in value. A scan period that is too long or too short may result in poor keyed-in effect, which can be avoided by the following methods:
  a) If the scan period is too short, I/O may not be able to respond in time, resulting in not being able to read the keyed-in value correctly. In this case, please fix the scan time.
  b) If the scan period is too long, the key may respond slowly. In this case, write this instruction into the time interruption subroutine to fix the time for the execution of this instruction.
- Functions of M1167:
  a) When M1167 = On, HKY instruction will be able to input the hexadecimal value of 0 ~ F.
  b) When M1167 = Off, HKY instruction will see A ~ F as function keys.
- Functions of D1037 :

  Write D1037 to set the overlapping time for keys (unit: ms). The overlapping time will vary upon different program scan time and the settings in D1037.

ZL 72		DSW		S D1 D2 n	Digital Switch
	D				

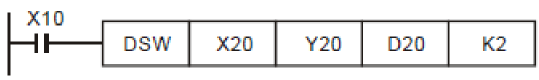
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															DSW: 9 steps 16-bit
D_1		*														
D_2											*	*	*	*	*	
n					*	*										

1、Explanations:

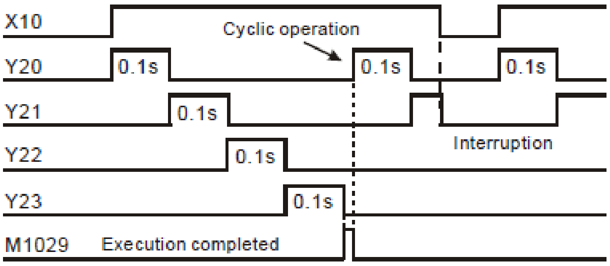
- S: Start device for switch scan input　D1: Start device for switch scan output
  D2: Device for storing the set value of switch　n: Groups of switches
- Range of n: 1 ~ 2.
- This instruction designates 4 or 8 consecutive external input points starting from S and 4 consecutive external input points starting from D1 to scan read 1 or 2 4-digit DIP switches.The set values of DIP switches are stored in D2. n decides to read 1 or 2 4-digit DIP switches.
- There is no limitation on the times of using this instruction in the program. However, two instructions are allowed to be executed at a time.

2、Program Example:

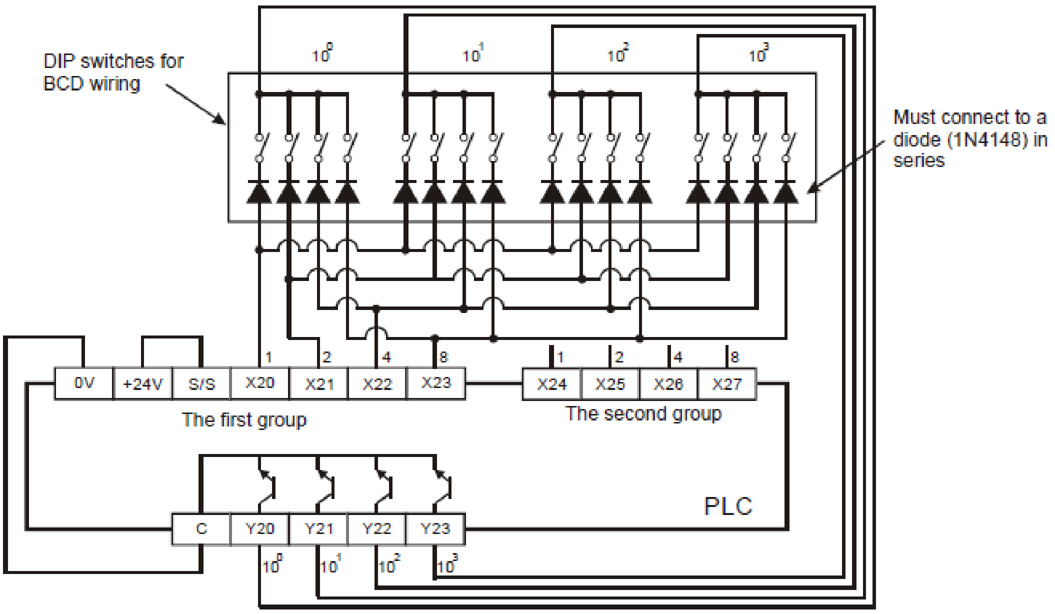
- The first group of DIP switches consist of X20 ~ X23 and Y20 ~ Y23. The second group of switches consist of X24 ~ X27 and Y20 ~ Y23. When X10 = On, the instruction will be executed and the set values of the first group switches will be read and converted into bin values before being stored in D20. The set values of the second group switches will be read, converted into bin values and stored in D21.



- When X10 = On, the Y20 ~ Y23 auto scan cycle will be On. Whenever a scan cycle is completed, M1029 will be On for a scan period.
- Please use transistor output for Y20 ~ Y23. Every pin 1, 2, 4, 8 shall be connected to a diode (0.1A/50V) before connecting to the input terminals on PLC.



- Wiring for DIP swich input:



Remarks:

- When n = K1, D2 will occupy one register. When n = K2, D2 will occupy 2 consecutive registers.
- Follow the methods below for the transistor scan output:

  a) When X10 = On, DSW instruction will be executed. When X10 goes Off, M10 will keep being On until the scan output completes a scan cycle and go Off.

  b) When X10 is used as a button switch, whenever X10 is pressed once, M10 will be reset to Off when the scan output designated by DSW instruction completes a scan cycle. The DIP switch data will be read completely and the scan output will only operate during the time when the button switch is pressed. Therefore, even the scan output is a transistor type, the life span of the transistor can be extended because it does not operate too frequently.

ZL 73		SEGD		S D	Seven Segment Decoder

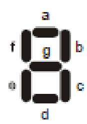
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SEGD: 5 steps 16-bit
S					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

1、Explanations:

- S: Source device to be decoded D: Output device after the decoding
- When X10 = On, the contents (0 ~ F in hex) of the lower 4 bits (b0 ~ b3) of D10 will be decoded into a 7-segment display for output. The decoded results will be stored in Y10 ~ Y17. If the content exceeds 4 bits, the lower 4 bits are still used for the decoding.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

```
 X10
─┤├────[ SEGD | D10 | K2Y10 ]
```

- Decoding table of the 7-segment display:

Hex	Bit combi-nation	Composition of the 7-segment display	Status of each segment							Data displayed
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)	
0	0000		ON	ON	ON	ON	ON	ON	OFF	0
1	0001		OFF	ON	ON	OFF	OFF	OFF	OFF	1
2	0010		ON	ON	OFF	ON	ON	OFF	ON	2
3	0011		ON	ON	ON	ON	OFF	OFF	ON	3
4	0100		OFF	ON	ON	OFF	OFF	ON	ON	4
5	0101		ON	OFF	ON	ON	OFF	ON	ON	5
6	0110		ON	OFF	ON	ON	ON	ON	ON	6
7	0111		ON	ON	ON	OFF	OFF	ON	OFF	7
8	1000		ON	ON	ON	ON	ON	ON	ON	8
9	1001		ON	ON	ON	ON	OFF	ON	ON	9
A	1010		ON	ON	ON	OFF	ON	ON	ON	A
B	1011		OFF	OFF	ON	ON	ON	ON	ON	b
C	1100		ON	OFF	OFF	ON	ON	ON	OFF	C
D	1101		OFF	ON	ON	ON	ON	OFF	ON	d
E	1110		ON	OFF	OFF	ON	ON	ON	ON	E
F	1111		ON	OFF	OFF	OFF	ON	ON	ON	F

ZL 74	SEGL	S D n														Seven Segment with Latch
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SEGL: 7 steps 16-bit
D		*														
n					*	*										

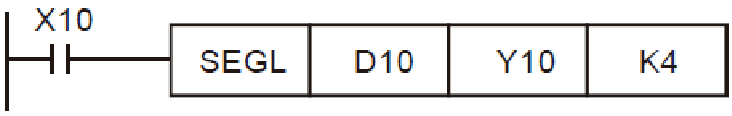
1、Explanations:

- S: Source device to be displayed in 7-segment display D: Start device for 7-segment display scan output

  n: Polarity setting of output signal and scan signal
- Range of n: 0 ~ 7. See remarks for more details.
- The instruction can be used twice in the program.
- This instruction occupies 8 or 12 continuous external input points starting from D for displaying 1 or 2 4-digit 7-segment display data and outputs of scanned signals. Every digit carries a 7-segment display drive (to convert the BCD codes into 7-segment display signal). The drive also carries latch control signals to retain the 7-segment display.
- n decides there be 1 group or 2 groups of 4-digit 7-segment display and designates the polarity for the output.
- When there is 1 group of 4-digit output, 8 output points will be occupied. When there are 2 groups of 4-digit output, 12 output points will be occupied.
- When this instruction is being executed, the scan output terminals will circulate the scan in sequence. When the drive contact of the instruction goes from Off to On again, the scan output terminal will restart the scan again.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- When X10 = On, this instruction starts to be executed, Y10 ~ Y17 construct a 7-segment display scan circuit. The value in D10 will be converted into BCD codes and sent to the first group 7-segment display. The value in D11 will be converted into BCD codes as well and sent to the second group 7-segment display. If the values in D10 and D11 exceed 9,999, operational error will occur



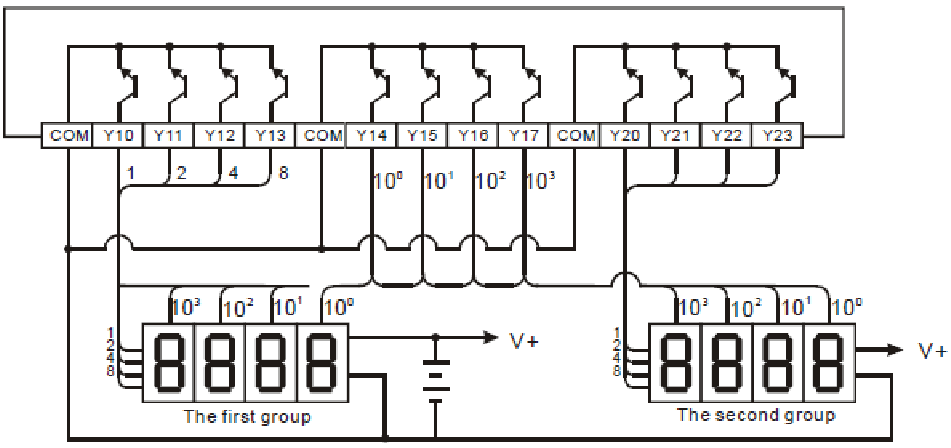
- When X10 = On, Y14 ~ Y17 will circulate the scan automatically. Every cycle requires 12 scan period. Whenever a cycle is completed, M1029 will be On for a scan period.
- When there is 1 group of 4-digit 7-segment display, n = 0 ~ 3.

  a) Connect the already decoded 7-segment display terminals 1, 2, 4, 8 in parallel an connect them to Y10 ~ Y13 on the PLC. Connect the latch terminals of each digit to Y14 ~ Y17 on the PLC.

  b) When X10 = On, the instruction will be executed and the content in D10 will be sent to the 7-segment displays in sequence by the circulation of Y14 ~ Y17.
- When there is 2 groups of 4-digit 7-segment display, n = 4 ~ 7.

  a) Connect the already decoded 7-segment display terminals 1, 2, 4, 8 in parallel an connect them to Y20 ~ Y23 on the PLC. Connect the latch terminals of each digit to Y14 ~ Y17 on the PLC.

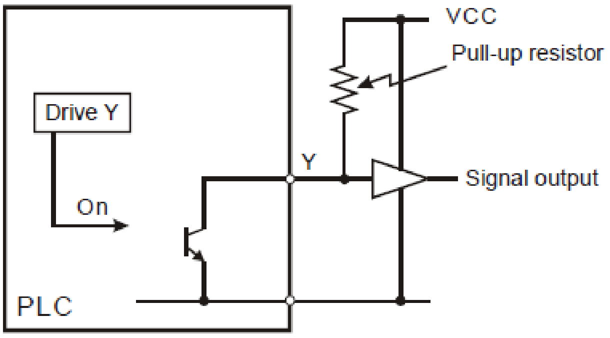
  b) The contents in D10 are sent to the first group 7-segment display. The contents in D11 are sent the the second group 7-segment display. If D10 = K1234 and D11 = K4321, the first group will display 1 2 3 4, and the second group will display 4 3 2 1.
- Wiring of the 7-segment display scan output.



Remarks:

- When this instruction is executed, the scan time has to be longer than 10ms. If the scan time is shorter than 10ms, please fix the scan time at 10ms.
- n is for setting up the polarity of the transistor output and the number of groups of the 4-digit 7-segment display.
- The output point must be a transistor module of NPN output type with open collector outputs. The output

has to connect to a pull-up resistor to VCC (less than 30VDC). Therefore, when output point Y is On, the signal output will be in low voltage.



- Positive logic (negative polarity) output of BCD code

BCD value				Y output (BCDcode)				Signal output			
b_3	b_2	b_1	b_0	8	4	2	1	A	B	C	D
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1	0

- Negative logic (positive polarity) output of BCD code.

BCD value				Y output (BCDcode)				Signal output			
b_3	b_2	b_1	b_0	8	4	2	1	A	B	C	D
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	0	1	0	0
0	1	0	1	1	0	1	0	0	1	0	1
0	1	1	0	1	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0	0	0
1	0	0	1	0	1	1	0	1	0	0	1

- Scan latched signal display

Positive logic (negative polarity)		Negative logic (positive polarity)	
Y output (latch)	Output signal	Y output (latch)	Output signal
1	0	0	1

- Settings of n:

Groups of 7-segment display	1 group				2 groups			
Y output of BCD code	+		−		+		−	
Scan latched signal display	+	−	+	−	+	−	+	−
n	0	1	2	3	4	5	6	7

+: Positive logic (negative polarity) output −: Negative logic (positive polarity) output

- The polarity of transistor output and the polarity of the 7-segment display input can be the same or different by the setting of n.

ZL 75	ARWS				S D_1 D_2 n											Arrow Switch
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												ARWS: 9 steps 16-bit
D_1											*	*	*	*	*	
D_2		*														
n					*	*										

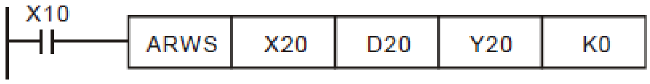
1、Explanations:

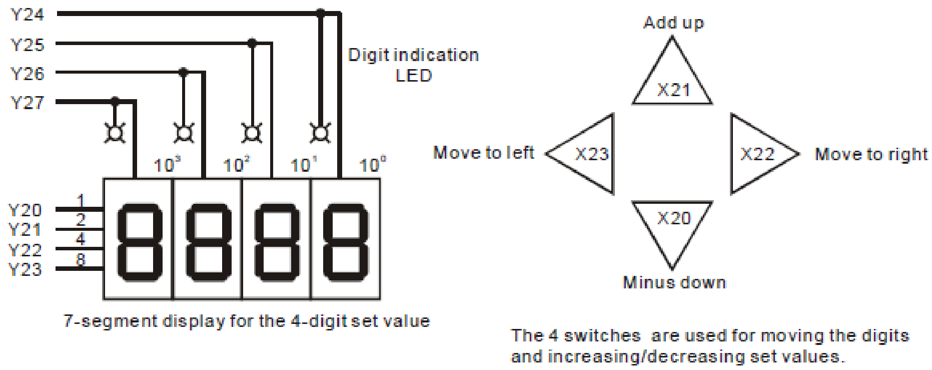
- S: Start device for key input D1: Device to be displayed in 7-segment display
  D2: Start device for 7-segmentdisplay scan output n: Polarity setting of output signal and scan signal
- S will occupy 4 consecutive points.
- Range of n: 0 ~ 3. See remarks of API 74 SEGL for more details.
- There no limitation on the times of using this instruction in the program. However, only one instruction is allowed to be executed at a time.
- The output points designated by this instruction shall be transistor output.
- When using this instruction, please fix the scan time, or place this instruction in the time interruption subroutine (I6□□ ~ I8□□).
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- When this instruction is executed, X20 is defined as down key, X21 is defined as up key, X22 is defined as right key and X23 is defined as left key. The keys are used for setting up and displaying external set values. The set values (range: 0 ~ 9,999) are stored in D20.
- When X10 = On, digit 103 will be the valid digit for setup. If you press the left key at this time, the valid digit will circulate as $10^3 \rightarrow 10^0 \rightarrow 10^1 \rightarrow 10^2 \rightarrow 10^3 \rightarrow 10^0$
- If you press the right key at this time, the valid digit will circulate as $10^3 \rightarrow 10^2 \rightarrow 10^1 \rightarrow 10^0 \rightarrow 10^3 \rightarrow 10^2$. During the circulation, the digit indicators connected Y24 ~ Y27 will also be On interchangeably following the circulation.
- If you press the up key at this time, the valid digit will change as 0 → 1 → 2 ⋯ → 8 → 9 → 0 →1. If you press the down key, the valid digit will change as 0 → 9 → 8 ⋯ → 1 → 0 → 9. The changed value will

also be displayed in the 7-segment display.





ZL 76 ASC S D — ASCII Code Conversion

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S															
D											*	*	*		

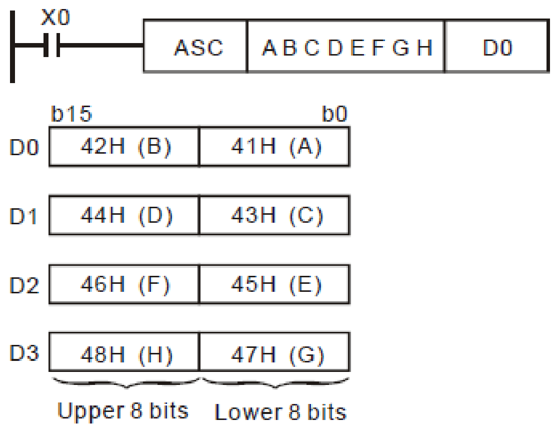
ASC: 11 steps 16-bit

1、Explanations:

- S: English letter to be converted into ASCII code D: Device for storing ASCII code
- If the execution of this instruction is connected to a 7-segment display, the error message can be displayed by English letters.

2、Program Example:

- When X0 = On, convert A ~ H into ASCII code and stored it in D0 ~ D3.



7.4 (ZL 80-89) External SER equipment

ZL 80	RS	S m D n	Serial Communication Instruction

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			RS: 9 steps 16-bit
m					*	*							*			
D													*			
n					*	*							*			

1、Explanations:

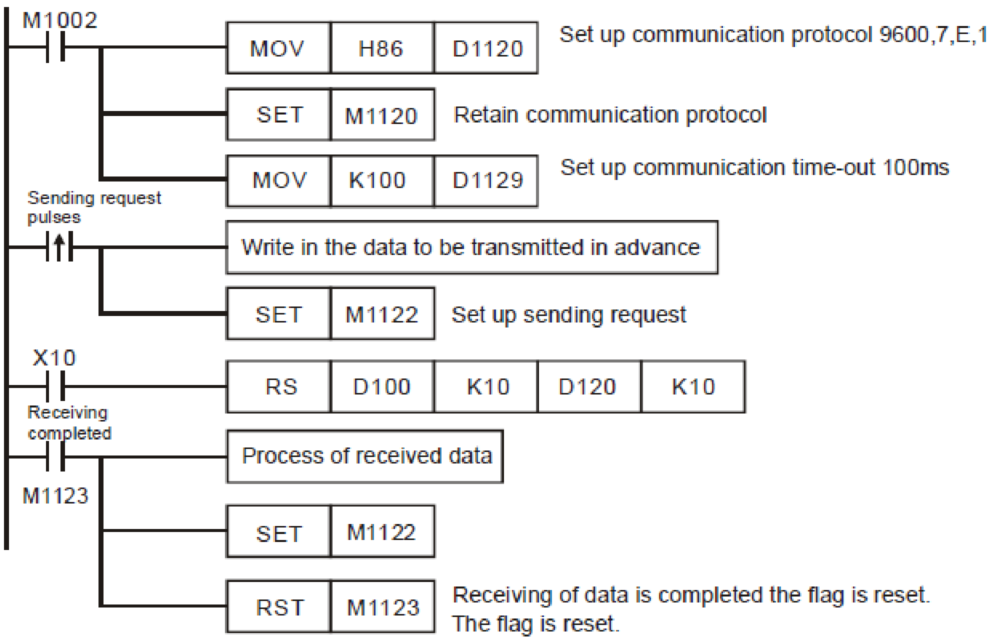
- S: Start device for the data to be transmitted m: Length of data to be transmitted

  D: Start device for receiving data n: Length of data to be received
- Range of m: 0 ~ 256
- Range of n: 0 ~ 256
- The instruction RS supports COM2 (RS-485)
- This instruction is a handy instruction exclusively for MPU to use RS-485 serial communication interface. The user has to pre-store word data in S data register, set up data length m and the data receiving register D and received data length n. If E, F index registers are used to modify S and D, the user cannot change the set values of E and F when the instruction is being executed; otherwise errors may cause in data writing or reading.
- Designate m as K0 if you do not need to send data. Designate n as K0 if you do not need to receive data.
- There is no limitation on the times of using this instruction in the program, but only one instruction is allowed to be executed at a time.
- During the execution of RS instruction, changing the data to be transmitted will be invalid.
- If the peripheral devices, e.g. AC motor drive, are equipped with RS-485 serial communication and its communication format is open, you can use RS instruction to design the program for the data transmission between PLC and the peripheral device.
- If the communication format of the peripheral device is Modbus, PLC offers handy communication instructions API 100 MODRD, API 101 MODWR, and API 150 MODRW, to work with the device. See explanations of the instructions in this application manual.

- For the special auxiliary relays M1120 ~ M1161 and special data registers D1120 ~ D1131 relevant to RS-485 communication, see remarks for more details.

2、Program Example 1:

- Use COM2 on the PLC to carry out RS-485 communication.
- Write the data to be transmitted in advance into registers starting from D100 and set M1122 (sending request flag) as On.
- When X10 = On, RS instruction will be executed and PLC will start to wait for the sending and receiving of data. D100 starts to continuousl send out 10 data and when the sending is over, M1122 will be automatically reset to Off (DO NOT use the program to execute RST M1122). After 1ms of waiting, PLC will start to receive the 10 data. Store the data in consecutive registers starting from D120.
- When the receiving of data is completed, M1123 will automatically be On. After the program finishes processing the received data, M1123 has to be reset to Off and the PLC will start to wait for the sending and receiving of data again. DO NOT use the program to continuously execute RST M1123.



3、Program Example 2:

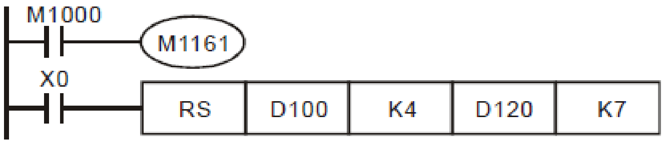
- Use COM2 on the PLC to carry out RS-485 communication

  Switching between 8-bit mode (M1161 = On) and 16-bit mode (M1161 = Off)

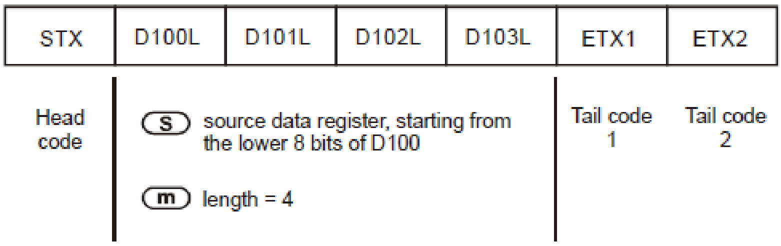
  8-bit mode:

  1) The head code and tail code of the data are set up by M1126 and M1130 together with D1124 ~ D1126. When PLC is executing RS instruction, the head code and tail code set up by the user will be

sent out automatically. M1161 = On indicates PLC in 8-bit conversion mode. The 16-bit data will be divided into the higher 8 bits and lower 8 bits. The higher 8 bits are ignored and only the lower 8 bits are valid for data transmission.

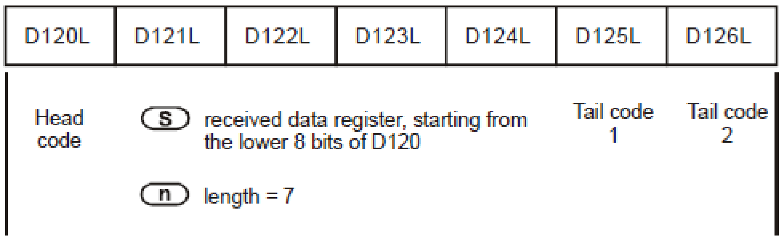


Sending data: (PLC -> external equipment)



Receiving data: (External equipment -> PLC)

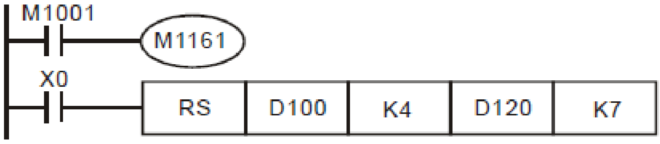
Receiving data: (External equipment -> PLC)



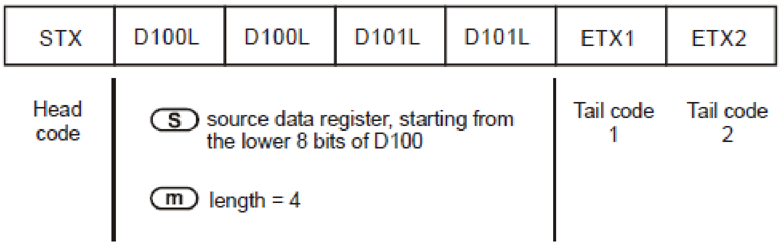
When receiving data, PLC will receive the head code and tail code of the data from the external equipment; therefore, the user has to be aware of the setting of data length n.

2) 16-bit mode:

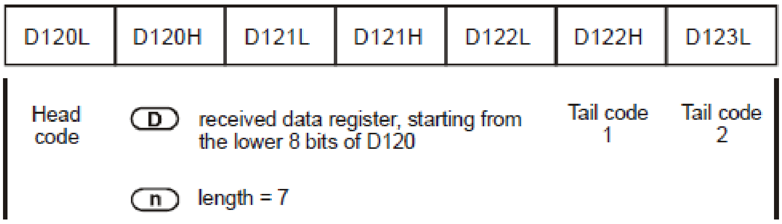
The head code and tail code of the data are set up by M1126 and M1130 together with D1124 ~ D1126. When PLC is executing RS instruction, the head code and tail code set up by the user will be sent out automatically. M1161 = Off indicates PLC in 16-bit conversion mode. The 16-bit data will be divided into the higher 8 bits and lower 8 bits for data transmission.



Sending data: (PLC -> external equipment)



Receiving data: (External equipment -> PLC)

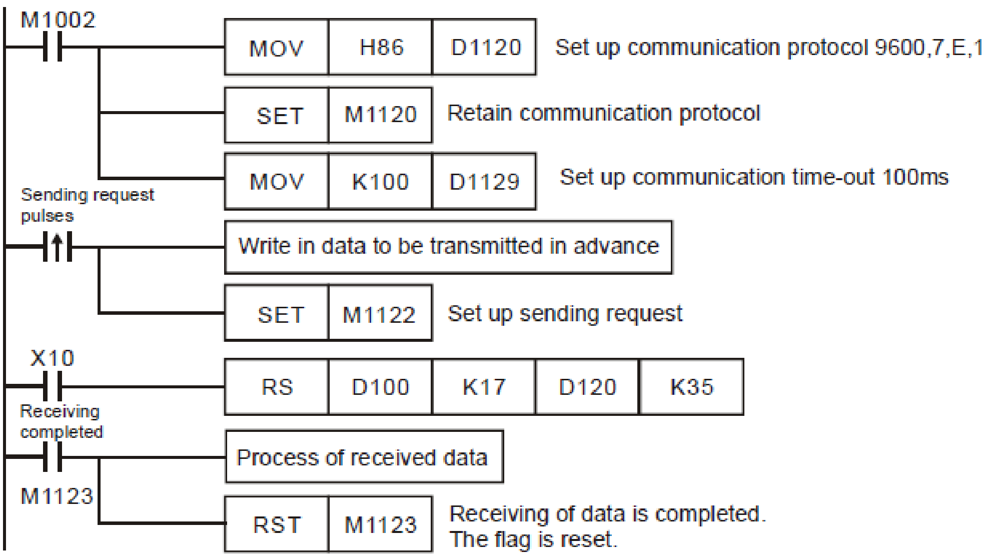


When receiving data, PLC will receive the head code and tail code of the data from the external equipment; therefore, the user has to be aware of the setting of data length n

4、Program Example 3:

Use COM2 on the PLC to carry out RS-485 communication.

Connect PLC to AC motor drives (AC motor drive in ASCII Mode; PLC in 16-bit mode and M1161 = Off). Write in the 6 data starting from parameter address H2101 in AC motor drive in advance as the data to be transmitted



PLC → AC motor drive, PLC sends “: 01 03 2101 0006 D4 CR LF “

AC motor drive → PLC, PLC receives “: 01 03 0C 0100 1766 0000 0000 0136 0000 3B CR

LF “ Registers for sent data (PLC sends out message)

Register	Data		Explanation	
D100 low	‘: ’	3A H	STX	
D100 high	‘0’	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D101 low	‘1’	31 H	ADR 0	
D101 high	‘0’	30 H	CMD 1	Instruction code: CMD (1,0)
D102 low	‘3’	33 H	CMD 0	
D102 high	‘2’	32 H	Start data address	
D103 low	‘1’	31 H		
D103 high	‘0’	30 H		
D104 low	‘1’	31 H		
D104 high	‘0’	30 H	Number of data (counted by words)	
D105 low	‘0’	30 H		
D105 high	‘0’	30 H		
D106 low	‘6’	36 H		
D106 high	‘D’	44 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D107 low	‘4’	34 H	LRC CHK 0	
D107 high	CR	D H	END	
D108 low	LF	A H		

Registers for received data (AC motor drive responds with messages)

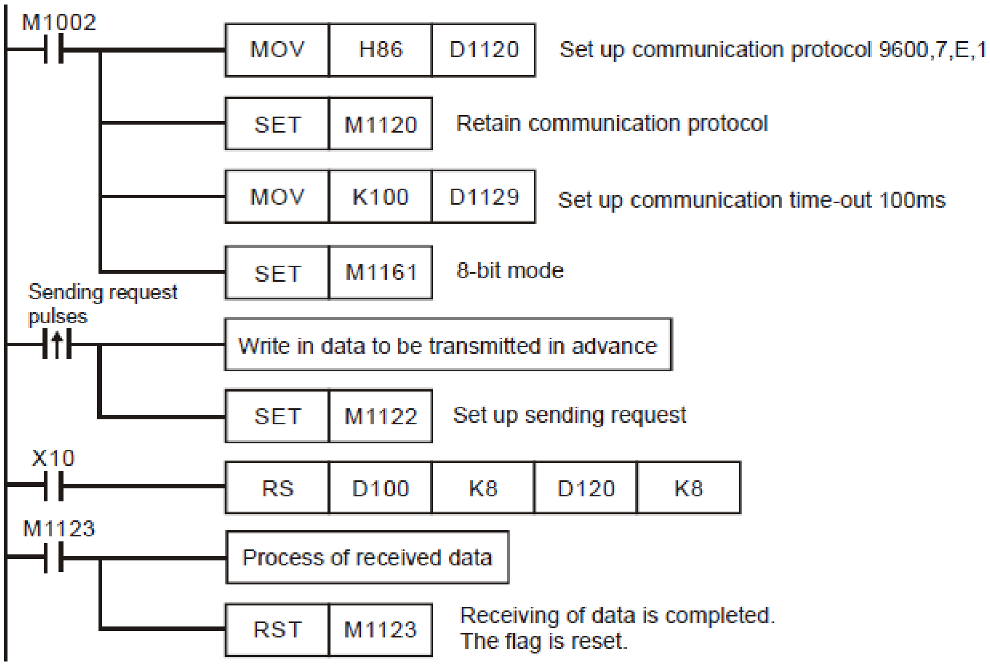
Register	Data		Explanation
D120 low	‘: ’	3A H	STX
D120 high	‘0’	30 H	ADR 1
D121 low	‘1’	31 H	ADR 0
D121 high	‘0’	30 H	CMD 1
D122 low	‘3’	33 H	CMD 0
D122 high	‘0’	30 H	Number of data (counted by byte)
D123 low	‘C’	43 H	
D123 high	‘0’	30 H	Content of address 2101 H
D124 low	‘1’	31 H	
D124 high	‘0’	30 H	
D125 low	‘0’	30 H	
D125 high	‘1’	31 H	Content of address 2102 H
D126 low	‘7’	37 H	
D126 high	‘6’	36 H	
D127 low	‘6’	36 H	
D127 high	‘0’	30 H	Content of address 2103 H
D128 low	‘0’	30 H	
D128 high	‘0’	30 H	
D129 low	‘0’	30 H	
D129 high	‘0’	30 H	Content of address 2104 H
D130 low	‘0’	30 H	
D130 high	‘0’	30 H	
D131 low	‘0’	30 H	
D131 high	‘0’	30 H	Content of address 2105 H
D132 low	‘1’	31 H	
D132 high	‘3’	33 H	
D133 low	‘6’	36 H	
D133 high	‘0’	30 H	Content of address 2106 H
D134 low	‘0’	30 H	
D134 high	‘0’	30 H	
D135 low	‘0’	30 H	
D135 high	‘3’	33 H	LRC CHK 1
D136 low	‘B’	42 H	LRC CHK 0
D136 high	CR	D H	END
D137 low	LF	A H	

5、Program Example 4:

Use COM2 on the PLC to carry out RS-485 communication.

Connect PLC to AC motor drives (AC motor drive in RTU Mode; PLC in 16-bit mode and M1161 = On).

Write in H12 to parameter address H2000 in VFD-B in advance as the data to be transmitted.



PLC → AC motor drive, PLC sends: 01 06 2000 0012 02 07

AC motor drive → PLC, PLC receives: 01 06 2000 0012 02 07

Registers for sent data (PLC sends out messages)

Register	Data	Explanation
D100 low	01 H	Address
D101 low	06 H	Function
D102 low	20 H	Data address
D103 low	00 H	
D104 low	00 H	Data content
D105 low	12 H	
D106 low	02 H	CRC CHK Low
D107 low	07 H	CRC CHK High

Registers for received data (VFD-B responds with messages)

Register	Data	Explanation
D120 low	01 H	Address
D121 low	06 H	Function
D122 low	20 H	Data address
D123 low	00 H	
D124 low	00 H	Data content
D125 low	12 H	
D126 low	02 H	CRC CHK Low
D127 low	07 H	CRC CHK High

1、 PLC COM2 RS-485: Associated flags (Auxiliary relays) and special registers (Special D) for communication instructions RS / MODRD / MODWR / FWD / REV / STOP / MODRW.

2、 How to set up RS-485 communication protocol in D1120

	Content	0	1
b0	Data length	7	8
b1 b2	Parity bits	00: None 01: Odd 11: Even	
b3	Stop bits	1 bit	2 bits
b4 b5 b6 b7	0001 (H1) : 110 0010 (H2) : 150 0011 (H3) : 300 0100 (H4) : 600 0101 (H5) : 1200 0110 (H6) : 2400 0111 (H7) : 4800 1000 (H8) : 9600 1001 (H9) : 19200 1010 (HA) : 38400 1011 (HB) : 57600 1100 (HC) : 115200		
b8	Start word	None	D1124
b9	First end word	None	D1125
b10	Second end word	None	D1126
b15 ~ b11	Not defined		

3、 When RS instruction is in use, the frequently used communication format in the peripheral device will define the start word and end word of the control string. Therefore, you can set up the start word and end word in D1124 ~ D1126 for COM2 or use the start word and end word defined by the PLC. When you use M1126, M1130 and D1124 ~ D1126 to set up the start word and end word, b8 ~ b10 of D1120 have to be set as 1 to make valid the RS-485 communication protocol. See the table below for how to set up.

		M1130	
		0	1
M1126	0	D1124: user defined D1125: user defined D1126: user defined	D1124: H 0002 D1125: H 0003 D1126: H 0000 (no setting)
	1	D1124: user defined D1125: user defined D1126: user defined	D1124: H 003A (':') D1125: H 000D (CR) D1126: H 000A (LF)

4、 Example of how to set up the communication format of COM2:

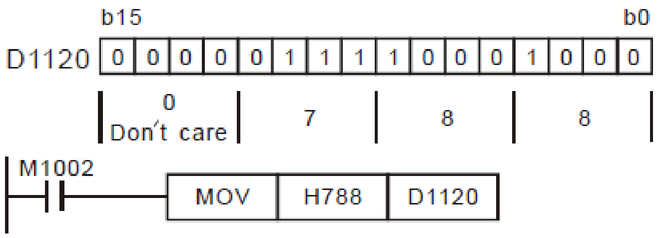
Assume there is a communication format: Baud rate 9600 7, N, 2

STX : ":"

ETX1 : "CR"

ETX2 : "LF"

Check the table and obtain the communication format H788 and write it into D1120.



When STX, ETX1 and EXT2 are in use, please be aware of the On and Off of the special auxiliary relays M1126 and M1130.

5、 M1143 is for the selection of ASCII mode or RTU mode. On = RTU mode; Off = ASCII mode.

Take the standard Modbus format for example:

In ASCII mode (M1143 = Off)

STX	Start word = ':' (3AH)
Address Hi	Communication address:
Address Lo	The 8-bit address consists of 2 ASCII codes
Function Hi	Function code:
Function Lo	The 8-bit function code consists of 2 ASCII codes
DATA (n-1)	Data:
…….	
DATA 0	The n × 8-bit data consists of 2n ASCII codes
LRC CHK Hi	LRC checksum:
LRC CHK Lo	The 8-bit checksum consists of 2 ASCII code
END Hi	End word:
END Lo	END Hi = CR (0DH), END Lo = LF(0AH)

The communication protocol is in Modbus ASCII mode, i.e. every byte is composed of 2 ASCII characters. For example, 64Hex is '64' in ASCII, composed by '6' (36Hex) and '4' (34Hex). Every hex '0'…'9', 'A'…'F' corresponds to an ASCII code.

Character	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'
ASCII code	30H	31H	32H	33H	34H	35H	36H	37H

Character	'8'	'9'	'A'	'B'	'C'	'D'	'E'	'F'
ASCII code	38H	39H	41H	42H	43H	44H	45H	46H

Start word (STX):

Fixed as ':' (3AH)

Address:

'0' '0': Broadcasting to all drivers

'0' '1': To the driver at address 01

'0' 'F': To the driver at address 15

'1' '0': To the driver at address 16

….and so on, maximum to the driver at address 254 ('F' 'E')

Function code:

'0' '1': Reading several bit devices

'0' '2': Reading several bit devices (read-only devices)

'0' '3': Reading several word devices

'0' '4': Reading several word devices (read-only devices)

'0' '5': Writing a state in a single bit device

'0' '6': Writing data in a single word device

'0' 'F': Writing states in bit devices

'1' '0': Writing data in word devices

'1' '7': Reading word devices and writing data in word devices

Data characters: The data sent by the user.

LRC checksum:

LCR checksum is 2's complement of the value added from Address to Data Content.

For example: 01H + 03H + 21H + 02H + 00H + 02H = 29H. 2's complement of 29H = D7H

End word (END):

Fixed as END Hi = CR (0DH), END Lo = LF (0AH)

End word (END):

Fixed as END Hi = CR (0DH), END Lo = LF (0AH)

For example: Read 2 continuous data stored in the registers of the driver at address 01H (see the table below). The start register is at address 2102H.

Inquiry message:

STX	':'
Slave station address	'0'
	'1'
Function code	'0'
	'3'
Start address	'2'
	'1'
	'0'
	'2'
Number of data (counted by words)	'0'
	'0'
	'0'
	'2'
LRC checksum	'D'
	'7'
END	CR
	LF

Responding message:

STX	':'
Slave station address	'0'
	'1'
Function code	'0'
	'3'
Number of data (counted by byte)	'0'
	'4'
Content in start address 2102H	'1'
	'7'
	'7'
	'0'
Content of address 2103H	'0'
	'0'
	'0'
	'0'
LRC check	'7'
	'1'
END	CR
	LF

In RTU mode (M1143 = On)

Name	Data (hexadecimal system)
START	See the following explanation
Address	Communication address: In 8-bit binary
Function	Function code: In 8-bit binary
DATA (n-1)	Data: n × 8-bit data
…….	
DATA 0	
CRC CHK Low	CRC checksum: 16-bit CRC consists of 2 8-bit binary
CRC CHK High	
END	See the following explanation

Address:

00H: Broadcasting to all drivers

01H: To the driver at address 01

0FH: To the driver at address 15

10H: To the driver at address 16…. And so on, maximum to the driver at address 254 (FE H)

Function code:

02H: Reading several bit devices

03H: Reading several word devices

04H: Reading several word devices (read-only devices)

05H: Writing a state in a single bit device

06H: Writing data in a single word device

0FH: Writing states in bit devices

10H: Writing data in word devices

17H: Reading word devices and writing data in word devices

Data characters: The data sent by the user.

CRC checksum: Starting from Address and ending at Data Content.

Step 1: Make the 16-bit register (CRC register) = FFFFH

Step 2: Exclusive OR the first 8-bit message and the low 16-bit CRC register. Store the result in the CRC register.

Step 3: Right shift CRC register for a bit and fill "0" into the high bit.

Step 4: Check the value shifted to the right. If it is 0, fill in the new value obtained in step 3 and store the value in CRC register; otherwise, Exclusive OR A001H and CRC register and store the result in the CRC register.

Step 5: Repeat step 3 – 4 and finish operations of all the 8 bits.

Step 6: Repeat step 2 – 5 for obtaining the next 8-bit message until the operation of all the messages are completed. The final value obtained in the CRC register is the CRC checksum. The CRC checksum has to be placed interchangeably in the checksum of the message.

START and END:

See the table below :

Baud rate(bps)	RTU timeout timer (ms)	Baud rate (bps)	RTU timeout timer (ms)
300	40	9,600	2
600	21	19,200	1
1,200	10	38,400	1
2,400	5	57,600	1
4,800	3	115,200	1

For example: Read 2 continuous data stored in the registers of the driver at address 01H (see the table below).

The start register is at address 2102H.

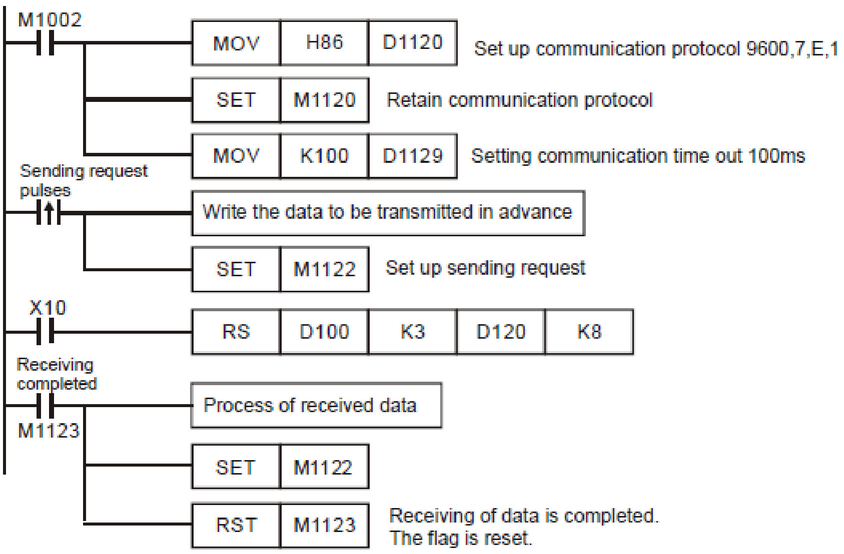
Inquiry message:

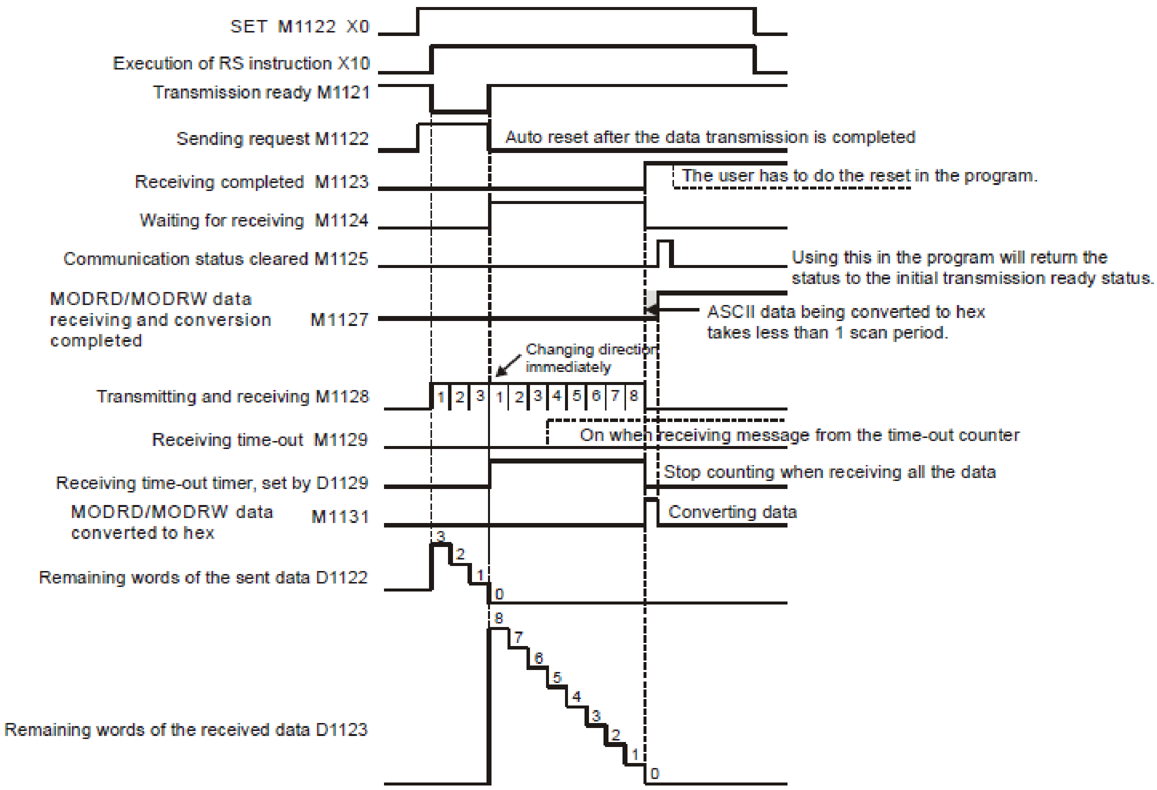
Name	Data (Hexadecimal value)
Address	01 H
Function code	03 H
Start data address	21 H
	02 H
Number of data (counted by words)	00 H
	02 H
CRC CHK Low	6F H
CRC CHK High	F7 H

Responding message:

Name	Data (Hexadecimal value)
Address	01 H
Function	03 H
Number of data (counted by byte)	04 H
Content in data address 2102H	17 H
	70 H
Content in data address 2103H	00 H
	00 H
CRC CHK Low	FE H
CRC CHK High	5C H

6、 Timing diagram of the RS-485 communication flag for COM2:




SET M1122 X0
Execution of RS instruction X10
Transmission ready M1121
Sending request M1122
Auto reset after the data transmission is completed
Receiving completed M1123
The user has to do the reset in the program.
Waiting for receiving M1124
Communication status cleared M1125
Using this in the program will return the status to the initial transmission ready status.
MODRD/MODRW data receiving and conversion completed M1127
ASCII data being converted to hex takes less than 1 scan period.
Changing direction immediately
Transmitting and receiving M1128
Receiving time-out M1129
On when receiving message from the time-out counter
Receiving time-out timer, set by D1129
Stop counting when receiving all the data
MODRD/MODRW data converted to hex M1131
Converting data
Remaining words of the sent data D1122
Remaining words of the received data D1123

ZL 81	D	PRUN	S D	Parallel Run

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S							*		*						
D								*	*						

PRUN: 5 steps 16-bit

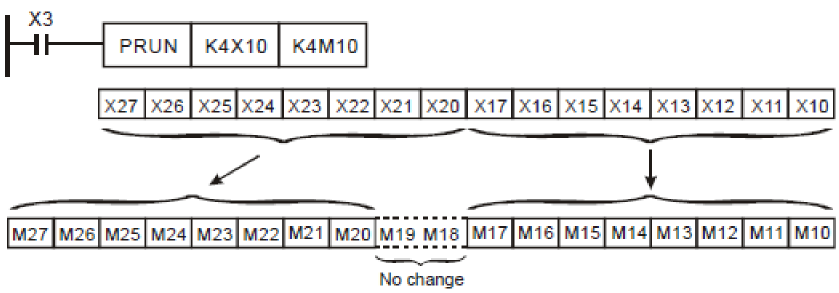
DPRUN: 9 steps 32-bit

1、Explanations:

- S: Source device D: Destination device
- The most right digit of X, Y and M of KnX, KnY and KnM has to be 0.
- When S designates KnX, D has to designate KnM; when S designates KnM, D has to designate KnY.
- This instruction sends the content in S to D in the form of octal system.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

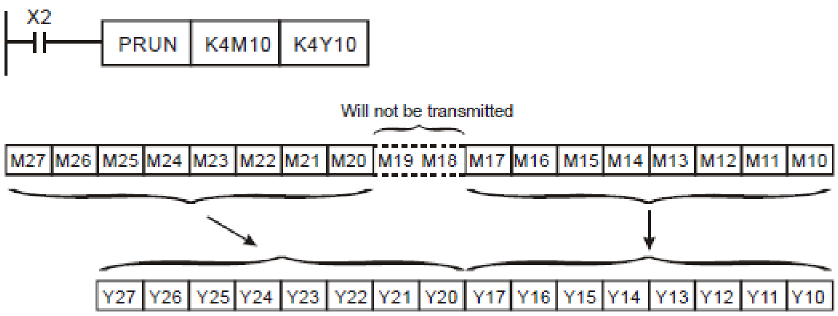
2、Program Example1:

When X3 = On, the content in K4X10 will be sent to K4M10 in octal form.



2、 Program Example 2:

When X2 = On, the content in K4M10 will be sent to K4Y10 in octal form.



ZL 82	ASCI	S D n	Converts Hex to ASCII

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			ASCI, ASCIP: 7 steps 16-bit
D								*	*	*	*	*	*			
n					*	*										

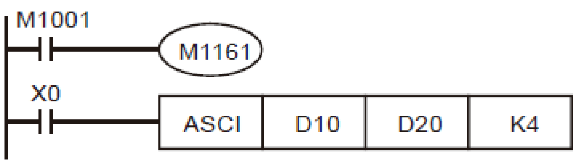
1、Explanations:

- S: Start device for source data D: Start device for storing the converted result

  n: Number of bits to be converted
- Range of n: 1 ~ 256
- 16-bit conversion mode: When M1161 = Off, the instruction converts every bit of the hex data in S into ASCII codes and send them to the 8 high bits and 8 low bits of D. n = the converted number of bits.
- 8-bit conversion mode: When M1161 = On, the instruction converts every bit of the hex data in S into ASCII codes and send them to the 8 low bits of D. n = the number of converted bits. (All 8 high bits of D = 0).
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example 1:

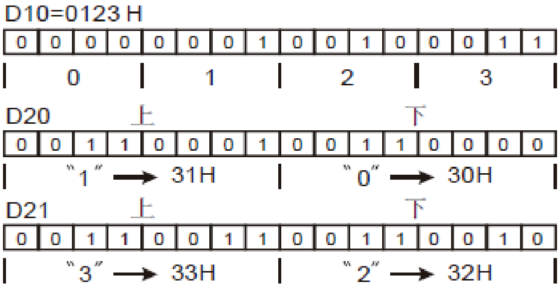
- M1161 = Off: The 16-bit conversion mode.
- When X0 = On, convert the 4 hex values in D10 into ASCII codes and send the result to registers starting from D20.



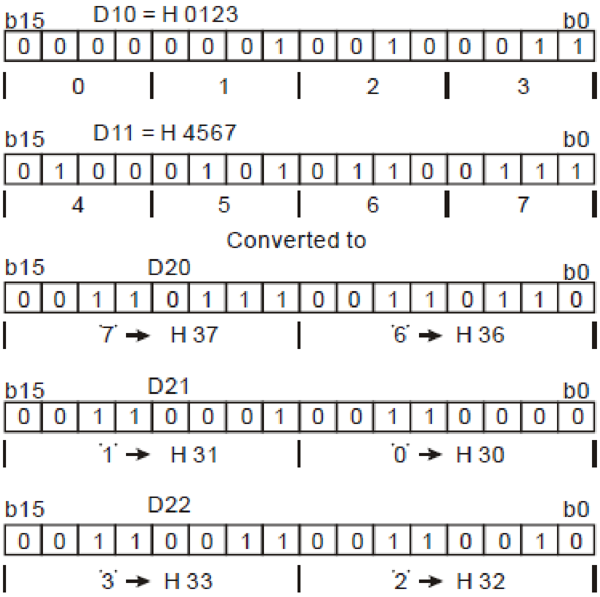
- Assume:

(D10)	=	0123 H	'0'	=	30H	'4'	=	34H	'8'	=	38H
(D11)	=	4567 H	'1'	=	31H	'5'	=	35H	'9'	=	39H
(D12)	=	89AB H	'2'	=	32H	'6'	=	36H	'A'	=	41H
(D13)	=	CDEF H	'3'	=	33H	'7'	=	37H	'B'	=	42H

- When n = 4, the bit structure will be as:



- When n = 6, the bit structure will be as:



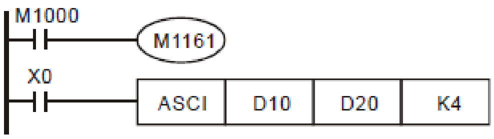
- When n = 1 ~ 16:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20 Low byte	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D20 High byte		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D21 Low byte			"3"	"2"	"1"	"0"	"7"	"6"
D21 High byte				"3"	"2"	"1"	"0"	"7"
D22 Low byte					"3"	"2"	"1"	"0"
D22 High byte						"3"	"2"	"1"
D23 Llow byte							"3"	"2"
D23 High byte								"3"
D24 Low byte								
D24 High byte				no change				
D25 Low byte								
D25 High byte								
D26 Low byte								
D26 High byte								
D27 Low byte								
D27 High byte								

D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20 Low byte	"B"	"A"	"9"	"8"	"F"	"E"	"D"	"C"
D20 High byte	"4"	"B"	"A"	"9"	"8"	"F"	"E"	"D"
D21 Low byte	"5"	"4"	"B"	"A"	"9"	"8"	"F"	"E"
D21 High byte	"6"	"5"	"4"	"B"	"A"	"9"	"8"	"F"
D22 Low byte	"7"	"6"	"5"	"4"	"B"	"A"	"9"	"8"
D22 High byte	"0"	"7"	"6"	"5"	"4"	"B"	"A"	"9"
D23 Llow byte	"1"	"0"	"7"	"6"	"5"	"4"	"B"	"A"
D23 High byte	"2"	"1"	"0"	"7"	"6"	"5"	"4"	"B"
D24 Low byte	"3"	"2"	"1"	"0"	"7"	"6"	"5"	"4"
D24 High byte		"3"	"2"	"1"	"0"	"7"	"6"	"5"
D25 Low byte			"3"	"2"	"1"	"0"	"7"	"6"
D25 High byte				"3"	"2"	"1"	"0"	"7"
D26 Low byte			no change		"3"	"2"	"1"	"0"
D26 High byte						"3"	"2"	"1"
D27 Low byte							"3"	"2"
D27 High byte								"3"

3、Program Example 2:

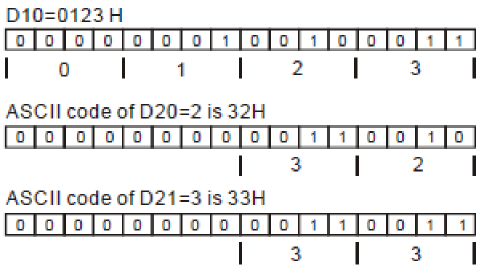
- M1161 = On: The 8-bit conversion mode.
- When X0 = On, convert the 4 hex values in D10 into ASCII codes and send the result to registers starting from D20.



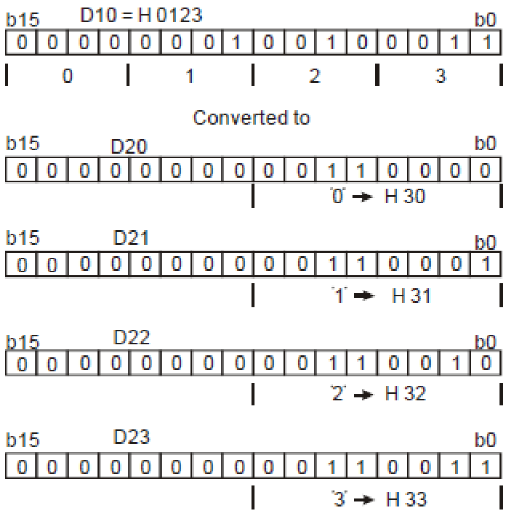
- Assume:

(D10)	=	0123 H	'0'	=	30H	'4'	=	34H	'8'	=	38H
(D11)	=	4567 H	'1'	=	31H	'5'	=	35H	'9'	=	39H
(D12)	=	89AB H	'2'	=	32H	'6'	=	36H	'A'	=	41H
(D13)	=	CDEF H	'3'	=	33H	'7'	=	37H	'B'	=	42H

- When n = 2, the bit structure will be as:



- When n = 4, the bit structure will be as:



- When n = 1 ~ 16:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D21		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D22			“3”	“2”	“1”	“0”	“7”	“6”
D23				“3”	“2”	“1”	“0”	“7”
D24					“3”	“2”	“1”	“0”
D25						“3”	“2”	“1”
D26							“3”	“2”
D27								“3”
D28								
D29				no change				
D30								
D31								
D32								
D33								
D34								
D35								

D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20	“B”	“A”	“9”	“8”	“F”	“E”	“D”	“C”
D21	“4”	“B”	“A”	“9”	“8”	“F”	“E”	“D”
D22	“5”	“4”	“B”	“A”	“9”	“8”	“F”	“E”
D23	“6”	“5”	“4”	“B”	“A”	“9”	“8”	“F”
D24	“7”	“6”	“5”	“4”	“B”	“A”	“9”	“8”
D25	“0”	“7”	“6”	“5”	“4”	“B”	“A”	“9”
D26	“1”	“0”	“7”	“6”	“5”	“4”	“B”	“A”
D27	“2”	“1”	“0”	“7”	“6”	“5”	“4”	“B”
D28	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D29		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D30			“3”	“2”	“1”	“0”	“7”	“6”
D31				“3”	“2”	“1”	“0”	“7”
D32					“3”	“2”	“1”	“0”
D33			no change			“3”	“2”	“1”
D34							“3”	“2”
D35								“3”

ZL 83	HEX	S D n	Converts ASCII to Hex

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			HEX, HEXP: 7 steps 16-bit
D								*	*	*	*	*	*			
n					*	*										

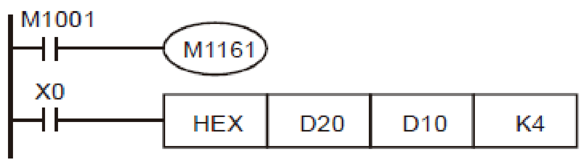
1、Explanations:

- S: Start device for source data D: Start device for storing the converted result

  n: Number of bits to be converted
- Range of n: 1 ~ 256
- 16-bit conversion mode: When M1161 = Off, the instruction is in 16-bit conversion mode. ASCII codes of the 8 high bits and 8 low bits of the hex data in S are converted into hex value and sent to D (every 4 bits as a group). n = the number of bits converted into ASCII codes.
- 8-bit conversion mode: When M1161 = On, the instruction is in 8-bit conversion mode. Every bit of the hex data in S are converted into ASCII codes and sent to the 8 low bits of D. n = the number of converted bits. (All 8 high bits of D = 0).
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example 1:

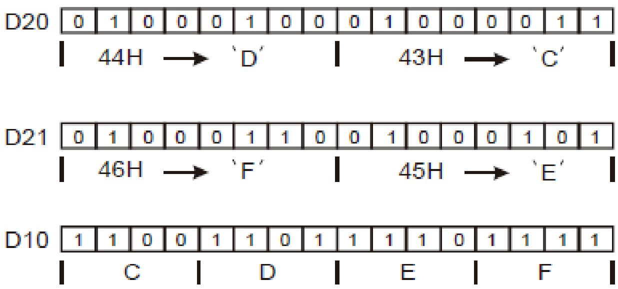
- M1161 = Off: The 16-bit conversion mode.
- When X0 = On, convert the ASCII codes stored in the registers starting from D20 into hex value and send the result (every 4 bits as a group) to registers starting from D10. n = 4.



- Assume:

S	ASCII code	Converted to hex	S	ASCII code	Converted to hex
D20 low byte	H 43	“C”	D24 low byte	H 34	“4”
D20 high byte	H 44	“D”	D24 high byte	H 35	“5”
D21 low byte	H 45	“E”	D25 low byte	H 36	“6”
D21 high byte	H 46	“F”	D25 high byte	H 37	“7”
D22 low byte	H 38	“8”	D26 low byte	H 30	“0”

- When n = 4, the bit structure will be as:

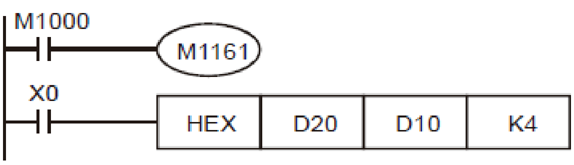


- When n = 1 ~ 16:

n \ D	D13	D12	D11	D10
1				***C H
2				**CD H
3				*CDE H
4				CDEF H
5	The undesignated parts in the registers in use are all 0.		***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

3、Program Example 2:

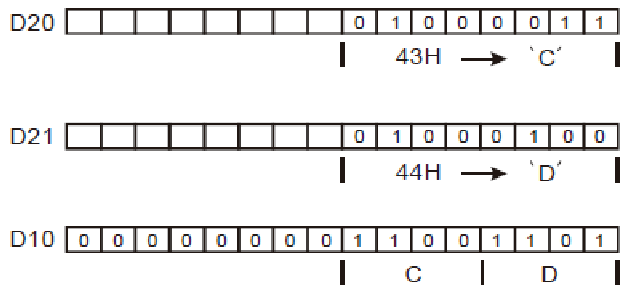
- M1161 = On: The 8-bit converstion mode.



- Assume:

S	ASCII code	Converted to hex	S	ASCII code	Converted to hex
D20	H 43	"C"	D28	H 34	"4"
D21	H 44	"D"	D29	H 35	"5"
D22	H 45	"E"	D30	H 36	"6"
D23	H 46	"F"	D31	H 37	"7"
D24	H 38	"8"	D32	H 30	"0"
D25	H 39	"9"	D33	H 31	"1"
D26	H 41	"A"	D34	H 32	"2"
D27	H 42	"B"	D35	H 33	"3"

- When n = 2, the bit structure will be as:



- When n = 1 ~ 16:

n \ D	D13	D12	D11	D10
1	The used registers which are not specified are all 0			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

ZL 87	ABS	D	Absolute Value
	D		

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
D								*	*	*	*	*	*	*	*

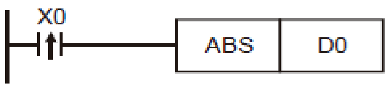
ABS: 3 steps	16-bit
DABS: 5 steps	32-bit

1、Explanations:

- D: Device of the absolute value.
- This instruction obtains the absolute value of the content in the designated in D.
- This instruction adopts pulse execution instructions (ABSP, DABSP).
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- When X0 = Off→On, obtain the absolute value of the content in D0.



<table>
<tr><td>ZL 88</td><td></td><td rowspan="2">PID</td><td></td><td rowspan="2" colspan="11">S_1 S_2 S_3 D</td><td rowspan="2" colspan="2">PID Control Loop</td></tr>
<tr><td></td><td>D</td><td></td></tr>
<tr><td></td><td colspan="4">Bit Devices</td><td colspan="11">Word Devices</td><td rowspan="6">
PID : 9 steps 16-bit
DPID: 17 steps 32-bit</td></tr>
<tr><td></td><td>X</td><td>Y</td><td>M</td><td>S</td><td>K</td><td>H</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>T</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td>S_1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>S_2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>S_3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
</table>

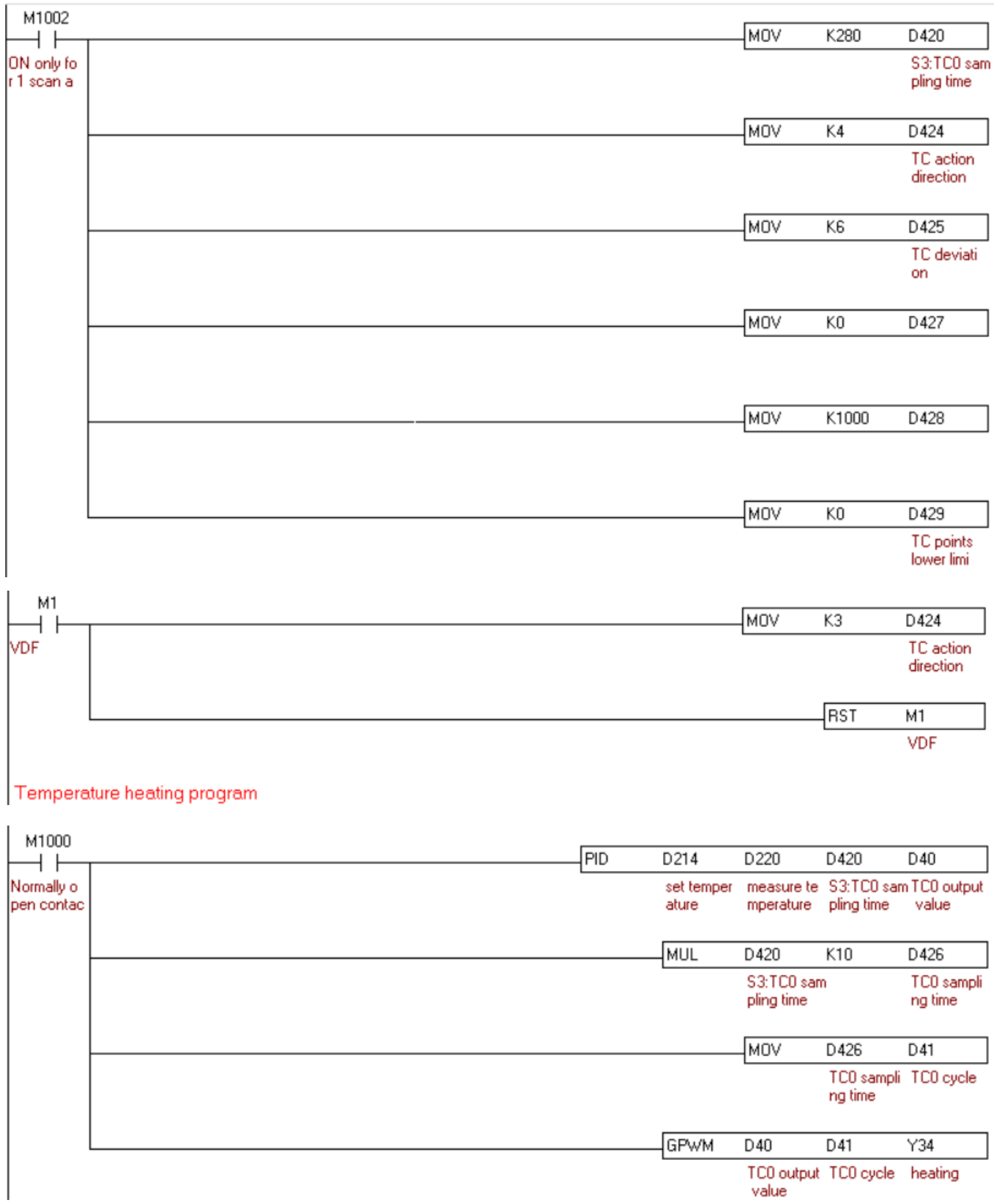
1、Explanations:

- S1: Set value (SV) S2: Present value (PV) S3: Parameter D: Output value (MV)
- In the 16-bit instruction, S3 will occupy 20 consecutive devices; in the 32-bit instruction, S3 will occupy 21 consecutive devices.
- This instruction is specifically for PID control. PID operation will be executed by the scan only when the sampling time is reached. PID refers to "proportion, integration and differential". PID control is widely applied to many machines, pneumatic and electronic equipments.
- For the 16-bit instruction, the parameters are S3 ~ S3+19; for the 32-bit instruction, the parameters are S3 ~ S3+20. After all the parameters are set up, PID instruction will start to be executed and the results will be stored in D. D has to be the data register area without latched function. (If you wish to designate a latched data register area, place the data register in the latched area at the beginning of the program and clear it as 0.)
- Application examples

  Use PID commands in temperature control system.

  Control purpose: make the control system reach the temperature target value.

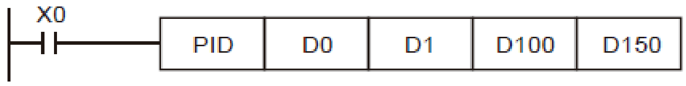
  The program example is as follows:



Note: The sampling time is set to 2.8S, the default D424 is K4 when power on, when M1 is ON, D424 is K3, and the system enters the temperature auto-tuning mode. After auto-tuning, the value of D424 changes from K3→K4.

2、Other Example:

- Complete the parameter setting before executing PID instruction.
- When X0 = On, the instruction will be executed and the result will be stored in D150. When X0 goes Off, the instruction will not be executed and the data prior to the instruction will stay intact.



Remarks:

- There is no limitation on the times of using this instruction. However, the register No. designated in S3 cannot be repeated.
- For the 16-bit instruction, S3 will occupy 20 registers. In the program example above, the area designated in S3 is D100 ~ D119. Before the execution of PID instruction, you have to transmit the setting value to the designated register area by MOV instruction, If the designated registers are latched, use MOVP instruction to transmit all setting value at a time.
- Settings of S3 in the 16-bit instruction：

Device No.	Function	Setup Range	Explanation
S_3 :	Sampling time (T_S) (unit: 10ms)	1~2,000 (unit： 10ms)	If TS is less than 1 program scan time, PID instruction will be executed for 1 program scan time. If TS= 0, PID instruction will not be enabled. The minimum TS has to be longer than the program scan time
S_3 +1：	Proportional gain (K_P)	0~30,000(%)	The magnified error proportional value between SV － PV
S_3 +2:	Integral gain （K_I）	0~30,000(%)	For control mode K0~K8
S_3 +3:	Differential gain （K_D）	-30,000~30,000(%)	For control mode K0~K8
S_3 +4:	Control mode	0： automatic control 1： forward control (E = SV - PV) 2： inverse control (E = PV - SV) 3： Auto-tuning of parameter exclusively for the temperature control. The device will automatically become K4 when the auto-tuning is completed and be filled in with the appropriate parameter KP, KI and KD (not avaliable in the 32-bit instruction). 4： Exclusively for the adjusted temperature control (not avaliable in the 32-bit instruction) 5： Auto direction control (limited integrall upper/lower limit) 7： Manual control 1: Users set an MV. The accumulated integral value increases according to the error. It is suggested that the control mode should be used in a control environment which changes more slowly. 8： Manual control 2: Users set an MV. The accumulated integral value will stop increasing. When the control mode becomes the	

		automatic mode (the control mode K5 is used), the instruction PID outputs an appropriate accumulated integral value according to the last MV.	
S_3 +5:	The range that error value (E) doesn't work	0~32,767	E = the error of SV – PV. When S3 +5 = K0, the function will not be enabled, e.g. when S3 +5 is set as 5, MV of E between -5 and 5 will be 0.
S_3 +6:	Upper bound of output value (MV)	-32,768~32,767	Ex: if S3 +6 is set as 1,000, the output will be 1,000 when MV is bigger than 1,000. S3 +6 has to be bigger or equal S3 +7; otherwise the upper bound and lower bound will switch.
S_3 +7:	Lower bound of output value (MV)	-32,768~32,767	Ex: if S3 +7 is set as -1,000, the output will be -1,000 when MV is smaller than -1,000.
S_3 +8:	Upper bound of integral value	-32,768~32,767	Ex: if S3 +8 is set as 1,000, the output will be 1,000 when the integral value is bigger than 1,000 and the integration will stop. S3 +8 has to be bigger or equal S3 +9; otherwier the upper bound and lower bound will switch.
S_3 +9:	Lower bound of integral value	-32,768~32,767	Ex: if S3 +9 is set as -1,000, the output will be -1,000 when the integral value is smaller than -1,000 and the integration will stop. If S3+8 and S3+9 are set to 0, there will be no upper limit for integration.
S_3 +10、11:	Accumulated integral value	32-bit floating point	The accumulated integral value is only for reference. You can still clear or modify it (in 32-bit floating point) according to your need.
S_3 +12:	The previous PV	-32,768~32,767	The previous PV is only for reference. You can still modify it according to your need.
S_3 +13: ~ S_3 +19:	For system use only.		

- When parameter setting exceeds its range, the upper bound and lower bound will become the setting value. However, if the motion direction (DIR) exceeds the range, it will be set to 0.
- PID instruction can be used in interruption subroutines, step points and CJ instruction.
- The maximum error of sampling time TS = - (1 scan time + 1ms) ~ + (1 scan time). When the error affects the output, please fix the scan time or execute PID instruction in the interruption subroutine of the timer.

- PV of PID instruction has to be stable before the execution of PID instruction.
- For the 32-bit instruction, If S3 designates the parameter setting area of PID instruction as D100 ~ D120, S3 occupies 21 registers. Before the execution of PID instruction, you have to use MOV instrction first to send the setting value to the register area for setup. If the designated registers are latched one, use MOVP instruction to send all the setting value at a time.
- Settings of S3 in the 32-bit instruction:

Device No.	Function	Setup range	Explanation
S_3 :	Sampling time (TS) (unit: 10ms)	1 ~ 2,000 (unit: 10ms)	If TS is less than 1 program scan time, PID instruction will be executed for 1 program scan time. If TS= 0, PID instruction will not be enabled. The minimum TS has to be longer than the program scan time.
S_3 +1:	Proportional gain (K_P)	0~30,000(%)	The magnified error proportional value between SV － PV
S_3 +2:	Integral gain (K_I)	0~30,000(%)	For control mode K0~K2, K5
S_3 +3:	Differential gain (K_D)	-30,000~30,000(%)	For control mode K0~K2, K5
S_3 +4:	Control direction (DIR)	0：automatic control 1：forward control (E=SV-PV) 2：inverse control (E=PV-SV) 5：Automatic mode with MV upper/lower bound control. When MV reaches upper/lower bound, the accumulation of integral value stops.	
S_3 +5、6:	The range that 32-bit error value (E) doesn't work	0~2,147,483,647	E = the error of SV － PV. When S3 +5,6 = K0, the function will not be enabled, e.g. when S3 +5,6 is set as 5, MV of E between -5 and 5 will be 0.
S_3 +:7、8:	Upper bound of 32-bit output value (MV)	-2,147,483,648~ 2,147,483,647	Ex: if S3 +7,8 is set as 1,000, the output will be 1,000 when MV is bigger than 1,000. S3 +7,8 has to be bigger or equal S3 +9,10; otherwise the upper bound and lower bound will switch.
S_3 +9、10:	Lower bound of 32-bit output value (MV)	-2,147,483,648~ 2,147,483,647	Ex: if S3 +9,10 is set as -1,000, the output will be -1,000 when MV is smaller than -1,000.
S_3 +11、12:	Upper bound of 32-bit integral value	-2,147,483,648~ 2,147,483,647	Ex: if S3 +11,12 is set as 1,000, the output will be 1,000 when the integral value is bigger than 1,000 and the integration will stop. S3 +11,12 has to be bigger or equal S3 +13,14; otherwier the

			upper bound and lower bound will switch.
S_3 +13、14:	Lower bound of 32-bit integral value	-2,147,483,648~ 2,147,483,647	Ex: if S3 +13,14 is set as -1,000, the output will be -1,000 when the integral value is smaller than -1,000 and the integration will stop.
S_3 +15、16:	32-bit accumulated integral value	32-bit floating point	The accumulated integral value is only for reference. You can still clear or modify it (in 32-bit floating point) according to your need.
S_3 +17、18:	32-bit previous PV	--	The previous PV is only for reference. You can still modify it according to your need.
S_3 +19、20:	For system use only.		

The explanation of 32-bit S3 and 16-bit S3 are almost the same. The difference is the capacity of S3+5 ~ S3+20.

3、PID Equations:

- The PID operation is conducted according to the speed and the differential PV.
- The PID operation has three control directions: automatic, foreward and inverse. Forward or inverse are, designated in S3 +4. Other relevant settings of PID operation are set by the registers designated in S3 ~ S3 +5.
- Basic PID equation:

$$MV = K_P * E(t) + K_I * E(t)\frac{1}{S} + K_D * PV(t)S$$

Control direction	PID equation
Forward, automatic	E(t) = SV – PV
Inverse	E(t) = PV – SV

$PV(t)S$ is the differential value of $PV(t)$; $E(t)\frac{1}{S}$ is the integral value of $E(t)$.

When E(t) is less than 0 as the control direction is selected as forward or inverse, E(t) will be regarded as “0".

The equation above illustrates that this instruction is different from a general PID instruction by the variable use of the differential value. To avoid the flaw that the transient differential value is too big when a general PID instruction is executed for the first time, our PID instruction monitors the differentiation status of the PV. When the variation of PV is too big, this instruction will reduce the output of MV.

- Symbol explanation:

MV: Output value

K_P: Proprotional gain

$E(t)$: Error value

PV: Present measured value

SV: Target value

K_D: Differential gain

$PV(t)S$: Differential value of PV(t)

K_I: Integral gain

$E(t)\frac{1}{S}$: Integral value of E(t)

- Temperature Control Equation:

When S_3 +4 is K3 and K4, the equation used in diagram 2 (see below) will be changed as:

$$MV = \frac{1}{K_P}\left[E(t) + \frac{1}{K_I}\left(E(t)\frac{1}{S}\right) + K_D * PV(t)S\right]$$

In which the error value is fixed as E(t) = SV – PV

This equation is exclusively designed for temperature control. Therefore, when the sampling time (TS) is set as 4 seconds (K400), the range of output value (MV) will be K0 ~ K4,000 and the cycle time of GPWM instruction used together has to be set as 4 seconds (K4000) as well.

If you have no idea how to adjust the parameters, you can select K3 (auto-tuning) and after all the parameters are adjusted (the control direction will be automatically set as K4), you can modify your parameters to better ones according to the result of the control.

- Control diagrams:

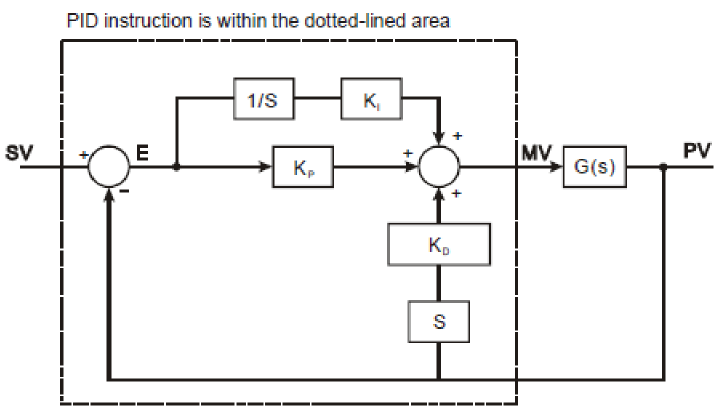


Diagram 1:S_3 + 4 = K0 ~ K2

In Diagram 1, S is differentiation, referring to "PV – previous PV / sampling time". 1 / S is integration, referring to "(previous integral value + error value) × sampling time". G(S) refers to the device being controlled.

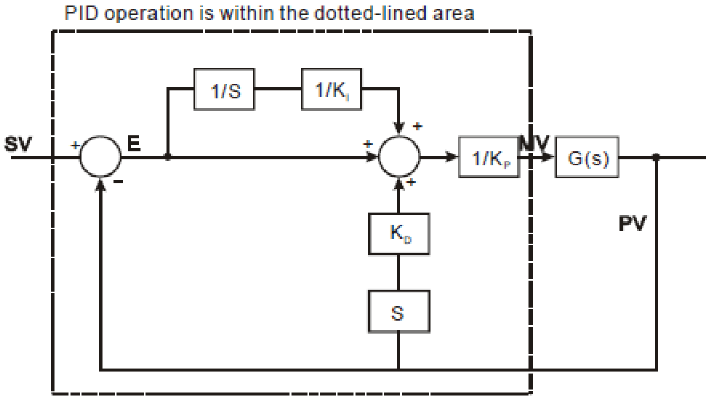
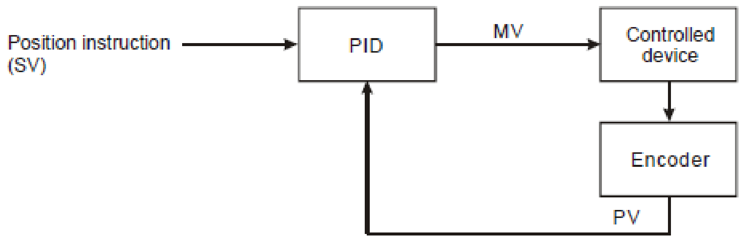


Diagram 2: S_3 + 4 = K3 ~ K4

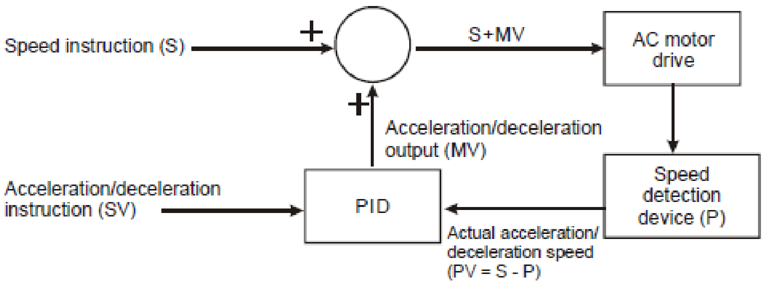
In Diagram 2, 1/KI and 1/KP refer to "divided by KI" and "divided by KP". Due to that this is exclusively for temperature control, you have to use PID instruction together with GPWM instruction. See Application 3 more details.

- Notes:
  1) There are a lot of circumstances where PID instruction can be applied; therefore, please choose the control functions appropriately. For example, when you select parameter auto-tuning for the temperature (S3 + 4 = K3), you cannot use it in a motor control environment in case improper control may occur.
  2) When you adjust the three main parameters, KP, KI and KD (S3 + 4 = K0 ~ K2), you have to adjust KP first (according to your experiences) and set KI and KD as 0. When you can roughly handle the control, you then adjust KI (increasingly) and KD (increasingly) (see example 4 below for how to adjust). KP = 100 refers to 100%, i.e. the gain of the error is 1. KP < 100% will decrease the error and KP > 100% will increase the error.
  3) When you select the parameter exclusively for temperature control (S_3 + 4 = K3, K4), it is suggested that you store the parameter in D register in the latched area in case the automatically adjusted parameter will disappear after the power is cut off. There is no guarantee that the adjusted parameter is suitable for every control. Therefore, you can modify the adjusted parameter according to your actual need, but it is suggested that you modify only K_I or K_D
  4) PID instruction can to work with many parameters; therefore please do not randomly modify the parameters in case the control cannot be executed normally.

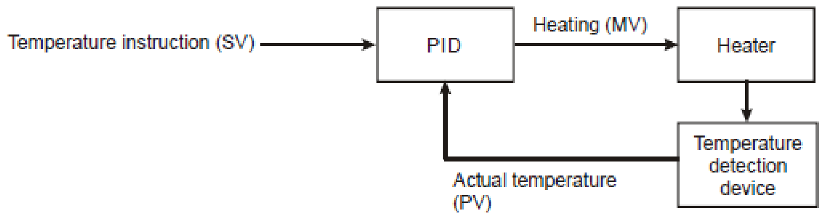
3、 Example 1: Diagram of using PID instruction in position control (S3 + 4 = 0)



Example 2: Diagram of using PID instruction with AC motor drive on the control (S3 + 4 = 0)



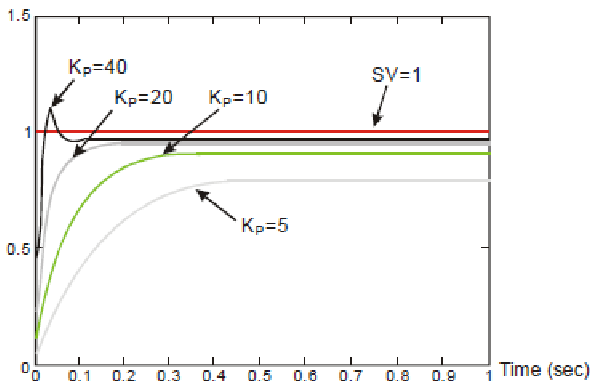
Example 3: Diagram of using PID instruction in temperature control (S3 + 4 = 1)



Example 4: How to adjust PID parameters

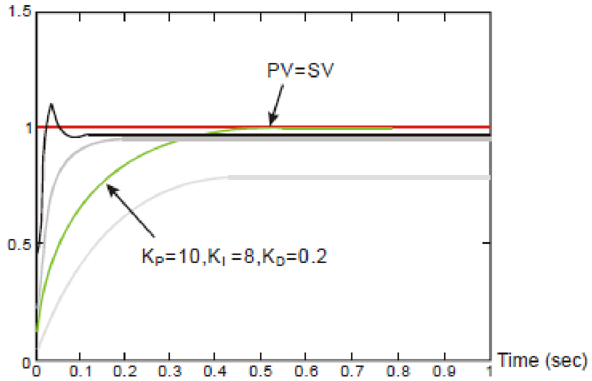
Assume that the transfer function of the controlled device G(S) in a control system is a first-order function $G(s)=\frac{b}{s+a}$ (most models of motors are first-order function), SV = 1, and sampling time (T_S) = 10ms, we suggest you to follow the steps below for adjusting the parameters.

Step 1: Set K_I and K_D as 0 and K_P as 5, 10, 20 and 40. Record the SV and PV respectively and the results are as the figure below.



Step 2: From the figure, we can see that when K_P = 40, there will be over-reaction, so we will not select it. When K_P = 20, the PV reaction curve will be close to SV and there will not be over-reaction, but due to its fast start-up with big transient MV, we will consider to put it aside. When K_P = 10, the PV reaction curve will get close to SV value more smoothly, so we will use it. Finally when K_P = 5, we will not consider it due to the slow reaction.

Step 3: Select K_P = 10 and adjust K_I from small to big (e.g. 1, 2, 4 to 8). K_I should not be bigger than K_P. Adjust K_D from small to big (e.g. 0.01, 0.05, 0.1 and 0.2). K_D should not exceed 10% of K_P. Finally we obtain the figure of PV and SV below.



Note: This example is only for your reference. Please adjust your parameters to proper ones according to your actual condition of the control system.

8 Application Instructions ZL100~ZL149

8.1 (ZL 100-109) Communication instructions

ZL 100	MODRD	S_1 S_2 D	Read Modbus Data

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODRD: 7 steps 16-bit
S_2					*	*							*			
n					*	*							*			

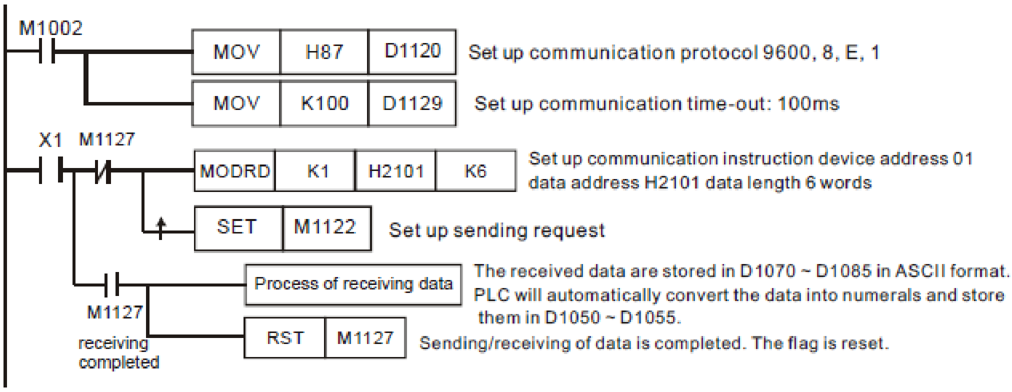
1、Explanations:

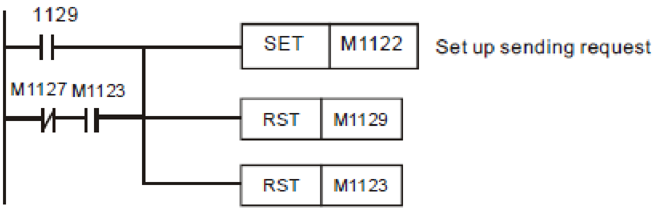
- S1: Address of communication device S2: Address of data to be read n: Length of read data
- Range of S1: K0 ~ K254

  Range of n: K1 ⩽ n ⩽ K6
- MODRD is a drive instruction exclusively for peripheral communication equipment in MODBUS ASCII mode/RTU mode.
- If the address of S2 is illegal to the designed communication device, the device will respond with an error, PLC will records the error code in D1130 and M1141 will be On.
- The feedback (returned) data from the peripheral equipment will be stored in D1070 ~ D1085. After receiving the feedback data is completed, PLC will auto-check if all data are correct. If there is an error, M1140 will be On.
- In ASCII mode, due to that the feedback data are all in ASCII, PLC will convert the feedback data into numerals and store them in D1050 ~ D1055. D1050 ~ D1055 will be invalid in RTU mode.
- After M1140 or M1141 turn On, the program will send a correct datum to the peripheral equipment. If the feedback datum is correct, M1140 and M1141 will be reset.

2、Program Example 1:

Communication between PLC and AC motor drives (ASCII Mode, M1143 = Off)





PLC → AC motor drives, PLC sends: “01 03 2101 0006 D4”

AC motor drives → PLC , PLC receives: “01 03 0C 0100 1766 0000 0000 0136 0000 3B”

Registers for sent data (sending messages)

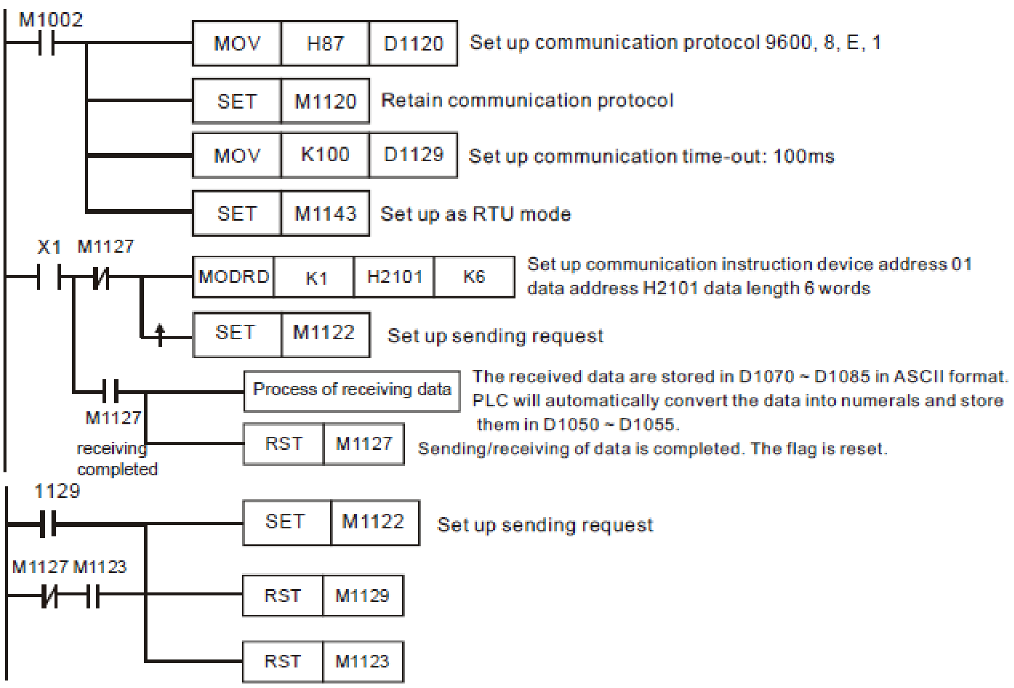
Register	DATA		Explanation	
D1089 low	‘0’	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D1089 high	‘1’	31 H	ADR 0	
D1090 low	‘0’	30 H	CMD 1	Instruction code: CMD (1,0)
D1090 high	‘3’	33 H	CMD 0	
D1091 low	‘2’	32 H	Starting data address	
D1091 high	‘1’	31 H		
D1092 low	‘0’	30 H		
D1092 high	‘1’	31 H		
D1093 low	‘0’	30 H	Number of data (counted by words)	
D1093 high	‘0’	30 H		
D1094 low	‘0’	30 H		
D1094 high	‘6’	36 H		
D1095 low	‘D’	44 H	LRC CHK 1	Checksum: LRC CHK (0,1)
D1095 high	‘4’	34 H	LRC CHK 0	

Registers for received data (responding messages)

Register	DATA		Explanation	
D1070 low	'0'	30 H	ADR 1	
D1070 high	'1'	31 H	ADR 0	
D1071 low	'0'	30 H	CMD 1	
D1071 high	'3'	33 H	CMD 0	
D1072 low	'0'	30 H	Number of data (counted by byte)	
D1072 high	'C'	43 H		
D1073 low	'0'	30 H	Content of address 2101 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1050 = 0100 H
D1073 high	'1'	31 H		
D1074 low	'0'	30 H		
D1074 high	'0'	30 H		
D1075 low	'1'	31 H	Content of address 2102 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1051 = 1766 H
D1075 high	'7'	37 H		
D1076 low	'6'	36 H		
D1076 high	'6'	36 H		
D1077 low	'0'	30 H	Content of address 2103 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1052 = 0000 H
D1077 high	'0'	30 H		
D1078 low	'0'	30 H		
D1078 high	'0'	30 H		
D1079 low	'0'	30 H	Content of address 2104 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1053 = 0000 H
D1079 high	'0'	30 H		
D1080 low	'0'	30 H		
D1080 high	'0'	30 H		
D1081 low	'0'	30 H	Content of address 2105 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1054 = 0136 H
D1081 high	'1'	31 H		
D1082 low	'3'	33 H		
D1082 high	'6'	36 H		
D1083 low	'0'	30 H	Content of address 2106 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1055 = 0000 H
D1083 high	'0'	30 H		
D1084 low	'0'	30 H		
D1084 high	'0'	30 H		
D1085 low	'3'	33 H	LRC CHK 1	
D1085 high	'B'	42 H	LRC CHK 0	

3、Program Example 2:

Communication between PLC and AC motor drives (RTU Mode, M1143 = On)



PLC → AC motor drives, PLC sends: "01 03 2102 0002 6F F7"

AC motor drives → PLC , PLC receives: "01 03 04 1770 0000 FE 5C"

Registers for sent data (sending messages)

Register	DATA	Explanation
D1089 low	01 H	Address
D1090 low	03 H	Function
D1091 low	21 H	Starting data address
D1092 low	02 H	
D1093 low	00 H	Number of data (counted by words)
D1094 low	02 H	
D1095 low	6F H	CRC CHK Low
D1096 low	F7 H	CRC CHK High

Registers for received data (responding messages)

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	03 H	Function
D1072 low	04 H	Number of data (counted by bytes)
D1073 low	17 H	Content of address 2102 H
D1074 low	70 H	
D1075 low	00 H	Content of address 2103 H
D1076 low	00 H	
D1077 low	FE H	CRC CHK Low
D1078 low	5C H	CRC CHK High

ZL 101	MODWR	S_1 S_2 n	Write Modbus Data

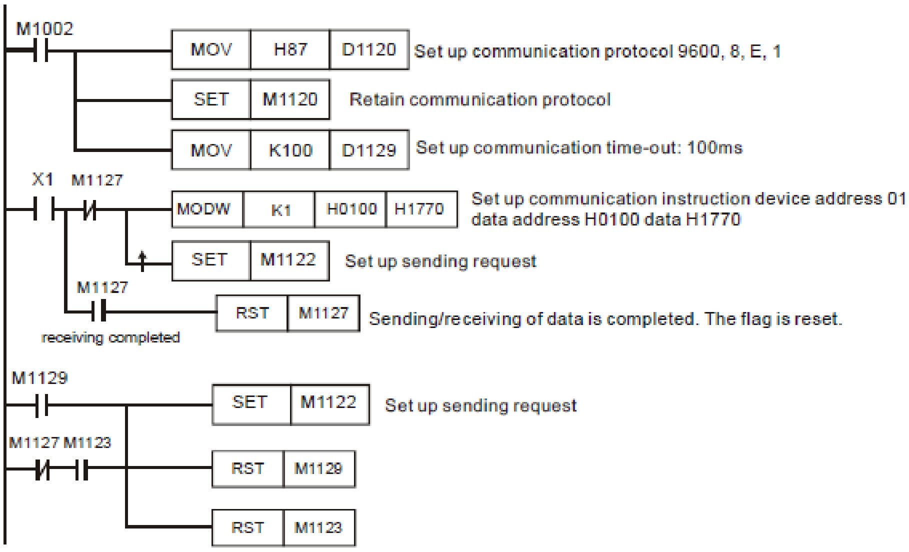
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODWR: 7 steps 16-bit
S_2					*	*							*			
n					*	*							*			

1、Explanations:

- S1: Address of communication device S2: Address of data to be read n: Data to be written
- Range of S1: K0 ~ K254
- MODWR is a drive instruction exclusively for peripheral communication equipment in MODBUS ASCII mode/RTU mode.
- The feedback (returned) data from the peripheral equipment will be stored in D1070 ~ D1076. After receiving the feedback data is completed, M1127 will be On.

2、Program Example 1:

Communication between PLC and AC motor drives (ASCII Mode, M1143 = Off)



PLC → AC motor drives, PLC sends: "01 06 0100 1770 71"

AC motor drives → PLC , PLC receives: "01 06 0100 1770 71"

Registers for sent data (sending messages)

Register	DATA		Explanation	
D1089 low	'0'	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D1089 high	'1'	31 H	ADR 0	
D1090 low	'0'	30 H	CMD 1	Instruction code: CMD (1,0)
D1090 high	'6'	36 H	CMD 0	
D1091 low	'0'	30 H	Data address	
D1091 high	'1'	31 H		
D1092 low	'0'	30 H		
D1092 high	'0'	30 H		
D1093 low	'1'	31 H	Data contents	
D1093 high	'7'	37 H		
D1094 low	'7'	37 H		
D1094 high	'0'	30 H		
D1095 low	'7'	37 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1095 high	'1'	31 H	LRC CHK 0	

PLC receiving data register (response messages)

Register	DATA		Explanation
D1070 low	'0'	30 H	ADR 1
D1070 high	'1'	31 H	ADR 0
D1071 low	'0'	30 H	CMD 1
D1071 high	'6'	36 H	CMD 0
D1072 low	'0'	30 H	Data address
D1072 high	'1'	31 H	
D1073 low	'0'	30 H	
D1073 high	'0'	30 H	
D1074 low	'1'	31 H	Data content
D1074 high	'7'	37 H	
D1075 low	'7'	37 H	
D1075 high	'0'	30 H	
D1076 low	'7'	37 H	LRC CHK 1
D1076 high	'1'	31 H	LRC CHK 0

ZL 102	RS1				S_1 S_2 S_3 S_4 n										COM1: RS232 data read and write
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1					*	*							*			RS1: 7 steps 16-bit
S_2					*	*							*			
n					*	*							*			

1、Explanations:

- This command has the functions of MODRD, MODWR and MODRW commands at the same time, which is more convenient to use.
- S1: Address of communication device (Unit Address) S2: Communication function code (Function Code).

Function code	Command description
K1	Reading several bit devices (same function as MODRD instruction)
K3	Reading single or several word devices (same function as MODRD instruction)
K6	Writing data in a single word device (same function as the MODWR instruction)
K15	Writing states in bit devices (same function as MODRW instruction)
K16	Writing data in word devices (same function as MODRW instruction)

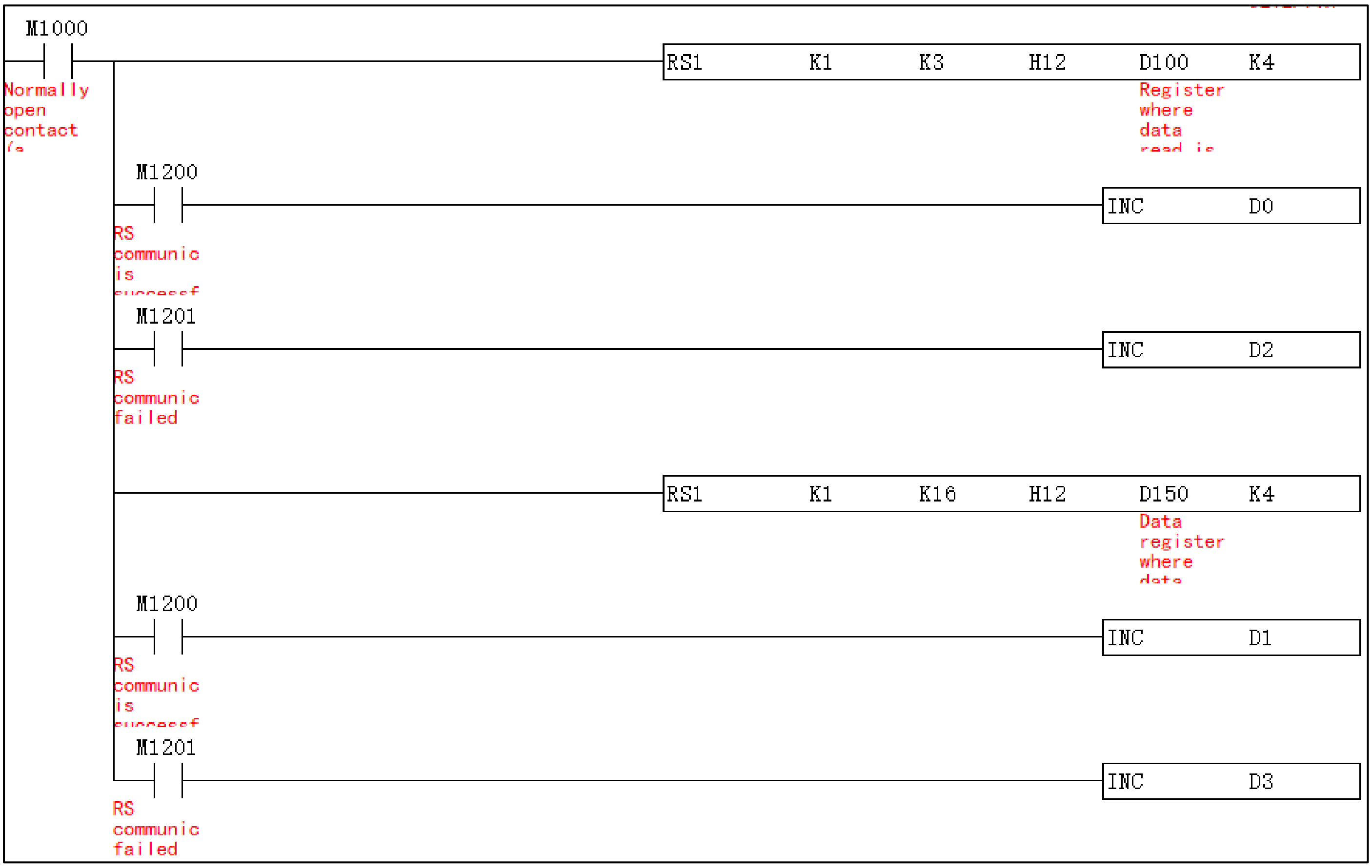
- S3, S4, n: Its functions vary according to different function codes.
- S3: The address of the data to be read and written.
- S4: Data to be read and written.
- n: read and write data length.
- S3, S4, n operands have the following functions according to different function codes:

Function code	S_3	S_4	n
K1(Reading several bit devices)	Address from which data is read	Register where data read is stored	Number of data read
K3 （Reading single or several word devices)	Address from which data is read	Register where data read is stored	Number of data read
K6（Writing data in a single word device)	Address from which data is read	Data register where data written is stored	None

K15 (Writing states in bit devices)	Address into which data is written	Data register where data written is stored	Number of data written
K16 (Writing data in word devices)	Address into which data is written	Data register where data written is stored	Number of data written

- There is no limit to the number of times this command can be used in the program, and multiple commands can be executed at the same time.

Eg.



- pecial M

special M	function
M1200	ON when the RS1/RS2/RS3 command communication is successful, the system will automatically OFF every time it is ON
M1201	ON when the RS1/RS2/RS3 command communication fails, the system will automatically OFF every time it is ON

ZL 103	RS2	S_1 S_2 S_3 S_4 n														COM2: RS485data read and write
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			RS2: 7 steps 16-bit
S_2					*	*							*			
n					*	*							*			

1、Instruction description: refer to ZL102 RS1 instruction description

ZL 104	RS3	S_1 S_2 S_3 S_4 n														COM3: RS485 data read and write
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			RS3: 7 steps 16-bit
S_2					*	*							*			
n					*	*							*			

1、Instruction description: refer to ZL102 RS1 instruction description

ZL 108		CRC		S n D											Checksum CRC Mode
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			CRC, CRCP: 7 steps 16-bit
n					*	*							*			
D													*			

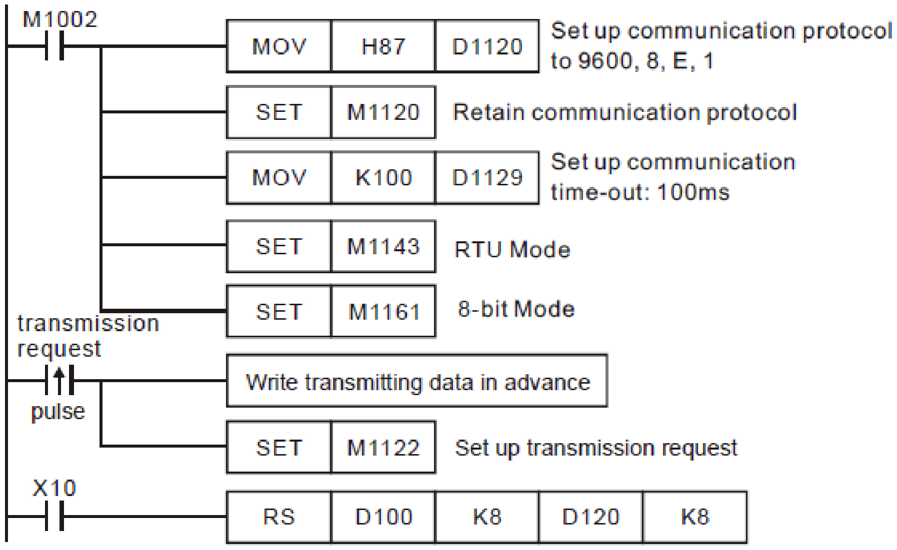
1、Explanations:

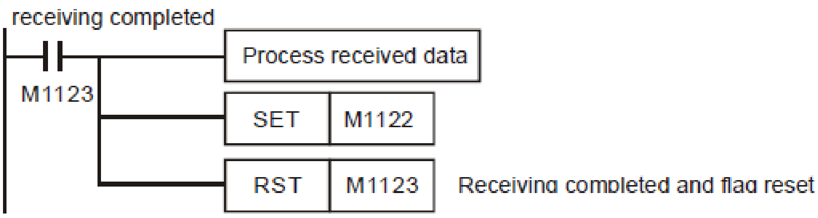
- S: Start operation device for RTU mode checksum n: Number of calculated bits

  D: Start device for storing the operation result CRC checksum: See remarks
- Range of n: K1 ~ K256
- If n does not fall within its range, an operation error will occur, the instruction will not be executed, M1067, M1068 = On and D1067 will record the error code H’0E1A.
- In 16-bit conversion mode: When M1161 = Off, S divides its hex data area into higher 8 bits and lower 8 bits and performs CRC checksum operation on each bit. The data will be sent to the higher 8 bits and lower 8 bits in D. n = the number of calculated bits.
- In 8-bit conversion mode: When M1161 = On, S divides its hex data area into higher 8 bits (invalid data) and lower 8 bits and performs CRC checksum operation on each bit. The data will be sent to the lower 8 bits in D and occupy 2 registers. n = the number of calculated bits. (All higher 8 bits in D are “0”.)

2、Program Example:

- When PLC communicates with AC motor drives (In RTU mode, M1143 = On), (In 16-bit mode, M1161 = On), the sent data write in advance H12 into H2000 of AC motor drives.



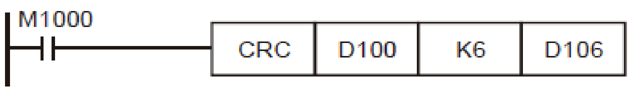


PLC → AC motor drives, PLC sends: “01 06 2000 0012 02 07”

Registers for sent data (sending messages)

Register	DATA	Explanation
D100 low	01 H	Address
D101 low	06 H	Function
D102 low	20 H	Data address
D103 low	00 H	
D104 low	00 H	Data content
D105 low	12 H	
D106 low	02 H	CRC CHK 0
D107 low	07 H	CRC CHK 1

The error checksum CRC CHK (0,1) can be calculated by CRC instruction (in 8-bit mode, M1161 = On).



CRC checksum: 02 H is stored in the lower 8 bits of D106 and 07 H in the lower 8 bits of D107

Remarks:

1) The format of RTU mode with a communication datum:

START	Time interval
Address	Communication address: 8-bit binary
Function	Function code: 8-bit binary
DATA (n-1)	Data content:

…….	n × 8-bit data
DATA 0	
CRC CHK Low	CRC checksum:
CRC CHK High	16-bit CRC checksum consists of 2 8-bit binaries
END	Time interval

2) CRC checksum starts from Address and ends at Data content.

The operation of CRC checksum:

Step 1: Make the 16-bit register (CRC register) = FFFFH

Step 2: Exclusive OR the first 8-bit byte message instruction and the low-bit 16-bit CRC register. Store the result in CRC register.

Step 3: Shift the CRC register one bit to the right and fill 0 in the higher bit.

Step 4: Check the value that shifts to the right. If it is 0, store the new value from Step 3 into the CRC register, otherwise, Exclusive OR A001H and the CRC register, and store the result in the CRC register.

Step 5: Repeat Step 3 ~ 4 and finish calculating the 8 bits.

Step 6: Repeat Steps 2 ~ 5 for obtaining the next 8-bit message instruction until all the message instructions are calculated. In the end, the obtained CRC register value is the CRC checksum. Be aware that CRC checksum should be placed in the checksum of the message instruction.

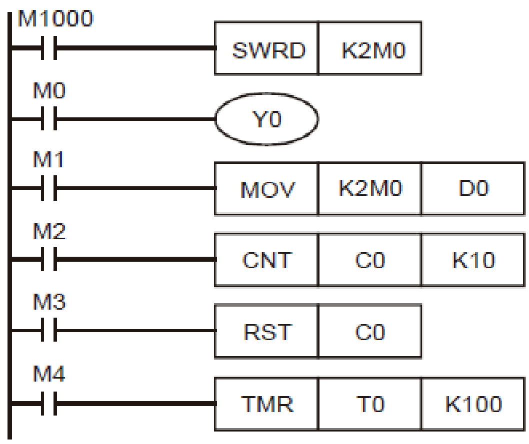
ZL 109	SWRD					D										Read Digital Switch
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SWRD: 3 steps 16-bit
D								*	*	*	*	*	*	*	*	

1、Explanations:

- D: Device for storing the read value.
- This instruction stores the value read from digital switch function card into D.
- The read value is stored in the low byte in D. Every switch has a corresponding bit.
- When there is no digital function card inserted, the error message C400 (hex) will appear in grammar check.

2、Program Example:

- There are I 8 DIP switches on the digital switch function card. After the switches are read by SWRD instruction, the status of each switch will correspond to M0 ~ M7.



- The status of M0 ~ M7 can be executed by each contact instruction.
- The execution of END instruction indicates that the process of input is completed. REF (I/O refresh) instruction will be invalid.
- When SWRD instruction uses the data in digital switch function card, it can read minimum 4 bits (K1Y*, K1M* or K1S*).

Remarks:

When digital switch function card is inserted, the status of the 8 DIP switches will correspond to M1104 ~ M1111。

8.2 (ZL 110-119) Floating point arithmetic

ZL 110	D	ECMP		S_1 S_2 D	Floating Point Compare

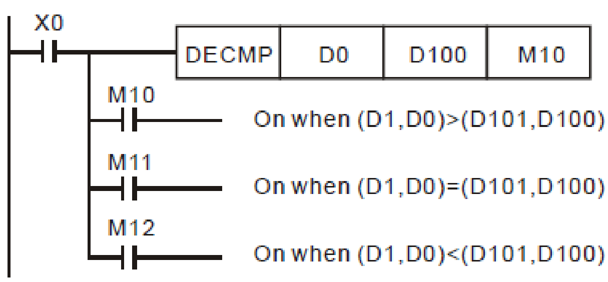
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DECMP, DECMPP: 13 steps 32-bit
S_2					*	*							*			
D		*	*	*												

1、Explanations:

- S1: Binary floating point comparison value 1 S2: Binary floating point comparison value 2
  D: Comparison result
- D occupies 3 consecutive devices.
- The binary floating point values S1 and S2 are compared with each other. The comparison result (>, =, <) is stored in D.
- If S1 or S2 is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the comparison.

2、Program Example:

- Designated device M10 and M10 ~ M12 are automatically occupied.
- When X0 = On. DECMP instruction will be executed and one of M10 ~ M12 will be On. When X0 = Off, DECMP instruction will not be executed and M10 ~ M12 will remain their status before X0 = Off.
- To obtain results ≧, ≦, ,≠ serial-parallel M10 ~ M12.
- Use RST or ZRST instruction to clear the result.



ZL 111	D	EZCP						S_1 S_2 S D								Floating Point Zone Compare
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEZCP: 17 steps 32-bit
S_2					*	*							*			
S					*	*							*			
D		*	*	*												

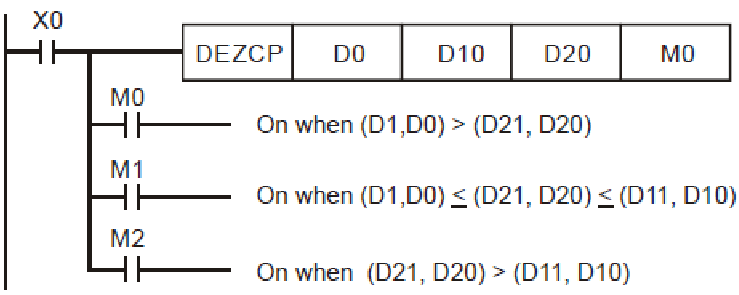
1、Explanations:

- S1: Lower bound of binary floating point S2: Upper bound of binary floating point

  S: Binary floating point comparison result D: Comparison result
- D occupied 3 consecutive devices.
- S1 ⩽ S2. See the specifications of each model for their range of use.
- S is compared with S1 and S2 and the result (>, =, <) is stored in D.
- If S1 or S2 is andesignated constant K or H, the instruction will convert the constant into a binary floating point value before the comparison.
- When S1 > S2, S1 will be used as upper/lower bound for the comparison.

2、Program Example:

- Designated device M0 and M0 ~ M2 are automatically occupied.
- When X0 = On. DEZCP instruction will be executed and one of M0 ~ M2 will be On. When X0 = Off, EZCP instruction will not be executed and M0 ~ M2 will remain their status before X0 = Off.
- Use RST or ZRST instruction to clear the result.



ZL 112	D	MOVR	S D	Move Floating Point Data

	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
	Bit Devices				Word Devices										
S															
D								*	*	*	*	*	*		

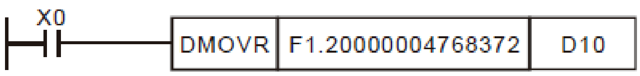
DMOVR: 9 steps 32-bit

1、Explanations:

- S: Source floating point data D: Destination device
- S can only be a floating-point constant value.
- This instruction is able to enter floating point values directly in S.
- When the instruction is executed, the content in S is moved directly into D. When the instruction is not executed, the content in D will not be modified.
- If users want to move the floating-point value in registers, they have to use DMOV.
- To execute the pulse type, add the NP rising edge "↑" command before the command.

2、Program Example:

- User DMOVR instruction to move 32-bit floating point data.
- When X0 = Off, the content in (D11、D10) remains unchanged. When X0 = On, the present value F1.20000004768372 will be moved to data registers (D11, D10).



ZL 114		MUL16	S1 S2 D	16-bit Multiplication
	D	MUL32		32-bit Multiplication

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*	*	
S2					*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	

MUL ,: 7 steps 16-bit

DMUL : 13 steps 32-bit

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S1: Multiplicand S2: Multiplicator D: Product
- In 16-bit instruction, D occupies one device.
- In 32-bit instruction, D occupies 2 consecutive devices.
- This instruction multiplies S1 by S2 in BIN format and stores the result in D. Be careful with the positive/negative signs of S1, S2 and D when doing 16-bit and 32-bit operations.
- In 16-bit BIN multiplication.

S1	S2	D
b15................ b0 x	b15................ b0 =	b15................ b0
b15 is a symbol bit.	b15 is a symbol bit.	b15 is a symbol bit.

16 bits x 16 bits = 16 bits
Symbol bit = 0 refers to a positive value.
Symbol bit = 1 refers to a negative value.

When D serves as a bit device, it can designate K1 ~ K4 and construct a 16-bit result, occupying 16-bit data.

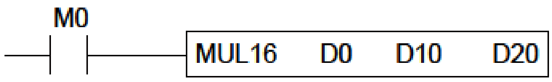
- In 32-bit BIN multiplication.

S1+1	S1	S2+1	S2	D+1	D
b31.......b16	b15.........b0 x	b31.......b16	b15.........b0 =	b31.......b16	b15.........b0
b31 is a symbol bit		b31 is a symbol bit		b31 is a symbol bit	

32 bits x 32 bits = 32 bits
Symbol bit = 0 refers to a positive value.
Symbol bit = 1 refers to a negative value.

2、Program Example 1:

- If M0 is On, the 16-bit D0 is multiplied by the 16-bit D10 and a 16-bit product is produced. The 16-bit data is stored in D20. On/Off of the most left bit indicates the positive/negative status of the result value.



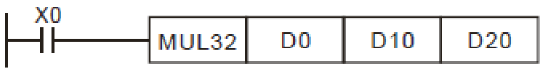
16 bits × 16 bits = 16 bits

D0 × D10 = D20

D0=K100, D10=K200, D20=K10,000

3、Program Example 2:

If X0 is On, the 32-bit value K10,00 in (D1, D0) is multiplied by the 32-bit value K20,000 in (D11, D10) and a 32-bit product is produced. The 32-bit data is stored in (D21, D20). On/Off of the most left bit indicates the positive/negative status of the result value.



32 bits × 32 bits = 32 bits

(D1,D0) ×(D11,D10) = (D21,D20)

(D1,D0)=K10,000, (D11,D10)=K20,000, (D21, D20)=K200,000,000

Remarks:

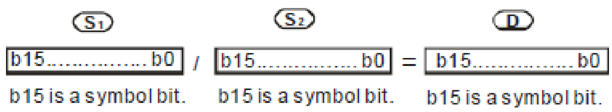
1) If the value gotten from the 16-bit multiplication can not be represented by a 16-bit signed value, and is greater than the maximum 16-bit positive value K32767 or less than the minimum 16-bit negative value K-32768, the low 16-bit data is stored.
2) If users need to get a complete value (32-bit value) from a 16-bit multiplication, they have to use API22 MUL/MULP. Please refer to the explanation of API22 MUL/MULP for more information.
3) If the value gotten from the 32-bit multiplication can not be represented by a 32-bit signed value, and is greater than the maximum 32-bit positive value K2147483647 or less than the minimum 16-bit negative value K-2147483648, the low 32-bit data is stored.
4) If users need to get a complete value (64-bit value) from a 32bit multiplication, they have to use ZL 22 DMUL/DMULP. Please refer to the explanation of API22 DMUL/DMULP for more information.

ZL 115	DIV16 DIV32	S1 S2 D	16-bit Division 32-bit Division

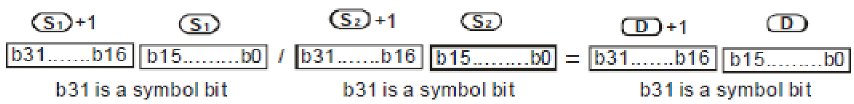
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DIV: 7 steps 16-bit DDIV: 13 steps 32-bit
S1					*	*	*	*	*	*	*	*	*	*		
S2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S1: Dividend S2: Divisor D: Quotient and remainder.
- In 16-bit instruction, D occupies one device.
- In 32-bit instruction, D occupies 2 consecutive devices.
- This instruction divides S1 and S2 in BIN format and stores the result in D. Be careful with the positive/negative signs of S1, S2 and D when doing 16-bit and 32-bit operations.
- This instruction will not be executed when the divisor is 0.
- In 16-bit BIN division:



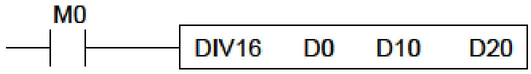
- When D serves as a bit device, it can designate K1 ~ K4 and construct a 16-bit result, occupying a 16-bit quotient.
- In 32-bit BIN division:



When D serves as a bit device, it can designate K1 ~ K8 and construct a 32-bit result, occupying a 32-bit quotient.

2、Program Example 1:

- If M0 = On, the value in D0 (K103) will be divided by the value in D10 (K5) and the quotient will be stored in D20. On/Off of the highest bit indicates the positive/negative status of the result value.



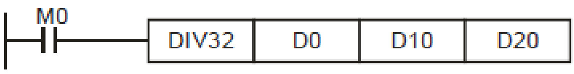
D0/D10=D20

⇨ K103/K5=K20. The remainder is K3.

⇨ D20=K20 (The remainder is left out.)

3、Program Example 2:

- If M0 = On, the value in (D1, D0) (K81,000) will be divided by the value in (D11, D10) (K40,000) and the quotient will be stored in (D21, D20). On/Off of the highest bit indicates the positive/negative status of the result value.



(D1,D0)/(D11,D10)=(D21,D20)

⇨ K81,000/K40,000=K2. The remainder is K1,000.

⇨ (D21,D20)=K2 (The remainder is left out.)

Remarks：

- If users need to record a remainder by a 16-bit division, they have to use API23 DIV/DIVP. Please refer to the explanation of API23 DIV/DIVP for more information.
- If users need to record a remainder by a 32-bit division, they have to use API23 DDIV/DDIVP. Please refer to the explanation of API23 DDIV/DDIVP for more information.

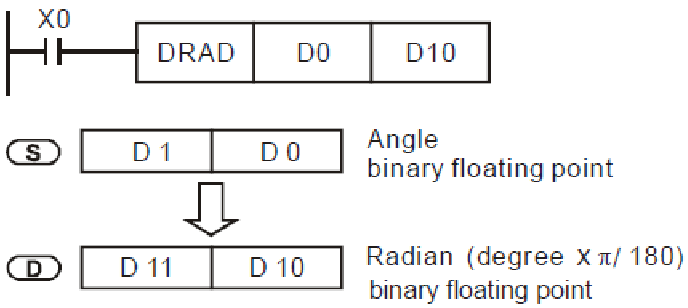
ZL 116	D	RAD		S D												Angle → Radian
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DRAD, DRADP: 9 steps
S					*	*							*			32-bit
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source (angle)   D: Result (radian)
- Radian = degree × (π/180)

2、Program Example:

- When X0 = On, designate the degree of binary floating point (D1, D0). Convert the angle into radian and store the result in binary floating point in (D11, D10)。



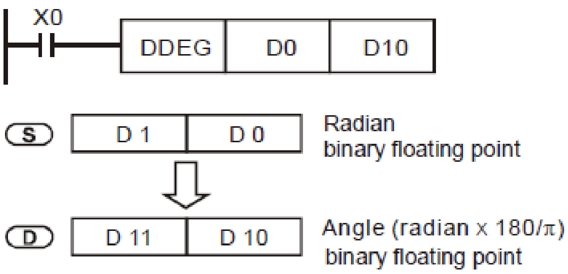
ZL 117		DEG		S D												Radian→Angle
	D															
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DDEG, DDEGP: 9 steps
S					*	*							*			32-bit
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source (radian)   D: Result (angle)
- Degree = radian × (180/π)

2、Program Example:

When X0 = On, designate the angle of binary floating point (D1, D0). Convert the radian into angle and store the result in binary floating point in (D11, D10).



ZL 118		EBCD		S D	Float to Scientific Conversion
	D				

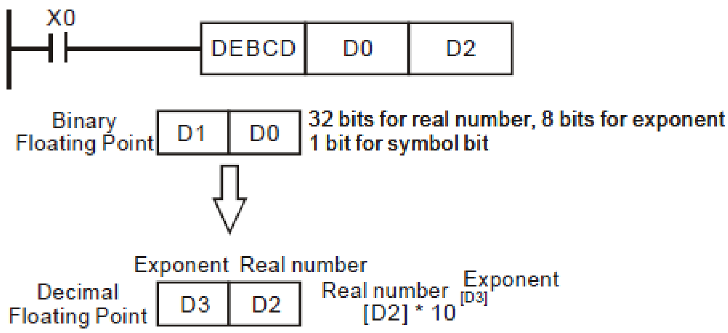
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DEBCD, DEBCDP: 9 steps
S													*			32-bit
D													*			

1、Explanations:

- S: Source D: Result
- This instruction converts binary floating point value in the register designated by S into decimal floating point value and stores it in the register designated by D.
- PLC conducts floating point operation in binary format. DEBCD instruction is exclusively for converting floating points from binary to decimal.

2、Program Example:

- When X0 = On, the binary floating points in D1 and D0 will be converted into decimal floating points and stored in D3 and D2.



ZL 119		EBIN		S	D	Scientific to Float Conversion
	D					

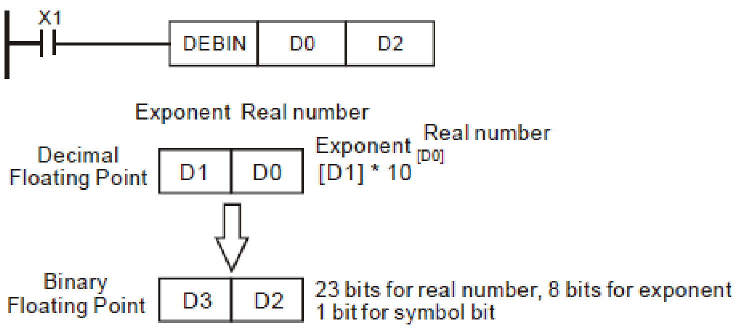
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
	Bit Devices				Word Devices											
S													*			DEBIN, DEBINP: 9 steps
D													*			32-bit

1、Explanations:

- S: Source　　D: Result
- This instruction converts decimal floating point value in the register designated by S into binary floating point value and stores it in the register designated by D.
- DEBIN instruction is exclusively for converting floating points from decimal to binary.
- Range of decimal floating point real numbers: -9.999 ~ +9,999. Range of exponants: -41 ~ +35. Range of PLC decimal floating points: $\pm 1,175 \times 10^{-41} \sim \pm 3,402 \times 10^{+35}$

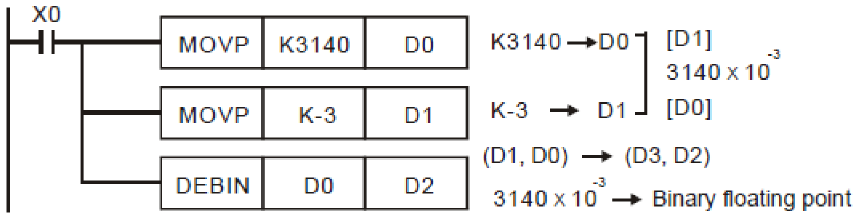
2、Program Example 1:

- When X1 = On, the decimal floating points in D1 and D0 will be converted into binary floating points and stored in D3 and D2.



3、Program Example 2:

- Use FLT instruction (API 149) to convert BIN integer into binary floating point before performing floating point operation. The value to be converted must be BIN integer and use DEBIN instruction to convert the floating point into a binary one.
- When X0 = On, move K3,140 to D0 and K-3 to D1 to generate decimal floating point ($3.14 = 3140 \times 10^{-3}$)。


X0
MOVP
K3140
D0
MOVP
K-3
D1
DEBIN
D0
D2
K3140 → D0
K-3 → D1
[D1]
3140 x 10^-3
[D0]
(D1, D0) → (D3, D2)
3140 x 10^-3 → Binary floating point

8.3 (ZL 120-129) Floating point arithmetic

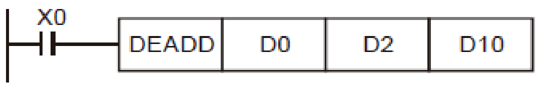
<table>
<tr><td>ZL 120</td><td></td><td colspan="4" rowspan="2">EADD</td><td></td><td colspan="8" rowspan="2">S1 S2 D</td><td colspan="2" rowspan="2">Floating Point Addition</td></tr>
<tr><td></td><td>D</td><td></td></tr>
<tr><td></td><td colspan="4">Bit Devices</td><td colspan="11">Word Devices</td><td rowspan="5">DEADD: 13 steps
32-bit</td></tr>
<tr><td></td><td>X</td><td>Y</td><td>M</td><td>S</td><td>K</td><td>H</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>T</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td>S1</td><td></td><td></td><td></td><td></td><td>*</td><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>S2</td><td></td><td></td><td></td><td></td><td>*</td><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
</table>

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S1: Summand S2: Addend D: Sum
- S1 + S2 = D. The floating point value in the register designated by S1 and S2 are added up and the result is stored in the register designated by D. The addition is conducted in binary floating point system.
- If S1 or S2 is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
- S1 and S2 can designate the same register. In this case, if the "continuous execution" instruction is in use, during the period when the criteria contact in On, the register will be added once in every scan by pulse execution instruction DEADDP.

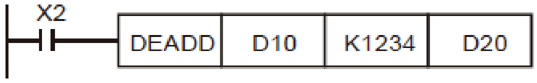
2、Program Example 1:

- When X0 = On, binary floating point (D1, D0) + binary floating point (D3, D2) and the result is stored in (D11, D10).



3、Program Example 2:

- When X2 = On, binary floating point (D11, D10) + K1234 (automatically converted into binary floating point) and the result is stored in (D21, D20).


X2
DEADD
D10
K1234
D20

ZL 121	D	ESUB	S1 S2 D	Floating Point Subtraction

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*							*		
S2					*	*							*		
D													*		

DESUB: 13 steps

32-bit

1、Explanations:

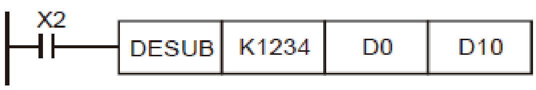
- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S1: Minuend S2: Subtrahend D: Remainder
- S1 − S2 = D. The floating point value in the register designated by S2 is subtracted from the floating point value in the register assigned by S1 and the result is stored in the register designated by D. The subtraction is conducted in binary floating point system.
- If S1 or S2 is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
- S1 and S2 can designate the same register. In this case, if the "continuous execution" instruction is in use, during the period when the criteria contact in On, the register will be subtracted once in every scan by pulse execution instruction DESUBP.

2、Program Example 1:

- When X0 = On, binary floating point (D1, D0) – binary floating point (D3, D2) and the result is stored in (D11, D10).

3、Program Example 2:

- When X2 = On, K1234 (automatically converted into binary floating point) – binary floating point (D1, D0) and the result is stored in (D11, D10).



ZL 122	D	EMUL	S1 S2 D	Floating Point Multiplication

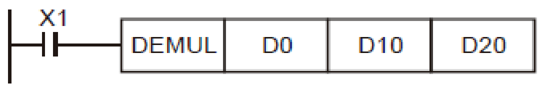
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DEMUL: 13 steps 32-bit
S1					*	*							*			
S2					*	*							*			
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S1: Multiplicand   S2: Multiplicator   D: Product
- S1 × S2 = D. The floating point value in the register assigned by S1 is multiplied with the floating point value in the register designated by S2 and the result is stored in the register designated by D. The multiplication is conducted in binary floating point system.
- If S1 or S2 is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
- S1 and S2 can designate the same register. In this case, if the "continuous execution" instruction is in use, during the period when the criteria contact in On, the register will be multiplied once in every scan by pulse execution instruction DEMULP.

2、Program Example :

- When X1 = On, binary floating point (D1, D0) × binary floating point (D11, D10) and the result is stored in (D21, D20).



- When X2 = On, K1234 (automatically converted into binary floating point) × binary floating point (D1, D0) and the result is stored in (D11, D10).

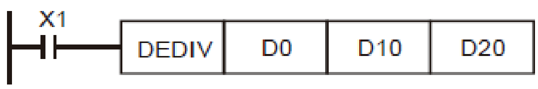
ZL 123	D	EDIV						S1 S2 D								Floating Point Division
	Bit Devices				Word Devices											DEDIV: 13 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*							*			
S2					*	*							*			
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge " ↑ " command before the command.
- S1: Dividend S2: Divisor D: Quotient and remainder
- S1 ÷ S2 = D. The floating point value in the register designated by S1 is divided by the floating point value in the register assigned by S2 and the result is stored in the register designated by D. The division is conducted in binary floating point system.
- If S1 or S2 is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
- If S2 = 0, operation error will occur, the instruction will not be executed.

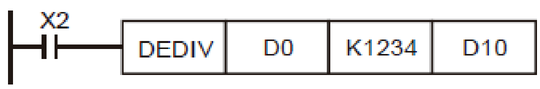
2、Program Example 1:

- When X1 = On, binary floating point (D1, D0) ÷ binary floating point (D11, D10) and the quotient is stored in (D21, D20).



3、Program Example 2:

- When X2 = On, binary floating point (D1, D0) ÷ K1234 (automatically converted into binary floating point) and the result is stored in (D11, D10).



ZL 124			
D	EXP	S D	Exponent of Binary Floating Point

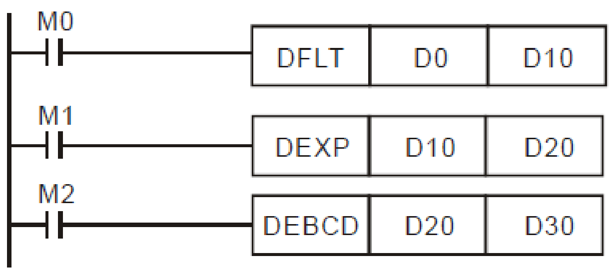
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
	Bit Devices				Word Devices											DEXP: 13 steps
S					*	*							*			32-bit
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Device for operation source D: Device for operation result
- e = 2.71828 as the base and S as exponent for EXP operation: EXP $^{[D+1, D]}$ = [S +1, S] .
- Both positive and negative values are valid for S. When designating D registers, the data should be 32-bit and the operation should be performed in floating point system. Therefore, S should be converted into a floating point value.
- The content in D = e^{S}; e = 2.71828, S = designated source data

2、Program Example:

- When M0 = On, convert (D1, D0) into binary floating point and store it in register (D11, D10).
- When M1= On, use (D11, D10) as the exponent for EXP operation and store the binary floating point result in register (D21, D20).
- When M2 = On, convert the binary floating point (D21, D20) into decimal floating point (D30 × 10[D31]) and store it in register (D31, D30).



ZL 125	D	LN	S D	Natural Logarithm of Binary Floating Point

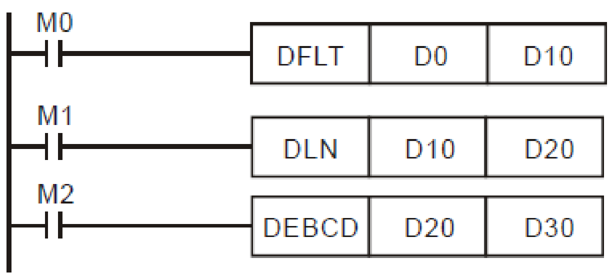
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DLN: 9 steps
S					*	*							*			32-bit
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Device for operation source D: Device for operation result
- This instruction performs natural logarithm "LN" operation by S: LN〔S + 1, S〕=〔D + 1, D〕.
- Only positive values are valid for S. When designating D registers, the data should be 32-bit and the operation should be performed in floating point system. Therefore, S should be converted into a floating point value.
- $e^D = S$ .The content in D = lnS; S = designated source data.

2、Program Example:

- When M0 = On, convert (D1, D0) into binary floating point and store it in register (D11, D10).
- When M1= On, use register (D11, D10) as the real number for LN operation and store the binary floating point result in register (D21, D20).
- When M2 = On, convert the binary floating point (D21, D20) into decimal floating point ($D30 \times 10^{[D31]}$) and store it in register (D31, D30)



ZL 126	D	LOG					S1 S2 D									Logarithm of Binary Floating Point
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DLOG: 13 steps 32-bit
S1					*	*							*			
S2					*	*							*			
D													*			

1、Explanations:

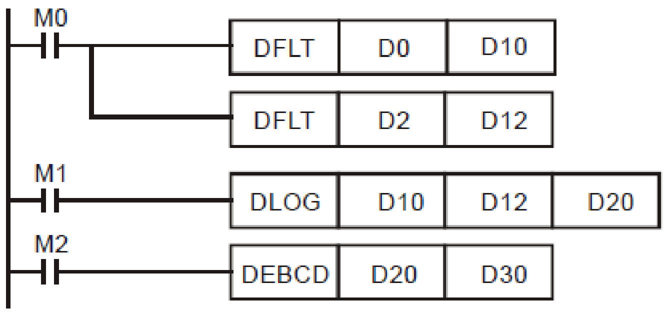
- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S1: Device for base S2: Device for operation source D: Device for operation result
- This instruction performs "log" operation of the content in S1 and S2 and stores the result in D.
- Only positives are valid for the content in S1 and S2. When designating D registers, the data should be 32-bit and the operation should be performed in floating point system. Therefore, S1 and S2 should be converted into floating point values.
- $S1^{D} = S2$, $D = ?$ → $Log_{S1}{}^{S2} = D$

  Example: Assume $S_1 = 5$, $S_2 = 125$, $D = \log_5{}^{125} = ?$

  $S_1{}^{D} = S_2 \rightarrow 5^{D} = 125 \rightarrow D = \log_5{}^{125} = 3$

2、Program Example:

- When M0 = On, convert (D1, D0) and (D3, D2) into binary floating points and store them in the 32-bit registers (D11, D10) and (D13, D12).
- When M1= On, perform log operation on the binary floting points in 32-bit registers (D11, D10) and (D13, D12) and store the result in the 32-bit register (D21, D20).
- When M2 = On, convert the binary floating point (D21, D20) into decimal floating point ($D30 \times 10^{[D31]}$) and store it in register (D31, D30).



ZL 127		ESQR				S D										Floating Point Square Root
	D															
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DESQR: 9 steps
S					*	*							*			32-bit
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source device D: Operation result
- Range of S: ⩾ 0
- This instruction performs a square root operation on the content in the register designated by S and stores the result in the register designated by D. The square root operation is performed in floating point system.
- If S is an designated constant K or H, the instruction will convert the constant into a binary floating point value before the operation.
- S can only be a positive value. Performing any square root operation on a negative value will result in an "operation error" and this instruction will not be executed.

2、Program Example 1:

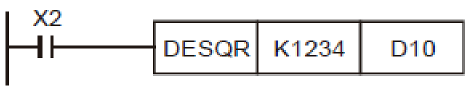
- When M0 = On, calculate the square root of the binary floating point (D1, D0) and store the result in register (D11, D10)。

$\sqrt{(D1, D0)}$ → (D11, D10)

binary floating point binary floating point

3、Program Example 2:

- When M2 = On, calculate the square root of K1,234 (automatically converted into binary floating point) and store the result in register (D11, D10).



ZL 128		POW		S1 S2 D	Floating Point Power Operation
	D				

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DPOW: 13 steps 32-bit
S1					*	*							*			
S2					*	*							*			
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge " ↑ " command before the command.
- S1: Device for base. S2: Device for exponent. D: Device for operation result
- This instruction performs power multiplication of binary floating point S1 and S2 and stores the result in D.

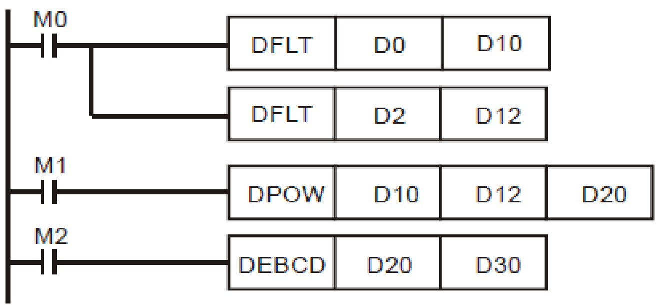
  D = POW〔S1 + 1, S1〕^〔S2 + 1, S2〕
- Only positives are valid for the content in S1. Both positives and negatives are valid for the content in S2. When designating D registers, the data should be 32-bit and the operation should be performed in floating point system. Therefore, S1 and S2 should be converted into floating point values.

  Example: When $S1^{S2}$ = D, D = ?

  Assume S1 = 5, S2 = 3, D = 5^3 =125

2、Program Example:

- When M0 = On, convert (D1, D0) and (D3, D2) into binary floating points and store them in the 32-bit registers (D11, D10) and (D13, D12).
- When M1= On, perform POW operation on the binary floting points in 32-bit registers (D11, D10) and (D13, D12) and store the result in the 32-bit register (D21, D20).
- When M2 = On, convert the binary floating point (D21, D20) into decimal floating point (D30 × 10[D31]) and store it in register (D31, D30).



ZL 129	D	INT					S	D								Float to Integer
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	INT: 5 steps 16-bit
S													*			DINT: 9 steps 32-bit
D													*			

1、Explanations:

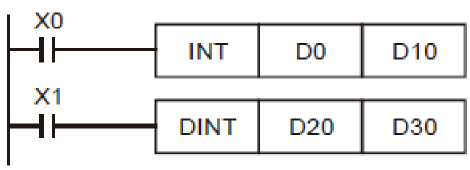
- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source device   D: Converted result
- S occupies 2 consecutive devices. See the specifications of each model for their range of use.
- The binary floating point value of the register designated by S is converted to BIN integer and stored in the register designated by D. The decimal of BIN integer is left out.
- 3. This instruction is the inverse operation of API 49 FLT instruction.

  16- bit instruction: -32,768 ~ 32,767

  32-bit instruction: -2,147,483,648 ~ 2,147,483,647

2、Program Example:

- When X0 = On, the binary floating point (D1, D0) will be converted into BIN integer and the result will be stored in (D10). The decimal of BIN integer will be left out.
- When X1 = On, the binary floating point (D21, D20) will be converted into BIN integer and the result will be stored in (D31, D30). The decimal of BIN integer will be left out.

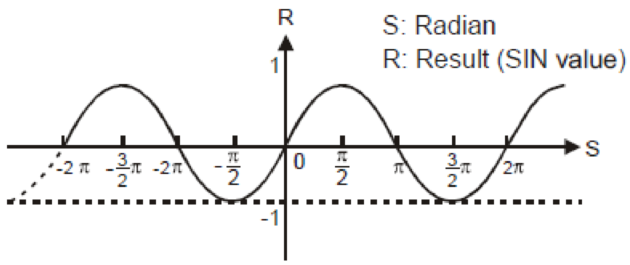


8.4 (ZL 130-139) Trigonometric operation

ZL 130		SIN			S D											Sine
	D															
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DSIN 9 steps
S					*	*							*			32-bit
D													*			

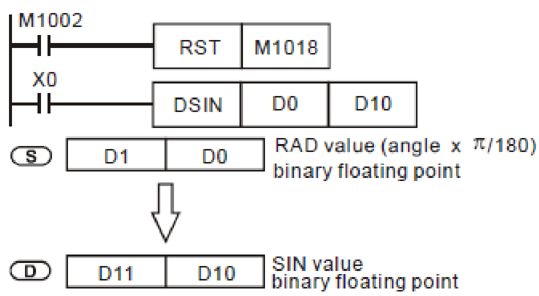
1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source value D: SIN result
- 0° ⩽ S < 360° . See the specifications of each model for their range of use.
- The program will be in radian mode and the RAD value = angle ×π/180.
- The SIN value obtained by S is calculated and stored in the register designated by D. The figure below offers the relation between radian and the result.



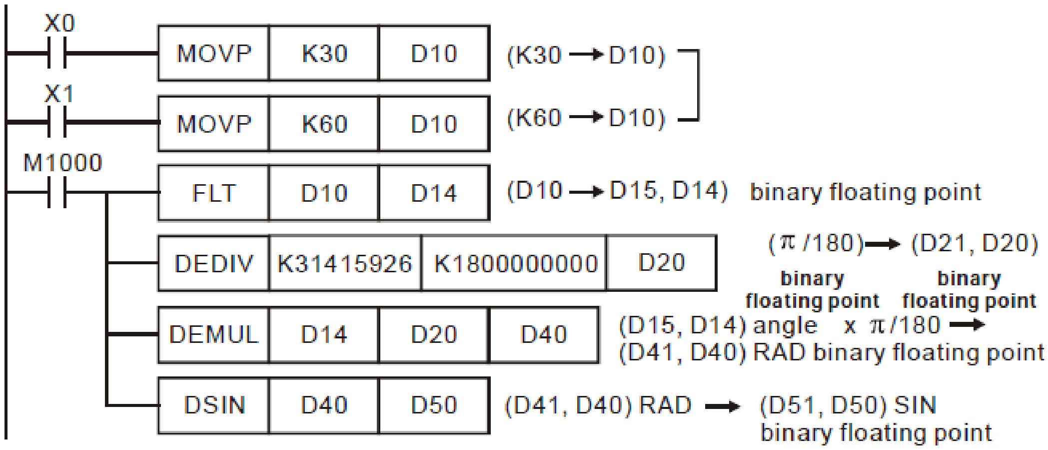
2、Program Example 1

- When X0 = On, use the RAD value of binary floating point (D1, D0) and obtain its SIN value. The binary floating point result will be stored in (D11, D10).



3、Program Example 2

- Input terminals X0 and X1 select the angle. The angles are converted into RAD value for calculating the SIN value.

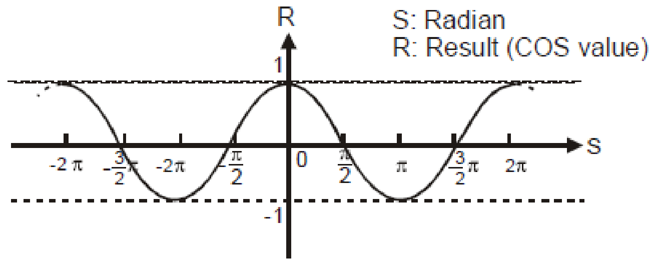

X0
MOVP K30 D10
(K30 → D10)
X1
MOVP K60 D10
(K60 → D10)
M1000
FLT D10 D14
(D10 → D15, D14) binary floating point
DEDIV K31415926 K1800000000 D20
(π /180) → (D21, D20)
binary floating point
binary floating point
DEMUL D14 D20 D40
(D15, D14) angle x π/180 →
(D41, D40) RAD binary floating point
DSIN D40 D50
(D41, D40) RAD → (D51, D50) SIN
binary floating point

ZL 131		COS		S D	Cosine
	D				

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DCOS: 9 steps
S					*	*							*			32-bit
D													*			

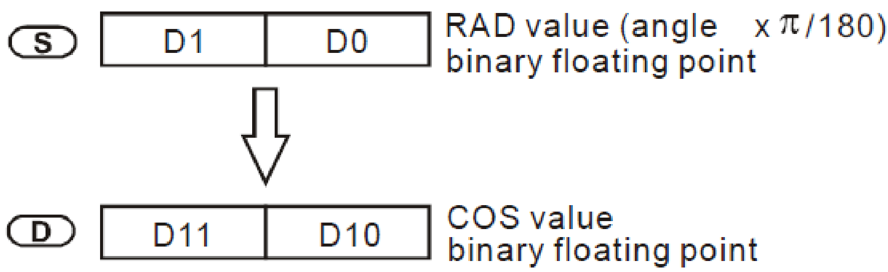
1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source value D: COS result
- the program will be in radian mode and the RAD value = angle ×π/180.



2、Program Example 1:

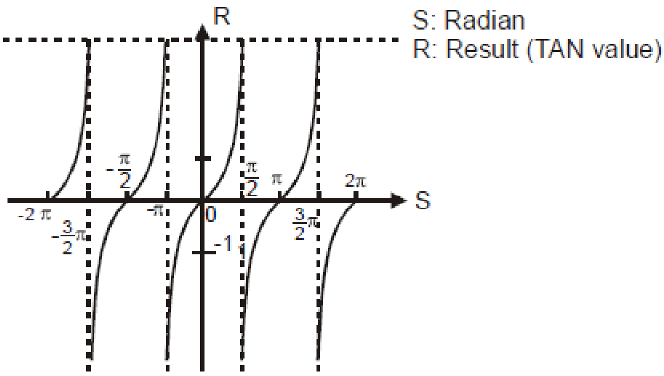
- When X0 = On, use the RAD value of binary floating point (D1, D0) and obtain its COS value. The binary floating point result will be stored in (D11, D10).



ZL 132		TAN		S D												Tangent
	D															
	Bit Devices				Word Devices											DTAN: 9 steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			
D													*			

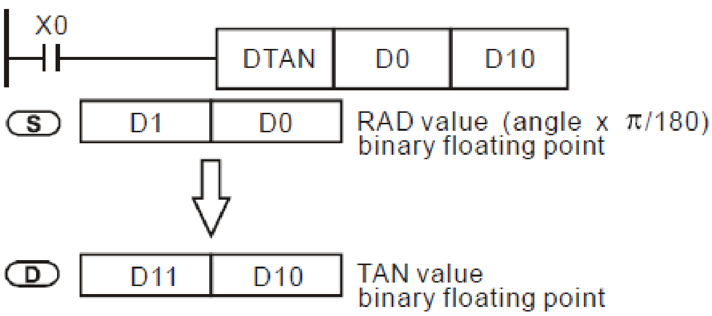
1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source value D: TAN result
- The program will be in radian mode and the RAD value = angle ×π/180.
- The TAN value obtained by S is calculated and stored in the register designated by D. The figure below offers the relation between radian and the result.



2、Program Example 1:

- When X0 = On, use the RAD value of binary floating point (D1, D0) and obtain its TAN value. The binary floating point result will be stored in (D11, D10).

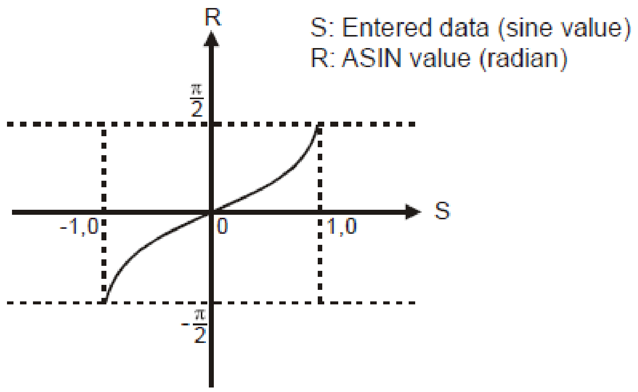


ZL 133		ASIN		S D	Arc Sine
	D				

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DASIN: 9 steps
S					*	*							*			32-bit
D													*			

1、 Explanations:

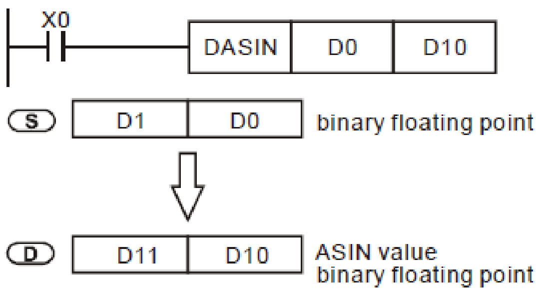
- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source value (binary floating point) D: ASIN result.
- ASIN value=sin – 1. The figure below offers the relation between the entered sin value and the result.



- The decimal floating point of the SIN value designated by S should be within -1.0 ~ +1.0.

2、Program Example:

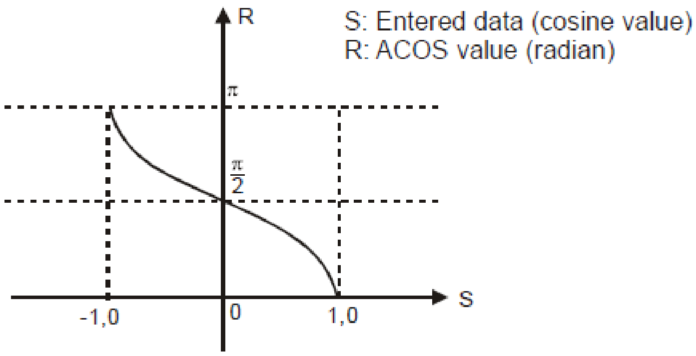
- When X0 = On, obtain the ASIN value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



ZL 134	D	ACOS		S D												Arc Cosine
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DACOS: 9 steps
S					*	*							*			32-bit
D													*			

1、Explanations:

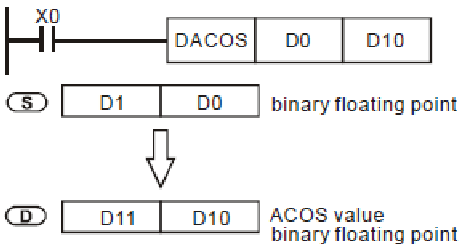
- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source value (binary floating point)   D: ACOS result
- ACOS value=$\cos^{-1}$. The figure below offers the relation between the entered cos value and the result.



- The decimal floating point of the COS value designated by S should be within -1.0 ~ +1.0.

2、Program Example:

- When X0 = On, obtain the ACOS value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).

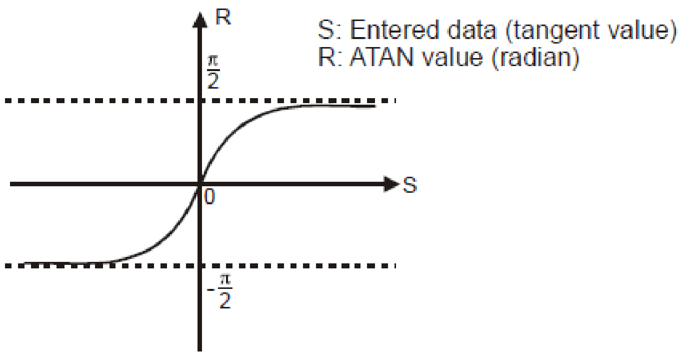


ZL 135		ATAN		S D	Arc Tangent
	D				

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DATAN: 9 steps
S					*	*							*			32-bit
D													*			

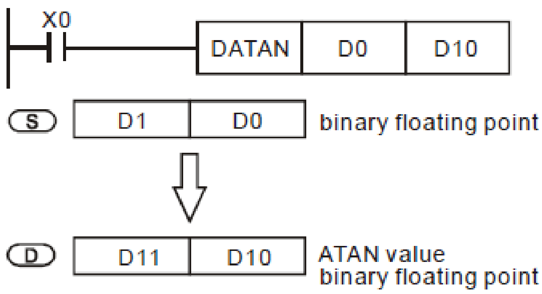
1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source value (binary floating point) D: ATAN value
- ATAN value=$\tan^{-1}$. The figure below offers the relation between the entered tan value and the result.



2、Program Example:

- When X0 = On, obtain the ATAN value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



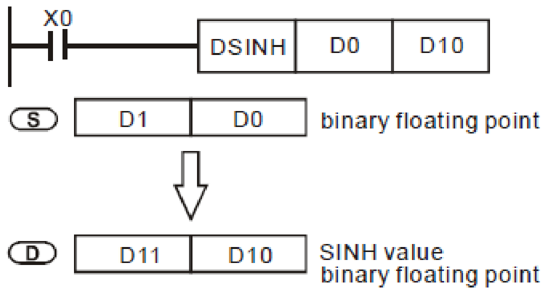
ZL 136	D	SINH		S D												Hyperbolic Sine
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DSINH: 9 steps
S					*	*							*			32-bit
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source value (binary floating point)   D: SINH value.
- SINH value=(e^{s}-e^{-s})/2. The result is stored in D.

2、Program Example:

- When X0 = On, obtain the SINH value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



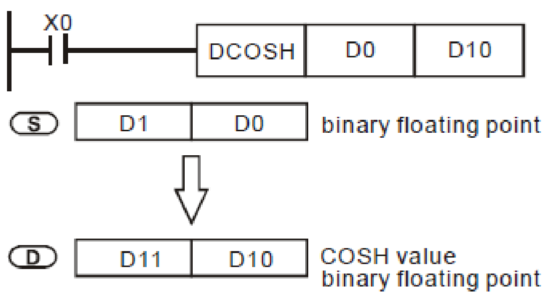
ZL 137	D	COSH		S D												Hyperbolic Cosine
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DCOSH: 9 steps
S					*	*							*			32-bit
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source value (binary floating point) D: COSH value
- COSH value=(e^{s}+e^{-s})/2. The result is stored in D.

2、Program Example:

- When X0 = On, obtain the COSH value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



ZL 138		TANH		S D	Hyperbolic Tangent
	D				

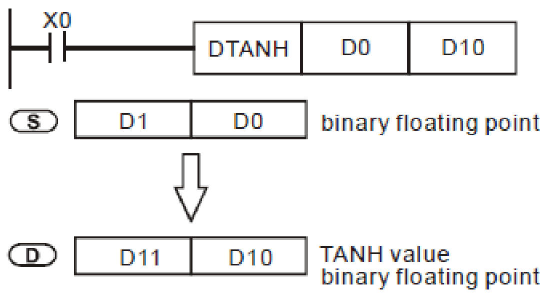
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DTANH: 9 steps
S					*	*							*			32-bit
D													*			

1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: Source value (binary floating point)　　D: TANH result
- TANH value=$(e^{s}-e^{-s})/(e^{s}+e^{-s})$. The result is stored in D.

2、Program Example:

- When X0 = On, obtain the TANH value of binary floating point (D1, D0) and store the binary floating point result in (D11, D10).



8.5 (ZL 140-149) Special function instructions

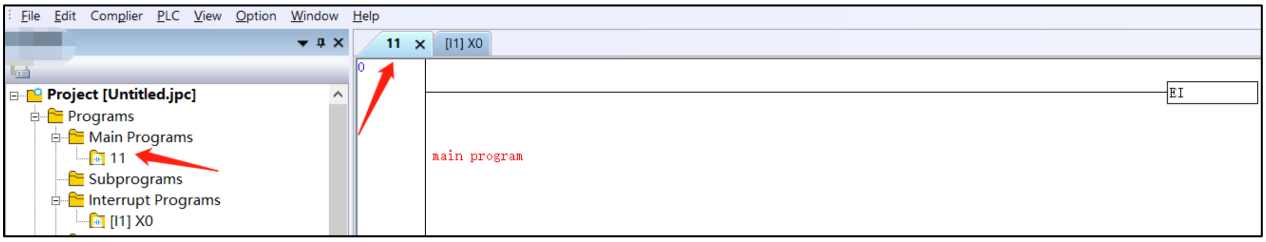
ZL 143	DELAY				S											Delay Instruction
	Bit Devices				Word Devices											DELAY: 3 steps 16-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			

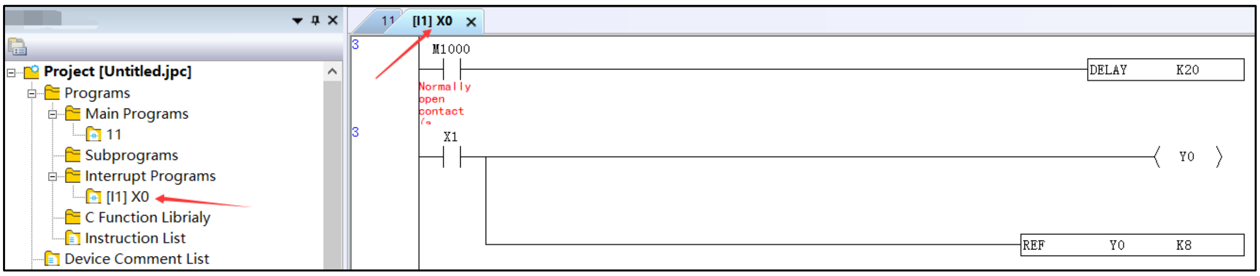
1、Explanations:

- To execute the pulse type, add the NP rising edge "↑" command before the command.
- S: delay time (unit: 100ms).
- Range of S: K1 ~ K1,000. See the specifications of each model for their range of use.
- After DELAY instruction is executed, the program after DELAY in every scan period will execute delay outputs according to the delay time designated by the user.

2、Program Example:

- If X0 is turned from Off to On, the external interruption will be generated. DELAY in the interrupt subroutine will be execute for 2 ms before the next step (X1 = On and Y0 = On) is executed.





ZL 144	GPWM	S1 S2 D	General PWM Output

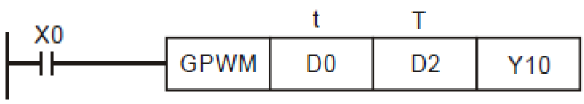
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GPWM: 7 steps 16-bit
S1													*			
S2													*			
D		*	*	*												

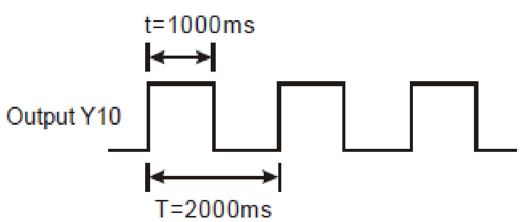
1、Explanations:

- S1: Width of output pulse S2: Pulse output cycle D: Pulse output device
- S2 occupies 3 consecutive devices.
- S1 ⩽ S2. See the specifications of each model for their range of use.
- Range of S1: t = 0 ~ 32,767ms.
- Range of S2: t = 1 ~ 32,767ms.
- S2 +1 and S2 +2 are parameters for the system. Do not occupy them.
- Pulse output devices D: Y, M, S.
- When being executed, GPWM instruction designates S1 and S2 and that pulses output will be from device D.
- When S1 ⩽ 0, there will be no pulse output. When S1 ⩾ S2, the pulse output device will keep being On.
- S1 and S2 can be modified when GPWM instruction is being executed.

2、Program Example:

- When X0 = On, D0 = K1,000, D2 = K2,000, and Y10 will output the pulse illustrated below. When X0 = Off, Y10 output will be Off.





ZL 147		SWAP		S	Byte Swap
	D				

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S								*	*	*	*	*	*	*	*

SWAP: 3 steps

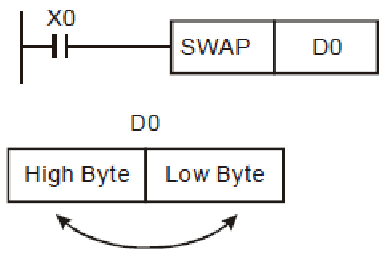
DSWAP: 5 steps

1、Explanations:

- S: Device for swapping 8 high/low byte.
- If D is used in device F, only 16-bit instruction is applicable.
- As 16-bit instruction: the contents in the 8 high bytes and 8 low bytes are swapped.
- As 32-bit instruction: the 8 high bytes and 8 low bytes in the two registers swap with each other respectively.
- This instruction adopts pulse execution instructions (SWAPP, DSWAPP).

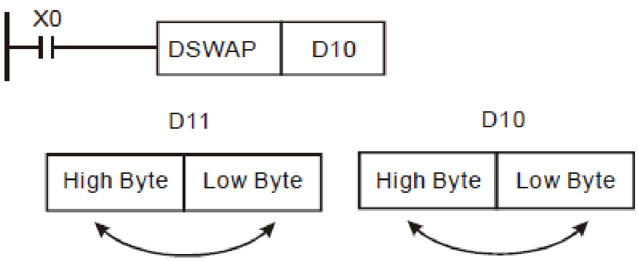
2、Program Example 1:

- When X0 = On, the high 8 bytes and low 8 bytes in D0 will swap with each other.



3、Program Example 2:

- When X0 = On, the high 8 bytes and low 8 bytes in D11 will swap with each other and the high 8 bytes and low 8 bytes in D10 will swap with each other.



9 Application instructions ZL150~ZL199

9.1 (ZL 150-154) Special function instructions

ZL 151	PWD					S D										Detection of Input Pulse Width
	Bit Devices				Word Devices											PWD: 5 steps 16-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															
n													*			

1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instructio n.
- S: Source device D: Destination device for storing the detected result
- Range of S: X10 ~ X15
- D must be in the range of D0 to D999., it occupies two consecutive devices.
- PWD instruction is for detecting the interval between the input signals; the valid frequency range is 1 ~1kHz. If M1169 = Off, the instruction will continuously detect the intervals between the rising edges of the input signals and the falling edges of the input signals (time unit: 100us). If M1169 = On, the instruction will continuously detect the intervals between rising edges of the input signals (time unit: 1us). It cannot designate the same X10 ~ X17 as DCNT and ZRN instructions.
- D occupies two consecutive devices. The longest detection time is 21,474.83647 seconds, about 357.9139 minutes or 5.9652 hours.
- There is no limitation on the times of using this instruction. However, only one instruction can be executed at a time.

2、Program Example:

- When X0 = On, record the time span of X10 = On and store it in D1 and D0.


X0
PWD
X10
D0

ZL 154	RAND	P	S1 S2 D	Random Number

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

RAND, RANDP: 7 steps 16-bit

DRAND: 13 steps

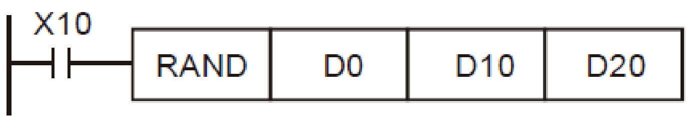
1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instructio n.
- S1: Lower bound of the random number S2: Upper bound of the random number

  D: The random number produced
- S1≦S2; K0 ≦ S1 , S2 ≦K32,767 If the user enters S1 > S2, the PLC determines that the operation is wrong and the instruction is not executed.
- Entering S1 > S2 will result in operation error. The instruction will not be executed at this time, M1067, M1068 = On and D1067 records the error code 0E1A (hex).

2、Program Example:

- When X10 = On, RAND will produce the random number between the lower bound D0 and upper bound D10 and store the result in D20.



9.2 (ZL 155-159) Positioning control

ZL 156		ZRN			S1 S2 S3 D											Zero Return
	D															
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*	*	ZRN: 9 steps 16-bit
S2					*	*	*	*	*	*	*	*	*	*	*	DZRN: 17 steps 32-bit
S3	*	*	*	*												
D		*														

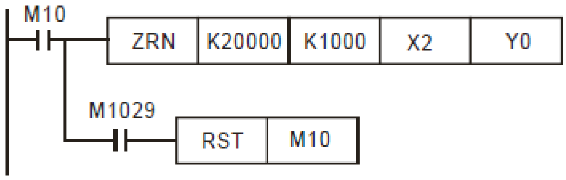
1、Explanations:

- S1: Zero return speed   S2: Creep speed
  S3: Near p oint signal (DOG)   D: Pulse output device (please use transistor output module)
- When S1 and S2 are used in device F, only 16-bit instruction is applicable.
- S1 specifies the speed at the beginning of home return. The 16-bit command can be specified in the range of 10~32,767Hz, and the 32-bit command can be specified in the range of 10~200,000Hz. When the specified speed is less than 10Hz, 10Hz is regarded as the homing speed; when the specified speed is greater than 200kHz, 200kHz is the homing speed.
- S2 specifies the inching speed, after the DOG signal On, specifies the speed of the low-speed part, the specified range is 10~32,767Hz.
- S3 Designated DOG signal input (A contact input).
- D pulse output device.
- When performing ZL 158 DRVI relative positioning or ZL 159 DRVA absolute addressing, the PLC has automatically generated forward/reverse pulses to increase and decrease the current value registers as shown in the table below. Therefore, the mechanical position can be grasped at any time, but since the data will disappear when the power of the PLC is turned off, the home position data of the mechanical action must be input when the home position return is performed for the first time.

Group No	PUL	DIR	current number of output pulses（32-bit integer)	Pulse complete	Pulse sending	Emergency stop	Start frequencyK10-K32767 Default K200	Accel/Decel timeK10-K10000 defaultK100
CH0（Y0,Y1)	Y0	Y1	D1648	M1029	M1344	M1308	D1340	D1343
CH1（Y2,Y3)	Y2	Y3	D1664	M1030	M1345	M1309	D1352	D1353
CH2（Y4,Y5)	Y4	Y5	D1680	M1036	M1346	M1310	D1379	D1381
CH3（Y6,Y7)	Y6	Y7	D1696	M1037	M1347	M1311	D1380	D1382
CH4（Y10,Y11)	Y10	Y11	D1712	M1102	M1348	M1312	D1400	D1383
CH5（Y12,Y13)	Y12	Y13	D1728	M1103	M1349	M1313	D1401	D1384
CH6（Y14,Y15)	Y14	Y15	D1744	M1104	M1350	M1314	D1402	D1385
CH7（Y16,Y17)	Y16	Y17	D1760	M1105	M1351	M1315	D1403	D1386
CH8（Y20,Y21)	Y20	Y21	D1776	M1106	M1352		D1404	D1387
CH9（Y22,Y23)	Y22	Y23	D1792	M1107	M1353		D1405	D1388
CH10（Y24,Y25)	Y24	Y25	D1808	M1108	M1354		D1406	D1389
CH11（Y26,Y27)	Y26	Y27	D1824	M1109	M1355		D1407	D1390
CH12（Y30,Y31)	Y30	Y31	D1840	M1110	M1356		D1408	D1391
CH13（Y32,Y33)	Y32	Y33	D1856	M1111	M1357		D1409	D1392
CH14（Y34,Y35)	Y34	Y35	D1872	M1112	M1358		D1410	D1393
CH15（Y36,Y37)	Y36	Y37	D1888	M1113	M1359		D1411	D1394
CH16（Y40,Y41)	Y40	Y41	D1904	M1114	M1360		D1412	D1395
CH17（Y42,Y43)	Y42	Y43	D1920	M1115	M1361		D1413	D1396
CH18（Y44,Y45)	Y44	Y45	D1472	M1116	M1362		D1414	D1397
CH19（Y46,Y47)	Y46	Y47	D1488	M1117	M1363		D1415	D1398
CH20（Y50,Y51)	Y50	Y51	D1504	M1118	M1364		D1416	D1399
CH21（Y52,Y53)	Y52	Y53	D1520	M1119	M1365		D1417	D1420
CH22（Y54,Y55)	Y54	Y55	D1536	M1205	M1366		D1418	D1421
CH23（Y56,Y57)	Y56	Y57	D1552	M1206	M1367		D1419	D1422
Remarks			D1648: Low word of the current number of output pulses from CH0. D1649: High word of the current number of output pulses from CH0.	After CH0-CH23 pulse output is completed, the corresponding flag bit is ON	Only when the pulse is being sent, the flag bit corresponding to CH0-CH23 is ON	Off->On : The high-speed pulse output pauses immediately. On->Off: Continuing to output the pulses which have not been output		

2、Program Example:

- When M10=On, start the home return action from Y0 output pulse at 20kHz frequency. When it touches the DOG signal X2=On, it will run in the opposite direction at 1kHz frequency of inching speed, output pulse from Y0 to X2 =Off and stop.



- The zero return operation:

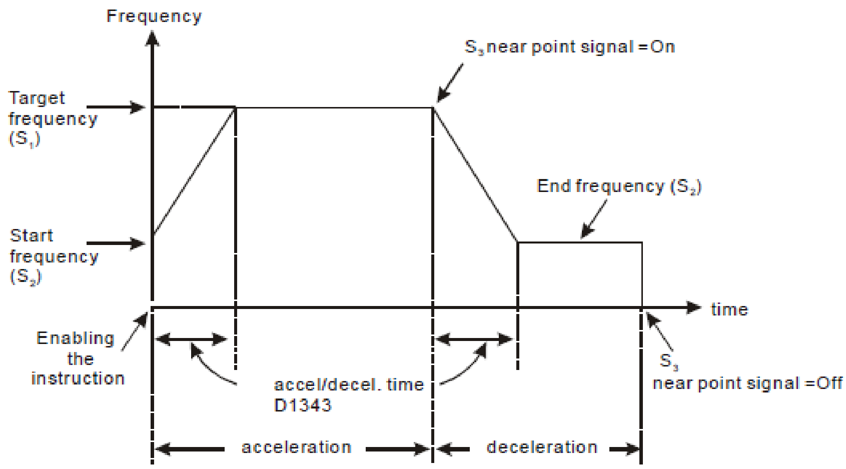
1) When ZRN instruction is executed, set the frequency of the first acceleration segment as the start frequency. The acceleration time of special D is used for reference.S1 will start to move when the acceleration reaches the zero return speed.

2) When the DOG signal goes from Off to On, the zero return speed will decelerate to S2 in the acceleration/deceleration time.

3) When the DOG signal goes from On to Off, the pulse output will immediately stop, 0 will be written in the present value.

4) When the pulse output is completed, the completion flag is ON and the in operation flag is OFF.

5) The ZRN (DZRN) instruction cannot search for the position of the near-point signal (DOG), and the homing operation can only be performed in one direction. The content of the current value register of the pulse amount corresponding to each channel in the home return will change towards the decreasing direction.



6) When the conditions for the start of the return to origin command are met, CH0 (CH1) will read the value set by D1343 (D1353) as the acceleration and deceleration time. After accelerating to the origin

return speed, wait for the DOG origin signal to enter and then decelerate from the origin return speed to inches Moving speed until the DOG origin signal is OFF and immediately stop outputting pulses.

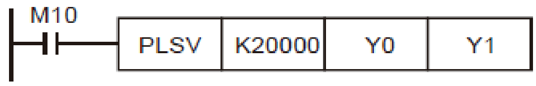
ZL 157	D	PLSV				S D1 D2										Adjustable Speed Pulse Output
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PLSV: 7 steps 16-bit DPLSV: 13 steps 32-bit
S1					*	*	*	*	*	*	*	*	*	*	*	
D1		*														
D2		*	*	*												

1、Explanations:

- S: Pulse output frequency D1: Pulse output device (please use transistor output module)
  D2: Output device for the signal of rotation direction
- See remarks for the setting range of S, D1 and D2.
- 3. S is the designated pulse output frequency. The 16-bit instruction can designate its range 0 ~ +32,767Hz, 0 ~- 32,768Hz. the ranges designated by 32-bit instruction are 0 ~ +200,000Hz and 0 ~ -200,000Hz. "+/-" signs indicate forward/backward directions. During the pulse output, the frequency can be changed, but not the frequencies of different directions.
- D1 is the pulse output device:
  Y0.Y2.Y4.Y6.Y10.Y12.Y14.Y16.Y18.Y20.Y22.Y24.Y26.Y30.Y32.Y34.Y36.Y40.Y42.Y44.Y46.
- The operation of D2 corresponds to the "+" or "- " of S. When S is "+", D2 will be On; when S is "- ", D2 will be Off.
- Anyway, when M1207 is OFF, the PLSV instruction does not set acceleration or deceleration, so it doesn't perform acceleration and deceleration at the beginning and stop.
- Anyway, when M1207 is ON, PLSV instruction takes acceleration and deceleration Settings, so the acceleration and deceleration actions start and stop are performed. The acceleration and deceleration time is set according to JC156 ZRN instruction.
- Anyway, when M1207 is OFF, PLSV instructions execute pulse output, and stop directly without decelerating if the driving condition changes to OFF.
- Virtual gateway When M1207 is ON, the PLSV instruction executes pulse output, decelerating and stopping if the driving condition changes to Off.

2、Program Example:

- When M10 = On, Y0 will output pulses at 20kHz. Y1 = On indicates forward pulses.



Group No	PUL	DIR	current number of output pulses（32-bit integer)	Pulse complete	Pulse sending	Accel/Decel timeK10-K10000 defaultK100
CH0（Y0,Y1)	Y0	Y1	D1648	M1029	M1344	D1343
CH1（Y2,Y3)	Y2	Y3	D1664	M1030	M1345	D1353
CH2（Y4,Y5)	Y4	Y5	D1680	M1036	M1346	D1381
CH3（Y6,Y7)	Y6	Y7	D1696	M1037	M1347	D1382

ZL 158		DRVI		S1 S2 D1 D2											Drive to Increment
	D														
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*	*	*	DRVI: 9 steps 16-bit
S2					*	*	*	*	*	*	*	*	*	*	*	DDRVI: 17 steps 32-bit
D1		*														
D2		*	*	*												

1、Explanations:

- S1: Number of output pulses (relative designation)

  S2: Pulse output frequency

  D1: Pulse output device (please use transistor output module)

  D2: Output device for the signal of rotation direction
- See remarks for the setting range of S1, S2, D1 and D2.
- S1 is the number of output pulses (relative designation). The 16-bit instruction can designate the range -32,768 ~ +32,767. The range designated by 32-bit instruction is -2,147,483,648 ~+2,147,483,647. If the value in S1 is 0, that means no output and no action.
- S2 is the designated pulse output frequency. The 16-bit instruction can designate its range 10 ~ 32,767Hz. The range designated by 32-bit instruction is 10 ~ 200,000Hz.
- The operation of D2 corresponds to the "+" or "- " of S1. When S1 is "-", D2 will be Off; when S1 is "+ ", D2 will be On. D2 will not be Off immediately after the pulse output is over; it will be Off only when the drive contact of the instruction turns Off.
- Specify the number of pulse output S1 will become the current value register of CH0 (Y0, Y1) pulse (D1648 high bit, D1649 low bit) 32-bit data, CH1 (Y2, Y3) pulse current value register (D1664 high bit, D1665 low bit) 32 bit Data, and so on. In the reverse direction, the content of the current value register will decrease.
- When DRVI instruction is executing pulse output, you cannot change the content of all operands. The changes will be valid next time when DRVI instruction is enabled.
- When the driving condition of the DRVI command becomes Off, even if the CH0 (CH1) pulse sending indicator M1344 (M1345) is On, the DRVI command cannot be driven again.
- DRVI and DDRVI commands output at 200kHz when the absolute value of the input frequency>200kHz,

and output at 10Hz when the absolute value of the input frequency<10Hz.

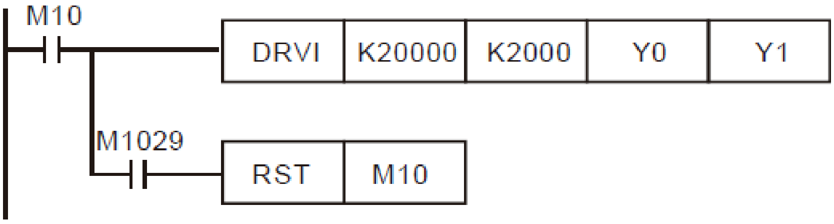
- D1343 (D1353) is the acceleration and deceleration time setting of CH0 (CH1) for the first stage of acceleration and the last stage of deceleration. The acceleration and deceleration time is 1~10,000 ms. If it is higher than 10,000ms, the factory default value is 100ms.
- D1340 (D1352) is the CH0 (CH1) start/stop frequency setting. If the pulse output frequency specified by S2 is less than or equal to the start/stop frequency, the start/stop frequency will be used as the pulse output frequency.
- Please refer to the table for host pulse output channels:

Group No	PUL	DIR	current number of output pulses (32-bit integer)	Pulse complete	Pulse sending	Emergency stop	Start frequencyK10-K32767 Default K200	Accel/Decel timeK10-K10000 defaultK100
CH0 (Y0,Y1)	Y0	Y1	D1648	M1029	M1344	M1308	D1340	D1343
CH1 (Y2,Y3)	Y2	Y3	D1664	M1030	M1345	M1309	D1352	D1353
CH2 (Y4,Y5)	Y4	Y5	D1680	M1036	M1346	M1310	D1379	D1381
CH3 (Y6,Y7)	Y6	Y7	D1696	M1037	M1347	M1311	D1380	D1382
CH4 (Y10,Y11)	Y10	Y11	D1712	M1102	M1348	M1312	D1400	D1383
CH5 (Y12,Y13)	Y12	Y13	D1728	M1103	M1349	M1313	D1401	D1384
CH6 (Y14,Y15)	Y14	Y15	D1744	M1104	M1350	M1314	D1402	D1385
CH7 (Y16,Y17)	Y16	Y17	D1760	M1105	M1351	M1315	D1403	D1386
CH8 (Y20,Y21)	Y20	Y21	D1776	M1106	M1352		D1404	D1387
CH9 (Y22,Y23)	Y22	Y23	D1792	M1107	M1353		D1405	D1388
CH10 (Y24,Y25)	Y24	Y25	D1808	M1108	M1354		D1406	D1389
CH11 (Y26,Y27)	Y26	Y27	D1824	M1109	M1355		D1407	D1390
CH12 (Y30,Y31)	Y30	Y31	D1840	M1110	M1356		D1408	D1391
CH13 (Y32,Y33)	Y32	Y33	D1856	M1111	M1357		D1409	D1392
CH14 (Y34,Y35)	Y34	Y35	D1872	M1112	M1358		D1410	D1393
CH15 (Y36,Y37)	Y36	Y37	D1888	M1113	M1359		D1411	D1394
CH16 (Y40,Y41)	Y40	Y41	D1904	M1114	M1360		D1412	D1395
CH17 (Y42,Y43)	Y42	Y43	D1920	M1115	M1361		D1413	D1396
CH18 (Y44,Y45)	Y44	Y45	D1472	M1116	M1362		D1414	D1397
CH19 (Y46,Y47)	Y46	Y47	D1488	M1117	M1363		D1415	D1398
CH20 (Y50,Y51)	Y50	Y51	D1504	M1118	M1364		D1416	D1399
CH21 (Y52,Y53)	Y52	Y53	D1520	M1119	M1365		D1417	D1420
CH22 (Y54,Y55)	Y54	Y55	D1536	M1205	M1366		D1418	D1421
CH23 (Y56,Y57)	Y56	Y57	D1552	M1206	M1367		D1419	D1422

Remarks			D1648: Low word of the current number of output pulses from CH0. D1649: High word of the current number of output pulses from CH0.	After CH0-CH23 pulse output is completed, the corresponding flag bit is ON	Only when the pulse is being sent, the flag bit corresponding to CH0-CH23 is ON	Off->On : The high-speed pulse output pauses immediately. On->Off: Continuing to output the pulses which have not been output		

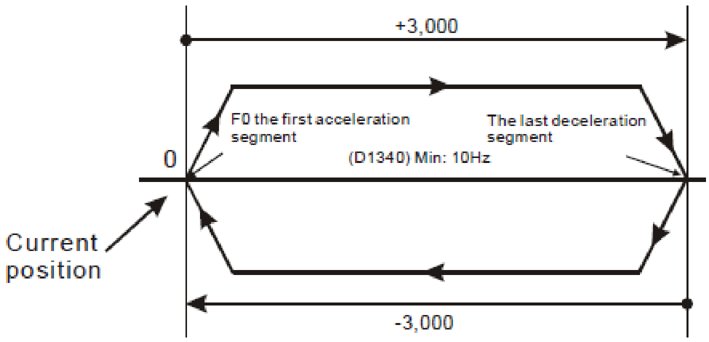
2、Program Example:

- When M10= On, Y0 will output 20,000 pulses (relative designation) at 2kHz. Y1 = On indicates the pulses are executed in forward direction.

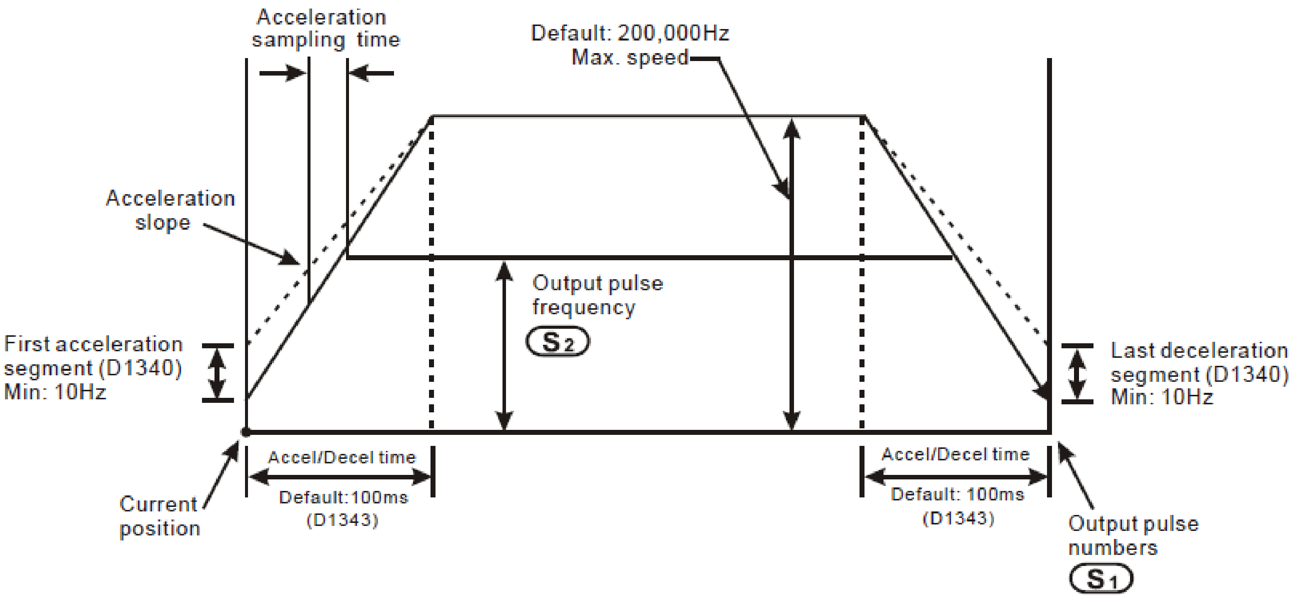


Remarks:

1) Relative position control: Designating the traveling distance starting from the current position by “+/ -” signs; also known as a relative driving method.



2) Settings of relative positioning and the acceleration/deceleration speed:


Acceleration sampling time
Default: 200,000Hz
Max. speed
Acceleration slope
Output pulse frequency
S2
First acceleration segment (D1340) Min: 10Hz
Last deceleration segment (D1340) Min: 10Hz
Current position
Accel/Decel time
Default:100ms (D1343)
Accel/Decel time
Default: 100ms (D1343)
Output pulse numbers
S1

ZL 159	D	DRVA		S1 S2 D1 D2												Drive to Absolute
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*	*	DRVA: 9 steps
S2					*	*	*	*	*	*	*	*	*	*	*	DDRVA: 17 steps
D1		*														
D2		*	*	*												

1、Explanations:

- S1: Number of output pulses (absolute designation)   S2: Pulse output frequency

  D1: Pulse output device (please use transistor output module)   D2: Output device for the signal of rotation direction

- S1 is the number of output pulses (absolute designation). The 16-bit instruction can designate the range -32,768 ~ +32,767. The range designated by 32-bit instruction is -2,147,483,648 ~ +2,147,483,647. If the absolute position and the current position in S1 are the same, which means the relative output pulse is 0. Then to execute this instruciton will NOT output any pulse but the special M flag will be ON, indicating the output is complete.
- S2 is the designated pulse output frequency. The 16-bit instruction can designate its range 10 ~ 32,767Hz. The range designated by 32-bit instruction is 10 ~ 200,000Hz.
- D2 The output device of the rotation direction signal. When S1 is greater than the current relative position, D2: Off. When S1 is less than the current relative position, D2: On, D2 will not be Off immediately after the pulse output ends, you must wait for the command to execute when the contact switch is Off D2: Off.
- Specify the number of pulse output S1 will become the current value register of CH0 (Y0, Y1) pulse (D1648 high bit, D1649 low bit) 32-bit data, CH1 (Y2, Y3) pulse current value register (D1664 high bit, D1665 low bit) 32 bit Data, and so on. In the reverse direction, the content of the current value register will decrease.
- When DRVA instruction is executing pulse output, you cannot change the content of all operands. The changes will be valid next time when DRVA instruction is enabled.
- When the driving condition of the DRVA command becomes Off, even if the CH0 (CH1) pulse sending indicator M1344 (M1345) is On, the DRVA command cannot be driven again.
- DRVI and DDRVI commands output at 200kHz when the absolute value of the input frequency>200kHz,

and output at 10Hz when the absolute value of the input frequency<10Hz.

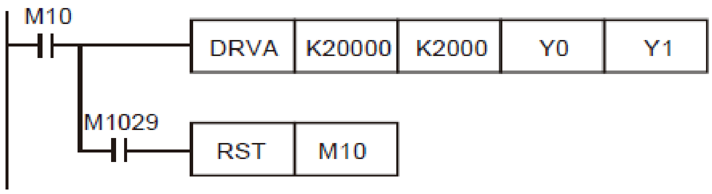
- D1343 (D1353) is the acceleration and deceleration time setting of CH0 (CH1) for the first stage of acceleration and the last stage of deceleration. The acceleration and deceleration time is 1~10,000 ms. If it is higher than 10,000ms, the factory default value is 100ms.
- D1340 (D1352) is the CH0 (CH1) start/stop frequency setting. If the pulse output frequency specified by S2 is less than or equal to the start/stop frequency, the start/stop frequency will be used as the pulse output frequency.
- Please refer to the table for host pulse output channels:

Group No	PUL	DIR	current number of output pulses（32-bit integer)	Pulse complete	Pulse sending	Emergency stop	Start frequencyK 10-K32767 Default K200	Accel/Decel timeK10-K10000 defaultK100
CH0（Y0,Y1)	Y0	Y1	D1648	M1029	M1344	M1308	D1340	D1343
CH1（Y2,Y3)	Y2	Y3	D1664	M1030	M1345	M1309	D1352	D1353
CH2（Y4,Y5)	Y4	Y5	D1680	M1036	M1346	M1310	D1379	D1381
CH3（Y6,Y7)	Y6	Y7	D1696	M1037	M1347	M1311	D1380	D1382
CH4（Y10,Y11)	Y10	Y11	D1712	M1102	M1348	M1312	D1400	D1383
CH5（Y12,Y13)	Y12	Y13	D1728	M1103	M1349	M1313	D1401	D1384
CH6（Y14,Y15)	Y14	Y15	D1744	M1104	M1350	M1314	D1402	D1385
CH7（Y16,Y17)	Y16	Y17	D1760	M1105	M1351	M1315	D1403	D1386
CH8（Y20,Y21)	Y20	Y21	D1776	M1106	M1352		D1404	D1387
CH9（Y22,Y23)	Y22	Y23	D1792	M1107	M1353		D1405	D1388
CH10（Y24,Y25)	Y24	Y25	D1808	M1108	M1354		D1406	D1389
CH11（Y26,Y27)	Y26	Y27	D1824	M1109	M1355		D1407	D1390
CH12（Y30,Y31)	Y30	Y31	D1840	M1110	M1356		D1408	D1391
CH13（Y32,Y33)	Y32	Y33	D1856	M1111	M1357		D1409	D1392
CH14（Y34,Y35)	Y34	Y35	D1872	M1112	M1358		D1410	D1393
CH15（Y36,Y37)	Y36	Y37	D1888	M1113	M1359		D1411	D1394
CH16（Y40,Y41)	Y40	Y41	D1904	M1114	M1360		D1412	D1395
CH17（Y42,Y43)	Y42	Y43	D1920	M1115	M1361		D1413	D1396
CH18（Y44,Y45)	Y44	Y45	D1472	M1116	M1362		D1414	D1397
CH19（Y46,Y47)	Y46	Y47	D1488	M1117	M1363		D1415	D1398
CH20（Y50,Y51)	Y50	Y51	D1504	M1118	M1364		D1416	D1399
CH21（Y52,Y53)	Y52	Y53	D1520	M1119	M1365		D1417	D1420
CH22（Y54,Y55)	Y54	Y55	D1536	M1205	M1366		D1418	D1421
CH23（Y56,Y57)	Y56	Y57	D1552	M1206	M1367		D1419	D1422

Remarks			D1648: Low word of the current number of output pulses from CH0. D1649: High word of the current number of output pulses from CH0.	After CH0-CH23 pulse output is completed, the corresponding flag bit is ON	Only when the pulse is being sent, the flag bit corresponding to CH0-CH23 is ON	Off->On : The high-speed pulse output pauses immediately. On->Off: Continuing to output the pulses which have not been output		

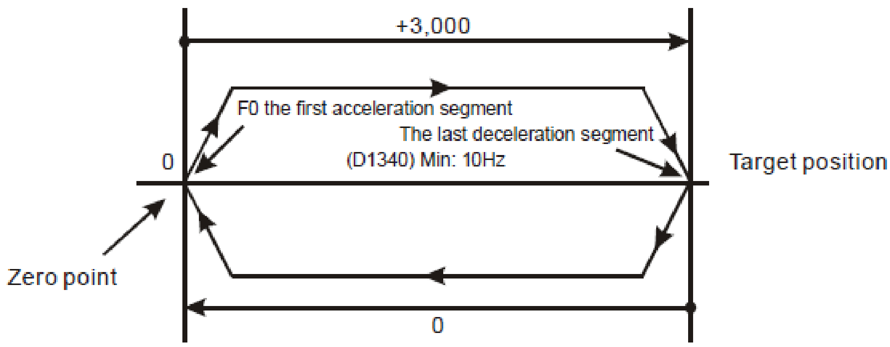
2、Program Example:

- When M10= On, Y0 will output 20,000 pulses (absolute designation) at 2kHz. Y1 = On indicates the pulses are executed in forward direction.

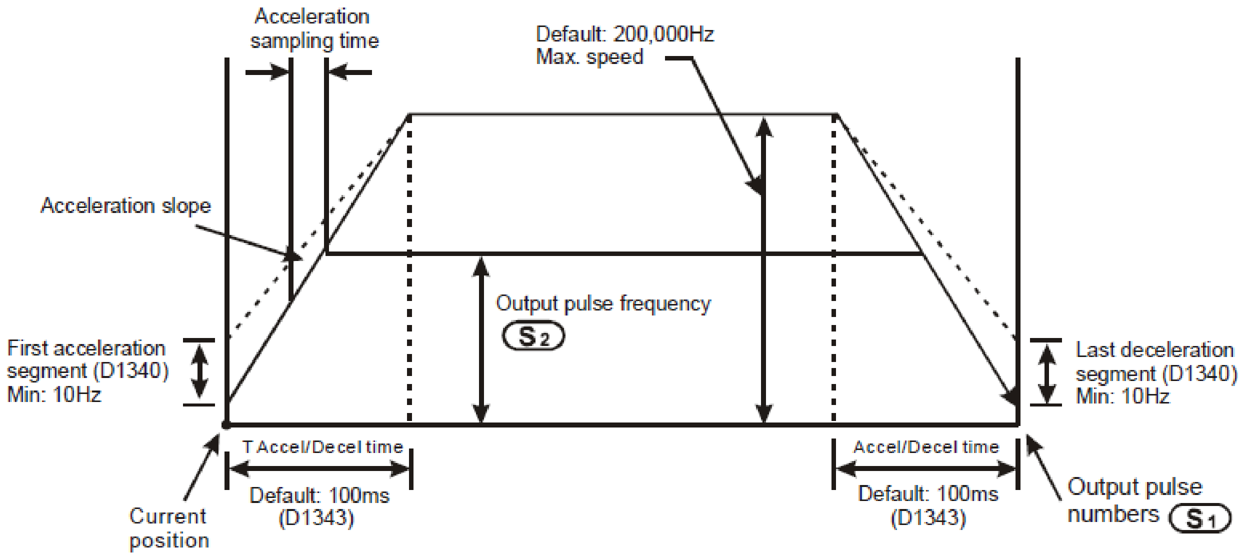


Remarks:

1) Absolute position control: Designating the traveling distance starting from the zero point (0); also known as a absolute driving method.



2) Settings of absolute positioning and the acceleration/deceleration speed:


Acceleration
sampling time
Default: 200,000Hz
Max. speed
Acceleration slope
Output pulse frequency
S2
First acceleration
segment (D1340)
Min: 10Hz
Last deceleration
segment (D1340)
Min: 10Hz
T Accel/Decel time
Accel/Decel time
Default: 100ms
(D1343)
Default: 100ms
(D1343)
Current
position
Output pulse
numbers S1

9.3 (ZL 160-169) Perpetual calendar

ZL 160	D	TCMP		S1 S2 S3 D1 D2											Time Compare
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
S3					*	*	*	*	*	*	*	*	*	*	*
D1											*	*	*		
D2		*	*	*											

TCMP, TCMPP: 11 steps

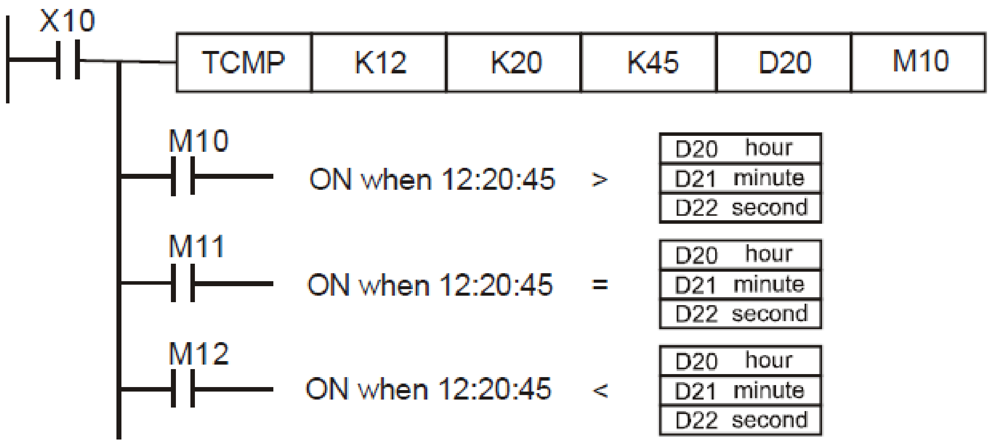
16-bit

1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- Range of S1: K0 ~ K23; range of S2 and S3: K0 ~ K59.
- S will occupy 3 consecutive devices; D will occupy 3 consecutive points.
- S1, S2 and S3 are compared with the present values of "hour", "minute" and "second" starting from S. The comparison result is stored in D.
- S is the "hour" of the current time (K0 ~ K23) in RTC; S + 1 is the "minute" (K0 ~ K59) and S + 2 is the "second" (K0 ~ K59).
- S is read by TRD instruction and the comparison is started by TCMP instruction. If S exceeds the range, the program will regard this as an operation error and the instruction will not be executed.

2、Program Example:

- When X10= On, the instruction will compare the current time in RTC (D20 ~ D22) with the set value 12:20:45 and display the result in M10 ~ M12. When X10 goes from On to Off, the instruction will not be executed, but the On/Off stauts prior to M10 ~ M12 will remain.
- Connect M10 ~ M12 in series or in parallel to obtain the result of ≧, ≦, and ≠.


X10
TCMP
K12
K20
K45
D20
M10
M10
ON when 12:20:45 >
D20 hour
D21 minute
D22 second
M11
ON when 12:20:45 =
D20 hour
D21 minute
D22 second
M12
ON when 12:20:45 <
D20 hour
D21 minute
D22 second

ZL 161		TZCP		S1 S2 S D2	Time Zone Compare
	D				

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1											*	*	*			TZCP, TZCPP: 9 steps
S2											*	*	*			16-bit
S											*	*	*			
D		*	*	*												

1、Explanations:

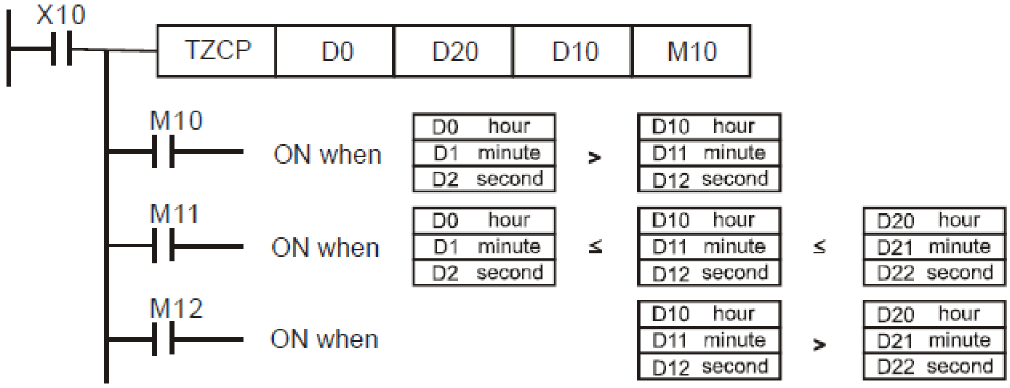
- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Lower bound of the time for comparison

  S2: Upper bound of the time for comparison

  S: Current time of RTC D: Comparison result
- S1, S2, and S will occupy 3 consecutive devices.
- The content in S1 must be less than the content in S2.
- D will occupy 3 consecutive points.
- S is compared with S1 and S2. The comparsion result is stored in D.
- S1 , S1 +1, S1 +2: The "hour", "minute" and "second" of the lower bound of the time for comparison.
- S2, S2 +1, S2 +2: The "hour", "minute" and "second" ond" of the upper bound of the time for comparison.
- S , S +1, S +2: The "hour", "minute" and "second" of the current time of RTC.
- D0 designated by S is read by TRD instruction and the comparison is started by TZCP instruction. If S1, S2, and S exceed their ranges, the program will regard this as an operation error and the instruction will not be executed.
- When S < S1 and S < S2, D will be On. When S > S1 and S > S2, D + 2 will be On. In other occasions, D + 1 will be On.

2、Program Example:

- When X10= On, TZCP instruction will be executed and one of M10 ~ M12 will be On. When X10 = Off, TZCP instruction will not be executed and the status of M10 ~ M12 prior to X10 = Off will remain unchanged.


X10
TZCP
D0
D20
D10
M10
M10
ON when
D0 hour
D1 minute
D2 second
>
D10 hour
D11 minute
D12 second
M11
ON when
D0 hour
D1 minute
D2 second
≤
D10 hour
D11 minute
D12 second
≤
D20 hour
D21 minute
D22 second
M12
ON when
D10 hour
D11 minute
D12 second
>
D20 hour
D21 minute
D22 second

ZL 162	D	TADD	S1 S2 D	Time Addition

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	TADD, TADDP: 7 steps 16-bit
S1											*	*	*			
S2											*	*	*			
D											*	*	*			

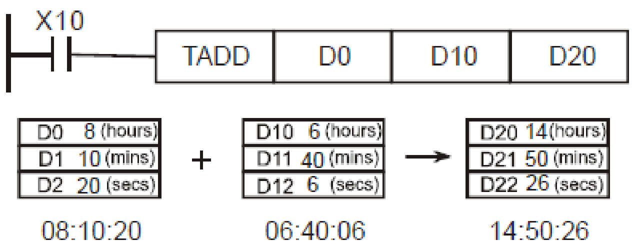
1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge " ↑ " instruction to the front of the instruction.

- S1, S2, and D will occupy 3 consecutive devices.
- S1 + S2 = D. The hour, minute, and second of the RTC designated in S1 plus the hour, minute, and second designated in S2. The result is stored in the hour, minute, and second of the register designated in D.
- If S1 and S2 exceed their ranges, the program will regard this as an operation error and the instruction will not be executed.
- If the sum is larger than 24 hours, the value in D will be the result of “sum minuses 24 hours” .

2、Program Example:

- When X10= On, TADD instruction will be executed and the hour, minute and second in RTC designated in D0 ~ D2 will plus the hour, minute and second in RTC designated in D10 ~ D12. The sum is stored in the hour, minute and second of the register designated in D20 ~ D22.



ZL 163	D	TSUB		S1 S2 D	Time Subtraction

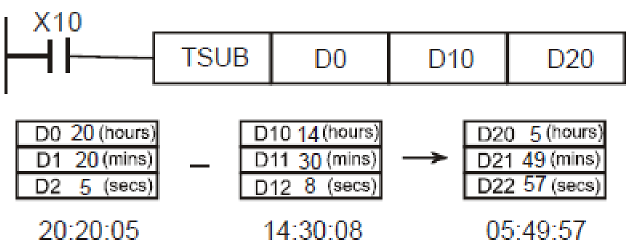
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1											*	*	*			Time Subtraction
S2											*	*	*			
D											*	*	*			

1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Time minuend S2: Time subtrahend D: Time remainder
- S1, S2, and D will occupy 3 consecutive devices.
- S1 − S2 = D. The hour, minute, and second of the RTC designated in S1 minus the hour, minute, and second designated in S2. The result is stored in the hour, minute, and second of the register designated in D.
- If S1 and S2 exceed their ranges, the program will regard this as an operation error and the instruction will not be executed.
- If the remainder is a negative value, the value in D will be the result of "the negative value pluses 24 hours" .

2、Program Example:

- When X10= On, TADD instruction will be executed and the hour, minute and second in RTC designated in D0，~ D2 will minus the hour, minute and second in RTC designated in D10 ~ D12. The remainder is stored in the hour, minute and second of the register designated in D20 ~ D22.



ZL 166		TRD					D									Calendar data read-out
	Bit Devices				Word Devices											A 16-bit instructions TRD continuous execution 32-bit instruction There is no
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D											*	*	*			
● Note that the D operand uses seven devices in a row Please refer to the functional specification table for the use range of each device																

1、Instructions

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- A device that stores a perpetual calendar when the time is read out.
- According to a clock, seven data sets -- year, week, month, day, hour, minute, second -- are stored in D1319 to D1313, according to the TRD instruction, which lets programmers read the time directly into a specified set of seven registers.
- Anyway, D1319 reads only the right two bits of THE AD year, according to the supplementary instructions for reading all four bits.

2、Sample application

- According to the system, when X0=On, the clock reads the time of the calendar into the specified REGISTERS D0~D6.
- Buy a ticket for D1318, using 1 for Monday, 2 for Tuesday, and so on, and 7 for Sunday.

Description:

1) Mark and special register of perpetual calendar clock:

number	The name says	Action function
M1016	Perpetual calendar AD year display	Off when D1319 shows AD 2 to the right On D1319 shows the year AD 2 digits to the right plus 2000
M1017	Plus or minus 30 seconds	Off→On for correction when triggered. (0~29 seconds return 0, 30~59 seconds, minute plus 1, second return 0).
M1076	The calendar is faulty	Set value ON when out of set range (this check will only be done when starting up)

number	The name says	Action function
D1313	second	0~59
D1314	points	0~59
D1315	when	0~23
D1316	day	1~31
D1317	month	1~12
D1318	week	1~7
D1319	years	0~99 (2nd from the right of AD)

- A method of correcting a perpetual calendar clock
- built-in perpetual calendar clock, its correction method is correction time special instruction, please refer to TWR instruction (ZL 167) for details.
- Lent The YEAR displays a 4-digit number:

1) The year normally displays only 2 digits (for example: 2003 displays only 03). If you want to display 4 digits, please type the following program at the beginning of the program.
2) The AD year display is changed from 2 digits to 4 digits, showing the year of THE right 2 digits plus 2000.
3) If you want to write the new setting time in the mode of 4-digit display in THE YEAR of AD, only 2-digit can be written, and the valid value of this 2-digit is "0~99", which reflects the year of AD is "2000 ~2099", the relationship between them is as follows. Example: 00= year 2000 03= year 2003 50= year 2050 99= year 2099.

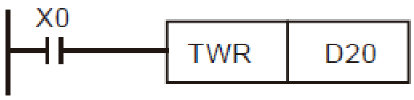
<table>
<tr><td>JC 167</td><td></td><td colspan="2" rowspan="2">TWR</td><td></td><td colspan="10" rowspan="2">S</td><td colspan="2" rowspan="2">Write calendar data</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td rowspan="2"></td><td colspan="4">Bit Devices</td><td colspan="11">Word Devices</td><td rowspan="4">A 16-bit instructions
TRD continuous execution
32-bit instruction
There is no</td></tr>
<tr><td>X</td><td>Y</td><td>M</td><td>S</td><td>K</td><td>H</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>T</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td>*</td><td>*</td><td></td><td></td></tr>
<tr><td colspan="16">● Note that the D operand uses seven devices in a row Please refer to the functional specification table for the use range of each device</td></tr>
</table>

1、Instructions

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- Anyway, S: The device that stores new values to be written to the calendar.
- To adjust a calendar clock, you use this command to write the correct current time into the built-in calendar clock, anyway.
- According to the scheme, when the command is executed, the new set time is written into the PLC's internal calendar clock immediately, so when running the command, pay attention to whether the new set time is written to the current time at the time of writing.
- Anyway, if the value of S content is out of the range, the operation is regarded as an error and the command is not executed.

2、Sample application

- Buy a way to write the correct current time into an implicit calendar clock when X0=On.

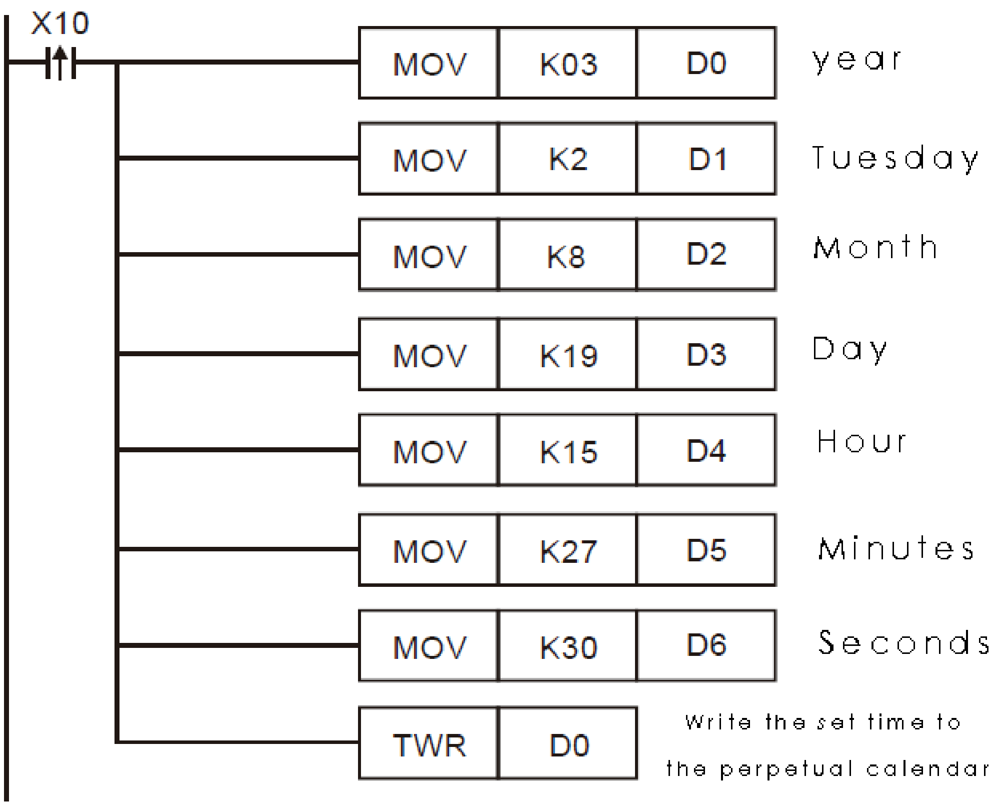


	General D	project	content		Te D	project	
New setting time	D20	years	00~99	→	D1319	years	Perpetual calendar clock
	D21	week	1~7	→	D1318	week	
	D22	month	1~12	→	D1317	month	
	D23	day	1~31	→	D1316	day	
	D24	when	0~23	→	D1315	when	
	D25	points	0~59	→	D1314	points	
	D26	second	0~59	→	D1313	second	

3、Example program 2

- Perpetual calendar current time setting, adjust the current time to 15:27:30 on Tuesday, August 19, 2003.
- The content of D0~D6 sets the time for the new perpetual calendar.
- X10=On can replace the current time of the perpetual calendar clock as the set value.

Every time X11 is On, the perpetual calendar clock will perform a correction action of ±30 seconds. The so-called correction is that when the second hand of the perpetual calendar clock is between 1 and 29, it will be automatically classified as "0" seconds and the minute hand will remain unchanged. is automatically reclassified to "0" seconds and the minute hand adds 1 minute.



ZL 168		MVM		S1 S2 D												Move the Designated Bit
	D															
	Bit Devices				Word Devices											MVM, MVMP: 7 steps 16-bit DMVM,DMVMP: 13 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1							*	*	*	*	*	*	*	*	*	
S2					*	*	*	*	*	*	*	*	*	*	*	
D							*	*	*	*	*	*	*	*	*	

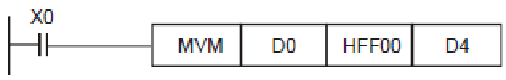
1、Explanations:

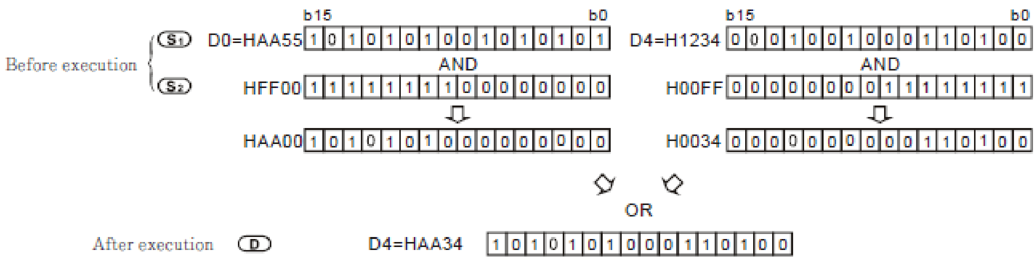
- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Source device 1 S2: Bits to be masked (OFF)

  D: Source device 2 / Operation results [D = (S1 & S2) | (D & ~S2)]
- The instruction conducts logical AND operation between S1 and S2 first, logical AND operation between D and ~S2 secondly, and combines the 1st and 2nd results in D by logical OR operation.
- Rule of Logical AND operation: 0 AND 1 = 0, 1 AND 0 = 0, 0 AND 0 = 0, 1 AND 1 = 1.
- Rule of Logical OR operation: 0 OR 1= 1, 1 OR 0 = 1, 0 OR 0 = 0, 1 OR 1 = 1.

2、Program Example 1:

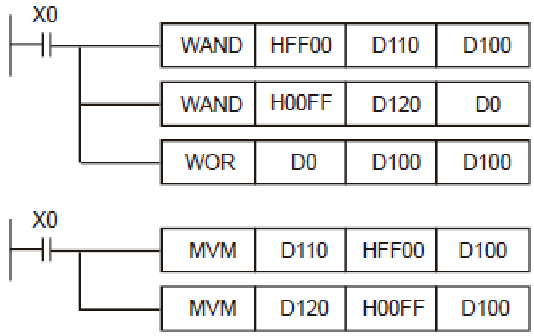
- When X0 = ON, MVM instruction conducts logical AND operation between 16-bit register D0 and H’FF00 first, logical AND operation between D4 and H’00FF secondly, and combines the 1st and 2nd results in D4 by logical OR operation.





3、Program Example 2:

- Simplify instructions:


X0
WAND
HFF00
D110
D100
WAND
H00FF
D120
D0
WOR
D0
D100
D100
X0
MVM
D110
HFF00
D100
MVM
D120
H00FF
D100

ZL 169				
D	HOUR	S D1 D2		Hour Meter

	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S					*	*	*	*	*	*	*	*	*	*	*
D1													*		
D2		*	*	*											

HOUR: 7 steps 16-bit

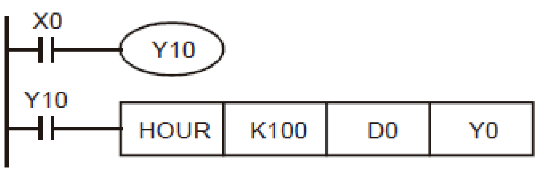
DHOUR: 13 steps 32-bit

1、Explanations:

- S: Period of time when D2 is On (in hour) D1: Current value being measured (in hour)
  D2: Output device
- If S is used in device F, only 16-bit instruction is applicable.
- D1 will occupy 2 consecutive points. D1 + 1 uses 16-bit register in 16-bit or 32-bit instruction.
- Range of S: K1 ~ K32,767 (unit: hour); range of D1: K0 ~ K32,767 (unit: hour). D1 + 1 refers to the current time that is less than an hour (range: K0 ~K3,599; unit: second).
- This instruction times the time and when the time reaches the set time (in hour), D2 will be On. This function allows the user to time the operation of the machine or conduct maintenance works.
- After D2 is On, the timer will resume the timing.
- In the 16-bit instruction, when the current time measured reaches the maximum 32,767 hours/3,599 seconds, the timing will stop. To restart the timing, D1 and D1 + 1 have to be reset to "0".
- n the 32-bit instruction, when the current time measured reaches the maximum 2,147,483,647 hours/3,599 seconds, the timing will stop. To restart the timing, D1 ~ D1 + 2 have to be reset to "0".

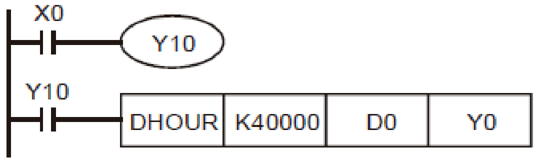
2、Program Example 1:

- In 16-bit instruction, when X0 = On, Y10 will be On and the timing will start. When the timing reaches 100 hours, Y0 will be On and D0 will record the current time measured (in hour) and D1 will record the current time that is less than an hour (0 ~ 3,599; unit: second).



3、Program Example 2:

- In 32-bit instruction, when X0 = On, Y10 will be On and the timing will start. When the timing reaches 40,000 hours, Y0 will be On. D1 and D0 will record the current time measured (in hour) and D2 will record the current time that is less than an hour (0 ~ 3,599; unit: second).



9.4 (ZL 170-179) Gray code conversion/floating point arithmetic

ZL 170	D	GRY					S D									BIN→Gray Code
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GRY: 5 steps 16-bit
S					*	*	*	*	*	*	*	*	*	*	*	DGRY: 9 steps 32-bit
D								*	*	*	*	*	*	*	*	

1、Explanations:

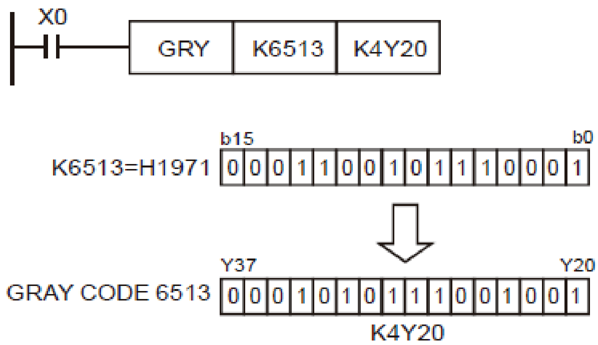
- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S: Source device for BIN value D: Device for storing Gray code
- If S and D are used in device F, only 16-bit instruction is applicable.
- This instruction converts the BIN value in the device designated in S into Gray code and stores the value in D.
- See the ranges of S as indicated below. If S exceeds the ranges, the program will regard it as an operation error and the instruction will not be executed. M1067 and M1068 will be On and D1067 will record the error code 0E1A (hex).

  In 16-bit instruction: 0 ~ 32,767

  In 32-bit instruction: 0 ~ 2,147,483,647

2、Program Example:

- When X0 = On, the instruction will convert constant K6,513 into Gray code and store the result in K4Y20.



ZL 171		GBIN		S D	Gray Code→BIN
	D				

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GBIN: 5 steps 16-bit
S					*	*	*	*	*	*	*	*	*	*	*	DGBIN: 9 steps 32-bit
D								*	*	*	*	*	*	*	*	

1、Explanations:

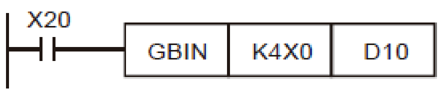
- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S: Source device for Gray code D: Device for storing BIN value
- If S and D are used in device F, only 16-bit instruction is applicable.
- This instruction converts the Gray code in the device designated in S into BIN value and stores the value in D.
- This instruction converts the content (in Gray code) in the absolute position encoder connected at the PLC input terminal into BIN value and store the result in the designated register.
- See the ranges of S as indicated below. If S exceeds the ranges, the program will regard it as an operation error and the instruction will not be executed.

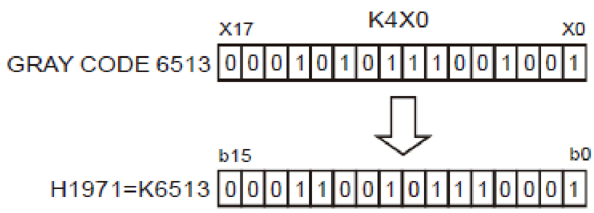
  In 16-bit instruction: 0 ~ 32,767

  In 32-bit instruction: 0 ~ 2,147,483,647

2、Program Example:

- When X20 = On, the Gray code in the absolute position encoder connected at X0 ~ X17 will be converted into BIN value and stored in D10.





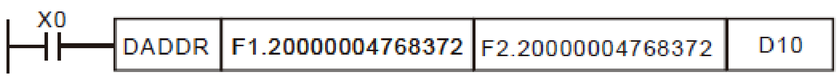
ZL 172	D	ADDR	S1 S2 D													Floating Point Addition
	Bit Devices				Word Devices											DADDR 13 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1													*			
S2													*			
D													*			

1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Floating point summand   S2: Floating point addend   D: Sum
- S1 and S2 can be floating point values (FX.XX).
- In DADDR instruction, floating point values (e.g. F1.2) can be entered directly into S1 and S2 or stored in register D for operation. When the instruction is being executed, operand D will store the operation result.
- When S1 and S2 stores the floating point values in register D, their functions are the same as API 120 EADD.
- S1 and S2 can designate the same register. In this case, if the "continuous execution" type instruction is in use and during the On period of the drive contact, the register will be added once in every scan by a "pulse execution" type instruction (DADDRP).

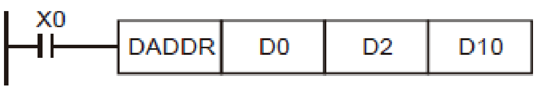
2、Program Example 1:

- When X0 = On, the floating point F1.20000004768372 will plus F2.20000004768372 and the result F3.40000009536743 will be stored in the data registers (D10, D11).



3、Program Example 2:

- When X0 = On, the floating point value (D1, D0) + floating point value (D3, D2) and the result will be stored in the registers designated in (D11, D10).



ZL 173	D	SUBR	S1 S2 D	Floating Point Subtraction

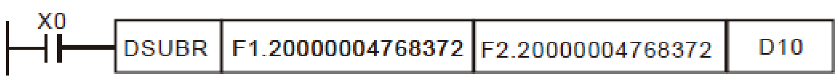
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DSUBR: 13 steps 32-bit
S1													*			
S2													*			
D													*			

1、Explanations:

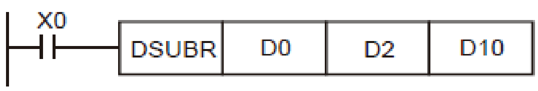
- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Floating point minuend S2: Floating point subtrahend D: Remainder
- S1 and S2 can be floating point values (FX.XX).
- In DSUBR instruction, floating point values (e.g. F1.2) can be entered directly into S1 and S2 or stored in register D for operation. When the instruction is being executed, operand D will store the operation result.
- When S1 and S2 stores the floating point values in register D, their functions are the same as API 121 ESUB.
- S1 and S2 can designate the same register. In this case, if the "continuous execution" type instruction is in use and during the On period of the drive contact, the register will be subtracted once in every scan by a "pulse execution" type instruction (DSUBRP).

2、Program Example 1:

- When X0 = On, the floating point F1.20000004768372 will minus F2.20000004768372 and the result F-1 will be stored in the data registers (D10, D11).



- When X0 = On, the floating point value (D1, D0) – floating point value (D3, D2) and the result will be stored in the registers designated in (D11, D10).



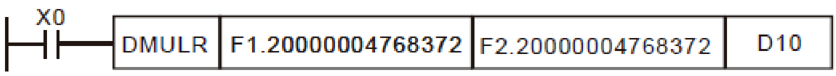
ZL 174	D	MULR				S1 S2 D										Floating Point Multiplication
	Bit Devices				Word Devices											DMULR: 13 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1													*			
S2													*			
D													*			

1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Floating point multiplicand S2: Floating point multiplicator D: Product
- S1 and S2 can be floating point values (FX.XX).
- In DMULR instruction, floating point values (e.g. F1.2) can be entered directly into S1 and S2 or stored in register D for operation. When the instruction is being executed, operand D will store the operation result.
- When S1 and S2 stores the floating point values in register D, their functions are the same as API 122 EMUL.
- S1 and S2 can designate the same register. In this case, if the "continuous execution" type instruction is in use and during the On period of the drive contact, the register will be multiplied once in every scan by a "pulse execution" type instruction (DMULRP).

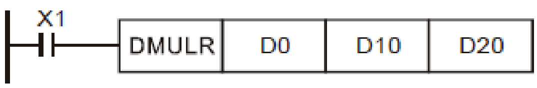
2、Program Example 1:

- When X0 = On, the floating point F1.20000004768372 will multiply F2.20000004768372 and the result F2.64000010490417 will be stored in the data registers (D10, D11).



3、Program Example 2:

- When X1 = On, the floating point value (D1, D0) × floating point value (D11, D10) and the result will be stored in the registers designated in (D21, D20).



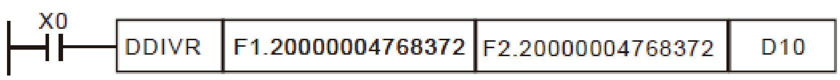
ZL 175		DIVR		S1 S2 D												Floating Point Division
	D															
	Bit Devices				Word Devices											DDIVR: 13 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1													*			
S2													*			
D													*			

1、Explanations:

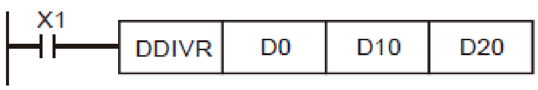
- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Floating point dividend S2: Floating point divisor D: Quotient
- S1 and S2 can be floating point values.
- In DDIVR instruction, floating point values (e.g. F1.2) can be entered directly into S1 and S2 or stored in register D for operation. When the instruction is being executed, operand D will store the operation result.
- When S1 and S2 stores the floating point values in register D, their functions are the same as API 123 EDIV.
- If S2 is "0", the program will regard it as an operation error and the instruction will not be executed.

2、Program Example 1:

- When X0 = On, the floating point F1.20000004768372 will be divided by F2.20000004768372 and the result F0.545454561710358 will be stored in the data registers (D10, D11).



- When X1 = On, the floating point value (D1, D0) ÷ floating point value (D11, D10) and the quotient will be stored in the registers designated in (D21, D20).



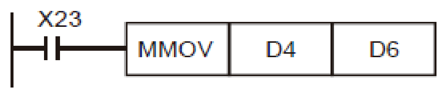
ZL 176	MMOV				S D										Magnifying Transfer with Sign Extension
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MMOV: 5 steps
S					*	*	*	*	*	*	*	*	*			16-bit
D											*	*	*			

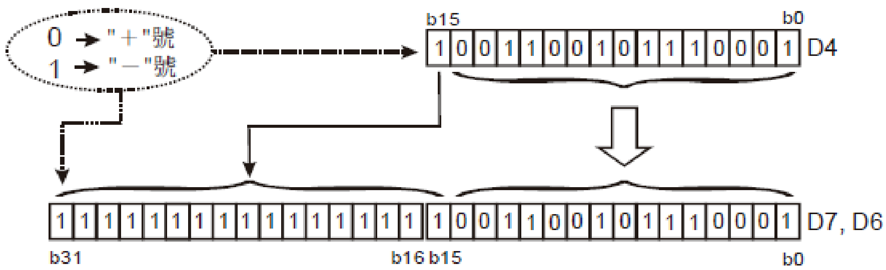
1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S: Data source (16-bit)   D: Data destination (32-bit)
- MMOV instruction sends the data in the 16-bit S device to the 32-bit D device. The designated sign bit will be copied and stored in the destination device.

2、Program Example 1:

- When X23 = On, the data in D4 will be sent to D6 and D7.





ZL 179	D	WSUM	S D n	Sum of multiple devices

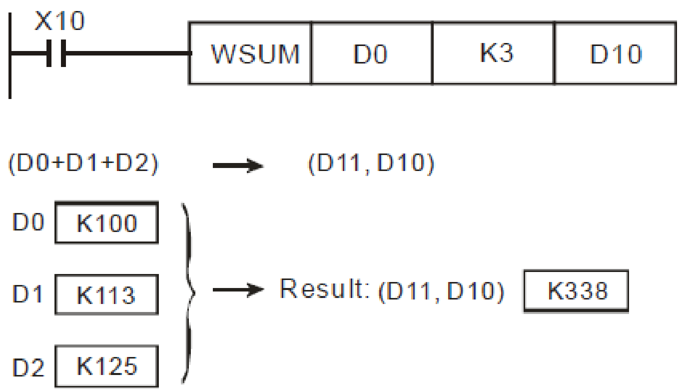
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WSUM: 7 steps DWSUM: 13 steps
S											*	*	*			
D											*	*	*			
n					*	*							*			

1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S: Source device n: Data length to be summed up D: Device for storing the result
- WSUM instruction sums up n devices starting from S and store the result in D.
- If the specified source devices S are out of valid range, only the devices in valid range will be processed.
- Valid range for n: 1~64. If the specified n value is out of the available range (1~64), PLC will take the upper (64) or lower (1) bound value as the set value.
- D used in the 16-bit/32-bit instruction is a 32-bit register.

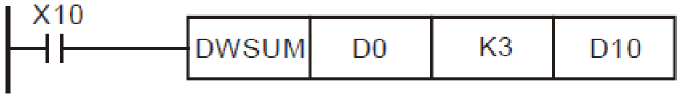
2、Program example 1:

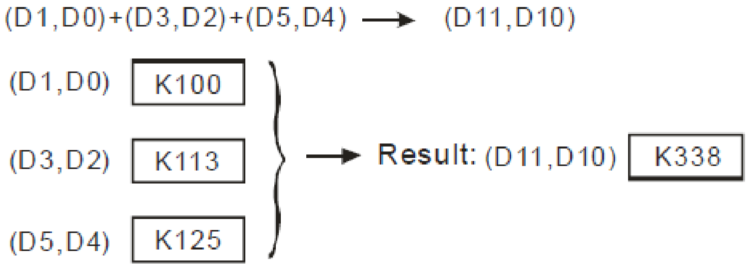
- When X10 = ON, 3 consecutive devices (n = 3) from D0 will be summed up and the result will be stored in (D11, D10).



3、Program example 2:

- When X10 = ON, 3 consecutive devices (n = 3) from (D1, D0) will be summed up and the result will be stored in (D11, D10).


X10
DWSUM
D0
K3
D10


(D1,D0)+(D3,D2)+(D5,D4)
(D11,D10)
(D1,D0)
K100
(D3,D2)
K113
(D5,D4)
K125
Result: (D11,D10)
K338

9.5 (ZL 180-190) Matrix processing

ZL 180	MAND					S1 S2 D n									Matrix 'AND' Operation
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1							*	*	*	*	*	*	*			MAND: 9 steps
S2							*	*	*	*	*	*	*			16-bit
D								*	*	*	*	*	*			
n					*	*							*			

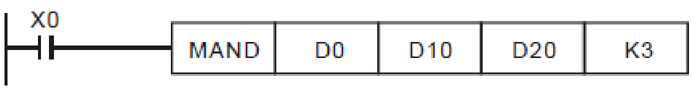
1、Explanations:

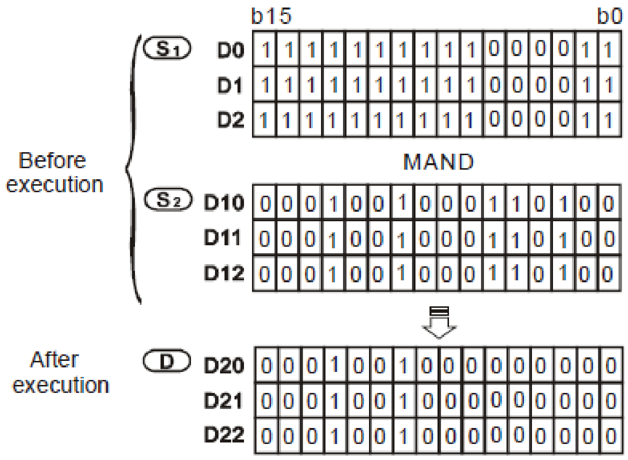
- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Matrix source device 1　　S2: Matrix source device 2

  D: Operation result　　n: Array length
- Range of n: K1 ~ K256.
- S1, and S2 designate KnX, KnY, KnM and KnS; D designates KnYm KnM and KnS.
- esignate n ≦ 4.
- The two matrix sources S1 and S2 perform matrix 'AND' operation according to the array length n. The result is stored in D.
- Operation rule of matix 'AND' : The result will be 1 if both two bits are 1; otherwise the result will be 0.

2、Program Example:

- When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 and the 3 arrays of 16-bit registers D10 ~ D12 will perform a matrix 'AND' operation. The result will be stored in the 3 arrays of 16-bitd registers D20 ~ D22.





Remarks:

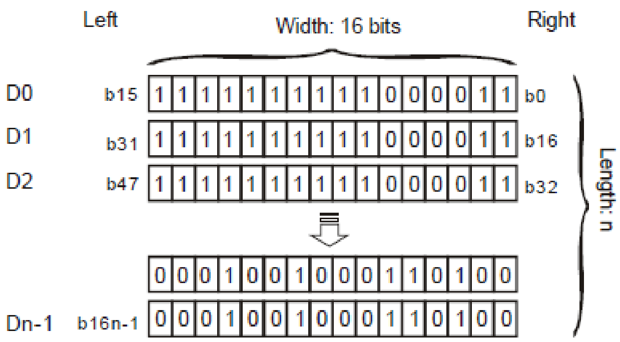
1) A matix consists of more than 1 consecutive 16-bit registers. The number of registers in the matrix is the length of the array (n). A matrix contains 16 × n bits (points) and there is only 1 bit (point) offered for an operand at a time.

2) The matrix instruction gathers a series of 16 × n bits (b0 ~ b16n-1) and designates a single point for operation. The point will not be seen as a value.

3) The matrix instruction processes the moving, copying, comparing and searching of one-to-many or many-to-many matrix status, which is a very handy and important application instruction.

4) The matrix operation will need a 16-bit register to designate a point among the 16n points in the matrix for the operation. The register is the Pointer (Pr) of the matrix, designated by the user in the instruction. The vaild range of Pr is 0 ~ 16n -1, corresponding to b0 ~ b16n-1 in the matrix.

5) There are left displacement, right displacement and rotation in a matrix operation. The bit number decreases from left to right (see the figure below).



6) The matrix width (C) is fixed at 16 bits.

7) Pr: matrix pointer. E.g. if Pr is 15, the designated point will be b15.

8) Array length (R) is n: n = 1 ~ 256.

Example: The matrix is composed of D0, n = 3; D0 = HAAAA, D1 = H5555, D2 = HAAFF

	C_{15}	C_{14}	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0	
R_0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	D0
R_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D1
R_2	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	D2

Example: The matrix is composed of K2X0, n = 3; K2X0 = H37, K2X10 = H68, K2X20 = H45

	C_{15}	C_{14}	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0	
R_0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	X_0~X_7
R_1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	X_{10}~X_{17}
R_2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	X_{20}~X_{27}

ZL 181	MOR	S1 S2 D n	Matrix 'OR' Operation

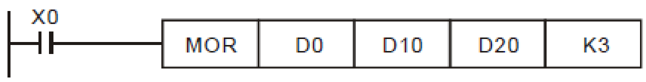
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1							*	*	*	*	*	*	*			MOR: 9 steps
S2							*	*	*	*	*	*	*			16-bit
D								*	*	*	*	*	*			
n					*	*							*			

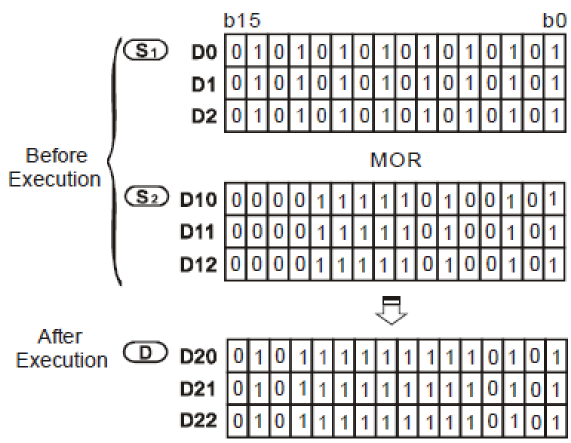
1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Matrix source device 1   S2: Matrix source device 2.   D: Operation result   n: Array length
- Range of n: K1 ~ K256.
- S1, and S2 designate KnX, KnY, KnM and KnS; D designates KnYm KnM and KnS.
- esignate n ≦ 4.
- The two matrix sources S1 and S2 perform matrix 'OR' operation according to the array length n. The result is stored in D.
- Operation rule of matrix 'OR' : The result will be 1 if either of the two bits is 1. The result is 0 only when both two bits are 0.

2、Program Example:

- When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 and the 3 arrays of 16-bit registers D10 ~ D12 will perform a matrix 'OR' operation. The result will be stored in the 3 arrays of 16-bit registers D20 ~ D22.





ZL 182		MXOR		S1 S2 D n											Matrix 'XOR' Operation
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1							*	*	*	*	*	*	*		
S2							*	*	*	*	*	*	*		
D								*	*	*	*	*	*		
n					*	*							*		

MXOR: 9 steps

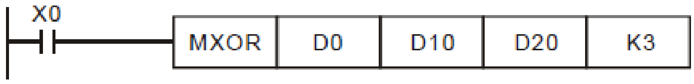
16-bit

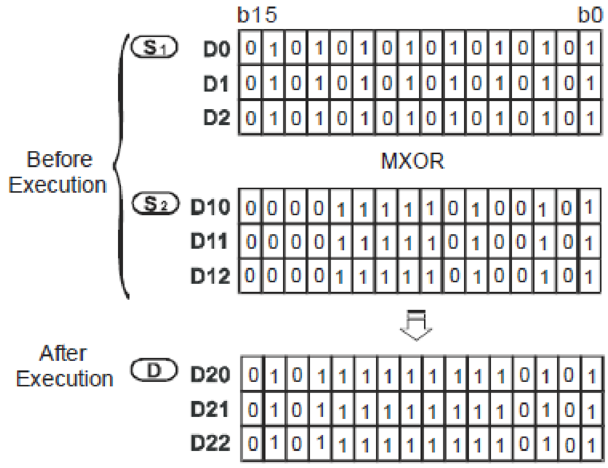
1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Matrix source device 1 S2: Matrix source device 2 D: Operation result n: Array length
- Range of n: K1 ~ K256.
- S1, and S2 designate KnX, KnY, KnM and KnS; D designates KnYm KnM and KnS.
- The two matrix sources S1 and S2 perform matrix 'XOR' operation according to the array length n. The result is stored in D.
- Operation rule of matrix 'XOR': The result will be 1 if the two bits are different. The result will be 0 if the two bits are the same.

2、Program Example:

- When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 and the 3 arrays of 16-bit registers D10 ~ D12 will perform a matrix 'XOR' operation. The result will be stored in the 3 arrays of 16-bit registers D20 ~ D22.


X0
MXOR
D0
D10
D20
K3


b15
b0
Before Execution
S1
D0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
D1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
D2 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
MXOR
S2
D10 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1
D11 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1
D12 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1
After Execution
D
D20 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1
D21 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1
D22 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1

ZL 183	MXNR	S1 S2 D n	Matrix ‘XNR’ Operation

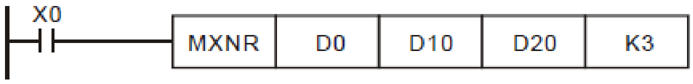
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1							*	*	*	*	*	*	*			MXNR: 9 steps
S2							*	*	*	*	*	*	*			16-bit
D								*	*	*	*	*	*			
n					*	*							*			

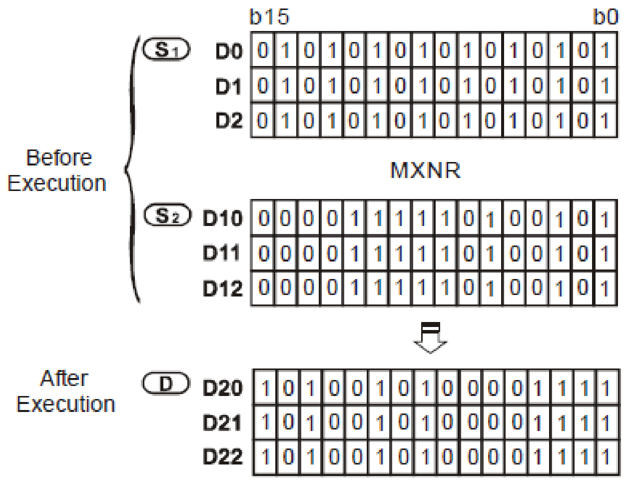
1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Matrix source device 1 S2: Matrix source device 2 D: Operation result n: Array length
- Range of n: K1 ~ K256.
- S1, and S2 designate KnX, KnY, KnM and KnS; D designates KnYm KnM and KnS.
- The two matrix sources S1 and S2 perform matrix ‘XNR’ operation according to the array length n. The result is stored in D.
- Operation rule of matrix ‘XNR’: The result will be 1 if the two bits are the same. The result will be 0 if the two bits are different.

2、Program Example:

- When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 and the 3 arrays of 16-bit registers D10 ~ D12 will perform a matrix ‘XNR’ operation. The result will be stored in the 3 arrays of 16-bit registers D20 ~ D22.


X0
MXNR
D0
D10
D20
K3


b15
b0
S1
D0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
D1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
D2 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
Before Execution
MXNR
S2
D10 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1
D11 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1
D12 0 0 0 0 1 1 1 1 1 0 1 0 0 1 0 1
After Execution
D
D20 1 0 1 0 0 1 0 1 0 0 0 0 1 1 1 1
D21 1 0 1 0 0 1 0 1 0 0 0 0 1 1 1 1
D22 1 0 1 0 0 1 0 1 0 0 0 0 1 1 1 1

ZL 184	MINV	S D n	Matrix Inverse Operation

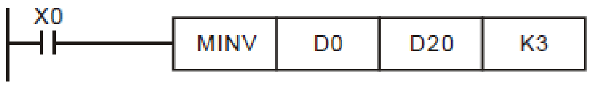
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
	Bit Devices				Word Devices											
S							*	*	*	*	*	*	*			MINV: 7 steps
D								*	*	*	*	*	*			16-bit
n					*	*							*			

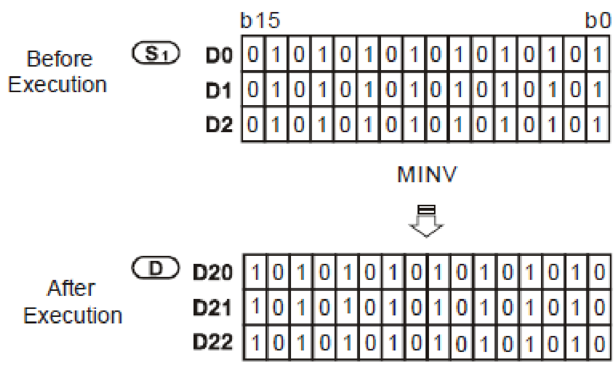
1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S: Matrix source device D: Operation result n: Array length
- Range of n: K1 ~ K256
- S designates KnX, KnY, KnM and KnS; D designates KnY, KnM and KnS.
- S performs an inverse matrix operation according to the array length n. The result is stored in D.

2、Program Example:

- When X0 = On, the 3 arrays of 16-bit registers D0 ~ D2 perform a matrix inverse operation. The result will be stored in the 3 arrays of 16-bit registers D20 ~ D22.





9.6 (ZL 191-199) Positioning instructions

ZL 192	D	PPMA	S1 S2 S D	2-Axis Absolute Point to Point Motion

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*							*			DPPMA: 17 steps
S2					*	*							*			32-bit
S					*	*							*			
D		*														

1、Explanations:

- S1: Number of output pulses of X axis S2: Number of output pulses of Y axis

  S: Max. point to point output frequency D: Pulse output device
- In terms of pulse output methods, this instructin only supports "pulse + direction" mode.
- S1 and S2 are the designated (absolute designation) number of output pulses in X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of the number is -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in forward direction, the pulse present value registers CH0 (D1649 high word, D1648 low word), CH1 (D1665high word, D1664 low word), CH2 (D1681high word, D1680 low word) and CH3 (D1697 high word, D1696 low word) will increase. When in backward direction, the present value will decrease.
- D can designate Y0 and Y4.

  When Y0 is designated:

  Y0 refers to 1st group X-axis pulse output device.

  Y1 refers to 1st group X-axis direction signal.

  Y2 refers to 1st group Y-axis pulse output device.

  Y3 refers to 1st group Y-axis direction signal.

  Y4 refers to 2nd group X-axis pulse output device.

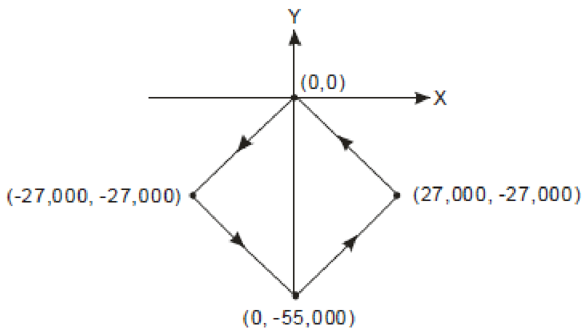
  Y5 refers to 2nd group X-axis direction signal.

  Y6 refers to 2nd group Y-axis pulse output device.

  Y7 refers to 2nd group Y-axis direction signal.

- When direction signal outputs, Off will not occur immediately after the pulse output is over. Direction signal will turn Off when the drive contact is Off.
- Refer to DDRVI and DDRVA instructions for special M and D corresponding to each channel.
- The time shall be longer than 10ms. If the time is shorter than 10ms or longer than 10,000ms, the output will be operated at 10ms. Default setting = 100ms.
- If the maximum output frequency setting is less than 10Hz, the output will be operated at 10Hz. If the setting is more than 200kHz, the output will be operated at 200kHz.
- When the 2-axis synchronous motion instruction is enabled, the start frequency and acceleration/deceleration time in Y axis will be same as the settings in X axis.
- The number of output pulses for the 2-axis motion shall not be the values within 1 ~ 59; otherwise the line drawn will not be straight enough.
- There is no limitation on the number of times using the instruction. However, assume CH1 or CH2 output is in use, the 1st group X/Y axis will not be able to output. If CH3 or CH4 output is in use, the 2nd group X/Y axis will not be able to output.

2、Program Example：Draw a rhombus as the figure below

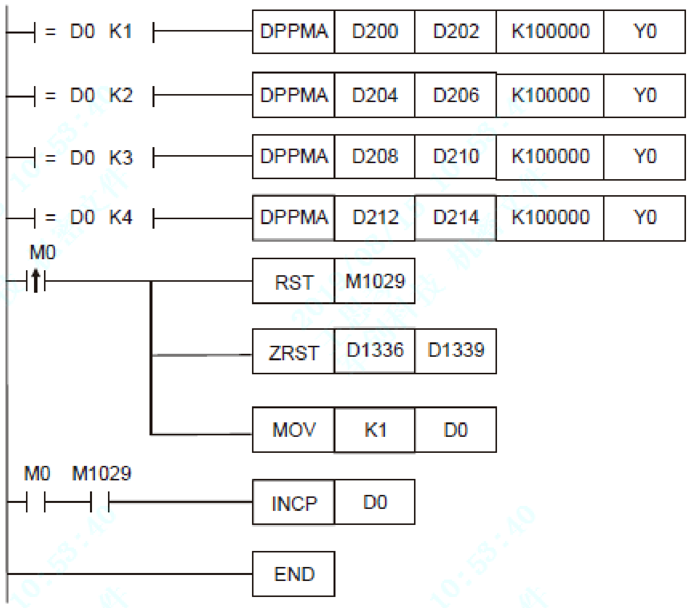


- Steps:

 1）Set the four coordinate (-27,000, -27,000), (0, -55,000), (27,000, -27,000), (0, 0) (as the figure above). Place them in the 32-bit (D200, D202), (D204, D206), (D208, D210), (D212, D214).

 2) Write program codes as follows.

 3) PLC RUN. Set M0 as On and start the 2-axis line drawing.



- Motion explanation:

  When PLC RUN and M0 = On, PLC will start the first point-to-point motion by 100kHz. D0 will plus 1 whenever a point-to-point motion is completed and the second point-to-point motion will start to execute automatically. The same motion will keep executing until the fourth point-to-point motion is completed.

ZL 194		CIMA		S1 S2 S D	2-Axis Absolute Position Arc Interpolation
	D				

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*							*			DCIMA: 17 steps
S2					*	*							*			32-bit
S													*			
D		*														

1、 Explanations:

Mode 0:

- S1: Number of output pulses of X axis S2: Number of output pulses of Y axis

  S: Parameter setting D: Pulse output device
- In terms of pulse output methods, this instructin only supports "pulse + direction" mode.
- S1 and S2 are the designated (absolute designation) number of output pulses in X axis (Y0 or Y4) and Y axis (Y2 or Y6). The range of the number is -2,147,483,648 ~ +2,147,483,647. The pulse present value register will increase when in the positive direction. In the opposite direction, it will decrease.
- S (direction and resolution setting): Set K0 to output 10 segments clockwise (normal resolution), and set K2 to output 20 segments clockwise (higher resolution), you can draw a 90° arc as shown in Figure (1) , (2); set K1 to output 10 counterclockwise segments (normal resolution), and set K3 to output 20 counterclockwise segments (higher resolution), you can draw a 90° arc as shown in (3), (4) shown.

  S is K0 or K1, which means working in mode 0, and S is K2 or K3, which means working in mode 1.
- S+1~S+2 (walking frequency setting): The general setting range is 10hz~K200000hz.

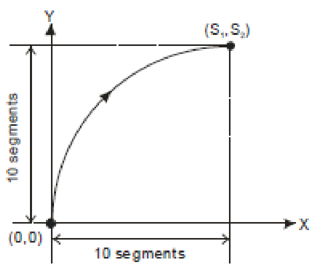


Figure 1

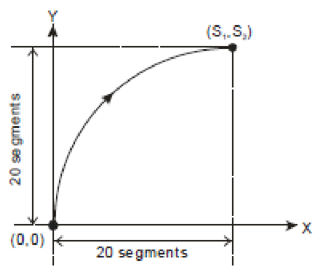


Figure 2

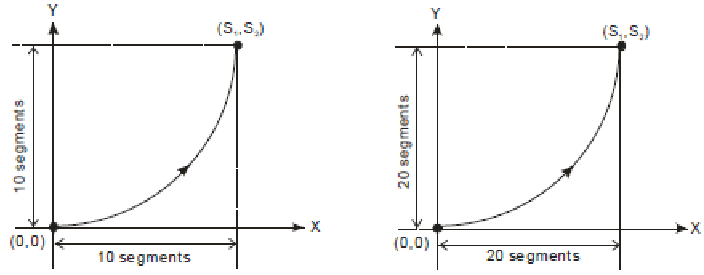


Figure 3 Figure 4

- D can designate Y0 and Y4.

  When Y0 is designated:

  Y0 refers to 1st group X-axis pulse output device.

  Y1 refers to 1st group X-axis direction signal.

  Y2 refers to 1st group Y-axis pulse output device.

  Y3 refers to 1st group Y-axis direction signal.

  When Y4 is designated:

  Y4 refers to 2nd group X-axis pulse output device.

  Y5 refers to 2nd group X-axis direction signal.

  Y6 refers to 2nd group Y-axis pulse output device.

  Y7 refers to 2nd group Y-axis direction signal.

  When direction signal outputs, Off will not occur immediately after the pulse output is over. Direction signal will turn Off when the drive contact is Off.

- Draw four 90° arcs.
- When the direction signal is On, the direction is positive. When the direction signal is Off, the direction is negative. When S is set as K0, K2, the arcs will be clockwise (see figure 5). When S is set as K1, K3, the arcs will be counterclockwise (see figure 6).

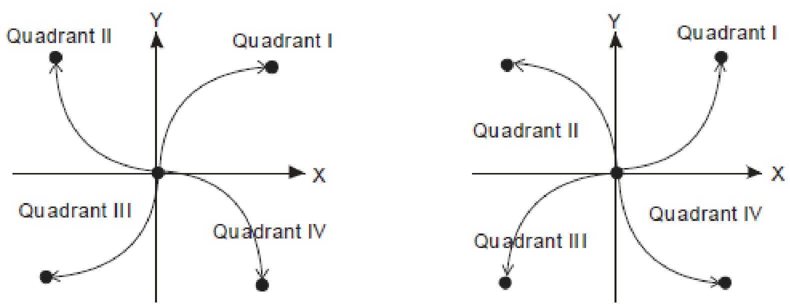


Figure 5 Figure 6

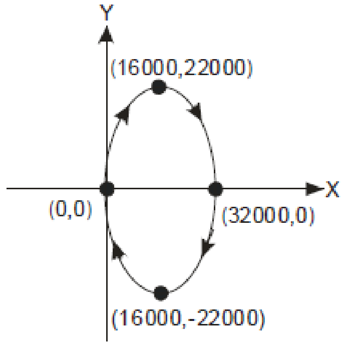
- When the 2-axis motion is being executed in 10 segments (of average resolution), the operation time of the instruction when the instruction is first enabled is approximately 5ms. The number of output pulses cannot be less than 100 and more than 1,000,000; otherwise, the instruction cannot be enabled.

- When the 2-axis motion is being executed in 20 segments (of high resolution), the operation time of the instruction when the instruction is first enabled is approximately 10ms. The number of output pulses cannot be less than 1,000 and more than 10,000,000; otherwise, the instruction cannot be enabled.
- If you wish the number of pulses in 10-segment or 20-segment motion to be off the range, you may adjust the gear ratio of the servo for obtaining your desired number.
- Every time when the instruction is executed, only one 90° arc can be drawn. It is not necessary that the arc has to be a precise arc, i.e. the numbers of output pulses in X and Y axes can be different.
- There are no settings of start frequency and acceleration/deceleration time.
- There is no limitation on the number of times using the instruction.
- The settings of motion time in the high 16 bits of S can be slower than the the fastest suggested time but shall not be faster than the fastest suggested time.
- The fastest suggested time for the arc interpolation:

Mode 1:

- S1~S1+1 represent the center of the X-axis. S2~S2+1 represent the center of the Y-axis. S: parameter setting. D: Pulse output device.
- This command pulse output mode only supports "pulse + direction" mode.
- S (direction and resolution setting): Set K0 to output 10 segments clockwise (normal resolution), and set K2 to output 20 segments clockwise (higher resolution), you can draw a 90° arc as shown in Figure (1) , (2); set K1 to output 10 counterclockwise segments (normal resolution), and set K3 to output 20 counterclockwise segments (higher resolution), you can draw a 90° arc as shown in (3), (4) shown.
- S is K0 or K1, which means working in mode 0, and S is K2 or K3, which means working in mode 1.
- S+1~S+2 (walking frequency setting): The general setting range is 10hz~K200000hz.
- S+3~S+4 indicates that the length of the arc to be executed is in degrees, and the format is a floating-point number. For example, F150.23, expressed as 150.23 degrees.
- S+5~S+6 represents the length of the arc that has been run (read-only, the format is a floating point number, such as F125.23, which means that it has run 125.23 degrees).

2、Program Example 1: Draw an ellipse as shown below, with mode 0.



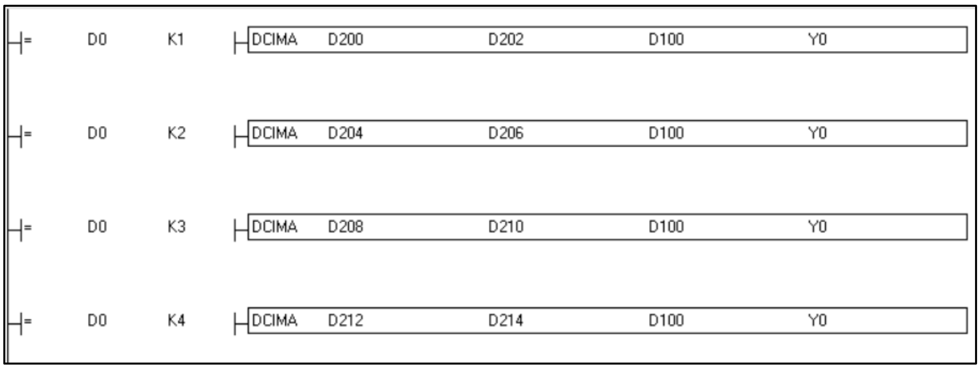
- Steps:

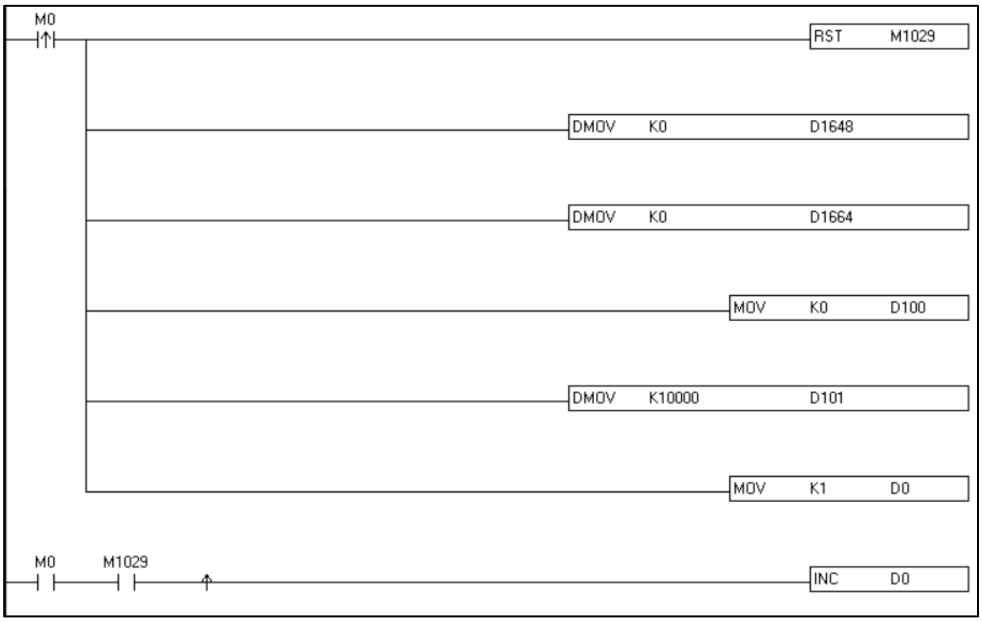
1) Set the four coordinates (0, 0), (16000, 22000), (32000, 0), (16000, -22000) (as the figure above). Place them in the 32-bit (D200, D202), (D204, D206), (D208, D210), (D212, D214).

2) Select “draw clockwise arc” and “average resolution” (S =D100= K0).

3) Select DCIMA instruction for drawing arc and write program codes as follows.

4) PLC RUN. Set M0 as On and start the drawing of the ellipse.



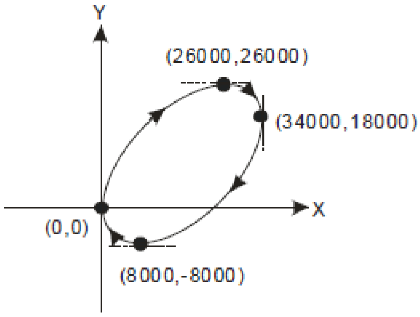


- Motion explanation:

When PLC RUN and M0 = On, PLC will start the drawing of the first segment of the arc. D0 will plus

1 whenever a segment of arc is completed and the second segment of the arc will start to execute automatically. The same motion will keep executing until the fourth segment of arc is completed。

3、Program Example 2：Draw a tilted ellipse as the figure below



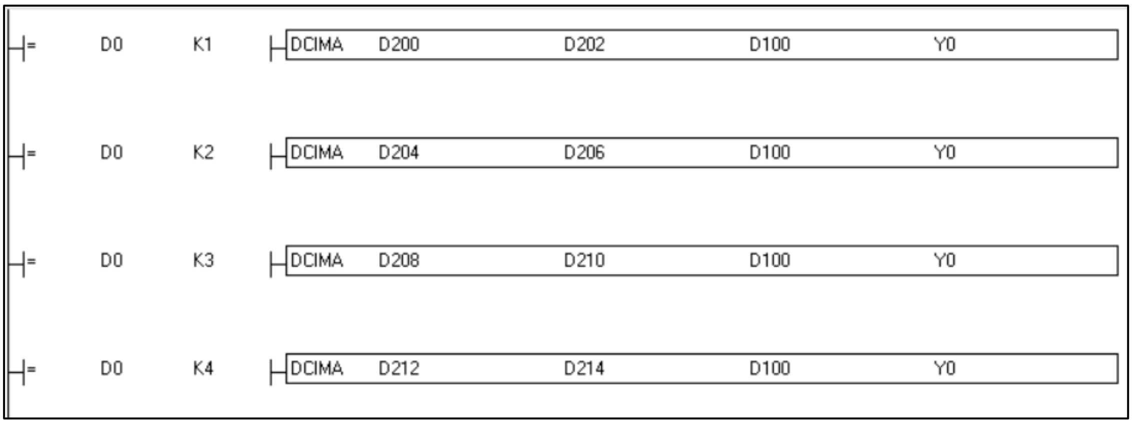
- Steps：

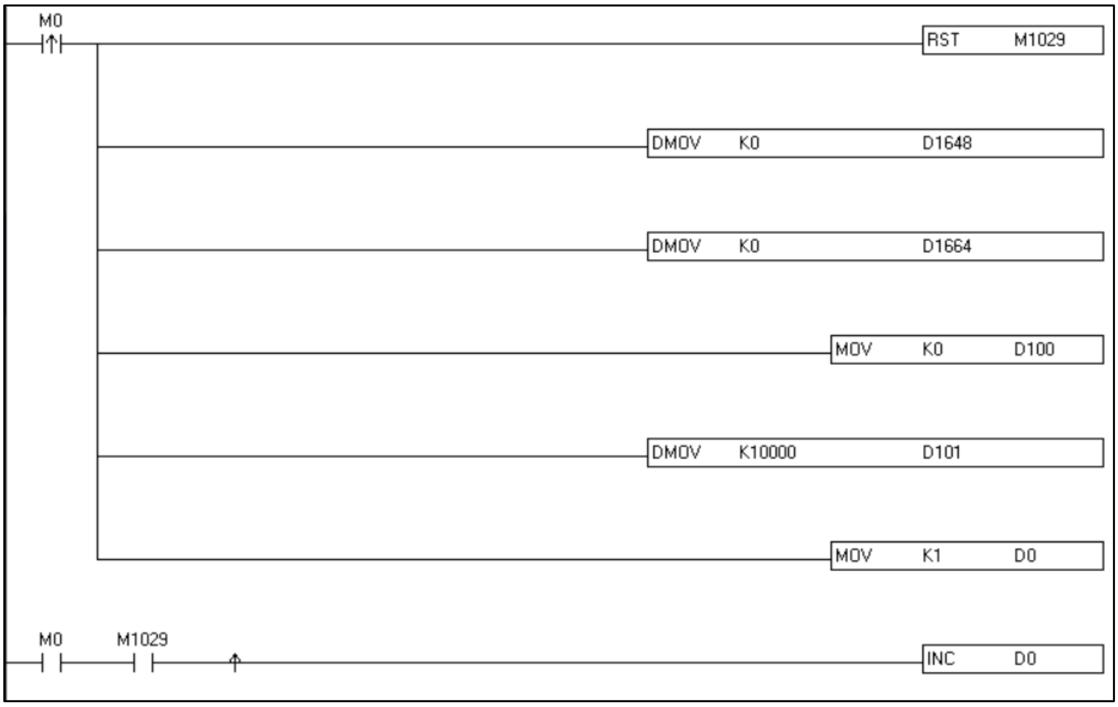
1） Find the max. and min. coordinates on X and Y axes (0, 0), (26000, 26000), (34000, 18000), (8000, -8000) (as the figure above). Place them respectively in the 32-bit (D200, D202), (D204, D206), (D208, D210) and (D212, D214).

2) Select “draw clockwise arc” and “average resolution” (S =D100= K0).

3) Select DCIMA instruction for drawing arc and write program codes as follows.

4) PLC RUN. Set M0 as On and start the drawing of the ellipse.





- Motion explanation:

  When PLC RUN and M0 = On, PLC will start the drawing of the first segment of the arc. D0 will plus 1 whenever a segment of arc is completed and the second segment of the arc will start to execute automatically. The same motion will keep executing until the fourth segment of arc is completed.

ZL 195	D	PTPO	S1 S2 D	Single-Axis Pulse Output by Table

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DPTPO: 13 steps 32-bit
S1													*			
S2													*			
D		*														

1、Explanations:

- S1: Source start device S2: Number of segments D: Pulse output device
- According to the value of S2 + 0, every segment consecutively occupy four register D. (S1 + 0) refers to output frequency. (S1 + 2) refers to the number of output pulses.
- When the output frequency of S1 is less than 1, PLC will automatically modify it as 1. When the value is larger than 200,000kHz, PLC will automatically modify it as 200,000kHz.
- S2 + 0: number of segments (range: 1 ~ 60). S2 + 1: number of segments being executed. Whenever the program scans to this instruction, the instruction will automatically update the segment No. that is currently being executed. D can only designate output devices Y0, Y2, Y4 and Y6 and can only perform pulse output control. For the pin for direction control, the user has to compile other programs to control.
- This instruction does not offer acceleration and deceleration functions. Therefore, when the instruction is disabled, the output pulses will stop immediately.
- In every program scan, each channel can only be executed by one instruction. However, there is no limitation on the number of times using this instruction.
- When the instruction is being executed, the user is not allowed to update the frequency or number of the segments. Changes made will not be able to make changes in the actual output.

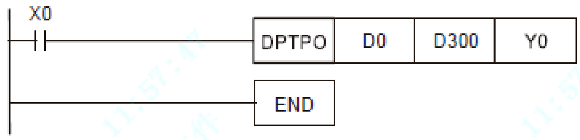
2、Program Example:

- When X0 = On, the output will be operated according to the set frequency and number of pulses in every segment.
- Format of the table：

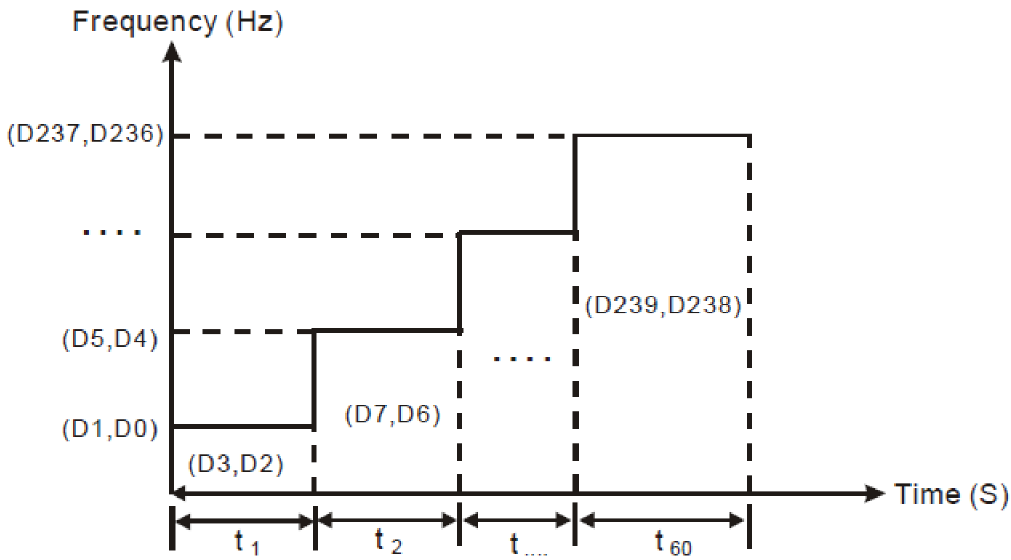
S2=D300， number of segments (D300=K60)	S1=D0，frequency (S1+0)	S1=D0，number of output pulses (S1+2)

K1 1st segment))	D1, D0	D3, D2
K2(2nd segment)	D5, D4	D7, D6
: :	: :	: :
K60(60th segment))	D237, D236	D239, D238

- Monitor the segment No. that is currently being executed in register D301.



- The pulse output curve:



- among them: t1=(D3,D2)÷(D1,D0); t2=(D7,D6)÷(D5,D4); t60=(D239,D238)÷(D237,D236)

ZL 197	D	CLLM	S1 S2 S D	Close Loop Position Control

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1	*											*				DCLLM: 17 steps
S2					*	*							*			32-bit
S					*	*							*			
D		*														

1、Explanations:

- S1: Feedback source device S2: Target number of feedbacks

  S3: Target frequency of output D: Pulse output device

  The corresponding interruption of S1:

Source device	X0	X1	X2	X3	C241-C254			
Corresponding outout	Y0	Y2	Y4	Y6	Y0	Y2	Y4	Y6
Interruption No.	I00□	I10□	I20□	I30□	I010	I020	I030	I040

  note: □ = 1: rising-edige trigger、□ = 0 falling-edge trigger

- 1) When S1 designates X as the input points and the pulse output reaches the set target number of feedbacks in S2, the output will continue to operate by the frequency of the last segment until the interruption of X input points occurs.

  2) When S1 designates a high speed counter and the pulse output reaches the set target number of feedbacks in S2, the output will continue to operate by the frequency of the last segment until the feedback pulses reaches the target number.

  3) The range of S2: -2,147,483,648 ~ +2,147,483,647 (+/- represents the forward/backward direction). When in forward direction, the pulse present value registers CH0 (D1649 high word, D1648 low word), CH1 (D1665 high word, D1664 low word), CH2 (D1681 high word, D1680 low word) and CH3 (D1697 high word, D1696 low word) will increase. When in backward direction, the present value will decrease..

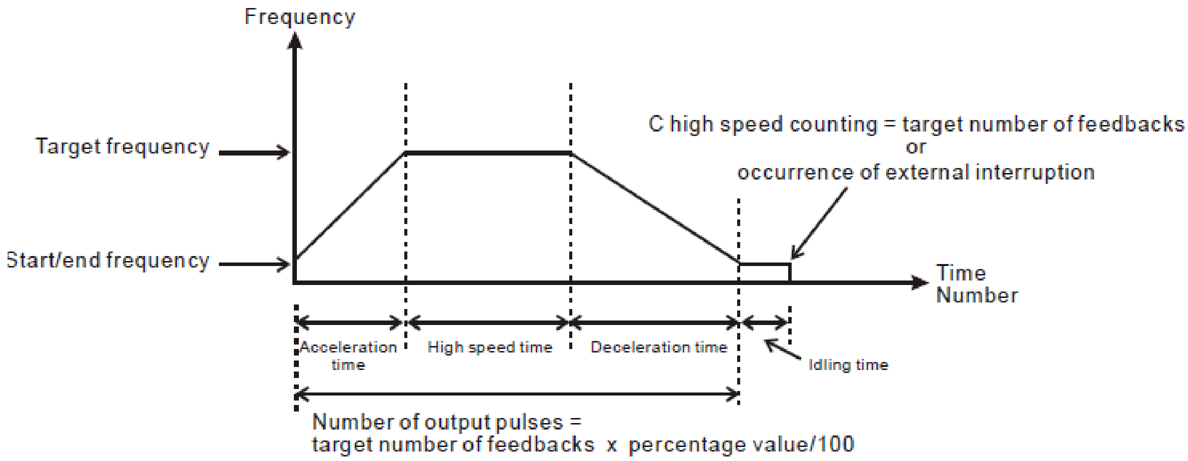
- If S3 is lower than 10Hz, the output will operate at 10Hz; if S3 is higher than 200kHz, the output will operate at 200kHz.

- D can only designate Y0, Y2, Y4 and Y6 and the direction signals repectively are Y1, Y3, Y5 and Y7. When there is a direction signal output, the direction signal will not be Off immediately after the pulse output is completed. The direction signal will be Off only when the drive contact is Off.
- D1340, D1352, D1379 and D1380 are the settings of start/end frequencies of CH0 ~ CH3. The minimun frequency is 10Hz and default is 200Hz.
- D1343, D1353, D1381 and D1382 are the settings of the time of the first segment and the last deceleration segment of CH0 ~ CH3. The acceleration/deceleration time cannot be shorter than 10ms. The outptu will be operated in 10ms if the time set is shorter than 10ms or longer than 10,000ms. The dafault setting is 100ms.
- D1131, D1132, D1478 and D1479 are the output/input ratio of the close loop control in CH0 ~ CH3. K1 refers to 1 output pulse out of the 100 target feedback input pulses; K200 refers to 200 output pulses out of the 100 target feedback input pulses. D1131, D1132, D1478 and D1479 are the numerators of the ratio (range: K1 ~ K10,000) and the denominator is fixed as K100 (the user does not have to enter a denominator).

Group No	PUL	DIR	current number of output pulses (32-bit integer)	Pulse complete flag	Pulse sending	Emergency stop without slowing down	Start frequencyK10-K32767 defaultK200	Accel/Decel timeK10-K10000 defaultK100	deceleration time
CH0（Y0,Y1）	Y0	Y1	D1648	M1029	M1344	M1308	D1340	D1343	D1936
CH1（Y2,Y3）	Y2	Y3	D1664	M1030	M1345	M1309	D1352	D1353	D1937
CH2（Y4,Y5）	Y4	Y5	D1680	M1036	M1346	M1310	D1379	D1381	D1938
CH3（Y6,Y7）	Y6	Y7	D1696	M1037	M1347	M1311	D1380	D1382	D1939

2、Close Loop Explanations:

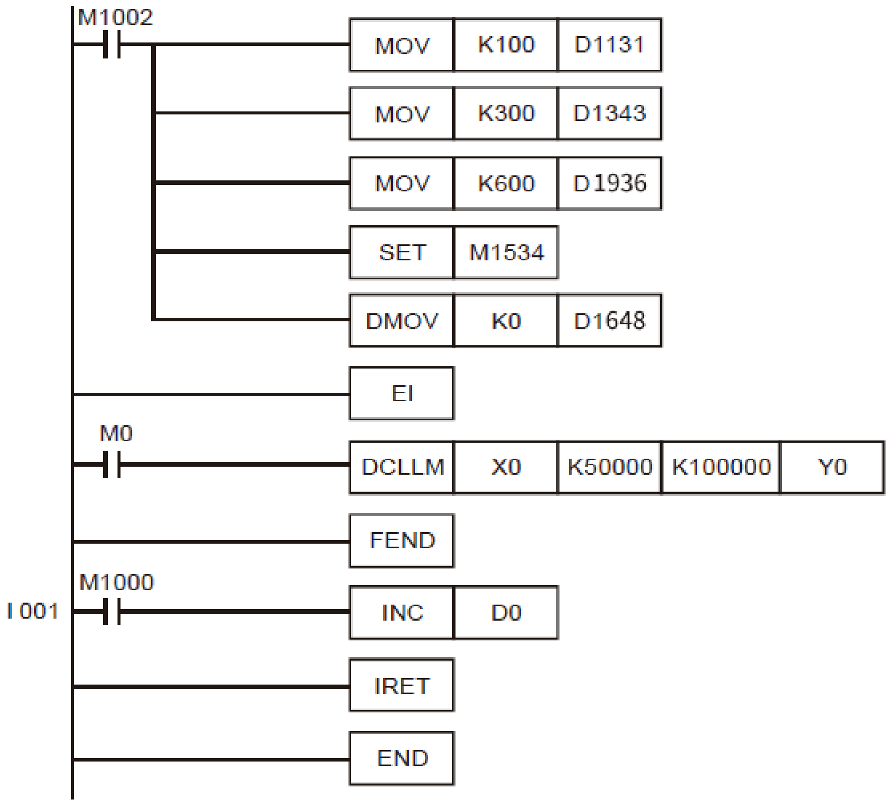
- Function: Immediately stop the high-speed pulse output according to the number of feedback pulses or external interruption signals.
- The execution：

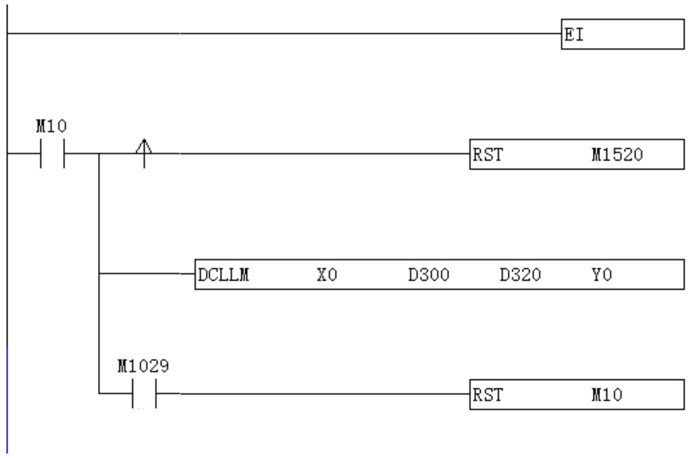


- How to adjust the time for the completion of the positioning:
  1) The time for the completion of the positioning refers to the time for "acceleration + high speed + deceleration + idling" (see the figure above). For example, you can increase or decrease the entire number of output pulses by making adjustment on the percentage value and further increase or decrease the time required for the positioning.
  2) Among the four segments of time, only the idling time cannot be adjusted directly by the user. However, you can determine if the execution result is good or bad by the length of the idling time. In theory, a bit of idling left is the best result for a positioning.
  3) Owing to the close loop operation, the length of idling time will not be the same in every execution. Therefore, when the content in the special D for displaying the actial number of output pulses is smaller or larger than the calculated number of output pulses (taget number of feedbacks x percentage value/100), you can improve the situation by adjusting the percentage value, acceleration/decelartion time or target frequency.

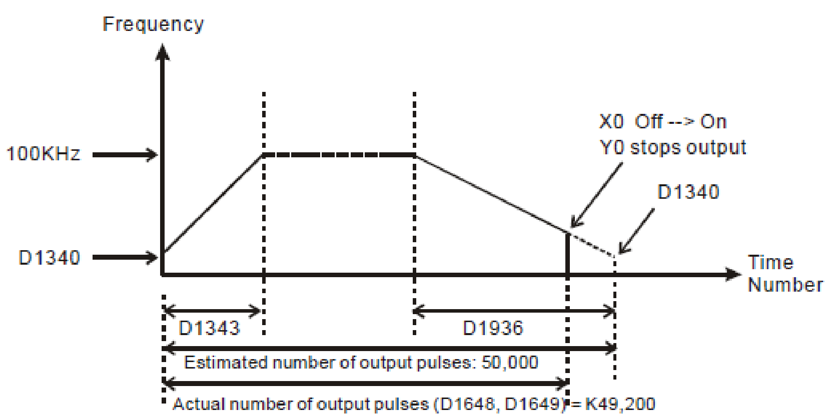
3、Program Example:

- Assume we adopt X0 as the external interruption, together with I001 (rising-edge trigger) interruption program; target number of feedbacks = 50,000; target frequency = 10kHz; Y0, Y1 (CH0) as output pulses; start/end frequency (D1340) = 200Hz; acceleration time (D1343) = 300ms; deceleration time (D1936) = 600ms; percentage value (D1131) = 100; current number of output pulses (D1648, D1649) = 0.





- Assume the first execution result as:

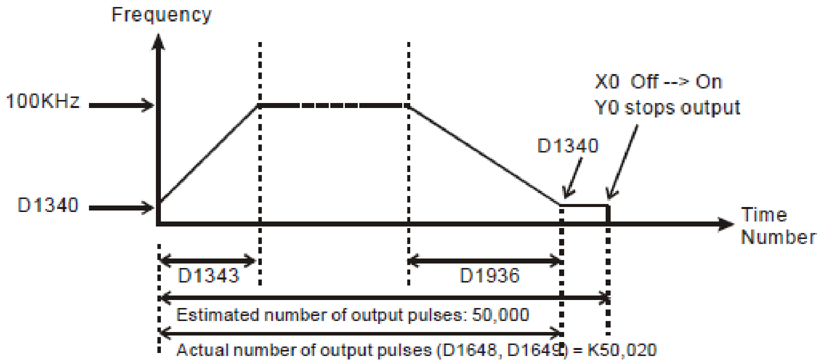


- Observe the result of the first execution:

  1) The actual output number 49,200 – estimated output number 50,000 = -800 (a negative value). A negative value indicates that the entire execution finishes earlier and has not completed yet.

  2) Try to shorten the acceleration time (D1343) into 250ms and deceleration time (D1936) into 550ms.

- Obtain the result of the second execution:



- Observe the result of the second execution:

  1) The actual output number 50,020 – estimated output number 50,000 = 20

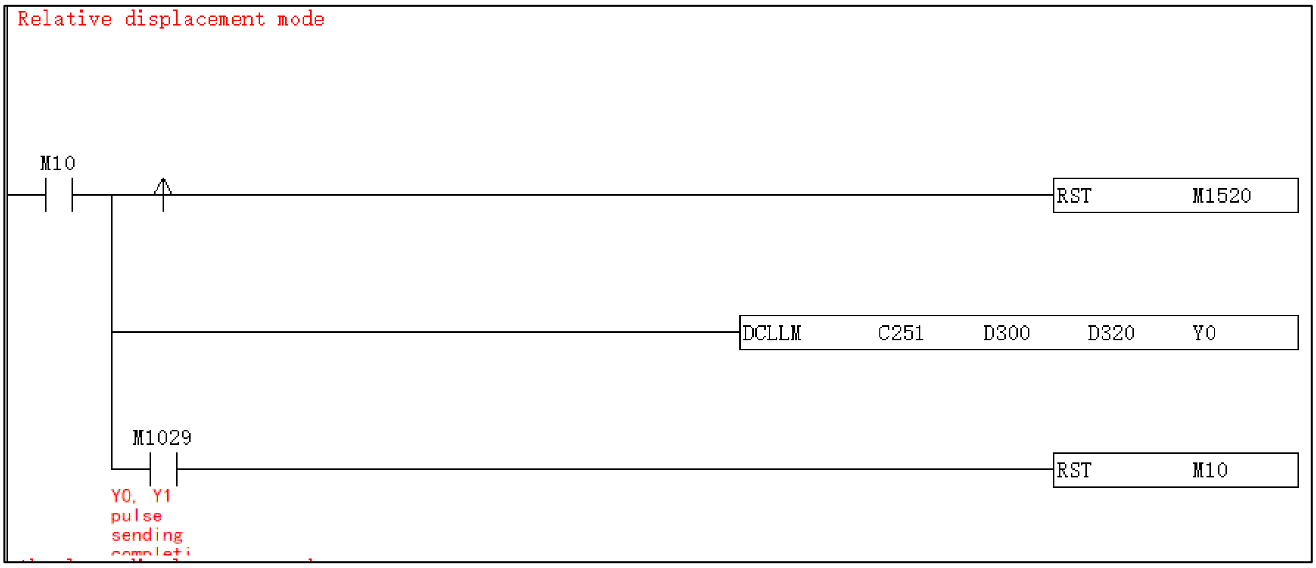
  2) 20 x (1/200Hz) = 100ms (idling time)

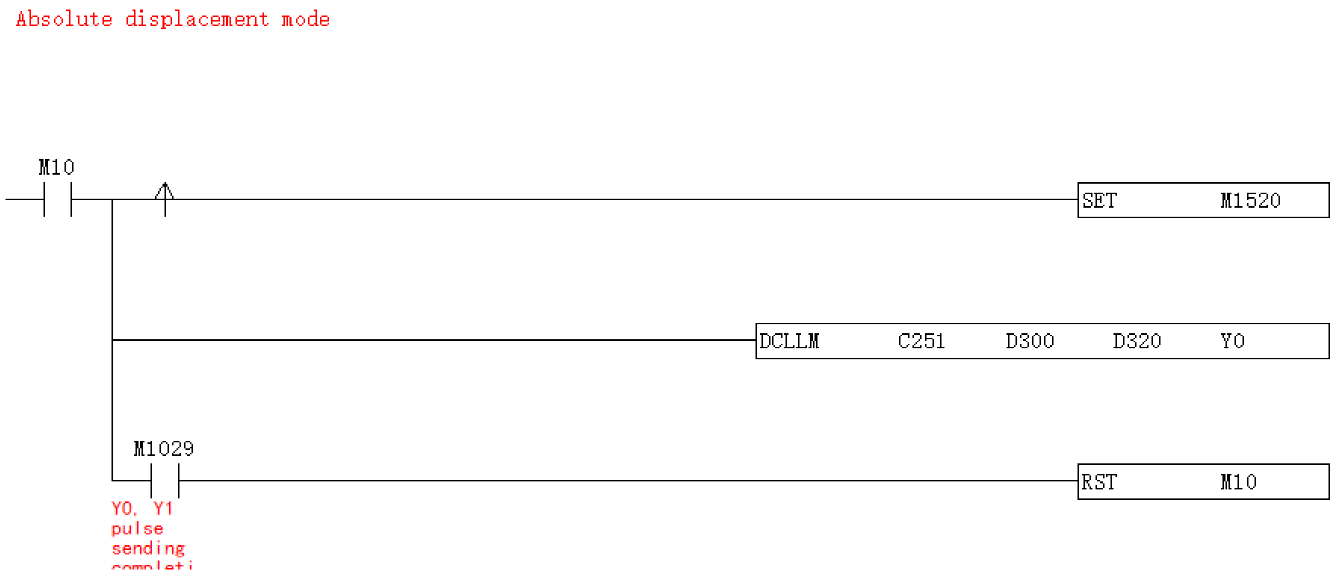
  3) 100ms is an appropriate value. Therefore, set the acceleration time as 250ms and deceleration time as 550ms to complete the design.

4、Program Example 2:

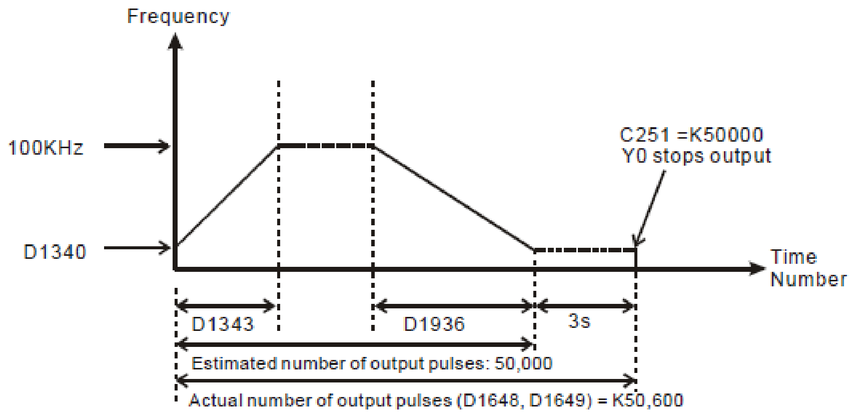
- Assume the feedback of the encoder is an A/B phase input and we adopt C251 timing (we suggust you clear it to 0 before the execution); target number of feedbacks = 50,000; target output frequency = 100kHz; Y0, Y1(CH0) as output pulses; start/end frequency (D1340) = 200Hz; acceleration time (D1343) = 300ms; deceleration time (D1936) = 600ms; precentage value (D1131) = 100; current number of output pulses (D1648, D1649) = 0.


M1002
ON only
for 1
scan
MOV K100 D1131
MOV K100 D1244
MOV K100 D1340
The 1st
step
start
MOV K100 D1343
Accelera
/Deceler
time of
MOV K0 D1696
Y6, Y7
sent
pulse
DMOV K0 D1648
Y0, Y1
sent
pulse
DMOV K0 C251


Relative displacement mode
M10
RST M1520
DCLLM C251 D300 D320 Y0
M1029
RST M10
Y0, Y1
pulse
sending



- Assume the first execution result as:



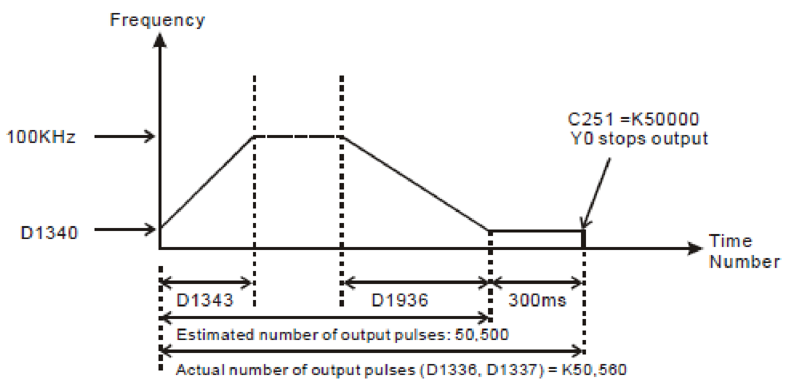
- Observe the result of the first execution:

  1) The actual output number 50,600 – estimated output number 50,000 = 600

  2) 600 x (1/200Hz) = 3s (idling time)

  3) 3 seconds are too long. Therefore, increase the percentage value (D1131) to K101.

- Obatin the result of the second execution:



- Observe the result of the second execution:

1) a) The actual output number 50,560 – estimated output number 50,500 = 60

2) b) 60 x (1/200Hz) = 300ms (idling time)

3) 300ms is an appropriate value. Therefore, set the percentage value (D1131) as K101 to complete the design.

ZL 198		VSPO	S1 S2 S3 D	Variable speed pulse output
	D			

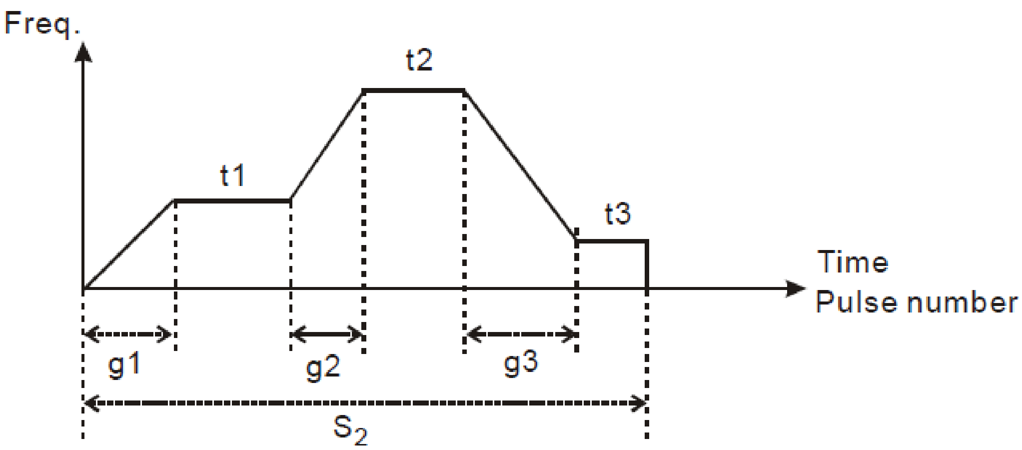
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1													*			DVSPO: 17 steps
S2					*	*							*			32-bit
S3					*	*							*			
D		*														

1、Operands:

- S1: Target frequency of output S2: Target number of pulses

  S3: Gap time and gap frequency D: Pulse output device (Y0, Y2, Y4, and Y6.)
- Max frequency for S1: 200kHz. Target frequency can be modified during the execution of instruction. When S1 is modified, VSPO will ramp up/down to the target frequency according to the ramp-up gap time and gap frequency set in S3.
- S2 target number of pulses is valid only when the instruction is executed first time. S2 can NOT be modified during the execution of instruction. S2 can be a negative value. When target number of pulses are specified with 0, PLC will perform continuous output and the special D shows the current value that is counting and going in the forward direction but that does NOT include any control over the output point direction.
- The gap frequency in S3+0 is in the range of 6Hz to 32767Hz, and the gap time in S3+1 is in the range of 1ms to 80ms. If a setting value exceeds the available range, the PLC will take the maximum or the minimum value.
- D pulse output device supports Y0, Y2, Y4 and Y6. Y1, Y3, Y5 and Y7 are corresponding output direction. The forward direction is On
- Parameters set in S3 can only be modified while modifying the value in S1. When target frequency is set as 0, PLC will ramp down to stop according to parameters set in S3. If target frequency other than 0 is specified again, pulse output will ramp up to target frequency and operates untill target number of pulses are completed
- Function Explanations::

  1) Pulse output diagram



a. Definitions:

t1→ target frequency of 1st shift

t2 → target frequency of 2nd shift

t3 → target frequency of 3rd shift

g1 → ramp-up time of 1st shift

g2 → ramp-up time of 2nd shift

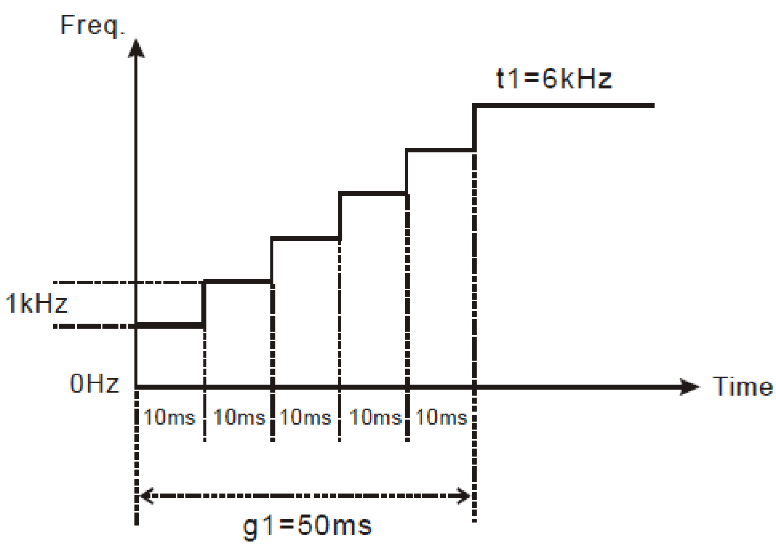
g3 → ramp-down time of 3rd shift

S2 → total output pulses

b. Explanations on each shift

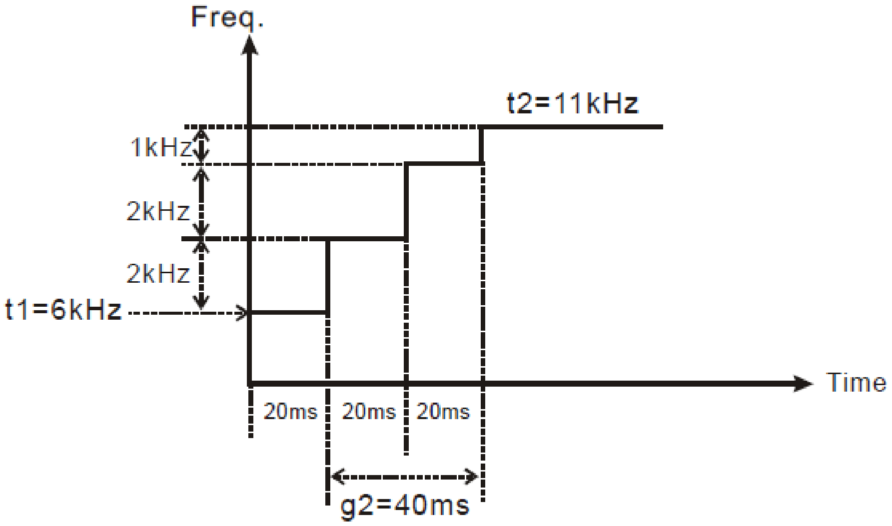
1、1st shift:

Assume t1 = 6kHz, gap freqency = 1kHz, gap time = 10ms Ramp-up steps of 1st shift:



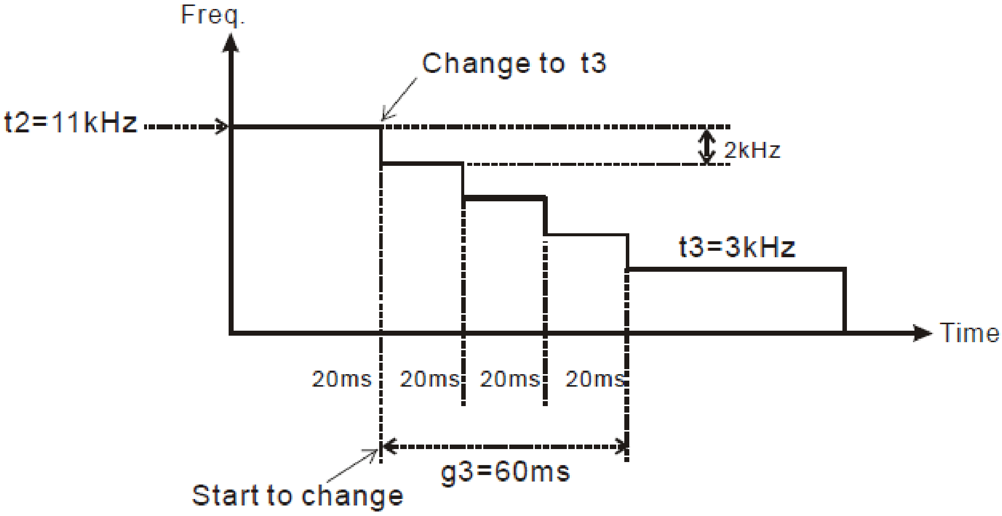
2、2nd shift:

Assume t2 = 11kHz, internal frequency = 2kHz, gap time = 20ms Ramp-up steps of 2nd shift:



3、3rd shift:

Assume t3 = 3kHz, gap frequency = 2kHz, gap time = 20ms Ramp-down steps of 3rd shift:



Points to note:

1. Associated flags:

   M1029: CH0 pulse output execution is completed

   M1030: CH1 pulse output execution is completed

   M1036: CH2 pulse output execution is completed

   M1037: CH3 pulse output execution is completed

10 **Application instructions** ZL200~ZL313

10.1 (ZL 202-203) Special function instructions

ZL 202	SCAL						S1 S2 S3 D								Proportional Value Calculation
	Bit Devices				Word Devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*							*			SCAL: 9 steps 16-bit
0					*	*							*			
S3					*	*							*			
D													*			
Use pay attention to the operands: Range of S1, S2, S3: -32,768 ~ 32,767 S2 operand unit of the input value is 0.001 Each device using range please refer to the functional specification table																

1、Explanations:

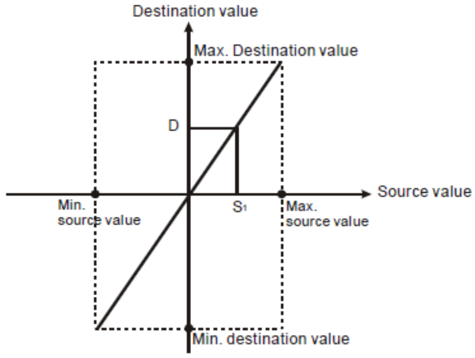
- To perform pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction
- S1: Source value S2: Slope S3: Offset D: Destination device.
- Operation equation in the instruction: D = (S1 × S2) ÷ 1,000 + S3.

  Users have to obtain S2 and S3 (decimals are rounded up into 16-bit integers) by using the slope and offset equations below.

  Slope equation: S2 = [(max. destination value − min. destination value) ÷ (max. source value − min. source value)] × 1,000.

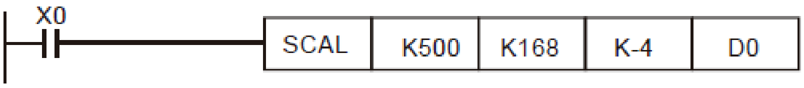
  Offset equation: S3 = min. destination value − min. source value × S2 ÷ 1,000

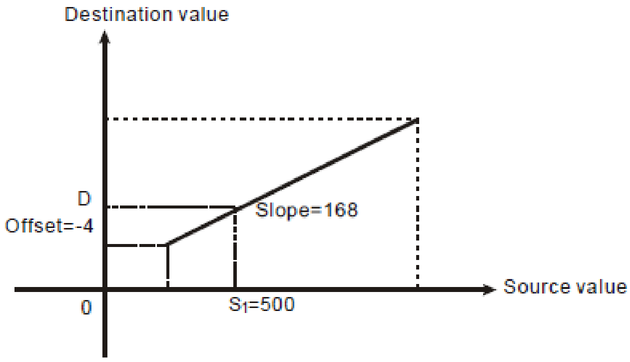
  4



2、Program Example 1:

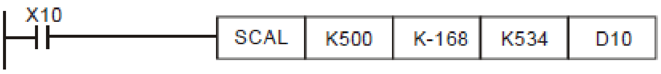
- Assume S1 = 500, S2 = 168, S3 = -4. When X0 = On, SCAL instruction will be executed and obtain the proportional value at D0.
- Equation: D0 = (500 × 168) ÷ 1,000 + (-4) = 80.

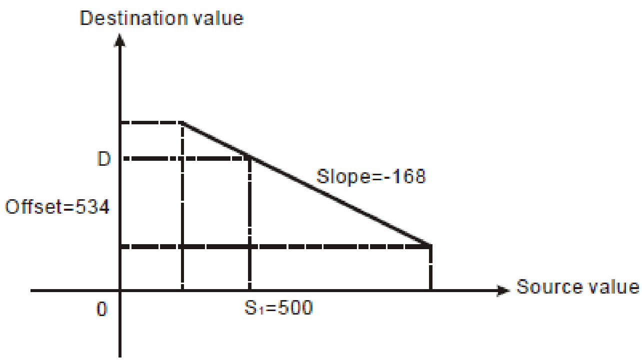




3、Program Example 2:

- Assume S1 = 500, S2 = -168, S3 = 534. When X10 = On, SCAL instruction will be executed and obtain the proportional value at D10.
- Equation: D0 = (500 × -168) ÷ 1,000 + 534 = 450





Remarks:

- This instruction is applicable for known slope and offset. If slope and offset are unknown, use SCLP instruction for the calculation.
- S2 has to be within the range -32,768 ~ 32,767. If S2 falls without the range, use SCLP instruction for the calculation.
- When using the slope equation, please be aware that the max. source value must > min. source value, but it is not necessary that max. destination value > min. destination value.
- If the value of D > 32,767, D = 32,767; if the value of D < -32,768, D = -32,768.

<table>
<tr><td>ZL 203</td><td></td><td rowspan="2" colspan="4">SCLP</td><td></td><td colspan="9">S1 S2 D</td><td>Parameter Proportional Value Calculation</td></tr>
<tr><td></td><td>D</td><td></td><td colspan="9"></td><td></td></tr>
<tr><td></td><td colspan="4">Bit Devices</td><td colspan="11">Word Devices</td><td rowspan="5">SCLP: 7 steps
DSCLP: 13 steps</td></tr>
<tr><td></td><td>X</td><td>Y</td><td>M</td><td>S</td><td>K</td><td>H</td><td>KnX</td><td>KnY</td><td>KnM</td><td>KnS</td><td>T</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td>S1</td><td></td><td></td><td></td><td></td><td>*</td><td>*</td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>S2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
<tr><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>*</td><td></td><td></td></tr>
</table>

1、Explanations:

- S1: Source value. S2: Parameter. D: Destination device
- Settings of S2 for 16-bit instruction:

  S2 occupies 4 consecutive devices in 16-bit instruction:

Device No.	Parameter	Range	
S2:	Maximum source value	Integer	Floating point
S2 +1	Minimum source value	-2,147,483,648~ 2,147,483,647	Range of 32-bit floating point
S2 +2	Maximum destination value		
S2 +3	Minimum destination value		

- Settings of S2 for 32-bit instruction:
- S2 occupies 8 consecutive devices in 32-bit instruction.

Device No.	Parameter	Range	
		Integer	Floating point
S2, S2 + 1	Maximum source value	-2,147,483,648 ~ 2,147,483,647	Range of 32-bit floating point
S2 + 2, 3	Minimum source value		
S2 + 4, 5	Maximum destination value		
S2 + 6, 7	Minimum destination value		

- Operation equation in the instruction: D = [(S1 − min. source value) × (max. destination value − min. destination value)] ÷ (max. source value − min. source value) + min. destination value.
- The operational relation between source value and destination value is as stated below:

y = kx + b

y= Destination value (D)

k= Slope = (max. destination value − min. destination value) ÷ (max. source value − min. source value)

b= Offset = Min. destination value − Min. source value × slope

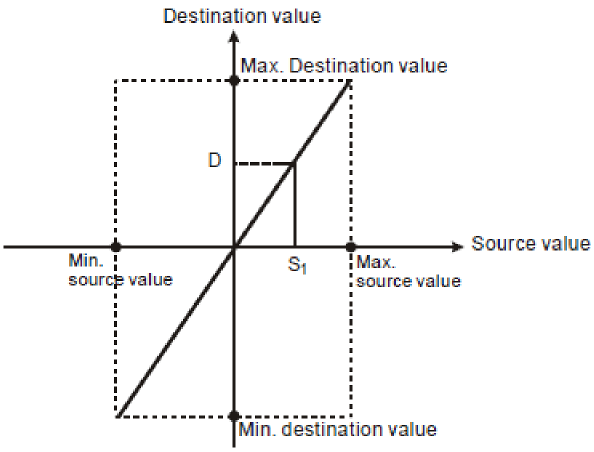
Bring all the parameters into equation y = kx + b and obtain the equation in the instruction:

y = kx + b = D = k S1 + b = slope ×S1 + offset = slope ×S1 + min. destination value − min. source value × slope = slope × (S1 − min. source value) + min. destination value = (S1 − min. source value) × (max. destination value − min. destination value) ÷ (max. source value − min. source value) + min. destination value.

- If S1 > max. source value, S1 = max. source value

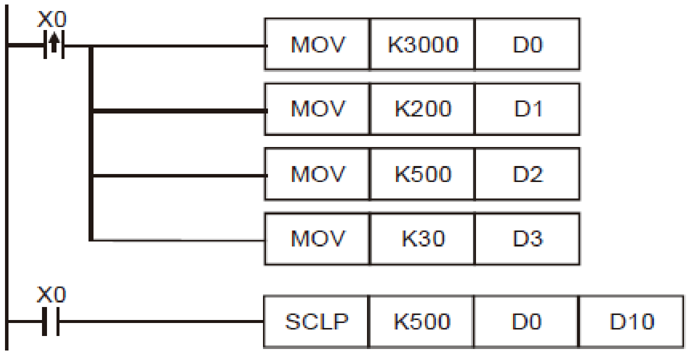
  If S1 < min. source value, S1 = min. source value

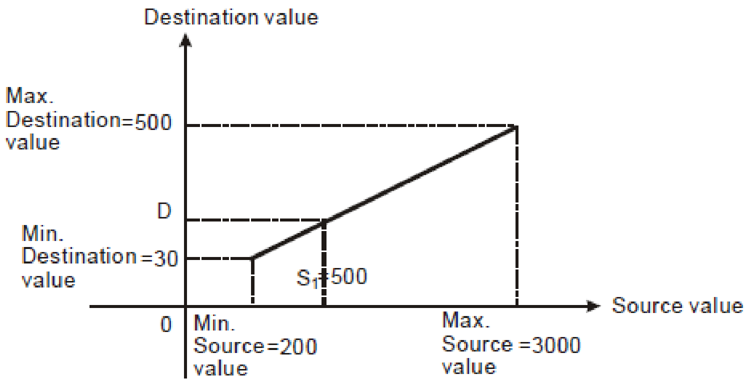
  When all the input values and parameters are set, the output curve is shown as the figure:



2、Program Example 1:

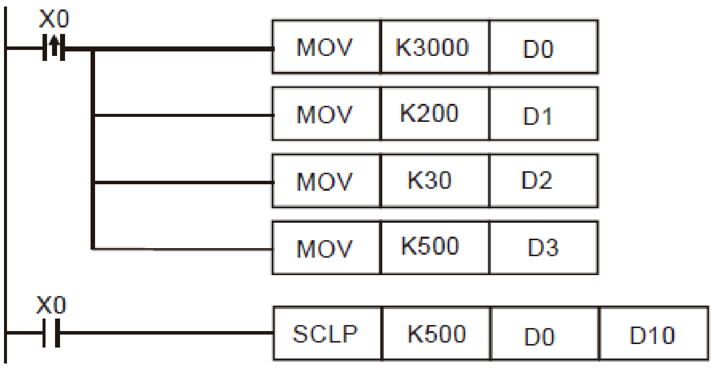
- Assume S1 = 500, max. source value D0 = 3,000, min. source value D1 = 200, max. destination value D2 = 500, and min. destination value D3 = 30. When X0 = On, SCLP instruction will be executed and obtain the proportional value at D10.
- Equation: D10 = [(500 − 200) × (500 − 30)] ÷ (3,000 − 200) + 30 = 80.35. Round off the result into an integer D10 = 80.

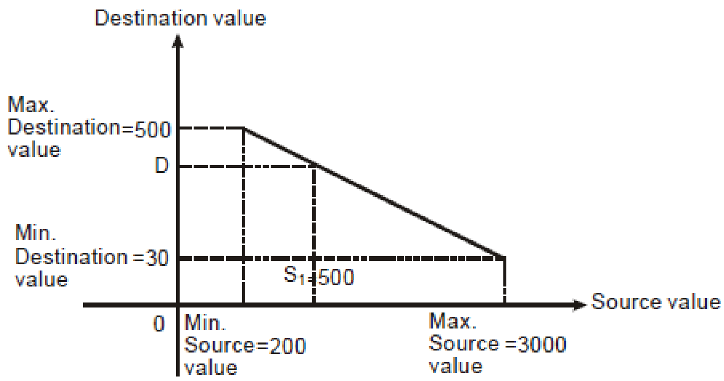




3、Program Example 2:

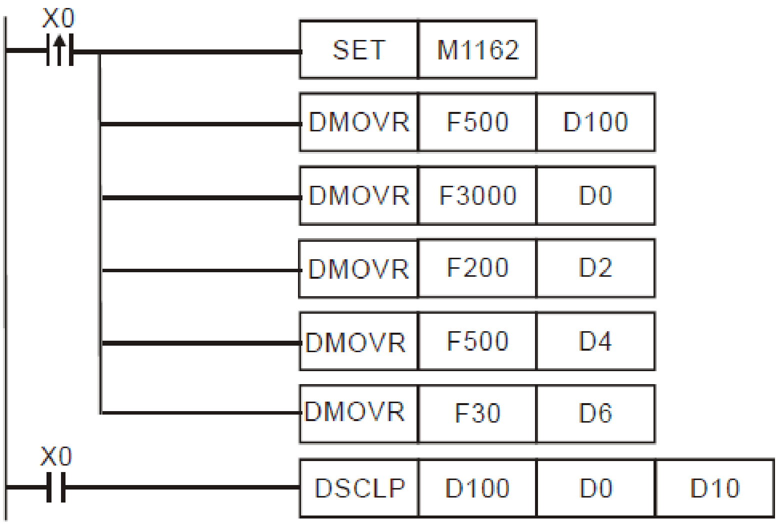
- Assume S1 = 500, max. source value D0 = 3,000, min. source value D1 = 200, max. destination value D2 = 30, and min. destination value D3 = 500. When X0 = On, SCLP instruction will be executed and obtain the proportional value at D10.
- Equation: D10 = [(500 − 200) × (30 − 500)] ÷ (3,000 − 200) + 500 = 449.64. Round off the result into an integer D10 = 450.

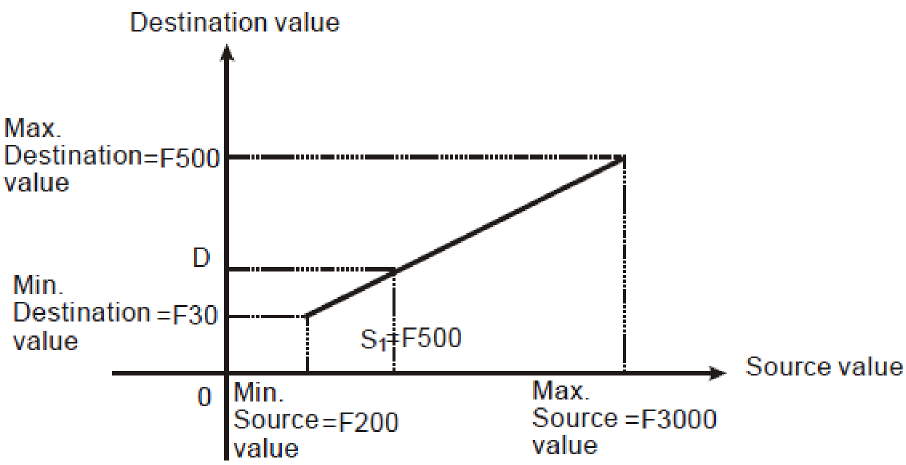




4、Program Example 3:

- Assume the source of S1 D100 = F500, max. source value D0 = F3000, min. source value D2 = F200, max. destination value D4 = F500, and min. destination value D6 = F30. When X0 = On, set up M1162, adopt floating point operation and execute DSCLP instruction. The proportional value will be obtained at D10.
- Equation: D10 = [(F500 － F200) × (F500 － F30)] ÷ (F3000 － F200) + F30 = F80.35. Round off the result into an integer D10 = F80.





Remarks:

- Range of S1 for 16-bit instruction: max. source value ⩾ S1 ⩾ min. source value; -32,768 ~ 32,767. If the value falls without the bounds, the bound value will be used for calculation.
- Range of integer S1 for 32-bit instruction: max. source value ⩾ S1 ⩾ min. source value; -2,147,483,648 ~ 2,147,483,647. If the value falls without the bounds, the bound value will be used for calculation.
- Range of floating point S1 for 32-bit instruction: max. source value ⩾ S1 ⩾ min. source value; following the range of 32- bit floating point. If the value falls without the bounds, the bound value will be used for calculation.
- Please be aware that the max. source value must > min. source value, but it is not necessary that max. destination value > min. destination value.

ZL 205	D	CMPT	P	S1 S2 n D	Compare table

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1											*	*	*			CMPT: 9 steps
S2											*	*	*			DCMPT: 17 steps
n					*	*							*			DCMPTP: 17 steps
D								*	*	*	*	*	*			

1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instruction.
- S1: Source device 1 S2: Source device 2

  n: Data length/function D: Destination device
- S1 and S2 can be T/C/D devices, for C devices only 16-bit devices are applicable (C0~C199).
- The high 16-bit value in the operand n used in the 32-bit instruction is an invalid value.
- The low 8-bit value in the operand n indicates the data length. The operand n used in the 16-bit instruction should be within the range between 1 and 16. The operand n used in the 32-bit instruction should be within the range between 1 and 32. PLC will take the upper/lower bound value if set value exceeds the available range.
- The high 8-bit value in the operand n indicates the comparison condition.

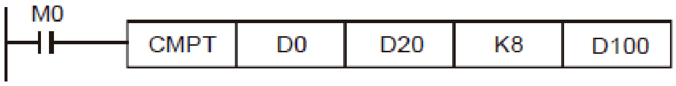
Value	K0	K1	K2	K3	K4
Comparison condition	S1 = S2	S1 < S2	S1 <= S2	S1 > S2	S1 >= S2

- If n used in the 16-bit instruction is set to H0108, it means that 8 pieces of data are compared to 8 pieces of data, and the "larger than" comparison is performed. If n used in the 32-bit instruction is set to H00000320, it means that 32 pieces data are compared to 32 pieces of data, and the "less than" comparison is performed.
- If the setting value for the comparison condition exceeds the range, or the firmware version does not support the comparison condition, the default "equal to" comparison is performed.
- The comparison values used in the 16-bit instruction are signed values. The comparison values used in the 32- bit instruction are 32-bit values (M1162=Off), or floating-point values (M1162=On).

- Data written in operand D will all be stored in 16-bit format or in 32-bit format. When data length is less than 16 or 32, the null bits are fixed as 0, e.g. if n = K8, bit 0~7 will be set according to compare results, and bit 8~15 will all be 0.
- If the comparison result meets the condition, the corresponding bit is set to 1. Otherwise, it is set to 0.

2、Program example:

- When M0 = ON, compare the 16-bit value in D0~D7 with D20~D27 and store the results in D100.



◆ Content in D0~D7:

number	D0	D1	D2	D3	D4	D5	D6	D7
value	K10	K20	K30	K40	K50	K60	K70	K80

◆ Content in D20~D27:

number	D20	D21	D22	D23	D24	D25	D26	D27
value	K12	K20	K33	K44	K50	K66	K70	K88

◆ After the comparison of CMPT instruction, the associated bit will be 1 if two devices have the same value, and other bits will all be 0. Therefore the results in D100 will be as below:

<table>
<tr><td rowspan="3">D100</td><td>Bit0</td><td>Bit1</td><td>Bit02</td><td>Bit03</td><td>Bit04</td><td>Bit05</td><td>Bit06</td><td>Bit07</td><td>Bit8~15</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0…0</td></tr>
<tr><td colspan="9">H0052 (K82)</td></tr>
</table>

ZL 207	CSFO	S S1 S	Catch speed and proportional output

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CSFO: 7 steps 16-bit
S	*															
S1													*			
D													*			

1、Explanations:

- To perform a pulse type, queue by adding an NP rising edge "↑" instruction to the front of the instructio n.
- S: Source device of signal input (Only X0 and X1 are available)

  S1: Sample time setting and the input speed information

  D: Output proportion setting and output speed information
- When S specifies X0, PLC only uses X0 input point and its associated high speed pulse output: Y0, in this case Y1 is normal output point. When S specifies X1, PLC uses X0 (A phase) and X1 (B phase) input points and their associated output: Y0 (Pulse) / Y1 (Dir).
- If S specifies X1 with 2-phase 2 inputs, the counting mode is fixed as quadruple frequency.
- During pulse output process of Y0, special registers (D1649, D1648) storing the current number of output pulses will be updated when program scan proceeds to this instruction.
- S1 occupies consecutive 4 16-bit registers. S1 +0 specifies the sampling times, i.e. when S1 +0 specifies K1, PLC catches the speed every time when 1 pulse is outputted. Valid range for S1 +0 in 1-phase 1-input mode: K1~K100, and 2-phase 2-input mode: K2~K100. If the specified value exceeds the valid range, PLC will take the lower/upper bound value as the set value. Sample time can be changed during PLC operation, however the modified value will take effect until program scan proceeds to this instruction. S1+1 indicates the latest speed sampled by PLC (Read-only). Unit: 1Hz. Valid range: ±10kHz. S1+2 and S1+3 indicate the accumulated number of pulses in 32-bit data (Read-only).
- D occupies 3 consecutive 16-bit registers. D +0 specifies the output proportion value. Valid range: K1 (1%) ~ K10000 (10000%). If the specified value exceeds the valid range, PLC will take the lower/upper bound value as the set value. Output proportion can be changed during PLC operation, however the modified value will take effect until program scan proceeds to this instruction. D+2 and D+1 indicates the

output speed in 32-bit data. Unit: 1Hz. Valid range: ±200kHz. When selecting the MPG mode, it takes one more 16-bit register. D+3 indicates the pulse output channel, ranging from K0 to K3, indicating the output channels CH0~CH3. When selecting the general mode, the pulse output channel is fixed to CH0. Note: if you need to change the mode from the MPG mode to the general mode or vise versa, you need to close the instruction and re-execute the instruction to ensure the channel switching can be normally done。

- The pulse output channel selecting: when S input point uses X0 as the source, the corresponding pulse output points are Y0, Y2, Y4, Y6 and the general pulse output points are Y1, Y3, Y5, Y7. When S input point uses X1 as the source, the corresponding output points are Y0(Pulse) / Y1(Dir) or Y2(Pulse) / Y3(Dir) or Y4(Pulse) /Y5(Dir) or Y6(Pulse) / Y7(Dir) high speed output.
- The speed sampled by the PLC will be multiplied with the output proportion D+0, then the PLC will generate the actual output speed. The PLC will take the integer of the calculated value, i.e. if the calculated result is smaller than 1Hz, the PLC will output with 0Hz. For example, input speed: 10Hz, output proportion: K5 (5%), then the calculation result will be 10 x 0.05 = 0.5Hz. Pulse output will be 0Hz; if output proportion is modified as K15 (15%), then the calculation result will be 10 x 0.15 = 1.5Hz. Pulse output will be 1Hz.

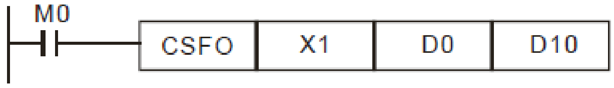
2、Program Example:

- If D0 is set as K2, D10 is set as K100:

  When the sampled speed on (X0, X1) is +10Hz (D1 = K10), (Y0, Y1) will output pulses with +10Hz (D12, D11 = K10); When the sampled speed is -10Hz (D1 = K-10), (Y0, Y1) will output pulses with -10Hz (D12, D11 = K-10).
- If D0 is set as K2, D10 is set as K1000:

  When the sampled speed on (X0, X1) is +10Hz (D1 = K10), (Y0, Y1) will output pulses with +100Hz (D12, D11 = K100); When the sampled speed is -100Hz (D1 = K-100), (Y0, Y1) will output pulses with -100Hz (D12, D11 =K-100).
- If D0 is set as K10, D10 is set as K10:

  When the sampled speed on (X0, X1) is +10Hz (D1 = K10), (Y0, Y1) will output pulses with +1Hz (D12, D11 = K1); When the sampled speed is -10Hz (D1 = K-10), (Y0, Y1) will output pulses with -1Hz (D12, D11 = K-1) .



10.2 (ZL 215-223) Contact type logic operation instructions

ZL 215~217	D	LD#	S1 S2	Contact Logical Operation LD#

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	LD#: 5 steps 16-bit
S1					*	*	*	*	*	*	*	*	*	*	*	DLD#: 9 steps 32-bit
S2					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

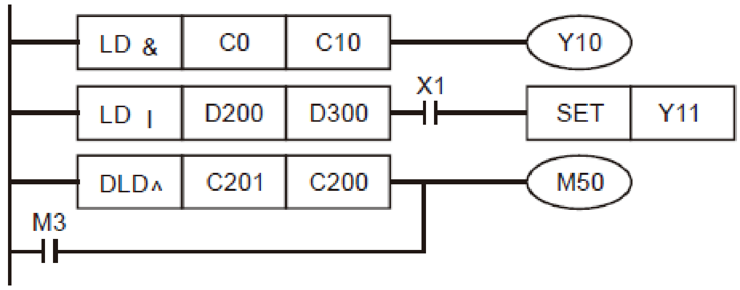
- S1: Data source device 1 S2: Data source device 2
- This instruction compares the content in S1 and S2. If the result is not "0", the continuity of the instruction is enabled. If the result is "0", the continuity of the instruction is disabled.
- LD# instruction is used for direct connection with BUS

ZL No.	16-bit instruction	32-bit instruction	Continuity condition	No-continuity condition
215	LD&	DLD&	S1 & S2≠ 0	S1 & S2 = 0
216	LD|	DLD|	S1 | S2≠ 0	S1 | S2= 0
217	LD^	DLD^	S1 ^ S2≠ 0	S1 ^ S2= 0

- &: Logical "AND" operation.
- |: Logical "OR" operation.
- ^: Logical "XOR" operation
- When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DLD#). If 16-bit instructions (LD#) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

2、Program Example:

- When the result of logical AND operation of C0 and C10 ≠ 0, Y10 = On.
- When the result of logical OR operation of D200 and D300 ≠ 0 and X1 = On, Y11 = On will be retained.
- When the result of logical XOR operation of C201 and C200 ≠ 0 or M3 = On, M50 = On.



ZL 218~220	D	AND#	S1 S2													Contact Logical Operation AND#
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	AND#: 5 steps 16-bit
S1					*	*	*	*	*	*	*	*	*	*	*	DAND#: 9 steps 32-bit
S2					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

- S1: Data source device 1 S2: Data source device 2
- This instruction compares the content in S1 and S2. If the result is not "0", the continuity of the instruction is enabled. If the result is "0", the continuity of the instruction is disabled.
- AND# is an operation instruction used on series contacts.

ZL No.	16-bit instruction	32-bit instruction	Continuity condition	No-continuity condition
218	AND&	DAND&	S1 & S2≠ 0	S1 & S2 = 0
219	AND |	DAND |	S1 | S2≠ 0	S1 | S2= 0
220	AND^	DAND^	S1 ^ S2≠ 0	S1 ^ S2= 0

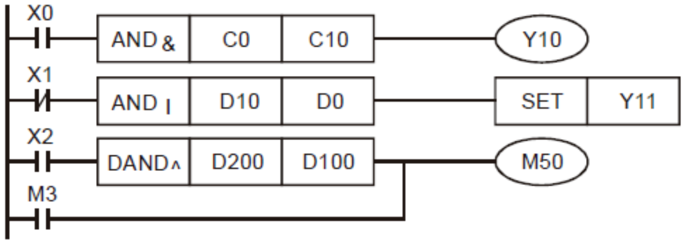
- &: Logical "AND" operation
- |: Logical "OR" operation
- ^: Logical "XOR" operation.
- When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DAND#). If 16-bit instructions (AND#) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

2、Program Example:

- When X0 = On and the result of logical AND operation of C0 and C10 ≠ 0, Y10 = On.
- When X1 = Off and the result of logical OR operation of D10 and D0 ≠ 0 and X1 = On, Y11 = On will be

retained.

- When X2 = On and the result of logical XOR operation of 32-bit register D200 (D201) and 32-bit register D100 (D101) ≠ 0 or M3 = On, M50 = On.



ZL 221~223	D	OR#	S1 S2	Contact Logical operation OR#

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	OR#: 5 steps
S1					*	*	*	*	*	*	*	*	*	*	*	DOR#: 9 steps
S2					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

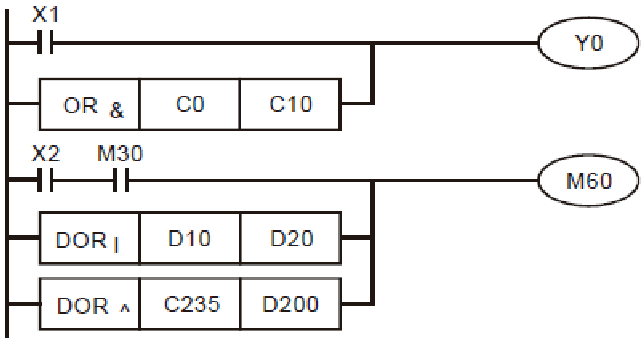
- S1: Data source device 1　　S2: Data source device 2
- This instruction compares the content in S1 and S2. If the result is not “0”, the continuity of the instruction is enabled. If the result is “0”, the continuity of the instruction is disabled.
- OR# is an operation instruction used on parallel contacts.

ZL No.	16-bit instruction	32-bit instruction	Continuity condition	No-continuity condition
221	OR&	DOR&	S1 & S2≠ 0	S1 & S2 = 0
222	OR |	DOR |	S1 | S2≠ 0	S1 | S2= 0
223	OR^	DOR^	S1 ^ S2≠ 0	S1 ^ S2= 0

- &: Logical “AND” operation.
- |: Logical “OR” operation.
- ^: Logical “XOR” operation.
- When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DOR#). If 16-bit instructions (OR#) is adopted, a “program error” will occur and the ERROR indicator on the MPU panel will flash.

2、Program Example:

- When X1 = On and the result of logical AND operation of C0 and C10 ≠ 0, Y10 = On.
- M60 will be On when X2 = On and M30 = On, or the result of logical OR operation of 32-bit register D10 (D11) and 32-bit register D20 (D21) ≠ 0, or the result of logical XOR operation of 32-bit register D200 (D201) and 32- bit counter C235 ≠ 0.


X1
Y0
OR &
C0
C10
X2
M30
M60
DOR |
D10
D20
DOR ^
C235
D200

10.3 (ZL 224-246) Contact type comparison command

ZL 224~230	D	LD※		S1 S2	Load Compare

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	LD※: 5 steps 16-bit
S1					*	*	*	*	*	*	*	*	*	*	*	DLD※: 9 steps 32-bit
S2					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

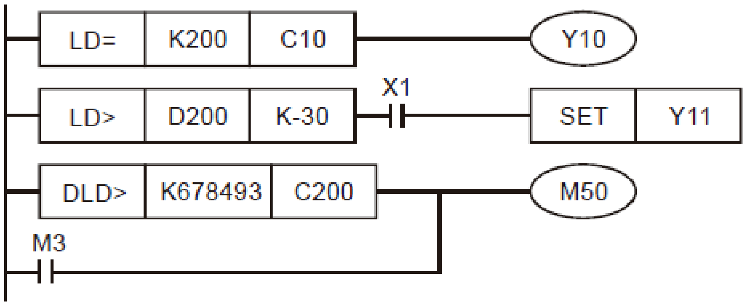
- S1: Data source device 1 S2: Data source device 2.
- This instruction compares the content in S1 and S2. Take API224 (LD=) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
- LD※ instruction is used for direct connection with BUS.

ZL No.	16-bit instruction	32-bit instruction	Continuity condition	No-continuity condition
224	LD=	DLD=	S1 = S2	S1 ≠ S2
225	LD>	DLD>	S1 > S2	S1 ≦ S2
226	LD<	DLD<	S1 < S2	S1 ≧ S2
228	LD<>	DLD<>	S1 ≠ S2	S1 = S2
229	LD<=	DLD<=	S1 ≦ S2	S1 > S2
230	LD>=	DLD>=	S1 ≧ S2	S1 < S2

- When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DLD※). If 16-bit instructions (LD※) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

2、Program Example:

- When the content in C10 = K200, Y10 = On.
- When the content in D200 > K-30 and X1 = On, Y11= On will be retained.
- When the content in C200 < K678,493 or M3 = On, M50 = On.


LD=
K200
C10
Y10
LD>
D200
K-30
X1
SET
Y11
DLD>
K678493
C200
M50
M3

ZL 232~238	D	**AND※**	S1 S2	AND Compare

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	AND※: 5 steps
S1					*	*	*	*	*	*	*	*	*	*	*	DAND※: 9 steps
S2					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

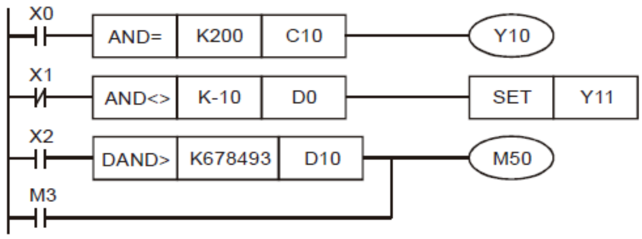
- S1: Data source device 1 S2: Data source device 2
- This instruction compares the content in S1 and S2. Take API232 (AND=) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
- AND※ is a comparison instruction is used on series contacts.

ZL No.	16-bit instruction	32-bit instruction	Continuity condition	No-continuity condition
232	AND=	DAND=	S1 = S2	S1 ≠ S2
233	AND>	DAND>	S1 > S2	S1 ≦ S2
234	AND<	DAND<	S1 < S2	S1 ≧ S2
236	AND<>	DAND<>	S1 ≠ S2	S1 = S2
237	AND<=	DAND<=	S1 ≦ S2	S1 > S2
238	AND>=	DAND>=	S1 ≧ S2	S1 < S2

- When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DAND※). If 16-bit instructions (AND※) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

2、Program Example:

- When X0 = On and the content in C10 = K200, Y10 = On.
- When X1 = Off and the content in D0 ≠ K-10, Y11= On will be retained.
- When X2 = On and the content in 32-bit register D0 (D11) < 678,493 or M3 = On, M50 = On.


X0
AND=
K200
C10
Y10
X1
AND<>
K-10
D0
SET
Y11
X2
DAND>
K678493
D10
M50
M3

ZL 240~246	D	OR※	S1 S2	OR Compare

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1					*	*	*	*	*	*	*	*	*	*	*	OR*: 5 steps 16-bit
S2					*	*	*	*	*	*	*	*	*	*	*	DOR*: 9 steps 32-bit

1、Explanations:

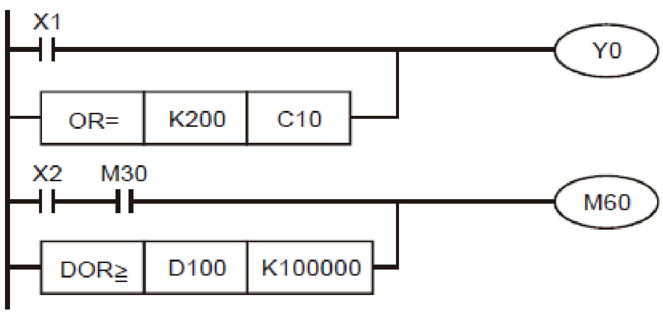
- S1: Data source device 1 S2: Data source device 2
- This instruction compares the content in S1 and S2. Take API240 (OR=) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
- OR※ is an comparison instruction used on parallel contacts.

ZL No.	16-bit instruction	32-bit instruction	Continuity condition	No-continuity condition
240	OR=	DOR=	S1 = S2	S1 ≠ S2
241	OR>	DOR>	S1 > S2	S1 ≦ S2
242	OR<	DOR<	S1 < S2	S1 ≧ S2
244	OR<>	DOR<>	S1 ≠ S2	S1 = S2
245	OR<=	DOR<=	S1 ≦ S2	S1 > S2
246	OR>=	DOR>=	S1 ≧ S2	S1 < S2

- When 32-bit counters (C200 ~ C255) are used in this instruction for comparison, make sure to adopt 32-bit instruction (DOR※). If 16-bit instructions (OR※) is adopted, a "program error" will occur and the ERROR indicator on the MPU panel will flash.

2、Program Example:

- When X1 = On and the present value of C10 = K200, Y0 = On.
- M60 will be On when X2 = On, M30 = On and the content in 32-bit register D100 (D101) ≥ K100,000.



10.4 (ZL 266-274) Character device bit command

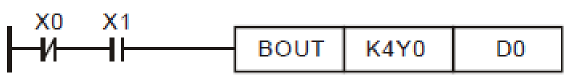
ZL 266		BOUT	D n													Output Specified Bit of a Word
	D															
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BOUT: 5 steps 16-bit
D								*	*	*	*	*	*	*	*	DBOUT: 9 steps 32-bit
n					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

- D: Destination output device n: Device specifying the output bit..
- BOUT instruction performs bit output on the output device according to the value specified by operand n. Status of Coils and Associated Contacts:

Evaluation result	BOUT instruction		
	Coil	Associated Contacts	
		NO contact (normally open)	NC contact (normally closed)
FALSE	Off	Current blocked	Current flows
TRUE	On	Current flows	Current blocked

2 Program Example:



Instruction:			Operation:
LDI	X0		Load NC contact X0
AND	X1		Connect NO contact X1 in series.
BOUT	K4Y0	D0	When D0 = k1, executes output on Y1 When D0 = k2, executes output on Y2

ZL 267		BSET		D n												Set ON Specified Bit of a Word
	D															
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BSET: 5 steps 16-bit
D								*	*	*	*	*	*			DBSET: 9 steps 32-bit
n					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

- D: Destination device to be Set ON n: Device specifying the bit to be Set ON
- When BSET instruction executes, the output device specified by operand n will be On and latched. To reset the On state of the device, BRST instruction is required.

2、Program Example:

Instruction:			Operation:
LDI	X0		Load NC contact X0
AND	X1		Connect NO contact X1 in series.
BSET	K4Y0	D0	When D0 = k1, Y1 is ON and latched When D0 = k2, Y2 = ON and latched

ZL 268		BRST		D n	Reset Specified Bit of a Word
	D				

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BRST: 5 steps 16-bit
D								*	*	*	*	*	*			DBRST: 9 steps 32-bit
n					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

- D: Destination device to be reset n: Device specifying the bit to be reset
- When BRST instruction executes, the output device specified by operand n will be reset (OFF).

2、Program Example:

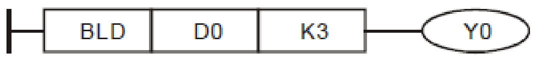
Instruction:			Operation:
LD	X0		Load NO contact X0
BRST	K4Y0	D0	When D0 = k1,
			Y1 is OFF
			When D0 = k2,
			Y2 = OFF

ZL 269	D	BLD						S n								Load NO Contact by Specified Bit
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BLD: 5 steps 16-bit
D								*	*	*	*	*	*			DBLD: 9 steps 32-bit
n					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

- S: Reference source device n: Reference bit
- Available range for the value in operand n: K0~K15 for 16-bit instruction; K0~K31 for 32-bit instruction.
- BLD instruction is used to load NO contact whose contact state is defined by the reference bit n in reference device D, i.e. if the bit specified by n is ON, the NO contact will be ON, and vice versa.

2、Program Example:



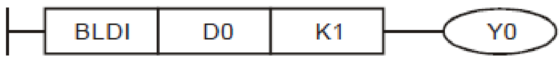
Instruction:			Operation:
BLD	D0	K3	Load NO contact X0 with bit Status of bit3 in D0
OUT	Y0		Device coil Y0

ZL 270		BLDI					S n									Load NC Contact by Specified Bit
	D															
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BLDI: 5 steps 16-bit
D								*	*	*	*	*	*			DBLDI: 9 steps 32-bit
n					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

- S: Reference source device n: Reference bit
- BLD instruction is used to load NC contact whose contact state is defined by the reference bit n in reference device D, i.e. if the bit specified by n is ON, the NC contact will be ON, and vice versa.

2、Program Example:



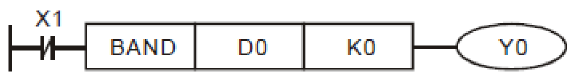
Instruction:			Operation:
BLDI	D0	K1	Load NC contact with bit status of bit1 in D0
OUT	Y0		Drive coil Y0

ZL 271	D	BAND		S n												Connect NO Contact in Series by Specified Bit
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BAND: 5 steps 16-bit
D								*	*	*	*	*	*			DBAND: 9 steps 32-bit
n					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

- S: Reference source device n: Reference bit
- BAND instruction is used to connect NO contact in series. The current state of the contact which is connected in series is read, and then the logical AND operation is performed on the current state and the previous logical operation result. The final result is stored in the accumulative register.

2、Program Example:



Instruction:			Operation:
LDI	X1		Load NC contact X1
BAND	D0	K0	Connect NO contact in series, whose state is defined by bit0 of D0
OUT	Y0		Drive coil Y0

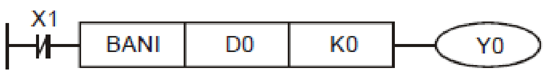
ZL 272		BANI	S n	Connect NC Contact in Series by Specified Bit
	D			

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BANI: 5 steps 16-bit
D								*	*	*	*	*	*			DBANI: 9 steps 32-bit
n					*	*	*	*	*	*	*	*	*	*	*	

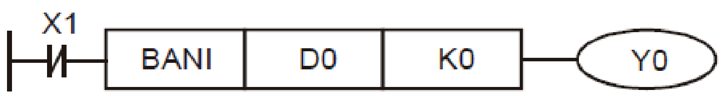
1、Explanations:

- S: Reference source device n: Reference bit
- BANI instruction is used to connect NC contact in series. The current state of the contact which is connected in series is read, and then the logical AND operation is performed on the current state and the previous logical operation result. The final result is stored in the accumulative register.

2、Program Example:



Instruction:			Operation:
LDI	X1		Load NC contact X1
BANI	D0	K0	Connect NC contact in series, whose state is defined by bit0 of D0
OUT	Y0		Drive coil Y0



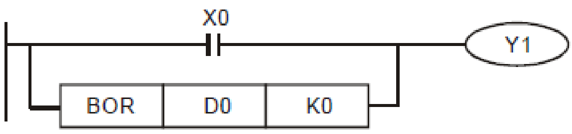
指令:			操作说明:
LDI	X1		载入 X1 的 B 接点
BANI	**D0**	**K0**	**串联 D0 Bit0 的 B 接点**
OUT	Y0		驱动 Y0 线圈

ZL 273	D	BOR					S n									Connect NO Contact in Parallel by Specified Bit
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BOR: 5 steps 16-bit
D								*	*	*	*	*	*			DBOR: 9 steps 32-bit
n					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

- S: Reference source device n: Reference bit
- BOR instruction is used to connect NO contact in parallel. The current state of the contact which is connected in series is read, and then the logical OR operation is performed on the current state and the previous logical operation result. The final result is stored in the accumulative register.

2、Program Example:



Instruction:			Operation:
LD	X0		Load NO contact X0
BOR	D0	K0	Connect NO contact in parallel, whose state is defined by bit0 of D0
OUT	Y1		Drive coil Y1

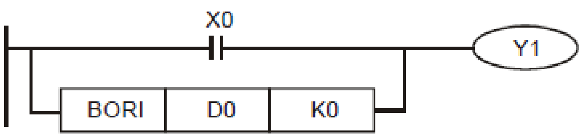
ZL 274	D	BORI	S n	Connect NC Contact in Parallel by Specified Bit

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BORI: 5 steps 16-bit
D								*	*	*	*	*	*			DBORI: 9 steps 32-bit
n					*	*	*	*	*	*	*	*	*	*	*	

1、Explanations:

- S: Reference source device n: Reference bit
- BORI instruction is used to connect NC contact in parallel. The current state of the contact which is connected in series is read, and then the logical OR operation is performed on the current state and the previous logical operation result. The final result is stored in the accumulative register.

2、Program Example:



Instruction:			Operation:
LD	X0		Load NO contact X0
BORI	D0	K0	Connect NC contact in parallel, whose state is defined by bit0 of D0
OUT	Y1		Drive coil Y1

10.5 (ZL 275-313) Floating point contact type command

ZL 275~280		FLD※				S1 S2										Floating Point Contact Type Comparison
	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FLD※: 9 steps
S1											*	*	*			32-bit
S2											*	*	*			

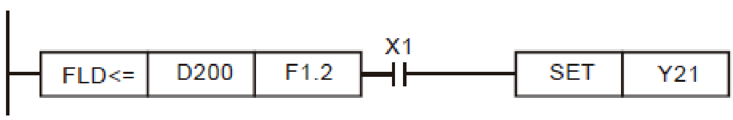
1、Explanations:

- S1: Source device 1 S2: Source device 2
- This instruction compares the content in S1 and S2. Take API275 (FLD=) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
- The user can specify the floating point value directly into operands S1 and S2 (e.g. F1.2) or store the floating point value in D registers for further operation.
- FLD※ (※: =, >, <, <>, ⩽, ⩾) instruction is used for direct connection with left hand bus bar.

ZL No.	32-bit instruction	Continuity condition	Discontinuity condition
275	FLD =	S1 = S2	S1 ≠ S2
276	FLD >	S1 > S2	S1 ≦ S2
277	FLD <	S1 < S2	S1 ≧ S2
278	FLD < >	S1 ≠ S2	S1 = S2
279	FLD < =	S1 ≦ S2	S1 > S2
280	FLD > =	S1 ≧ S2	S1 < S2

2、Program Example:

- When the content in D200 (D201) ⩽ F1.2 and X1 is ON, Y21 = ON and latched.



ZL 281~286	FAND※		S1 S2													Floating Point Serial Type Comparison
	Bit Devices				Word Devices											FAND※: 9 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S1											*	*	*			
S2											*	*	*			

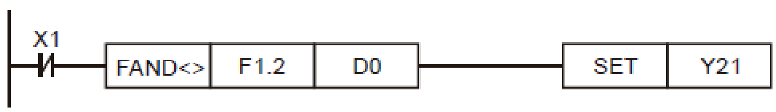
1、Explanations:

- S1: Source device 1   S2: Source device 2
- This instruction compares the content in S1 and S2. Take API281 (FAND =) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
- The user can specify the floating point value directly into operands S1 and S2 (e.g. F1.2) or store the floating point value in D registers for further operation.
- FAND※ (※: =, >, <, <>, ⩽, ⩾) instruction is used for serial connection with contacts.

ZL No.	32-bit instruction	Continuity condition	Discontinuity condition
281	FAND =	S1 = S2	S1 ≠ S2
282	FAND >	S1 > S2	S1 ≦ S2
283	FAND <	S1 < S2	S1 ≧ S2
284	FAND < >	S1 ≠ S2	S1 = S2
285	FAND < =	S1 ≦ S2	S1 > S2
286	FAND > =	S1 ≧ S2	S1 < S2

2、Program Example

- When X1 is OFF and the content in D0 (D1) does not equal to F1.2, Y21 = ON and latched.



ZL 287~292	FOR※	S1 S2	Floating Point Parallel Type Comparison

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FOR※: 9 steps
S1											*	*	*			32-bit
S2											*	*	*			

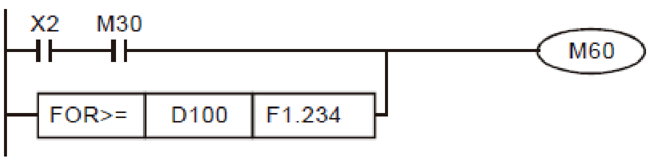
1、Explanations:

- S1: Source device 1　　S2: Source device 2
- This instruction compares the content in S1 and S2. Take API287 (FOR =) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
- The user can specify the floating point value directly into operands S1 and S2 (e.g. F1.2) or store the floating point value in D registers for further operation.
- OR※ (※: =, >, <, <>, ⩽, ⩾) instruction is used for parallel connection with contacts.

ZL No.	32-bit instruction	Continuity condition	Discontinuity condition
287	FOR =	S1 = S2	S1 ≠ S2
288	FOR >	S1 > S2	S1 ≦ S2
289	FOR <	S1 < S2	S1 ≧ S2
290	FOR < >	S1 ≠ S2	S1 = S2
291	FOR < =	S1 ≦ S2	S1 > S2
292	FOR > =	S1 ≧ S2	S1 < S2

2、Program Example:

- When both X2 and M30 are OFF and the content in D100 (D101) ⩾ F1.234, M60 = ON.



ZL 296~301	D	LDZ※	S1 S2 S3	Comparing contact type absolute values LDZ※

	Bit Devices				Word Devices											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	LDZ※: 7 steps 16-bit DLDZ※: 13 steps 32-bit
S1					*	*	*	*	*	*	*	*	*			
S2					*	*	*	*	*	*	*	*	*			
A3					*	*	*	*	*	*	*	*	*			

1、Explanations:

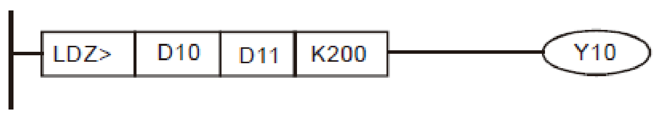
- S1: Source device 1　　S2: Source device 2　　S3: Source device 3
- The absolute value of the difference between S1 and S2 is compared with the absolute value of S3. Take LDZ> for example. If the comparison result is that the absolute value of the difference between S1 and S2 is greater than the absolute value of S3, the condition of the instruction is met. If the comparison result is that the absolute value of the difference between S1 and S2 is less than or equal to the absolute value of S3, the condition of the instruction is not met.
- The instruction can be connected to a busbar:

ZL No.	16-bit instruction	32-bit instruction	On	Off
296	LDZ>	DLDZ>	$\lvert S1-S2 \rvert > \lvert S3 \rvert$	$\lvert S1-S2 \rvert \leqq \lvert S3 \rvert$
297	LDZ>=	DLDZ>=	$\lvert S1-S2 \rvert \geqq \lvert S3 \rvert$	$\lvert S1-S2 \rvert < \lvert S3 \rvert$
298	LDZ<	DLDZ<	$\lvert S1-S2 \rvert < \lvert S3 \rvert$	$\lvert S1-S2 \rvert \geqq \lvert S3 \rvert$
299	LDZ<=	DLDZ<=	$\lvert S1-S2 \rvert \leqq \lvert S3 \rvert$	$\lvert S1-S2 \rvert > \lvert S3 \rvert$
300	LDZ =	DLDZ =	$\lvert S1-S2 \rvert = \lvert S3 \rvert$	$\lvert S1-S2 \rvert \neq \lvert S3 \rvert$
301	LDZ<>	DLDZ<>	$\lvert S1-S2 \rvert \neq \lvert S3 \rvert$	$\lvert S1-S2 \rvert = \lvert S3 \rvert$

- A 32-bit counter (C200~C255) must be used with the 32-bit instruction DLDZ※. If it is used with the 16-bit instruction LDZ※, a program error will occur, and the ERROR LED indicator on the PLC will blink.

2、Program Example:

- If the absolute value of the difference between D10 and D11 is greater than K200, Y0 will be On. If the absolute value of the difference between D10 and D11 is less than or equal to K200, Y0 will be Off.



ZL 302~307	D	ANDZ※		S1 S2 S3											Comparing contact type absolute values ANDZ※
	Bit Devices				Word Devices										ANDZ※: 7 steps 16-bit DANDZ※: 13 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*		
S2					*	*	*	*	*	*	*	*	*		
S3					*	*	*	*	*	*	*	*	*		

1、Explanations:

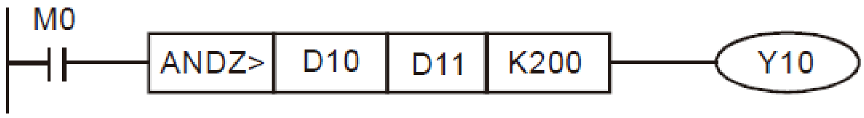
- S1: Source device 1 S2: Source device 2 S3: Source device 3
- The absolute value of the difference between S1 and S2 is compared with the absolute value of S3. Take ANDZ> for example. If the comparison result is that the absolute value of the difference between S1 and S2 is greater than the absolute value of S3, the condition of the instruction is met. If the comparison result is that the absolute value of the difference between S1 and S2 is less than or equal to the absolute value of S3, the condition of the instruction is not met.
- The instruction ANDZ※ is connected to a contact in series.

ZL No.	16-bit instruction	32-bit instruction	On	Off
302	ANDZ>	DANDZ>	|S1- S2| > |S3|	|S1- S2| ≦ |S3|
303	ANDZ>=	DANDZ>=	|S1- S2| ≧ |S3|	|S1- S2| < |S3|
304	ANDZ<	DANDZ<	|S1- S2| < |S3|	|S1- S2| ≧ |S3|
305	ANDZ<=	DANDZ<=	|S1- S2| ≦ |S3|	|S1- S2| > |S3|
306	ANDZ =	DANDZ =	|S1- S2| = |S3|	|S1- S2| ≠ |S3|
307	ANDZ<>	DANDZ<>	|S1- S2| ≠ |S3|	|S1- S2| = |S3|

- A 32-bit counter (C200~C255) must be used with the 32-bit instruction DANDZ※. If it is used with the 16-bit instruction ANDZ※, a program error will occur, and the ERROR LED indicator on the PLC will blink.

2、Program Example:

- If M0 is On, and the absolute value of the difference between D10 and D11 is greater than K200, Y0 will be On. If the absolute value of the difference between D10 and D11 is less than or equal to K200, Y0 will be Off.


M0
ANDZ>
D10
D11
K200
Y10

ZL 308~313		ORZ※		S1 S2 S3											Comparing contact type absolute values ORZ※
	D														
	Bit Devices				Word Devices										ORZ※: 7 steps 16-bit DORZ※: 13 steps 32-bit
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S1					*	*	*	*	*	*	*	*	*		
S2					*	*	*	*	*	*	*	*	*		
S3					*	*	*	*	*	*	*	*	*		

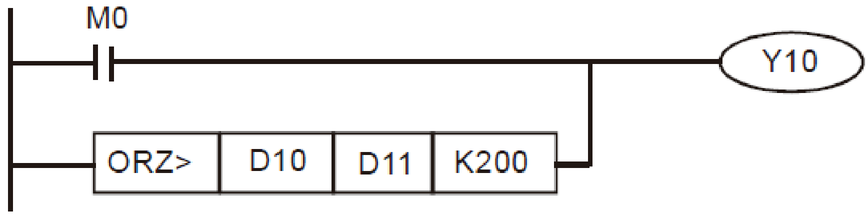
- S1: Source device 1 S2: Source device 2 S3: Source device 3
- The absolute value of the difference between S1 and S2 is compared with the absolute value of S3. Take ORZ> for example. If the comparison result is that the absolute value of the difference between S1 and S2 is greater than the absolute value of S3, the condition of the instruction is met. If the comparison result is that the absolute value of the difference between S1 and S2 is less than or equal to the absolute value of S3, the condition of the instruction is not met.
- The instruction ORZ※ is connected to a contact in parallel.

ZL No.	16-bit instruction	32-bit instruction	On	Off
308	ORZ>	DORZ>	|S1- S2| > |S3|	|S1- S2| ≦ |S3|
309	ORZ>=	DORZ>=	|S1- S2| ≧ |S3|	|S1- S2| < |S3|
310	ORZ<	DORZ<	|S1- S2| < |S3|	|S1- S2| ≧ |S3|
311	ORZ<=	DORZ<=	|S1- S2| ≦ |S3|	|S1- S2| > |S3|
312	ORZ =	DORZ =	|S1- S2| = |S3|	|S1- S2| ≠ |S3|
313	ORZ<>	DORZ<>	|S1- S2| ≠ |S3|	|S1- S2| = |S3|

- A 32-bit counter (C200~C255) must be used with the 32-bit instruction DORZ※. If it is used with the 16-bit instruction ORZ※, a program error will occur, and the ERROR LED indicator on the PLC will blink.

2、Program Example:

- If M0 is On, or the absolute value of the difference between D10 and D11 is greater than K200, Y0 will be On.


M0
Y10
ORZ>
D10
D11
K200