Featuring
Infinite
Technologies

smartPLC®

User's Manual

Firmware	v2-05 2018-12-26
Last updated	2019-02-04

株式会社エフ・アイ・ティ
FIT CO.,LTD.

Change log

Date	Remarks
2017-11-21	First edition
2018-01-16	7 - Trouble shooting. When the program can not be read. 4-2-1 Temporary memory TM923 fixed.
2018-02-20	3-5 Encoder - Program example comments
2018-06-01	- Adding "Info Pages" on Display 2-5-8, 2-5-8-4, 4-1-2, 4-2-1 - Adding a setting for the maximum of VR1, VR2 and VR3 (DM1190~DM1192) 3-6-2, 3-6-2-2, 4-2-2
2018-06-05	Adding note about a new reset sequence of M1200 4-1-2
2018-06-21	Adding information about a scan cycle 1-3-2, 1-3-2-1 Adding details about synchronization of operands 5-1-3-1
2019-01-17	- CNT bit relay operands T and C were missing - LDP, LDF updating text - Adding information about LDPB, ANP, ANPB, ORP, ORPB, LDFB, ANF, ANFB, ORF, ORFB New in v2.05 - CMP **Breaking change!** - LDA and MOV can read an actual value of a counter - TMR, TMH, TMS and TMI can use CM as operand - Monitor and Edit internal registers of smartPLC 1-2, 4
2019-02-04	- Updating information about firmware 2-5-2 - Updating Escape Character Text + replacing [￥] by [\] 2-5-8-3

Table of Contents

1 Introduction

Thank you for purchasing the General-purpose small controller smartPLC.
The smartPLC is a programmable logic controller for simple implementation of the control of small-scale devices.
The development of the smartPLC is based on our achievements and experience in an automatic machine design production.
By using the smartPLC, you can reduce the production cost by reducing number of design steps of the control panel, wiring man-hours, and programming man-hour.
Before you start using this product, read this manual thoroughly and understand fully to its content.
Use the smartPLC according to safety and proper use instructions.
Keep this manual in a safe place for a future reference.

1 - 1 Safety Precautions

1 - 1 - 1 About caution symbols

In this manual, caution symbols are shown as follows for using the smartPLC safely.

⚠Warning	If you do not handle properly, there is a serious risk which may result in injury, serious injury or even death. Also, there is a risk of serious physical damage as well.
⚠Caution	If you do not handle properly, there is a risk which may cause you to suffer minor injuries or moderate injuries. Also, there is a possibility of physical damage.
Note	It indicates additional information on proper use.

1 - 1 - 2 General Precautions

- Before you use the device, make sure to check the proper function and performance of this device.
- Proceed with care when modifying the device, or when using it in a manner that falls outside of the ranges indicated in its specifications, since FIT CO., LTD. is unable and do not guarantee device functionality or performance in such situations.
- When you are using the device in combination with other devices or equipment, it may not be able to satisfy an expected functionality or performance, as it depends on the use conditions, environment etc. Consider well before such use.
- Do not use the device for protection of human beings, or in applications where a failure of the device can lead to injuries, fatalities of humans or animals or where it can lead to high damages on equipment, environment, and so on.
- The device is not intended for use as an explosion-proof product. Do not use the device in a hazardous location or in a location that has a potentially explosive atmosphere.

1 - 2 Overview of the smartPLC

The smartPLC is a programmable logic controller which integrates following functions and features, realizes miniaturization, and it can be used to control small machines, equipment, etc.

●Production of the control panel is not necessary.

Reduced the number of design and manufacturing steps of the control panel, which had been manufactured to meet the specifications of the conventional equipment.
Standard equipped with a variety of features.

- DC24V/100W　Built-in power supply
- 21 inputs 21 outputs
- Two positioning pulse outputs for motors
- One thermocouple input (K type)
- One Load cell input (bridge type)
- One Relay output (maximum 250 V 10 A)
- SD card slot
- Liquid crystal display (8 digits 2 lines)

●Troublesome wiring is unnecessary.

Reduce wiring man-hours.

- Inputs of switches, sensors, etc. are connected by e-CON with one touch
- Outputs of DC solenoid valves etc. are connected by e-CON with one touch. (Maximum 400 mA)
- Crimp terminal wiring, crossover wiring unnecessary
- Easy installation of the main unit with a special bracket.

●Free ladder software

- Download the latest firmware and ladder software from https://smartplc.org/about/downloadsEN.html
- Connect to PC via USB cable (USB Mini-B)
- Monitor and Edit internal registers of smartPLC
- Free sample programs
- Programs tailored to the customer's specifications can be consulted.

1-3 Internal process of the smartPLC

1-3-1 Startup process

A start of the smartPLC goes in the following order.

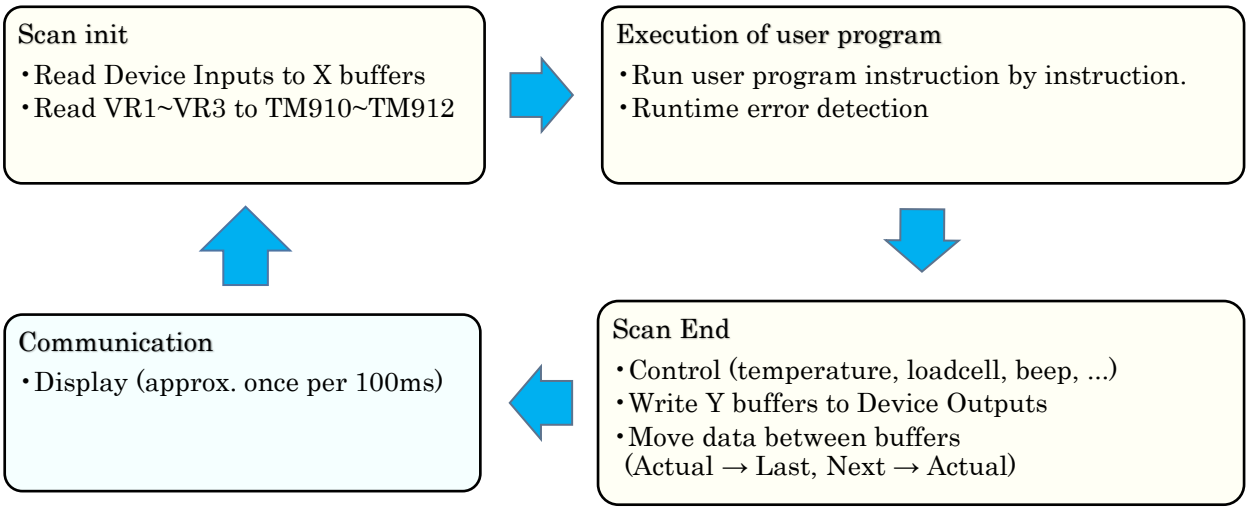
1) Hardware initialization
2) Reading and internal compiling of the program
3) Start of the program scan (program cycle)

* If there is an error in the program, it will display an error and stop.
Correct the program and reset. (Refer to 2-5-4 RUN-STOP switch)

1-3-2 Program scan

After starting up, a scan process is repeated until the smartPLC is manually stopped or the error occurs. A processing time of one scan is not constant as it is affected by many conditions (ex. program length, used instructions, used features, interrupts, …), it is normally between 1ms to 10ms.

1-3-2-1 Sequence of the scan



* If an error occurs during scanning, all outputs are turned off, an error is displayed and the running process is stopped.

1-4 Specification

1-4-1 Rating

Item	Specification
AC supply voltage	100~200 VAC 50/60Hz
Allowable supply voltage	85~220 VAC
Power consumption	100 W or less (Excluding external AC equipment)
Operating ambient temperature	10 to 50°C
Output voltage	Service power supply 24 VDC ± 10% (4 A) * Including internal consumption 5 VDC ± 10% (1.2 A) * Including internal consumption
	AC control power output depends on AC power supply voltage (125 VAC or less 10 A, 126 VAC or more 3A)
Storage temperature	0 to 75°C
Operating ambient humidity	10 to 90%RH (no condensation)
Operating environment	As little dust and corrosive gases as possible. Dust should not be conductive.
Installation method	Left side bracket installation
Outer dimension	Height 164 × width 56 × depth 165 (mm)
Weight	1.2 kg

1-4-2 Performance

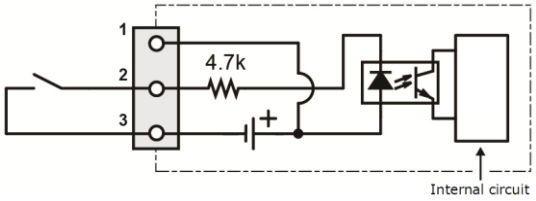
Item	Specification
Control system	Stored program system
I/O control system	Cycle scan refresh system
Programming language	Ladder diagram, Mnemonic language
Program capacity	3000 steps
Display	8 digits 2 lines LCD with backlight
General-purpose operation button	4 buttons on the front panel [X200 to X203]
General-purpose potentiometer	3 potentiometers on the front panel
Memory hold function	Built-in flash memory

Item	Specification
Power off hold function	DM area and CM area
Total I / O	42
DC general-purpose input	17, DC24V NPN type [X000 ~ X015, X204]
DC general-purpose output	17, DC24V NPN transistor type [Y000 to Y015, Y200]
DC high speed input	4, DC24V NPN type [X100 to X103]
DC high speed output	4, DC24V NPN transistor type [Y100 to Y103]
Thermocouple input	1, K type Measurement temperature range 0 to 700 ° C
Load cell input	1, bridge type
AC control power output	1, relay [Y201]
DC control supply output	24V for inputs/outputs is controlled by [Y202]. Terminal block DC 24 V is always on.
Internal auxiliary relay	800 bits
Internal holding relay/ Control memory	160 bits / 10 WORDs
Timer relay	100
Counter relay	100
Data memory	1700 WORDs
Temporary memory	100 WORDs
Control memory	10 WORDs

1-4-3 General-purpose input / output specification

DC General-purpose input X000 to X015 (16 inputs)

Item	Specification
Input type	Open collector input
Input voltage	24 VDC
Input impedance	4.7KΩ
Input current	4.8mA

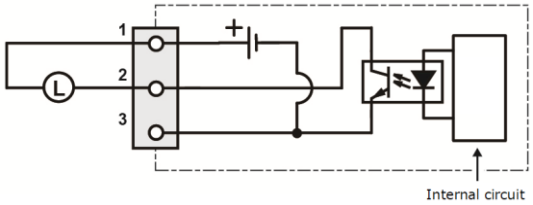


* Terminal 1 (+24 V), terminal 3 (GND) of all e-cons are internally short-circuited.

DC General-purpose output Y000 to Y015 (16 outputs)

Item	Specification
Output type	Transistor NPN output
Rated load	24 VDC 400mA

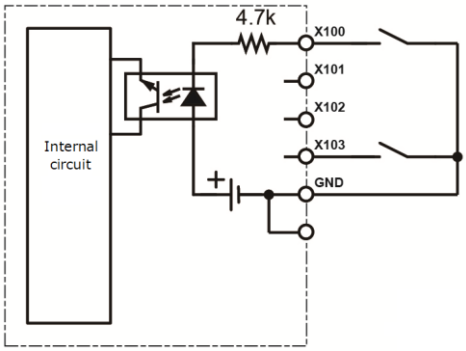
* Terminal 1 (+24 V), terminal 3 (GND) of all e-cons are internally short-circuited.



1-4-4 High speed input / output specification

DC high-speed input X100 to X103 (4 inputs)

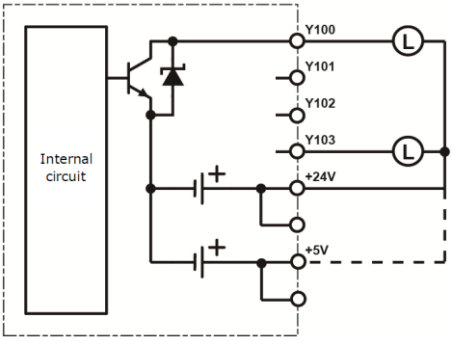
Item	Specification
Input type	Open collector input
Input voltage	24 VDC
Input impedance	4.7KΩ
Input current	4.8mA
Max. frequency	3.8kHz



DC high speed output Y100 to Y103 (4 outputs)

Item	Specification
Output type	Transistor NPN output
Rated load	24 VDC 400mA
Pulse max. frequency	50kHz

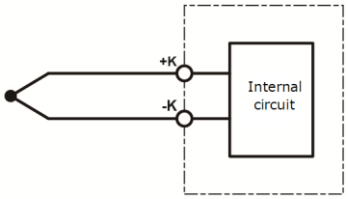
* It is possible to position control the maximum of 2 motors. Use + 5V power supply for the motor driver.



1-4-5 Thermocouple input specification

Thermocouple analog input

Item	Specification
Input sensor	Thermocouple K type
Sampling cycle	0.5s
Heater output	AC power supply control output [Y201]
control method	ON / OFF control
Measured temperature range	0～700℃（±2℃）

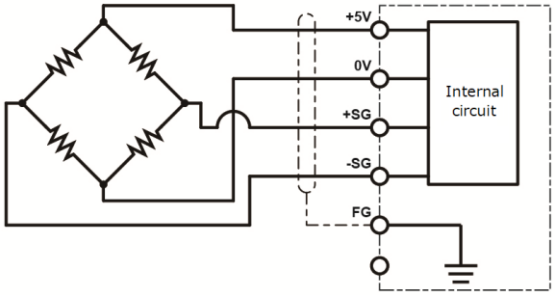


1-4-6 Load cell input specification

Load cell analog input

Item	Specification
Applied voltage	DC5V±5%
Resolution	24 bits
Maximum rated input range	1 to 3 mV / V
Measurement range	0 to 500 kg
Temperature drift	±10nV/℃

*The load cell is bridge type 100 Ω or more.



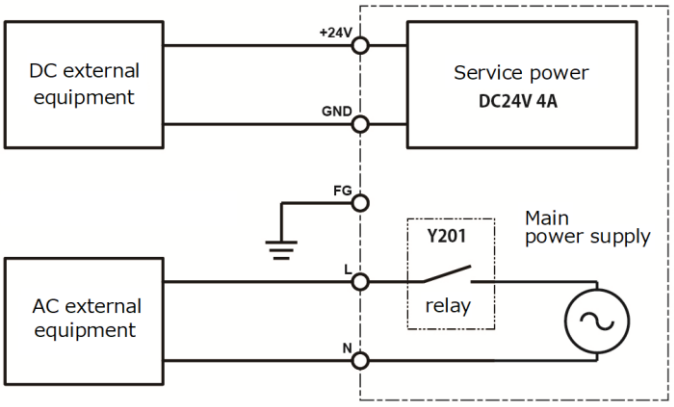
1-4-7 Terminal block

24 VDC service power supply

Item	Specification
Rated voltage	24 VDC±10%
Rated current	4 A (including internal consumption)

AC power supply control output

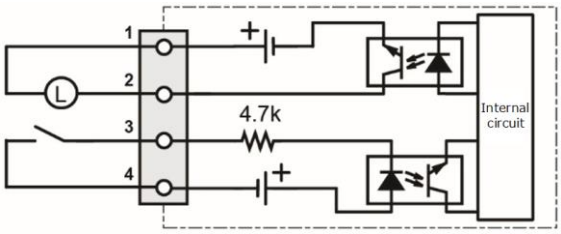
Item	Specification
Rated voltage	Depends on main unit power supply voltage 100~250 VAC
Rated current	Depends on main unit power supply voltage 10A/100V 3A/250V



1-4-8 Extended input / output specification

DC General-purpose input X204

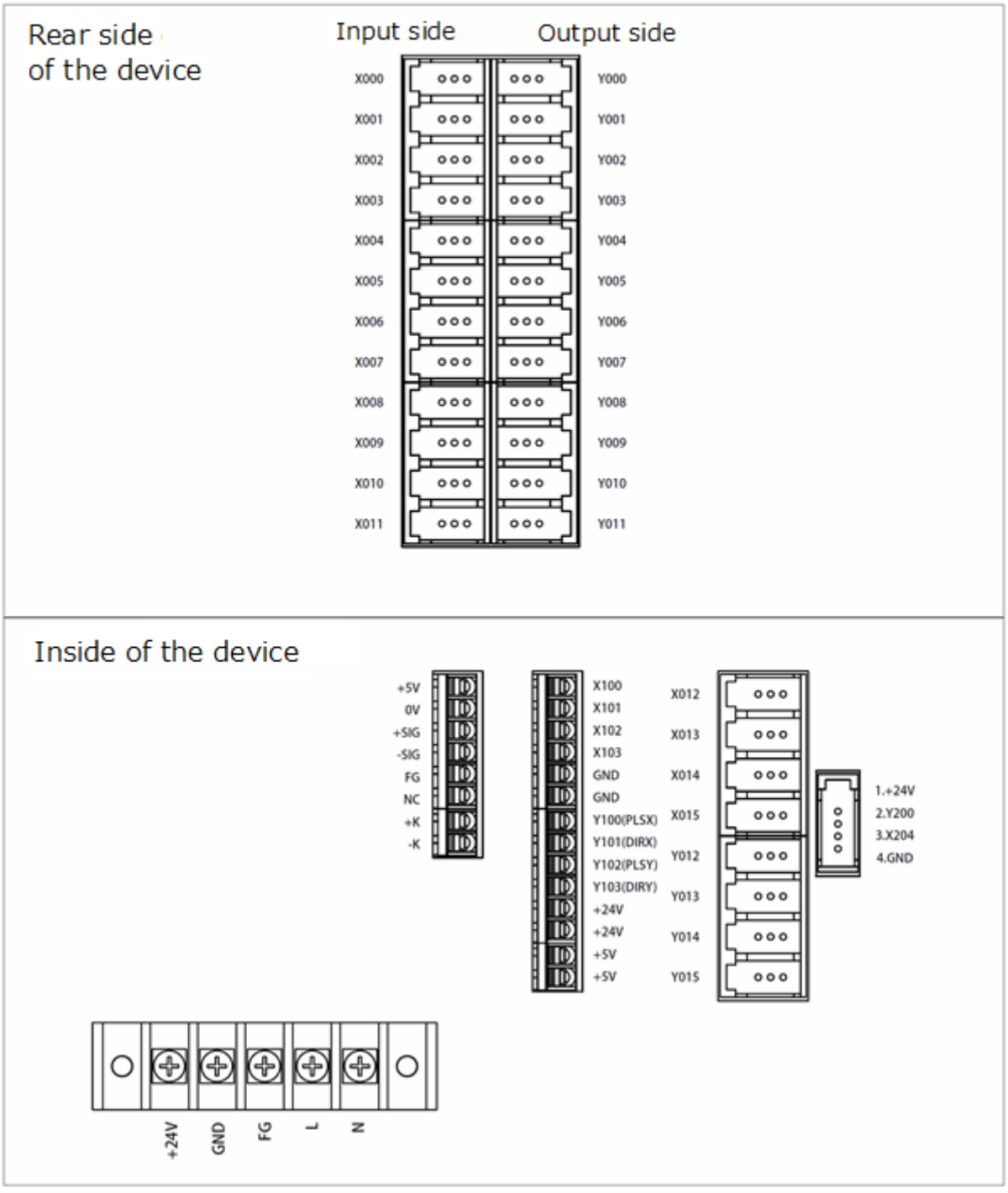
Item	Specification
Input type	Open collector input
Input voltage	24 VDC
Input impedance	4.7KΩ
Input current	4.8mA



DC General-purpose output Y200

Item	Specification
Output type	Transistor NPN output
Rated load	24 VDC 400mA

■ Terminal block · Connector wiring diagram

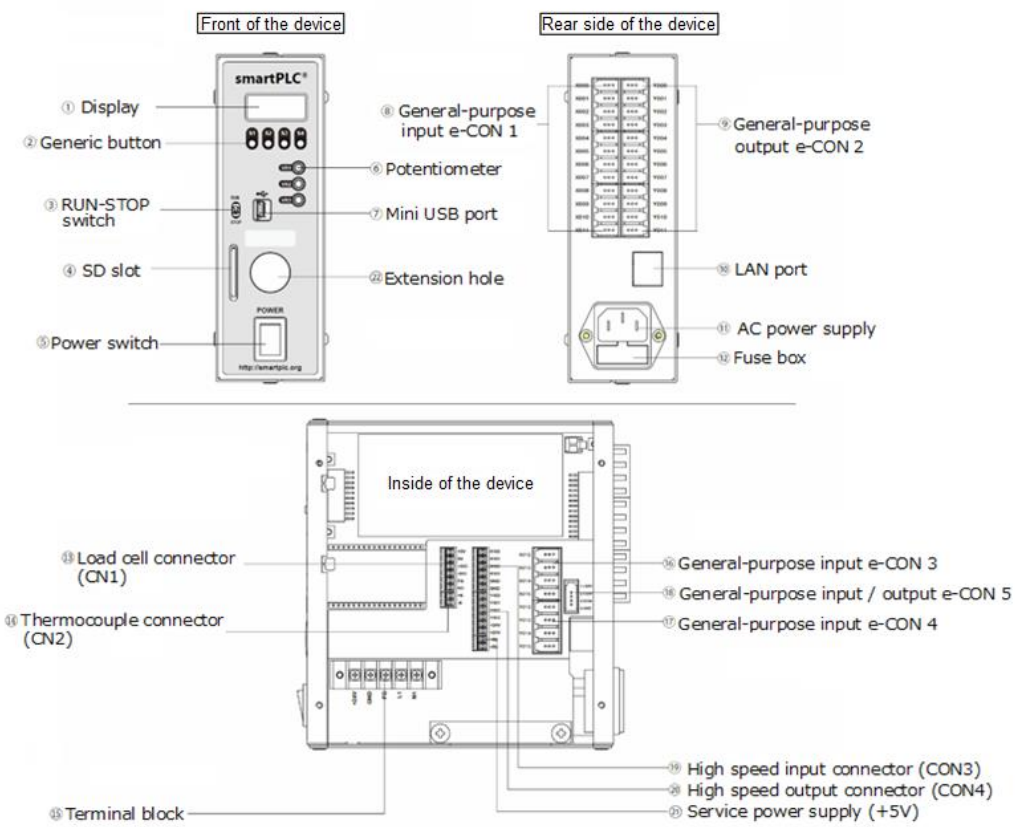


Warning	Make sure to unplug the power plug when wiring inside. Be careful as the terminal block (L, N) is using a one-way switch.

Warning	When you are doing an internal wiring. Do not touch the microcontroller or electronic parts. Do not to drop metal items inside to device.

2 Usage directions

2 - 1 Names and functions of parts



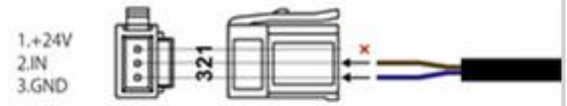
Num	Name	Feature Description
1	Display	Various values are displayed.
2	Generic button	The general-purpose buttons can be freely used in the user program.
3	RUN-STOP switch	It switches between RUN mode and STOP mode.
4	SD slot	A slot for SD card insertion.
5	Power switch	The main power switch of the device.
6	Potentiometer	Values (0~10) can be used as input in the user program.
7	Mini USB port	A port for communication with PC.
8	General-purpose input e-CON 1	An e-CON connector for 12 inputs.
9	General-purpose output e-CON 2	An e-CON connector for 12 outputs.
10	LAN port	A port for a network connection. *
11	AC power supply	A connector for a power input
12	Fuse box	A fuse inside.
13	Load cell connector	A connector for a bridge type load cell.
14	Thermocouple connector	A connector for a thermocouple (K).
15	Terminal block	A terminal block with output power (24VDC 4A) and (AC 125V 10A or AC 250V 3A).
16	General-purpose input e-CON 3	An e-CON connector for internal 4 inputs.
17	General-purpose input e-CON 4	An e-CON connector for internal 4 outputs.
18	General-purpose I / O e-CON 5	An e-CON connector for 1 input and 1 output.
19	High speed input connector	A connector for a high-speed pulse input.
20	High speed output connector	A connector for a positioning pulse output.
21	Service power supply (+5V)	A power supply for a motor driver 5V interface.
22	Extension hole	A mounting hole 16 mm which can be used for an attaching a switch etc.

* LAN port is currently not available for a use.

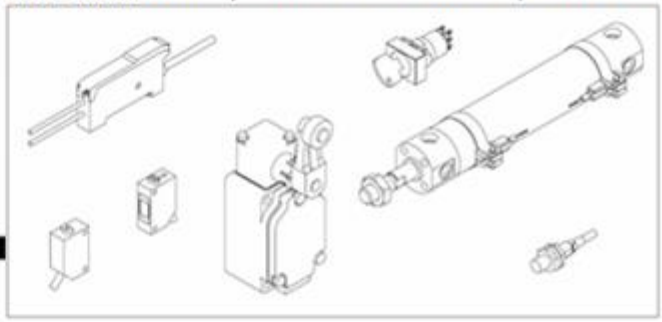
2-2 Device connection

■ Connector wiring of input X000 to X015

A contact or an NPN type sensor can be connected to these inputs. If you need a power supply (1A or less) for your sensor, connect to pin no.1 as appropriate.



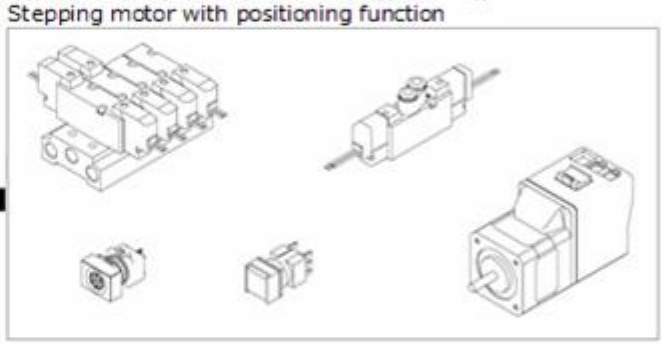
Example of an equipment which can be connected.
Sensor amplifier, Photoelectric sensor, Proximity sensor, Auto switch Various push buttons and switches, Limit switch



■ Connector wiring of output Y000 to Y015

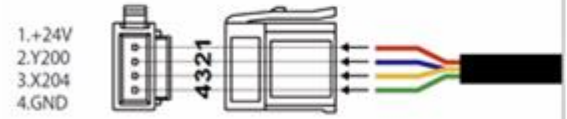
Connect devices with a rated voltage 24VDC and a current consumption of 400mA or less.

Example of an equipment which can be connected.
Solenoid valve, Buzzer, Indicator light, Relay, Stepping motor with positioning function

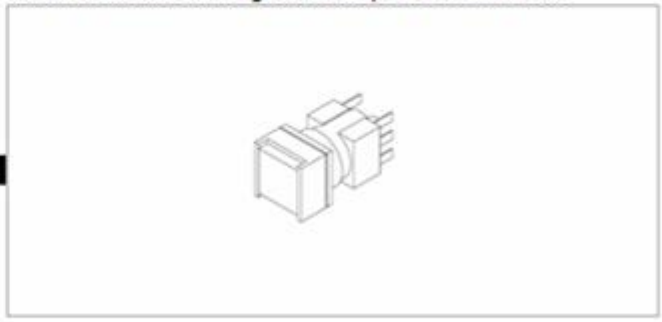


■ Connector wiring of Y200 and X204

A connector with an input / output pair.



Example of an equipment which can be connected.
Illuminated push button switch and etc.
It can be installed using the front panel 16 mm hole.



⚠ Caution	Before doing any wiring or connecting/disconnecting connectors or opening the device turn off the power supply and disconnect the main power cable from the device. Please, be careful. If you do not connect correctly +24V(+) and GND(-) pins the device or a connected equipment may be damaged. In case of misplacing the input/output connectors, the device or a connected equipment may be damaged. Do not connect the DC output of an another power supply in either the series or parallel to the terminal block output power 24VDC, as the power supply of the device can be damaged.

2-3 E-con mini clamp usage

① Select connector

1) Check the specifications (cross sectional area, outside diameter) of the electric wire.

2) Use a suitable connector based on the applicable table.

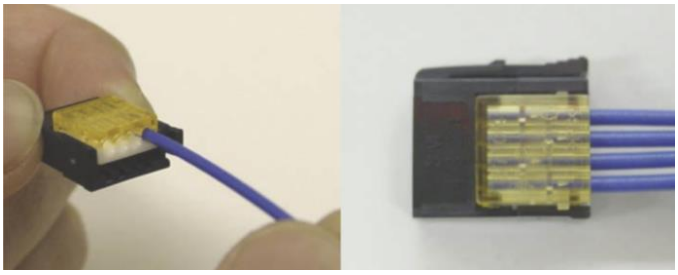
3 pole mini clamp connector (3 M)				
Part number	Applicable wire			Cover color
	AWG	Nominal cross-sectional area [mm^2]	outside diameter [mm]	
37103-2124-000FL	20 - 22	0.3 or more -0.5	1.0 - 1.2	Green
37103-2165-000FL	20 - 22	0.3 or more -0.5	1.2 – 1.6	Blue
37103-2206-000FL	20 - 22	0.3 or more -0.5	1.6 – 2.0	Gray

4 pole mini clamp connector (3 M)				
Part number	Applicable wire			Cover color
	AWG	Nominal cross-sectional area [mm^2]	outside diameter [mm]	
37104-2124-000FL	20 - 22	0.3 or more -0.5	1.0 - 1.2	Green
37104-2165-000FL	20 - 22	0.3 or more -0.5	1.2 – 1.6	Blue
37104-2206-000FL	20 - 22	0.3 or more -0.5	1.6 – 2.0	Gray

② Insert a wire

1) Check the pin number and insert the wire between the top cover (translucent part) and the base cover (white part).

2) Make sure that the wire is fully inserted as on the picture.



③ Crimp

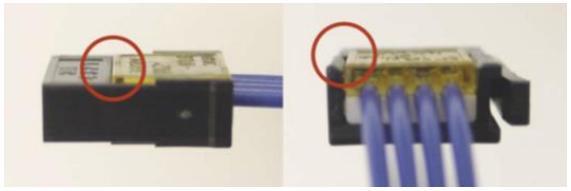
Push the cover into the connector body with pliers.

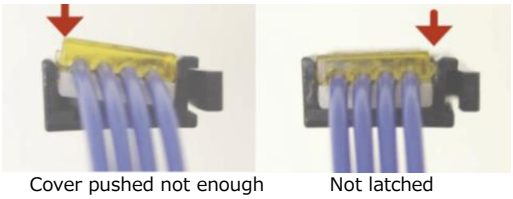
* Crimp with the pliers from the lateral direction of the connector as shown in the picture.

④ Check

Horizontal the cover to the body.

Make sure that there is no gap between the body and the cover.





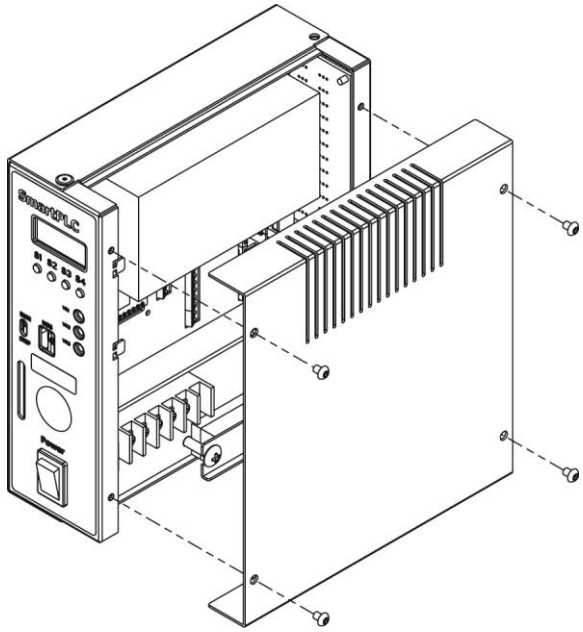
Cover pushed not enough　　Not latched

Push again at point marked by the red arrow.

2-4 Manipulation with the cover and bracket

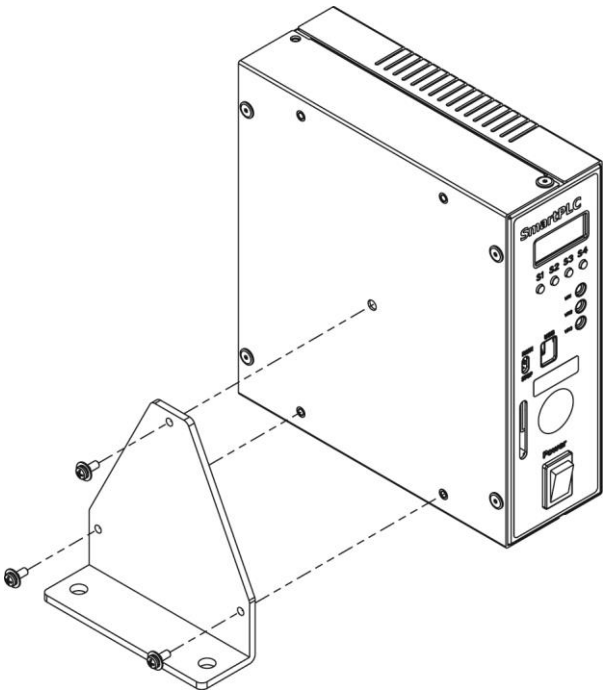
2-4-1 How to open the cover

To use the thermocouple, load cell, motor positioning control and other functions,
you need to remove the cover. Remove the four M3 screws with a Phillips screwdriver.

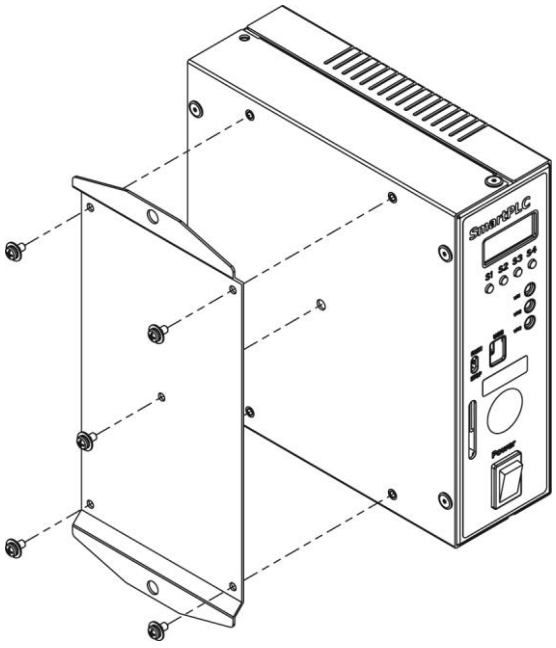


2-4-2 Install the bracket

Be sure to install the unit vertically.
The screw size attached to the right side is M3. Use the suitable length.
Two types of special brackets are available as options



Top mounting

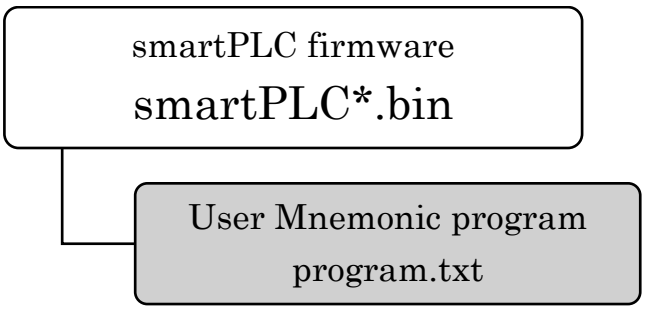


Side mounting

2-5 Basic usage

2-5-1 SmartPLC firmware and user program

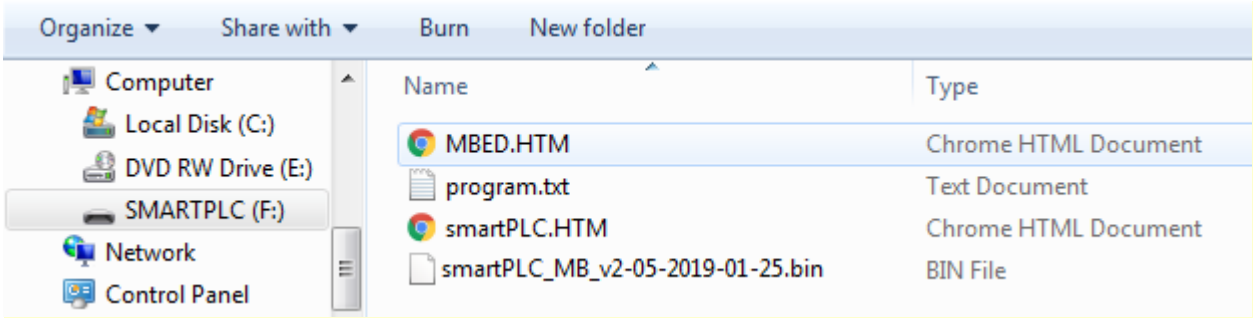
You can upgrade the firmware version by replacing smartPLC*.bin saved in smartPLC's virtual drive. (See Connection with a personal computer).
The firmware can be downloaded from https://smartplc.org/about/downloadsEN.html.
You can create a control program by editing program.txt.



* Firmware is software that controls the main unit such as user program execution and error handling.

2-5-2 Connection with a personal computer

You can change the program from the PC via USB. The supported OS is windows 7 and 10. When connecting with the PC with the included USB mini cable, it will be recognized as a smartPLC virtual drive.



Following files are saved on the smartPLC virtual drive.

File name	Description
smartPLC*.bin	Firmware, * represent version. v2 version, xx subversion, YYYY-MM-DD build date
smartPLC.HTM	smartPLC homepage, download site
program.txt	User Mnemonic program (See 5-1 Program)
MBED.HTM	Mbed microcontroller home page

About MBED.HTM
smartPLC uses the Arm Mbed IoT Device Platform developed by ARM, UK. Therefore, the Mbed home page MBED.HTM (English) is saved on the smartPLC virtual drive. There is no problem opening the link, but Mbed and smartPLC are not related. Please DO NOT ask about smartPLC at Mbed or ARM web pages or contacts.
* Mbed is a registered trademark of ARM, UK.

There are two versions of firmware smartPLC_v2*.bin and smartPLC_MB_v2*.bin, because there are two versions of the main board. A blue one uses the smartPLC_MB_v2*.bin firmware. When updating a firmware, just keep in mind that *_MB_* version has to be replaced by another *_MB_* version, and smartPLC_v2*.bin with smartPLC_v2*.bin.

2-5-3 Priority of a user mnemonic program execution

Priority	Folder where is the program.txt
1	Root folder of SD card
2	Root folder of smartPLC virtual drive

* If there is no program.txt in the inserted SD card, an error will be displayed.

2-5-4 RUN-STOP switch

RUN	Read the program and execute the control.
STOP	Firmware stops, all outputs are turned off.

Reset: Perform a hardware reset by switching the RUN-STOP switch in the sequence RUN → STOP → RUN.

The program file can be edited during RUN operation, you have to reset smartPLC to activate the changes.

2-5-5 DC 24V for outputs

The DC 24V internal power supply for e-con is controlled by a program.
The DC 24V of the terminal block is not controlled by the user program, it is permanently on.
To activate DC 24V for outputs, turn on Y202 as shown below. (See 5 Program and instructions)

```
LD    M1200          // Always ON
OUT   Y202           // DC24V output

… program …

END
```

DC24 V output example

2-5-6 Data memory (DM)

For motor position determination, load cell load measurement, and temperature control, it is necessary to write the setting values to the data memory (DM). (See 4 Device Organization)

Writing to the data memory is possible with program data transfer instructions (MOV, STA, etc.), or you can write and check it from the PC using the data memory setting application. (Refer to section 6 Data memory setting application (DM_Set))

Writing to data memory (DM) is much slower than writing to temporary memory. Extensive writing to the data memory can significantly extend the scan time.

2-5-7 Numeric type

Data	Notation examples	Numeric type
Numeric constant (literal) 16bit	# 1 #65535	Unsigned 16bit decimal number (0 to 65535)
Numeric constant (literal) 32bit	#-1 #1L #65536 #65536L	Signed 32bit number (-2147483648 ～ 2147483647)
16bit data register	DM1000 TM902	Unsigned decimal number (0 to 65535)
32bit data made from 2 continuous 16bit registers	TM902L (32bit notation, is made from TM902 (low16bits) and TM903 (high16bits))	Signed 32bit number (-2147483648 ～ 2147483647)

*For 32bit notation refer to 5-2-3-1 32bit instruction notation and usage.

2-5-7-1 Example of 32bit number split into to two 16bit numbers

The following table shows how is the 32bit number split into two 16bit registers (high and low).
If the value is negative, it is saved as 2's complement.

32bit decimal	32bit hex		16bit hex	16bit decimal
100000	000186A0	Low 16bits	86A0	34464
		High 16bits	0001	1
-20000	FFFFB1E0	Low 16bits	B1E0	45536
		High 16bits	FFFF	65535

2-5-8 Display

2-5-8-1 Display settings in temporary memory

There are special registers in temporary memory dedicated to control a display.
By default, zero is displayed on the both rows of the display, because default mode (TM900, TM901) is 0 (display number) which is in TM902L/TM904L and default unit (TM906, TM907) is 0 (no unit).
For more details look on the following tables.
Note if memory bit M1215 is ON an info page (defined by TM908, refer to 2-5-8-4) will be displayed.

Table 1 Display data in Temporary memory (TM)

Temporary memory	Description	Remarks
TM900	Display lower row - Mode selection	Refer to Table 2.
TM901	Display upper row - Mode selection	Refer to Table 2.
TM902 (low 16bits)	Display lower row - Numerical value (*1)	Signed 32bit number
TM903 (high 16bits)		
TM904 (low 16bits)	Display upper row - Numerical value (*1)	Signed 32bit number
TM905 (high 16bits)		
TM906	Display lower row - Unit	Refer to Table 3.
TM907	Display upper row - Unit	Refer to Table 3.
TM908	Info page to be displayed (if M1215)	Refer to 2-5-8-4 Table 4.
TM914 ~ TM917	Display lower row - User defined characters	Refer to section "About user-defined characters"
TM924 ~ TM927	Display upper row - User defined characters	Refer to section "About user-defined characters"

(*1) 32bit signed integer has up to 11 characters, but display has only 8 characters per row and so if the number does not fit in, the left pointing arrow character '←' will be displayed on the most left character position of the row.

Table 2 Display string number table (TM900, TM901)

Number	Description
0	Numerical value in TM902L, TM904L + unit (TM906, TM907)
1	RUN
2	STANDBY
3	START
4	STOP
5	Temp
6	Temp-Hi
7	Temp-Low
8	TotalCNT
9	VR1
10	VR2
11	VR3
12	VR4
13	DM

Number	Description
14	Load(kg)
15	Load-Hi
16	Load-Low
17	Error!
18	ALM!
19	smartPLC
20	Hello!
21	Pos.END
22	Pos. A
23	Pos. B
24	STEP
25	AUTO-M
26	MANU-M
27	STEP-M
28	TEACH-M
1000	User defined characters in TM914~TM917, TM924~TM927

Table 3 Display unit number table (TM906 TM907)

Number	Description	Example		
		Result on Display lower row	TM902L	TM906
0	Signed Integer - without unit	-5	-5	0
1	Absolute Value without unit	5	-5	1
2	Real number 0.1 no unit	0.5	-5	2
3	Real number 0.01 no unit	0.05	-5	3
4	Signed Integer g	-5g	-5	4
5	Absolute value g	5g	-5	5
6	Real number 0.1g	0.5g	-5	6
7	Real number 0.01g	0.05g	-5	7
8	Signed Integer kg	-5kg	-5	8
9	Absolute value kg	5kg	-5	9
10	Real number 0.1kg	0.5kg	-5	10
11	Real number 0.01kg	0.05kg	-5	11
12	Signed Integer Hz	-5vHz	-5	12
13	Absolute value Hz	5vHz	-5	13
14	Real number 0.1 Hz	0.5vHz	-5	14
15	Real number 0.01 Hz	0.0Hz	-5	15
16	Signed Integer ms	-5ms	-5	16
17	Absolute value ms	5ms	-5	17
18	Real number 0.1 ms	0.5ms	-5	18
19	Real number 0.01 ms	0.05ms	-5	19
20	Signed Integer s	-5s	-5	20

Number	Description	Example		
		Result on Display lower row	TM902L	TM906
21	Absolute value s	5s	-5	21
22	Real number 0.1 s	0.5s	-5	22
23	Real number 0.01 s	0.05s	-5	23
24	Signed Integer mm	-5mm	-5	24
25	Absolute value mm	5mm	-5	25
26	Real number 0.1 mm	0.5mm	-5	26
27	Real number 0.01 mm	0.05mm	-5	27
28	Signed Integer m	-5m	-5	28
29	Absolute value m	5m	-5	29
30	Real number 0.1m	0.5m	-5	30
31	Real number 0.01m	0.05m	-5	31
32	Signed Integer ℃	-5℃	-5	32
33	Absolute value ℃	5℃	-5	33
34	Real number 0.1℃	0.5℃	-5	34
35	Real number 0.01℃	0.05℃	-5	35

Program example

```
LD   M1200           // Always ON
MOV  #0 TM900        // The lower row of the display shows numerical value
                     // (Refer to Table 2)
MOV  #1 TM901        // The upper row of the display shows the characters
                     // "RUN" (Refer to Table 2)
MOV  #1234 TM902L    // Specify 1234 as the lower figure
MOV  #27 TM906       // Display lower row - unit mm, display 2 decimal places
                     // (Refer to Table 3)

END
```

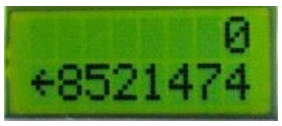
Display "RUN" on the upper row and "12.34mm" on the lower row

2-5-8-2 32bit numeric constant may not be fully displayed on the display.

32bit signed integer has up to 11 characters, but display has only 8 characters per row and so if the number does not fit in, the left pointing arrow character '←' will be displayed on the most left character position of the row.

Program example

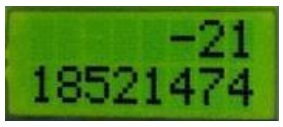
```
LD   M1202           // the First SCAN ON, after OFF
MOV  #-2118521474 TM0L   // a number to display
END
```



Display 32bit numeric constant on the lower row – the constant does not fit in the row

Program example

```
LD   M1202           // the First SCAN ON, after OFF
MOV  #-2118521474 TM0L   // a number to display

MOV  #0 TM906        // no unit on the display lower row
MOV  #0 TM907        // no unit on the display upper row
LDA  TM0L            // the number to internal register
DIV  #100000000      // divide (1 with 8 zeros)
STA  TM904L          // upper row of the display
CMP  #0              // compare internal register with #0

ANB  M1211           // not equal #0
MOV  #1 TM906        // absolute value the display lower row

LD   M1202           // the First SCAN ON, after OFF
MUL  #100000000      // multiple (1 with 8 zeros)
STA  TM2L            // #-2100000000 to TM2L
LDA  TM0L            // the number (#-2118521474) to internal register
SUB  TM2L            // subtract #-2118521474 - #-2100000000
STA  TM902L          // #-18521474 to the display lower row

AND  M1211           // if in previous comparison – equal #0
MOV  #"    " TM924L  // space chars to display upper row – left side
MOV  #"    " TM926L  // space chars to display upper row – right side
MOV  #1000 TM901     // display upper row mode – user characters
CMP  #0              // compare internal register with #0
AND  M1210           // and less than #0
DIV  #10000000       // divide (1 with 7 zeros)
CMP  #0              // compare internal register with #0
AND  M1210           // and less than #0
LDA  TM902L          // the number from display lower row to internal register
MUL  #-1             // multiple by -1 (change sign from – to +)
STA  TM902L          // internal register to the display lower row
MOV  #" -" TM927     // display '–' char on the upper row

END
```



Displays numerical values exceeding 8 digits using the upper and lower rows of the display

2-5-8-3 User defined characters

Any character of display font can be displayed on the display by setting "1000" in TM900, TM901. Each of rows (upper and lower) have 8 characters.

Escape character

The escape character is backslash [\] on English keyboard or a halfwidth "en" character [¥] on Japanese keyboard, it has 8bit code 92 (5Ch). There is no backslash [\] character in display font, the "en" character [¥] will be displayed instead.

To write a Double Quotation Mark ["] an escape character has to be used before the character [\"] (or [¥"] in Japanese environment).

To write the [¥] character use two backslashes [\\] (or [¥¥] in Japanese environment).

Japanese katakana character

Display supports only half-width katakana characters. For program file format refer to 5 - 1 Program.

Table 4. – Display characters in relation to temporary memory 16bit access

TM924	TM925	TM926	TM927
"CH"	**"AR"**	**"#0"**	**"13"**
TM914	TM915	TM916	TM917
"ｹﾞ"	**"-ﾑ"**	**"\\\""**	**"1\""**

As you can see from table 4 there is two characters in one temporary memory register.
The first character is a lower byte of the register and the second character is a higher byte of the register.

Table 5. – Display characters in relation to temporary memory 32bit access

TM924L	TM926L
"CHAR"	**"#013"**
TM914L	TM916L
"ｹﾞｰﾑ"	**"\\\"1\""**

There is used 32bit access to temporary memory registers in Table5. The 32bit access to the register is marked by 'L' next to the register number. The 32bit access allows to write 4 characters on the display by one instruction.

```
//16bit access - 2 characters
MOV  #"\\\"" TM916   // Display [¥"] on the middle right side of the lower row
MOV  #"1\"" TM917    // Display [1"] on the right side of the lower row
```

```
//32bit access - 4 characters
MOV  #"\\\"1\"" TM916L    // Display [¥"1"] on the right side of the lower row
```

Note: The 16bit access and 32bit access from the example will display the same characters on the same position.

Program example

```
LD   M1200           // Always ON
MOV  #1000 TM900     // Display user characters
MOV  #1000 TM901     // Display user characters
MOV  #"ｹﾞｰﾑ" TM914L  // Displayed on the lower left side
MOV  #"\\\"1\"" TM916L // Displayed on the lower right side
MOV  #"CHAR" TM924L  // Displayed on the upper left side
MOV  #"#013" TM926L  // Displayed on the upper right side
END
```

User characters on the display

2-5-8-4 Info pages on display

The info pages are easy way to show a state of inputs and outputs, or firmware build date (page #0 or any invalid value in TM908). To activate info pages on display, set the memory bit M1215 to ON.

Memory Bit	Description	Read	Write
M1215	Show info page (defined by TM908) on display	○	○

Table 4 Info page numbers with description (TM908)

Number	Description
0	Firmware build date (ex. smartPLC_v2-03-2018-06-05.bin will display 18-06-05)
1	Inputs X0~X7
2	Inputs X8~X15
3	Inputs X100~X103 (Only in input mode -> When Encoder Control (M1507) is ON, the last value of the inputs will be on display)
4	Inputs X200~X204
5	Outputs Y0~Y7
6	Outputs Y8~Y15
7	Outputs Y100~Y103 (Only in output mode -> When sending the pulses to Motor, the data on page will not change)
8	Outputs Y200~Y202

Program example

```
LD M1200            // Always ON
OUT Y202            // 24V ON

MOV #"Pres" TM924L  //display upper row left
MOV #"s S1" TM926L  //display upper row right
MOV #1000 TM901     //display upper row - mode user characters
MOV #"Info" TM914L  //display lower row left
MOV #"Page" TM916L  //display lower row right
MOV #1000 TM900     //display lower row - mode user characters
//S1 -> info-page ON/OFF
LDP X200            //Rising Edge on S1
ANB M1215           //Show Info Page
LDP X200            //Rising Edge on S1
AND M1215           //Show Info Page
KEEP M1215          //Show Info Page
//VR1 -> change info page number
LD M1215
MOV TM910 TM908     //VR1 -> INFO_PAGE_NUMBER
```

Show Info Page

3 Control method

3 - 1 Use relays

3 - 1 - 1 - 1 Connection

	range	Description	Place
Input	X0~11	General-purpose input	Back
	X12~15	General-purpose input	Internal
	X100~X103	High speed input	Internal
	X200~X203	Generic button	Front
	X204	General-purpose input	Internal
Output	Y0~11	General-purpose output	Back
	Y12~15	General-purpose output	Internal
	Y100~Y103	High speed output	Internal
	Y200	General-purpose output	Internal
	Y201	AC power supply control output	Internal terminal block (L, N)
	Y202	DC 24V output	-

Program example

```
LD   M1200          // Always on
OUT  Y202           // DC24V output

LD   X0
OUT  Y0             // Output the state of X0 to Y0

END
```

Output the state of X0 to Y0

3-2 Stepping motor

3-2-1 Common

3-2-1-1 Connection

Motor	Operand	Direction of rotation	Output destination (1 pulse system)	Output destination (2 pulse system)
Motor 1	Y100	C W	Y 100: Pulse output ON Y 101: rotation direction OFF	Y 100: Pulse output ON Y 101: Pulse output OFF
		CCW	Y 100: Pulse output ON Y 101: rotation direction ON	Y 100: Pulse output OFF Y 101: Pulse output ON
Motor 2	Y102	C W	Y 102: Pulse output ON Y 103: rotation direction OFF	Y 102: Pulse output ON Y 103: Pulse output OFF
		CCW	Y 102: Pulse output ON Y 103: rotation direction ON	Y 102: Pulse output OFF Y 103: Pulse output ON

3-2-1-2 Condition setting: current position reset, pulse method

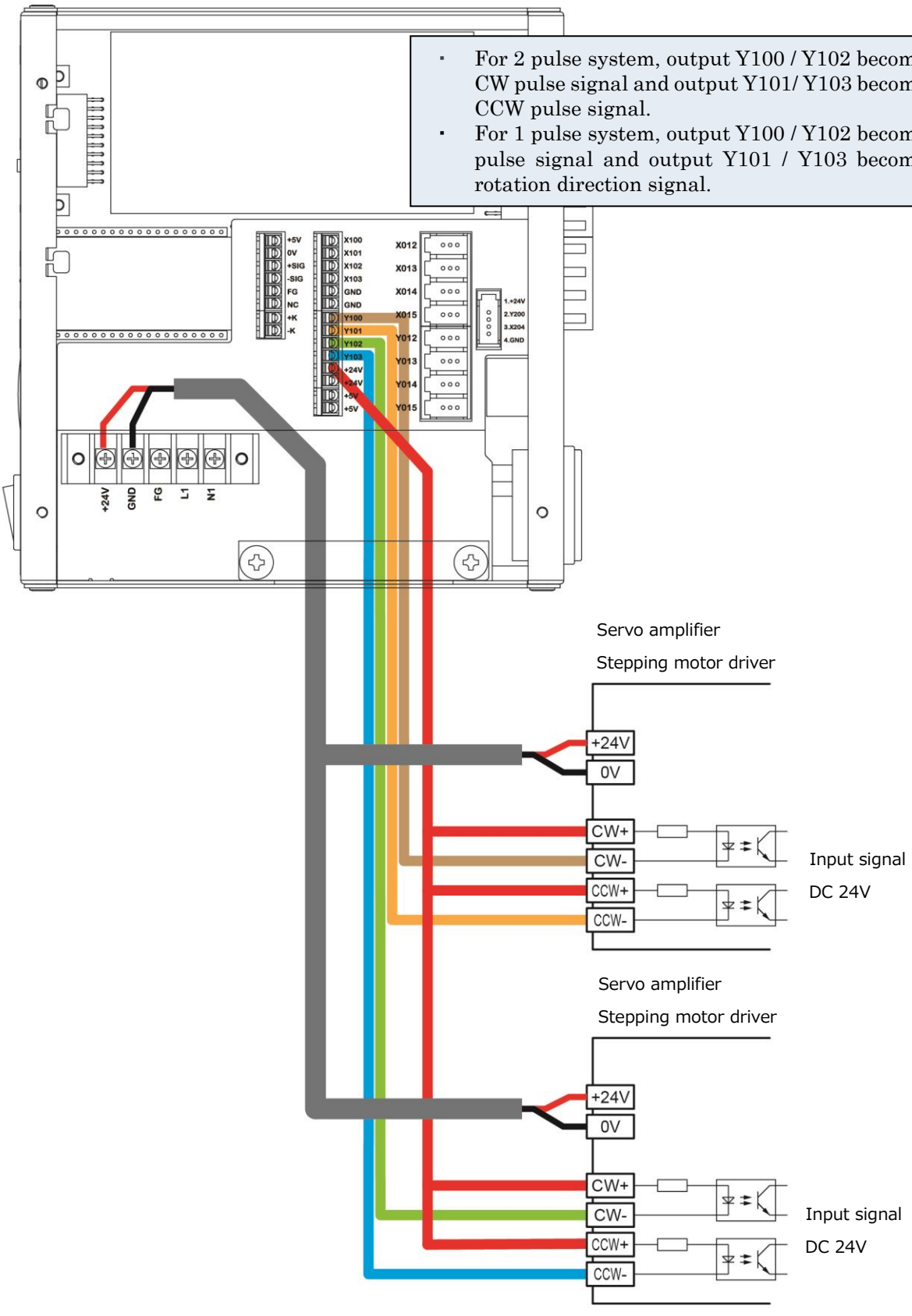
Memory Bit	Description	Read	Write
M1500	Immediately stop sending the pulses to Motor 1.	○	○
M1501	Immediately stop sending the pulses to Motor 2.	○	○
M1502	Reset the current position counter of Motor 1	○	○
M1503	Reset the current position counter of Motor 2	○	○
M1504	Sending the pulses to Motor 1.	○	×
M1505	Sending the pulses to Motor 2.	○	×
M1506	ON - 2 pulse mode, OFF - 1 pulse mode (Default value: OFF)	○	○
M1512	Motor 1 positioning complete (PLS instruction only)	○	×
M1513	Motor 2 positioning complete (PLS instruction only)	○	×

3-2-1-3 Current position (32 bit) output

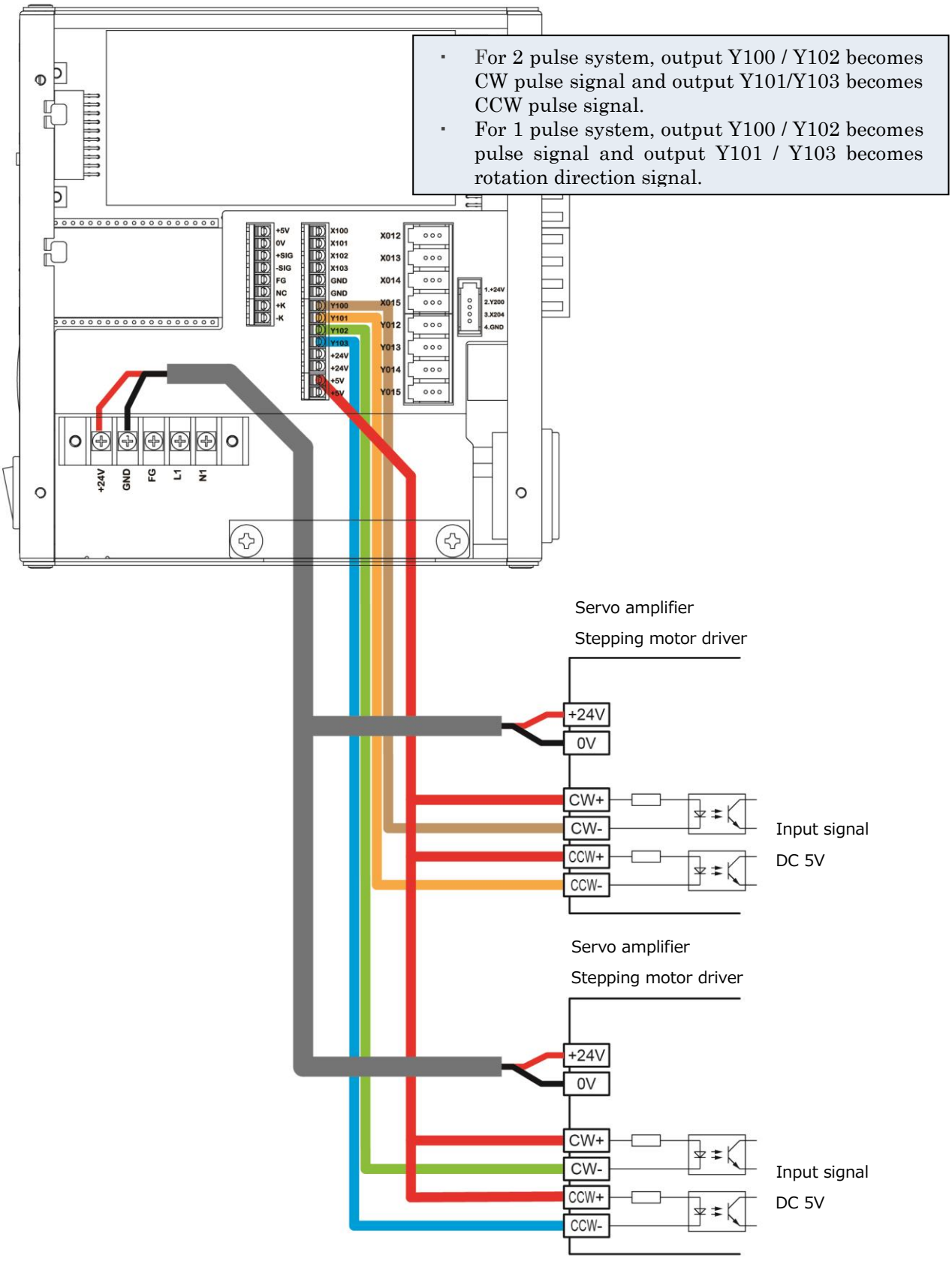
Special register in temporary memory		Description
TM940L (32bit)	TM940 low 16bits	Motor 1 current position counter
	TM941 high 16bits	
TM942L (32bit)	TM942 low 16bits	Motor 2 current position counter
	TM943 high 16bits	

3-2-2 Stepping motor connection example DC 24V

The example of the connection of smartPLC and stepping motor driver or servo amplifier. Check the input voltage specification on the driver side before connecting.



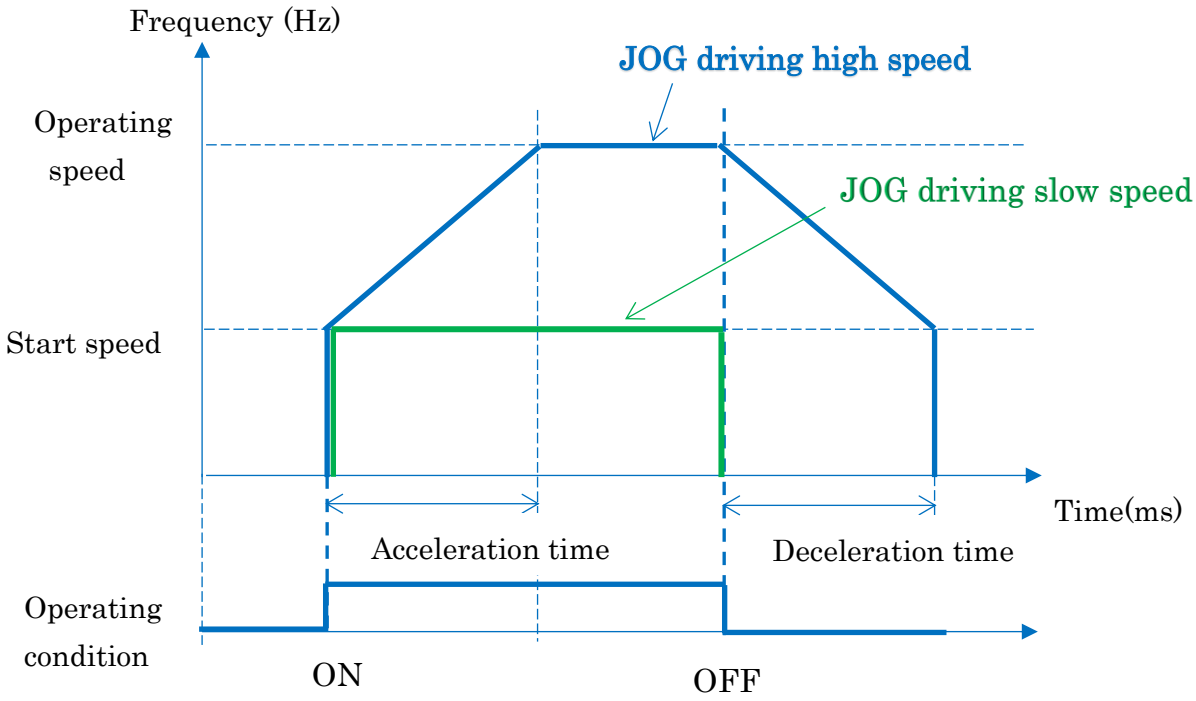
3-2-3 Stepping motor connection example DC 5V



3-2-4 Motor positioning: JOG driving

3-2-4-1 Operation

When the operation condition is ON, it outputs to Y100, Y102 as shown below.



JOG driving

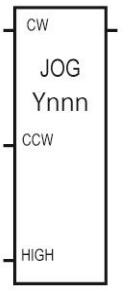
3-2-4-2 Data memory setting

Description	Address	Setting range	Setting Example	Unit
JOG starting speed	DM1103	200～50000	200	Hz
JOG acceleration / deceleration time	DM1104	1～4000	1000	ms
JOG operating speed	DM1105	200～50000	10000	Hz

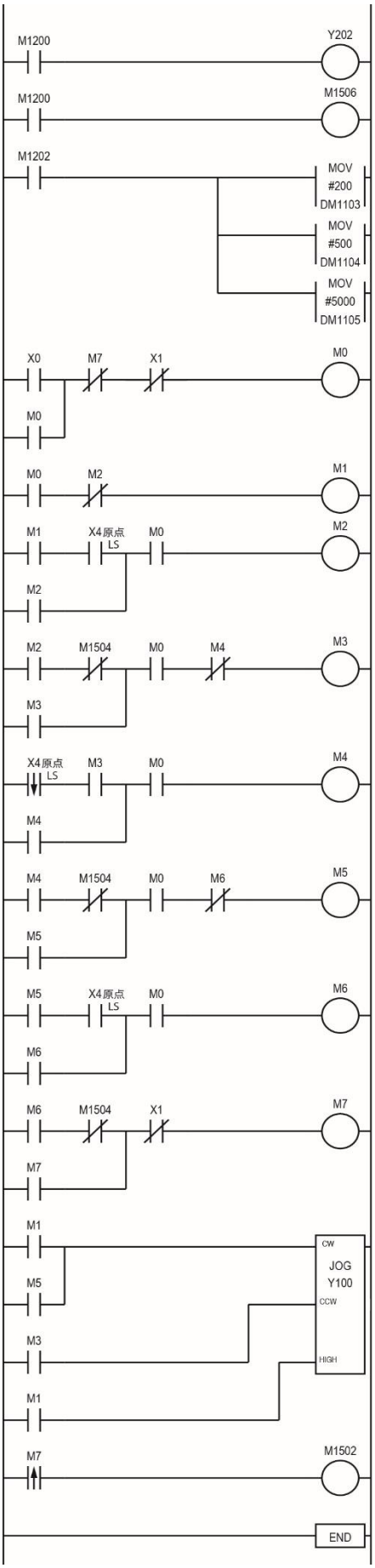
3-2-4-3 Operand of JOG instruction (output destination)

Operand (Ynnn)	Description
Y100	Motor 1
Y102	Motor 2

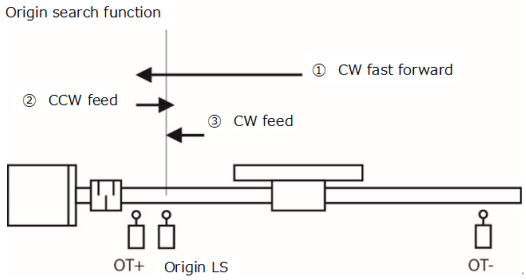
*Y101/Y103 can not be used with the instruction,
because it is automatically assigned by using Y100/Y102.

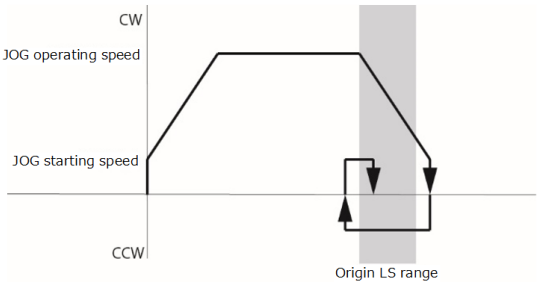


3-2-4-4 Program example: Origin search by JOG




```
LD      M1200
OUT     Y202            // DC 24V ON
OUT     M1506           // 2 pulse system

LD      M1202
MPS
MOV     #200 DM1103     // JOG startup speed
MRD
MOV     #500 DM1104     // JOG Acceleration /
                        // deceleration time
MPP
MOV     #5000 DM1105    // JOG operation speed

LD      X0
OR      M0
ANB     M7
ANB     X1
OUT     M0              // Start of origin search

LD      M0
ANB     M2
OUT     M1              // CW fast forward

LD      M1
AND     X4              // Origin sensor (N.O)
OR      M2
AND     M0
OUT     M2              // CW Fast Forward Stop

LD      M2
ANB     M1504           // Sending pulses to output
OR      M3
AND     M0
ANB     M4
OUT     M3              // CW Feed

LDF     X4
AND     M3
OR      M4
AND     M0
OUT     M4              // CCW stop

LD      M4
ANB     M1504
OR      M5
AND     M0
ANB     M6
OUT     M5              // CW stop

LD      M5
AND     X4
OR      M6
AND     M0
OUT     M6

LD      M6
ANB     M1504
OR      M7
ANB     X1
OUT     M7              // Origin search complete

LD      M1              // JOG CW
OR      M5
LD      M3              // JOG CCW
LD      M1              // JOG HIGH
JOG     Y100

END
```





Origin search start input	X0
Origin LS (Limit Switch)	X4
Output Axis	Y100
Origin search direction	CW

Function:

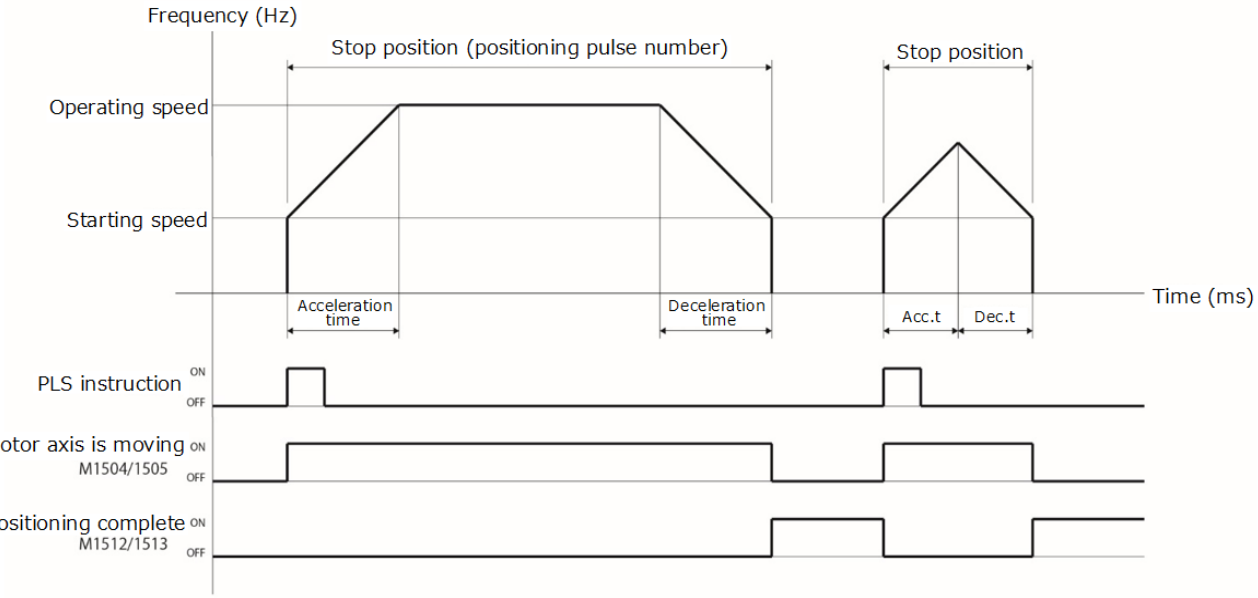
1. Start the origin search with the input of X0.
2. Move at JOG operating speed in the CW direction.
3. Start deceleration when X4 turns ON and continue until Motor Axis stops.
4. Move at JOG starting/slow speed in CCW direction and stop at ON falling edge of X4.
5. Move at JOG starting/slow speed in the CW direction and complete the origin search at X4 ON rising edge.
6. When the origin search is completed, the current Motor Axis Position value is cleared to zero.

* Error occurs when any of start speed, acceleration / deceleration time, operation speed is 0 in data memory.

3-2-5 Trapezoidal control: PLS driving

3-2-5-1 Operation explanation

When the operation condition is ON, it pulses are send to outputs Y100, Y101 / Y102, Y103 as shown below (PLS driving).



PLS driving

3-2-5-2 Data memory setting

Save the driving conditions as 10 action patterns (# 0 to # 9). (Example of pattern #0 below)

Settings	Address	Setting range	Example	Unit
Positioning Mode	DM1000	0: Relative value position, move to Current Position + Stop Position 1: Absolute value position, move to Stop Position	0	
Stop Position	DM1001 (low 16bits)	-2147483648~2147483647 (Refer to 2-7 numerical type for 32bit value setting)	10000	
	DM1002 (high 16bits)		0	
Starting speed	DM1003	200～50000	200	Hz
Acceleration / deceleration time	DM1004	1～4000 (100ms step)	500	ms
Operating speed	DM1005	200～50000	1000	Hz

* When the operation mode is 1 (absolute value position) and the current position is the stop position, the motor will not move even if the operating condition is turned on.

Up to 10 operation patterns can be set.

Positioning data No.	DM (data memory) area
#0	DM1000～DM1005
#1	DM1010～DM1015
#2	DM1020～DM1025
・・・	・・・
#9	DM1090～DM1095

Program example

```
LD   M1200
OUT  Y202

LDP  X0
PLS  #0 Y100         // motor1, pattern #0
LDP  X1
PLS  #0 Y102         // motor2, pattern #0

LD   M1200
MOV  TM940L TM902L   // Display the current position of the motor 1 shaft
                     // on the lower row of the display

END
```

Trapezoidal control of motor

3-3 Load cell control

In addition to measurement of the load, the comparison of the measured value with the target value and checking if the value is within the range can be used.

3-3-1-1 Connection

Connect the power supply of the load cell and the signal wires to the terminal inside the main body.

Terminal	Description
5V	5 V power output
0V	GND
+SIG	Signal+
−SIG	Signal−

3-3-1-2 Load cell setting

Specification of load cell amplifier

Item	Specification
Applied voltage	5V
resolution	24bits
Signal input range	±19.5mV （5÷128÷2）
Sampling cycle	around 20ms
Average processing	5 sample moving average
Capacity	1N～500KN（0.1kgf～50tf）

1N≒0.102kgf

Input voltage - A/D conversion value

Input SIG	A/D conversion value (hexadecimal)	A/D conversion value (decimal)
19.5mV	7FFFFFh	8388607
0V	000000h	0
-19.5mV	800000h	-8388608

3-3-1-3 Data memory setting

Load cell: In case of rated input 5kgf, rated output 2mV

Device	Description	Setting Example	Setting range	Unit
DM1210	Rated capacity	5000	1～65535	g : gf kg : kgf
DM1212	Rated output voltage	2	1~3	mV/V
DM1213	Offset value (Automatic setting, change prohibited)			
DM1214				
DM1215	Unused			
DM1216	Load target value	500	0~65535	g / kg
DM1217	Load upper limit	600	0~65535	g / kg
DM1218	Load lower limit value	400	0~65535	g / kg

3-3-1-4 Load cell control Special auxiliary relay (M)

Memory Bit	Description	Read	Write
M1400	Load cell control start	○	○
M1401	Load average value > Load target value	○	×
M1402	Load average value > Load upper limit value	○	×
M1403	Load average value < Load lower limit value	○	×
M1404	Load Zero Set (reset with pulse (LDP))	×	○

When the Load Zero Set(M1404) is ON, the offset value is updated with the average measured Load Value and so the Load Average Value become 0.

3-3-1-5 Load measurement data in Temporary memory (TM)

Temporary memory		Description
TM930L (32bit)	TM930 (low 16bits)	Load cell A/D conversion value (raw data)
	TM931 (high 16bits)	
TM932L (32bit)	TM932 (low 16bits)	Load conversion value (without offset, unit: 0.1g/kg)
	TM933 (high 16bits)	
TM934L (32bit)	TM934 (low 16bits)	Load Average Value (offset included, unit: 0.1g/kg)
	TM935 (high 16bits)	

Load average value is a moving average from last 5 samples.

3-3-1-6 Unit designation Temporary memory (TM)

Specify g/kg and display range. (TM 906, 907)

Device	Description	Remarks
TM906	Display lower row unit	Specify a number 4 to 10 from the table below.
TM907	Display upper row unit	Specify a number 4 to 10 from the table below.

Number	Description	Example
4	Signed Integer g	-5g
5	Absolute value g	5g
6	Real number 0.1g	0.5g
8	Signed Integer kg	-5kg
9	Absolute value kg	5kg
10	Real number 0.1kg	0.5kg

The target value, upper limit value and lower limit value are integer numbers in g/kg to display the correct unit use 4, 5, 8 or 9 in TM906 / TM907.
The Load Average Value and the Load conversion value have unit is 0.1g or 0.1kg and so the registers' content is for example 59 for 5.9g/kg. To display the correct unit, use 6 or 10 in TM906 / TM907.

Program example
When M1400 turns ON, measurement starts

```
LD   M1200
OUT  Y202            // 24V ON

LD   M1202           // The First SCAN ON, after OFF
MOV  #6 TM906        // Unit g, one digit after the decimal point (0.1g)

LD   X0
OUT  M1400           // Control start

LD   M1400
OUT  Y0

LD   M1401           // load average value > load target value
OUT  Y1

LD   M1402           // load average value > load upper limit value
OUT  Y2

LD   M1403           // load average value < load lower limit value
OUT  Y3

LD M1200
MOV TM930L TM904L    // load cell AD conversion value to display upper row

LD M1200
MOV TM934L TM902L    // Load Average Value to display lower row

LDP  X1              // Load Zero Set
OUT  M1404

END
```

Load cell control example

Show measured value in g/kg without decimal places
The average weight value is measured at 0.1g or 0.1kg. When you need to show the weight without decimal places you can simply divide the weight by 10 (**Note** a number rounding is not applied, ex. 59/10 the result will be 5 not 6).

Number	Description	Example TM934L = -59		
		Display lower row	TM902L	TM906
4	Signed Integer g	-5g	TM934L/10	4
5	Absolute value g	5g	TM934L/10	5
8	Signed Integer kg	-5kg	TM934L/10	8
9	Absolute value kg	5kg	TM934L/10	9

3-3-1-7 Program example

When M1400 turns ON, measurement starts. The measured value is displayed in grams without decimal places.

```
LD   M1200
OUT  Y202            // 24V ON

LD   M1202           // The first SCAN ON, after OFF
MOV  #4 TM906        // Unit g (grams without decimal places)

LD   X0
OUT  M1400           // Control start

LD   M1400
OUT  Y0

LD   M1401           // load average value > load target value
OUT  Y1

LD   M1402           // load average value > load upper limit value
OUT  Y2

LD   M1403           // load average value < load lower limit value
OUT  Y3

LD   M1200
LDA  TM934L          // Load average value
DIV  #10             // divide by 10
STA  TM902L          // display on lower row

LDP  X1              // Load Zero Set
OUT  M1404

END
```

Load cell control example

3-4 Temperature control

In addition to measurement of the temperature, the comparison of the measured value with the target value and checking if the value is within the range can be used.

3-4-1-1 Connetction

Connect a thermocouple (K type) to the terminal inside the main body.

Terminal name	Description
K+	Thermocouple +
K−	Thermocouple -

3-4-1-2 Data memory setting

Data Memory	Description	Unit
DM1200	Target value	℃
DM1201	Adjustment sensitivity (hysteresis value)	℃
DM1202	Alarm upper limit relative to Target value	℃
DM1203	Alarm lower limit relative to Target value	℃

Upper limit and lower limit values are relative to Target Value.

Example: Settings in data memory when the requested target value is 100 °C., the alarm upper limit value is 110 °C., and the alarm lower limit value is 95 °C.
DM1200: 100 DM1202: 10 DM1203: 5

3-4-1-3 Temperature control Special auxiliary relay (M)

Memory bit	Description	Read	Write
M1300	Temperature control start	○	○
M1301	Temperature average value < (Target value - Sensitivity adjustment) **Note**: Only if Error is 0.	○	×
M1302	Error (ex. Thermocouple disconnection)	○	×
M1303	Temperature average value > Upper limit value	○	×
M1304	Temperature average value < Lower limit value	○	×
M1305	Lower limit value < Temperature average value < Upper limit value	○	×

3-4-1-4 Temperature measurement data in Temporary memory (TM)

Temporary memory	Description
TM920	Temperature from Thermocouple-to-Digital Converter (°C, direct value)
TM921	Thermocouple-to-Digital Converter internal substrate temperature (°C)
TM923	Temperature average value Unit: 0.1°C

Temperature average value is a moving average from last 5 samples.

3-4-1-5 Unit designation Temporary memory (TM)

Specify °C and display range. (TM 906, 907)

Device	Description	Remarks
TM906	Display lower row unit	Designate number 32 to 34 from the table below.
TM907	Display upper row unit	Designate number 32 to 34 from the table below.

Number	Description	Example
32	Signed Integer ℃	5℃
33	Absolute value℃	5℃
34	Real number 0.1℃	0.5℃

The target value, Adjustment sensitivity, upper limit value and lower limit value are unsigned 16bit integer numbers in °C to display the correct unit use 32 or 33 in TM906 / TM907.
Temperature average value have unit is 0.1°C and so the register's content is for example 59 for 5.9°C. To display the correct unit, use 34 in TM906 / TM907.

3-4-1-6 Program example

When M1300 turns ON, measurement starts

```
LD   M1200
OUT  Y202            // 24V ON

LD   M1202           // The first SCAN ON, after OFF
MOV  #34 TM906       // Unit 0.1 °C (display lower row)
MOV  #32 TM907       // Unit °C (display upper row)

LD   X0
OUT  M1300           // Control start

LD   M1301           // temperature average value < (target value-sensitivity
                     // adjustment)
OUT  Y1

LD   M1301
OUT  Y201            // AC control power output
LD   M1302
OUT  Y2
LD   M1303
OUT  Y3
LD   M1304
OUT  Y4

LD   M1305
OUT  Y5

LD   X0
MOV  TM923 TM902L    // Temperature average value to display lower row
LD   X0
MOV  DM1200 TM904L   // Display target value to upper row

END
```

Show measured value in °C without decimal places
The temperature average value is measured at 0.1°C. When you need to show the temperature without decimal places you can simply divide the temperature by 10 (**Note** a number rounding is not applied, ex. 59/10 the result will be 5 not 6).

Number	Description	Example TM923 = 59		
		Display lower row	TM902L	TM906
32	Signed Integer ℃	5℃	TM923/10	32
33	Absolute value℃	5℃	TM923/10	33

Program example
When M1300 turns ON, measurement starts. The measured value is displayed in ℃ without decimal places.

```
LD   M1200
OUT  Y202            // 24V ON

LD   M1202           // The First SCAN ON, after OFF
MOV  #32 TM906       // Unit °C (display lower row)
MOV  #32 TM907       // Unit °C (display upper row)

LD   X0
OUT  M1300           // Control start

LD   M1301           // Temperature avr. value < (target value – sensitivity
                     // adjustment)
OUT  Y1

LD   M1301           // AC control power output
OUT  Y201

LD   M1302
OUT  Y2

LD   M1303
OUT  Y3

LD   M1304
OUT  Y4

LD   M1305
OUT  Y5

LD   X0
LDA  TM923           // load Temperature average value
DIV  #10             // divide by 10
STA  TM902L          // Display temp. value/10 to display lower row
LD   X0
MOV  DM1200 TM904L   // Display target value to upper row

END
```

3-5 Encoder

Inputs from X100 to X103 can be used to connect encoder or as input relays.
If the encoder is connected, the encoder data are located in temporary memory.

3-5-1-1 Connection

Terminal	Description
X100	Encoder 1 phase A
X101	Encoder 1 phase B
X102	Encoder 2 phase A
X103	Encoder 2 phase B

3-5-1-2 Encoder control - Special auxiliary relay (M)

Memory Bit	Description	Read	Write
M1507	Encoder Control start	○	○
M1508	Clear value of the encoder 1	○	○
M1509	Clear value of the encoder 2	○	○
M1510	Reverse a count direction of the encoder 1	○	○
M1511	Reverse a count direction of the encoder 2	○	○

By default, CW rotation is counting up. If Reverse a count direction is ON then it counts down.

3-5-1-3 Encoder data in Temporary memory (TM)

Temporary memory		Description
TM944L (32bit)	TM944 (low 16bits)	Encoder 1 value
	TM945 (high 16bits)	
TM946L (32bit)	TM946 (low 16bits)	Encoder 2 value
	TM947 (high 16bits)	

Program example

When M1507 turns ON, encoder control/counting starts.

```
LD   M1200
OUT  Y202            // 24V ON

LD   X0
OUT  M1507           // Encoder control start

LDP  X1
OUT  M1508           // Clear value of encoder1
LDP  X1
OUT  M1509           // Clear value of encoder2

LD   X2
OUT  M1510           // Encoder1 count direction reversal
LD   X2
OUT  M1511           // Encoder2 count direction reversal

LD   X0
MOV  TM944L TM902L   // Encoder1 value to display lower row
LD   X0
MOV  TM946L TM904L   // Encoder2 value to display upper row

LDB  X0
AND  X100            // Use X100 as input relay
OUT  Y0
LDB  X0
AND  X101            // Use X101 as input relay
OUT  Y1
LDB  X0
AND  X102            // Use X102 as input relay
OUT  Y2
LDB  X0
AND  X103            // Use X103 as input relay
OUT  Y3

END
```

Encoder control example

3-6 Other features

3-6-1 Front panel General-purpose buttons

The General-purpose buttons correspond to the input relay X200 to X203

General-purpose button	S1	S2	S3	S4
Input relay	X200	X201	X202	X203

Program example

```
LD   M1200
OUT  Y202

LD   X200
OUT  Y0
LD   X201
OUT  Y1
LD   X202
OUT  Y2
LD   X203
OUT  Y3

END
```

Example of a usage of General-purpose buttons

3-6-2 Front panel Potentiometers

3-6-2-1 Temporary memory (result storage address)

Temporary memory	Description	Range
TM910	Potentiometer VR1	0～10; 0~DM1190
TM911	Potentiometer VR2	0～10; 0~DM1191
TM912	Potentiometer VR3	0～10; 0~DM1192

Program example

Turning potentiometers 1 and 2 changes the display value.

```
LD   M1200
OUT  Y202

LD   M1200
MOV  TM910 TM902L    // Display VR1 to display lower row
LD   M1200
MOV  TM911 TM904L    // Display VR2 to display upper row

END
```

Show values of potentiometers 1 and 2

3-6-2-2 Data memory (VR maximum settings)

By default, values of potentiometers VR1, VR2 and VR3 are mapped to the range 0~10. The maximal value of the range can be changed by a setting of the appropriate data memory register (viz. following table).
The maximal value of VR1, VR2 or VR3 is recommended to be set in range 1~100. If it is set to 0 (zero), the default value 10 will be used. Note the higher value of the maximum the higher drift of the VR value.

Data Memory	Description
DM1190	The maximum value of VR1
DM1191	The maximum value of VR2
DM1192	The maximum value of VR3

3-6-3 Buzzer

3-6-3-1 Buzzer

Beep (1kHz) appears when M1515 is ON.

Buzzer control - Special auxiliary relay (M)

Memory Bit	Description	Read	Write
M1515	Beep ON	○	○

Program example

```
LD   M1200
OUT  Y202

LD   X0
OUT  M1515

END
```

Buzzer control example

4 Device Organization

A device element (relay, timer, data memory, etc.) is used as the operand of the instruction in the program.
From the firmware version 2.05 the internal registers of smartPLC can be monitored and edited from PC connected via USB. An FS Ladder version 2 application that provides the functionality can be downloaded from https://smartplc.org/about/downloadsEN.html. The USB driver has to be also installed (refer to 6 - 1).

4 - 1 Device bit memory and relays

4 - 1 - 1 I / O relay

Bit device name	Range	Description	At reset time	Read	Write
Input relay X	X0~X15	General-purpose input	Clear	○	×
	X100	General-purpose input / Encoder 1 Phase A	Clear	○	×
	X101	General-purpose input / Encoder 1 Phase B	Clear	○	×
	X102	General-purpose input / Encoder 2 Phase A	Clear	○	×
	X103	General-purpose input / Encoder 2 Phase B	Clear	○	×
	X200~ X203	Front Panel Buttons	Clear	○	×
	X204	General-purpose input	Clear	○	×
Output relay Y	Y0~15	General-purpose output	Clear	○	○
	Y100	1 pulse system: Motor 1 PLS output 2 pulse system: Motor 1 CW	Clear	○	○
	Y101	1 pulse system: Motor 1 DIR output 2 pulse system: Motor 1 CCW	Clear	○	○
	Y102	1 pulse system: Motor 2 PLS output 2 pulse system: Motor 2 CW	Clear	○	○
	Y103	1 pulse system: Motor 2 DIR output 2 pulse system: Motor 2 CCW	Clear	○	○
	Y200	General-purpose output	Clear	○	○
	Y201	AC power supply control output	Clear	○	○
	Y202	24V output	Clear	○	○
Timer T	T0~T99	Timer (TMR, TMH, TMS shared) 100 bits	Clear	○	×
Counter C	C0~C99	Counter 100 bits	Clear	○	×
Control memory CR (*1)	CR000- CR015 CR100- CR115 CR200- CR215 ... CR900- CR915	Control memory 160 bits (16 bits x 10 channels)	Hold (*3)	○	○

(*1): Control memory has two types of access CM for a word access and CR for a bit access. For example, CM0 shares the memory with CR0 ~ CR15

Example for CM0:

CM0	CR15 (CM0 MSB)	CR14	.....	CR2	CR1	CR0 (CM0 LSB)

4-1-2 Internal auxiliary relay - Special auxiliary relay

Bit device name	Range	Description	At reset time	Read	Write
Internal auxiliary relay	M000-M015 … M1100-M1115	User area 192 bits (*2)	Clear	○	○
Special auxiliary relay	M1200	Always ON	0→1(*5)	○	×
	M1201	Always OFF	Clear	○	×
	M1202	The first scan ON, else OFF	Clear	○	×
	M1203~M1209	Unavailable		×	×
	M1210	The internal register < CMP operand	Clear	○	×
	M1211	The internal register = CMP operand	Clear	○	×
	M1212	The internal register > CMP operand	Clear	○	×
	M1213~M1214	Unavailable		×	×
	M1215	Show info page (defined by TM908) on display	Clear	○	○
	M1300	Temperature control start	Clear	○	○
	M1301	Temperature average value < (Target value - Sensitivity adjustment) **Note**: Only if Error is 0.	Clear	○	×
	M1302	Error (ex. Thermocouple disconnection)	Clear	○	×
	M1303	Temperature average value > Upper limit value	Clear	○	×
	M1304	Temperature average value < Lower limit value	Clear	○	×
	M1305	Lower limit value < Temperature average value < Upper limit value	Clear	○	×
	M1306~M1315	Unavailable			
	M1400	Load cell control start	Clear	○	○
	M1401	Load average value > Load target value	Clear	○	×
	M1402	Load average value > Load upper limit value	Clear	○	×
	M1403	Load average value < Load lower limit value	Clear	○	×
	M1404	Load Zero Set (ON to Load average value = 0)	Clear	○	○
	M1405~M1415	Unavailable		×	×
	M1500	Immediately stop sending the pulses to Motor 1.	Clear	○	○
	M1501	Immediately stop sending the pulses to Motor 2.	Clear	○	○
	M1502	Reset the current position counter of Motor 1	Clear	○	○
	M1503	Reset the current position counter of Motor 2	Clear	○	○
	M1504	Sending the pulses to Motor 1.	Clear	○	×
	M1505	Sending the pulses to Motor 2.	Clear	○	×
	M1506	ON - 2 pulse mode, OFF - 1 pulse mode (Default value: OFF)	Clear	○	○
	M1507	Encoder Control start	Clear	○	○
	M1508	Clear value of the encoder 1	Clear	○	○
	M1509	Clear value of the encoder 2	Clear	○	○
	M1510	Reverse a count direction of the encoder 1	Clear	○	○
	M1511	Reverse a count direction of the encoder 2	Clear	○	○
	M1512	Motor 1 positioning complete (PLS instruction only)	Clear	○	×
	M1513	Motor 2 positioning complete (PLS instruction only)	Clear	○	×
	M1514	Unavailable		×	×
	M1515	ON - Beep On, OFF - Beep Off	Clear	○	○
Internal auxiliary relay	M1600-M1615 … M4900-M4915	User area 544 bits (*2) (192 bits + 544 bits = 736 bits in total)	Clear	○	○

(*2): The user area bits can be freely used in the program; the rest of bit memory is dedicated to the specific use.
(*5): During the reset sequence the last value of M1200 is set OFF and the actual value is set ON. In the first scan LDP M1200 is ON and after is OFF. (Before v2-03 2018-06-05 LDP M1200 was only OFF).

4-2 Device word memory registers

4-2-1 Temporary memory

Word device name	Address	Description	Value	At reset time	R	W
Temporary memory	TM0~TM99	User area 100 words (*2)	0~65535	Clear	○	○
	TM900	Display lower row - Mode selection	0~27, 1000	Clear	○	○
	TM901	Display upper row - Mode selection	0~27, 1000	Clear	○	○
	TM902(low 16b)	Display lower row - Numerical value	0~65535	Clear	○	○
	TM903(high 16b)		0~65535	Clear	○	○
	TM904(low 16b)	Display upper row - Numerical value	0~65535	Clear	○	○
	TM905(high 16b)		0~65535	Clear	○	○
	TM906	Display lower row - Unit	0~35	Clear	○	○
	TM907	Display upper row - Unit	0~35	Clear	○	○
	TM908	Info page to be displayed (if M1215)	0~8	Clear	○	○
	TM909	Unavailable		—	×	×
	TM910	Potentiometer VR1 value	0~10 (*4)	Clear	○	×
	TM911	Potentiometer VR2 value	0~10 (*4)	Clear	○	×
	TM912	Potentiometer VR3 value	0~10 (*4)	Clear	○	×
	TM913	Unavailable		—	×	×
	TM914~TM917	Display lower row - User defined characters	0~65535	Clear	○	○
	TM918~TM919	Unavailable		—	×	×
	TM920	Temperature from Thermocouple-to-Digital Converter (℃, direct value)	0~65535	Clear	○	×
	TM921	Thermocouple-to-Digital Converter internal substrate temperature (℃)	0~65535	Clear	○	×
	TM923	Temperature average value (0.1℃)	0~65535	Clear	○	×
	TM924~TM927	Display upper row - User defined characters	0~65535	Clear	○	○
	TM928~TM929	Unavailable		—	×	×
	TM930(low 16b)	Load cell A/D conversion value (raw data)	0~65535	Clear	○	×
	TM931(high 16b)		0~65535	Clear	○	×
	TM932	Load conversion value (without offset, unit: 0.1g / 0.1kg)	0~65535	Clear	○	×
	TM933		0~65535	Clear	○	×
	TM934	Load Average Value (offset included, unit: 0.1g / 0.1kg)	0~65535	Clear	○	×
	TM935		0~65535	Clear	○	×
	TM936~TM939	Unavailable		—	×	×
	TM940(low 16b)	Motor 1 current position counter	0~65535	Clear	○	×
	TM941(high 16b)		0~65535	Clear	○	×
	TM942(low 16b)	Motor 2 current position counter	0~65535	Clear	○	×
	TM943(high 16b)		0~65535	Clear	○	×
	TM944 (low 16b)	Encoder 1 value	0~65535	Clear	○	×
	TM945 (high 16b)		0~65535	Clear	○	×
	TM946	Encoder 2 value	0~65535	Clear	○	×
	TM947		0~65535	Clear	○	×
Control memory (*1)	CM0-CM9	Control memory 10 words	0~65535	Hold (*3)	○	○

(*3): Data is stored even when the power is turned off.
(*4): Default maximal value, it can be overridden viz. 3-6-2-2 Data memory (VR maximum settings)

4-2-2 Data memory

Word device name	Address	Description	Value	At reset time	R	W
Data memory	DM0000~DM999 DM1300~DM1999	User area 1700 words (*2)	0~65535	Hold (*3)	○	○
	DM1000	Pattern # 0 Positioning mode	0: Relative Stop position 1: Absolute Stop position	Hold	○	○
	DM1001(low 16b)	Pattern # 0 Stop position	-2147483648 ~ 2147483647	Hold	○	○
	DM1002(high 16b)					
	DM1003	Pattern # 0 Starting speed (Hz)	200~50000	Hold	○	○
	DM1004	Pattern # 0 Acceleration / deceleration time (ms)	1~4000	Hold	○	○
	DM1005	Pattern # 0 Operating speed (Hz)	200~50000	Hold	○	○
	DM1010~DM1015	pattern #1		Hold	○	○
	DM1020~DM1025	pattern #2		Hold	○	○
	DM1030~DM1035	pattern #3		Hold	○	○
	DM1040~DM1045	pattern #4		Hold	○	○
	DM1050~DM1055	pattern #5		Hold	○	○
	DM1060~DM1065	pattern #6		Hold	○	○
	DM1070~DM1075	pattern #7		Hold	○	○
	DM1080~DM1085	pattern #8		Hold	○	○
	DM1090~DM1095	pattern #9		Hold	○	○
	DM1103	JOG Starting speed (Hz)	200~50000	Hold	○	○
	DM1104	JOG Acceleration / deceleration time (ms)	1~4000	Hold	○	○
	DM1105	JOG Operating speed (Hz)	200~50000	Hold	○	○
	DM1190	VR1 - Maximum	0~65535 (*4)	Hold	○	○
	DM1191	VR2 - Maximum	0~65535 (*4)	Hold	○	○
	DM1192	VR3 - Maximum	0~65535 (*4)	Hold	○	○
	DM1200	Temperature control Target value (° C)	0~1000	Hold	○	○
	DM1201	Temperature control Adjustment sensitivity (℃)	0~100	Hold	○	○
	DM1202	Temperature control Alarm upper limit (℃)	0~1000	Hold	○	○
	DM1203	Temperature control Alarm lower limit (℃)	0~1000	Hold	○	○
	DM1204~DM1209	Unavailable		—	×	×
	DM1210	Load cell Rated capacity (g/kg)	0~65535	Hold	○	○
	DM1212	Load cell rated output (mV/V)	0~3	Hold	○	○
	DM1213(low 32b)	Load cell offset value	0~65535	Hold	○	×
	DM1214(high 32b)		0~65535	Hold	○	×
	DM1215	Usage prohibited		—	×	×
	DM1216	Load cell Target load value (g/kg)	0~65535	Hold	○	○
	DM1217	Load cell Load upper limit value (g/kg)	0~65535	Hold	○	○
	DM1218	Load cell Load lower limit (g/kg)	0~65535	Hold	○	○

(*2): The user area words can be freely used in the program; the rest of data memory words are dedicated to the specific use.
(*3): Data is stored even when the power is turned off.
(*4): The maximal value of VR1, VR2 or VR3 is recommended in range 1~100. (see 3-6-2-2)

5 Program and instructions

5 - 1 Program

5 - 1 - 1 Program notation rules

· The maximum number of characters per line is 500 characters (full-width characters are counted as 2 characters each). If it exceeds, an error will be displayed.
· Comments are started by two slashes "//" and ended with end of line. The characters used in the program are as shown in the table below.
· END instruction is required at the end of the program.

Object	Characters
Program	alphanumeric characters, Hash, Slash, Double Quotation Mark, Backslash
Comment	Half size, full width (including Japanese)

· Settings of text editor
In case of troubles (ERR#002 – Char???), Check the character encoding of program.txt.

Character encoding	Line feed code
SHIFT-JIS, ANSI (initial value of Notepad)	CR＋LF

8bit codes of important characters used in program.txt

Glyph		Code decimal	Code hexadecimal	Name	
		09	09	Horizontal Tab	
		32	20	Space	
"		34	22	Double Quotation Mark	
#		35	23	Hash / sharp sign	
/		47	2F	Slash	
\	¥	92	5C	Backslash	en character

5 - 1 - 2 Instruction

・Instruction name can be followed with one, two or three operands (depends on instruction type) and also it can be followed with "//" and comment
・There have to be a Space or Tab between the instruction and operands.
・There can be maximum one instruction per a program line. **Note**: The second instruction on the line will rise error.

5 - 1 - 3 Operand

・Operand name is made from operand type followed with its value without space between.
・Operand type for the numeric constant, is marked by hash (sharp sign) "#". A signed 32bit numeric constant can be marked by 'L' on the end. Examples: `#0, #1L, #10, #-10, #-5L, #65535, #65536`
・The operand name of word registers and numeric constants can be ended by letter 'L' to indicate 32bit access/number. Examples: `#-1L, TM940L, DM1001L`
・The order of operands following instruction name is very important and will be explained for each instruction in following section
・The number of operands depends on the instruction. The maximum is 3 (CNT), and the minimum 0 (MPP, ANL, …).
・Examples: `#1, #-1L, X10, Y2, T11, C10, M1200, CR3, CM0, CM0L, TM0, TM902L, DM3, DM10L`

5-1-3-1 Synchronization of the operands

To make clear the program behavior, it is important to know when the operand actual value changes. Some of the registers and bit relays are buffered (last, actual, next value buffers) and so the synchronization time depends on the operand type. During the program instruction processing, the instruction can update directly the actual value buffer (ex. TM, T, C) or the buffer with the next scan values (ex. M, Y). Data from the buffer with the next scan values are copied to the buffer with the actual values at the program scan end, see 1-3-2-1 Sequence of the scan.

Operand type	The time point when the actual value changes
TM	Instruction
DM	Instruction (including save to the permanent memory)
CM	Instruction (save to the permanent memory during the Scan End) Note: CR will reflect the change during the Scan End.
T	Instruction
C	Instruction
X	Scan Init (device inputs are read during the Scan Init)
Y	Scan End (device outputs will be updated during the Scan End)
M	Scan End, Note: M1210~M1212 change during CMP instruction and during Scan End are reset to zero.
CR	Scan End Note: CM will reflect the change immediately during the Instruction processing.

5-1-4 Example of a program notation

```
// You can include blank lines and comment lines.
LD          X0          // Full-width alphanumeric character ABC 123
[Instruction]  [Operand 1]  [Comment]


// ******** Timer instruction ***********
TMR         #10         TM0         // Alphabetical comment
[Instruction]  [Operand 1]  [Operand 2]  [Comment]


// You can put tabulator and space characters at the beginning of program line,
// between instruction and operands and at the end.
CNT         #0          #10         X0          //   Indentation is also possible
[Instruction]  [Operand 1]  [Operand 2]  [Operand 3]  [Comment]

END

// If there is no END instruction at the end of the program,
// an error will occur (Err#001 - END???)
// Comment line is possible even after END.
// smartPLC internal compiler does not read the content of program.txt after the
// program line with the END instruction
```

5 - 2 Instructions

5 - 2 - 1 Basic instructions

LD Load

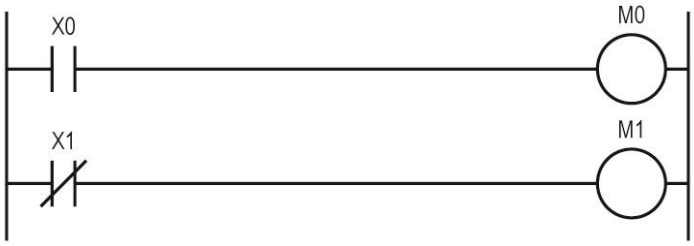
Connect an A contact (N.O – normally open) to the beginning of the circuit.

LDB Load bar

Connect a B contact (N.C – normally closed) to the beginning of the circuit.

Program example

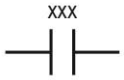
Instruction	Operand
LD	X0
OUT	M0
LDB	X1
OUT	M1



Operand xxx	
I / O relay	X, Y
Internal auxiliary relay	M
Timer	T
Counter	C
Control Memory	CR

When the input relay X0 is ON, the output relay M0 turns ON.
When the input relay X1 is OFF, the output relay M1 turns ON.
LD, LDB are the beginning of the circuit.

AND And



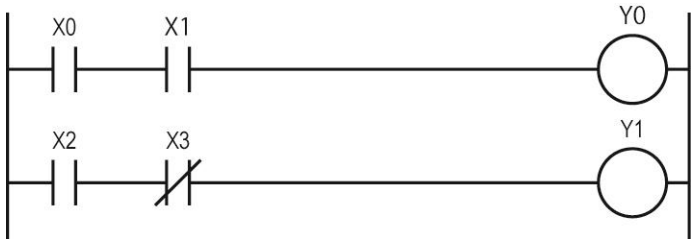
Connect an A contact (N.O) in series.

ANB And bar

Connect a B contact (N.C) in series.

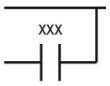
Program example

Instruction	Operand
LD	X0
AND	X1
OUT	Y0
LD	X2
ANB	X3
OUT	Y1

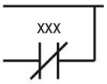


Operand xxx	
I / O relay	X, Y
Internal auxiliary relay	M
Timer	T
Counter	C
Control Memory	CR

When the input relay X0 is ON and X1 is ON, the output relay Y0 turns ON.
When the input relay X2 is ON and X3 is OFF, the output relay Y1 turns ON.
There is no limit to the number of contacts connected in series.

OR **Or** 

Connect an A contact (N.O) in parallel.

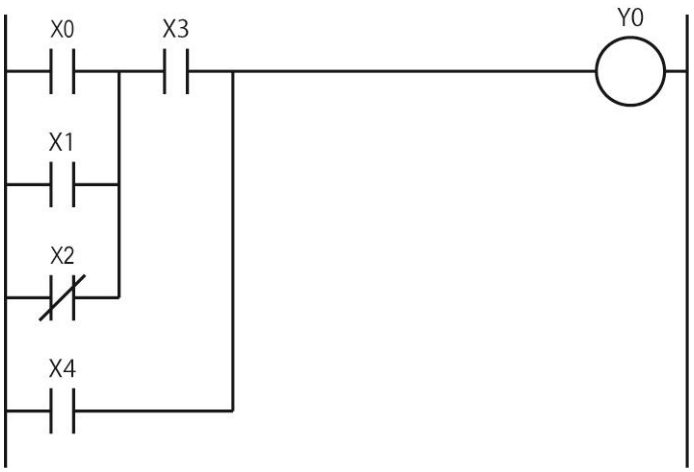
ORB **Or bar** 

Connect a B contact (N.C) in parallel.

Program example

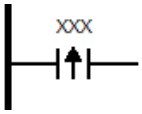
Instruction	Operand
LD	X0
OR	X1
ORB	X2
AND	X3
OR	X4
OUT	Y0

Operand xxx	
I / O relay	X, Y
Internal auxiliary relay	M
Timer	T
Counter	C
Control Memory	CR



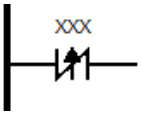
It is connected in parallel to LD and LDB rows.
There is no limit to the number of contacts to be connected in parallel.

LDP Load pulse



Connect an A contact (N.O) of the Raising Edge Detector to the beginning of the circuit.

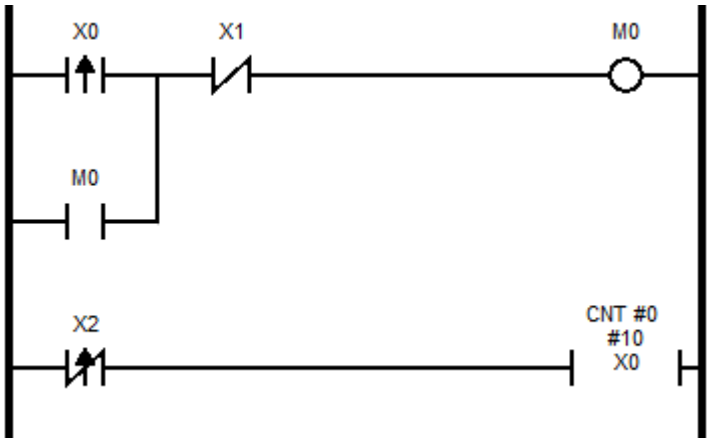
LDPB Load pulse bar



Connect an B contact of the Raising Edge Detector to the beginning of the circuit.

Program example

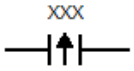
Instructions	Operands
LDP	**X0**
OR	**M0**
ANB	**X1**
OUT	**M0**
LDPB	**X2**
CNT	**#0 #10 X0**



Operand xxx	
I / O relay	X, Y
Internal auxiliary relay	M
Timer	T
Counter	C
Control Memory	CR

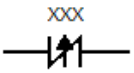
When the rising edge of the operand is detected, the A contact will be ON and the contact B will be OFF for one scan.
LDP, LDPB are the beginning of the circuit.

ANP And pulse



Connect an A contact (N.O) of the Raising Edge Detector in series.

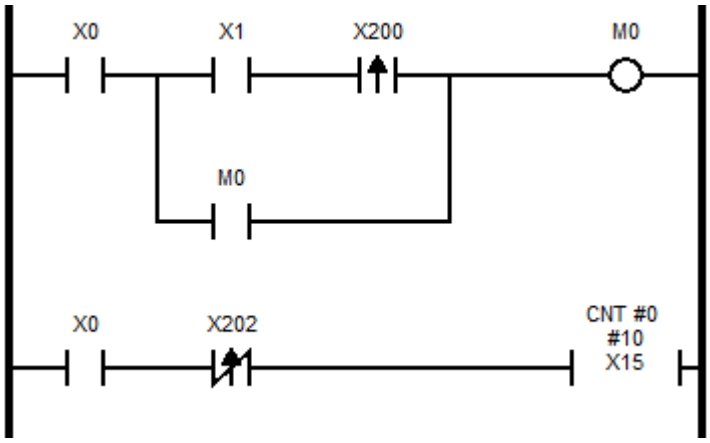
ANPB And pulse bar



Connect an B contact (N.C) of the Raising Edge Detector in series.

Program example

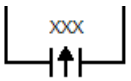
Instructions	Operands
LD	X0
LD	X1
ANP	X200
OR	M0
ANL	
OUT	M0
LD	X0
ANPB	X202
CNT	#0 #10 X15



Operand xxx	
I / O relay	X, Y
Internal auxiliary relay	M
Timer	T
Counter	C
Control Memory	CR

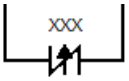
When the rising edge of the operand is detected, the A contact will be ON and the contact B will be OFF for one scan.

ORP Or pulse



Connect an A contact (N.O) of the Raising Edge Detector in parallel.

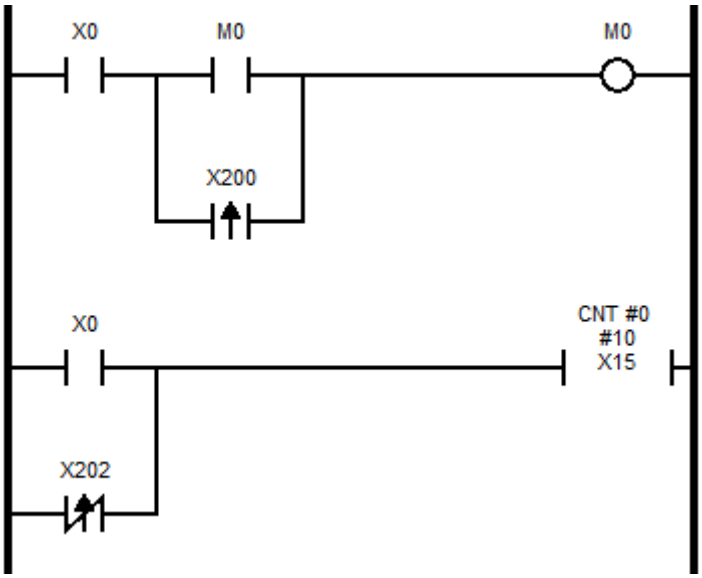
ORPB Or pulse bar



Connect an B contact (N.C) of the Raising Edge Detector in parallel.

Program example

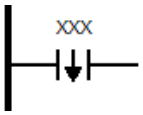
Instructions	Operands
LD	X0
LD	M0
ORP	X200
ANL	
OUT	M0
LD	X0
ORPB	X202
CNT	#0 #10 X15



Operand xxx	
I / O relay	X, Y
Internal auxiliary relay	M
Timer	T
Counter	C
Control Memory	CR

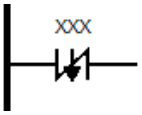
When the rising edge of the operand is detected, the A contact will be ON and the contact B will be OFF for one scan.

LDF Load fall



Connect an A contact (N.O) of the Falling Edge Detector to the beginning of the circuit.

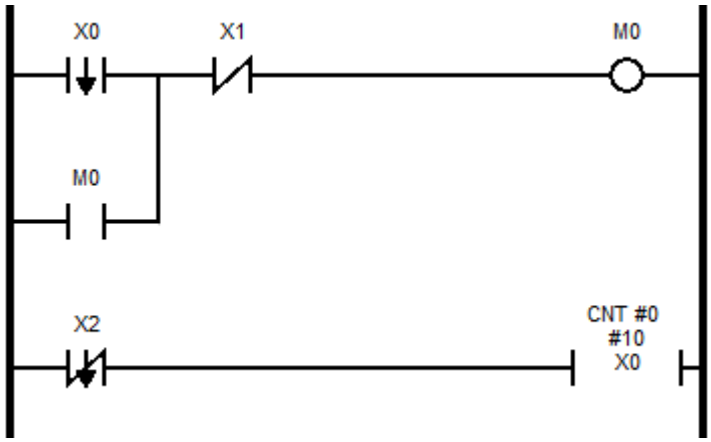
LDFB Load fall bar



Connect an B contact of the Falling Edge Detector to the beginning of the circuit.

Program example

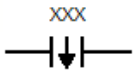
Instructions	Operands
LDF	**X0**
OR	**M0**
ANB	**X1**
OUT	**M0**
LDFB	**X2**
CNT	**#0 #10 X0**



Operand xxx	
I / O relay	X, Y
Internal auxiliary relay	M
Timer	T
Counter	C
Control Memory	CR

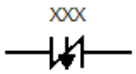
When the falling edge of the operand is detected, the A contact will be ON and the contact B will be OFF for one scan.
LDF, LDFB are the beginning of the circuit.

ANF And fall



Connect an A contact (N.O) of the Falling Edge Detector in series.

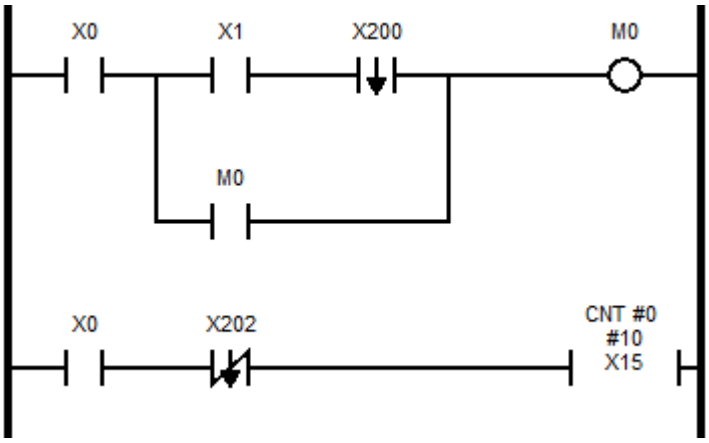
ANFB And fall bar



Connect an B contact (N.C) of the Falling Edge Detector in series.

Program example

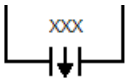
Instructions	Operands
LD	X0
LD	X1
ANF	X200
OR	M0
ANL	
OUT	M0
LD	X0
ANFB	X202
CNT	#0 #10 X15



Operand xxx	
I / O relay	X, Y
Internal auxiliary relay	M
Timer	T
Counter	C
Control Memory	CR

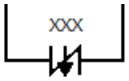
When the falling edge of the operand is detected, the A contact will be ON and the contact B will be OFF for one scan.

ORF Or fall



Connect an A contact (N.O) of the Falling Edge Detector in parallel.

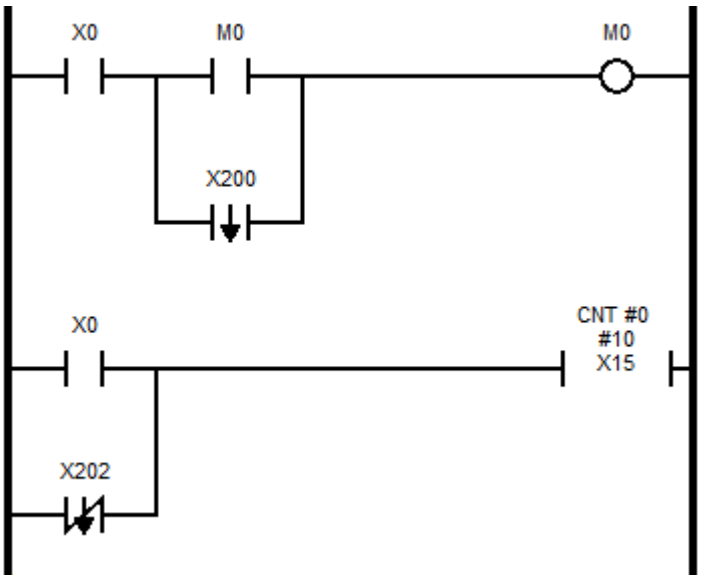
ORFB Or fall bar



Connect an B contact (N.C) of the Falling Edge Detector in parallel.

Program example

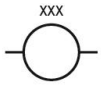
Instructions	Operands
LD	X0
LD	M0
ORF	X200
ANL	
OUT	M0
LD	X0
ORFB	X202
CNT	#0 #10 X15



Operand xxx	
I / O relay	X, Y
Internal auxiliary relay	M
Timer	T
Counter	C
Control Memory	CR

When the falling edge of the operand is detected, the A contact will be ON and the contact B will be OFF for one scan.

OUT Out



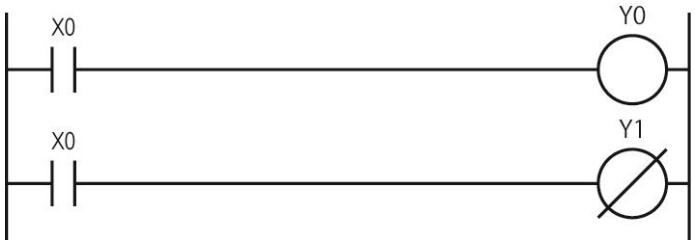
ON / OFF Status of the Circuit to the Output Relay.

OUB Out bar

Inverted ON / OFF Status of the Circuit to the Output Relay.

Program example

Instruction	Operand
LD	X0
OUT	Y0
LD	X0
OUB	Y1

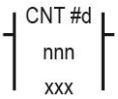


Operand xxx	
Output relay	Y
Internal auxiliary relay	M
Control Memory	CR

When the input relay X0 is ON, the output relay Y0 turns ON and the output relay Y1 turns OFF.

Note:
It is not possible to connect two or more identical output relay numbers.
It will lead to error (Err#017 – 2OUT->1Y).

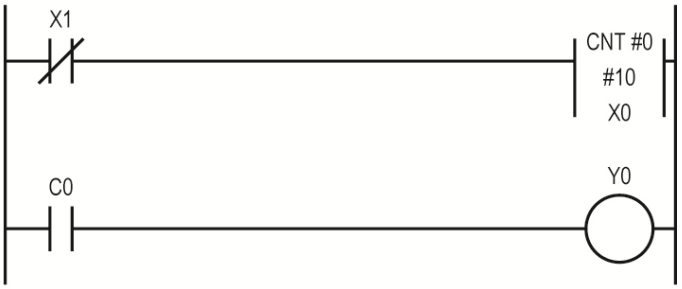
CNT Counter



It is 16bit incremental counter.

Program example

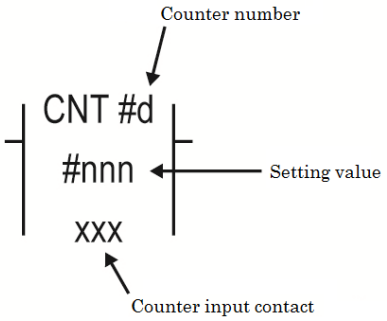
Instruction	Operands
LDB	X1
CNT	#0 #10 X0
LD	C0
OUT	Y0



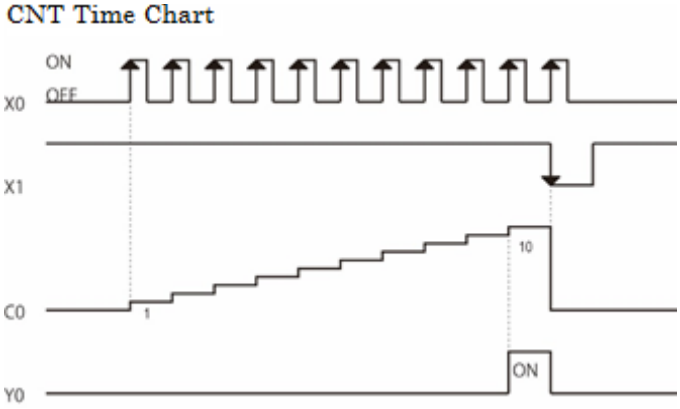
Operand #d	
Counter number	#0～99

Operand nnn	
Setting value	#0～65535, DM, TM

Operand xxx	
Bit relay	X, Y, M, T, C, CR



CNT Time Chart



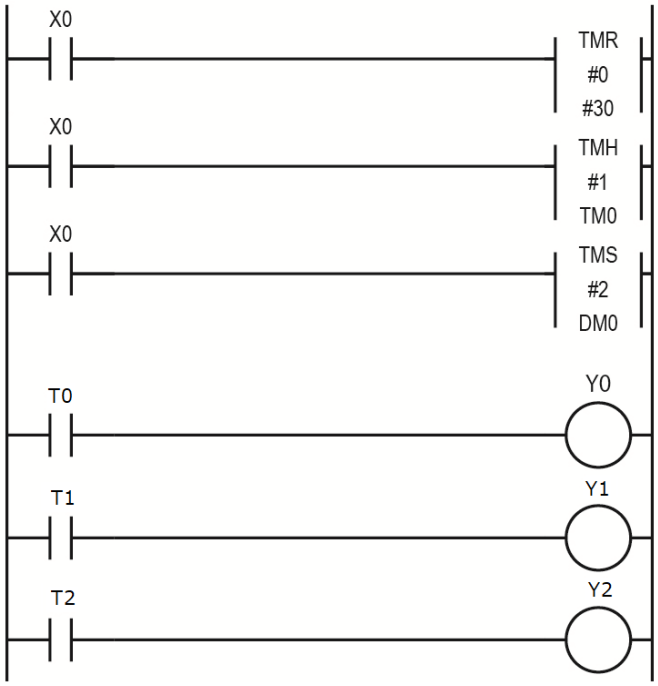
When the X1 is OFF, the counter value is incremented by 1 at every rising edge of the input contact X0.
When the counter value reaches the set value #10, the contact C0 turns ON.
When X1 turns ON during counting, the counter value is reset to zero.

Note:
The Counter Setting value numeric constant is 16bit unsigned integer.

TMR 100ms timer	TMR #d xxx	A timer instruction with delay in 0.1s.
TMH 10ms timer	TMH #d xxx	A timer instruction with delay in 0.01s.
TMS 1ms timer *1	TMS #d xxx	A timer instruction with delay in 0.001s.

Program example

Instruction	Operands
LD	X0
TMR	#0 #30
LD	X0
TMH	#1 TM0
LD	X0
TMS	#2 DM0
LD	T0
OUT	Y0
LD	T1
OUT	Y1
LD	T2
OUT	Y2



Operand #d	
Timer number	#0～99

Operand xxx	
Setting value	#0～65535, DM, TM, CM

When the X0 is ON, the timers are measuring time and the timers contacts will turn ON with their preset values.
When the X0 is OFF, the current values of the timers are reset to zero.
T0: set time = 30 × 100ms
T1: set time = TM0 × 10ms
T2: set time = DM0 × 1ms

Note:
The Timer Setting value numeric constant is 16bit unsigned integer.
TMR, TMH, TMS, TMI shares timer numbers and one timer number can be used only with one timer.

(*1): The accuracy of TMS timer is limited by the scan time, and so the timer setting value is recommended to be 10ms or more. For short time measurements (less than 1s) is recommended to use short programs as they have a shorter scan time and that means higher accuracy.

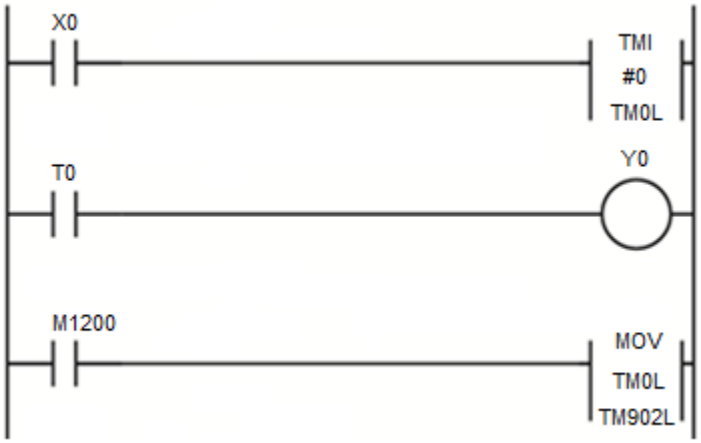
TMI Timer-interval ms

TMI
#d
xxx

A timer measures time in ms (*1).

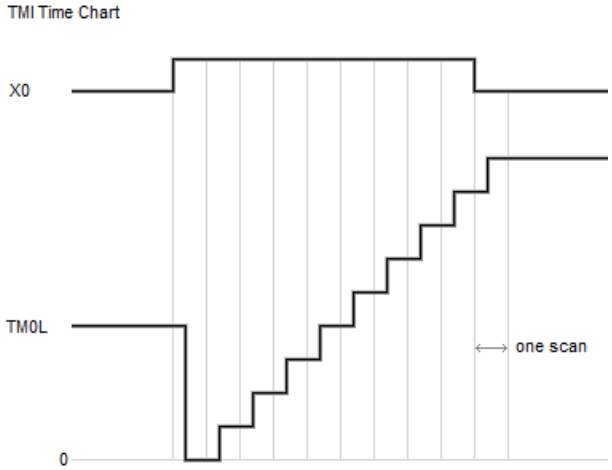
Program example

Instruction	Operands
LD	X0
TMI	#0 TM0L
LD	T0
OUT	Y0
LD	M1200
MOV	TM0L TM902L



Operand #d	
Timer number	#0～99

Operand xxx (32bit only)	
Word device	DMxxL, TMxxL, CMxxL

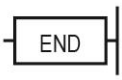


When the X0 is ON, the timer measures time and saves it to the TM0L (TM0, TM1). If the timer reaches the maximum possible value (2147483647ms → about 596hours), the contact T0 will turn ON and timer will stop counting. Value of TM0L is displayed on lower row of the display.
When the X0 is OFF, the last measured value is kept in TM0L.
When a rising edge is detected on X0, TM0L will be set to zero.

Note:
TMR, TMH, TMS, TMI shares timer numbers and one timer number can be used only with one timer.

(*1): The accuracy of TMI timer is limited by the scan time, for short time measurements (less than 1s) is recommended to use short programs as they have a shorter scan time and that means higher accuracy.

END End

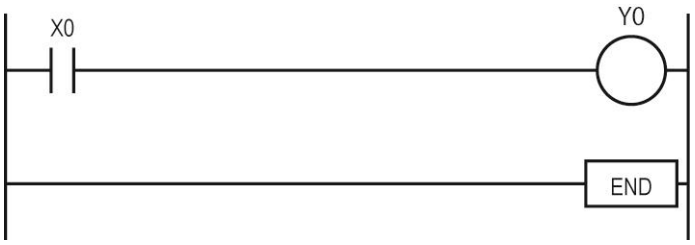


Indicates the end of the program.

Program example

Instruction	Operand
LD	X0
OUT	Y0
END	

No operand



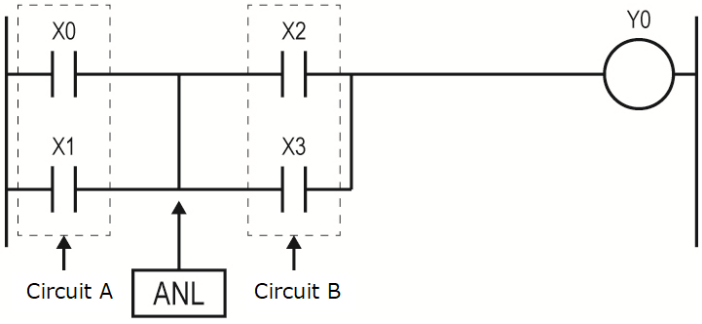
There has to be END instruction at the end of the program.
If there is no END instruction in the program, an error (Err#001 – END???) will occur.

ANL And Load

Connect the blocks of contacts in series.

Program example

Instruction	Operand
LD	X0
OR	X1
LD	X2
OR	X3
ANL	
OUT	Y0



No operand

Connect circuit A and circuit B in series.

Note:
The combination of block circuits can be translated into a sequence of up to 31 consecutive ANL / ORL instructions.

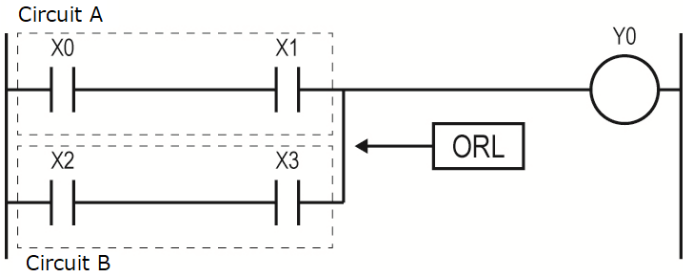
ORL Or Load

Connect the blocks of contacts in parallel.

Program example

Instruction	Operand
LD	X0
AND	X1
LD	X2
AND	X3
ORL	
OUT	Y0

No operand



Connect circuit A and circuit B in parallel.

Note:
The combination of block circuits can be translated into a sequence of up to 31 consecutive ANL / ORL instructions.

5-2-2 Application instructions

DIFU Differentiate UP

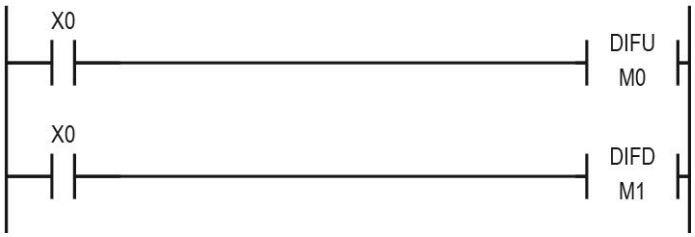
When a rising edge on the input is detected then the Relay specified by the operand is set ON for one scan cycle.

DIFD Differentiate Down

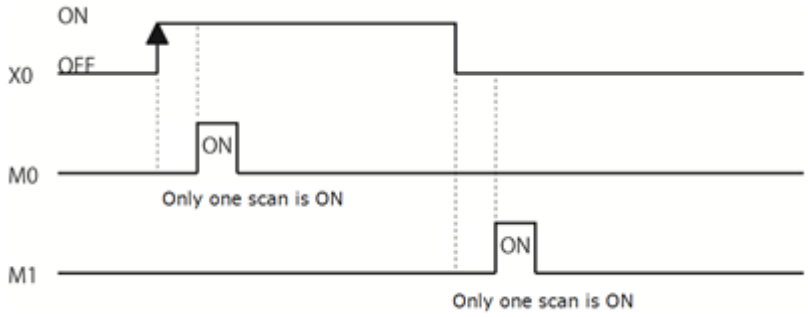
When a falling edge on the input is detected then the Relay specified by the operand is set ON for one scan cycle.

Program example

Instruction	Operand
LD	X0
DIFU	M0
LD	X0
DIFD	M1



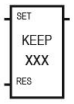
Operand xxx	
Internal auxiliary relay	M



The DIFU instruction turns on M0 for one scan at the rising edge of input X0.
The DIFD instruction turns on M1 for one scan at the falling edge of input X0.

The DIFU and DIFD instructions cannot use two or more identical Internal auxiliary relays. It will lead to error (Err#016 – 2DIF1ADR).

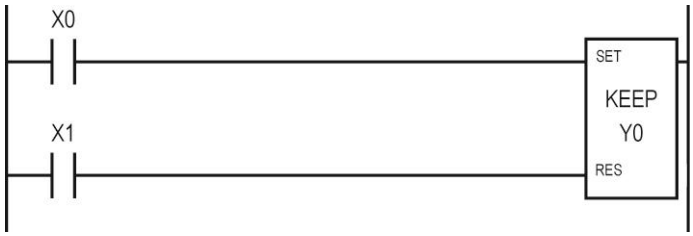
KEEP Keep relay



When input SET is ON, the relay specified by the operand is set ON and when input RES is ON the relay is set OFF.

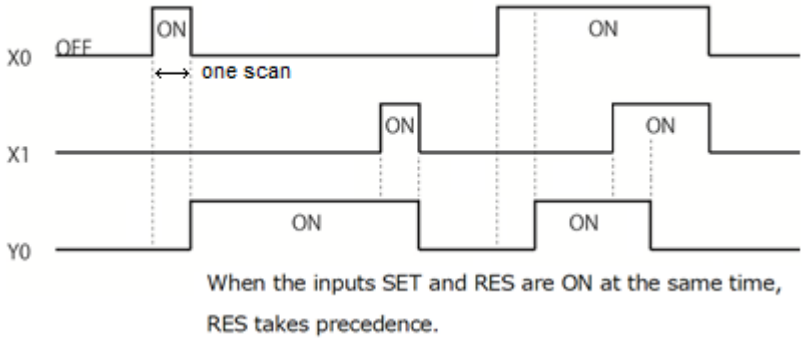
Program example

Instruction	Operand
LD	X0
LD	X1
KEEP	Y0



Operand xxx	
Output relay	Y
Internal auxiliary relay	M
Control Memory	CR

Time chart



When the inputs SET and RES are ON at the same time,
RES takes precedence.

When SET input is ON and RES input is OFF, output relay Y0 turns ON.
Output relay Y0 turns OFF if RES input is ON.

SET Set

SET
xxx

When the input condition is ON then set ON the Relay specified by the operand

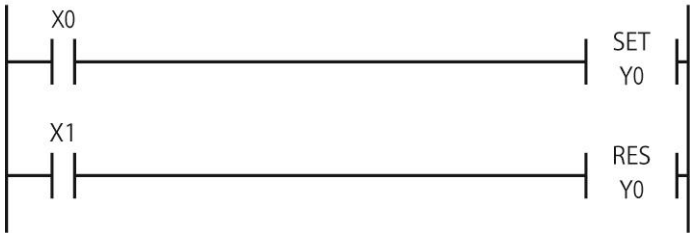
RES Reset

RES
xxx

When the input condition is ON then set OFF the Relay specified by the operand.

Program example

Instruction	Operand
LD	X0
SET	Y0
LD	X1
RES	Y0



Operand xxx	
Output relay	Y
Internal auxiliary relay	M
Control Memory	CR

When the input contact X0 is ON, the output relay turns ON.
When the input contact X1 is ON, the output relay turns OFF.
Multiple SET and RES can be placed on the circuit.

The SET and RES instructions can use identical relays specified by the operand.

MPS Push

Pushes the state of the input to the stack.

MRD Read

Reads the last pushed state of input from the stack.

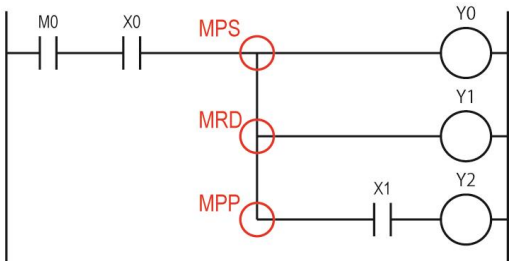
MPP Pop

Pops the last pushed state of input from the stack.

Program example

Instruction	Operand
LD	M0
AND	X0
MPS	
OUT	Y0
MRD	
OUT	Y1
MPP	
AND	X1
OUT	Y2

No operand



A result of input condition defined by contacts M0, X0 is stored by MPS and continues to output relay Y0.
The result is recalled be the MRD instruction and goes to output relay Y1.
On the last branch is the result recalled and removed from stack by the MPP instruction and it is connected in series with the A contact of input relay X1 and finally to output relay Y2

Note:
MPS and MPP are used in pairs.
The maximum number of mutually nested MPS-MPP pairs is 63.

5-2-3 Word register instructions

32bit instruction

5-2-3-1 32bit instruction notation and usage

・32bit instruction are those instructions which allows 32bit operands.
・32bit operand is marked 'L' next to the word register number. It is stored in the marked word register and the following register. For example, TM10L is stored in the temporary memory word registers TM10 (low 16bits) and TM11 (high 16bits).

32bit instruction	
Instruction	Arithmetic instructions, CMP, LDA, STA, MOV and TMI
Operand	Word device with 32bit notation (with 'L' at the end) Numeric constants marked by 'L' or numeric constants lower than 0 or higher than 65535

Example of 32bit instruction MOV

Instruction	Operands	Remarks
LD	M1202	The first scan ON, else OFF
MOV	TM10L TM12L	TM10 → TM12 //low 16bits TM11 → TM13 //high 16bits

Example same as previous example, but executed with 16bit instruction MOV

Instruction	Operands	Remarks
LD	M1202	The first scan ON, else OFF
MOV	TM10 TM12	TM10 → TM12
MOV	TM11 TM13	TM11 → TM13

・In case of numerical constant, 'L' at the end is unnecessary. (Example: #65536, #-2)
・For additional info about the numeric type, refer to 2-5-8 Numeric type.

Example of 32bit instruction MOV

Instruction	Operands	Remarks
LD	M1202	The first scan ON, else OFF
MOV	#65536 DM0L	#65536 is equal to 10000h (base 16 number) #0 → DM0 // low 16bits #1 → DM1 // high 16bits

Example same as previous example, but executed with 16bit instruction MOV

Instruction	Operands	Remarks
LD	M1202	The first scan ON, else OFF
MOV	#0 DM0	# 0 → DM 0
MOV	#1 DM1	# 1 → DM 1

Notes

1) Instruction is 32bit instruction, if it has at least one 32bit operand.

Example of 16bit and 32bit operands are mixed

Instruction	Operands	Remarks
LD	M1202	The first scan ON, else OFF
MOV	TM10 DM1L	TM10 is 16bit operand DM1L is 32bit operand Result in DM1L: TM10 → DM1 // low 16bits #0 → DM2 // high 16bits

Example of a **wrong usage** of 32bit and 16bit operands → **Err#004 - Operand?**

Instruction	Operands	Remarks
LD	M1202	The first scan ON, else OFF
MOV	#65536 DM1	#65536 is 32bit operand DM1 is 16bit operand Result: **Err#004 - Operand?**

2) **Do not use contiguous addresses of 32bit operands** to specify source and destination addresses.

Be careful, because error will not occur, but the high 16bits of source operand will be overwritten!

Instruction	Operands	Remarks
LD	M1202	The first scan ON, else OFF
MOV	TM10L TM11L	TM10 → TM11 TM11 → TM12 Note: **TM10L has been changed!**

Instruction	Operands	Remarks
LD	M1202	The first scan ON, else OFF
MOV	#131073 TM66L	131073 is equal to 20001h (base 16 number) #1 → TM66 #2 → TM67 **Note**: **TM66L = #131073**
MOV	TM66L TM67L	TM66 → TM67 // TM67 = #1 TM67 → TM68 // TM68 = #2 **Note**: TM66L has been **changed!** TM67L = #131073, but **TM66L = #65537** (10001h)

Instruction	Operands	TM0L	TM0	TM1	TM2
LD	M1202	*	*	*	*
MOV	#65536 TM0L	65536	0	1	*
LDP	X0	65536	0	1	*
LDA	TM0L	65536	0	1	*
ADD	#2	65536	0	1	*
STA	TM1L	131072	0	2	1

5-2-3-2 Internal register

- There is one 32bit internal register used by most of 32bit instructions.
- The internal 32bit register is used with arithmetic operations (ADD, SUB, MUL, DIV), comparison (CMP), load (LDA) and store (STA)
- Even if 16bit the operand is used, instructions using the internal register are always 32bit instructions with one exception, the STA instruction. The STA instruction with 16bit operand stores only the low 16bits of the internal register.

Instruction	Operands	TM0	TM1	Internal register (32bit)
LD	M1202	*	*	*
MOV	#65535 TM0	#65535 (ffffh)	*	*
LDP	X0	#65535 (ffffh)	*	*
LDA	TM0	#65535 (ffffh)	*	#65535 (0000 ffffh)
ADD	#3	#65535 (ffffh)	*	#65537 (0001 0002h)
STA	TM1	#65535 (ffffh)	#2 (0002h)	#65537 (0001 0002h)

LDA Load

Load the value specified by the operand to the internal register.

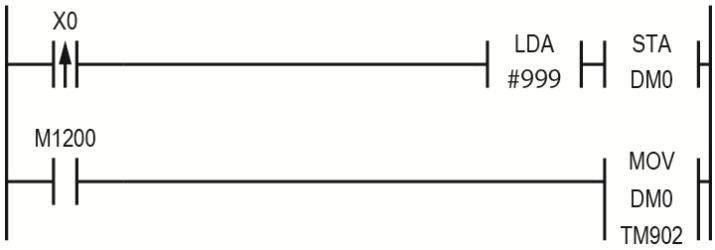
STA Store

Store the value of the internal register to the destination specified by the operand.

32bit instruction with 16bit operand

Program example

Instruction	Operands
LDP	X0
LDA	#999
STA	DM0
LD	M1200
MOV	DM0 TM902



LDA Operand ddd	
Numeric constant	#0～65535
Word device	DM, TM, CM, C

When a rise edge is detected on X0, numeric constant #999 is loaded to the internal register and after the low 16 bits of the internal register are stored in DM0.

STA Operand ddd	
Word device	DM, TM, CM

32bit instruction with 32bit operand

Program example

Instruction	Operands
LDP	X0
LDA	#65536
STA	DM0L
LD	M1200
MOV	DM0L TM902L

When a rise edge is detected on X0, 32bit numeric constant #65536 is loaded to the internal register and after the internal register is stored in DM0L.

Always move DM0L to TM902L to display it on the lower row of the display.

Note: #65536 is 10000h (base 16 number) and so #0 will be in DM0 (low 16bits of DM0L), #1 will be in DM1 (high 16bits of DM0L).

LDA Operand ddd	
Numeric constant	#-2147483648～2147483647 (signed 32bit number)
Word device	DMxxL, TMxxL, CMxxL

STA Operand ddd	
Word device	DMxxL, TMxxL, CMxxL

Note

There has to be an LDA instruction somewhere before the first STA instruction. There can be multiple STA instructions after one LDA instruction.

The LDA instruction with 16bit operand saves data to the 32bit internal register and so high 16 bits of internal register will become 0.

The STA instruction with 16bit operand stores only the low 16bits of the internal register.

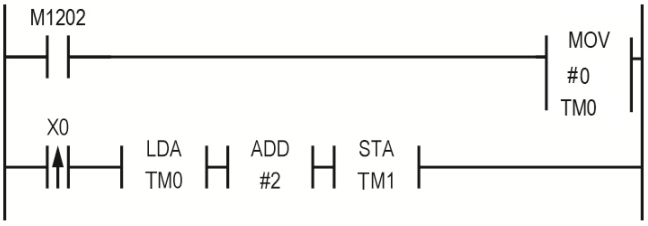
ADD addition

Add a value specified by operand to the value of the internal register, the result is saved to the internal register.

32bit instruction with 16bit operand

Program example

Instruction	Operands
LD	M1202
MOV	#0 TM0
LDP	X0
LDA	TM0
ADD	#2
STA	TM1



Operand ddd	
Numeric constant	#0～65535
Word device	DM、TM, CM

When a rise edge is detected on X0, TM0 and #2 are added and the result is stored in TM1. Even if 16bit the operand is used, ADD instruction is always 32bit instruction, because the internal register is 32bit.
If the result is less than -2147483648 or more than 2147483647 (signed 32bit number minimal and maximal values), an error (Err#029 - Overflow) will occur.

32bit instruction with 32bit operand

Program example

Instruction	Operands
LD	M1202
MOV	#65536 TM0L
LDP	X0
LDA	TM0L
ADD	#-2
STA	TM2L

+

Operand ddd	
Numeric constant	#-2147483648～2147483647 (signed 32bit number)
Word device	DMxxL, TMxxL, CMxxL

When a rise edge is detected on X0, TM0L and #-2 are added, and the result is stored in TM2L (TM2 low 16bits, TM3 high 16bits).

If the result is less than -2147483648 or more than 2147483647 (signed 32bit number minimal and maximal values), an error (Err#029 - Overflow) will occur.

SUB Subtraction

SUB ddd

Subtract a value specified by operand from the value of the internal register, the result is saved to the internal register.

32bit instruction with 16bit operand

Program example

Instruction	Operands
LD	M1202
MOV	#100 TM0
LDP	X0
LDA	TM0
SUB	#2
STA	DM1

Operand ddd	
Numeric constant	#0～65535
Word device	DM, TM, CM

When a rise edge is detected on X0, #2 is subtracted from TM0 and the result is saved In DM1. The SUB instruction is connected between LDA and STA.
Even if 16bit the operand is used, SUB instruction is always 32bit instruction, because the internal register is 32bit.
If the result is less than -2147483648 or more than 2147483647 (signed 32bit number minimal and maximal values), an error (Err#029 - Overflow) will appear.

32bit instruction with 32bit operand

Program example 1

Instruction	Operands
LD	M1202
MOV	#65536 TM0L
LD	M1202
MOV	#2 TM2L
LDP	X0
LDA	TM0L
SUB	TM2L
STA	DM1L

Program example 2

Instruction	Operands
LD	M1202
MOV	#100 TM0
LDP	X0
LDA	TM0L
SUB	#200
STA	DM1L

Operand ddd	
Numeric constant	#-2147483648～2147483647 (signed 32bit number)
Word device	DMxxL, TMxxL, CMxxL

Program example 1

When a rise edge is detected on X0, TM2L is subtracted from TM0L and the result is saved in DM1L.

Program example 2

When X0 is ON, #200 is subtracted from TM0L and the result is saved in DM1L.
Note: Negative results are saved as two's complement.

MUL Multiplication

MUL ddd

Multiply the value of the internal register by the value specified by the operand, the result is saved to the internal register.

32bit instruction with 16bit operand

Program example

Instruction	Operands
LD	M1202
MOV	#3 TM0
LDP	X0
LDA	TM0
MUL	#2
STA	TM1

Operand ddd	
Numeric constant	#0～65535
Word device	DM, TM, CM

When a rise edge is detected on X0, TM0 is multiplied by #2 and the result is stored in TM1. If the result is less than -2147483648 or more than 2147483647 (signed 32bit number minimal and maximal values), an error (Err#029 - Overflow) will appear.

32bit instruction with 32bit operand

Program example

Instruction	Operands
LD	M1202
MOV	#65536 TM0L
LDP	X0
LDA	TM0L
MUL	#-2
STA	DM1L

When a rise edge is detected on X0, TM0L is multiplied by #-2 and the result is stored in TM1L.

If the result is less than -2147483648 or more than 2147483647 (signed 32bit number minimal and maximal values), an error (Err#029 - Overflow) will appear.

Operand ddd	
Numeric constant	#-2147483648～2147483647 (signed 32bit number)
Word device	DMxxL, TMxxL, CMxxL

DIV Division

DIV
ddd

Divide the value of the internal register by the value specified by the operand, the result is saved to the internal register.

32bit instruction with 16bit operand

Program example

Instruction	Operands
LD	M1202
MOV	#8 TM0
LDP	X0
LDA	TM0
DIV	#2
STA	TM1

Operand ddd	
Numeric constant	#0～65535
Word device	DM, TM, CM

When a rise edge is detected on X0, TM0 is divided by #2 and the result is stored in TM1.
If you divide by zero, an error (Err#028 – div zero) will occur.

32bit instruction with 32bit operand

Program example

Instruction	Operands
LD	M1202
MOV	#65536 TM0L
LDP	X0
LDA	TM0L
DIV	#-2
STA	DM1L

When a rise edge is detected on X0, TM0L is divided by #-2 and the result is stored in DM1L.
If you divide by zero, an error (Err#028 – div zero) will occur.

Operand ddd	
Numeric constant	#-2147483648～2147483647 (signed 32bit number)
Word device	DMxxL, TMxxL, CMxxL

CMP Comparison

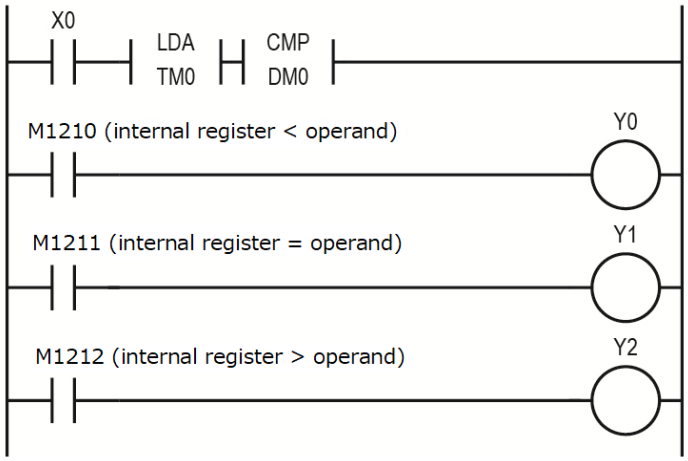
CMP
ddd

Compare the internal register with the value specified by the operand.

32bit instruction with 16bit operand

Program example

Instruction	Operand
LD	X0
LDA	TM0
CMP	DM0
LD	M1210
OUT	Y0
LD	M1211
OUT	Y1
LD	M1212
OUT	Y2



Operand ddd	
Numeric constant	#0~65535
Word device	DM, TM, CM

Compare TM0 and DM0 and reflect the result on special auxiliary relays (M1210 to M1212).
When TM0 < DM0, M1210 and output relay Y0 is ON.
When TM0 = DM0, M1211 and output relay Y1 is ON.
When TM0 > DM0, M1212 and output relay Y2 is ON.
The comparison is always 32bit, because the internal register is 32bit.

32bit instruction with 32bit operand

Program example

Instruction	Operand
LD	X0
LDA	TM0L
CMP	DM0L
LD	M1210
OUT	Y0
LD	M1211
OUT	Y1
LD	M1212
OUT	Y2

Compare the 32bit numbers, TM0L with the DM0L and reflect the result on special auxiliary relays (M1210 to M1212).

Breaking change! (from v2.05)
If the result of a logic operation (in program example `LD X0`) is OFF the CMP instruction will reset M1210-M1212 to OFF.
NOTE:
The firmware v2.04 (and older) is not changing M1210-M1212 when the result of logic operation is OFF.

Operand ddd	
Numeric constant	#-2147483648~2147483647 (signed 32bit number)
Word device	DMxxL, TMxxL, CMxxL

INC Increment

INC
ddd

Increment by 1 a value of the memory (DM, TM, CM) specified by the operand.

DEC Decrement

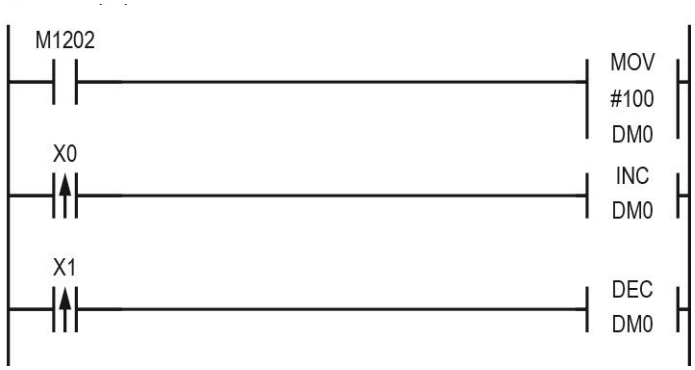
DEC
ddd

Decrement by 1 a value of the memory (DM, TM, CM) specified by the operand.

16bit instruction

Program example

Instruction	Operands
LD	M1202
MOV	#100 DM0
LDP	X0
INC	DM0
LDP	X1
DEC	DM0



Operand ddd	
Word device	DM, TM, CM

When a rise edge is detected on X0, the INC instruction adds 1 to value in DM0.
When a rise edge is detected on X1, the DEC instruction subtracts 1 from value in DM0.
If the result exceeds 65535 or becomes negative, an error (Err#029 - Overflow) will occur.

32bit instruction

Program example

Instruction	Operands
LD	M1202
MOV	#65535 DM0L
LDP	X0
IN	DM0L
LDP	X1
DEC	DM0L

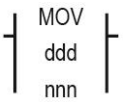
Operand ddd	
Word device	DMxxL, TMxxL, CMxxL

When a rise edge is detected on X0, the INC instruction adds 1 to value in DM0L.

When a rise edge is detected on X1, the DEC instruction subtracts 1 from value in DM0L.

If the result is less than -2147483648 or more than 2147483647 (signed 32bit number minimal and maximal values), an error (Err#029 - Overflow) will occur.

MOV Move



Transfer the value specified by the first operand to the destination specified by the second operand.

16bit instruction

Program example

Instruction	Operands
LD	M1202
MOV	#100 DM0
LD	M1200
MOV	DM0 TM902
LD	M1200
MOV	#0 TM903

Operand ddd	
Numeric constant	#0～65535
Word device	DM, TM, CM, C

Operand nnn	
Word device	DM, TM, CM

Transfer #100 to DM0 during the first scan (M1202 - the first scan ON, else OFF).

Always transfer DM0 to TM902 to display it on the lower row of the display.

Always transfer #0 to TM903 (high 16 bits of TM902L) to remove any previous values, because display uses 32bit access to TM902L.

32bit instruction

Program example

Instruction	Operands
LD	M1202
MOV	#65536 DM0L
LD	M1200
MOV	DM0L TM902L

Transfer #65536 to DM0L during the first scan (M1202 - the first scan ON, else OFF).
Always transfer DM0L to TM902L to display it on the lower row of the display.

When displaying numerical values on the display a 32bit instruction is recommended.

Operand ddd	
Numeric constant	#-2147483648～2147483647 (signed 32bit number)
Word device	DMxxL, TMxxL, CMxxL

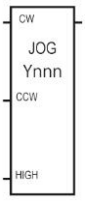
オペランド nnn	
Word device	DMxxL, TMxxL, CMxxL

Note

Second operand "nnn" specifies destination (DM, TM, CM have to be used), a numeric constant cannot be used.
The MOV instruction does **not use** the internal register for the transfer.

5-2-4 Positioning instructions

JOG Jog



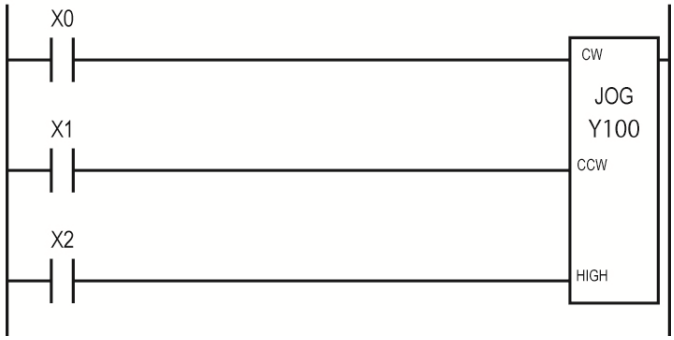
Send positioning pulses to the output relay specified by the operand.

Program example

Instruction	Operand
LD	X0 // CW-clockwise
LD	X1 // CCW // counter clockwise

LD	X2
	// OFF : low speed
	// ON : high speed

JOG	Y100



Output	Operand Ynnn
Motor1	Y100
Motor2	Y102

When X0 is ON, send CW pulses to outputs Y100, Y101.
When X1 is ON, send CCW pulses to outputs Y100, Y101.
When X2 is ON, a high speed is selected, else low speed will be used.
It is not possible to change the direction of rotation, low/high speed during motor operation.

Notes
If CW and CCW are ON at the same time, positioning pulses will slow down to stop.
Multiple JOG instructions with the same operand cannot be placed in the program.
Refer to 3-2 Stepping motor for details about operating conditions etc., 3-2-4 Motor positioning: JOG driving.

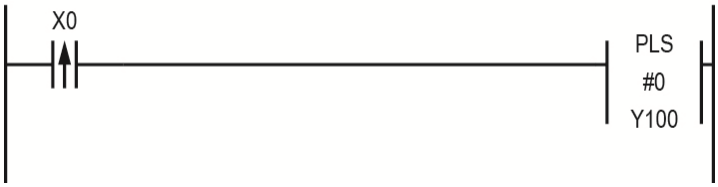
PLS Pulse

PLS
#ddd
Ynnn

Send positioning pulses to the output specified by the operand2 according to the selected preset position specified by the operand1.

Program example

Instruction	Operands
LDP	X0
PLS	#0 Y100



Operand	#ddd
Positioning data No	#0～9

Output Motor	Operand Ynnn
Motor 1	Y100
Motor 2	Y102

When a rise edge is detected on X0, trapezoidal control is executed on motor 1 with settings defined by the positioning data No. #0.

Notes

Refer to 3 - 2 Stepping motor for details about operating conditions etc., 3 - 2 - 5 Trapezoidal control: PLS driving.

6 Data memory setting application (DM_Set)

The data memory setting application reads and writes data memory DM and control memory CM.

6-1 Installing the USB driver

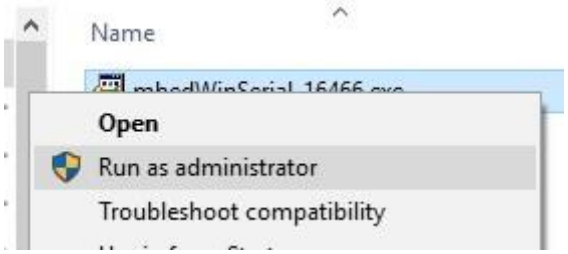
Before installation, it is necessary to install USB driver for serial communication on the PC side.

6-1-1 Download USB driver

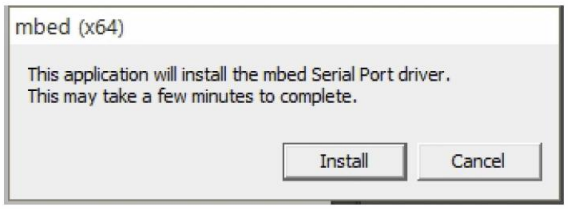
USB driver for serial communication can be downloaded from the following URL.
https://smartplc.org/about/downloadsEN.html

6-1-2 Driver installation procedure

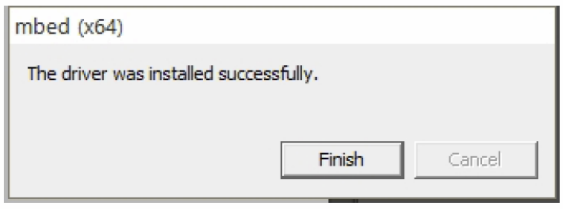
① Turn off the smartPLC's power switch and connect it to the personal computer with a USB cable.
② Open the downloaded file by right-clicking it and execute it as an administrator.



③ Click Install.



④ Click Finish to complete the installation.



⑤ Open the Device Manager and check that mbed Serial Port (comXX) is added to the port.
* The XX number of comXX changes according to the environment of the personal computer.

6-2 Installation

Operating environment: Windows7 64bit Microsoft.NET Framework version 4.0 or later
or Windows 10

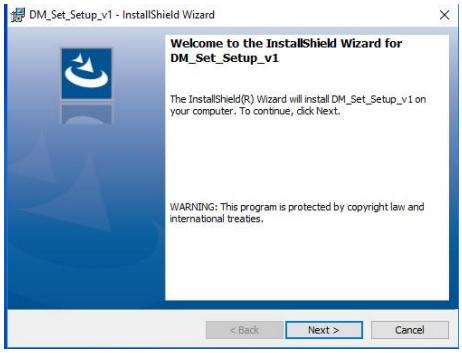
This application is tested in the following environment.
Windows 7 64bit with Microsoft .NET Framework 4.5, Windows 10
Microsoft Excel 2013 (corresponding to 97-2003 book)

6-2-1 Download

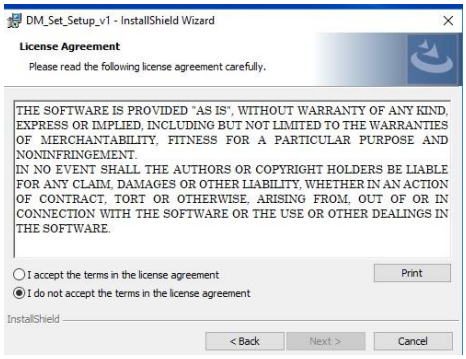
You can download the DM_Set application from the URL below. Move the downloaded file into a suitable folder.
https://smartplc.org/about/downloadsEN.html

Double click DM_Set_Setup.exe to install the application.

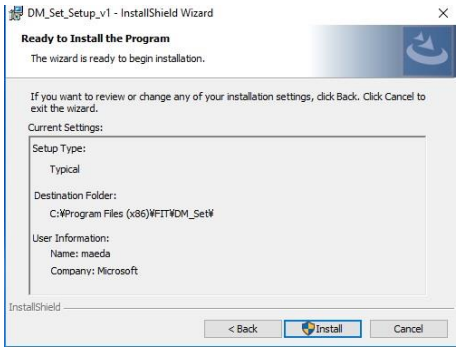
1. Click Next



2. Click "I accept the..." and Next



3. Click Install



6-2-2 Setup

Files are installed as follows.

File name	Destination	Remarks
DM_set.xls (or DM_set.xlsx)	Document folder or Application folder	Excel file for setting
DM_set.exe	Application folder (It is chosen during install.)	Executable file
DM_set.lnk	Desktop folder	Shortcut

6-3 How to use

6-3-1 DM set mode

Before you execute the DM_Set application, set the smartPLC to DM_Set mode. User program is not running when the smartPLC is in DM_Set mode.

DM_Set mode ON:
Press and hold button S1 on the front panel of the smartPLC and at same time restart ([RUN →] STOP→RUN) the smartPLC. It will boot to DM_Set mode. When “DM_Set” is Displayed, release button S1.

DM_Set mode OFF:
restart (RUN→STOP→RUN) the smartPLC

6-3-2 Connection with a PC

1. Connect to the computer and the smartPLC with USB cable.
2. When you start up DM_Set.exe by clicking the desktop icon, COM port is recognized automatically.

*If the smartPLC is not connected to the PC an error message will be displayed. When you plug in the smartPLC, COM port will be recognized automatically.)

* Do not change COM port, unless it cannot be recognized automatically.
* To avoid recognition errors, don’t connect another smartPLC or mbed device.

6-3-3 Setting up Excel

1. Open DM_set.xls with the Open Excel button and select Sheet1.
2. Edit Sheet1 and write the DM address, CM address and value to be set.
3. Save Sheet1.

	A	B	C	D	E	F	G
1	DM Addres	DM Value			CM Addres	CM Value	
2	1000	0	#0: operating mode (0:relative value, 1:al		0	10	test data1
3	1001	50000	#0: stop position (low 16bits)		1	20	test data2
4	1002	0	#0: stop position (high 16bits)		2	30	test data3
5	1003	200	#0: starttup speed (Hz)				
6	1004	1000	#0: acceleration/deceleration time (ms)				

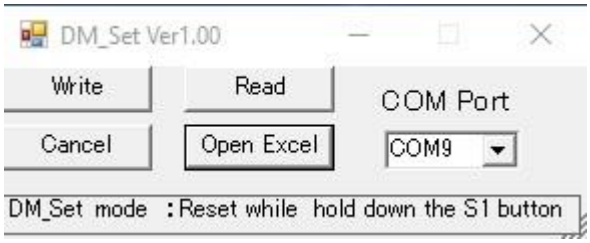
DM_Set.xls setting example

＜Caution＞

- Only "Sheet1" is used for DM / CM setting area.
- Set the DM and CM on the first row. Columns can be moved.
- Fill in the addresses and data in order from the top.
- In other columns there can be user comments. Cell filling can be used.
- Use halfwidth characters for the entering of values, because the values written by fullwidth characters (used in Chinese, Japanese and Korean languages) will not work.
- The ranges of addresses are following.

  DM address : 0～1999
  CM address : 0～9
  Value : 0～65535(unsigned16bit)
- If you forget to save the worksheet, the changes will not be reflected and the previous values will be used. (It may cause troubles.)
- You can also use Excel (2007 or later) ".xlsx" format.

6-3-4 Buttons



Button	Behavior
Write	Write the settings of DM_set.xls to smartPLC DM/CM memory. Excel will open automatically, write and close. (About 5 seconds) Do not operate the PC until Excel is closed.
Read	Add sheet to Excel and write DM and CM values from the smartPLC to the sheet. Excel stays open after reading.
Cancel	Quit the application. (Any ongoing communication will be canceled.)
Open Excel	Open DM_set.xls.

6-3-5 Return to normal mode

Restart (RUN→STOP→RUN) the smartPLC, it will enter to the normal mode and run the user program.

7 Troubleshooting

This section describes what to do in case of troubles. Read it, when trouble occurs.

Trouble Description	Possible Cause	Solution
The smartPLC disk drive is not recognized.	The USB cable is not properly connected.	Reconnect the USB cable. Check if the USB cable is connected to the computer and if the other end of the cable is connected to smartPLC.
program.txt is broken and can not be read.	The file was corrupted due to some trouble while writing the program.	Delete the corrupted file and copy a new program.txt to the smartPLC drive.
An input does not turn on. An output does not turn on.	There is no DC 24V power supply output command in the program.	Add "LD M1200", "OUT Y202" to the beginning of the program.
	Wiring is not correct.	Check the connection of E-CON.
	Voltage is low.	Check the voltage between 1 and 3 pins of E - CON.
The program is not working properly.	The RUN - STOP switch is set to STOP.	Set the switch to RUN.
	24 VDC power supply output is not ON.	Add "LD M1200", "OUT Y202" to the beginning of the program.
	DM is not set (temperature control, load cell, motor)	Set DM with DM_Set application or with MOV instruction.
	Wiring is not correct. (Temperature control, load cell, motor, encoder)	Check the connection of the terminal.
	The power is not turned on.	Make sure the power switch is on.
	An error has occurred.	Check the error code and its description and take appropriate action.
	A fail at the start of the microcontroller.	Do reset.
	There is no firmware.	Download the firmware and copy it to the smartPLC disk drive.

8 Error code list

Program errors which appear before SCAN cycle starts.
The error code and the line number alternately blink at the top of the display and the error number is displayed at the bottom.
The instruction on the line number can be correct, as the error can depend on the instructions or missing instructions before the line.

Error code Line number	Error number	Error Description	Description
1st row upper row	2nd row lower row		
END???	001	There is no END instruction	Enter the END command at the end of the program
Char???	002	Invalid or unexpected character	Check the program line for wrong characters and remove them. Check character encoding of program.txt. (Refer to 5-1. Program)
Command?	003	Invalid instruction (command) name.	Check and fix the instruction name.
Operand?	004	Operand error	Check validity of operand and fix it. Remove spaces, tabs between operand type and address/number. (Example ○: M1, ×: M 1) Check if numeric constant fits to 32bit value (# -2147483648 to 2147483647) Remove leading zeroes of the number. Check number of operands. Check if there is one instruction on one line. MOV: Check if the first operand is not 32bit and the second 16bit
LD?ANL?O	005	LD, ANL, ORL mismatch	Check if there is enough LD (LD, LDB, LDP, LDF) instructions before the ANL or ORL instruction. Minimum two LD instructions have to be before one ANL/ORL.
Too long	006	The program line has more than the maximum number of characters	Make the number of characters of the line 500 characters or less. **Note** one fullwidth character counts as two characters. Check character encoding of program.txt. (Refer to 5-1. Program)
MPS?MPP!	007	MPS and MPP mismatch	Check if there is MPS before MRD or MPP. Check if number of MPS equals to number of MPP instructions. (Refer to 5-2-2 MPS instruction)
No LD!	008	LD is missing or there is insufficient number of LD instructions.	Insert the LD instruction line before the error line. **Note** KEEP needs two and JOG require three LD instructions.
No LDA!	009	There is no LDA instruction before STA, CMP, ADD, SUB, MUL or DIV instruction.	Insert the LDA instruction line before the error line.

Error code Line number	Error number	Error Description	Description
1st row upper row	2nd row lower row		
No Prog.	010	Program reading error	Check the file name of program.txt. Check if there is program.txt in the inserted SD card.
Prog.OVR	011	The instructions do not fit to the memory.	Reduce number of instructions.
MPS > 63	013	The number of mutually nested MPS-MPP pairs is more than 63.	Reduce number of mutually nested MPS-MPP pairs.
Many OPR	014	Many operands	Delete extra operands.
2CNT1ADR	015	Multiple CNT instructions with the same counter number.	Change the counter number to a different number.
2DIF1ADR	016	Multiple DIFU, DIFD instructions use the same memory bit.	Change the operand to a different bit memory address.
2OUT->1Y	017	There are multiple OUT, OUB instructions using the same output or the output is already used by SET, RES, KEEP, PLS, JOG instruction. There are multiple JOG instructions using the same output.	Change the operand to different output.
2TMx1ADR	018	There are multiple timers with the same timer number.	Change the timer number to a different number.
Unhandl.	020	Unhandled error	There is a possible problem with the firmware. Contact us with firmware version number and program.txt.

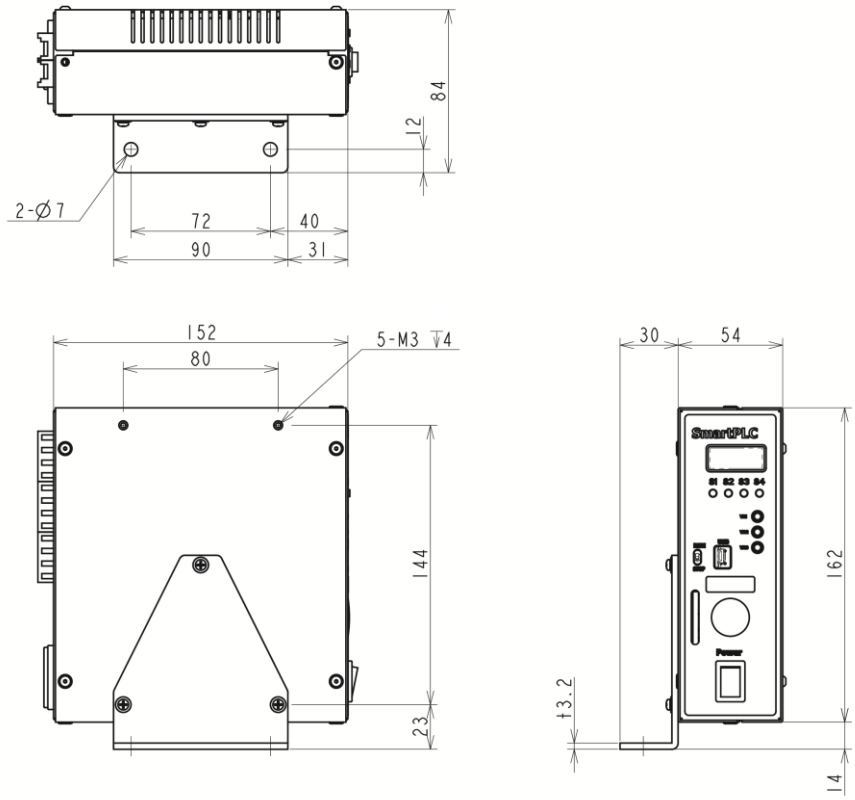
Scan error occurred during run of user program.

Error code Line number	Error number	Error Description	Description
First row (upper row)	2nd line (bottom line)		
END?!?	021	END is missing in instruction memory	There is a possible problem with the firmware. Contact us with firmware version number and program.txt.
INScode?	022	Invalid instruction code in instruction memory	There is a possible problem with the firmware. Contact us with firmware version number and program.txt.
OPERcode	023	Invalid operand code in instruction memory	There is a possible problem with the firmware. Contact us with firmware version number and program.txt.
JOG DM !	024	DM not set for JOG operation (start speed, acceleration / deceleration time or operation speed is 0)	In the DM_Set application or with MOV instruction, set the values for JOG control in data memory(DM) other than 0.
LDCEL DM	025	DM not set for loadcell control (rated capacity or rated output is 0)	In the DM_Set application or with MOV instruction, set the values for loadcell control in data memory(DM) other than 0.
PLS DM !	026	Invalid positioning data for PLS in DM (start speed, acceleration / deceleration time or operation speed is 0)	In the DM_Set application or with MOV instruction, set the values for positioning data in data memory(DM) other than 0.
div zero	028	Divide by 0 (zero)	Division by 0 is not possible. Set a not zero divisor.
Overflow	029	Result overflow/underflow	32bit instructions (ADD, SUB, MUL) can lead to underflow/overflow of internal 32bit register. INC, DEC instruction can lead to overflow/underflow of 16bit or 32bit registers. (Refer to 5-2-3. Arithmetic instruction).

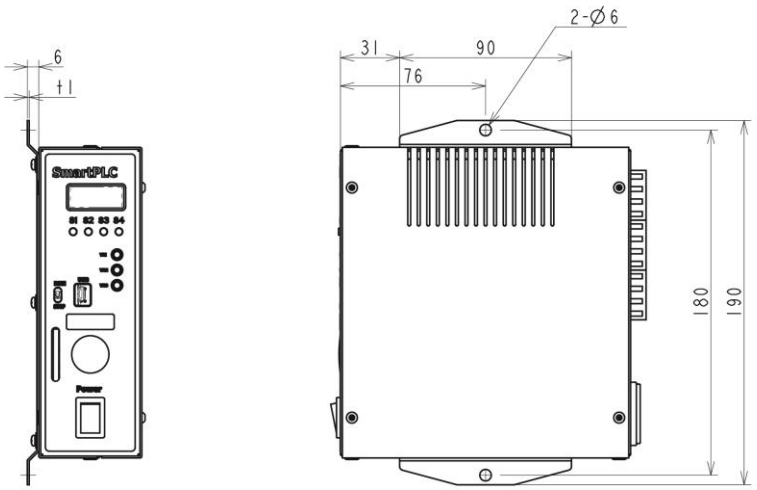
9 Mounting

Dimensions and dedicated bracket mounting dimensions

9-1-1-1 Top mounting bracket type



9-1-1-2 Side mounting bracket type



10 About warranty

1. Warranty Period

The warranty period of our products shall be one year after purchase or after delivery to the designated location.

2. Scope of Warranty

In the event of a failure of our products under the responsibility of the Company during the above warranty period, we will provide alternative products or repair defective products at no charge at the purchase site of the product. However, if the cause of the failure is any of following, it is excluded from the scope of this warranty.

Following cases are exclusive from the warranty:

a) Failure and damages caused by improper use or by conditions other than those stated in the catalog or in the user manual, for example outdoor applications, chemical contamination, flooding, physical damage, incorrect electric connection, electric shock, wrong handling, etc.

b) Failure and damages caused by the devices or the control software produced by other than our company.

c) The product was modified or repaired by other than our company.

d) Usage different to the original purpose of our product.

e) Usage of the device for protection of human beings, or in applications where a failure of the device can lead to injuries, fatalities of humans or animals or where it can lead to high damages on equipment, environment, etc.
Usage of the device in a hazardous location or in a location that has a potentially explosive atmosphere.

f) Failure to foresee at the science and technology level at the time of shipment of the product.

g) Failure and damages caused by natural disasters, disasters, terrorist attacks, radiation, fire, theft or loss of the product, and any other causes which are not our responsibility.

The guarantee here means the warranty for our product alone, and any damage induced by the breakdown of our product shall be excluded from the scope of the warranty.

3. Limitation of Liability

The Company shall not be liable in any case for special, indirect, or passive damages caused by our products. We are not responsible for programs programmed other than by our company, or the results caused by it.

4. Applications

Our products are designed and manufactured as General-purpose products for general industries. If our products are used in combination with other products, check standards, norms and regulations that should by applied by the customer. In addition, confirm the suitability of our products for your system, machinery and equipment. We are not responsible for the suitability of our products if these are not followed.
Our products are not intended for use in the following applications:

a) Use in outdoor applications, in applications where are potential chemical contamination or electrical disturbances, or in conditions and environments not listed in the catalog or in the user manual, etc.

b) Facilities for nuclear power control, incineration facilities, railway / aviation / vehicle facilities, medical equipment, entertainment machines, safety equipment, facilities complying with administrative agencies and individual industry regulations.

c) Systems, machinery and equipment that can be dangerous to health, human life and property

d) Gas, water supply, electricity supply system and 24 hours facilities requiring high reliability such as continuous operation system

e) Other applications requiring a high degree of safety pursuant to a) to d) above.

In case you use our product, in place which could be potentially dangerous to health, human life or to property, make sure you can guarantee doing things such as systems informing about overall hazards, a redundant safety design.

5. Change of Specifications

Specifications and accessories of products described in catalog, user manual, etc. may be changed as necessary due to improvement or other reasons without any notice.

6. Scope of application

The above content is based on transactions and use in Japan.
For transactions and use outside Japan, consult our company sales office.

Visit our webpage for the product inquiries

https://smartplc.org