HITACHI PROGRAMMABLE CONTROLLER

IEC 61131-3 Compliant PLC

MICRO-EHV+

APPLICATION MANUAL

(SERVICE MANUAL)

NJI-611(X)

○ Warranty period and coverage

The warranty period is the shorter period either 18 months from the data of manufacture or 12 months from the date of installation.
However within the warranty period, the warranty will be void if the fault is due to;

(1) Incorrect use as directed in this manual and the application manual.

(2) Malfunction or failure of external other devices than this unit.

(3) Attempted repair by unauthorized personnel.

(4) Natural disasters.

The warranty is for the PLC only, any damage caused to third party equipment by malfunction of the PLC is not covered by the warranty.

○ Repair

Any examination or repair after the warranty period is not covered. And within the warranty period ant repair and examination which results in information showing the fault was caused by ant of the items mentioned above, the repair and examination cost are not covered. If you have ant questions regarding the warranty please contact with your supplier or the local Hitachi Distributor. (Depending on failure part, examination might be impossible.)

○ Ordering parts or asking questions

When contacting us for repair, ordering parts or inquiring about other items, please have the following details ready before contacting the place of purchase.

(1) Model

(2) Manufacturing number (MFG.No.)

(3) Details of the malfunction

○ Reader of this manual

This manual is described for the following person.

・Person considering the introduction of PLC
・PLC system engineer
・Person handling PLC
・Manager after installing PLC

Warning

(1) Reproduction of the contents of this manual, in whole or in part, without written permission of Hitachi-IES, is prohibited.
(2) The content of this document may be changed without notice.
(3) While efforts have been made to be accurate, if any wrong or missing information is found, please contact us.

MS-DOS®, Windows®, and Windows NT® are registered trademarks of America and other registered countries of Microsoft Corp. of the United States.

Safety Precautions

Read this manual and related documents thoroughly before installing, operating, performing preventive maintenance or performing inspection, and be sure to use the unit correctly. Use this product after acquiring adequate knowledge of the unit, all safety information, and all cautionary information. Also, make sure this manual enters the possession of the chief person in charge of safety maintenance.

Safety caution items are classifies as “Danger” and “Caution” in this document.

DANGER : Identifies information about practice or circumstances, which may lead to personal injury or death, property damage, or economic loss.

CAUTION : Identifies information about practice or circumstances, which may lead to personal injury, property damage, or economic loss.

However, depending on the circumstances, items marked with CAUTION may result in major accidents.

The both marks show important information. Be sure to follow the instructions.

Icons for prohibited items and required items are shown below:

: Identifies prohibition. For example, when open flames are prohibited, is indicated.

: Identifies requirement. For example, when grounding must be performed, is indicated.

1. Installation

⚠ CAUTION

- Use this product in an environment as described in the catalog or this document.
  If this product is used in an environment subject to high temperature, high humidity, excessive dust, corrosive gases, vibration or shock, it may result in electric shock, fire or malfunction.
- Be sure to install the PLC according to this manual. Failure to do so could result in damage by falling off, failure or malfunction.
- Do not allow foreign objects such as wire chips to enter the unit.
  They may become the cause of fire, malfunction or failure.

2. Wiring

REQUIRED

- The PLC must be grounded (FE terminal).
  Failure to do so could result in injury to personnel or causing it to malfunction.

⚠ CAUTION

- Always use the power supply voltage listed in specifications. Using other voltage may damage the equipment or present a risk of fire.
- The wiring operation should be performed by a qualified personnel.
  Failure to do so could result in fire, damage or electric shock.

3. Precautions when using the unit

DANGER

- Do not touch the terminals while the power is on. There is a risk of electric shock.
- Appropriate emergency stop circuit, interlock circuitry and similar safety measures should be added to the PLC system to ensure safety in the event of incorrect, missing or abnormal signals caused by broken signal lines, momentary power interruptions or other causes. Do not share the power supply of relay output and interlock circuitry because relay output might not work properly due to switching noise from interlock circuitry.

⚠ CAUTION

- When performing program change, forced output, RUN, STOP, etc., while the unit is running, be sure to check system safety carefully. Failure to do so could lead to damage to equipment.
- Supply power according to the power–up order.
  Failure to do so could lead to damage to equipment or malfunction.

4. Preventive maintenance

DANGER

- Do not connect the ⊕ / ⊖ of the battery in reverse polarity. Do not recharge, disassemble, heat, place in fire, or short circuit the battery. There is a risk of explosion or fire.

PROHIBITED

- Do not attempt to disassemble, repair or modify any part of the PLC.
  Electric shock, malfunction or failure may result.

⚠ CAUTION

- Turn off power to the PLC before mounting or dismounting the module
  Electric shock, malfunction or failure may result.

Revision History

No.	Description of revision	Date of revision	Manual number
1	The first edition	Feb. 2014	NJI-611(X)

MEMO

Table of Contents

MEMO

Chapter 1 Introduction

Thank you very much for choosing Hitachi Programmable Controller (hereinafter referred to as PLC), MICRO-EHV+ series PLC.

1.1 Unpacking

(1) Installation of a battery

The battery for MICRO-EHV+ series PLC is an optional extra. If you need real time clock function or retentive data memory, you need to purchase it. Refer to “Chapter 5 Maintenance” for further information.

(2) Initializing of user program

Since initial status of memory devices in the CPU is undefined, OK LED may blink at the first power up by reason of memory error. In order to initialize memory area, execute “Reset origin” in the first use.

1.2 Instruction Manuals

MICRO-EH series expansion units are available with MICRO-EHV+ series PLC as listed in page 2-3.
Besides this application manual, application manuals are available shown in Table 1.1.

Table 1.1 Related manuals to MICRO-EHV+ series PLC (1/2)

Product name	Model	Type	Application manual number
64 Points Expansion unit	EH-A64EDR	AC-powered, DC input×40, Relay output×24	NJI-522*(X)
	EH-D64EDR	DC-powered, DC input×40, Relay output×24	
	EH-D64EDT	DC-powered, DC input×40, Transistor output×24	
	EH-D64EDTPS	DC-powered, DC input×40, Transistor output×24 (short circuit protection)	
28 Points Expansion unit	EH-A28EDR	AC-powered, DC input×16, Relay output×12	NJI-419*(X)
	EH-D28EDR	DC-powered, DC input×16, Relay output×12	
	EH-D28EDT	DC-powered, DC input×16, Transistor output×12	
	EH-D28EDTP	DC-powered, DC input×16, Transistor output×12	
	EH-D28EDTPS	DC-powered, DC input×16, Transistor output×12 (short circuit protection)	
16 Points Expansion unit	EH-D16ED	DC-powered, DC input×16	NJI-467*(X)
	EH-D16ER	DC-powered, Relay output×16	
	EH-D16ET	DC-powered, Transistor output×16	
	EH-D16ETPS	DC-powered, Transistor output×16 (short circuit protection)	
14 Points Expansion unit	EH-A14EDR	AC-powered, DC input×8, Relay output×6	NJI-350*(X) (MICRO-EH application manual)
	EH-D14EDR	DC-powered, DC input×8, Relay output×6	
	EH-D14EDT	DC-powered, DC input×8, Transistor output×6	
	EH-D14EDTP	DC-powered, DC input×8, Transistor output×6	
	EH-D14EDTPS	DC-powered, DC input×8, Transistor output×6 (short circuit protection)	
8 Points Expansion unit	EH-D8ED	DC-powered, DC input×8	NJI-467*(X)
	EH-D8ER	DC-powered, Relay output×8	
	EH-D8ET	DC-powered, Transistor output×8	
	EH-D8ETPS	DC-powered, Transistor output×8 (short circuit protection)	
	EH-D8EDR	DC-powered, DC input×4, Relay output×4	
	EH-D8EDT	DC-powered, DC input×4, Transistor output×4	
	EH-D8EDTPS	DC-powered, DC input×4, Transistor output×4 (short circuit protection)	

Table 1.2 Related manuals to MICRO-EHV+ series PLC (2/2)

Product name	Model	Type	Application manual number
Analog Expansion unit	EH-A6EAN	AC-powered, Analog input×4, Analog output×2	NJI-424*(X)
	EH-D6EAN	DC-powered, Analog input×4, Analog output×2	
RTD Expansion unit	EH-A6ERTD	AC-powered, RTD input×4, Analog output×2	NJI-453*(X)
	EH-D6ERTD	DC-powered, RTD input×4, Analog output×2	
	EH-A4ERTD	AC-powered, RTD input×4	
	EH-D4ERTD	DC-powered, RTD input×4	
Thermocouple Expansion unit	EH-D6ETC	DC-powered, Thermocouple input×4, Analog output×2	NJI-515*(X)
	EH-D4ETC	DC-powered, Thermocouple input×4	

* The last alphabet of the manual No. stands for version starting from blank, A, B, C...

1.3 System overview

MICRO-EHV+ series PLC is all-in-one type programmable controller shown in Figure 1.1.

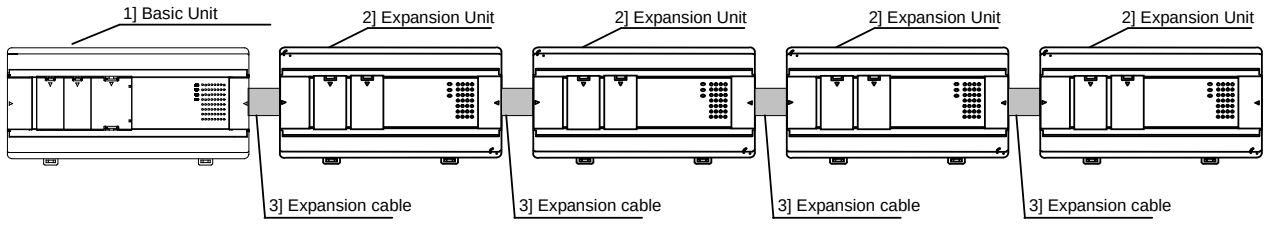


Figure 1.1 MICRO-EHV+ series PLC System configuration diagram

No.	Device name	Description of function
1]	Basic Unit	Reads input signals, executes user application program and writes output signals.
2]	Expansion Unit	Unit of external I / O Expansion
3]	Expansion cable	0.1m, 0.5m and 1m cable are available. The maximum cable length 2m in total.

EHV-CODESYS

EHV-CODESYS is IEC61131-3 compliant programming software for MICRO-EHV+ series PLC. “CODESYS” is a Trademark of the company 3S-Smart Software Solutions GmbH. “EHV-CODESYS” is same tool as “CODESYS” however, Hitachi specific device description files and libraries are preinstalled.

Chapter 2 Specifications

2.1 General specifications

Table 2.1 General specifications

Items	Specifications	
Power supply	AC-powered	DC-powered
Rated input voltage	100/110/120 V AC (50/60Hz) 200/220/240 V AC (50/60Hz)	24 V DC
Input voltage range	85 to 264 V AC wide range	19.2 to 30 V DC
Permissible instantaneous power failure	85 to 100 V AC: less than 10ms 100 to 264 V AC: less than 20ms	19.2 to 30 V DC: less than 10ms
Operational temperature	0 to 55 °C	
Storage temperature	−10 to 75 °C	
Operational humidity	5 to 95 % RH (no condensation)	
Storage humidity	5 to 95 % RH (no condensation)	
Pollution degree	Pollution degree 2 (IEC 61131-2)	
Altitude / Atmospheric pressure	UP to 2000 m (min. 70kPa during transportation)	
Vibration resistance	Conforms to IEC 60068-2-6	
Impact resistance	Peak acceleration: 147m/s^2, Duration: 11ms, Direction: 3, repeat each impact 3times	
Noise resistance	○ Noise voltage 1,500 Vpp, Noise pulse width 100 ns, 1μs (Noise input by a noise simulator across input terminals of a power module according to measuring method of Hitachi-IES.) ○ Static noise 3,000 V at electrode part	
Certifications	CE	
Insulation resistance	20 MΩ minimum between AC terminal and frame ground (FE) terminal (based on 500 V DC megger)	
Dielectric withstand voltage	1,500V AC for 1 minute between AC input terminal and frame ground (FE) terminal	
Ground	Class D grounding (grounding with the power supply module)	
Usage environment	No corrosive gases, no excessive dust	
Structure	Open wall-mount type	
Cooling	Natural air cooling	

2.2 Performance specifications

Table 2.2 Performance specifications

Item		Specification
		MV-*40** / MV-*64**
User program memory		1MB
Source file memory		1MB
Data memory (non retain)		640KB
Data memory (retain)		256KB (incl. 64KB persistent variables)
No. of expansion unit		4
No. of I/O (using 64 points unit)		320 (input: 200, output: 120)
Special I/O	Counter input	Single phase: 100kHz×5ch. (32bit) 2-phase: 60kHz×2ch. (32bit)
	Interrupt input	5 ch.
	Pulse / PWM output	65kHz×3ch.
Programming language		IEC61131-3 compliant 5 languages LD: Ladder Logic Diagram FBD: Function Block Diagram (incl. CFC : Continuous Function Chart) SFC: Sequential Function Chart IL: Instruction List ST: Structured Text
I/O updating cycle		Refresh processing
Communication	Protocol	CODESYS V3 protocol
	USB	USB 2.0 Full speed (Gateway *)
	Ethernet	10BASE-T / 100BASE-TX (Gateway *, Modbus-TCP slave)
	Serial	RS-232C (Modbus-RTU master/slave, General purpose)
	Option serial	RS-422/485 (Modbus-RTU master/slave, General purpose)
Switch, Indications	Indications	POW LED, RUN LED, OK LED, STATUS LED
	RUN switch	STOP / RUN (Remote STOP/RUN enabled when the switch position is RUN.)
Calendar / Clock		Support (Built-in RTC)
USB memory port		Support (Program transfer, Data logging, Web visualization)
Battery		MV-BAT (for retentive data and Real time clock)
Maintenance function		Diagnosis (micro processor error, watch dog timer error, memory error, battery error, etc.)

* Gateway: Communication with EHV-CODESYS

Table 2.3 Processing speed

Data type (number of bit)	Command	Processing time [µs / IL]
BOOL (1)	OR	0.54
BOOL (1)	AND	0.54
INT (16)	ADD	0.54
INT (16)	MUL	0.54
DINT (32)	ADD	0.68
DINT (32)	MUL	0.68
REAL (32)	ADD	0.71
REAL (32)	MUL	0.71
LREAL (64)	ADD	6.38
LREAL (64)	MUL	6.33

2.3 Product lineup

Table 2.4 List of system equipment

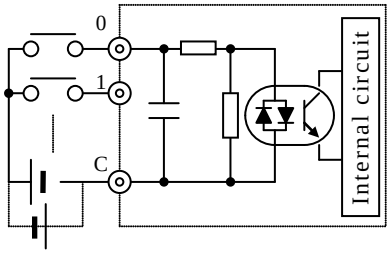
Product	Type	Specification	I/O type	Remarks
64 Points Basic unit	MV-A64DR	100/200 V AC, DC input×40, Relay output×24		
	MV-D64DR	24 V DC, DC input×40, Relay output×24		
	MV-D64DT	24 V DC, DC input×40, Transistor output×24		Sink
	MV-D64DTPS	24 V DC, DC input×40, Transistor output×24 (short circuit protection)		Source
40 Points Basic unit	MV-A40DR	100/200 V AC, DC input×24, Relay output×16		
	MV-D40DR	24 V DC, DC input×24, Relay output×16		
	MV-D40DT	24 V DC, DC input×24, Transistor output×16		Sink
	MV-D40DTPS	24 V DC, DC input×24, Transistor output×16 (short circuit protection)		Source
64 Points Expansion unit	EH-A64EDR	100/200 V AC, DC input×40, Relay output×24	64 DIO	
	EH-D64EDR	24 V DC, DC input×40, Relay output×24	64 DIO	
	EH-D64EDT	24 V DC, DC input×40, Transistor output×24	64 DIO	Sink
	EH-D64EDTPS	24 V DC, DC input×40, Transistor output×24 (short circuit protection)	64 DIO	Source
28 Points Expansion unit	EH-A28EDR	100/200 V AC, DC input×16, Relay output×12	8-28 DIO	
	EH-D28EDR	24 V DC, DC input×16, Relay output×12	8-28 DIO	
	EH-D28EDT	24 V DC, DC input×16, Transistor output×12	8-28 DIO	Sink
	EH-D28EDTP	24 V DC, DC input×16, Transistor output×12	8-28 DIO	Source
	EH-D28EDTPS	24 V DC, DC input×16, Transistor output×12 (short circuit protection)	8-28 DIO	Source
16 Points Expansion unit	EH-D16ED	24 V DC, DC input×16	8-28 DIO	
	EH-D16ER	24 V DC, Relay output×16	8-28 DIO	
	EH-D16ET	24 V DC, Transistor output×16	8-28 DIO	Sink
	EH-D16ETPS	24 V DC, Transistor output×16 (short circuit protection)	8-28 DIO	Source
14 Points Expansion unit	EH-A14EDR	100/200 V AC, DC input×8, Relay output×6	8-28 DIO	
	EH-D14EDR	24 V DC, DC input×8, Relay output×6	8-28 DIO	
	EH-D14EDT	24 V DC, DC input×8, Transistor output×6	8-28 DIO	Sink
	EH-D14EDTP	24 V DC, DC input×8, Transistor output×6	8-28 DIO	Source
	EH-D14EDTPS	24 V DC, DC input×8, Transistor output×6 (short circuit protection)	8-28 DIO	Source
8 Points Expansion unit	EH-D8ED	24 V DC, DC input×8	8-28 DIO	
	EH-D8ER	24 V DC, Relay output×8	8-28 DIO	
	EH-D8ET	24 V DC, Transistor output×8	8-28 DIO	Sink
	EH-D8ETPS	24 V DC, Transistor output×8 (short circuit protection)	8-28 DIO	Source
	EH-D8EDR	24 V DC, DC input×4, Relay output×4	8-28 DIO	
	EH-D8EDT	24 V DC, DC input×4, Transistor output×4	8-28 DIO	Sink
	EH-D8EDTPS	24 V DC, DC input×4, Transistor output×4 (short circuit protection)	8-28 DIO	Source
Analog Expansion unit	EH-A6EAN	100/200 V AC, Analog input×4, Analog output×2	AIO	
	EH-D6EAN	24 V DC, Analog input×4, Analog output×2	AIO	
RTD Expansion unit	EH-A6ERTD	100/200 V AC, RTD input×4, Analog output×2	AIO	
	EH-D6ERTD	24 V DC, RTD input×4, Analog output×2	AIO	
	EH-A4ERTD	100/200 V AC, RTD input×4	AIO	
	EH-D4ERTD	24 V DC, RTD input×4	AIO	
Thermocouple Expansion unit	EH-D6ETC	24 V DC, Thermocouple input×4, Analog output×2	AIO	
	EH-D4ETC	24 V DC, Thermocouple input×4	AIO	
Option board	OBV-NES	RS-485 communication board	OptionBoard	
	OBV-485A	RS-485 communication board with Analog input 2 ch. (10-bit)	OptionBoard	
Expansion cables	EH-MCB10	Expansion cable (1m)	—	
	EH-MCB05	Expansion cable (0.5m)	—	
	EH-MCB01	Expansion cable (0.1m)	—	
Battery	MV-BAT	Lithium battery for retentive data and RTC	—	

2.4 Consumption current

Table 2.5 List of consumption current

Type	weight (g)	Consumption current (A)			Remarks
		100VAC	264VAC	24VDC	
		Steady-state	Steady-state	Steady-state	
MV-A64DR	730	0.2	0.1	-	
MV-D64DR	655	-	-	0.5	
MV-D64DT	600	-	-	0.5	
MV-D64DTPS	600	-	-	0.5	
MV-A40DR	570	0.2	0.1	-	
MV-D40DR	500	-	-	0.4	
MV-D40DT	460	-	-	0.4	
MV-D40DTPS	460	-	-	0.4	
EH-A64EDR	720	0.4	0.2	-	
EH-D64EDR	640	-	-	0.5	
EH-D64EDT	640	-	-	0.4	
EH-D64EDTPS	640	-	-	0.4	
EH-A28EDR	600	0.2	0.06	-	
EH-D28EDR	500	-	-	0.3	
EH-D28EDT	500	-	-	0.2	
EH-D28EDTP	500	-	-	0.2	
EH-D28EDTPS	500	-	-	0.2	
EH-D16ED	260	-	-	0.13	
EH-D16ER	300	-	-	0.11	
EH-D16ET	260	-	-	0.03	
EH-D16ETPS	260	-	-	0.04	
EH-A14EDR	400	-	-	0.16	
EH-D14EDR	400	-	-	0.16	
EH-D14EDT	300	-	-	0.16	
EH-D14EDTP	300	-	-	0.16	
EH-D14EDTPS	300	-	-	0.16	
EH-D8ED	260	-	-	0.16	
EH-D8ER	280	-	-	0.16	
EH-D8ET	260	-	-	0.16	
EH-D8ETPS	260	-	-	0.16	
EH-D8EDR	300	-	-	0.16	
EH-D8EDT	260	-	-	0.16	
EH-D8EDTPS	260	-	-	0.16	
EH-A6EAN	400	0.1	0.06	-	
EH-D6EAN	300	-	-	0.16	
EH-A6ERTD	400	0.1	0.06	-	
EH-D6ERTD	300	-	-	0.16	
EH-A4ERTD	400	0.1	0.06	-	
EH-D4ERTD	300	-	-	0.16	
EH-D6ETC	300	-	-	0.16	
EH-D4ETC	300	-	-	0.16	

2.5 Input specifications

Item		Specification		Internal circuit diagram
		MVH: Input 0, 2, 4, 6, 8	Other than the left column	
Input voltage		24V DC		
Allowable input voltage range		0 to 30V DC		
Input impedance		Approx. 2.7 kΩ	Approx. 4.7 kΩ	
Input current		8 mA typical	4.8 mA typical	
Operating voltage	ON	18 VDC (min) / 4.5mA (max)	18 VDC (min) / 3.3mA (max)	
	OFF	5 VDC (max) / 1.8mA (max)	5 VDC (max) / 1.6mA (max)	
Input lag	OFF → ON	0.5 to 20 ms (configurable)		
	ON → OFF	0.5 to 20 ms (configurable)		
Input points		64-point type: 40 40-point type: 24		
Common		2		
Polarity		None		
Insulation system		Photocoupler insulation		
Input display		LED indication		
External		Removable type screw terminal block (M3)		

■ High-speed counter specifications

Item		Single phase	2-phase
Input		0, 2, 4, 6, 8	Combined use of 0 and 2 / 4 and 6
Input voltage	ON	18 V	
	OFF	5 V	
Count pulse width		10 μs	17 μs
Highest counting frequency		100 kHz each channels	60 kHz each channels
Count Resister		32-bit	
Coincidence output		Available	
On / Off-preset		Available	
Upper / Lower bound setting		Not supported (Ring counter of 0 to 4,294,967,295)	
Preload / Strobe		Available	

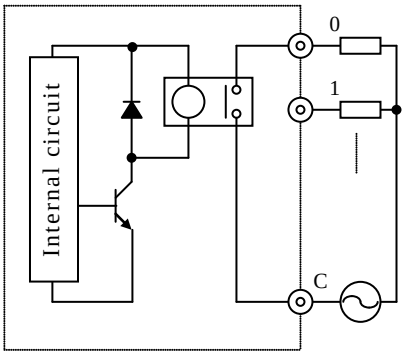
Refer to page 4-5 for wiring to a rotary encoder.

■ Interrupt input specifications

Item		Specification
Input		0, 2, 4, 6, 8
Input voltage	ON	18 V
	OFF	5 V

2.6 Output specifications

(1) Relay output (All output on MV-*64DR, MV-*40DR)

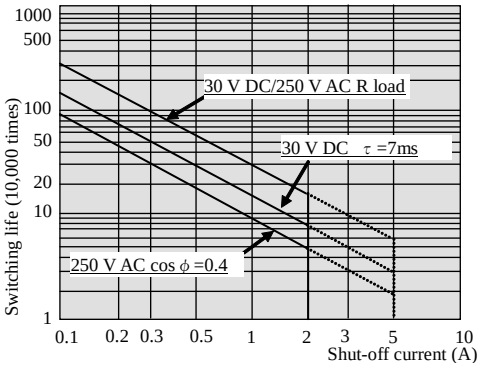
Item		Specifications	Internal circuit diagram
Rated load voltage		5 to 250V AC, 5 to 30V DC	
Minimum switching current		1 mA (5V DC) *1	
Maximum load current	1 circuit	2A (24V DC, 240V AC)	
	1 common	5A	
Output response time	OFF → ON	15 ms (max.)	
	ON → OFF	15 ms (max.)	
Output points		64-point type: 24 40-point type: 16	
Common		64-point type: 9 40-point type: 5	
Surge removing circuit		None	
Fuse		None	
Insulation system		Relay insulation	
Output display		LED indication	
External connection		Removable type screw terminal block (M3)	
Externally supplied power (for driving the relays)		Not required	
Contact life *2		20,000,000 times (mechanical) 200,000 times (electrical: 1.5A)	
Insulation		1,500V or more (external-internal) 500V or more (external-external)	

*1: Minimum switching current is the unofficial value as a guide. This value may vary with the switching frequency, environmental conditions, etc..

*2: Refer the diagram below

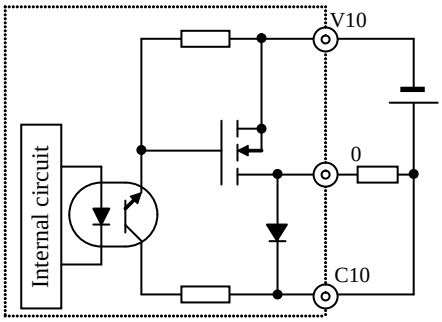
■ Life of relay contact

Life curve of relay contact



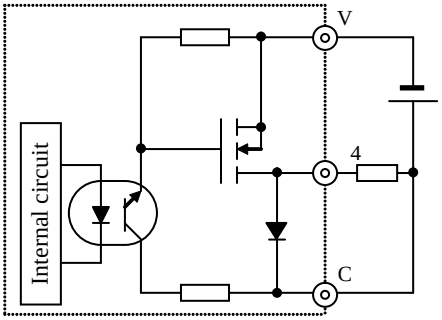
Life of the contact is also in squared reverse proportion to the current, so be aware that interrupting rush current or directly driving the capacitor load will drastically reduce the life of the relay.

(2) DC output（Output 0 to 2 on MV-D64DT, MV-D40DT）

Item		Specifications	Internal circuit diagram
Output specifications		Transistor output (sink type)	
Rated load voltage		24 / 12 V DC (+10%, -15%)	
Minimum switching current		10 mA	
Leak current		0.1 mA (max.)	
Maximum Load current	1 circuit	0.5 A 24VDC, 0.3 A 12VDC	
	1 common	2 A	
Output response time	OFF → ON	5 μs (max.) 24 V DC 0.2 A	
	ON → OFF	5 μs (max.) 24 V DC 0.2 A	
Output points		3	
Common		1	
Surge removing circuit		None	
Fuse		None	
Insulation system		Photocoupler insulation	
Output display		LED indication	
External connection		Removable type screw terminal block (M3)	
Externally supplied power *		12 to 30 V DC	
Insulation		1,500 V or more (external-internal) 500 V or more (external-external)	
Output voltage drop		0.3 V DC (max.)	

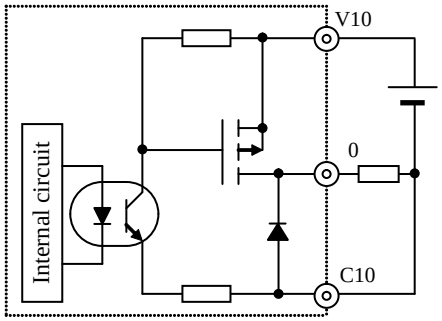
*: It is required to supply 12 to 30V DC between the V and C terminal externally.

(3) DC output (Except Output 0 to 2 on MV-D64DT, MV-D40DT)

Item		Specifications	Internal circuit diagram
Output specifications		Transistor output (sink type)	
Rated load voltage		24 / 12 V DC (+10%, -15%)	
Minimum switching current		10 mA	
Leak current		0.1 mA (max.)	
Maximum Load current	1 circuit	0.5 A	
	1 common	64-point type: 3 A 40-point type: 5 A	
Output response time	OFF → ON	0.1 ms (max.) 24 V DC	
	ON → OFF	0.1 ms (max.) 24 V DC	
Output points		64-point type: 21 40-point type: 13	
Common		64-point type: 3 40-point type: 1	
Surge removing circuit		None	
Fuse		None	
Insulation system		Photocoupler insulation	
Output display		LED indication	
External connection		Removable type screw terminal block (M3)	
Externally supplied power *		12 to 30 V DC	
Insulation		1,500 V or more (external-internal) 500 V or more (external-external)	
Output voltage drop		0.3 V DC (max.)	

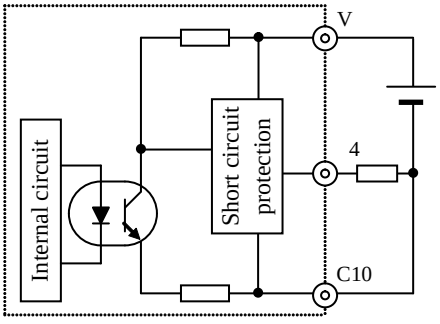
*: It is required to supply 12 to 30V DC between the V and C terminal externally.

(4) DC output (Output 0 to 3 on MV-D64DTPS, MV-D40DTPS)

Item		Specifications	Internal circuit diagram
Output specifications		Transistor output (Source type)	
Rated load voltage		24 / 12 V DC (+10%, -15%)	
Minimum switching current		10 mA	
Leak current		0.1 mA (max.)	
Maximum Load current	1 circuit	0.5 A 24VDC, 0.3 A 12VDC	
	1 common	2 A	
Output response time	OFF → ON	Output 0 to 2: 5 μs (max.) 24 V DC 0.2 A Output 3: 0.5 ms (max.) 24 V DC	
	ON → OFF	Output 0 to 2: 5 μs (max.) 24 V DC 0.2 A Output 3: 0.5 ms (max.) 24 V DC	
Output points		4	
Common		1	
Surge removing circuit		None	
Fuse		None	
Insulation system		Photocoupler insulation	
Output display		LED indication	
External connection		Removable type screw terminal block (M3)	
Externally supplied power *		12 to 30 V DC	
Insulation		1,500 V or more (external-internal) 500 V or more (external-external)	
Output voltage drop		0.3 V DC (max.)	

*: It is required to supply 12 to 30V DC between the V and C terminal externally.

(5) DC output (Except Output 0 to 3 on MV-D64DTPS, MV-D40DTPS)

Item		Specifications	Internal circuit diagram
Output specifications		Transistor output (Source, short circuit protection)	
Rated load voltage		24 / 12 V DC (+10%, -15%)	
Minimum switching current		10 mA	
Leak current		0.1 mA (max.)	
Maximum Load current	1 circuit	0.7 A	
	1 common	64-point type: 3.0 A 40-point type: 5.0 A	
Output response time	OFF → ON	0.5 ms (max.) 24 V DC	
	ON → OFF	0.5 ms (max.) 24 V DC	
Output points		64-point type: 20 40-point type: 12	
Common		64point type: 3 40-point type: 1	
Surge removing circuit		None	
Fuse		None	
Insulation system		Photocoupler insulation	
Output display		LED indication	
External connection		Removable type screw terminal block (M3)	
Externally supplied power *		12 to 30 V DC	
Insulation		1,500 V or more (external-internal) 500 V or more (external-external)	
Output voltage drop		0.3 V DC (max.)	

*: It is required to supply 12 to 30V DC between the V and C terminal externally.

■ Pulse train output / PWM output specifications

Item	64-point / 40-point transistor output
Output	0 to 2 (configurable)
Load voltage	12 / 24 V
PWM highest output frequency *	65,535Hz
Pulse train highest output frequency *	65,535Hz

*: Though Pulse train output and PWM output are available for relay output type, output cannot keep up with high frequencies. These output should be used at the operating frequency upon confirmation.

2.7 Power supply for sensor

24V terminal on the input terminal block can supply current to the external equipments. If this output is used as the power supply for the input of own unit, the remaining can be used as power supply for the sensors. The current (I) allowed using as the power supply for sensors can be calculated by the following formula.

$I = 430\ mA – (5\ mA^* \times NI + 5mA \times NO)$

NI: number of input that are turned on at the same time

NO: number of output that are turned on at the same time

*: Substitute 10mA in regard to Input 0, 2, 4, 6, 8.

2.8 Serial port specifications

2.8.1 Physical layer interface

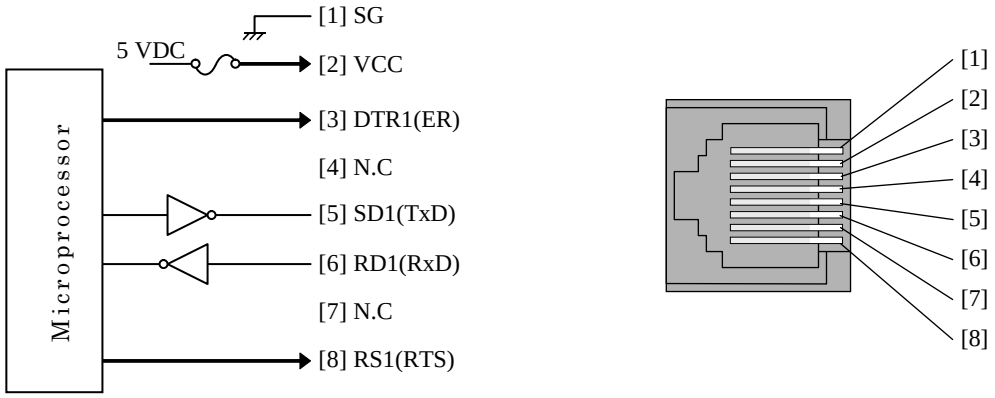


Figure 2.1 Circuit diagram and Pin No. of RS-232C

Table 2.6 List of signal of RS-232C

Pin No.	Signal name	Direction		Meaning
		PLC	Host	
[1]	SG	◄——►		Signal ground
[2]	VCC	——►		5 V DC is supplied. (Protective fuse is connected.)
[3]	DTR1(ER)	——►		When PLC is ready to communicate, this signal is high.
[4]	N.C			Don't connect.
[5]	SD1(TxD)	——►		Sending data from PLC
[6]	RD1(RxD)	◄——		Receiving data to PLC
[7]	N.C			Don't connect.
[8]	RS1(RTS)	——►		When PLC is ready to receive data, this signal is high.

2.8.2 RS-232C communication specifications

Specifications of RS-232C communication are shown in Table 2.7.

Table 2.7 RS-232C communication specifications

Item	Specifications
Transmission speed	4,800 bps / 9,600 bps / 19,200 bps / 38,400bps / 57,600bps / 115,200bps *
Interface	RS-232C
Maximum cable length	15 m
Connection mode	1 : 1
Synchronization method	Start-stop synchronization
Supported function	Modbus-RTU master/slave, General purpose communication
Transmission method	Serial transmission (bit serial transmission)
Transmission code outgoing sequence	Send out from the lowest bit in character
Error control	Vertical parity check, sum check, overrun check, framing check
Transmission unit	Message unit (variable length)
Maximum message length	1,024 bytes (including control characters)

*: Communication in 115.2kbps could be unstable depending on PC. If so, change the baud rate to 57.6kbps or slower.

2.9 USB communication port specifications

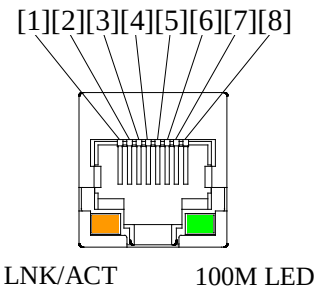
USB communication port is the dedicated one for Communication with EHV-CODESYS (Gateway).

Table 2.8 USB communication port specifications

Item	Specifications
Interface	Conforms to USB2.0 Full Speed
Maximum cable length	Less than 3m
Connection mode	1 : 1
Connector	USB Type B

2.10 Ethernet port specifications

2.10.1 Physical layer interface



Pin No.	Signal name	Remarks
[1]	TXD+	Transmit Data (+)
[2]	TXD-	Transmit Data (-)
[3]	RXD+	Received Data (+)
[4]	—	Unused
[5]	—	Unused
[6]	RXD-	Received Data (-)
[7]	—	Unused
[8]	—	Unused

2.10.2 Ethernet communication specifications

Table 2.9 Ethernet communication specifications

Item	Specifications
Ethernet standard	Conforms to IEEE802.3 (10BASE-T/100BASE-TX (Automatic recognition))
Transmission modulation	baseband
Media access method	CSMA / CD
Maximum segment length	100 m
Connector	8-pin modular connector (RJ-45)
Cable	Category 5 STP or UTP (STP recommended)

2.11 USB memory port specifications

Table 2.10 USB memory port specifications

Item	Specifications
Interface	USB2.0 Full Speed (12M)
Connect to USB-HUB	Not supported
Supported format	FAT32 (Recommended), FAT16 (2GB USB memory of FAT16 is not recommended because access speed is not fast enough.)
Max. USB memory size	32GB (FAT32), 2GB (FAT16)
Max. file size	2GB
Max. file name	99 characters (ASCII only, Unicode is not supported.)
Not allowed characters for file/directory name	¥ / : * ? " < > |
Max. number of files in root directory	7281
Max. simultaneous open file	10

Note

- 2GB USB memory of FAT16 is not recommended because access speed is not fast enough, which could result in overload exception especially for Webvisualization.
- To avoid contact failure due to vibration, be sure to use small type USB memory device as follows.

2.12 Backup

(1) Battery (optional)

Retentive data and RTC data can be kept by MV-BAT battery as below.
If the RTC function is used, be sure to use the battery.

Battery life (Total power failure time)[year]	
Guaranteed value (MIN) @55°C	Actual value (MAX) @25°C
5	10

(2) Capacitor

Retentive data can be kept for 12 hours (at 25 °C) by the capacitor.

*: To keep the retentive data or RTC data for 12 hours, it is required to charge the capacitor by supplying power to PLC for more than 20 minutes.

2.13 LED indication

The operating condition and the status of the external I/O can be checked on the LED display of the front cover.

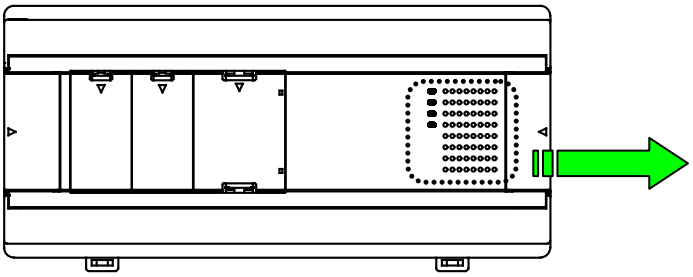


Figure 2.2 Front cover of MICRO-EHV+

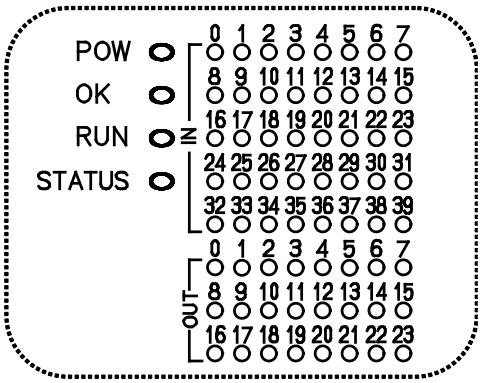


Figure 2.3 LED display (64-point type)

(1) POW LED

POW LED indicates whether power is properly supplied to the basic unit.

(2) OK LED

OK LED indicates the status of self-diagnostics.

(3) RUN LED

RUN LED indicates operating status.

(4) STATUS LED

STATUS LED indicates system initialization or FLASH memory writing.

(5) IN LED

IN LED indicates input status.

(6) OUT LED

OUT LED indicates output status.

Name	Status	Description
POW	ON	Power source is supplied.
	OFF	Power source is not supplied.
OK	ON	Normal operation (PLC is ready to run)
	Blinking	Exception or warning status. Refer to Section 3.25 Troubleshooting.
	OFF	Initialization status during power-up or Error status. Refer to Section 3.25 Troubleshooting.
RUN	ON	RUN status
	OFF	STOP status
STATUS	Blinking	Initialization status during power-up or flash memory is being written
	OFF	Normal operation
IN	ON	Input is ON status
	OFF	Input is OFF status
OUT	ON	Output is ON status
	OFF	Output is OFF status

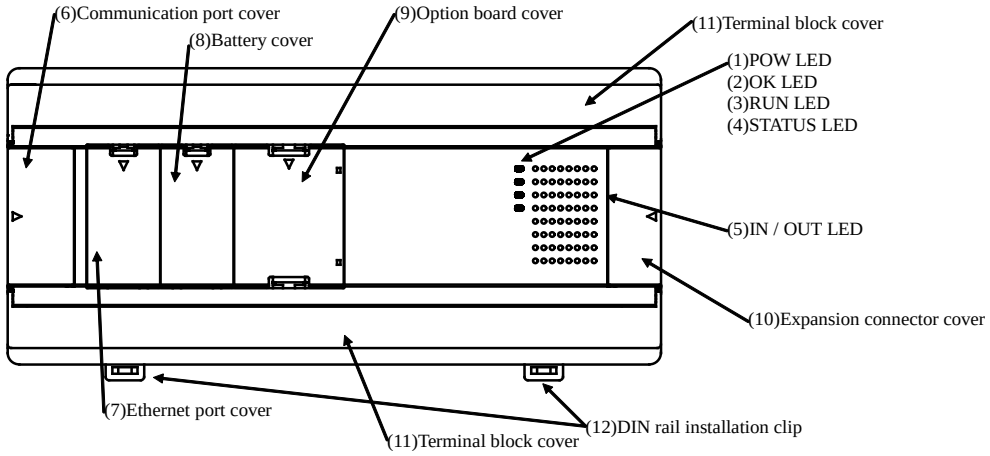
Note

If boot project file is corrupt or no boot project is stored in USB and internal FLASH memory initialization during power-up does not complete, which causes STATUS LED keeps blinking and OK LED does not light up. In this case, download a right project to recover.

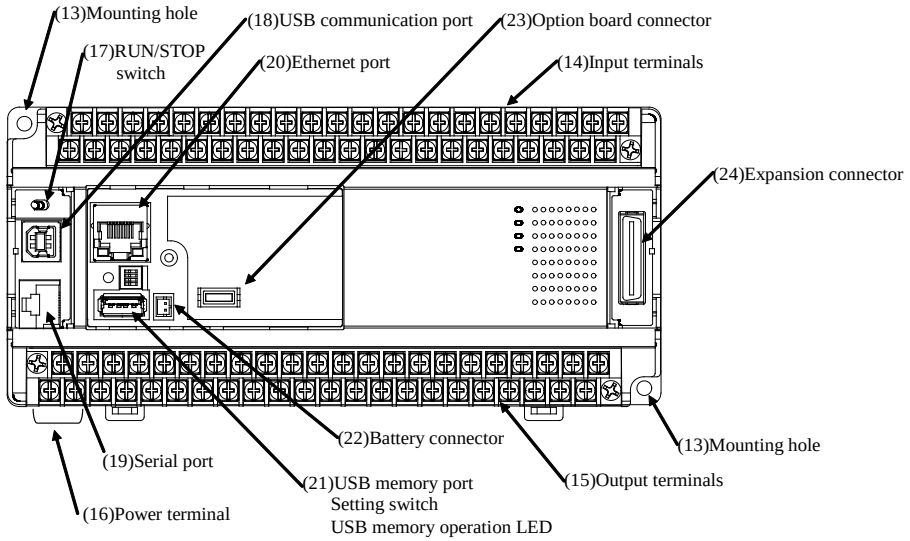
2.14 64-point Basic unit

2.14.1 Name and function of each part

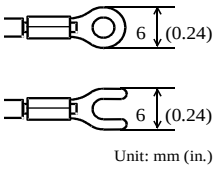
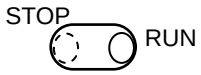
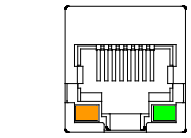
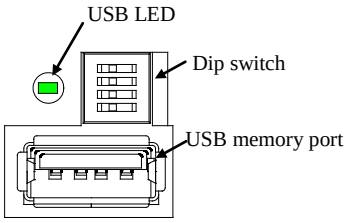
Name and function of each part	Type	MV-A64DR, MV-D64DR, MV-D64DT, MV-D64DTPS	
	Weight	MV-A64DR: 730g (1.61 lb.)	MV-D64DR: 655g (1.44 lb.)
		MV-D64DT: 600g (1.32 lb.)	MV-D64DTPS: 600g (1.32 lb.)



Without the front cover



No.	Item	Description
(1)	POW LED	indicates that the power is supplied.
(2)	OK LED	represents the result of the self-diagnosis in the basic unit. Normal: ON, Abnormal: blink or OFF (Refer to Section 3.25 Troubleshooting)
(3)	RUN LED	displays the operating condition. (RUN: ON, STOP: OFF)
(4)	STATUS LED	blinks until the state in which PLC can RUN from power-up (500ms ON / 500ms OFF) or during writing the program to FLASH memory.
(5)	IN / OUT LED	When signal status is ON, LED lights up accordingly.
(6)	Communication port cover	The cover for RUN/STOP switch, serial port and USB communication port.
(7)	Ethernet port cover	The cover for Ethernet port and USB memory port.
(8)	Battery cover	The cover for the battery connector.
(9)	Option board cover	The cover for the option board connector. Avoid contact with the printed circuit board when you remove the cover.
(10)	Expansion connector cover	The cover for the expansion connector.
(11)	Terminal block cover	The cover for the terminal block.

No.	Item	Description
(12)	DIN rail installation clip	This is used when mounting to a DIN rail.
(13)	Mounting hole	Use these holes when installing with screws. (M4×200 mm (0.79 in.))
(14)	Input terminals	The terminal block to connect input signals. The terminal screw size is M3. The terminal block supports 0.36 to 2.1 mm^2 (AWG22 to 14). If 2 crimping terminals are connected to one terminal screw, use 0.36 to 1.3 mm^2 (AWG20 to 16) cable.  (Recommended) Handle very carefully since cable could be detached when screw is loose.
(15)	Output terminals	The terminal block to connect output signals. The wiring Specification is the same as the input terminal.
(16)	Power terminal	The terminal for connecting the power supply. The wiring Specification is the same as the input terminal.
(17)	RUN/STOP switch	When this switch position is in RUN, CPU start executing program. At the same time, remote controlling is enabled, in which case, CPU is started or stopped by EHV-CODESYS over communication. When this switch position is in STOP, CPU stops executing program. In this status, remote controlling is disabled. 
(18)	USB communication port	USB port supports gateway function (with EHV-CODESYS) only. USB cable is not included with CPU package nor supplied by Hitachi-IES. Use type-B USB cable.
(19)	Serial port	Serial port supports IEC programming function supporting Modbus-RTU master/slave communication and general purpose communication. Port setting is fixed to RS-232C.
(20)	Ethernet port	Ethernet port has both gateway function (with EHV-CODESYS) and IEC programming function supporting Modbus-TCP server. In addition, network variables are transferred to/from other MICRO-EHV+ PLCs and EHV+ CPUs over Ethernet network. LNK/ACT LED lights when connecting the cable. The LED blinks when sending and receiving data. 100Mbps communication: 100M LED lights 10Mbps communication: 100M LED turns off 
(21)	- USB memory port - Dip switch - USB LED	USB memory port: USB host function (Program transfer, Data logging and Web visualization) is supported. Dip switch: User program can be downloaded, uploaded or verified according to switch position. Refer to section 3.21 for details. USB LED: LED indicates the status of USB memory function. 
(22)	Battery connector	This is a connector to battery. Following data are maintained by battery. (1) Data specified as VAR RETAIN and VAR PERSISTENT (2) RTC (real time clock) data ⚠ Caution - Replacement of the lithium battery shall be done by a trained technician only. - The battery has polarity. When plugging in, check the polarity carefully. - Refer to the table on section 5.2 for the life of battery.
(23)	Option board connector	This is a connector to option board.
(24)	Expansion connector	This is a connector to expansion cable. MICRO-EHV+ can connect maximum 4 expansion units.

⚠ Caution

Note the following matters for the communication port.

(1) Do not connect Ethernet cable to the serial port of CPU module. This could cause damage the CPU or connected equipment.

(2) In 100BASE-TX (100Mbps) communication of Ethernet, connection could be unstable due to external noise depending on cable length, installation environment and etc. In this case, increase the number of times to retry in connected device.

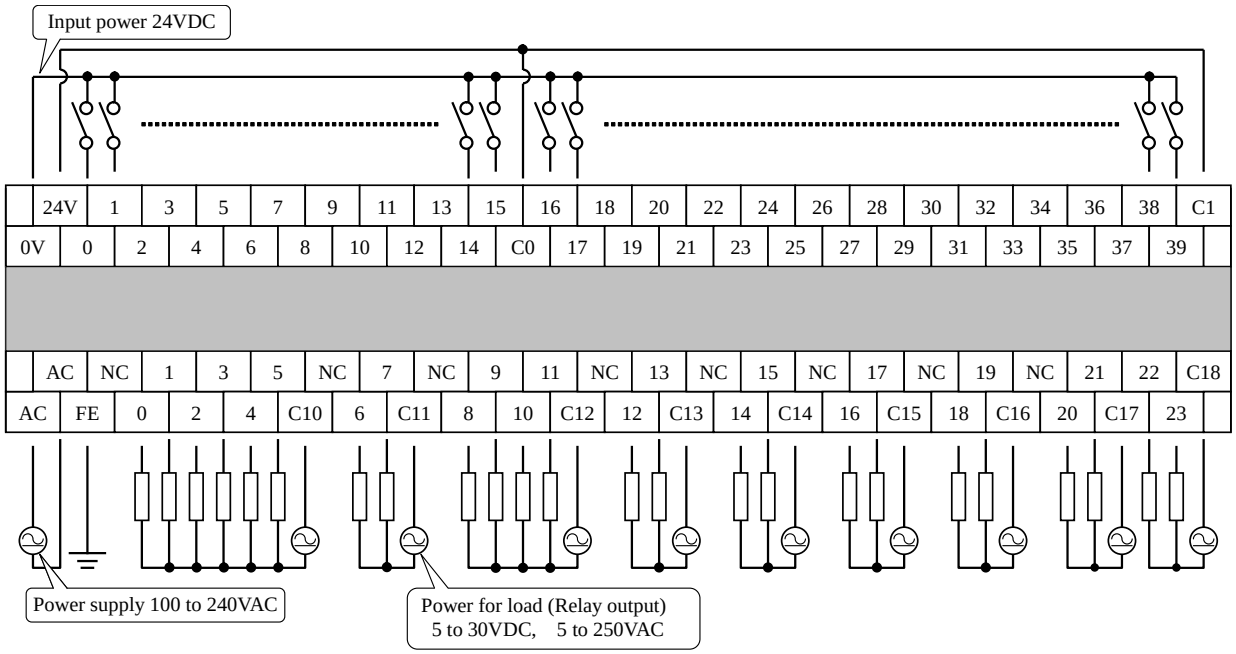
(3) USB communication could be unstable under severe noise environment. Be sure to use short cable and route apart from power line or other communication cables.

(4) Serial communication in 115.2kbps could be unstable depending on PC. If so, change the baud rate to 57.6kbps or slower.

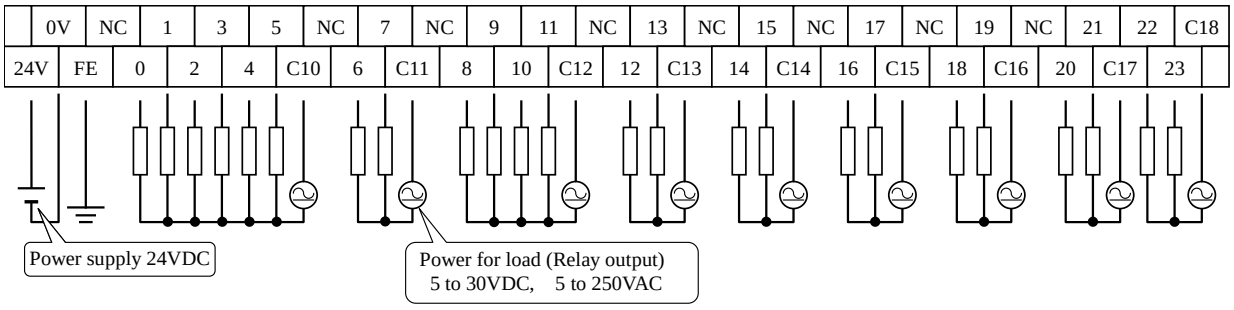
2.14.2 Terminal layout and wiring

(1) MV-A64DR (AC power type)

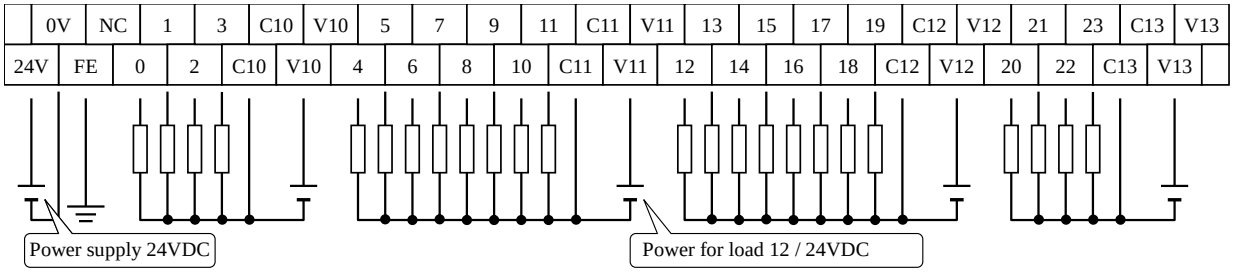
*: For the DC input, both sink and source types are available. It is possible to reverse the polarity of 24 V DC. Refer to page 4-5 for wiring to a rotary encoder.



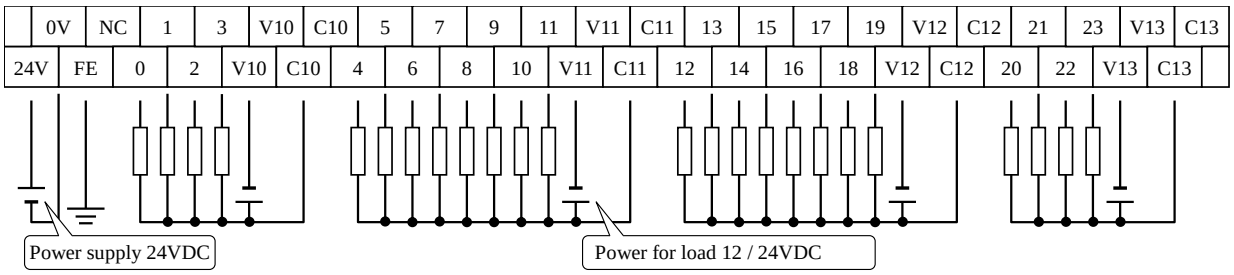
(2) MV-D64DR (DC power type) Input wiring is same as MV-A64DR.



(3) MV-D64DTPS (DC power type) Input wiring is same as MV-A64DR



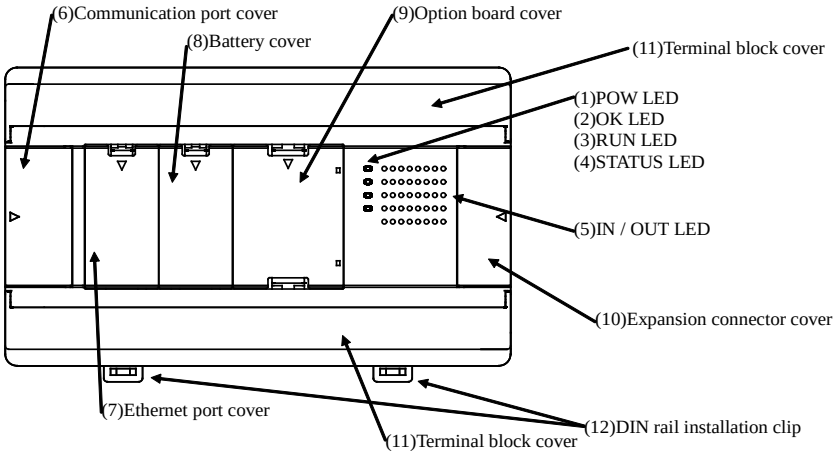
(4) MV-D64DT (DC power type) Input wiring is same as MV-A64DR



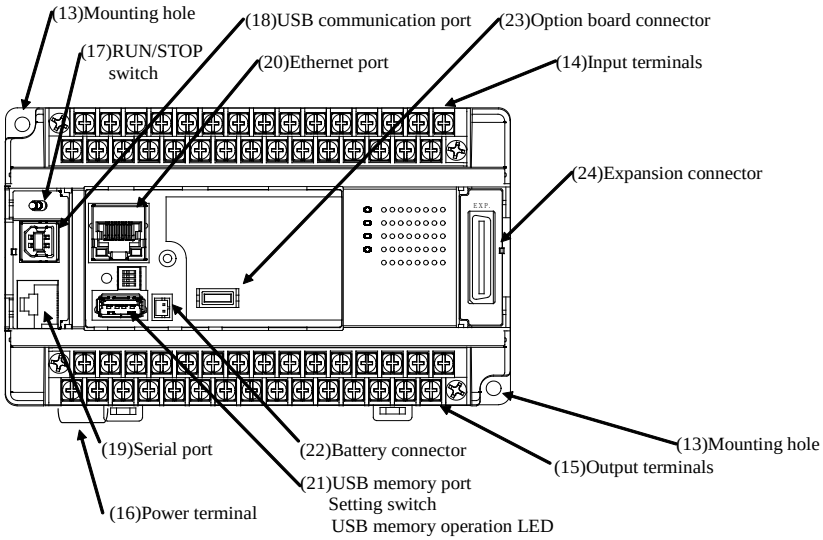
2.15 40-point Basic unit

2.15.1 Name and function of each part

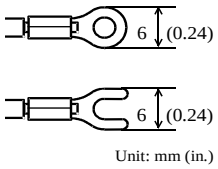
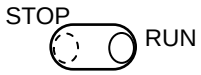
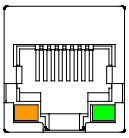
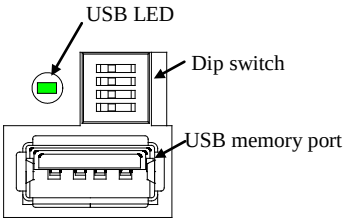
Name and function of each part	Type	MV-A40DR, MV-D40DR, MV-D40DT, MV-D40DTPS	
	Weight	MV-A40DR : 570g (1.25 lb.)	MV-D40DR : 500g (1.10 lb.)
		MV-D40DT : 460g (1.01 lb.)	MV-D40DTPS : 460g (1.01 lb.)



[Without the front cover]



No.	Item	Description
(1)	POW LED	indicates that the power is supplied.
(2)	OK LED	represents the result of the self-diagnosis in the basic unit. Normal: ON, Abnormal: blink or OFF (Refer to Section 3.25 Troubleshooting)
(3)	RUN LED	displays the operating condition. (RUN: ON, STOP: OFF)
(4)	STATUS LED	blinks until the state in which PLC can RUN from power-up (500ms ON / 500ms OFF) or during writing the program to FLASH memory.
(5)	IN / OUT LED	When signal status is ON, LED lights up accordingly.
(6)	Communication port cover	The cover for RUN/STOP switch, serial port and USB communication port.
(7)	Ethernet port cover	The cover for Ethernet port and USB memory port.
(8)	Battery cover	The cover for the battery connector.
(9)	Option board cover	The cover for the option board connector. Avoid contact with the printed circuit board when you remove the cover.
(10)	Expansion connector cover	The cover for the expansion connector.
(11)	Terminal block cover	The cover for the terminal block.

No.	Item	Description
(12)	DIN rail installation clip	This is used when mounting to a DIN rail.
(13)	Mounting hole	Use these holes when installing with screws. (M4×200 mm (0.79 in.))
(14)	Input terminals	The terminal block to connect input signals. The terminal screw size is M3. The terminal block supports 0.36 to 2.1 mm^2 (AWG22 to 14). If 2 crimping terminals are connected to one terminal screw, use 0.36 to 1.3 mm^2 (AWG20 to 16) cable.  (Recommended) Handle very carefully since cable could be detached when screw is loose.
(15)	Output terminals	The terminal block to connect output signals. The wiring Specification is the same as the input terminal.
(16)	Power terminal	The terminal for connecting the power supply. The wiring Specification is the same as the input terminal.
(17)	RUN/STOP switch	When this switch position is in RUN, CPU start executing program. At the same time, remote controlling is enabled, in which case, CPU is started or stopped by EHV-CODESYS over communication. When this switch position is in STOP, CPU stops executing program. In this status, remote controlling is disabled. 
(18)	USB communication port	USB port supports gateway function (with EHV-CODESYS) only. USB cable is not included with CPU package nor supplied by Hitachi-IES. Use type-B USB cable.
(19)	Serial port	Serial port supports IEC programming function supporting Modbus-RTU master/slave communication and general purpose communication. Port setting is fixed to RS-232C.
(20)	Ethernet port	Ethernet port has both gateway function (with EHV-CODESYS) and IEC programming function supporting Modbus-TCP server. In addition, network variables are transferred to/from other MICRO-EHV+ PLCs and EHV+ CPUs over Ethernet network. LNK/ACT LED lights when connecting the cable. The LED blinks when sending and receiving data. 100Mbps communication: 100M LED lights 10Mbps communication: 100M LED turns off 
(21)	- USB memory port - Dip switch - USB LED	USB memory port: USB host function (Program transfer, Data logging and Web visualization) is supported. Setting switch: User program can be downloaded, uploaded or verified according to switch position. Refer to section 3.21 for details. USB LED: LED indicates the status of USB memory function. 
(22)	Battery connector	This is a connector to battery. Following data are maintained by battery. (2) Data specified as VAR RETAIN and VAR PERSISTENT (2) RTC (real time clock) data ⚠ Caution - Replacement of the lithium battery shall be done by a trained technician only. - The battery has polarity. When plugging in, check the polarity carefully. - Refer to the table on section 5.2 for the life of battery.
(23)	Option board connector	This is a connector to option board.
(24)	Expansion connector	This is a connector to expansion cable. MICRO-EHV+ can connect maximum 4 expansion units.

⚠ Caution

Note the following matters for the communication port.

(1) Do not connect Ethernet cable to the serial port of CPU module. This could cause damage the CPU or connected equipment.

(2) In 100BASE-TX (100Mbps) communication of Ethernet, connection could be unstable due to external noise depending on cable length, installation environment and etc. In this case, increase the number of times to retry in connected device.

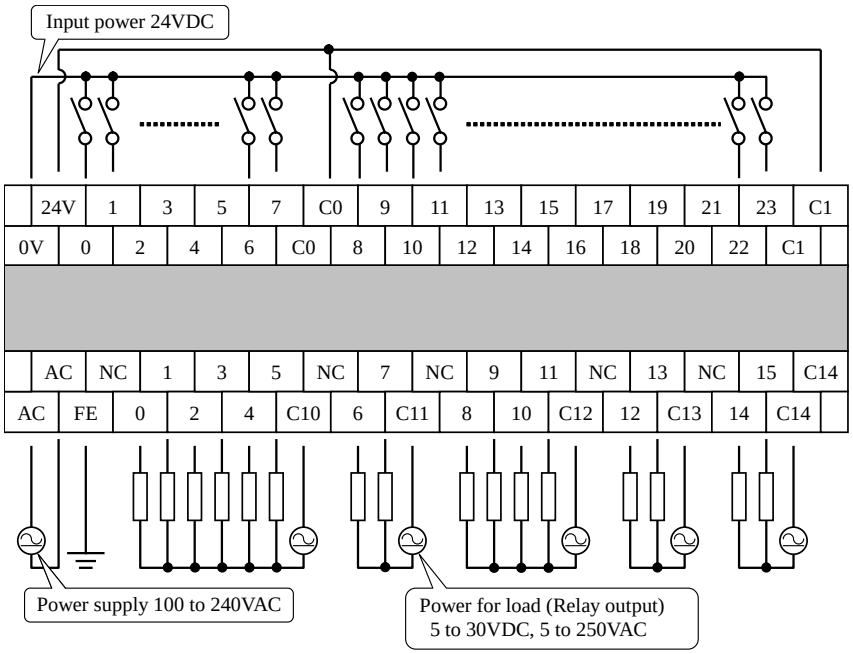
(3) USB communication could be unstable under severe noise environment. Be sure to use short cable and route apart from power line or other communication cables.

(4) Serial communication in 115.2kbps could be unstable depending on PC. If so, change the baud rate to 57.6kbps or slower.

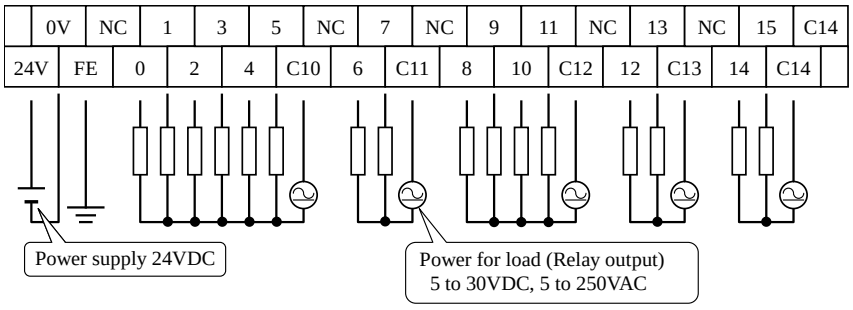
2.15.2 Terminal layout and wiring

(1) MV-A40DR (AC power type)

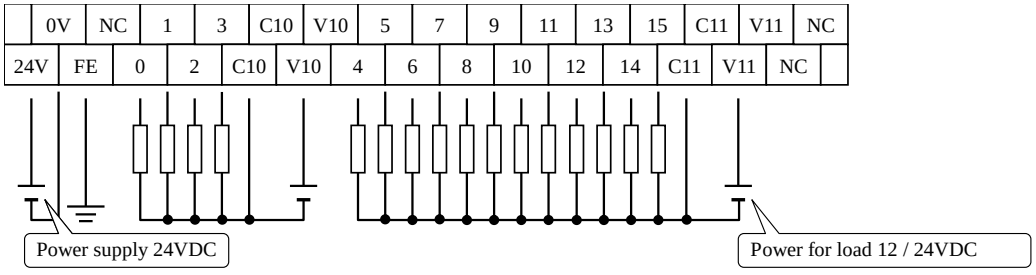
*: For the DC input, both sink and source types are available. It is possible to reverse the polarity of 24 V DC. Refer to page 4-5 for wiring to a rotary encoder.



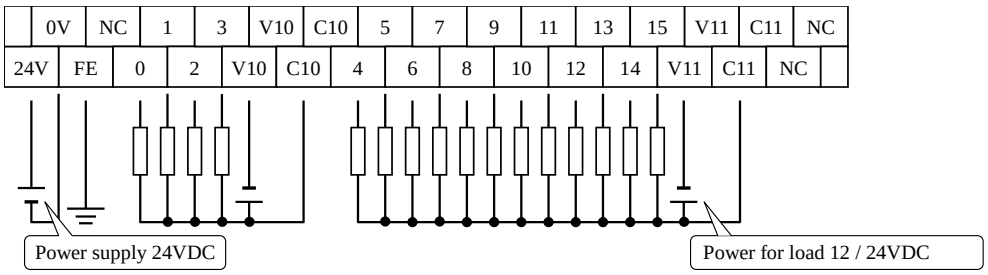
(2) MV-D40DR (DC power type) Input wiring is same as MV-A40DR.



(3) MV-D40DTPS (DC power type) Input wiring is same as MV-A40DR.



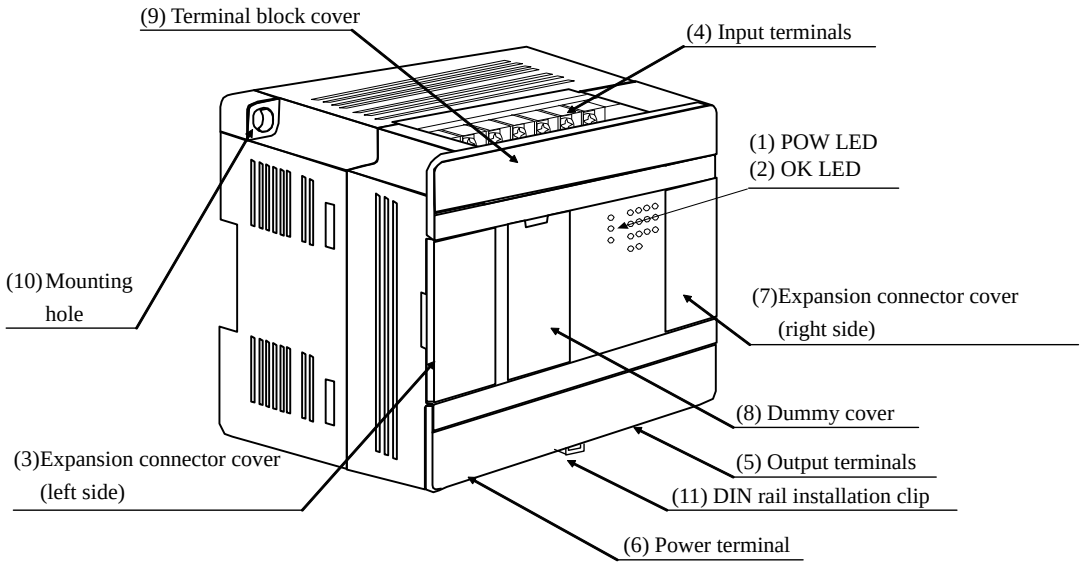
(4) MVH-D40DT (DC power type) Input wiring is same as MV-A40DR.

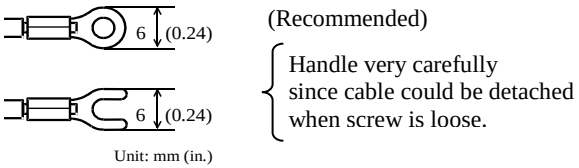


2.16 Expansion unit

2.16.1 Name and function of each part

Name and function of each part	Type	EH-A64EDR, EH-D64EDR, EH-D64EDT, EH-D64EDTPS
		EH-A28EDR, EH-D28EDR, EH-D28EDT, EH-D28EDTP, EH-D28EDTPS
		EH-D16ED, EH-D16ER, EH-D16ET, EH-D16ETPS
		EH-A14EDR, EH-D14EDR, EH-D14EDT, EH-D14EDTP, EH-D14EDTPS
		EH-D8ED, EH-D8ER, EH-D8ET, EH-D8ETPS, EH-D8EDR, EH-D8EDT, EH-D8EDTPS
		EH-A6EAN, EH-D6EAN
		EH-A6ERTD, EH-D6ERTD, EH-A4ERTD, EH-D4ERTD
		EH-D6ETC, EH-D4ETC
	Weight	Refer to the page 2-4.



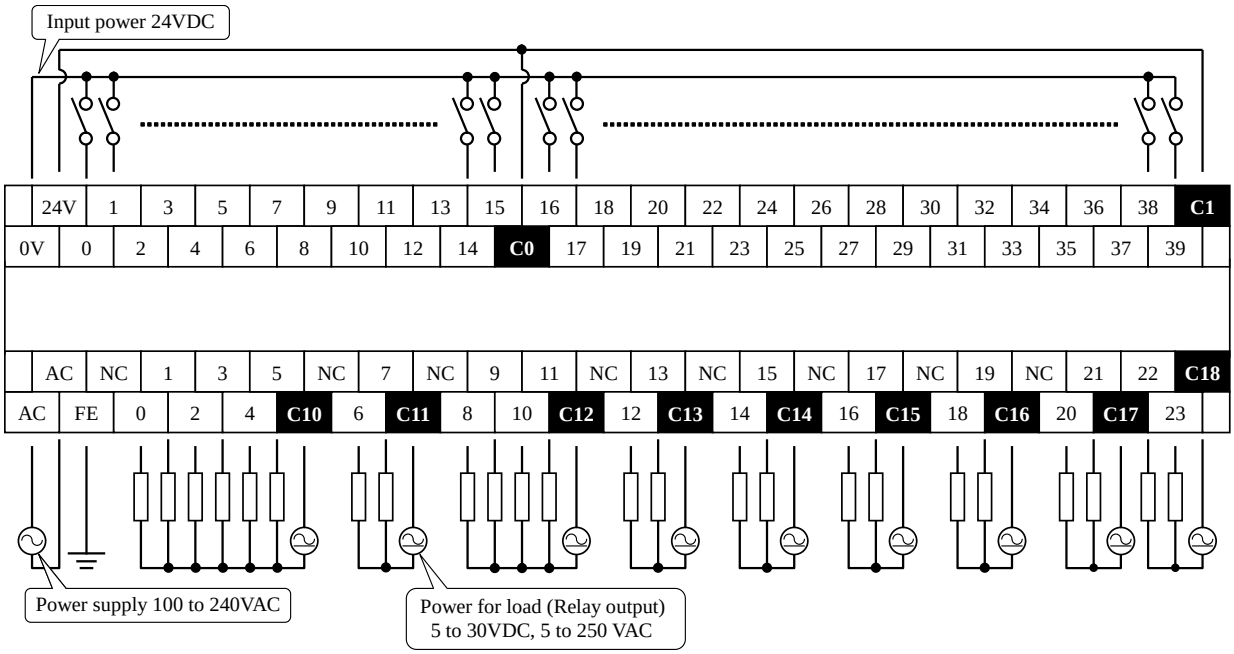
No.	Item	Description
(1)	POW LED	indicates that the power is supplied.
(2)	OK LED	represents the result of the self-diagnosis in the basic unit. Normal: ON, Abnormal: blink or OFF
(3)	Expansion cover (Left side)	The cover for the expansion connector. Used when connecting to the expansion cable from the front unit.
(4)	Input terminals	The terminal block to connect input signals. The terminal screw size is M3. The terminal block supports 0.36 to 2.1 mm^2 (AWG22 to 14). If 2 crimping terminals are connected to one terminal screw, use 0.36 to 1.3 mm^2 (AWG20 to 16) cable. 
(5)	Output terminals	The terminal block to connect output signals. The wiring Specification is the same as the input terminal.
(6)	Power terminal	The terminal for connecting the power supply. The wiring Specification is the same as the input terminal.
(7)	Expansion cover (Right side)	The cover for the expansion connector. Used when connecting to the next unit.
(8)	Dummy cover	The cover as a dummy.
(9)	Terminal block cover	The cover for the terminal block.
(10)	Mounting hole	Use these holes when installing with screws. (M4×200 mm (0.79 in.))
(11)	DIN rail installation clip	This is used when mounting to a DIN rail.

2.16.2 Terminal layout and wiring

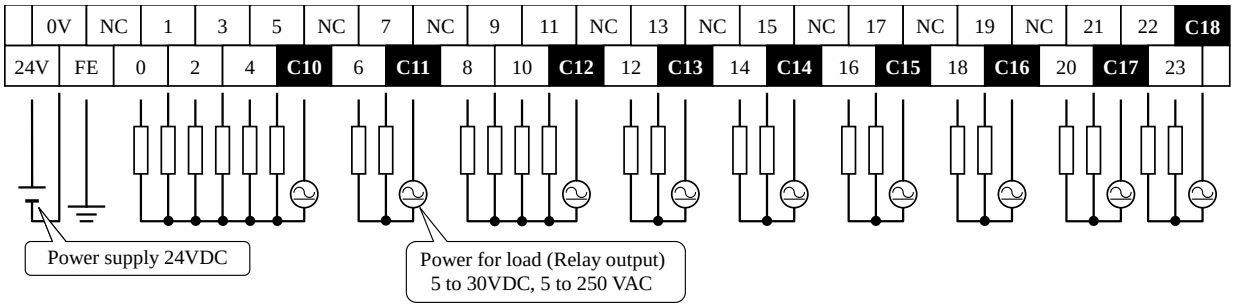
(1) 64-point type

[EH-A64EDR (AC power type)]

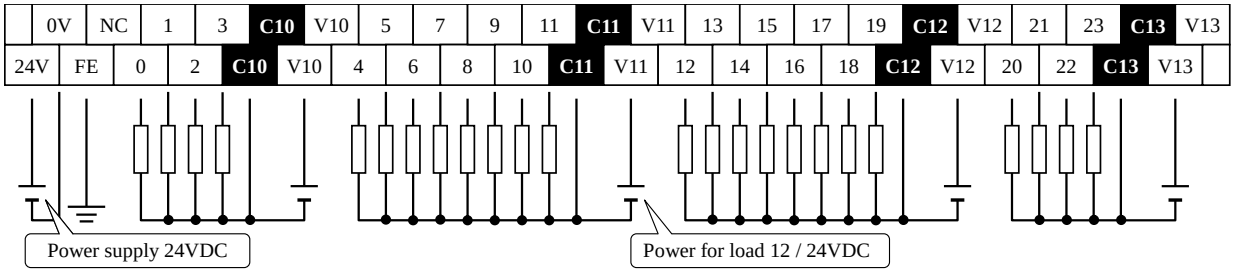
*: For the DC input, both sink and source types are available. It is possible to reverse the polarity of 24 V DC.



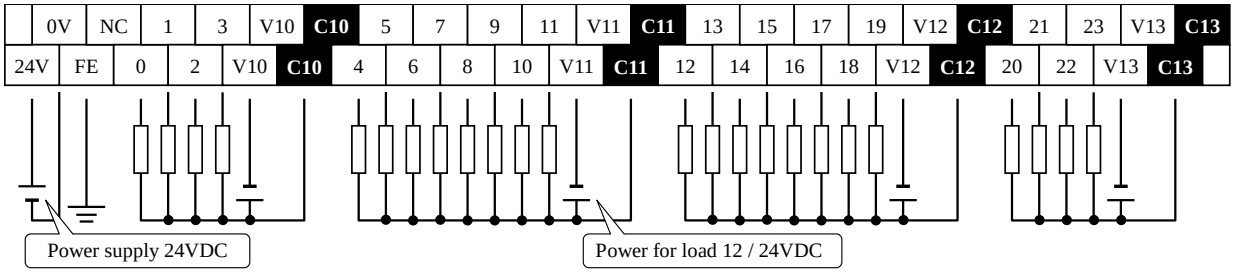
[EH-D64EDR (DC power type)] Input wiring is same as EH-A64EDR.



[EH-D64EDTPS (DC power type)] Input wiring is same as EH-A64EDR.



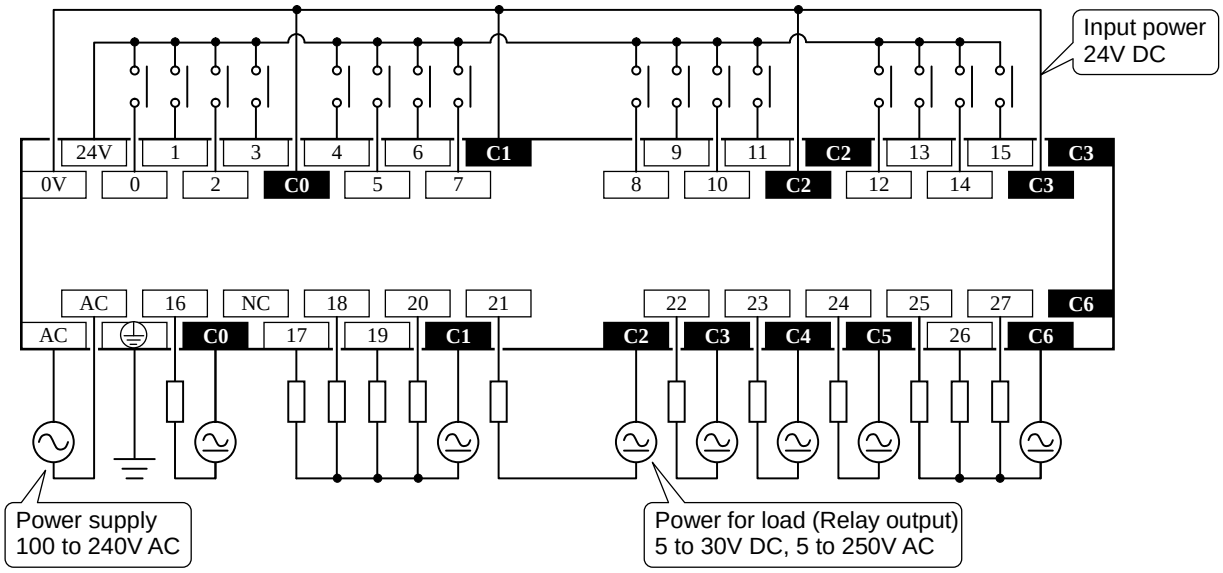
[EH-D64EDT (DC power type)] Input wiring is same as EH-A64EDR.



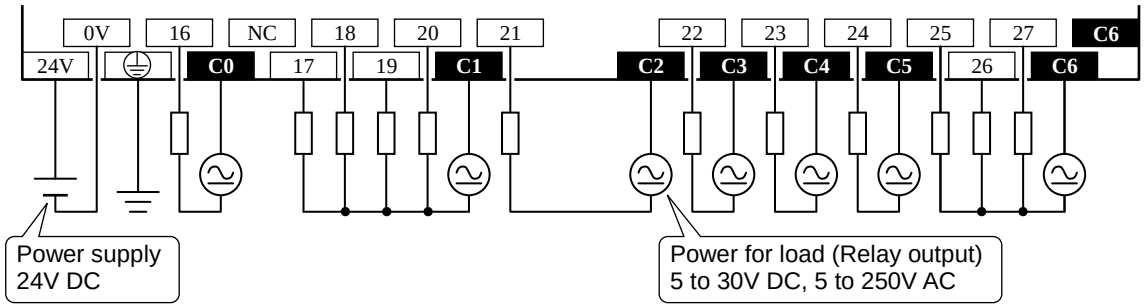
(2) 28-point type

[EH-A28EDR (AC power type)]

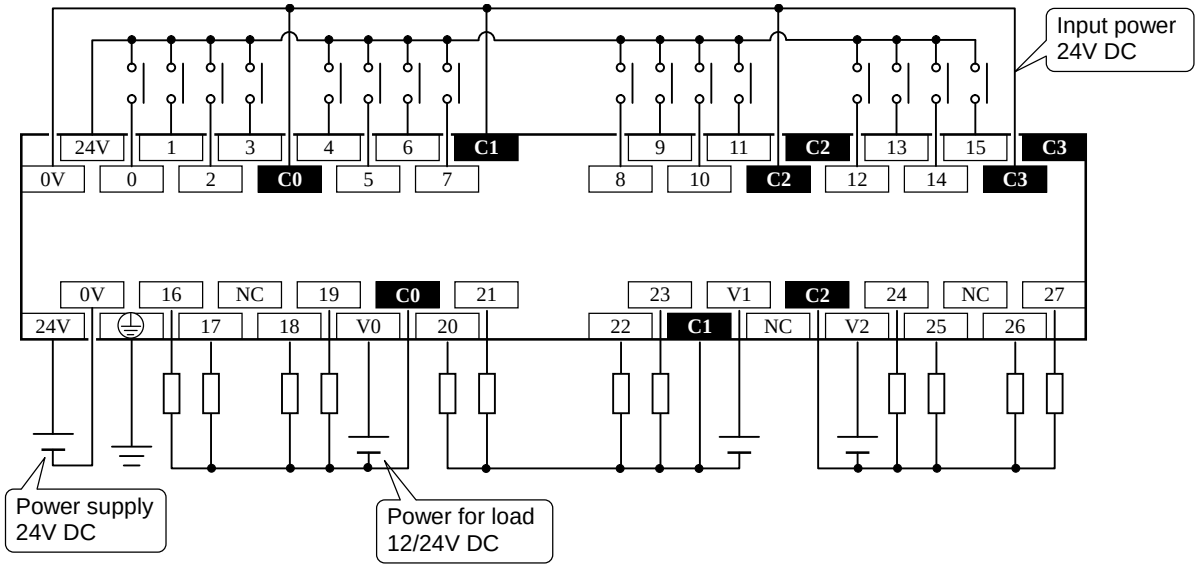
*: For the DC input, both sink and source types are available. It is possible to reverse the polarity of 24 V DC.



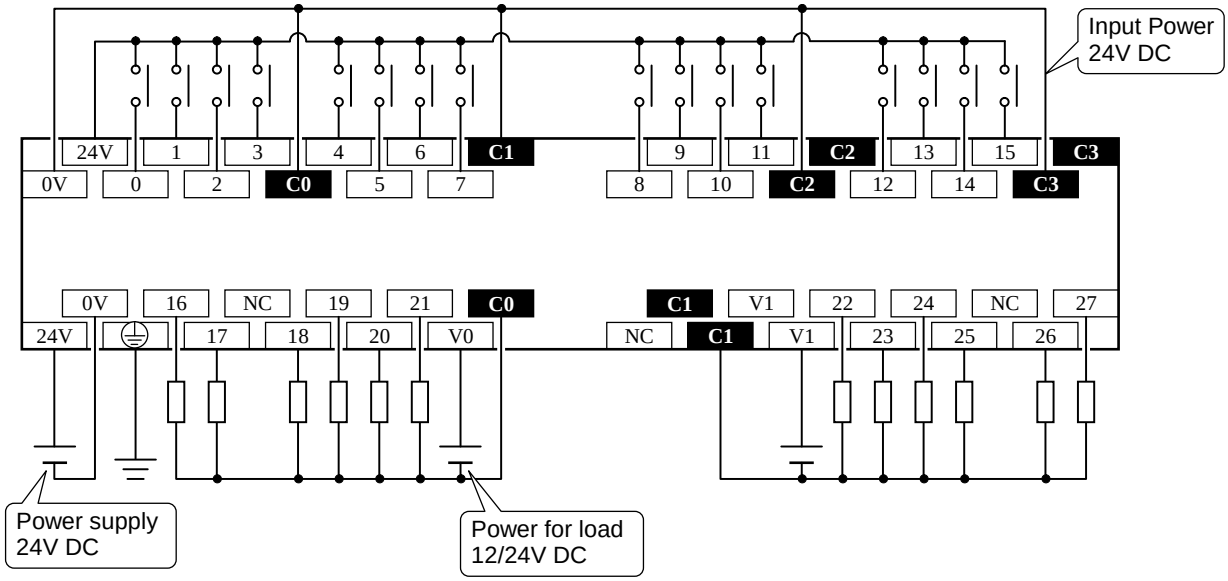
[EH-D28EDR (DC power type)] Input wiring is same as EH-A28EDR.



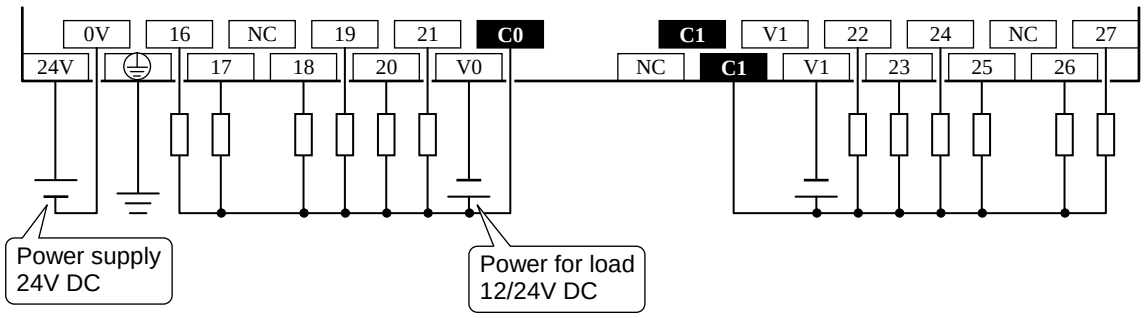
[EH-D28EDTP (DC power type)]



[EH-D28EDTPS (DC power type)]



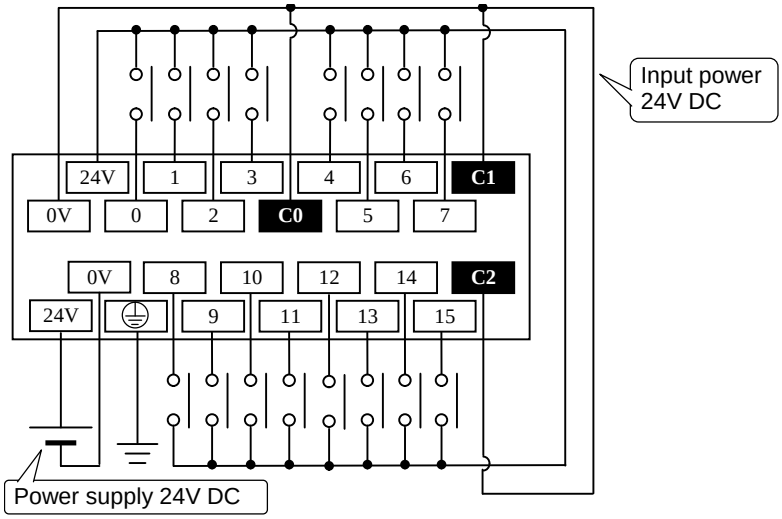
[EH-D28EDT (DC power type)] Input wiring is same as EH-D28EDTP.



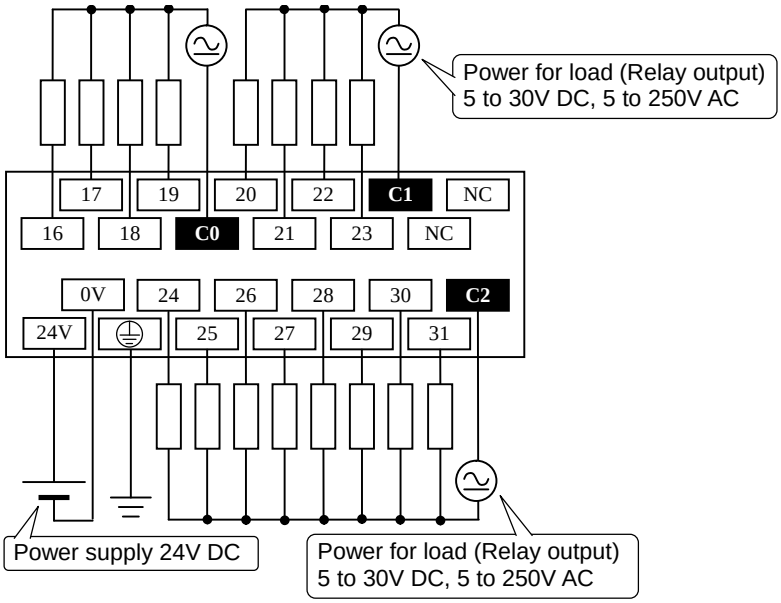
(3) 16-point type

[EH-D16ED (DC power type)]

*: For the DC input, both sink and source types are available. It is possible to reverse the polarity of 24 V DC.

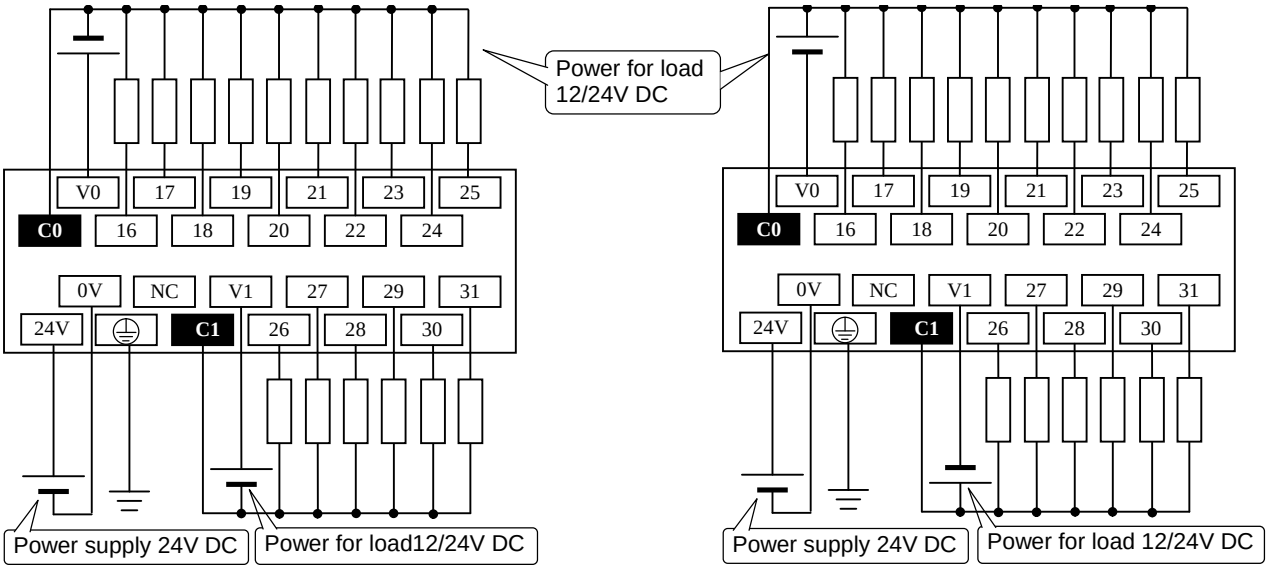


[EH-D16ER (DC power type)]



[EH-D16ETPS (DC power type)]

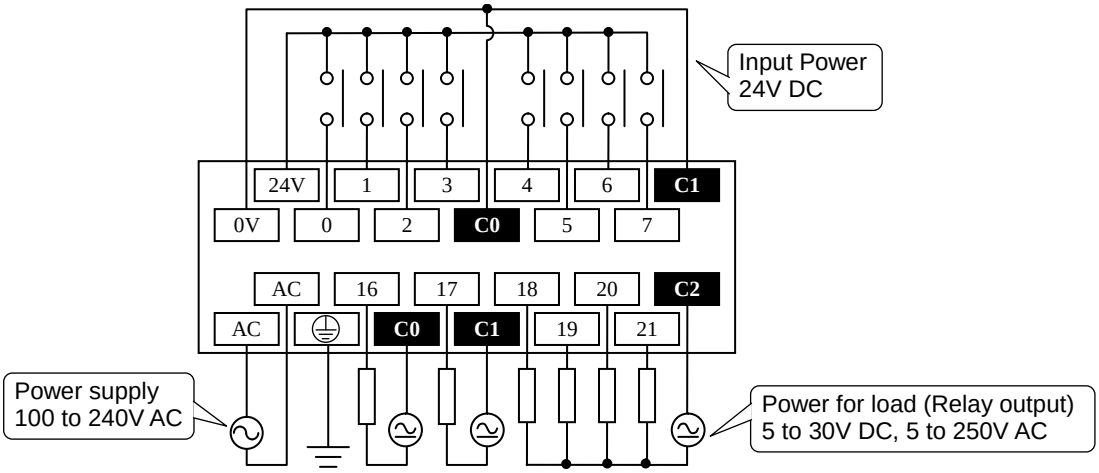
[EH-D16ET (DC power type)]



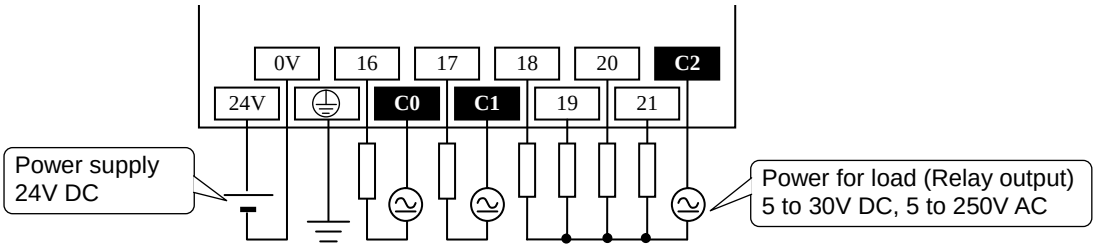
(4) 14-point type

[EH-A14EDR (AC power type)]

*: For the DC input, both sink and source types are available. It is possible to reverse the polarity of 24 V DC.

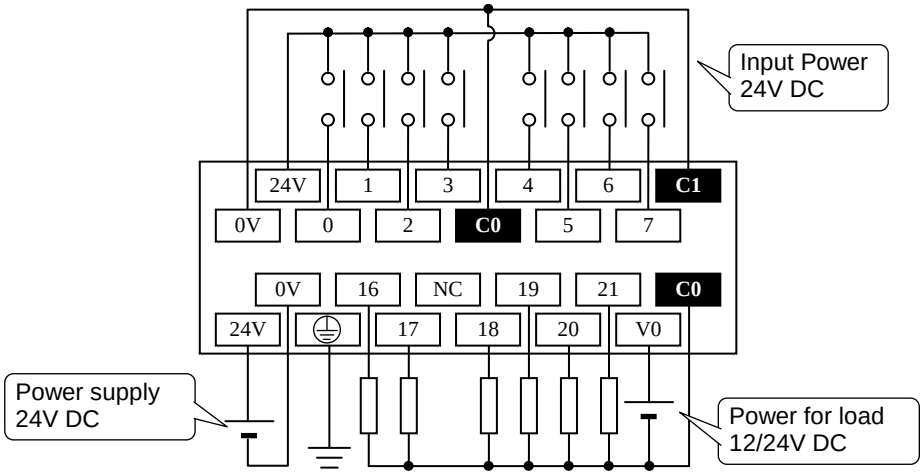


[EH-D14EDR (DC power type)] Input wiring is same as EH-A14EDR.

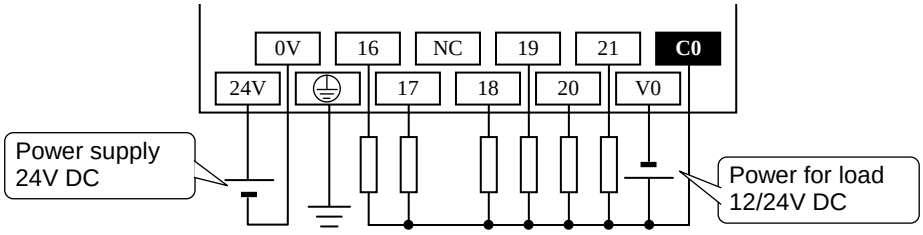


[EH-D14EDTP, EH-D14EDTPS (DC power type)]

*: For the DC input, both sink and source types are available. It is possible to reverse the polarity of 24 V DC.



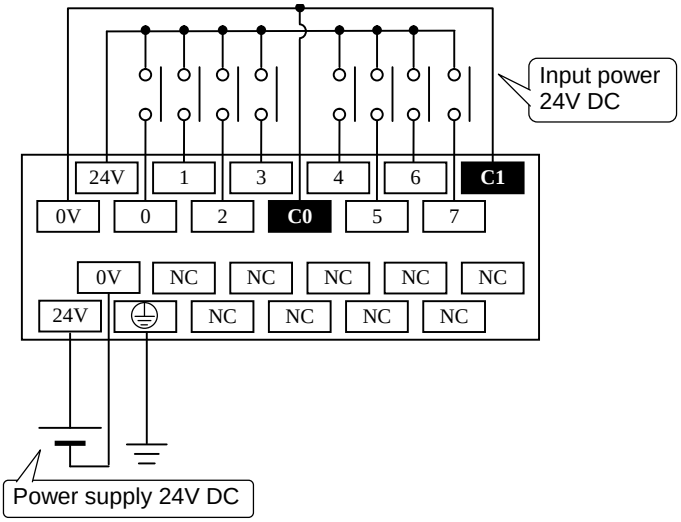
[EH-D14EDT (DC power type)] Input wiring is same as EH-D14EDTP.



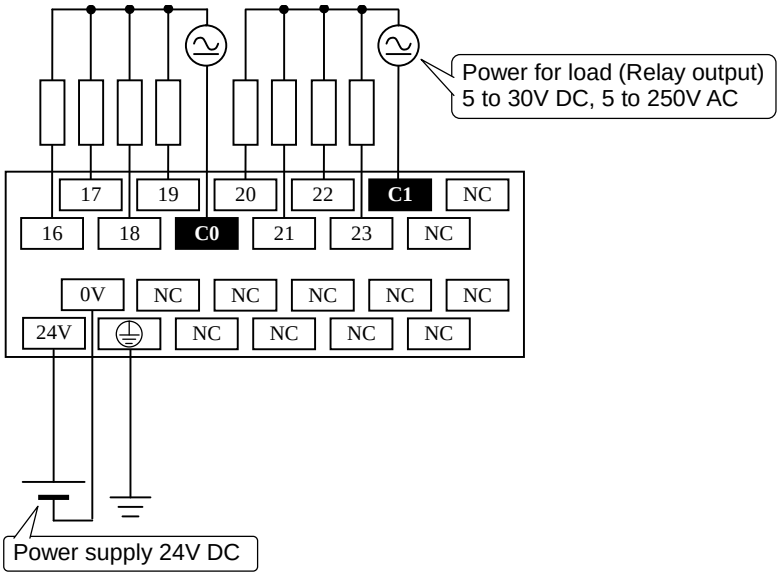
(5) 8-point type

[EH-D8ED (DC power type)]

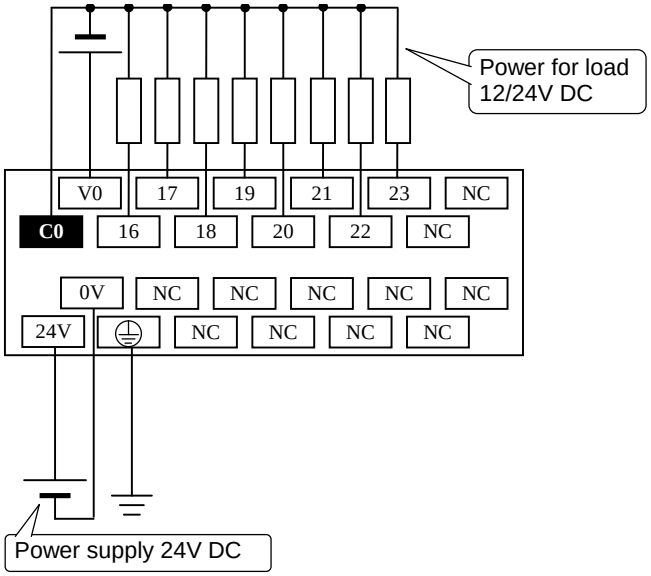
*: For the DC input, both sink and source types are available. It is possible to reverse the polarity of 24 V DC.



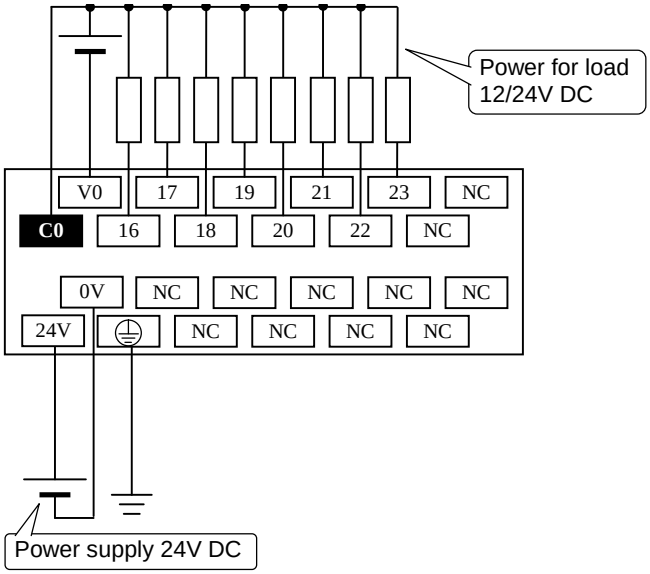
[EH-D8ER (DC power type)]



[EH-D8ETPS (DC power type)]

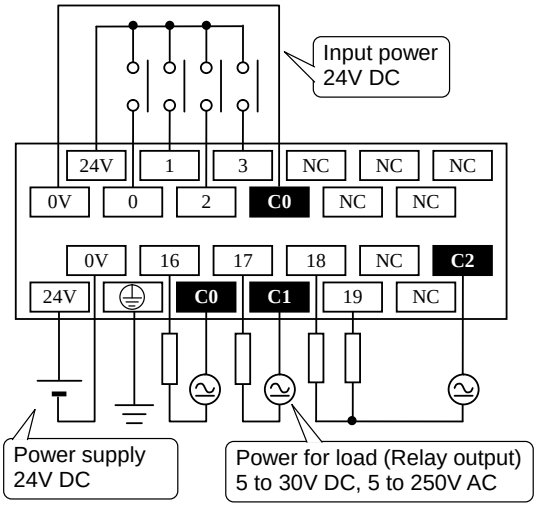


[EH-D8ET (DC power type)]

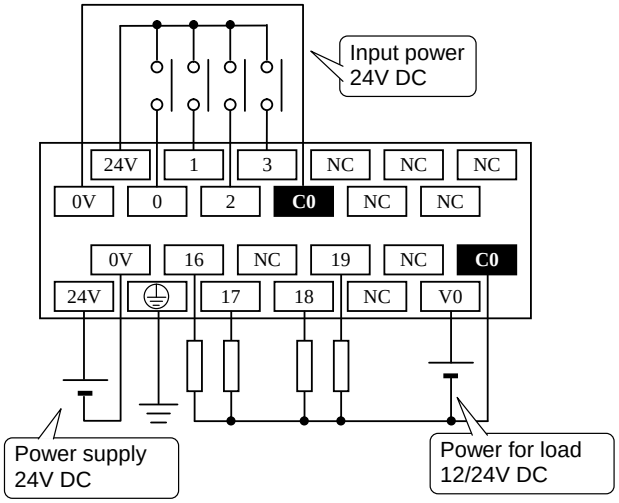


[EH-D8EDR (DC power type)]

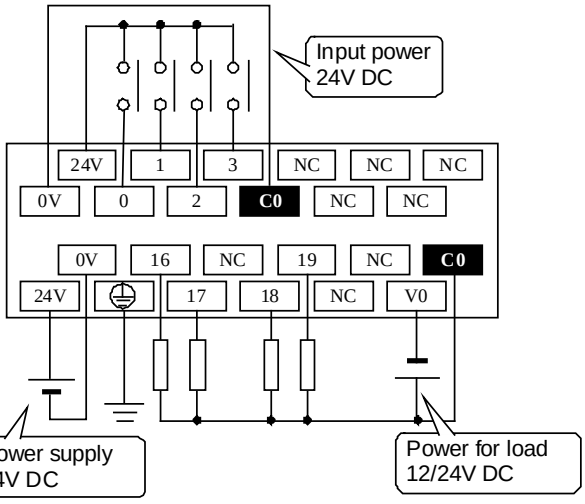
*: For the DC input, both sink and source types are available. It is possible to reverse the polarity of 24 V DC.



[EH-D8EDTPS (DC power type)]



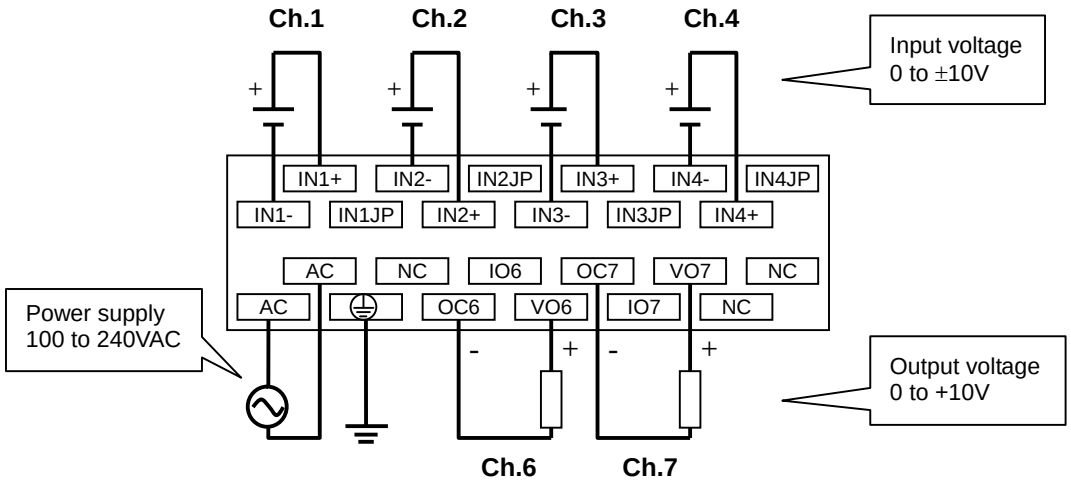
[EH-D8EDT (DC power type)]



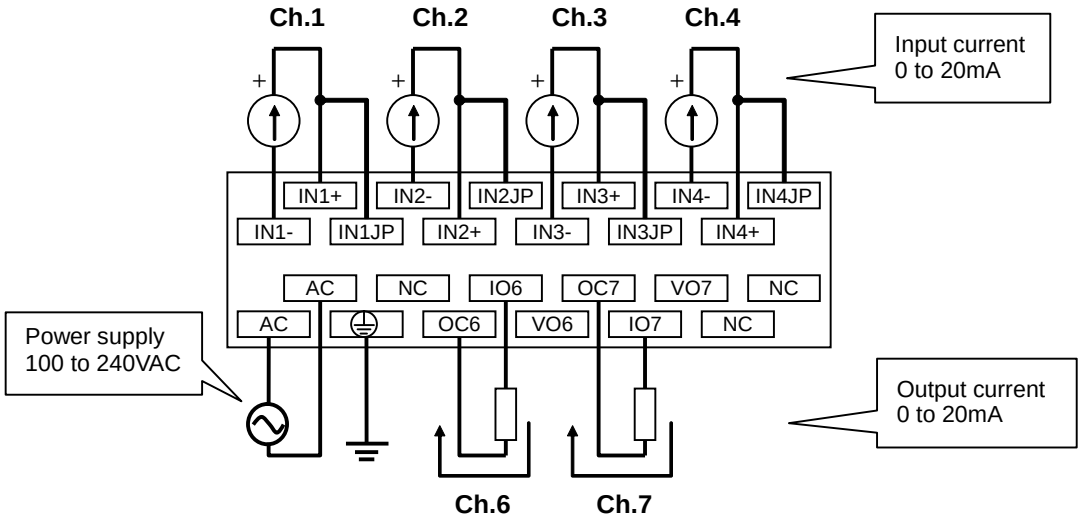
(6) Analog expansion unit

[EH-A6EAN (AC power type)]

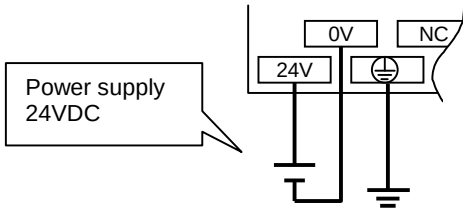
Voltage input and output (Input and output are configured separately.)



Current input and output (Input and output are configured separately.)



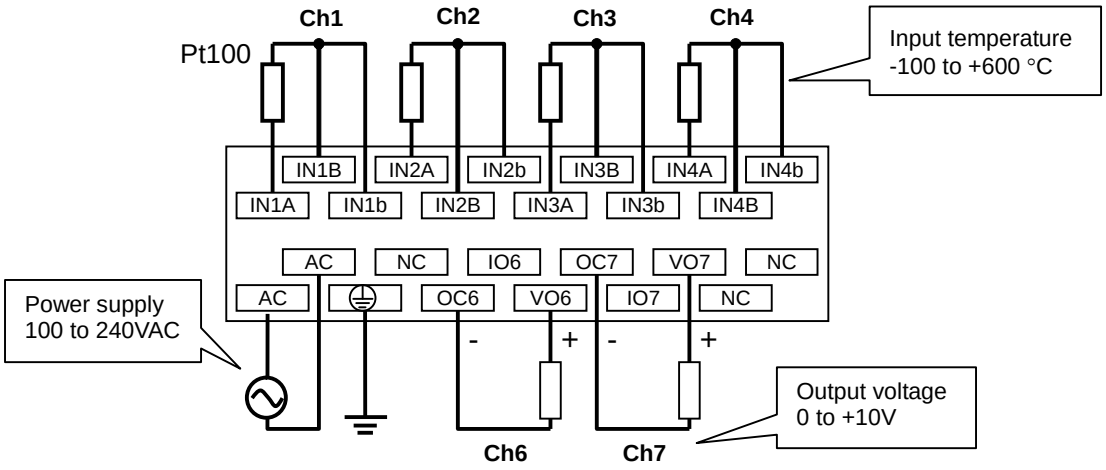
[EH-D6EAN (DC power type)] Input and output wirings are same as EH-A6EAN.



(7) RTD expansion unit

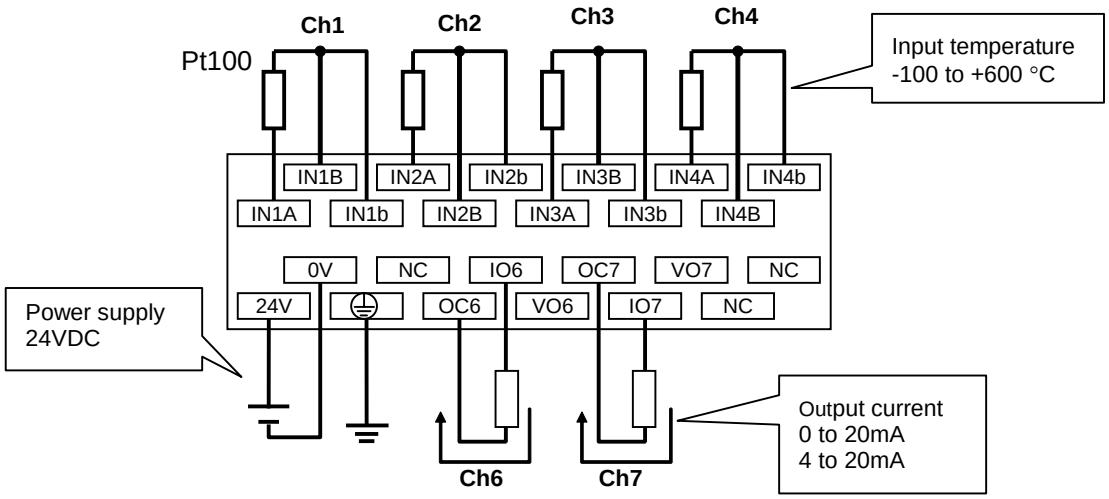
[EH-A6ERTD (AC power type)]

Example of RTD input and Voltage output

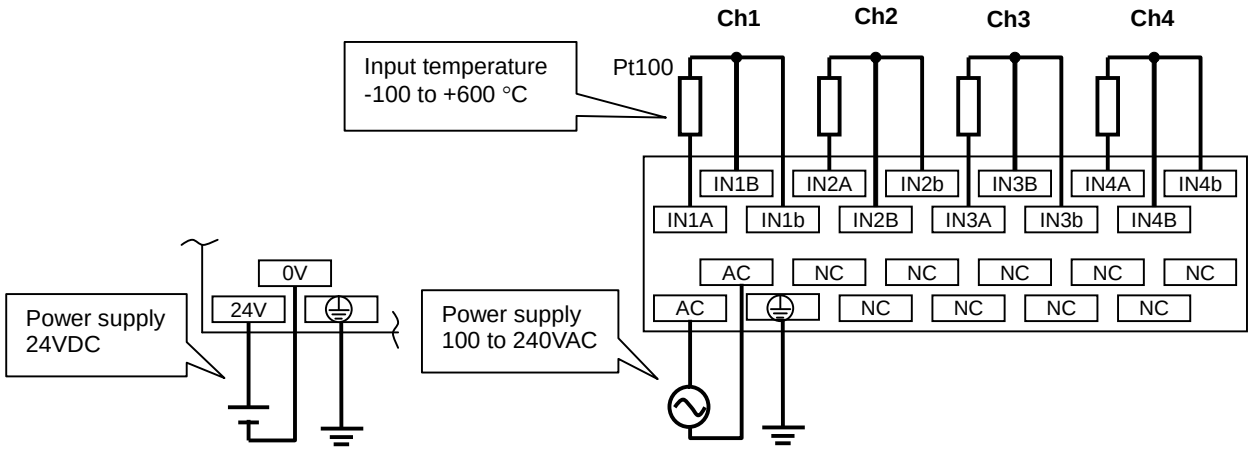


[EH-D6ERTD (DC power type)]

Example of RTD input and Current output



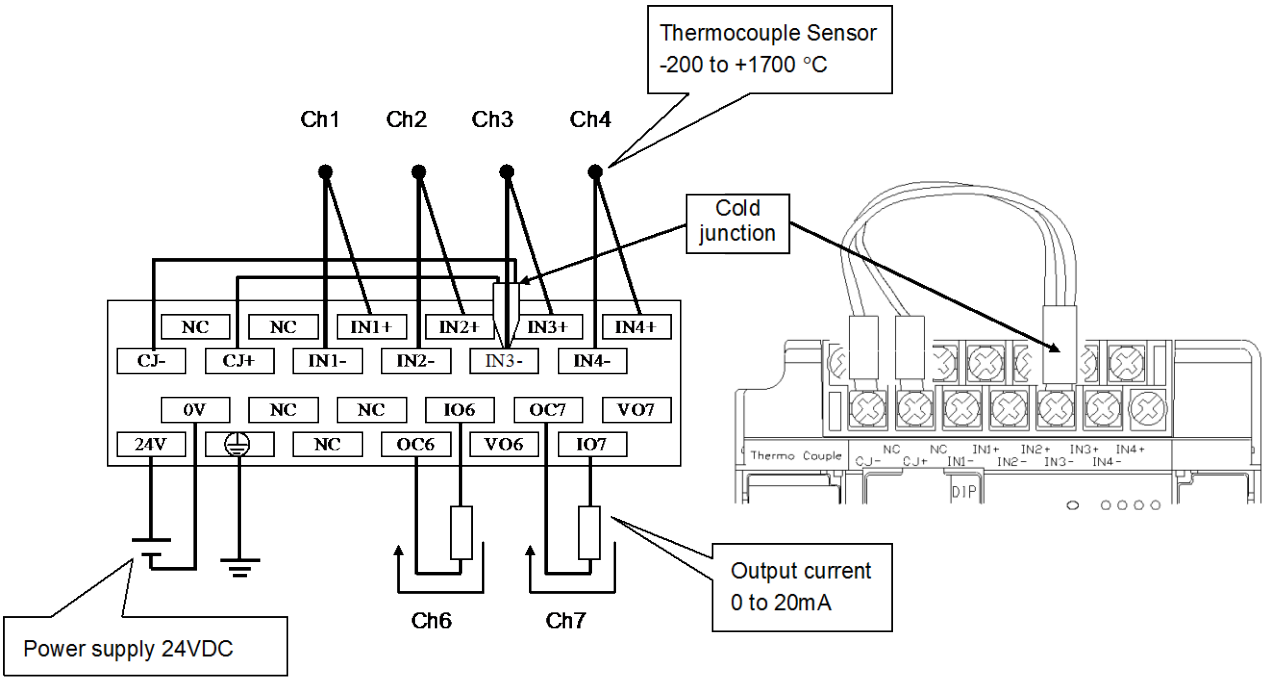
[EH-A4ERTD (AC power type), EH-D4ERTD (DC power type)]



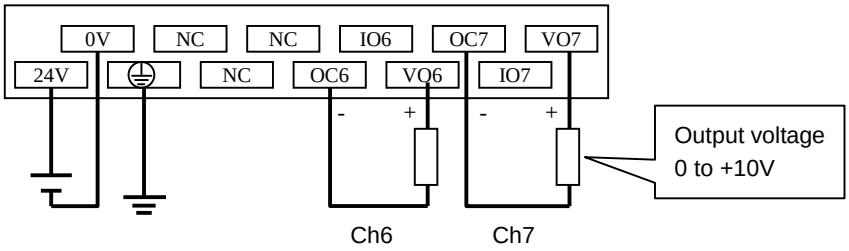
(8) Thermocouple expansion unit

[EH-D6ETC (DC power type)]

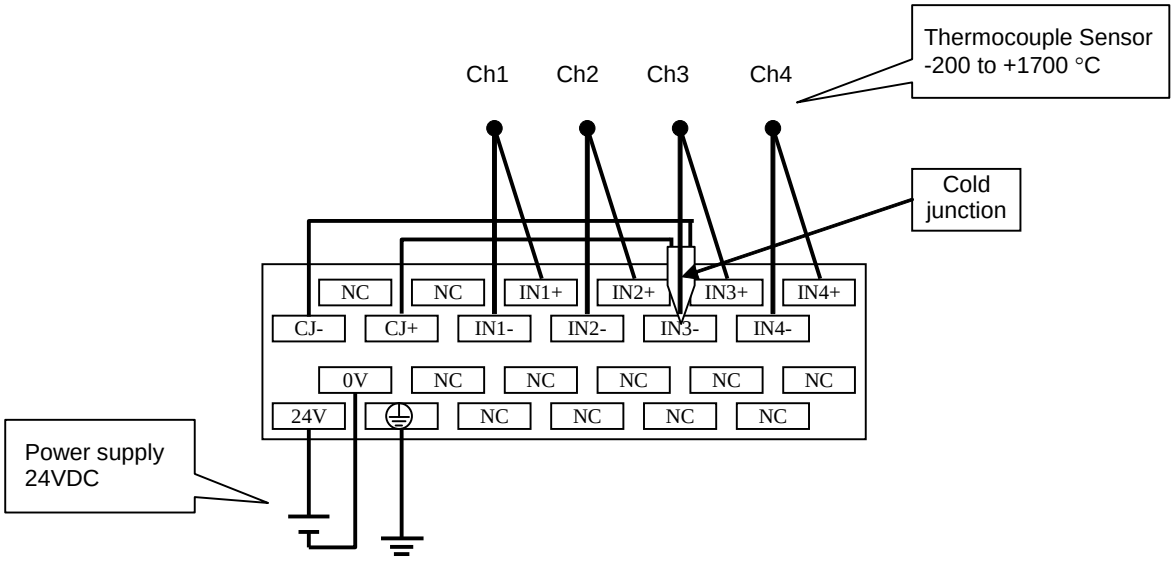
Current output



Voltage output



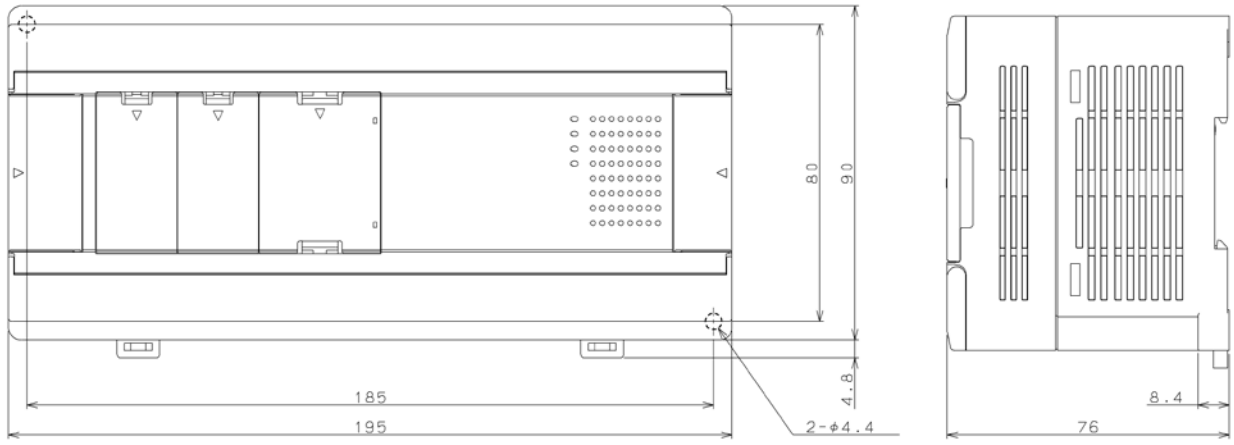
[EH-D4ETC (DC power type)]



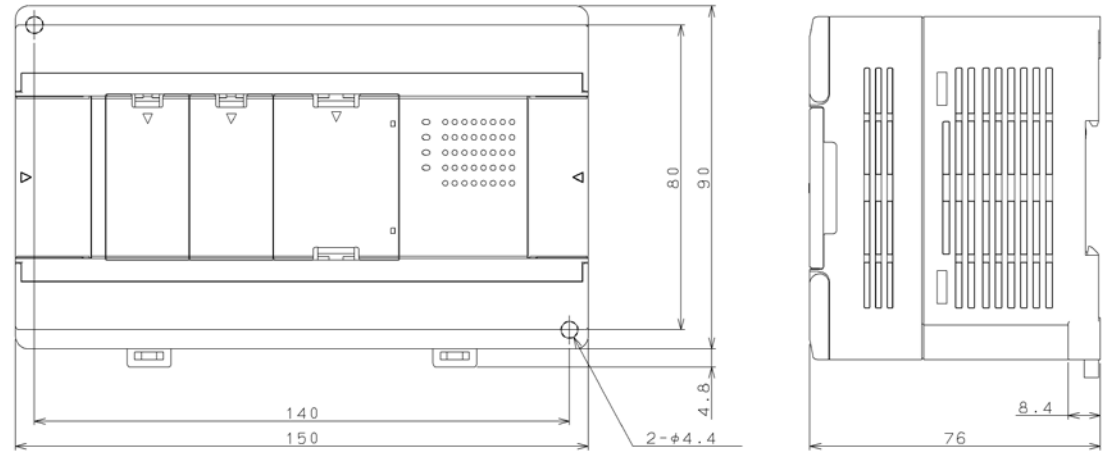
2.17 External dimensions

[64-point basic / expansion unit]

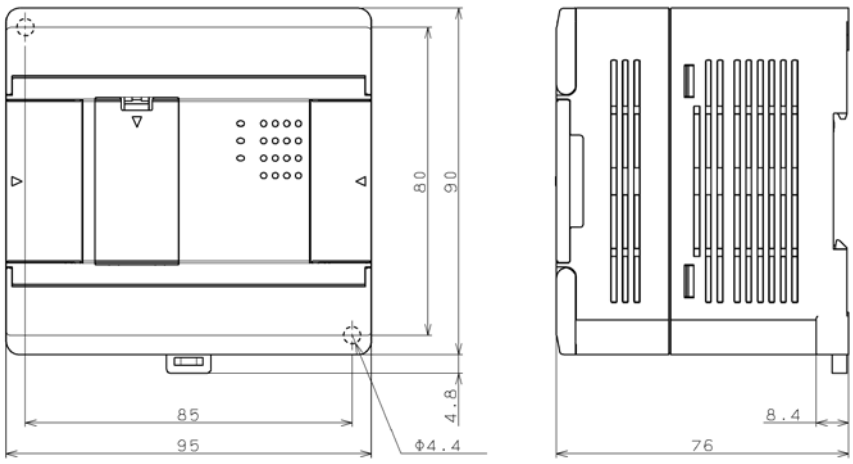
Unit: mm



[40-point basic unit and 28-point expansion unit]



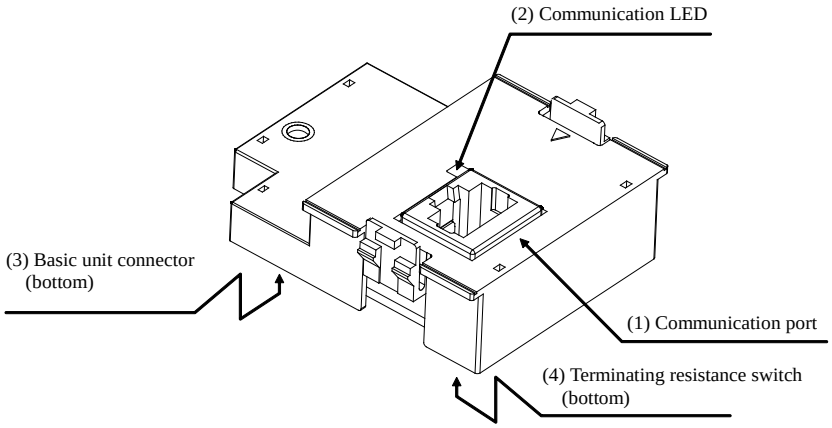
[16-point / 14-point / 8-point and Analog / RTD / Thermocouple expansion unit]



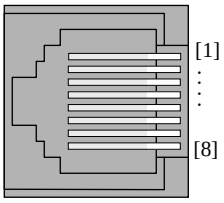
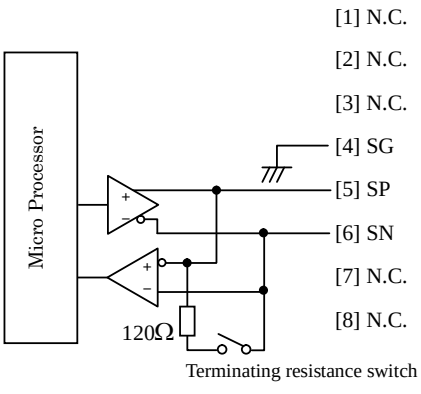
2.18 Option board

2.18.1 OBV-NES

Name and function of each part	Type	OBV-NES
	Weight	15g (0.03 lb.)

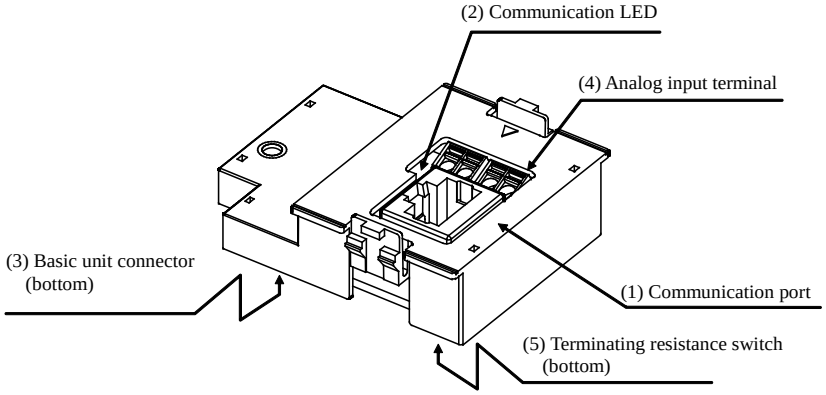


No.	Item	Description
(1)	Communication port	Communication port supports IEC programming function supporting Modbus-RTU master/slave and general purpose communication. *: IEC programming function is fixed to RS-485.
(2)	Communication LED	Flashes data transmission / reception.
(3)	Basic unit connector	This is a connector to basic unit.
(4)	Terminating resistance switch	This is a switch to select to enable / disable the 120Ω built-in termination resistor. ON: Enabled OFF: Disabled

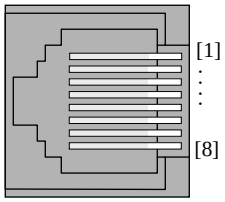
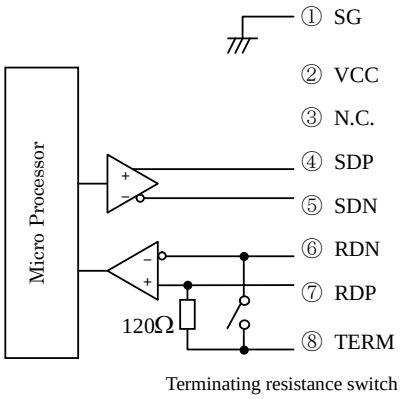
Terminal layout	Pin No.	Signal	Meaning	Internal circuit
 Socket connector (Top view)	[1]	N.C	Not used	
	[2]	N.C	Not used	
	[3]	N.C	Not used	
	[4]	SG	Signal ground	
	[5]	SP	Send / Receive data +	
	[6]	SN	Send / Receive data -	
	[7]	N.C	Not used	
	[8]	N.C	Not used	

2.18.2 OBV-485A

Name and function of each part	Type	OBV-485A
	Weight	20g



No.	Item	Description
(1)	Communication port	Communication port supports IEC programming function supporting Modbus-RTU master/slave and general purpose communication. *: IEC programming function is fixed to RS-485.
(2)	Communication LED	Flashes data transmission / reception.
(3)	Basic unit connector	This is a connector to basic unit.
(4)	Analog input terminal	Input terminal for analog voltage sigral Cable diameter: Single wire: 0.14 mm^2 to 1.5 mm^2 Strand wire: 0.14 mm^2 to 1.0 mm^2
(5)	Terminating resistance switch	This is a switch to select to enable / disable the 120Ω built-in termination resistor. ON: Enabled OFF: Disabled

Terminal layout	Pin No.	Signal	Meaning	Internal circuit
 Socket connector (Top view)	[1]	SG	Signal ground	
	[2]	VCC	5V DC output	
	[3]	N.C	Not used	
	[4]	SDP	Send data +	
	[5]	SDN	Send data -	
	[6]	RDN	Receive data -	
	[7]	RDP	Receive data +	
	[8]	TERM	Not used	

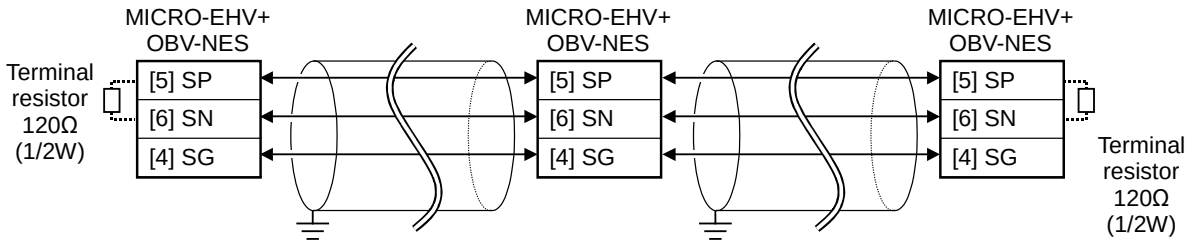
[Analog input specifications]

Item		Specification
Number of input		2 ch.
Input range		0 to 10V (10.24V max.)
Accuracy		±1% (For the full-scale value)
Resolution		10-bit
Maximum external wiring length		Less than 3m
Input impedance		Approx. 100 kΩ
Isolation	Between CPU and analog signal	Not isolated
	Between channels	Not isolated

2.18.3 Communication cable connection

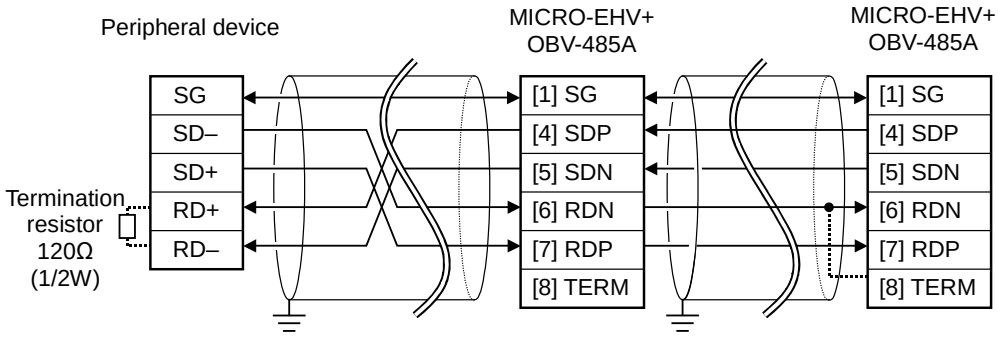
The following figure shows the communication cable connection example. Be sure to use shielded twisted pair cable.

[OBV-NES]

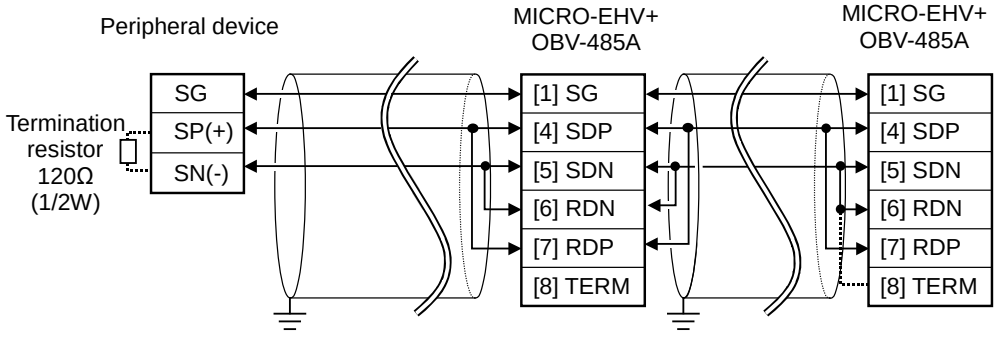


[OBV-485A]

The example of the cable connection of RS-422 I/F is shown below.



The example of the cable connection of RS-485 I/F is shown below.



Use the built-in termination resistors (120Ω) depending on the usage environment and the cable type. If the communication is unstable, perform the followings.

(1) Instead of the built-in termination resistor, attach termination resistors that match the characteristic impedance of the cable at both ends of the communication cable.

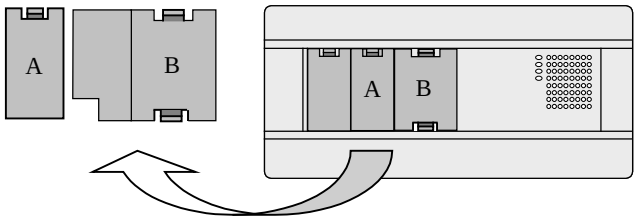
(2) Make wiring unconnected the SG (signal ground) of each device.

(3) Lower the transfer rate.

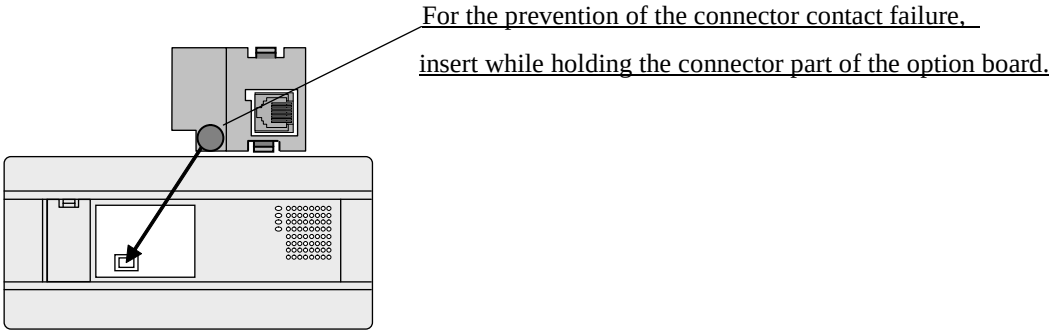
(4) Attach the ferrite core to the communication cable in a noisy environment.

2.18.4 Installation of option board

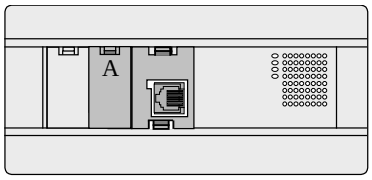
Remove the two covers (A and B) from the basic unit.



(1) Plug the connector of option board to the connector of the basic unit.



(2) After connecting, attach the cover A.

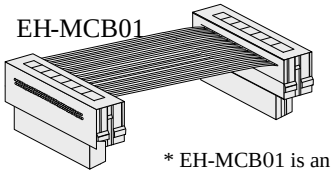


2.19 Accessories

2.19.1 Expansion cable

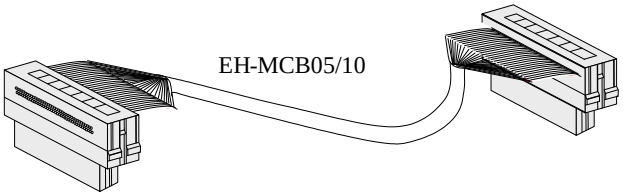
Features		Type	EH-MCB01 / 05 / 10
		Weight	Approx. 210 (0.46) / 240 (0.53)/ 300 g (0.66 lb.)
		Length	0.1 (0.33) / 0.5 (1.64)/ 1.0m (3.28 ft.)

EH-MCB01



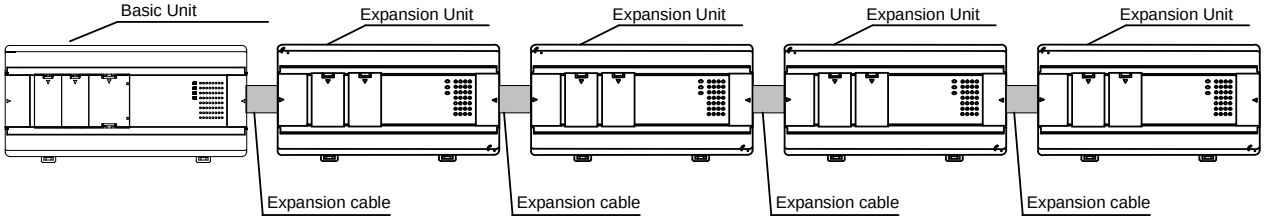
* EH-MCB01 is an accessory of expansion units.

EH-MCB05/10



Function	Connects the basic unit and the expansion unit or the expansion unit to each other. There is no directivity in the cable.

Connect the right side to the left side of each unit. PLC does not work correctly in improper connection.



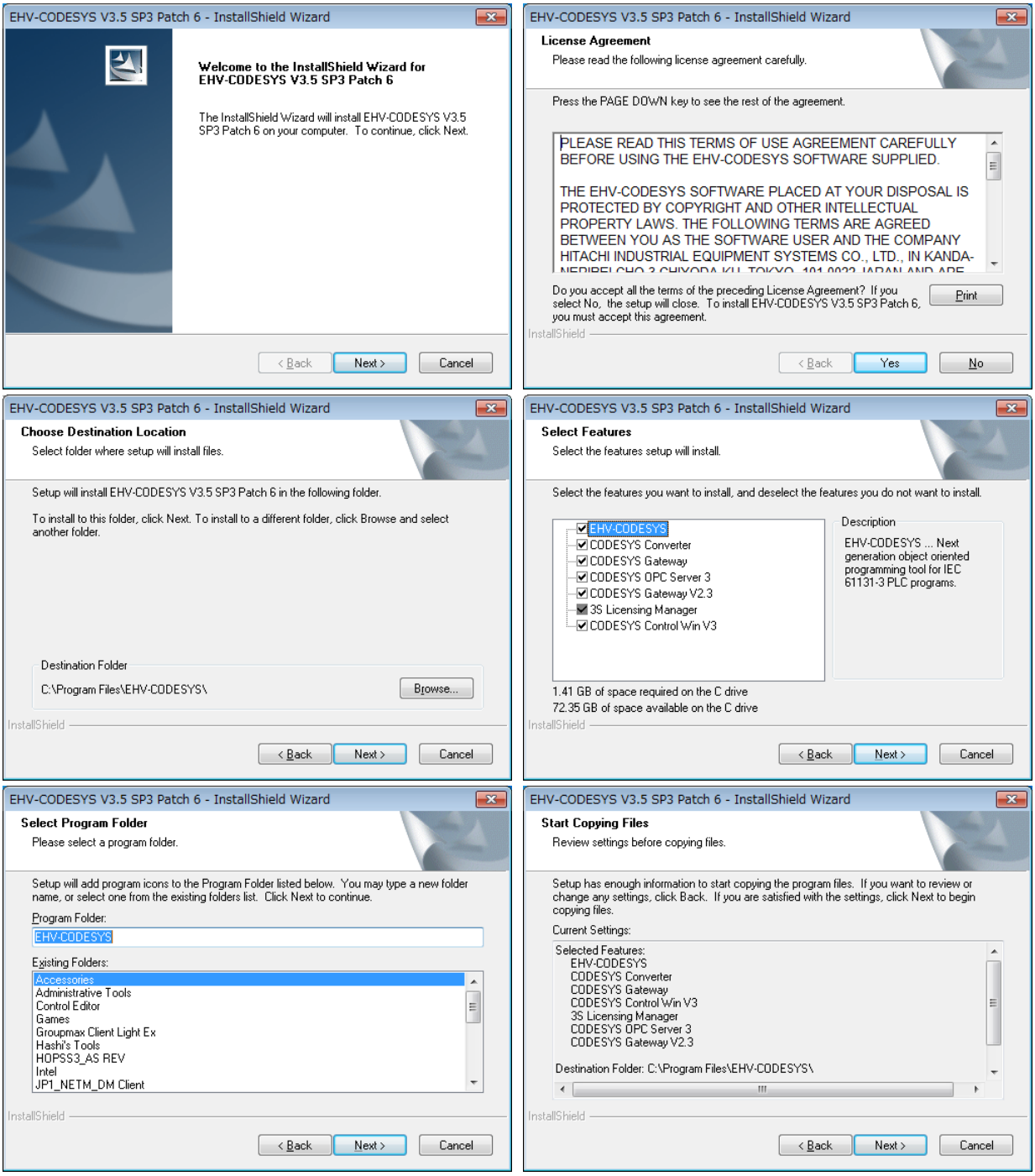
Chapter 3 Programming

3.1 Installation

3.1.1 Installation of EHV-CODESYS

1. The installation wizard starts up automatically on EHV-CODESYS installation CD.

2. Follow the instructions.



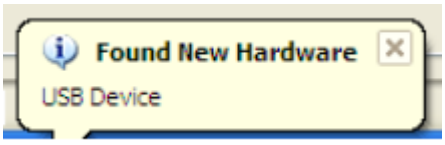
Note

Several Microsoft components are necessary to be installed for EHV-CODESYS. If components are not installed in your PC, the installation of EHV-CODESYS stops and a dialog appears. Click [Install] at the dialog to extract from setup file.

3.1.2 Installation of USB driver

1. Plug in USB cable to CPU module.

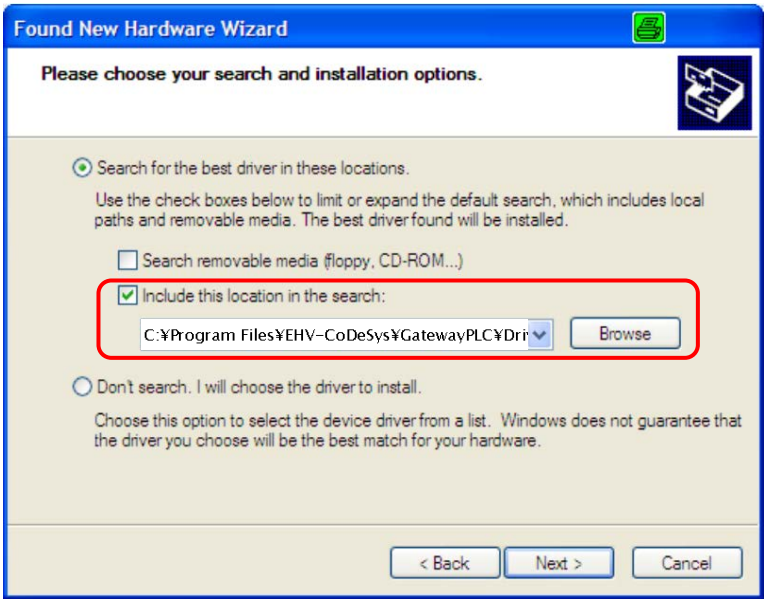
2. Popup window appears at right-bottom of screen. Click the popup window.



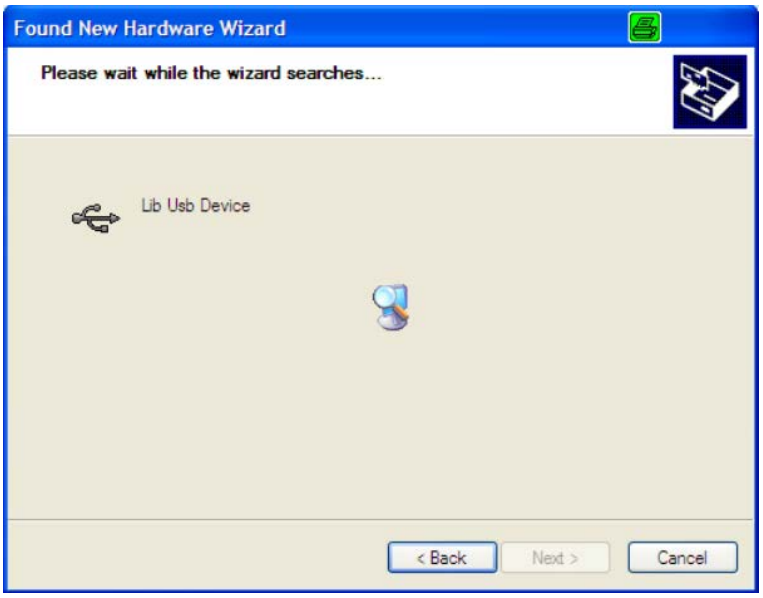
3. Click “Install from a list or specific location (Advanced)” and “Next” button.

4. Click “Include this location in the search” with the path C:¥Program Files¥EHV-CODESYS¥GatewayPLC¥Driver and “Next” button. USB driver for Windows 7 64-bit is in the separated folder as below.

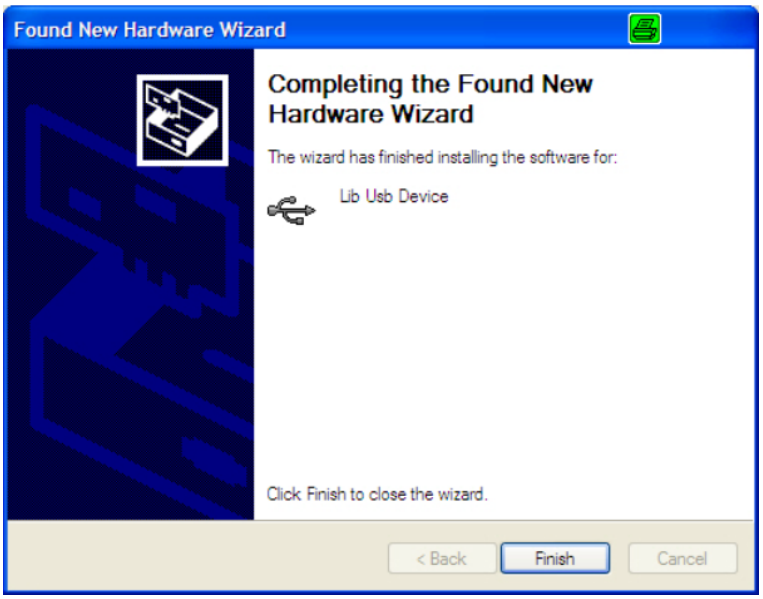
x64 ... Windows 7 64-bit
x86 ... Windows 2000, Windows XP, Windows Vista, Windows 7 32-bit



5. USB driver installation is in progress.



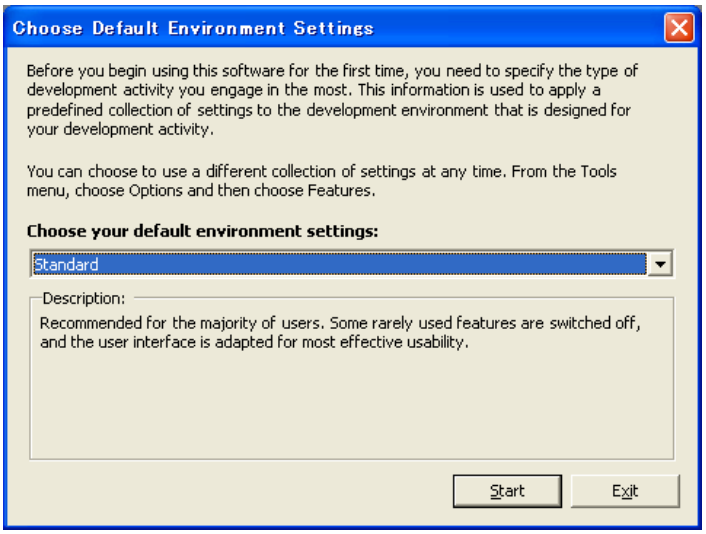
6. USB driver installation has been completed. Click “Finish” to close the wizard.



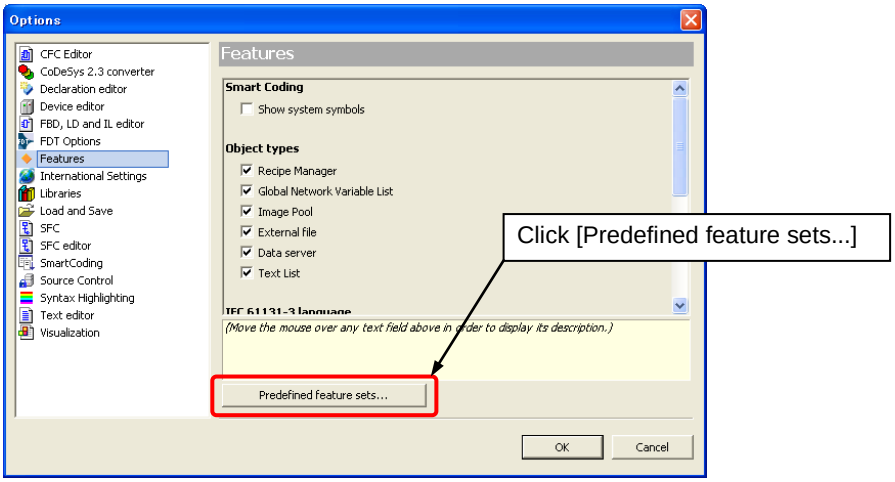
7. Reboot the PC to activate the USB driver.

3.2 Startup

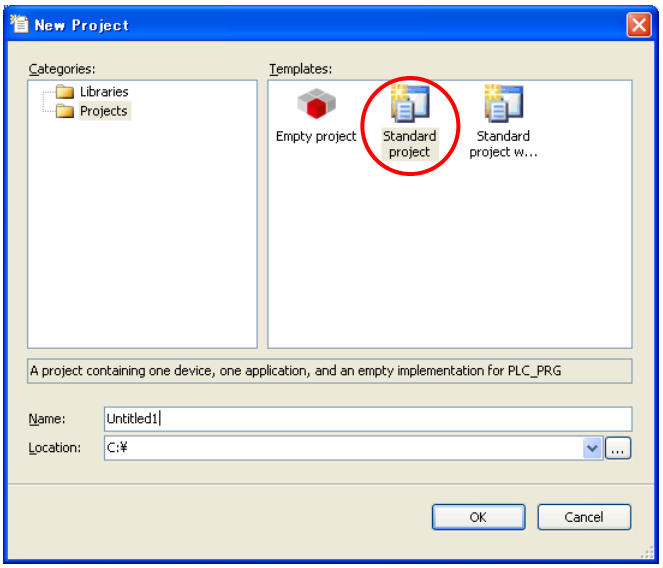
In the first use, you need to specify the type of development activity “Standard” or “Professional” you engage in the most.



Although the above dialog appears at the first use only, this setting can be changed later in the menu [Tools]-[Options]-[Features] as below.

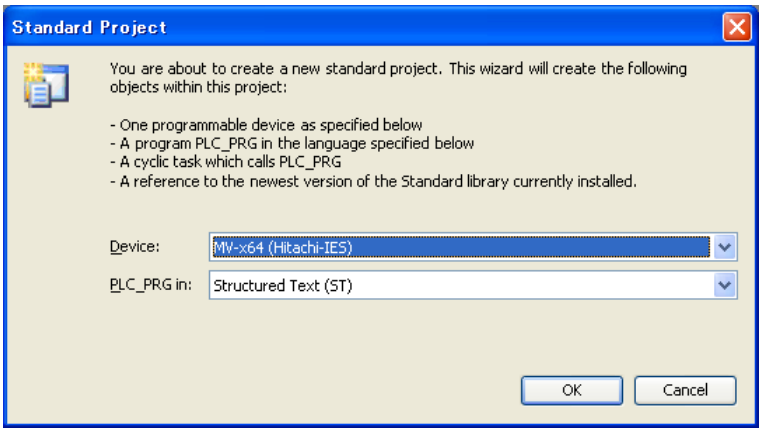


Click  icon or choose [File]-[New Project...] to create a new project file. Then New Project dialog box appears as below. Choose “Standard project”, enter new file name, specify location and click [OK].

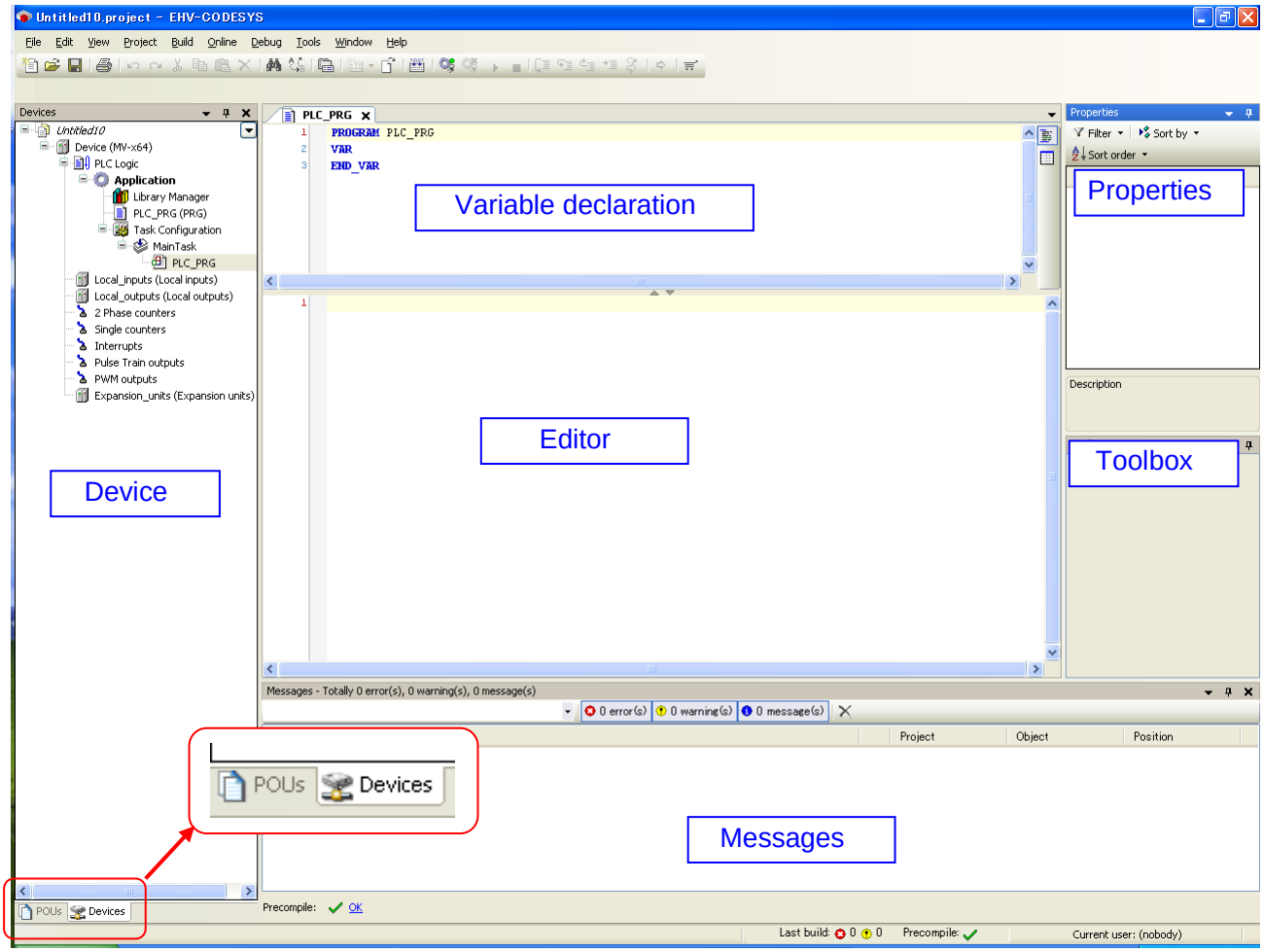


Choose CPU type and programming language and click [OK].
Available languages are as follows.

- Continuous Function Chart (CFC)
- Function Block Diagram (FBD)
- Instruction List (IL)
- Ladder Logic Diagram (LD)
- Sequential Function Chart (SFC)
- Structured Text (ST)



Initial screen shot of EHV-CODESYS is shown below.



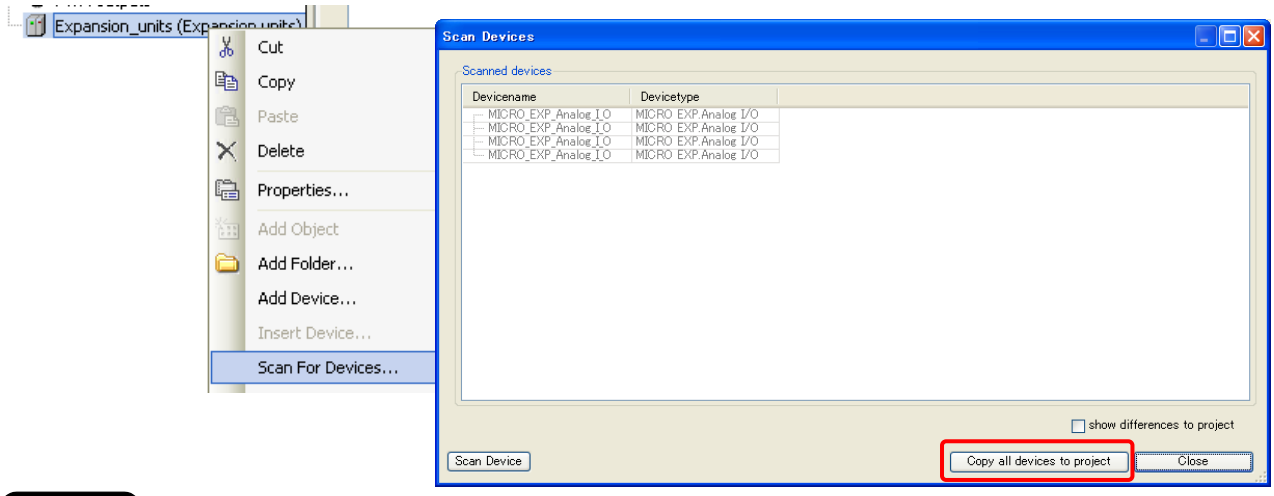
In the default setting, Device tree is behind the POU window. Click Devices tab to show it. “Toolbox” and “Properties” windows can be shown by [View] menu.

3.3 I/O Configuration

3.3.1 Scan For Devices

Actual I/O configuration can be read out from connected CPU.

Right click on “Expansion units” and choose “Scan For Devices...”. Then “Scan Devices” dialog appears. Click “Copy all devices to project”.

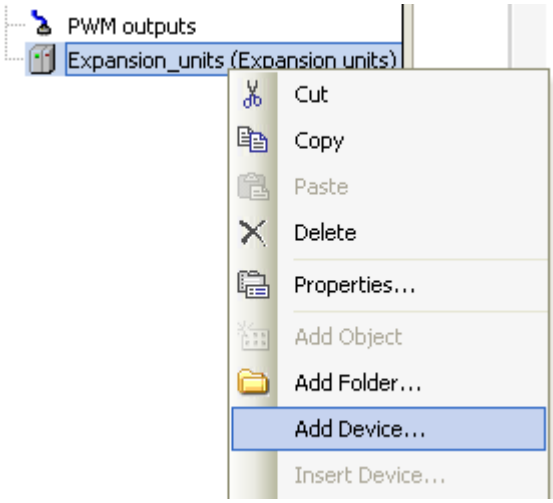


Note

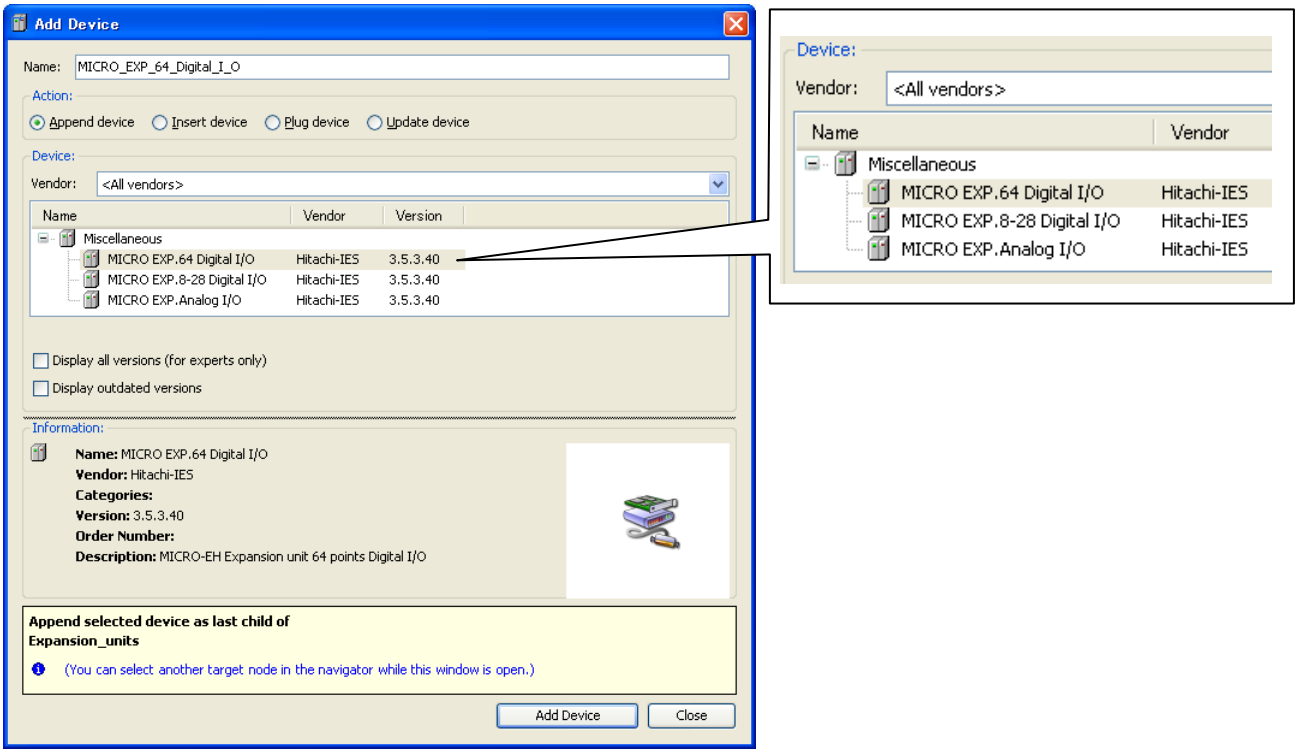
Be sure to perform “Scan For Devices” after login and logout. “Scan For Devices” works only when logout however, gateway and active path must be set and opened once in advance.

3.3.2 Expansion unit

Instead of “Scan For Devices”, expansion units can be plugged in one by one. Choose “Add Device” to configure expansion units.



MICRO-EHV+ allows to expand up to 4 expansion units.

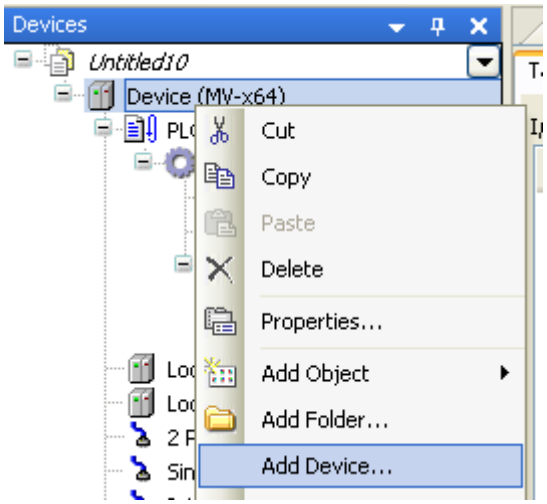


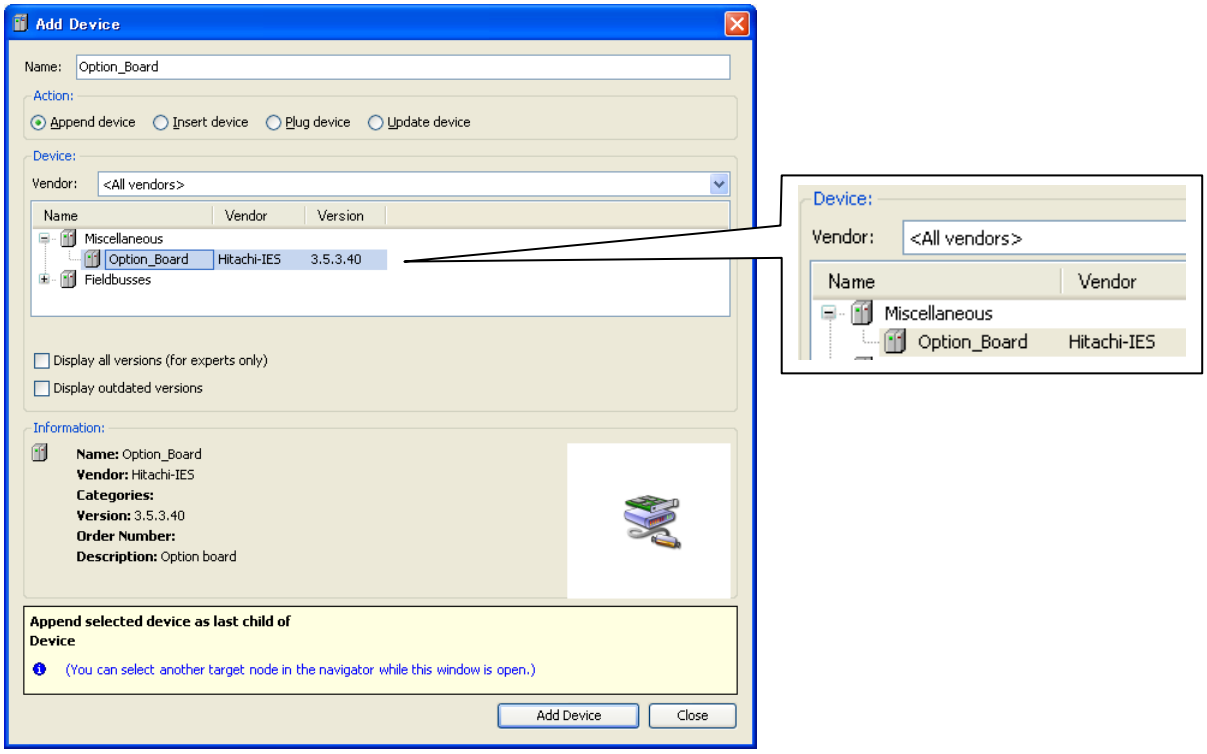
Configure expansion unit according to the list below.

Model names	Device Names
EH-A64EDR	MICRO EXP. 64 Digital I/O
EH-D64E**	
EH-A28EDR	MICRO EXP. 8-28 Digital I/O
EH-D28E**	
EH-D16E**	
EH-A14EDR	
EH-D14E**	
EH-D8E**	
EH-A6EAN	MICRO EXP. Analog I/O
EH-D6EAN	
EH-A6ERTD	
EH-D6ERTD	
EH-A4ERTD	
EH-D4ERTD	
EH-D6ETC	
EH-D4ETC	
EH-A2EP	
EH-D2EP	

3.3.3 Option board

Choose “Add Device” to configure option board.



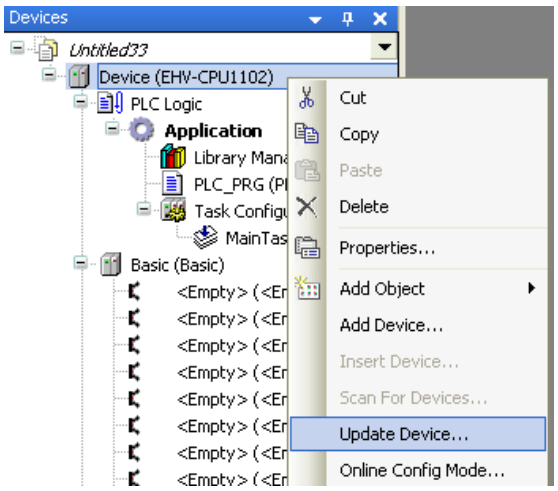


Configure option board according to the list below.

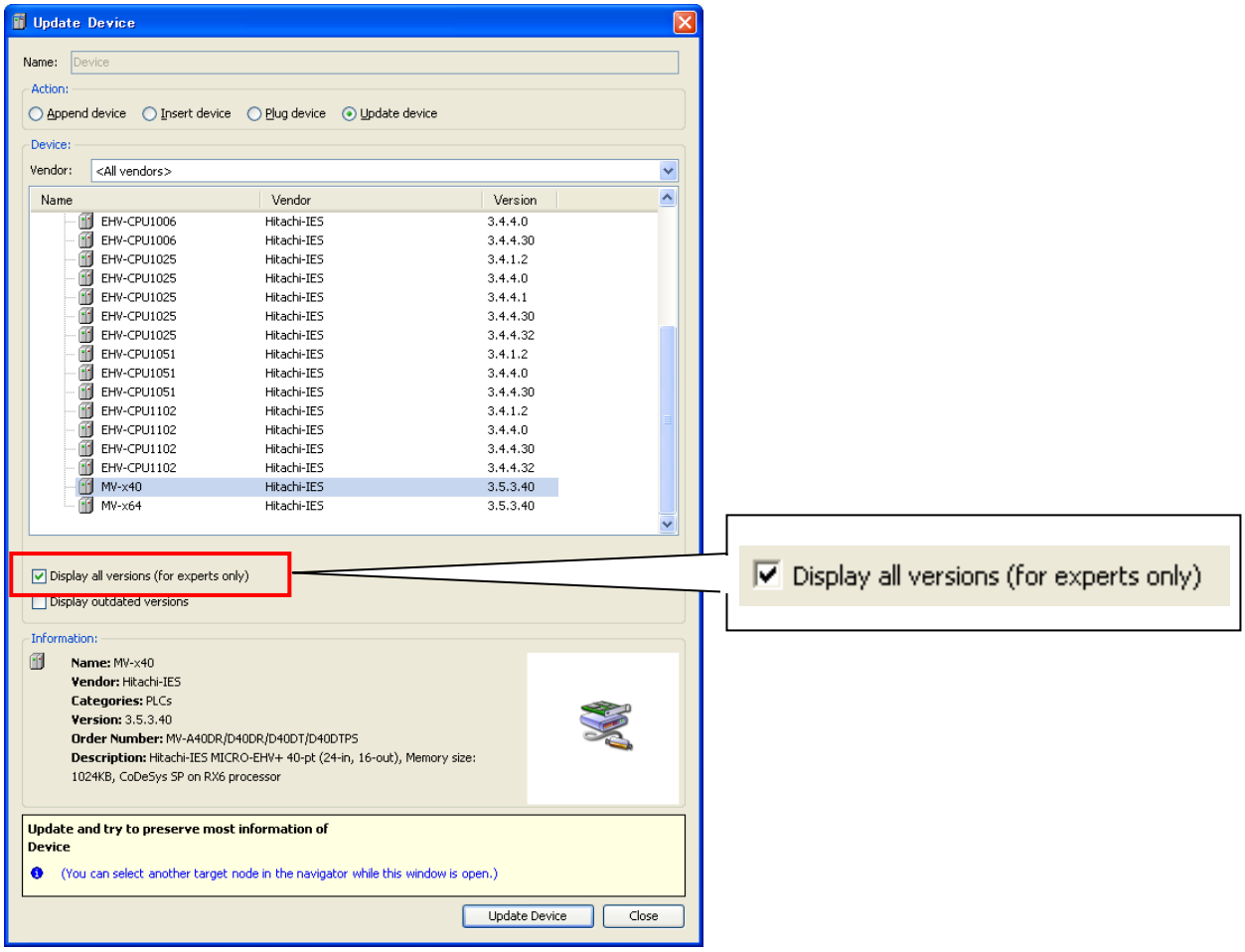
Model names	Device Names
OBV-NES	Option_Board
OBV-485A	

3.3.4 Update Device

Although device (CPU) type is required to set when creating new project, it can be changed later.
Right mouse click on the device and choose “Update Device”. Then “Update Device” windows appears.



Choose one of the devices and click [Update Device] button. If necessary, click at “Display all versions (for experts only)” and choose the certain device.



Note

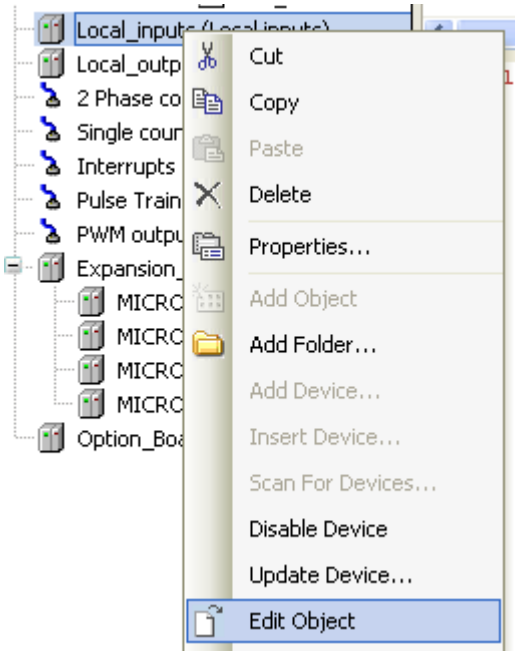
“Display all versions (for experts only)” is displayed in professional mode only. If you use standard mode, please switch to professional mode by choosing [Tool] – [Options] as shown in the section 3.2 Startup.

3.3.5 I/O address

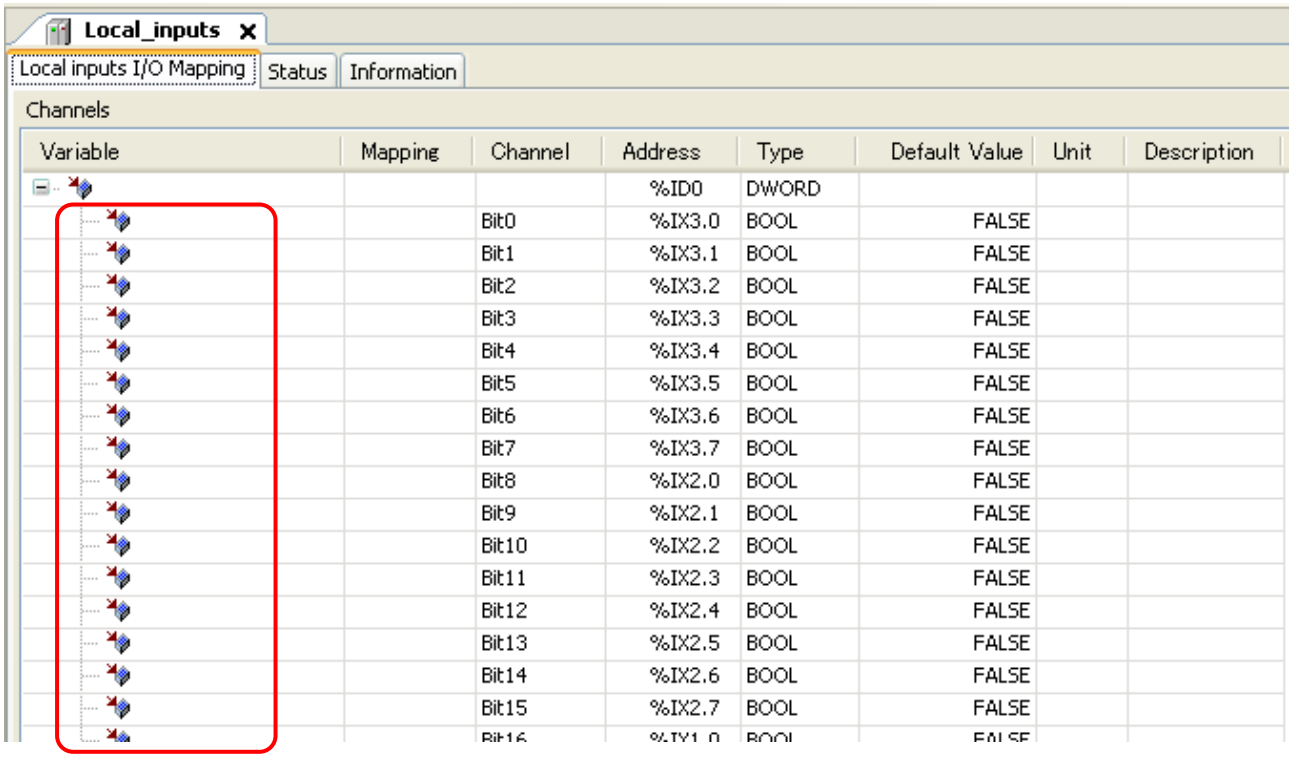
I/O addresses and variable names can be linked in two different ways: Global variable or Local variable as below.

[Global variable]

Double click on plugged I/O module or right click and choose “Edit Object”.

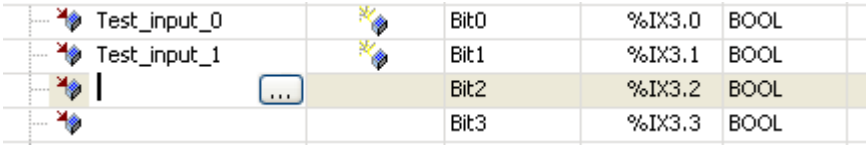


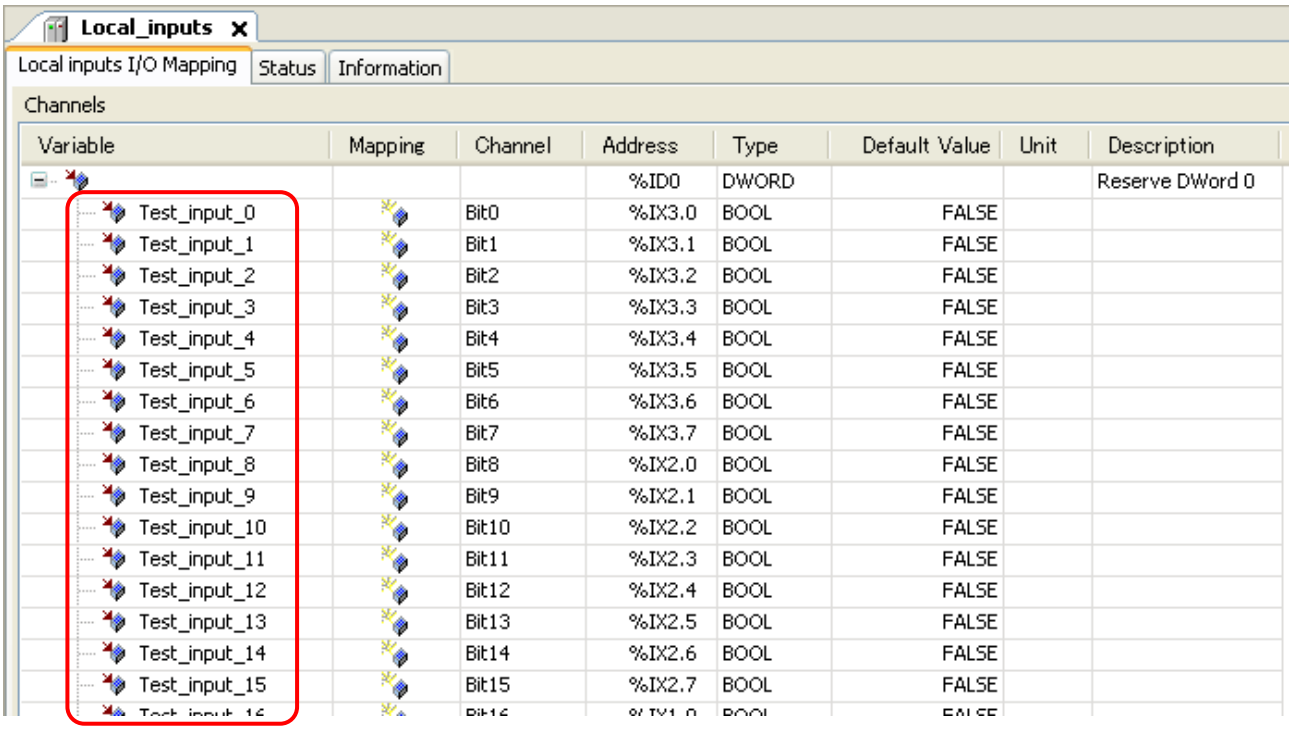
I/O Mapping window appears as below. Due to Motorola type byte order of RX processor, IEC address is not started with 0.0 as follows. Since the bit number shown at “Channel” corresponds to actual signal number, put variable names according to the bit number.



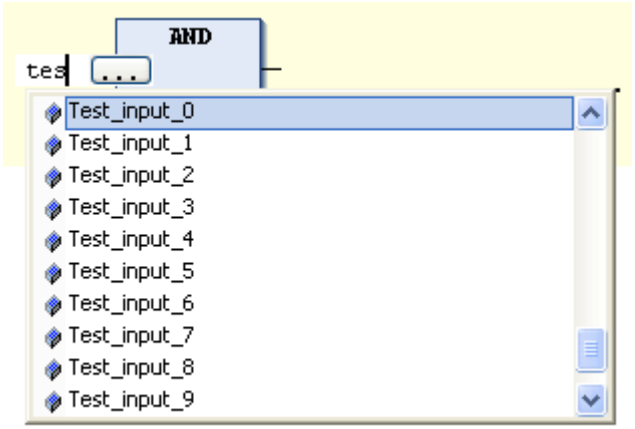
Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
			%ID0	DWORD			
		Bit0	%IX3.0	BOOL	FALSE		
		Bit1	%IX3.1	BOOL	FALSE		
		Bit2	%IX3.2	BOOL	FALSE		
		Bit3	%IX3.3	BOOL	FALSE		
		Bit4	%IX3.4	BOOL	FALSE		
		Bit5	%IX3.5	BOOL	FALSE		
		Bit6	%IX3.6	BOOL	FALSE		
		Bit7	%IX3.7	BOOL	FALSE		
		Bit8	%IX2.0	BOOL	FALSE		
		Bit9	%IX2.1	BOOL	FALSE		
		Bit10	%IX2.2	BOOL	FALSE		
		Bit11	%IX2.3	BOOL	FALSE		
		Bit12	%IX2.4	BOOL	FALSE		
		Bit13	%IX2.5	BOOL	FALSE		
		Bit14	%IX2.6	BOOL	FALSE		
		Bit15	%IX2.7	BOOL	FALSE		
		Bit16	%IX1.0	BOOL	FALSE		

Input any variable names in the field “Variable” according to your system.

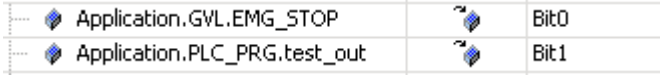




After defining variable names, they will be automatically listed up when it is used in all POU with assist of auto-complete.



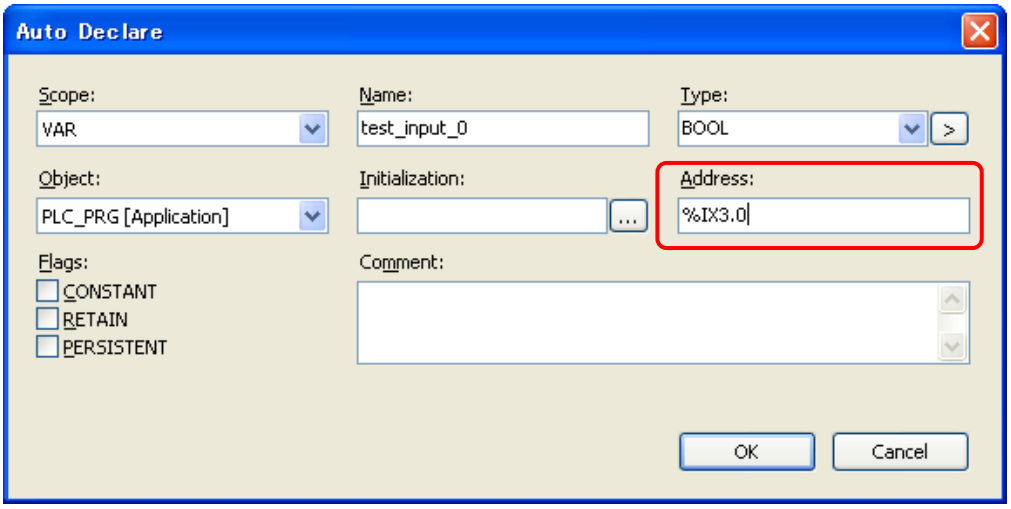
If a variable is already used (declared) in POU or global variable list, it can be taken by clicking ... icon in I/O mapping window. (... icon appears by clicking empty field.)



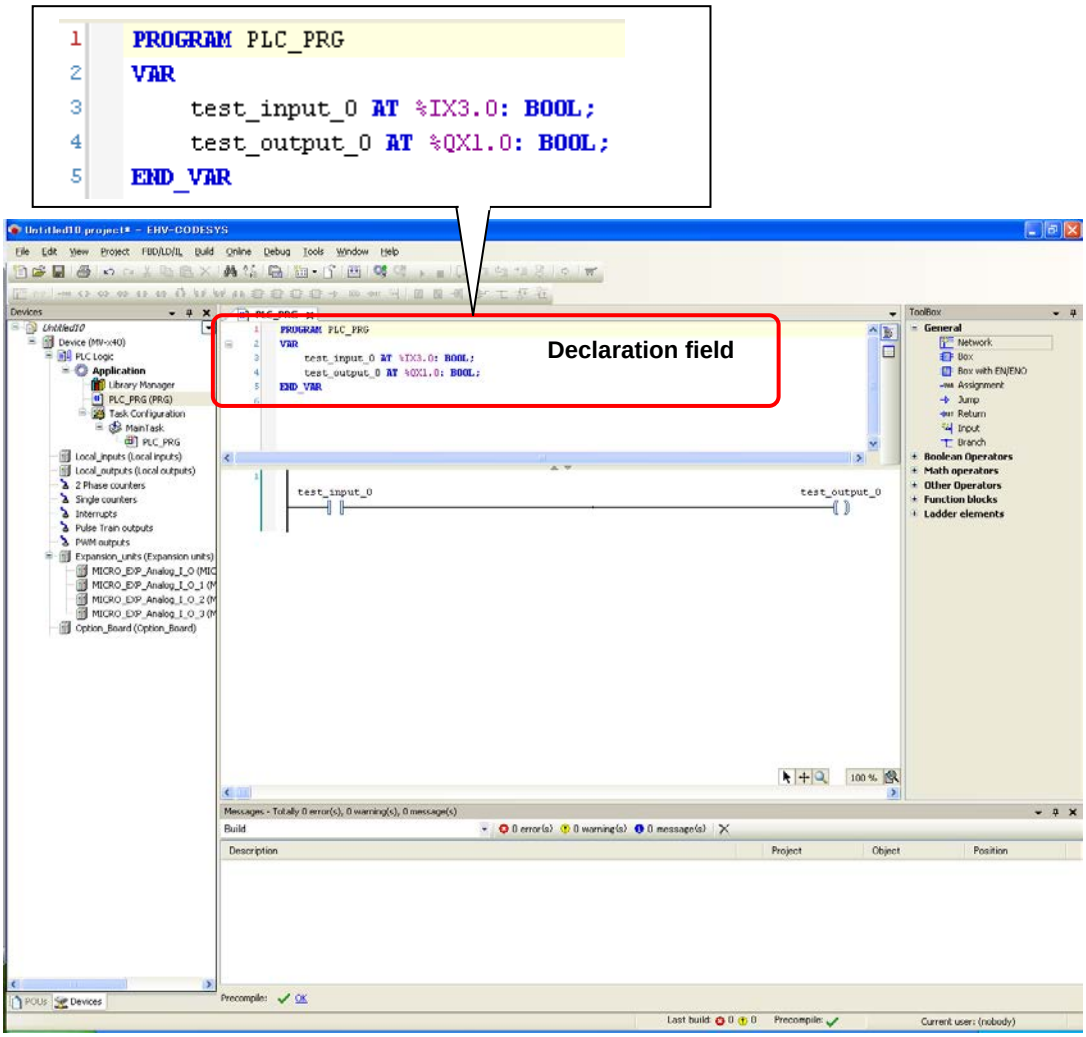
[Local variable]

Local variables are defined in each POU and valid only in the POU.

If new variable name is used in the first time, Auto Declare window will appear as below. In this window, there is an input field “Address”. Enter I/O address in this field according to data types. If it is remained as blank, the variable will be mapped in memory area.



After clicking [OK] button, declared information is added automatically as below.



I/O address example of 64 points basic / expansion unit

[Input]

Bit number	BOOL	LWORD	
Bit 0	%IX7.0	%IL0	LSB
Bit 1	%IX7.1		
Bit 2	%IX7.2		
Bit 3	%IX7.3		
Bit 4	%IX7.4		
Bit 5	%IX7.5		
Bit 6	%IX7.6		
Bit 7	%IX7.7		
Bit 8	%IX6.0		
⋮	⋮		
Bit 15	%IX6.7		
Bit 16	%IX5.0		
⋮	⋮		
Bit 23	%IX5.7		
Bit 24	%IX4.0		
⋮	⋮		
Bit 31	%IX4.7		
Bit 32	%IX3.0		
⋮	⋮		
Bit 39	%IX3.7		MSB

[Output]

Bit number	BOOL	DWORD	
Bit 0	%QX3.0	%QD0	LSB
Bit 1	%QX3.1		
Bit 2	%QX3.2		
Bit 3	%QX3.3		
Bit 4	%QX3.4		
Bit 5	%QX3.5		
Bit 6	%QX3.6		
Bit 7	%QX3.7		
Bit 8	%QX2.0		
⋮	⋮		
Bit 15	%QX2.7		
Bit 16	%QX1.0		
⋮	⋮		
Bit 23	%QX1.7		MSB

Internal I/O address example

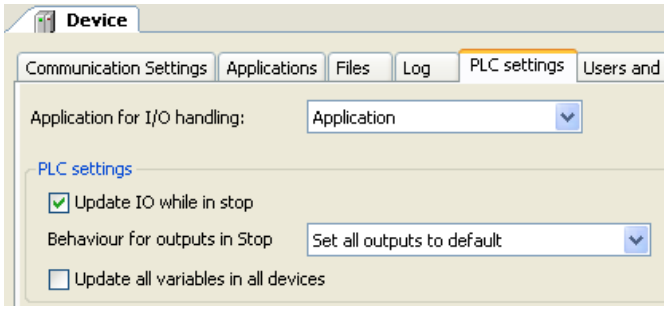
Bit number	BOOL	BYTE	WORD	DWORD	LWORD	
Bit 0	%MX7.0	%MB7	%MW3	%MD1	%ML0	LSB
⋮	⋮					
Bit 7	%MX7.7					
Bit 8	%MX6.0	%MB6				
⋮	⋮					
Bit 15	%MX6.7					
Bit 16	%MX5.0	%MB5	%MW2			
⋮	⋮					
Bit 23	%MX5.7					
Bit 24	%MX4.0	%MB4				
⋮	⋮					
Bit 31	%MX4.7					
Bit 32	%MX3.0	%MB3	%MW1	%MD0		
⋮	⋮					
Bit 39	%MX3.7					
Bit 40	%MX2.0	%MB2				
⋮	⋮					
Bit 47	%MX2.7					
Bit 48	%MX1.0	%MB1	%MW0			
⋮	⋮					
Bit 55	%MX1.7					
Bit 56	%MX0.0	%MB0				
⋮	⋮					
Bit 63	%MX0.7					MSB

Following 5 different codes access the same bit.

```
%MX7.0:=1;
%MB7  :=1;
%MW3  :=1;
%MD1  :=1;
%ML0  :=1;
```

3.4 I/O-update

Input data is read at the beginning of a task and output data is written at the end of a task. I/O-update settings are configured in “PLC settings” in Device tab. Be noted that only used I/Os in program are updated.



Update IO while in stop

If this option is activated (default), the values of the input and output channels get also updated when the PLC is stopped.

Behaviour for outputs in Stop

Keep current values: The current values will not be modified. If “Update IO while is stop” is deactivated, output data is not updated at CPU stopping.

Set all outputs to default: The default values resulting from the mapping will be assigned. If this setting is used, “Reset all outputs in STOP” of [Device]-[Configurtion] parameter must be set as “No”, otherwise default value of TRUE is not valid. Refer to the next page for further information.

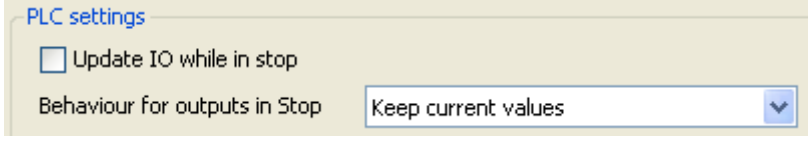
Execute program: You might determine the outputs behaviour by a program available within the project. Enter the name of this program here and it will be executed when the PLC gets stopped. Via button [...] the input Assistant can be used for this purpose.

Update all variables in all devices

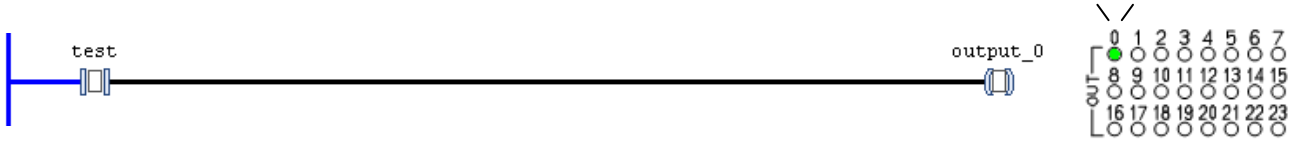
If this option is activated, then for all devices of the current PLC configuration all I/O variables will get updated in each cycle of the bus cycle task. This corresponds to option “Always update variables”, which can be set separately for each device in the "I/O Mapping" dialog.

Note

If PLC settings are configured as follows (“Update IO while in stop” disabled and “Keep current values” in “Behaviour for outputs in Stop”) and Reset warm or Reset cold is operated, the last status of output remains although monitored output status is reset.



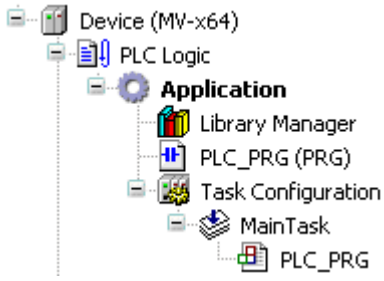
Actual output remains after Reset warm/cold



This is expected behaviour. If this setting combination is required, keep in mind this mismatching and be careful to use.

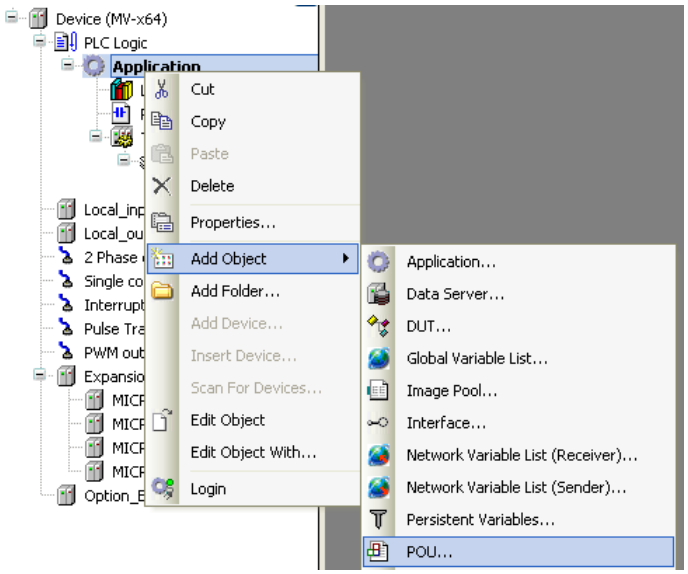
3.5 POU and task

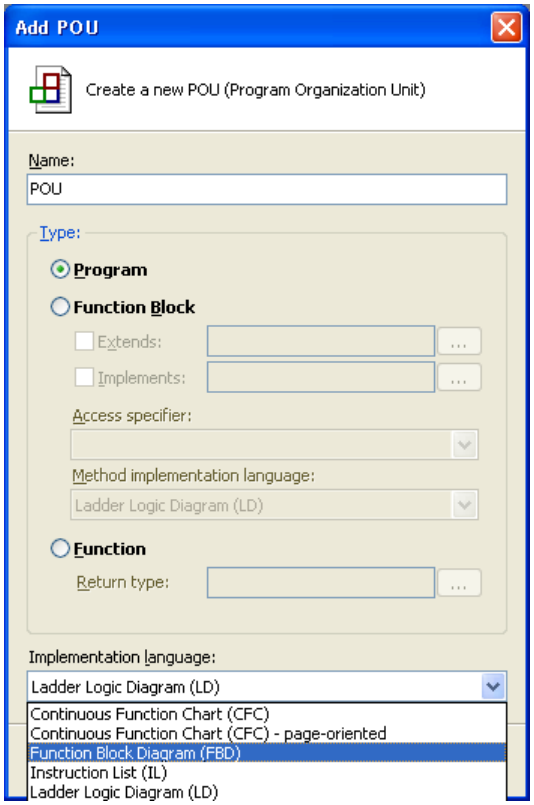
One application has at least one POU and one task as shown below.



POU

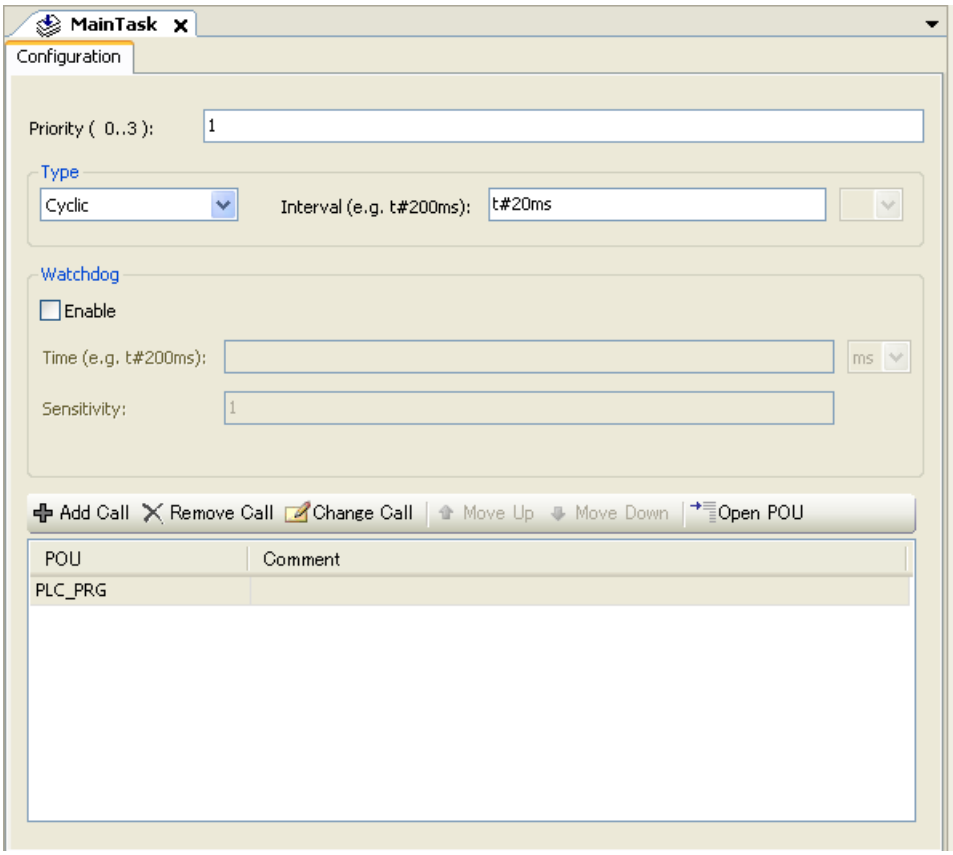
POU stands for Program Organization Unit. This can be assumed as a paper to create your program. Only one programming language can be used in one POU. If you need another language, add POU by right click on “Application” and choose “Add object”-“POU” and choose language.





Task

POU does not have information how to execute POU. This information is handled by task.
Put priority, choose type of task and add or remove POU accordingly.



Priority (0-3)

0 is the highest priority, 3 is the lowest.

Cyclic task

The task will be processed cyclic according to the time definition given in the field “Interval”. If the execution time of user program exceeds 80% of cycle time of the task, then CPU stops with processor load exception detected (error code 25).

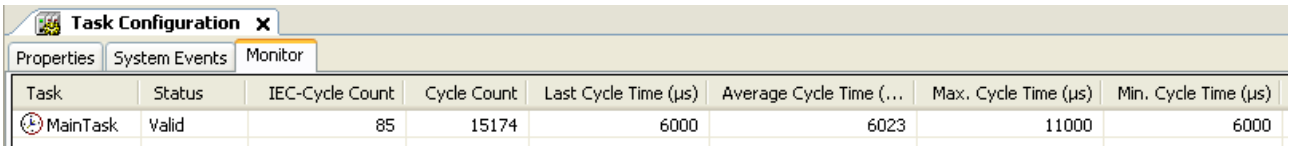
Event task

The task will be started as soon as the variable defined in the field gets a rising edge.

Freewheeling task

The task will be processed as soon as the program is started and at the end of one run will automatically restarted in a continuous loop. There is no cycle time defined. Be noted that the priority of this task is the lowest and 100ms of sleeping time is added at the end of each cycle for other tasks to be executed properly.

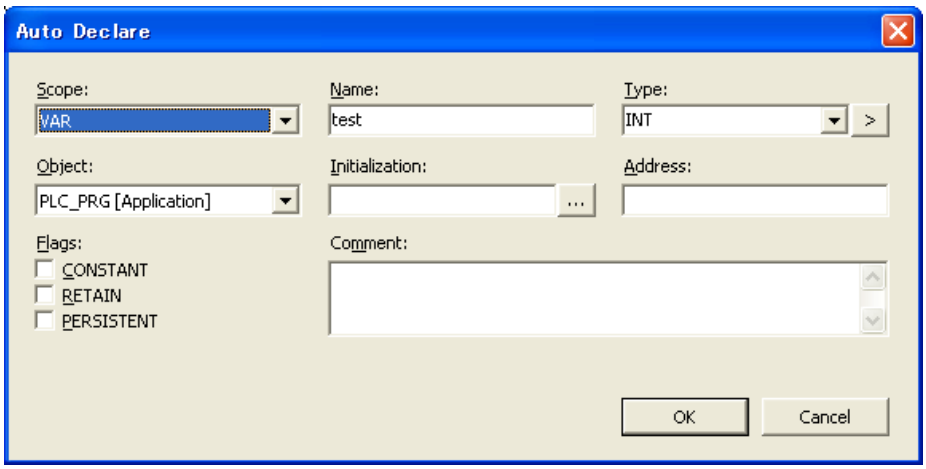
Actual cycle time of each task is monitored in Task configuration as below.



3.6 Variables

3.6.1 Data memory

In EHV-CODESYS programming, external I/Os and data memory (internal registers) are handled as variable names instead of direct I/O addresses, such like “A1_switch”. If new variable name is used, below Auto Declare window appears. Enter an each field according to following table.



Item		Descriptions
Scope		Choose “VAR” in normal use. If global variable is used, choose “VAR_GLOBAL”. Refer to section 3.6.7 Global variable for further information.
Name		Variable name is defined. Refer to section 3.6.3 Available characters for variable names.
Type		Data type is defined. Refer to section 3.6.5 Data type.
Object		In case of local variable, POU name is defined.
Initialization		Initial value when program starting can be set here. If it’s blank, initialization value is 0.
Address		No need to enter I/O address. EHV-CODESYS will assign to free address automatically.
Comment		Any text comment can be input.
Flags	CONSTANT	Enter a value in the Initialization field.
	RETAIN	The value is maintained by a battery after switch off of the PLC. If new application is downloaded, it will be initialized. (Refer to section 3.13 Run / Stop / Reset)
	PERSISTENT	The value is maintained by a battery after switch off of the PLC. If new application is downloaded, it will be maintained. (Refer to section 3.13 Run / Stop / Reset)

Bit access

Any bit data in integer type data can be accessed by adding suffix dot and number (decimal 0 to 63).

Example

```
wTest :=5;   <-- WORD type (16 bits)

wTest.0;
wTest.1;     <-- BOOL type (1 bit)
wTest.2;
wTest.3;
```

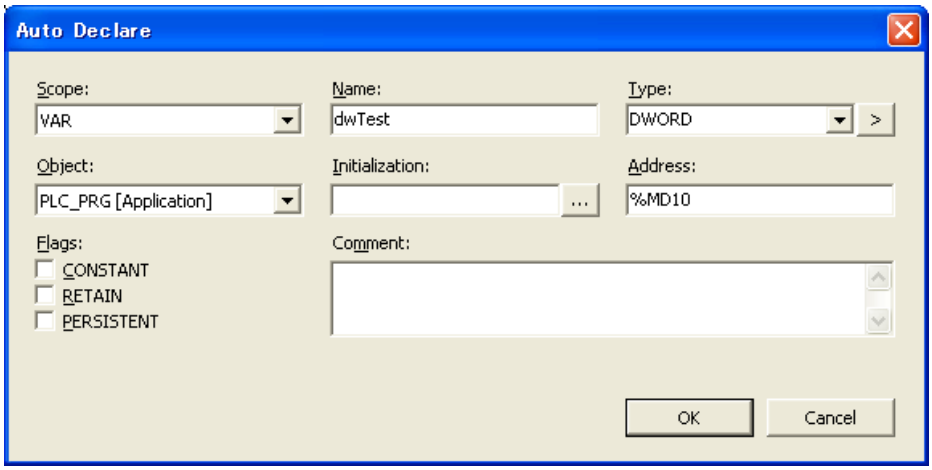
Login display

```
wTest 16#0005 :=5;

wTest 16#0005 . 0 TRUE;
wTest 16#0005 . 1 FALSE;
wTest 16#0005 . 2 TRUE;
wTest 16#0005 . 3 FALSE;
```

3.6.2 Marker memory

Normally users do not have to take care about internal address of data memory however, if needed, the marker memory is useful. The address of marker memory is %M.



For example, DWORD data dwTest, WORD data wTest_H and wTest_L are declared in the address %MD10, %MW20 and %MW21. Then high word and low word can be accessed separately with using %M addresses. The relation between each data types are same as page 3-14. Just replace “Q” with “M”. The marker memory does not support RETAIN nor PERSISTENT flags.

Variable declaration

```
VAR
    dwTest AT %MD10: DWORD;
    wTest_H AT %MW20: WORD;
    wTest_L AT %MW21: WORD;
END_VAR
```

Login display

Expression	Type	Value
dwTest	DWORD	16#12345678
wTest_H	WORD	16#1234
wTest_L	WORD	16#5678

The max. size of marker memory is 16KB. Supported address range is shown below.

Data type	Address range
BOOL	%MX0.0 to %MX16383.7
BYTE	%MB0 to %MB16383
WORD	%MW0 to %MW8191
DWORD	%MD0 to %MD4095
LWORD	%ML0 to %ML2047

3.6.3 Available characters for variable names

Available characters for variable names are only alphabet a to z, A to Z and number 0 to 9 and _ (underscore). The first character must not be numeric characters. Several words like BOOL, WORD, IF, FOR etc. are reserved.

Supported characters

Types	Supported	Remarks
Numerical	`0 to 9`	Not allowed to begin with numeric characters.
Alphabetical	`a to z, A to Z`	
Symbol	_	Trailing underscores are not allowed.

Examples for variable names

Allowed or not	Examples	Descriptions
Allowed	`Test_200`	
	`TEST`	
	`Test55`	
	`_Test`	
Not allowed	`2test`	Starting with numeric character.
	`test__200`	Trailing underscores are not allowed
	`test-5`	Minus sign is not allowed.
	`test#3`	other signs than underscore are not allowed.
	`test 3`	Space is not allowed.
	`IF`	Reserved word.

3.6.4 Constant

Constant values are expressed in the following format.

Types	Examples	Applicable for
Binary	`FALSE, TRUE, 0, 1, 2#1010`	
Decimal	`1234, 10#1234`	
Hexa-decimal	`16#1234, 16#7FFF`	
Real number	`3.14159`	data type REAL or LREAL
Time	`T#100ms, T#5.5s`	Timer (TON, etc.)
Date	`DT#2012-12-31-12:34:56`	RTC (Realtime clock)

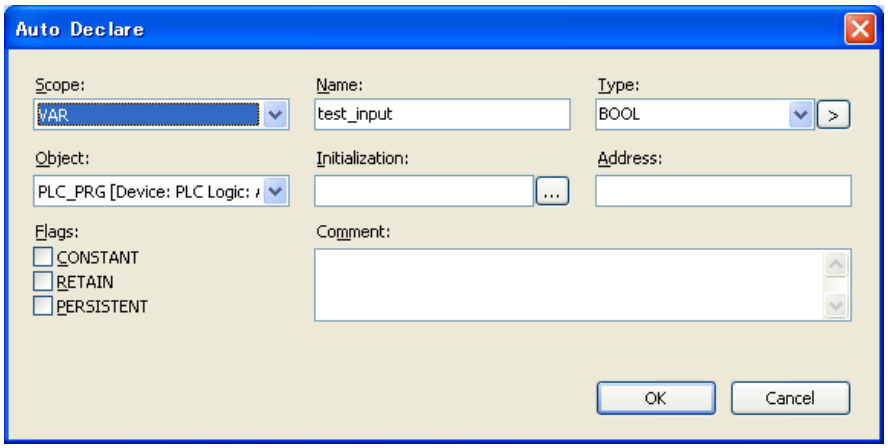
3.6.5 Data types

EHV-CODESYS supports below data types.

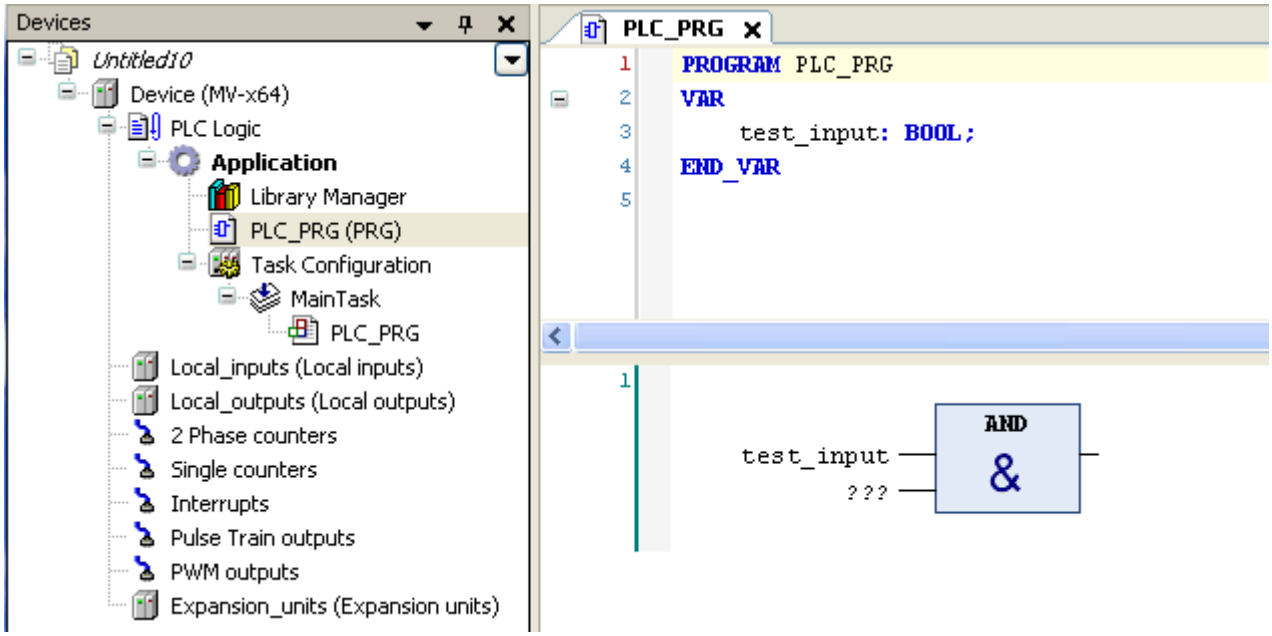
No.	Data types	Name	Size	Range
1	BOOL	Boolean	1	0 or 1
2	SINT	Short integer	8	-128 to 127
3	USINT	Unsigned short integer	8	0 to 255
4	BYTE	Bit string of length 8	8	0 to 255 (16#00 to 16#FF)
5	INT	Integer	16	-32,768 to 32,767
6	UINT	Unsigned integer	16	0 to 65,535
7	WORD	Bit string of length 16	16	0 to 65,535 (16#00 to 16#FFFF)
8	DINT	Double integer	32	-2,147,483,648 to 2,147,483,647
9	UDINT	Unsigned double integer	32	0 to 4,294,967,295
10	DWORD	Bit string of length 32	32	0 to 4,294,967,295 (16#00 to 16#FFFFFFFF)
11	REAL	Real numbers	32	±1.175494351 E-38 to 3.402823466E+38
12	TIME	Duration	32	0 to 4,294,967,295 ms Unit : ”d”: days, “h”: hours, “m”: minutes, “s”: seconds, “ms”: milliseconds Ex. T#100S12ms, t#0.1s
13	LREAL	Long reals	64	±1.7976931348623... E+308 to 2.2250738585072... E-308
14	STRING	Variable-length single-byte character string	8× n	1 to 255 char.
15	LINT	Long integer	64	-2^{63} ～ $2^{63}-1$
16	ULINT	Unsigned long integer	64	0 to $2^{64}-1$
17	LWORD	Bit string of length 64	64	0 to $2^{64}-1$
18	DATE	Date	32	year-month-day Ex. DATE#1996-05-06 d#1972-03-29
19	DATE_AND_TIME	Date and time of Day	32	year-month-day-hour:minute:second Ex. DATE_AND_TIME#1996-05-06-15:36:30 dt#1972-03-29-00:00:00
20	TIME_OF_DAY	Time of day	32	hour:minute:second Ex. TIME_OF_DAY#15:36:30.123 tod#00:00:00
21	LTIME	Long duration	64	Unit :“us”: microseconds, “ns”: nanoseconds Ex. LTIME#1000d15h23m12s34ms2us44ns tod#00:00:00
22	WSTRING	Variable-length double-byte character string	16× n	
23	ARRAY	Array	—	Ex. in variable declaration test: ARRAY[0..100] OF WORD; in user program test[5]:=20;

3.6.6 Local variable

If new variable name is used in POU, Auto Declare window appears as below. If the field “Address” is remained as empty, this variable will be assigned in a certain memory area of CPU.



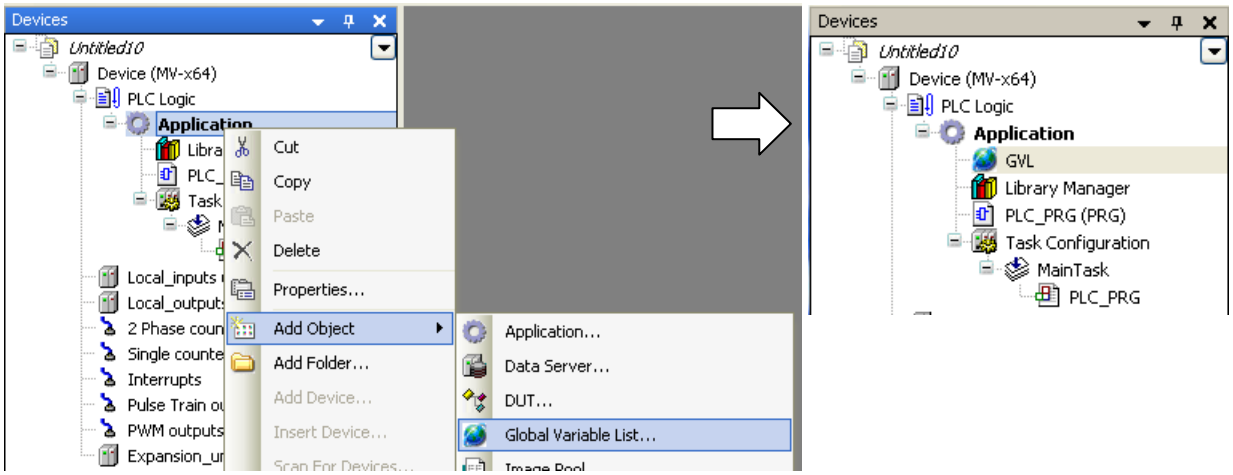
Click [OK] button, this variable is registered in declaration part of POU as below.



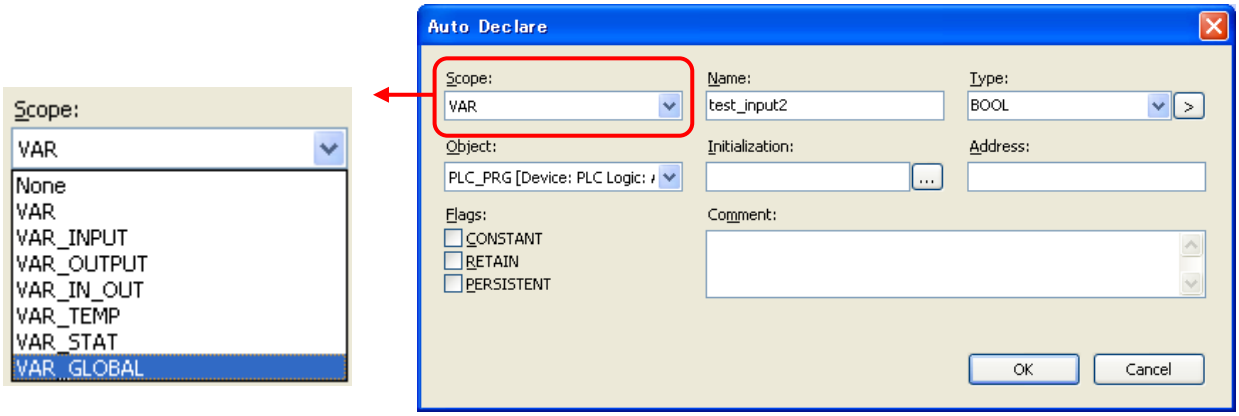
This variable is valid only in the POU. Even if same variable name is used in another POU, Auto Declare window will appear and it will be assigned in another memory location and handled as different variable.

3.6.7 Global variable

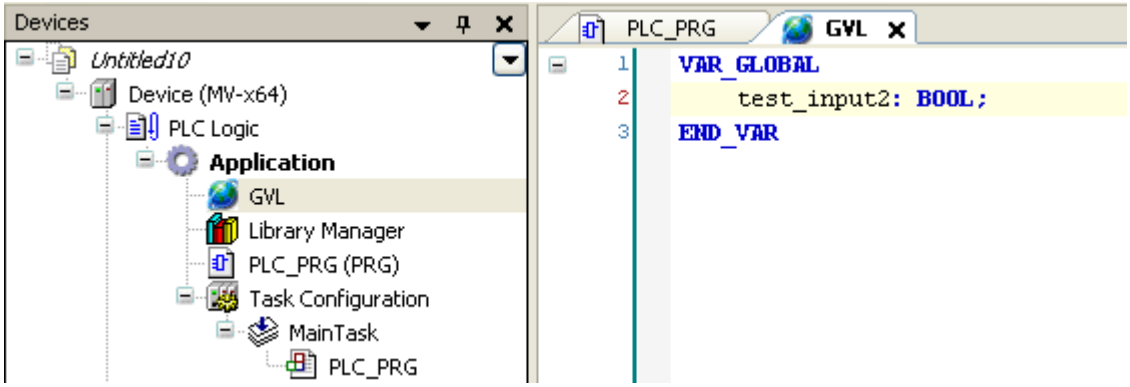
If variables need to be commonly used in all POUs, “Global Variable List” must be created by right click on Application as below.



If new variable name is used in POU, Auto Declare window appears as shown in local variables. Choose “VAR_GLOBAL” at “Scope” as below.



New variable name “test_input2” is registered in GVL as below instead of POU.

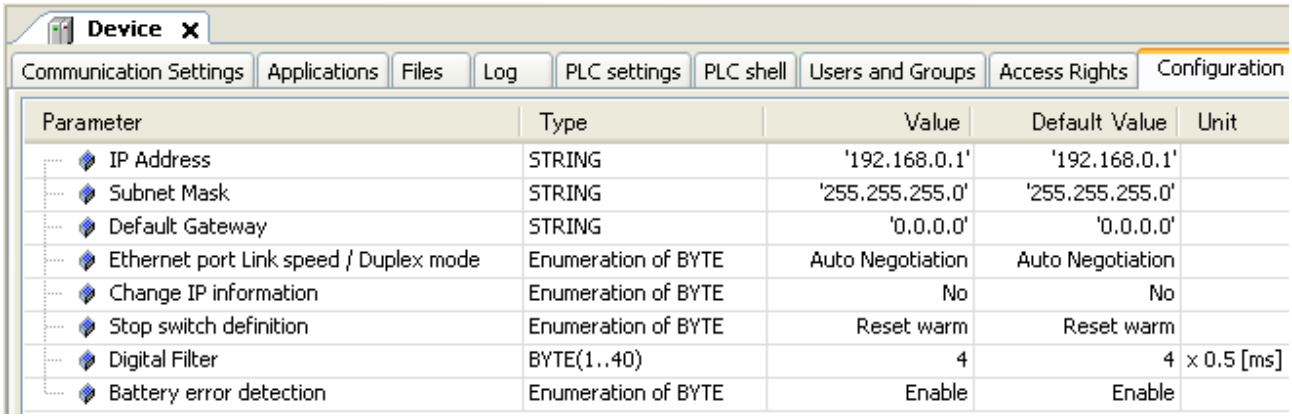


3.7 Configuration

Open device window and set CPU's parameters in Configuration tab.

Parameter	Description	
IP Address	When requesting to change the Ethernet port related parameters, be sure to set "**Yes**" in "Change IP information", otherwise parameters are not downloaded. Do not forget to set back to "**No**" after downloading.	
Subnet Mask		
Default Gateway		
Ethernet port Link speed / Duplex mode *		
Change IP information	No	IP information is not downloaded when application downloading.
	Yes	IP information is downloaded together with application.
Stop switch definition	Reset warm	When Run/stop switch is changed from run to stop, "Reset warm" operation is performed.
	Stop	When Run/stop switch is changed from run to stop, "Stop" operation is performed.
Digital Filter	Digital filter is applied on the input of both basic unit and expansion units. The setting range is 1 to 40 and this function works with the value multiplied by 0.5ms.	
Battery error detection	Enable	MICRO-EHV+ detects battery error (error code 71).
	Disable	MICRO-EHV+ does not detect battery error (error code 71).

*: The parameter of "Ethernet port Link speed / Duplex mode" should be fixed to "Auto Negotiation".



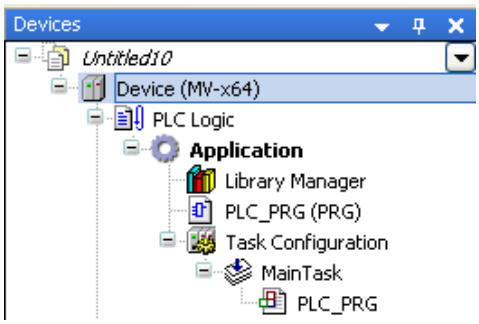
Note

- The parameter "Ethernet port Link speed / Duplex mode" requires power cycling to update parameter data. All the other parameters are updated when program is downloaded.

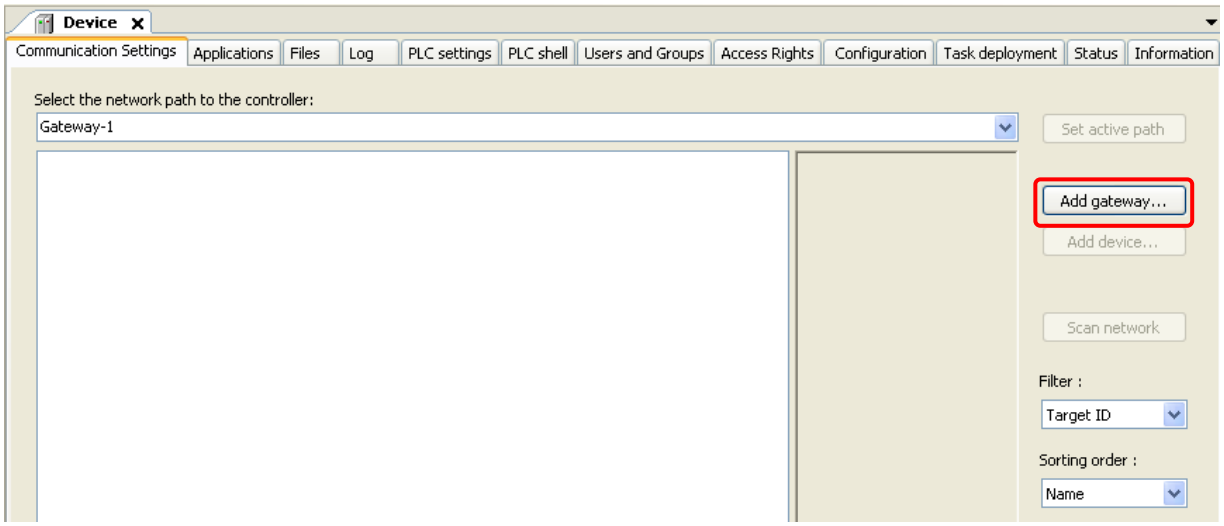
3.8 Communication settings

How to configure

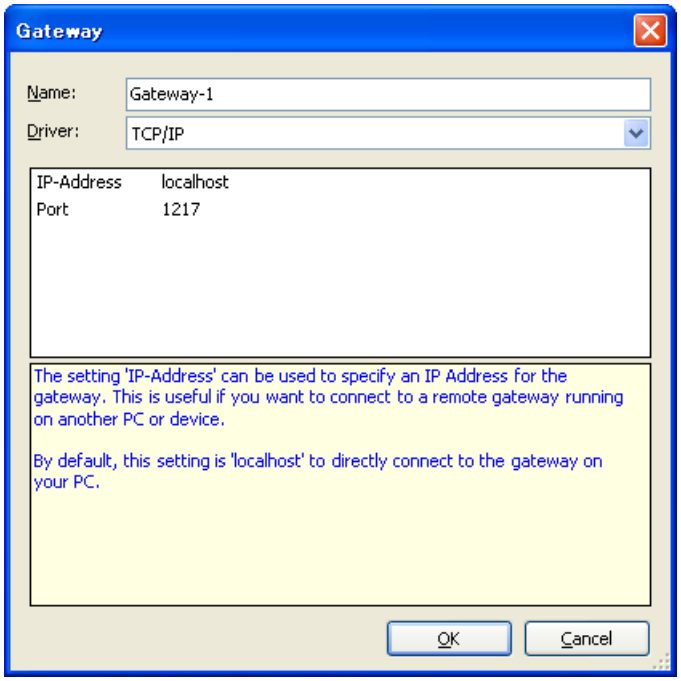
Double click on “Device (MV-xxx)” or right click and choose “Edit Object”.



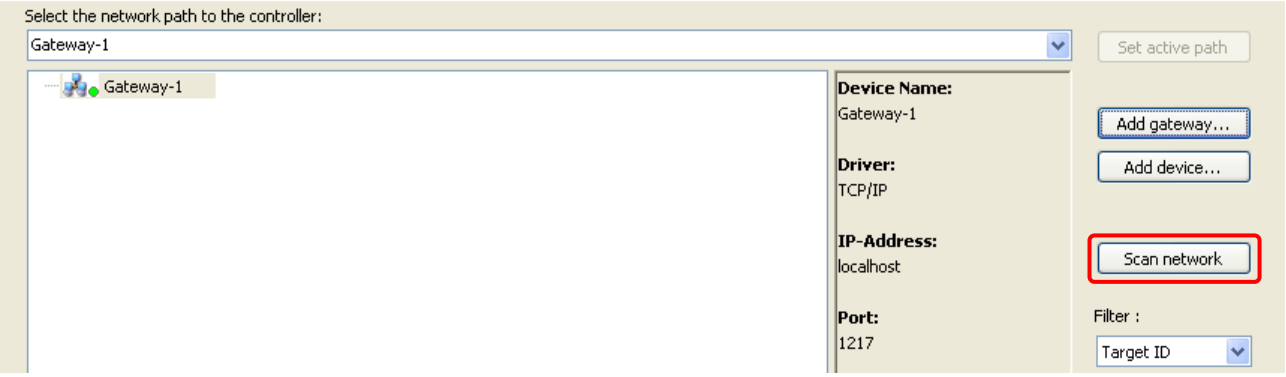
“Device” window will appear as below. Choose “Communication Settings” tab and click “Add gateway”.



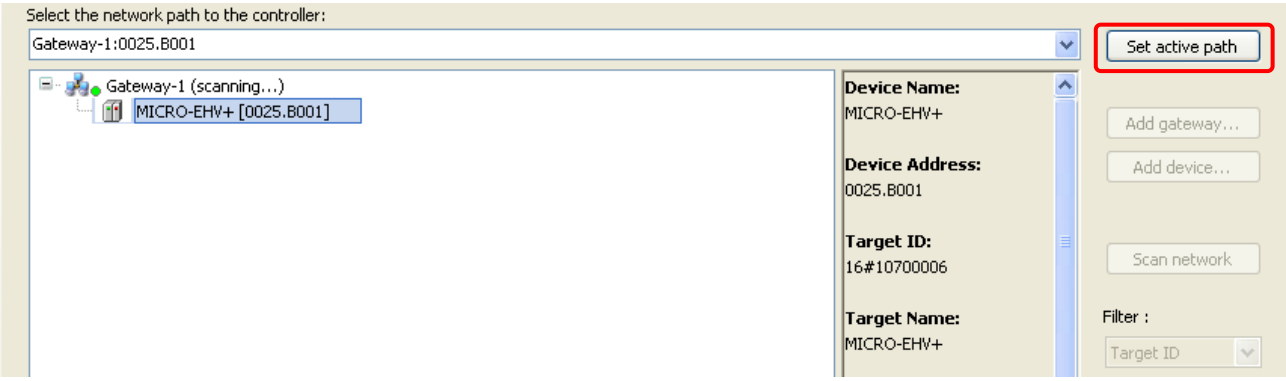
“Gateway” window will appear. Click “OK”. Sine the communication type between EHV-CODESYS and gateway (in PC) is TCP/IP, displayed driver name is “TCP/IP” regardless of CPU’s communication types.



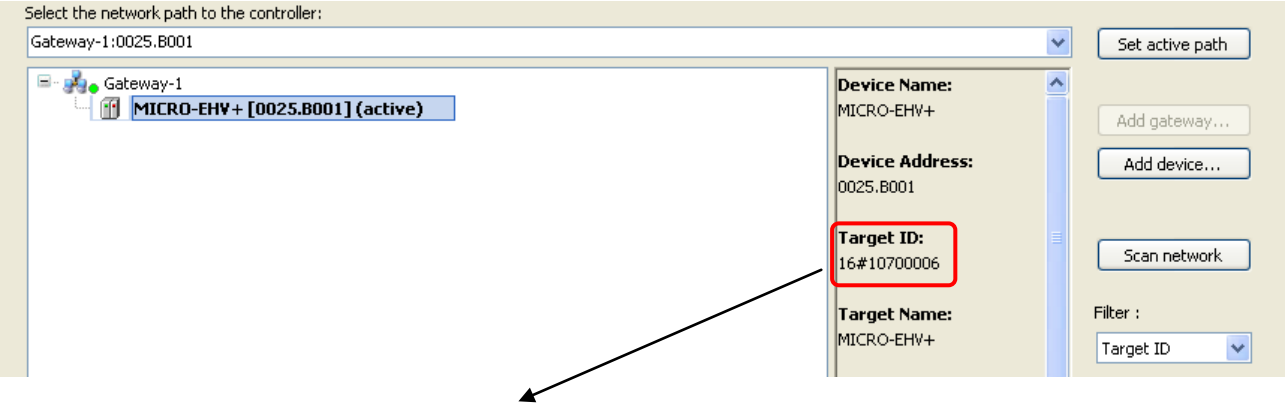
The gateway is displayed as below. Click “Scan network” to search available device in the network.



If CPU is found, it is displayed as below. Click “Set active path” to choose as the target device.



Target CPU is activated. Communication settings are completed.



CPU name	Target ID
MV-*64**	16#10700006
MV-*40**	16#10700007

Note

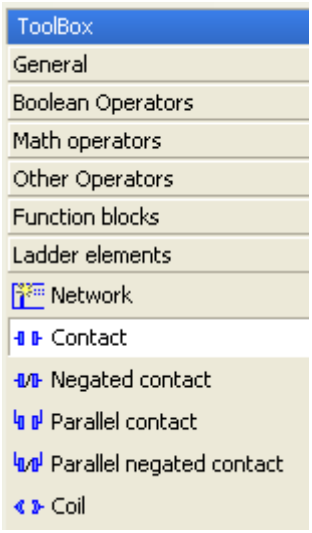
- Even if both Ethernet cable and USB cable are connected, only the first detected device is displayed.
  If the filter is changed from “Target ID” to “None”, all types of devices in the network are found.
- At the first time after USB driver installed, it could fail to found MICRO-EHV+ correctly in case of Window 7.
  In this case, login with Ethernet and then login with USB.

3.9 Programming

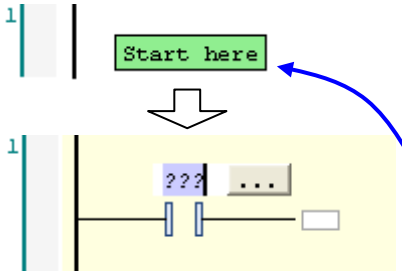
Ladder programming

Basic ladder programming is shown below as a first step. Please refer to online-help of EHV-CODESYS for further information about programming.

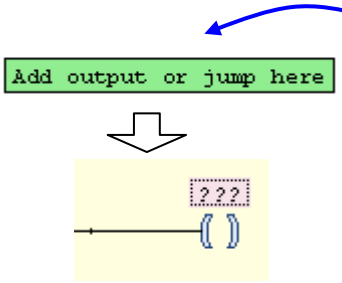
Several ways are available to add contact or coil to POU as below.

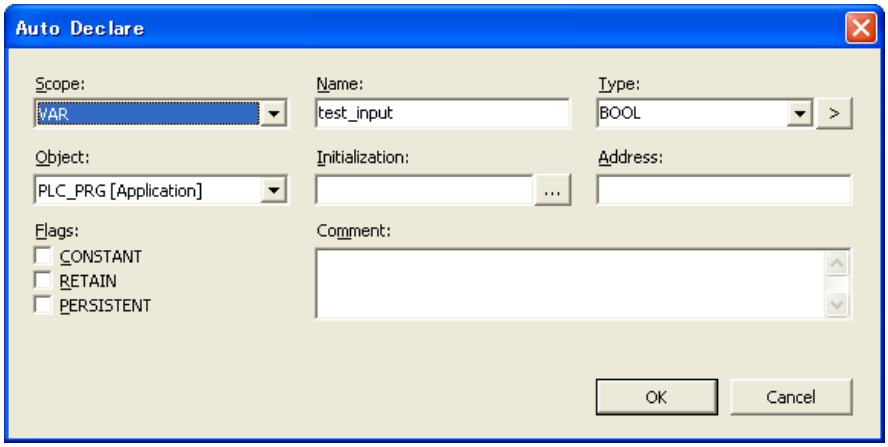


Contact
- Drag from ToolBox to [Start here].
- Menu [FBD/LD/IL]-[Insert Contact]
- Right mouse click [Insert Contact]
- Shortcut key [Ctrl + K]

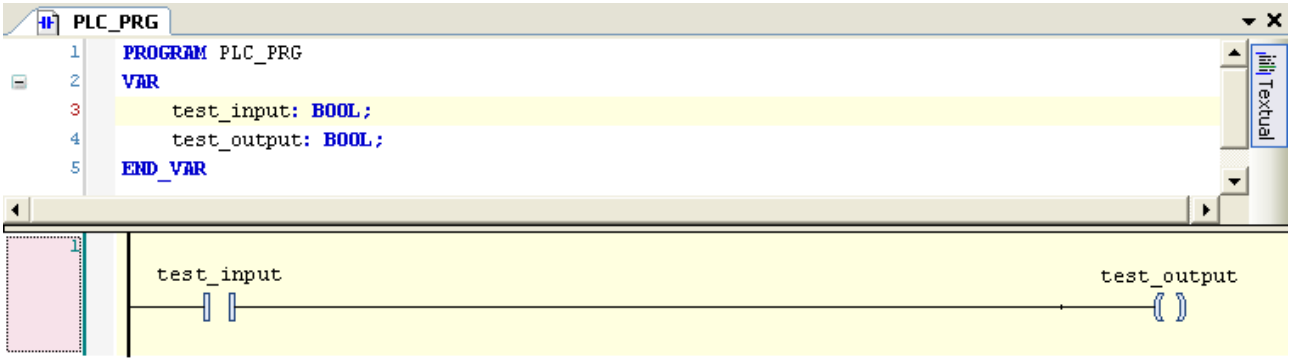


Coil
- Drag from ToolBox to [Add output or jump here].
- Menu [FBD/LD/IL]-[Insert Coil]
- Right mouse click [Insert Coil]
- Shortcut key [Ctrl + A]



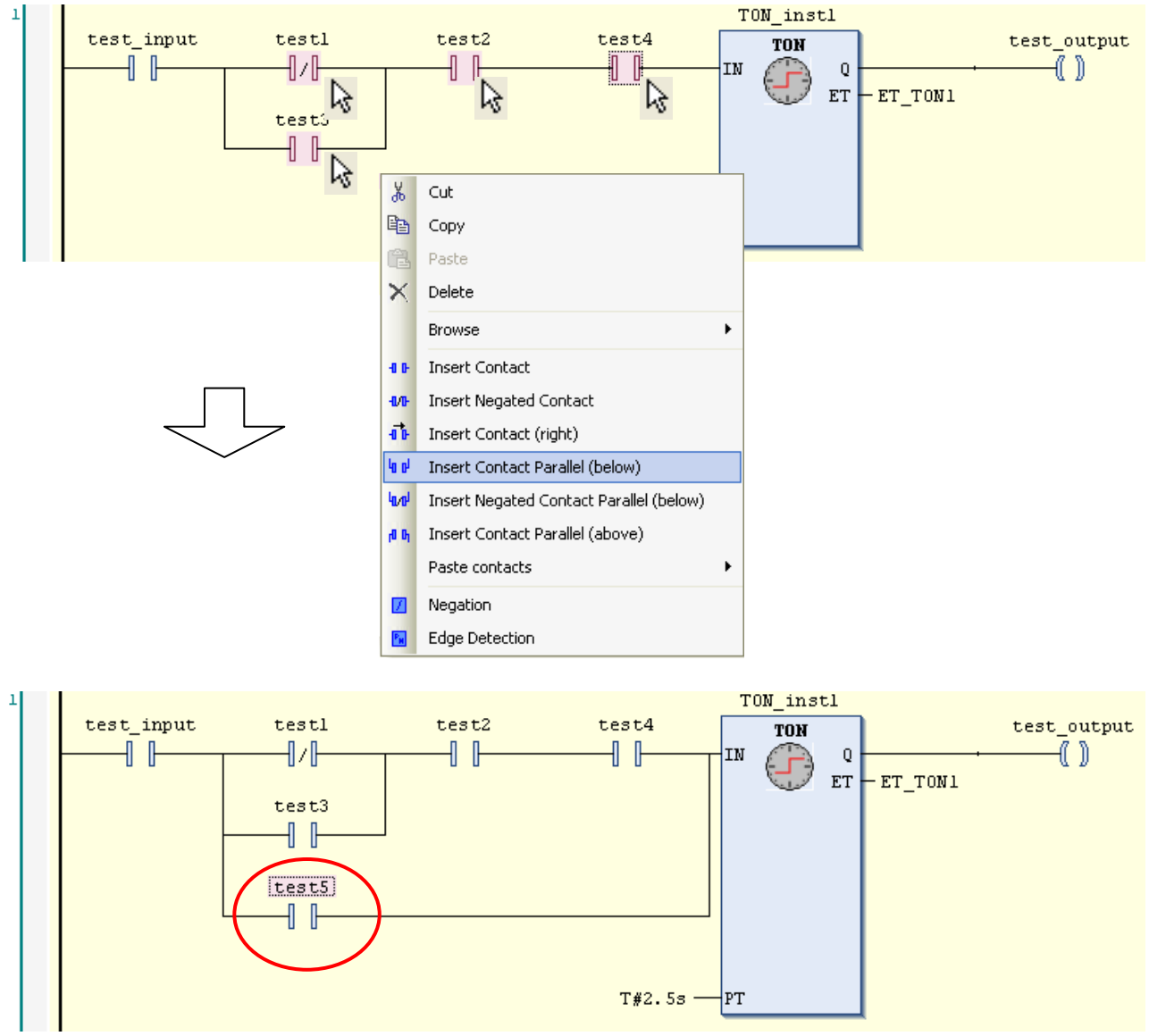


If new variable name is used, Auto Declare window appears automatically. Edit each input field and check-boxes if necessary, and Click [OK]. The variable is declared in declaration window as below.



Parallel contact across several contacts

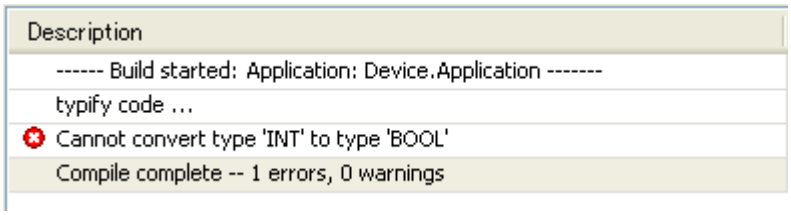
Click several contacts with shift key and choose [Insert Contact Parallel] in right-mouse click menu or press [Ctrl + R] key.



3.10 Login

Login

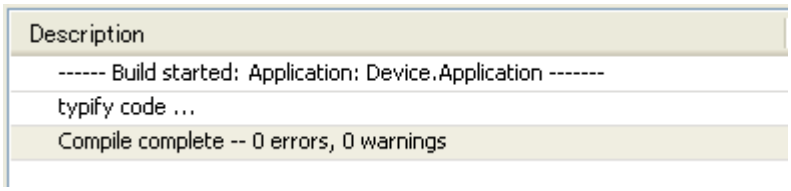
After programming, click  or choose [Build] in Build menu. If compiling fails, error information is shown at "Description" field as follows. Double click the message to jump to the part to be corrected.



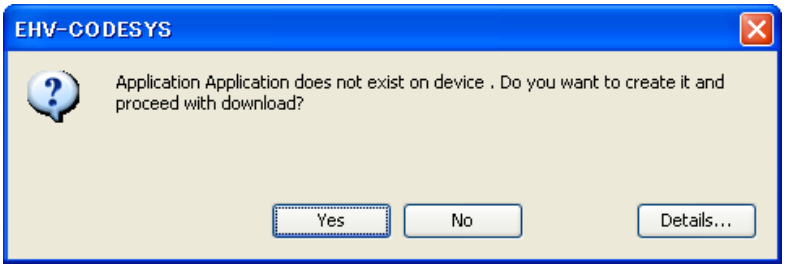
Note

If unknown message appears, it is recommended to [Clean all] in Build menu. All compile information is deleted by this operation.

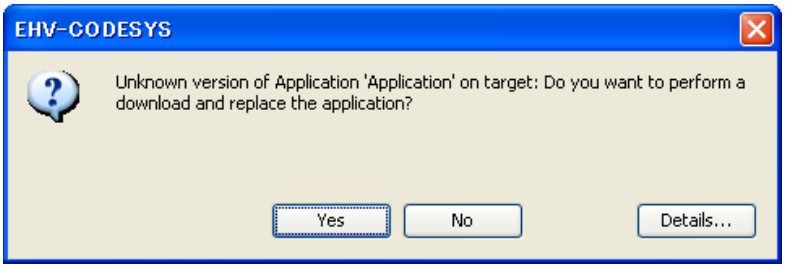
When all errors are removed as below, click  or choose [Login] in Online menu to download the program to CPU.



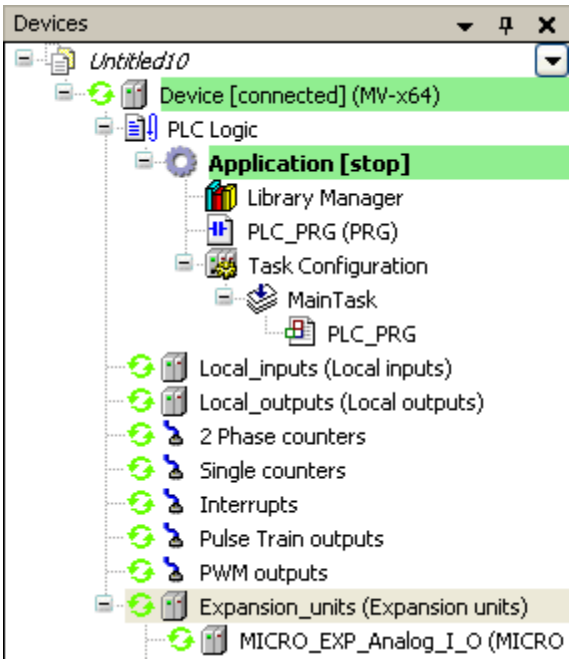
If no application is in the CPU, this message appears. Click [Yes] to download.

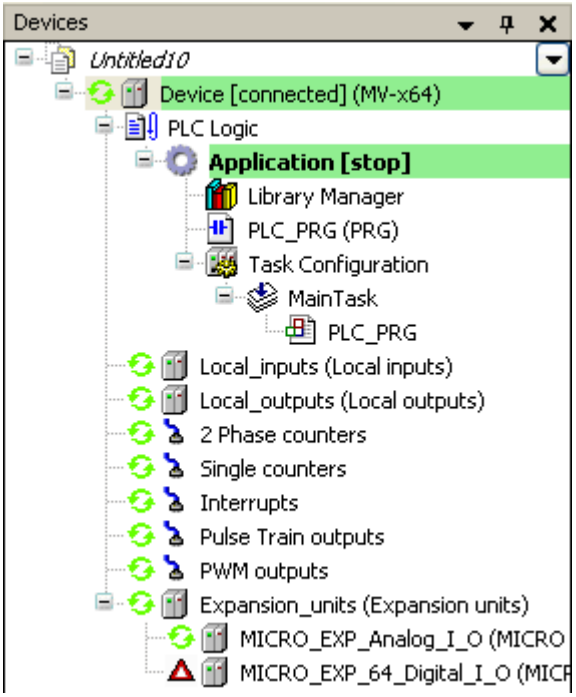


If unknown version of application is in the CPU, this message appears. Click [Yes] to download.



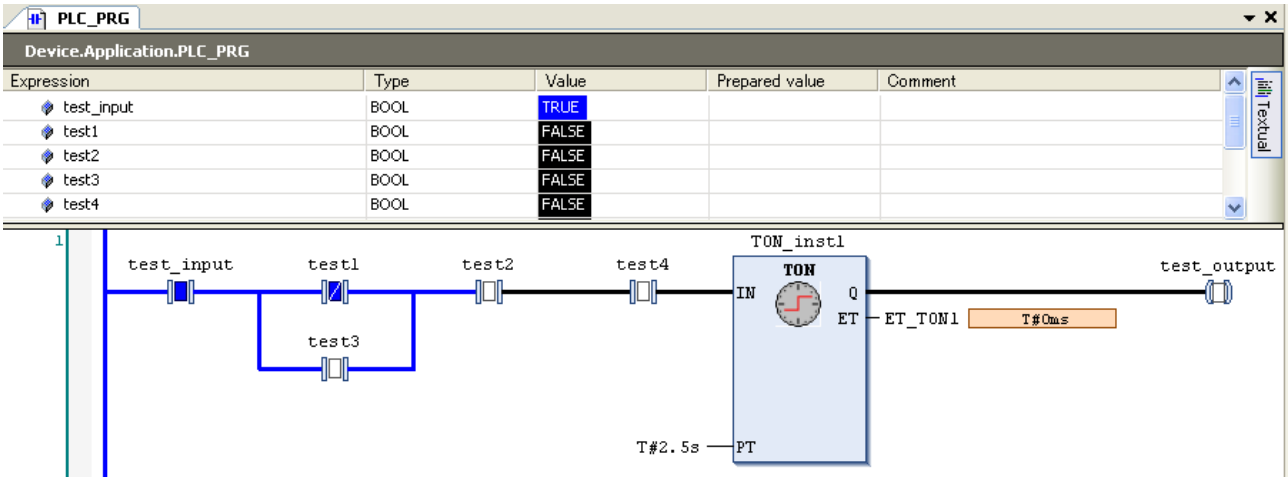
When logging in successfully, green circle icon is displayed at [Device]. If connected expansion units are matched with configured ones, green icon is displayed at each expansion unit also. If any mounted I/O module is mismatched, red triangle icon is displayed at mismatched module as below (right side).





Online monitoring

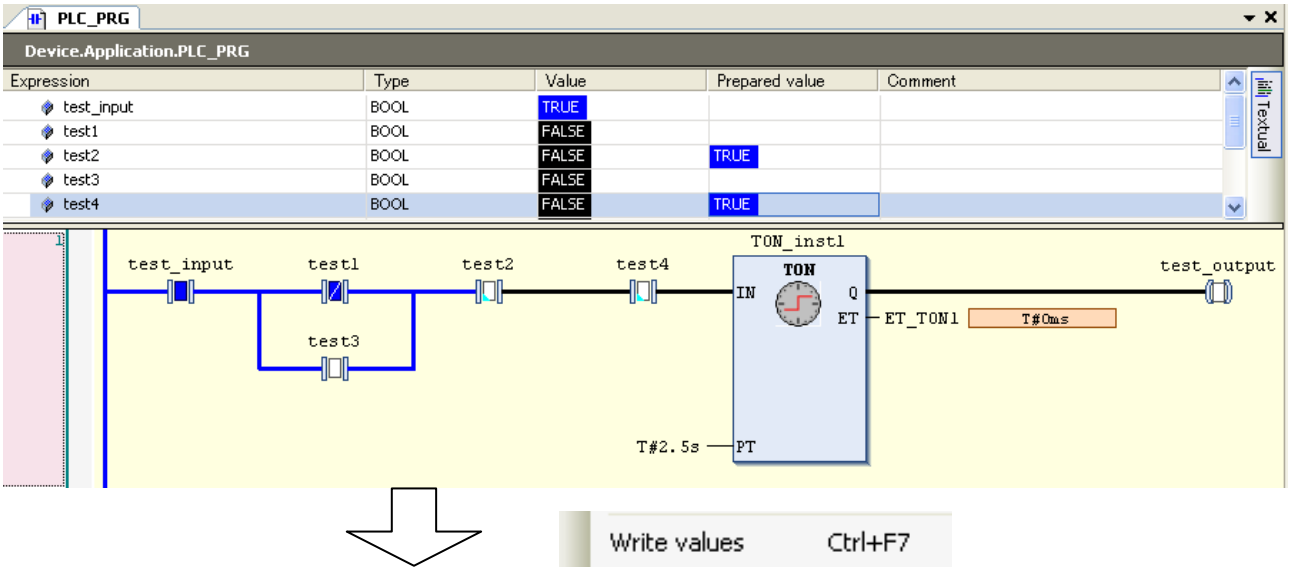
After login, actual status of variables are shown as below.



Write values

Prepare values for the variables by clicking at [Prepared value] or double clicking at ladder symbols.

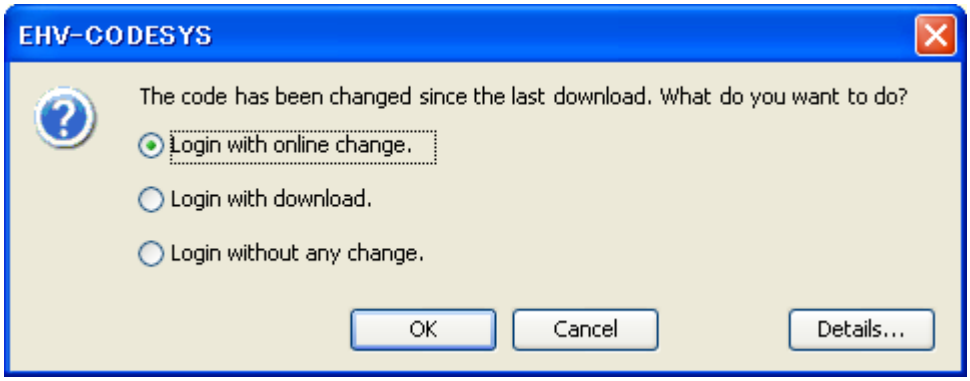
Choose the menu [Debug]-[Write values] or press [Ctrl + F7]. The prepared values will be transferred to PLC.



Online change

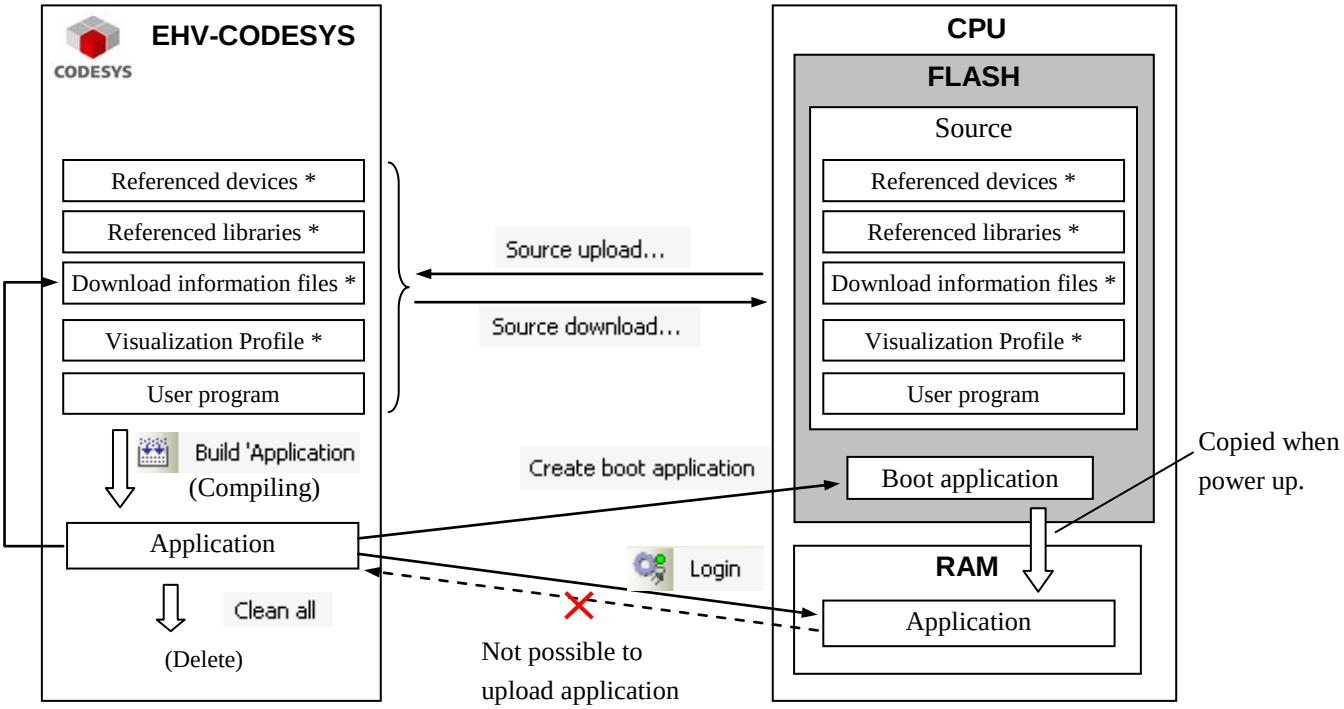
To change your program in running CPU (online change), you have to logout at first. After program changing, choose [Login] again. You will have 3 options as below.

Login with online change:	Only incremental program is downloaded without CPU stop.
Login with download:	Whole the program is downloaded. CPU is forced to stop.
Login without any change:	New program is not downloaded.



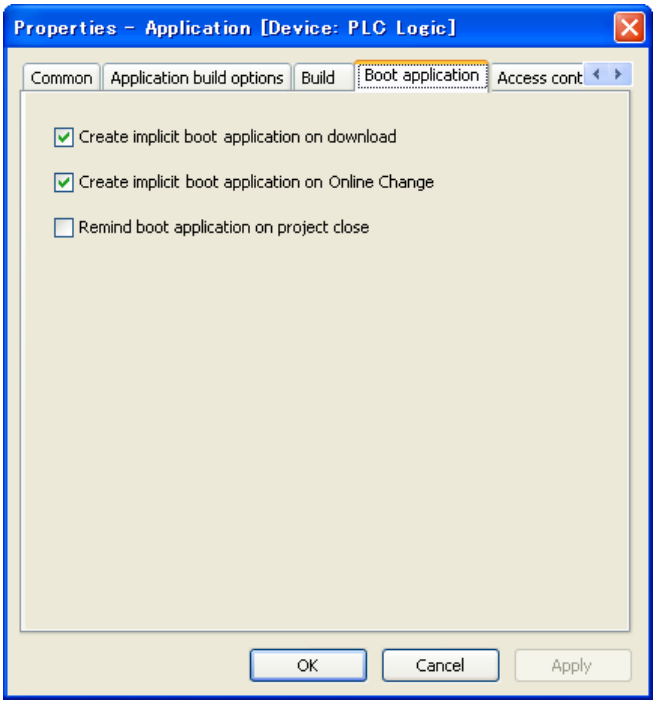
3.11 Boot application

The basic overview of downloading is shown as below picture. Be noted that an application (compiled user program) is downloaded to volatile RAM memory of the CPU, which means the application is lost when power is removed. If your application needs to be saved in non-volatile FLASH memory, choose [Create boot application] in Online menu while Login. When CPU is power up in the next time, the application is copied from FLASH to RAM and executed automatically if RUN/STOP switch is in RUN position.



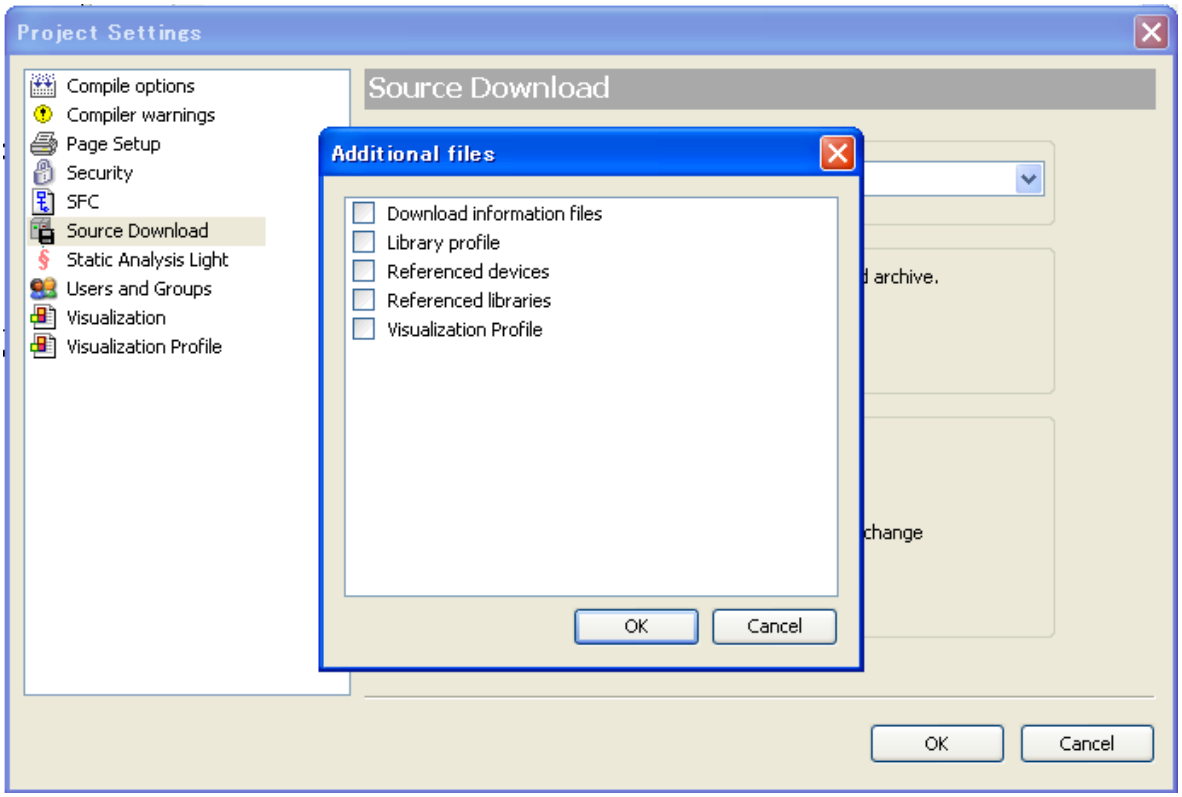
*: Optional

Timing to download boot application can be configured in [Properties] of [Application] (Right click on “Application” of the project tree). The default setting is shown below.



3.12 Source Download / Upload

Besides boot application, source file can be saved in the CPU, which enables you to upload original program file from PLC even if you don't have it in your PC. Some extra files can be added to source file as below. Choose according to your necessity.



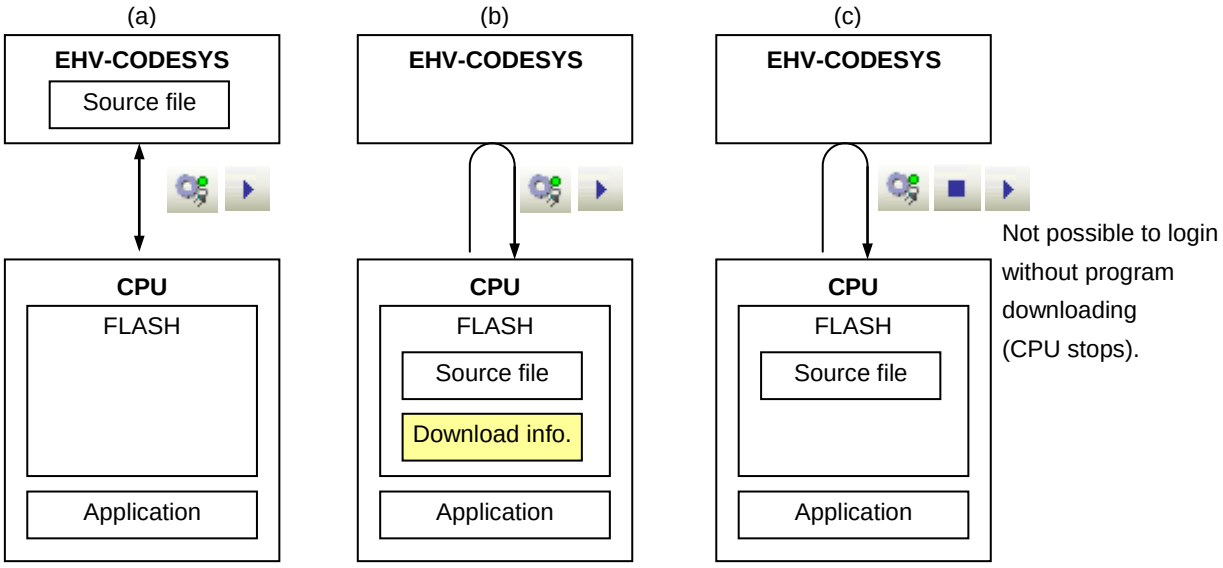
Download information files

"Download information files" in [additional files setting] is not necessary normally, but it is needed if you want to login without CPU stop from the PC which does not have original program file shown below as case (b) and (c).

(a) Online change from PC with source file to CPU without source file. ➔ Login

(b) Online change from PC without source file to CPU with source file and DL info. ➔ Source upload and Login

(c) Online change from PC without source file to CPU with source file. ➔ Source upload and Login, then program download is required because EHV-CODESYS is not able to verify program identity. It is possible to login after downloading, but CPU must stop at that time.



3.13 Run / Stop / Reset

Run/Stop

CPU can be started with EHV-CODESYS or Run/Stop switch on the CPU, but remote controlling with EHV-CODESYS is not allowed when the Run/stop switch is in Stop position as shown below.

Switch position / User operations	STOP	RUN
Stop with EHV-CODESYS 	Stop (no effect)	Stop
Run with EHV-CODESYS 	Stop (no effect)	Run
Reboot PLC (Cycle power)	Stop	Run *

* CPU starts running independent from the last status before power failure.

Reset

When CPU detects a serious error called “exception”, such as watchdog error, program execution stops. If EHV-CODESYS is connected, “Exception” indication blinks until this status is cleared. This exception status is cleared only by “Reset” operation. EHV-CODESYS has 3 different types of “Reset” operation: Reset warm, Reset cold and Reset origin. All of them can initialize exception status, but behaviors of CPU are different as shown below. Be noted that “Reset origin” initializes not only an exception but also your application and boot application in CPU module.

Operation	VAR	VAR RETAIN	VAR PERSISTENT	Application (in volatile memory)	Boot application (in non-volatile)
STOP	X	X	X	X	X
Reset warm	-	X	X	X	X
Reset cold	-	-	X	X	X
Reset origin	-	-	-	-	-
Download	-	-	X	(overwritten)	X
Online Change	X	X	X	(modified)	X
Reboot PLC	-	X	X	-	X

X = maintained, - = initialized

Stop switch definition

Definition of stop position of run/stop switch can be configured as “Stop” or “Reset warm” in CPU configuration. Default setting is “Reset warm” since it is almost same behavior of original “Stop” for existing Hitachi PLC.

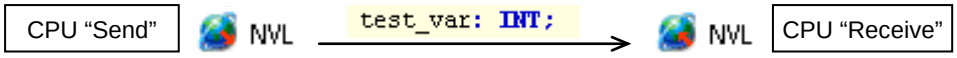
Parameter	Type	Value	Default Value
IP Address	STRING	'192.168.0.1'	'192.168.0.1'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'0.0.0.0'	'0.0.0.0'
Ethernet port Link speed / Duplex mode	Enumeration of BYTE	Auto Negotiation	Auto Negotiation
Change IP information	Enumeration of BYTE	No	No
Stop switch definition	Enumeration of BYTE	Reset warm	Reset warm
Digital Filter	BYTE(1..40)	4	4
Battery error detection	Enumeration of BYTE	Enable	Enable

3.14 Global network variables

Any variables can be listed in global network variable list, which are sent to all other CPUs in the network with broadcast address of UDP/IP. Global net work variable function is available only in professional setting. Refer to section 3.2 Start up how to change the environment setting.

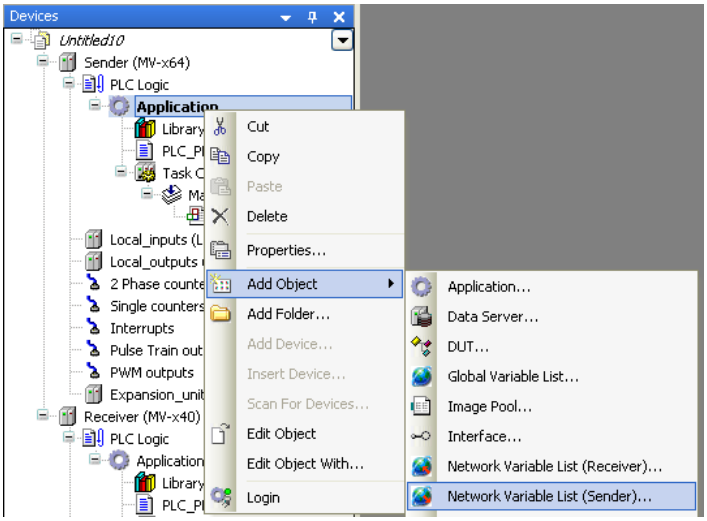
How to configure?

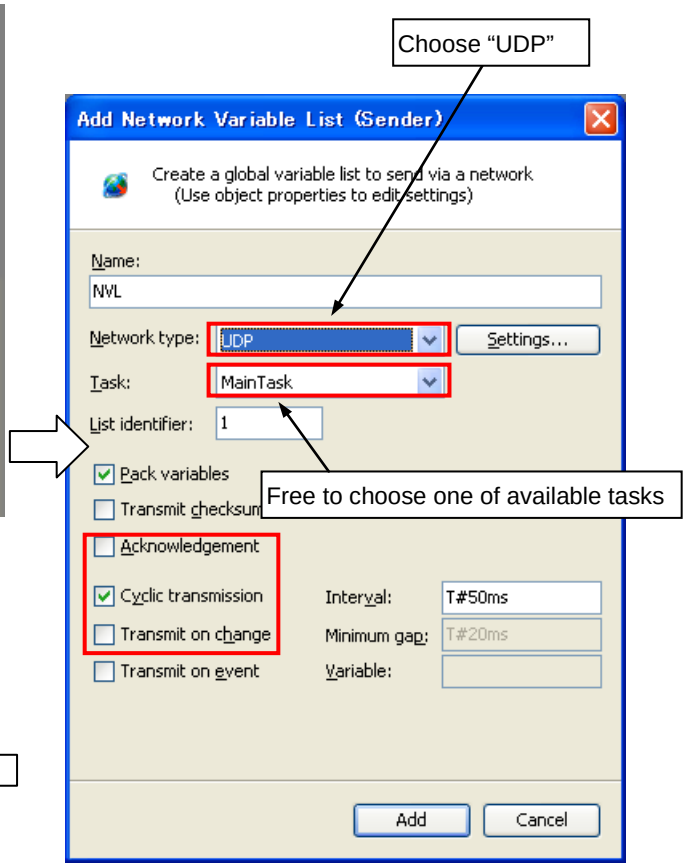
Procedure of configuration is shown below with a simple project: one CPU to send and the other CPU to receive.
Right click on the project and choose “Add Device” to add the 2nd CPU.

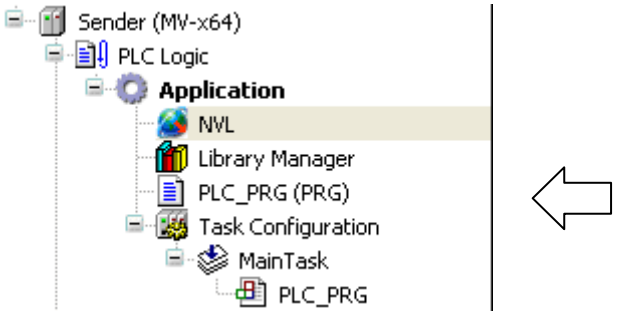


[CPU “Send”]

Right click on “Application” of send-CPU and choose “Network Variable List (Sender)...”.

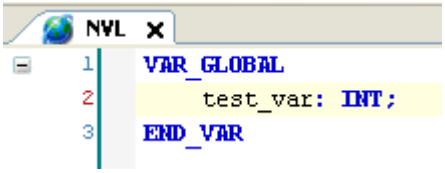


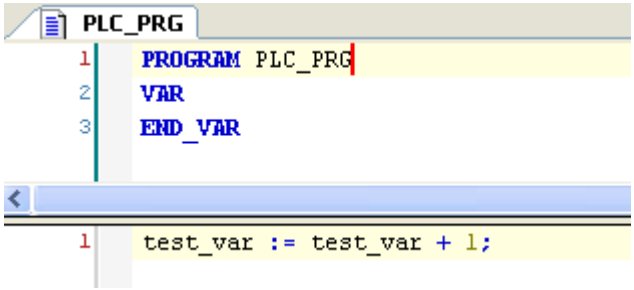




Network type: Choose “UDP”.
Task: Choose any one task. The variables are sent at the end of a task cycle.
List identifier: If more than 2 global variable list is configured, set a number in ascending order.
Cyclic transmission: Since variables are sent every task cycle, set interval time as same or bigger than cycle time of configured task. If smaller time than task cycle is set, actual sending cycle is limited by task cycle.
Transmit on change: Variables are sent only if their values have changed; the Minimum gap can define a minimum time lapse between transfers.
Transmit on event: Variables are sent while specified variable is TRUE. Be noted that it is not edge detection but level detection.
Refer to online help of EHV-CODESYS for further information.

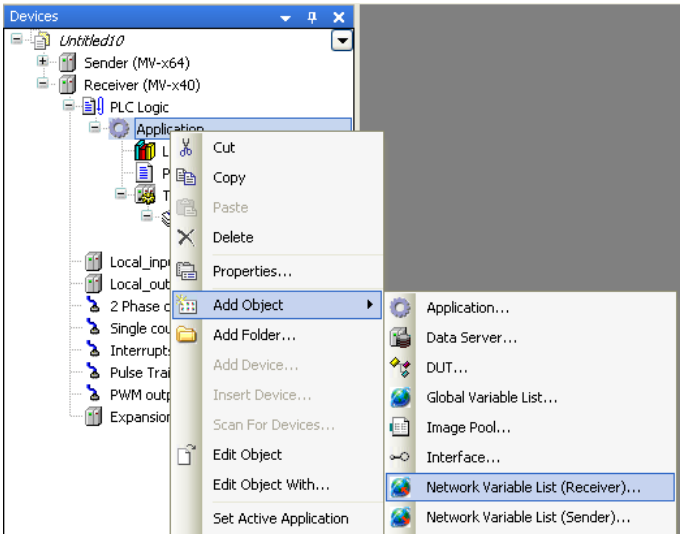
In this sample, one global variable “test_var” is defined and one-line program is written in POU as below.

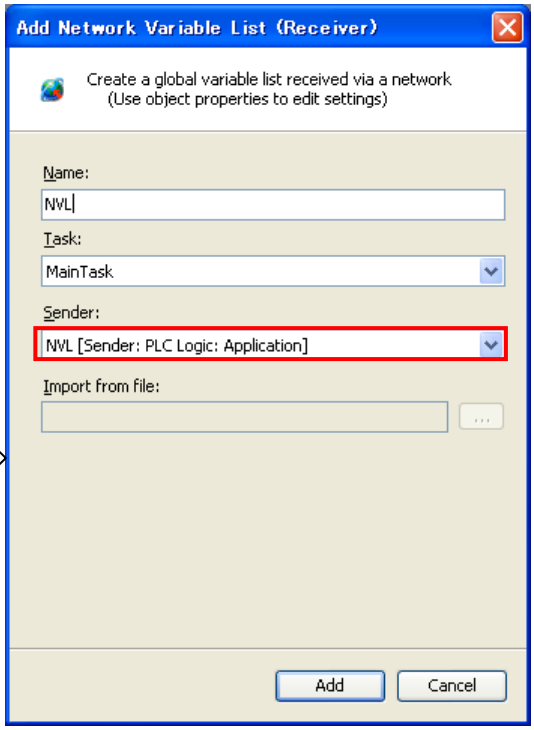


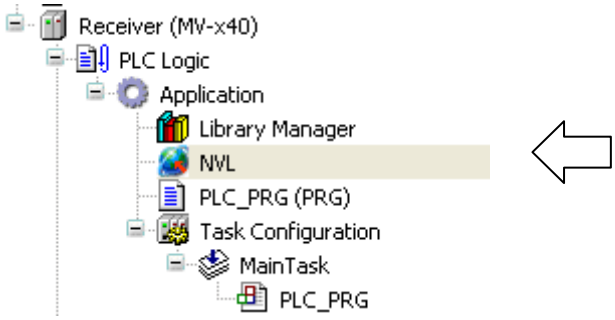


[CPU “Receive”]

The next step is configuration for receiving CPU. Right click on “Application” of Receive-CPU and choose “Network Variable List (Receiver)...” Be sure to check if Sender is properly set as configured list above.



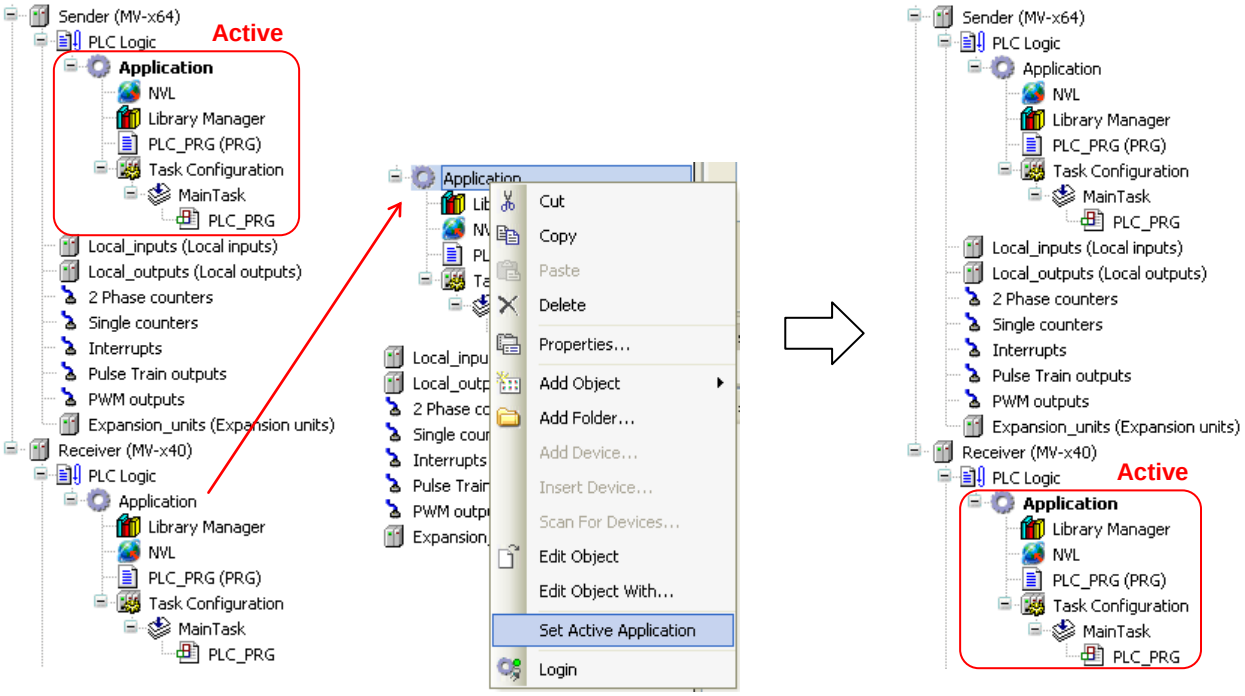




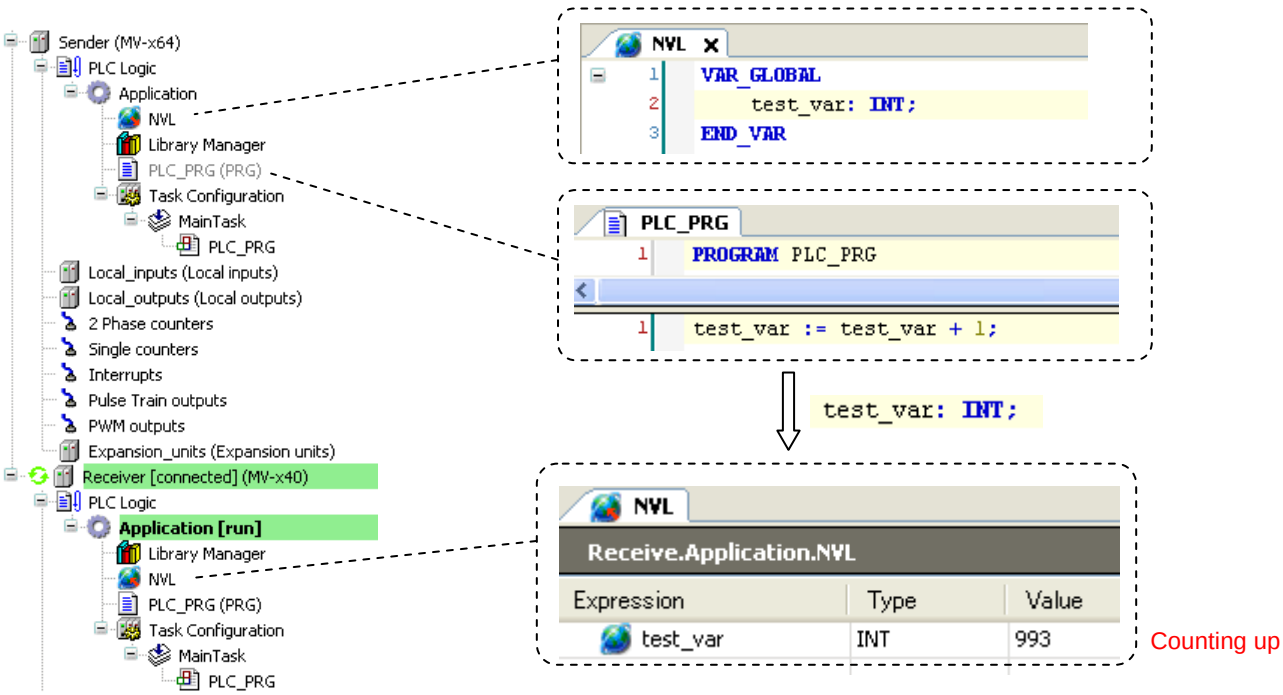
Configuration is completed for both send and receive-CPU.

Login

Set the communication path for Send-CPU and login (download application).
After logout, right click on “Application” of Receive-CPU and choose “Set Active Application”.
Set the communication path for Receive-CPU and login (download application).

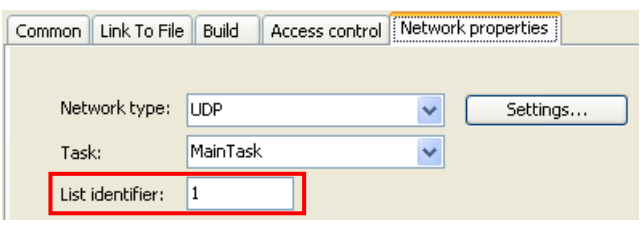


You can see the variable “test_var” is counting up in the global network variable list in the Receive-CPU.



Note

- If any parameters of global variable list is changed, be sure to execute “Clean” or “Clean All” before login.
- If more than 2 global variable lists are configured, be sure to set another “List identifier” in ascending order.



3.15 Modbus-TCP/RTU

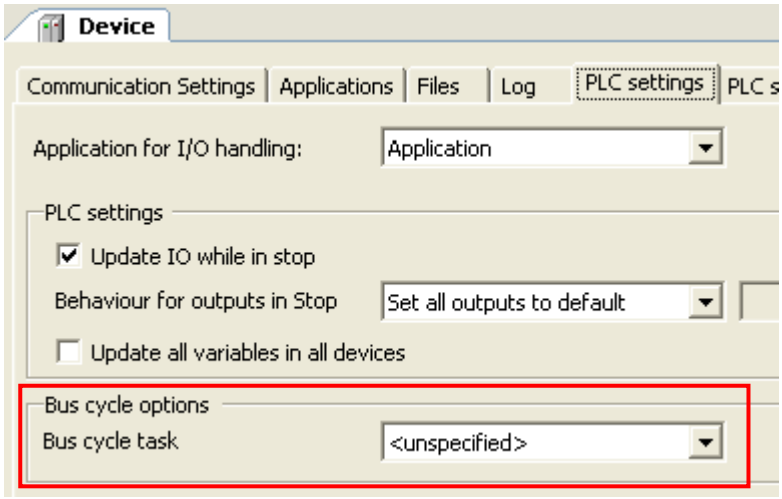
3.15.1 Introduction

Supported function codes are shown in the below table.

16#	10#	Function code	Modbus-RTU Master	Modbus-RTU Slave Modbus-TCP Slave
0x01	01	Read Coils	X	-
0x02	02	Read Discrete Inputs	X	-
0x03	03	Read Holding Registers	X	X
0x04	04	Read Input Registers	X	X
0x05	05	Write Single Coil	X	-
0x06	06	Write Single Register	X	X
0x0F	15	Write Multiple Coils	X	-
0x10	16	Write Multiple Registers	X	X
0x17	23	Read/Write Multiple Registers	X	X

X = Supported, - = Not supported

Modbus command processing is executed in bus cycle tack, which is configured in PLC settings of Device as below. You can specify any existing IEC tasks. If the bus cycle task is <unspecified>, task with the shortest cycle time is taken.

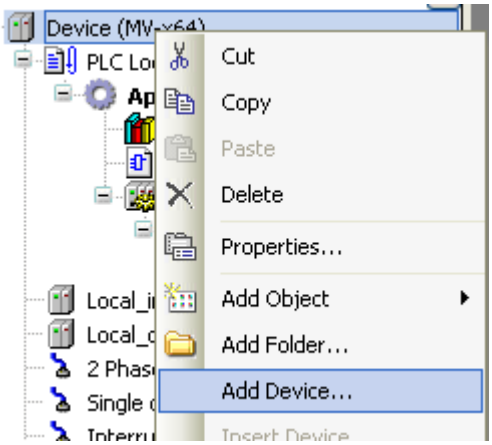


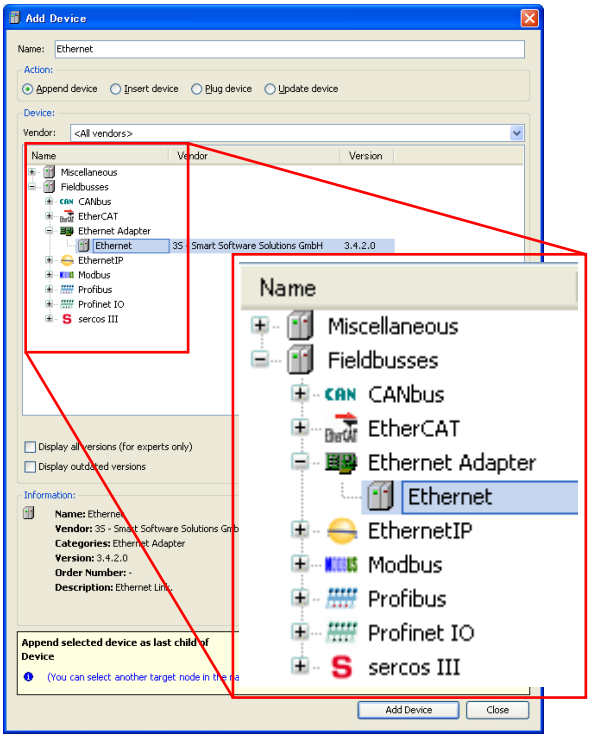
Note

When using Modbus-TCP slave or Modbus-RTU slave, be sure to send correct messages from master according to the Modbus specifications, otherwise slave could fail to respond correctly.

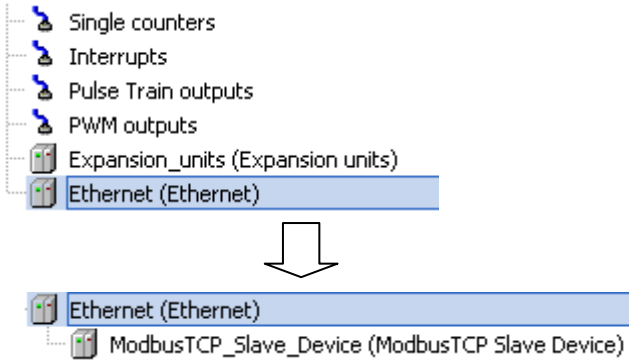
3.15.2 Modbus-TCP slave (server)

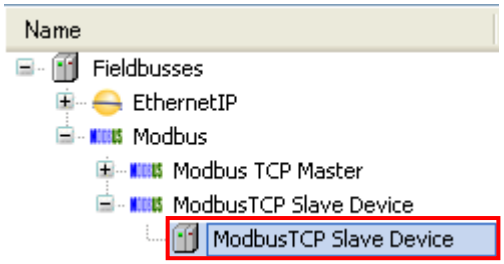
Right click on “Device” and choose “Add Device...”.
“Add Device” window appears.
Click “Ethernet” and [Add Device] button.



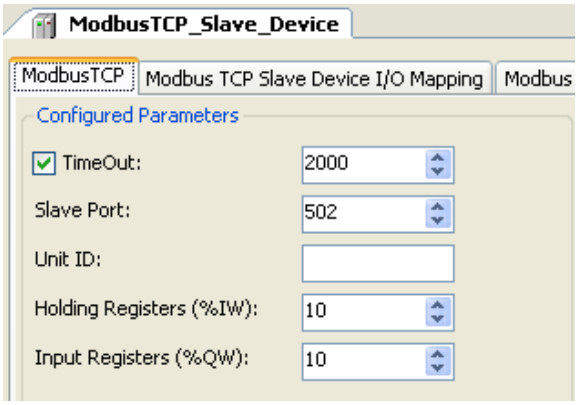


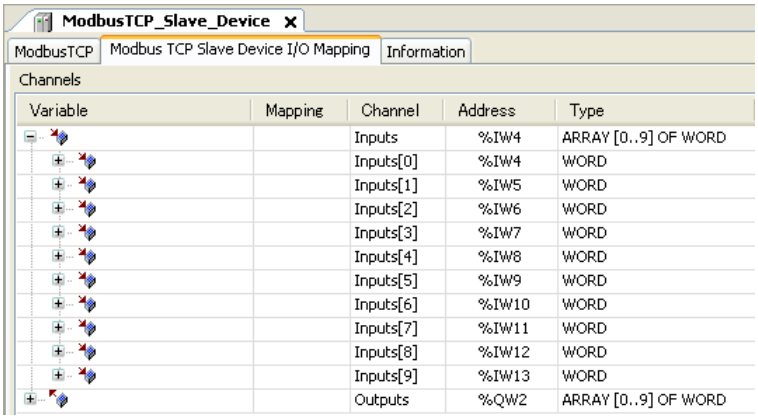
Right click on “Ethernet” and choose “Add Device...”. Click “Modbus TCP Slave Device” in the “Add Device” window and [Add Device] button.





Configure each parameter as below. According to the size of “Holding Registers” and “Input Registers”, data area will be assigned as seen in “Modbus TCP Slave Device I/O Mapping” tab.



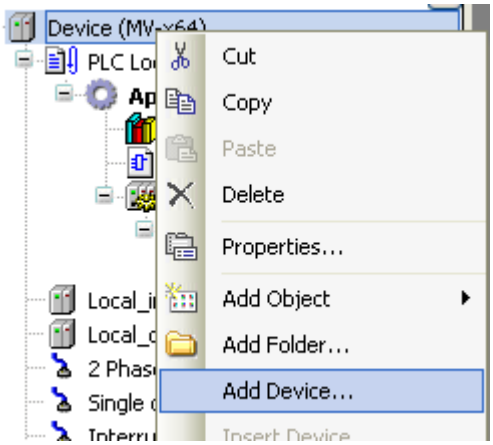


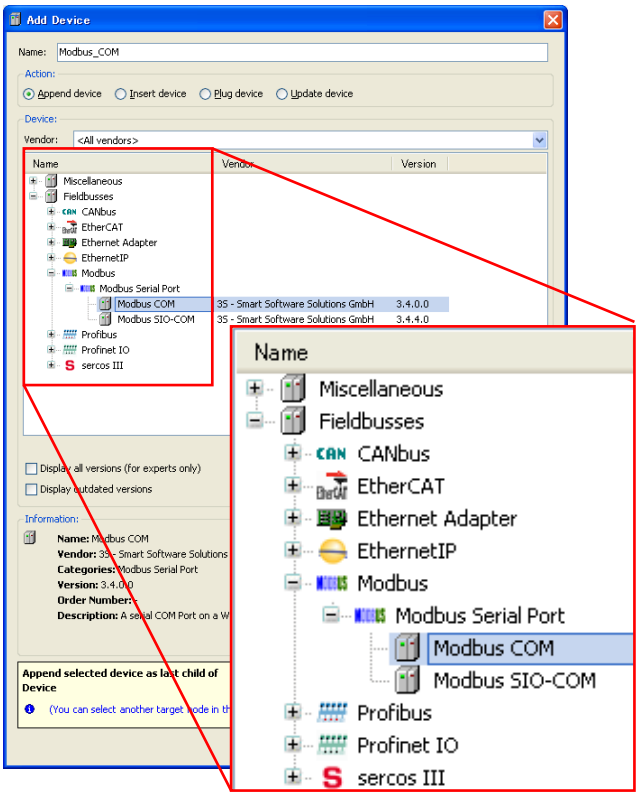
Note

It is necessary to supply the power to MICRO-EHV+ before establishing a TCP connection from master devices.

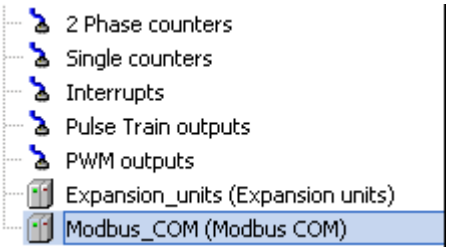
3.15.3 Modbus-RTU master

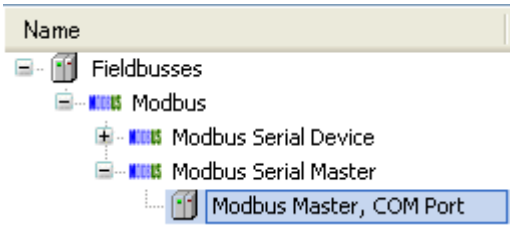
Right click on “Device” and choose “Add Device...”.
“Add Device” window appears.
Click “Modbus COM” and [Add Device] button.



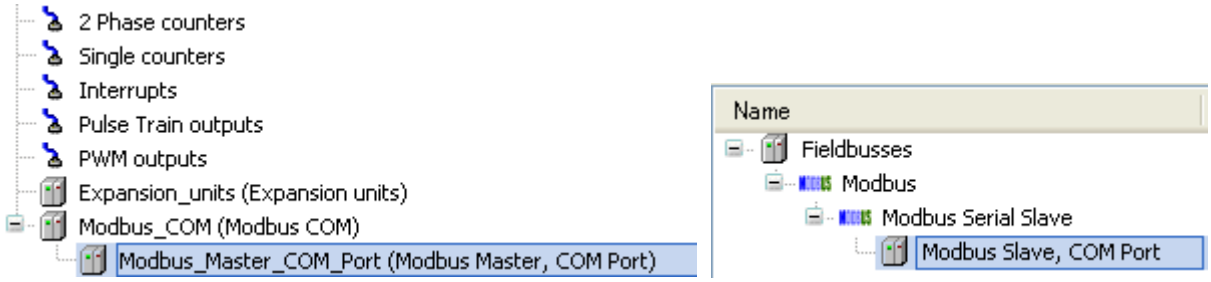


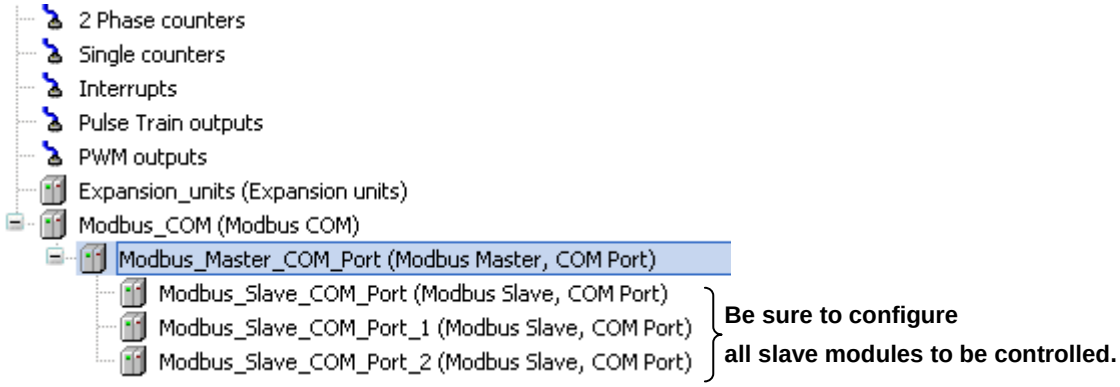
With “Add device” window opened, click “Modbus_COM” in the device tree. Then available devices will be shown in the “Add Device” window. Choose “Modbus Master, COM Port” and [Add Device] button.



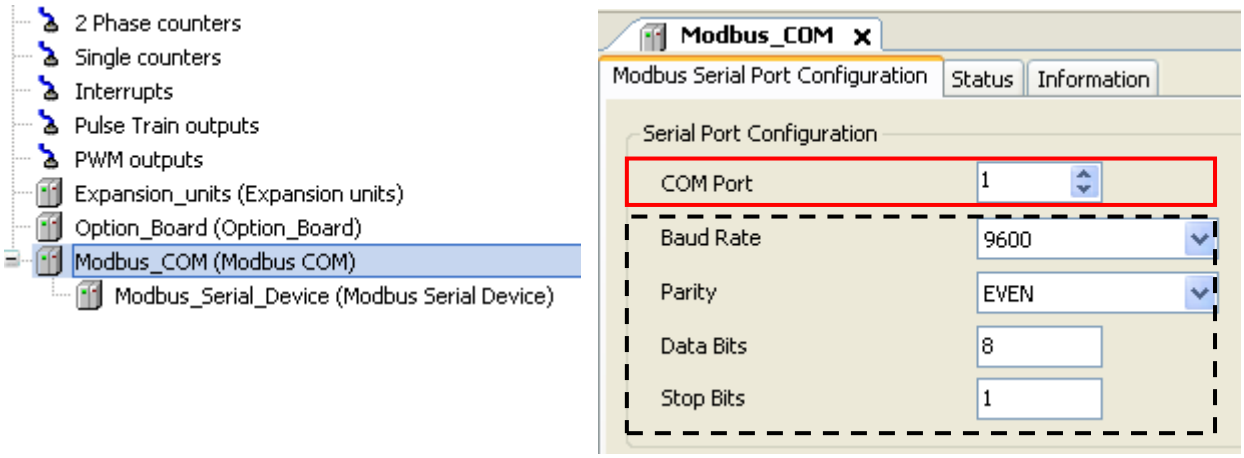


With “Add device” window opened, click “Modbus_Master_COM_Port” in the device tree. Then “Modbus Slave, COM Port” is shown in the “Add Device” window. Click “Modbus Slave, COM Port” and [Add Device] button according to your Modbus system configuration. e.g. if 3 slaves are to be controlled, add 3 times of slave devices.

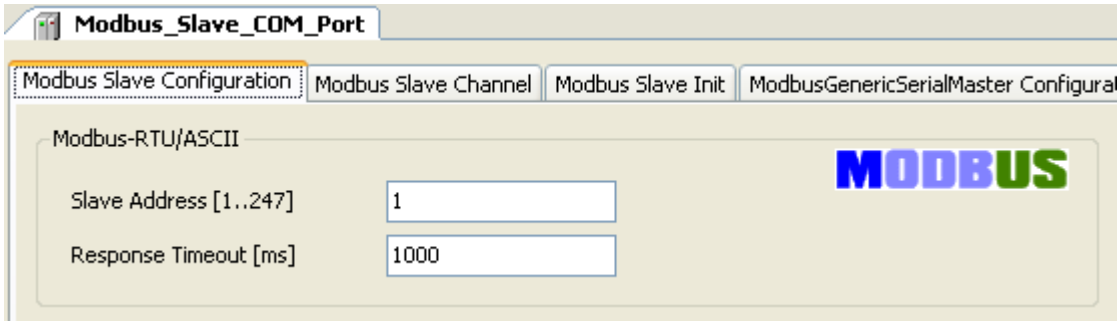




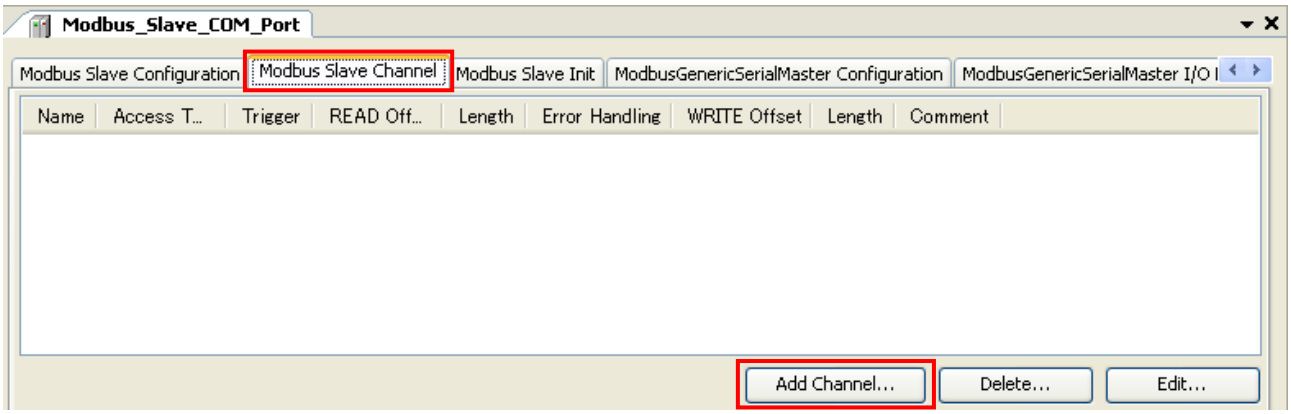
Double click on “Modbus_COM” or right click and choose “Edit Object”. Modbus_COM Configuration window appears. Set 1 (body) or 2 (option board) as COM port number. Set parameters in this configuration window.



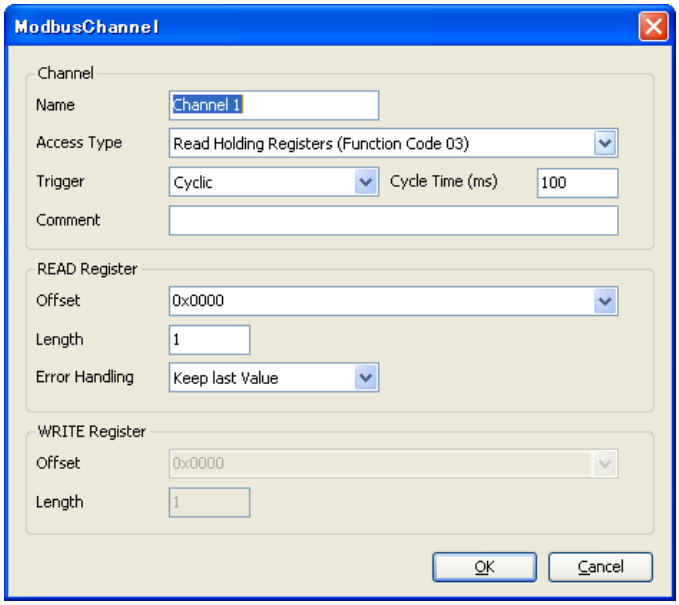
Function codes to be sent are configured in each slave. Double click a slave unit to open configuration window. Set slave address and response timeout.



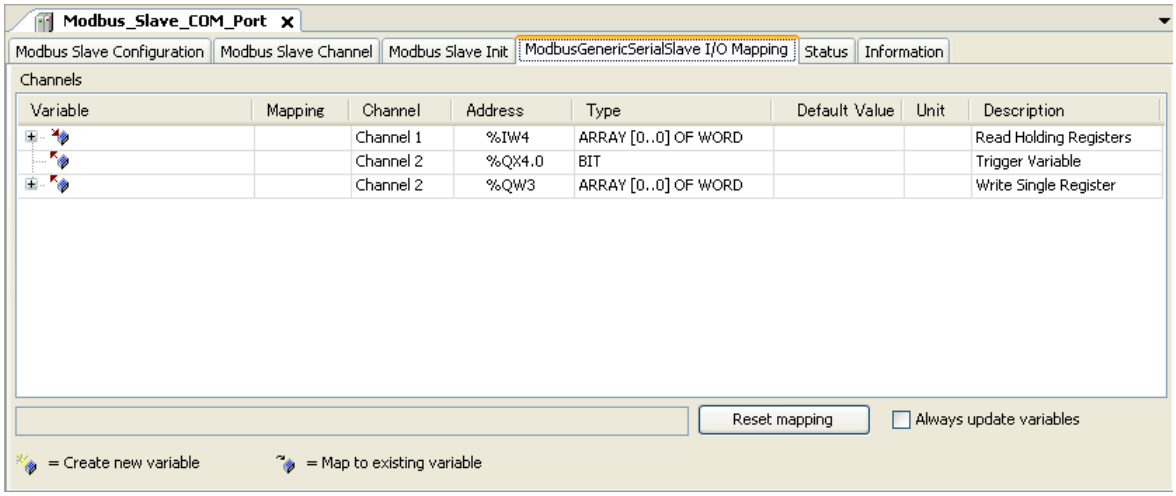
Open “Modbus Slave Channel” tab and click “Add Channel...” to add function codes.



Configure each parameter as below. If the Trigger setting is “Rising edge”, trigger variable (BOOL) will be automatically assigned in %QX address.

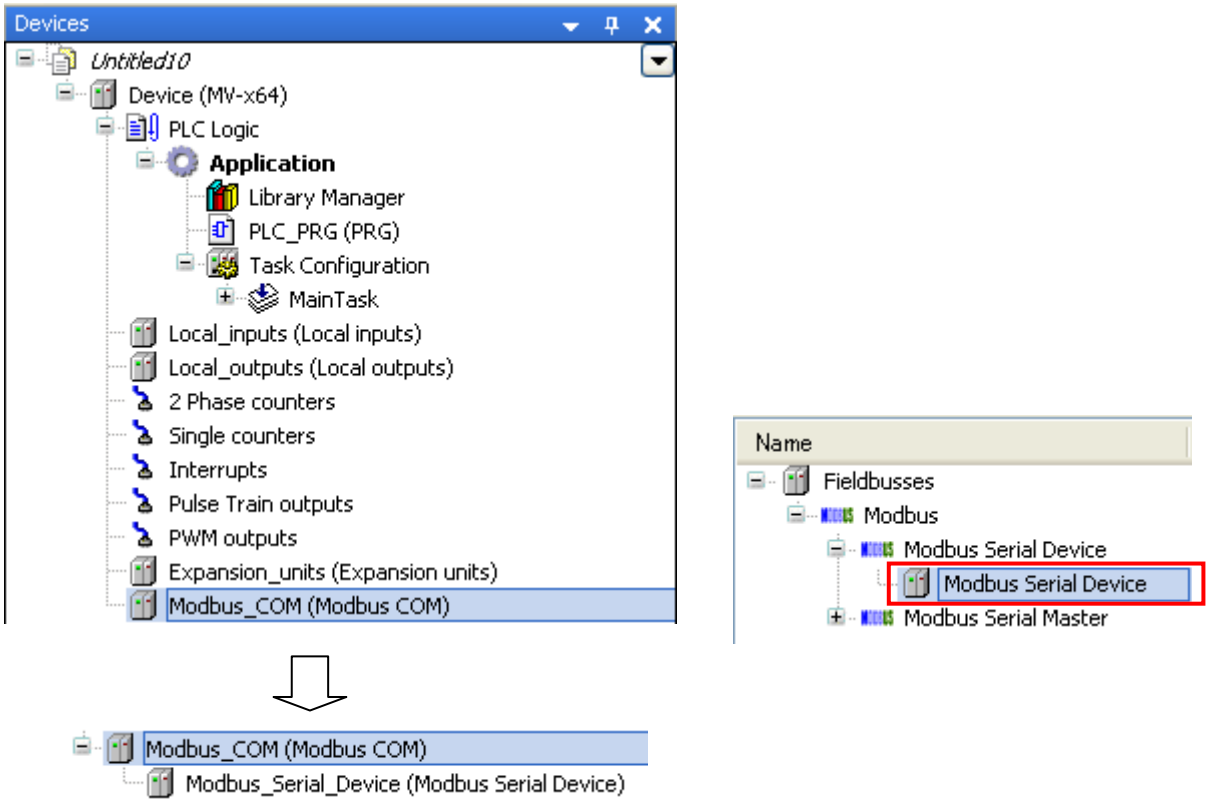


Data of Modbus will be assigned to %IW or %QW as seen in “ModbusGenericSerialMaster I/O Mapping” tab. Read data from slave is assigned to input area (%IW) and data to be written to slave is assigned to output area (%QW)

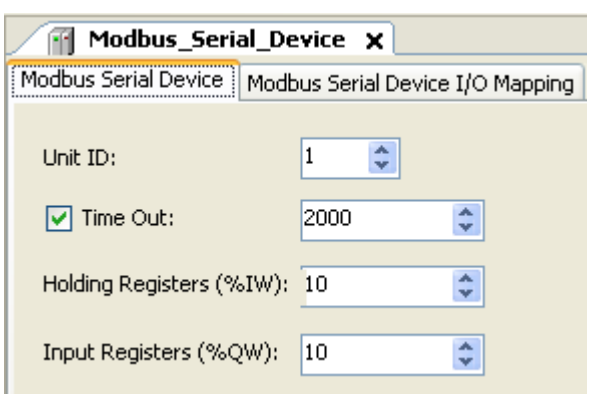


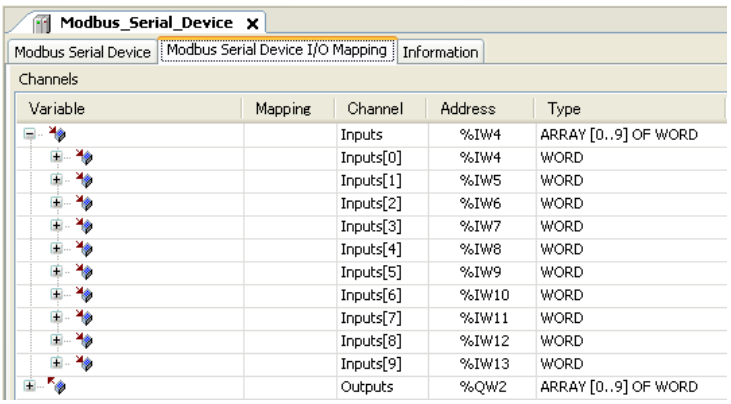
3.15.4 Modbus-RTU slave

Add “Modbus_COM” same as the setting of Modbus-RTU master. Right click on “Modbus_COM” and choose “Add Device...”. Click “Modbus Serial Device” in the “Add Device” window and [Add Device] button.



Configure each parameter as below. According to the size of “Holding Registers” and “Input Registers”, data area will be assigned as seen in “Modbus Serial Device I/O Mapping” tab.





3.16 High speed counter

DC inputs of MICRO-EHV+ are configurable for 32-bit high speed counters.
Each channel has two preset values set by special function block. If the counter value exceeds the preset value, then event task will be executed immediately, which leads POU (FUNCTION) under the task called accordingly. Desired program can be written in this FUNCTION.

Available I/O addresses of high speed counter and interrupt input are listed as below.

	bit 0 %IX7.0	bit 1 %IX7.1	bit 2 %IX7.2	bit 3 %IX7.3	bit 4 %IX7.4	bit 5 %IX7.5	bit 6 %IX7.6	bit 7 %IX7.7	bit 8 %IX6.0	bit 9 %IX6.1
Single phase	ch.1		ch.2		ch.3		ch.4		ch.5	
2-phase	ch.1-A		ch.1-B	ch.1-Z	ch.3-A		ch.3-B	ch.3-Z		
Interrupt		ch.1		ch.2		ch.3		ch.4		ch.5

Since the same input addresses are shared by single and 2-phase counters, available combinations of 2-phase and single counter are shown as below.

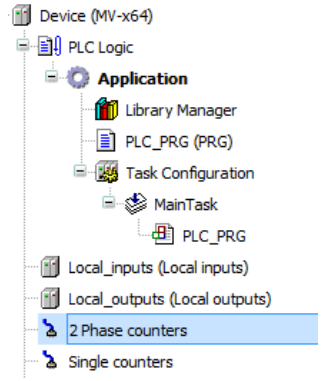
2-phase counter	Single counter	Interrupt input
2 (ch.1, 3)	1 (ch.5)	5 (ch.1,2,3,4,5) [3 (ch.1,3,5)]*
1 (ch.1)	3 (ch.3,4,5)	5 (ch.1,2,3,4,5) [4 (ch.1,2,3,5)]*
1 (ch.3)	3 (ch.1,2,5)	5 (ch.1,2,3,4,5) [4 (ch.1,3,4,5)]*
0	5 (ch.1,2,3,4,5)	5 (ch.1,2,3,4,5)

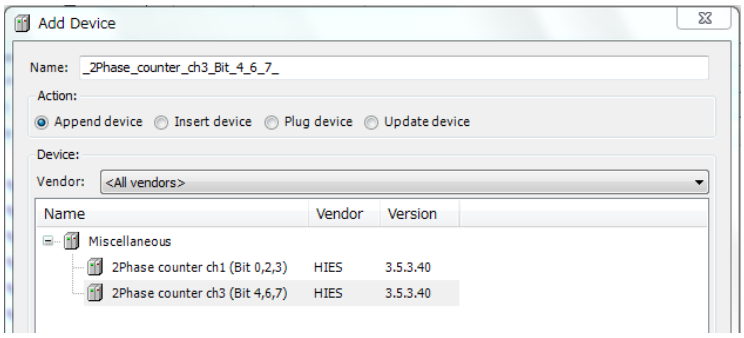
* In case marker input is enabled

Configuration of 2-phase counter

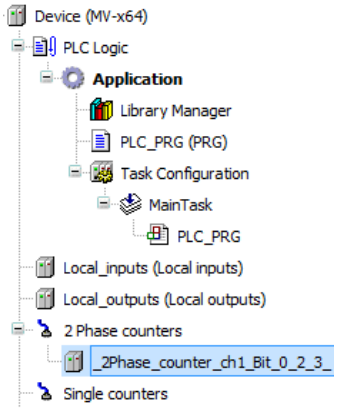
Right click on “2-phase counters” and choose “Add Device…”.

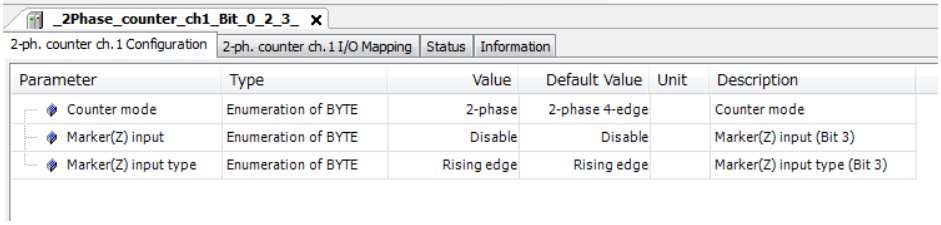
Then “Add Device” window appears. Click 2Phase counter ch1 or ch3 and [Add Device] button.

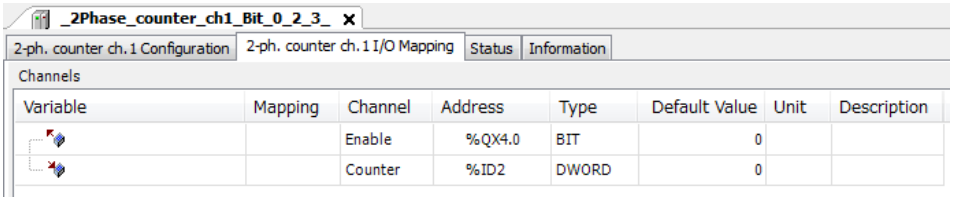




Double click on “_2Phase_counter_chx…” or right click and choose “Edit Object”. 2-phase counter configuration window appears. 2-phase counter has Configuration tab and Mapping tab as below.







Configuration

Parameter	Descriptions	Choices	Default
Counter mode	4 different counter modes are supported as shown below.	2-phase 4-edge	2-phase 4-edge
		2-phase	
		Pulse + Direction	
		CW/CCW	
Marker (Z) input	When Marker (Z) input is enabled and rising/falling edge of the input is detected, the counter value is reset (0).	Disable	Disable
		Enable	
Marker (Z) input type		Rising edge	Rising edge
		Falling edge	

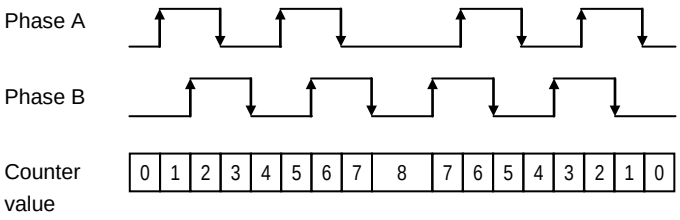
Mapping

Name	I/O	Type	Descriptions
Enable	Output (%QX)	BOOL	TRUE: Counter is enabled. FALSE: Counter is disabled.
Counter	Input (%ID)	DWORD	Counted value is stored in this register. 0 to 4,294,967,295

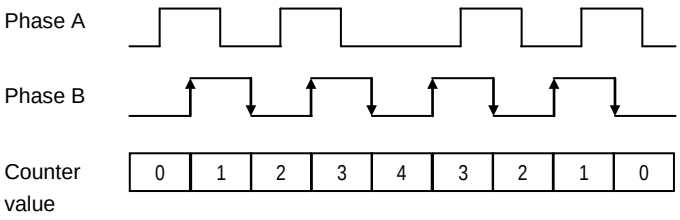
When the counter value exceeds the maximum or minimum value, it returns to 0 or 4,294,967,295.

Counter mode

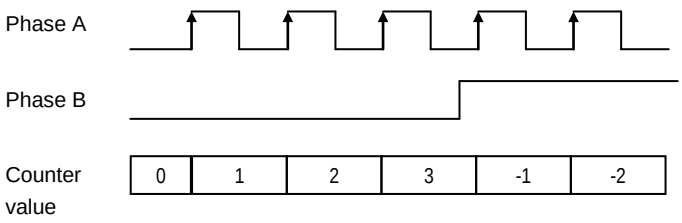
2-Phase 4-edge



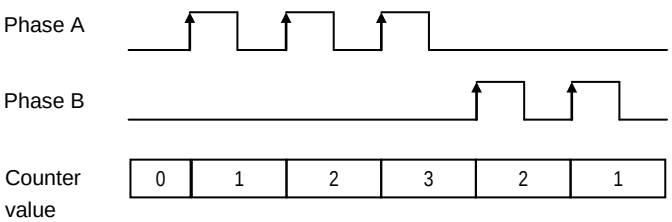
2-Phase



Pulse + Direction



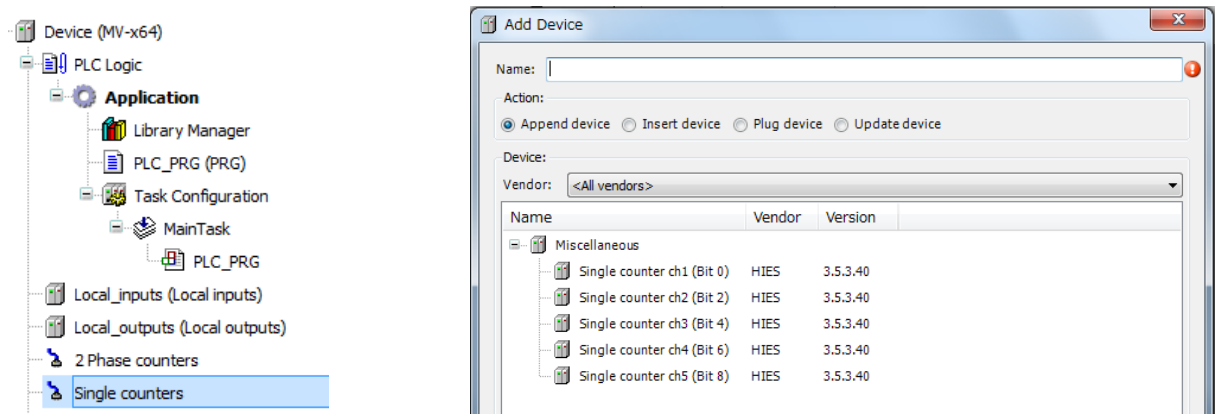
CW/CCW



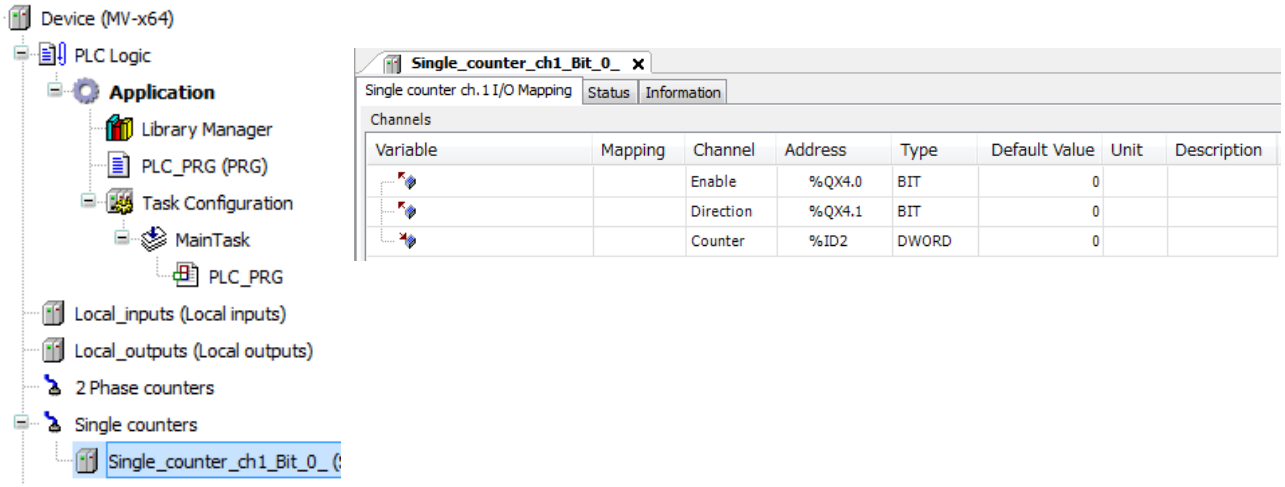
Configuration of single counter

Right click on “Single counters” and choose “Add Device…”.

Then “Add Device” window appears. Click Single counter chx and [Add Device] button.



Double click on “Single_counter_chx…” or right click and choose “Edit Object”. Single counter Mapping window appears. Single counter does not have Configuration tab, which is different from 2-phase counters.

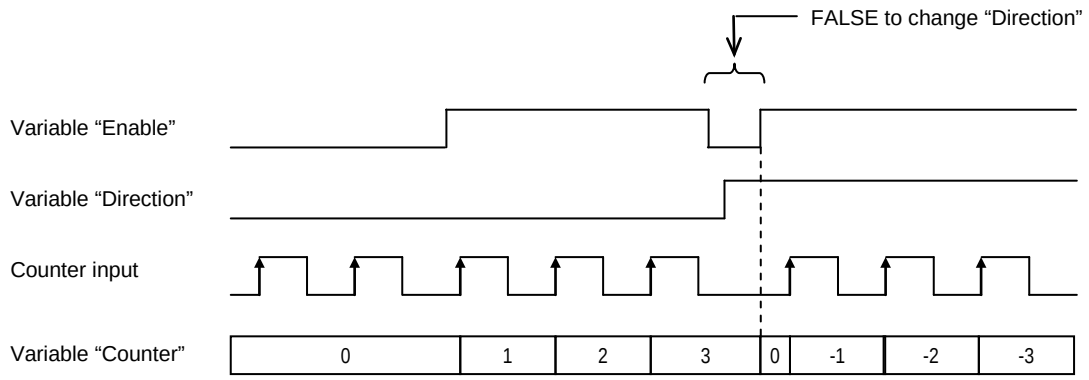


Mapping

Name	I/O	Type	Descriptions
Enable	Output (%QX)	BOOL	TRUE: Counter is enabled. FALSE: Counter is disabled.
Direction	Output (%QX)	BOOL	TRUE: Down counter FALSE: Up counter
Counter	Input (%ID)	DWORD	Counter value is increased or decreased with every rising edge and stored in this register. (0 to 4,294,967,295)

Direction can be changed only when counter is disabled.

When the counter value exceeds the maximum or minimum value, it returns to 0 or 4,294,967,295.



Operations

Counter can be controlled by variables and function blocks as below.

Operations	Variable or FB	Name
Start/Stop counter	Variable	Enable (%QX)
Read counter value	Variable	Counter (%ID)
Write current counter value	Function block	MV_CU_Write
Write preset values	Function block	MV_CU_preset

Start/Stop

Counter is started or stopped by the variable (%QX) configured in “Enable” in configuration window. While this variable is TRUE, counter is enabled. No function block is required to start or stop.

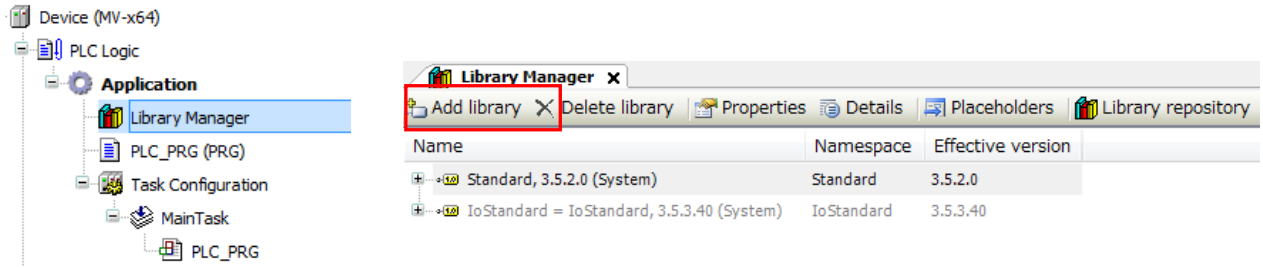
Read counter value

Counter value is stored automatically in the variable (%ID) configured in “Counter” in configuration window. No function block is required to read counter value.

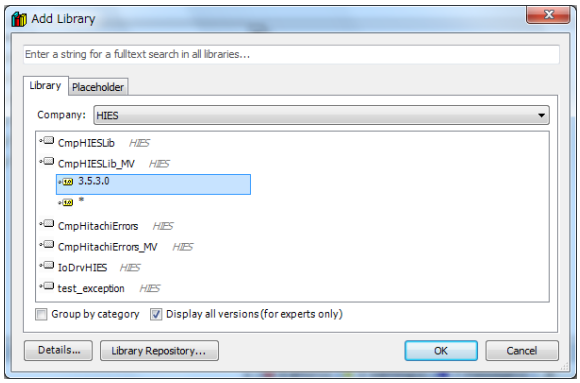
Write counter value / preset values

Dedicated library must be added to the project to write counter value and preset values.

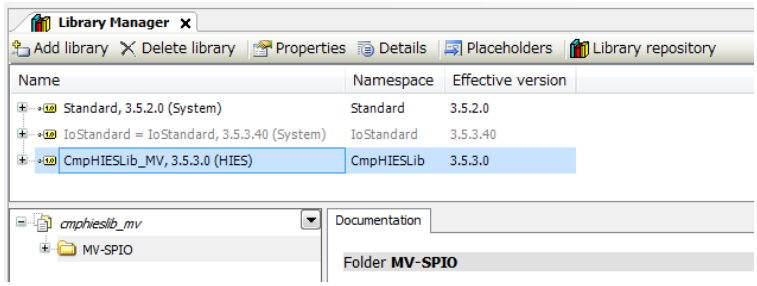
Double click on “Library Manager” or right click and choose “Edit Object”. Library Manager window appears.



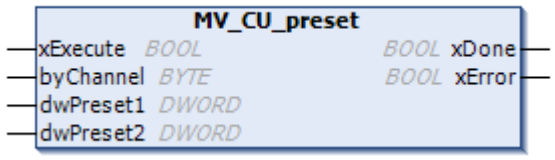
Click “Add library” and choose “CmpHIESLib_MV”.



CmpHIESLib_MV library is added in the project.

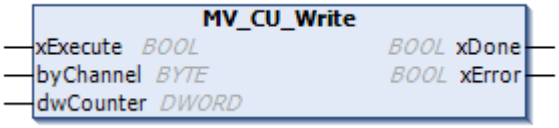


MV_CU_preset



FB name	Description		
MV_CU_preset	This function block is to set preset value 1 and 2 for specified channel of counter		
Name	**Type**	**I/O**	**Descriptions**
xExecute	BOOL	IN	A rising edge starts the execution.
byChannel	BYTE	IN	Channel number (2-phase counter: 1or 3, Single counter: 1 to 5)
dwPreset1	DWORD	IN	0 to 4,294,967,295
dwPreset2	DWORD	IN	0 to 4,294,967,295
xDone	BOOL	OUT	Execution done successfully
xError	BOOL	OUT	Error occurred

MV_CU_Write

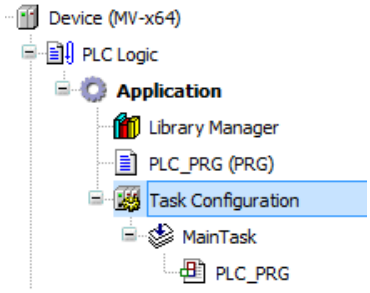


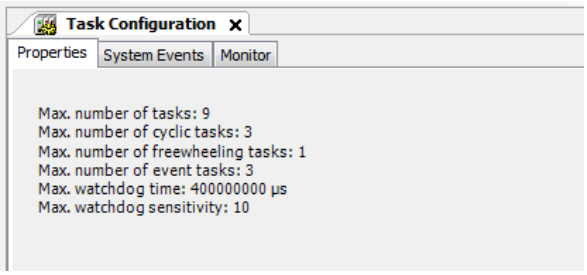
FB name	Description		
MV_CU_Write	This function block is to overwrite current counter value for specified channel of counter		
Name	**Type**	**I/O**	**Descriptions**
xExecute	BOOL	IN	A rising edge starts the execution.
byChannel	BYTE	IN	Channel number (2-phase counter: 1or 3, Single counter: 1 to 5)
dwCounter	DWORD	IN	0 to 4,294,967,295
xDone	BOOL	OUT	Execution done successfully
xError	BOOL	OUT	Error occurred

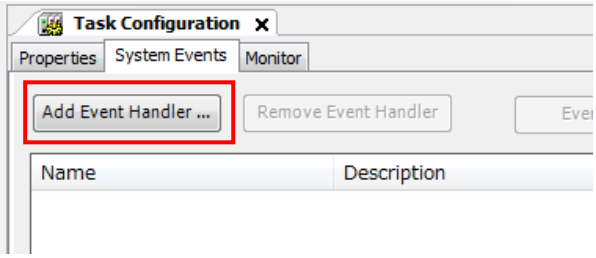
Preset value

If the counter value exceeds the preset value 1 or 2, then corresponding event task will be executed immediately, which leads POU (FUNCTION) under the task called accordingly. Desired program can be written in this FUNCTION. Event task is configured as follows.

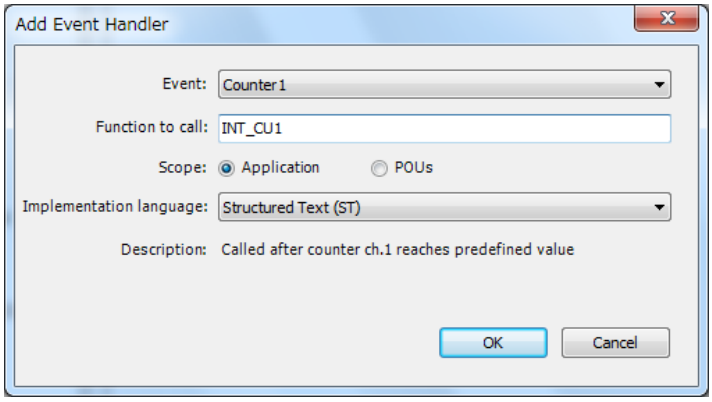
Double click on “Task Configuration” or right click and choose “Edit Object”. Task Configuration window appears. Click “System Events” tab.

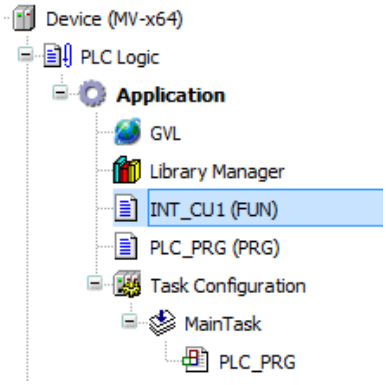






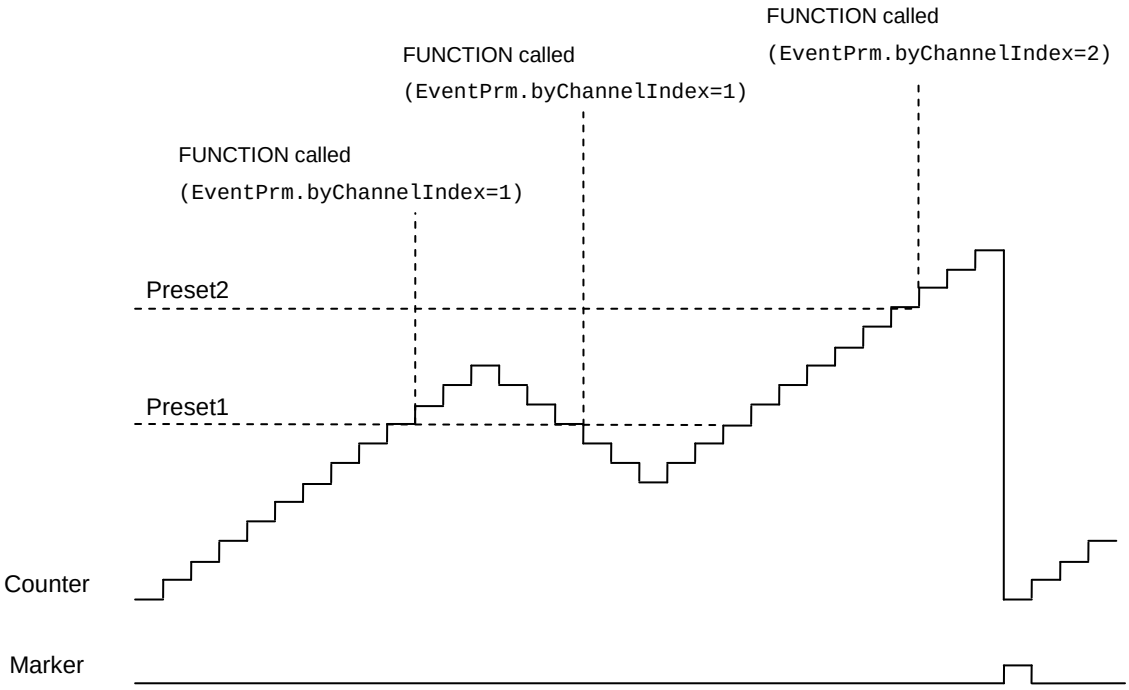
Click “Add Event Handler…” then ADD Event Handler window appears. Choose a counter event from Counter 1 to 5 in “Event” and enter a name in “Function to call”. Click OK button, then FUNCTION will be created.





When new FUNCTION is created, a variable “EventPrm” (type:STRUCT EVTPARAM_Counterset) is automatically defined in the FUNCTION. This variable consists of one BYTE variable named “byChannelIndex” indicating which preset value is exceeded.

(Library IoDrvHIES is implicitly added to Library Manager, and this STRUCT is defined in it.)



If counter value exceeds preset value 1 or 2, then program execution is interrupted and corresponding FUNCTION is called. After that, program execution returns back to the original position.
Each counter has two different preset values, but the same FUNCTION is called. This can be identified by the variable “EventPrm.byChannelIndex” in the FUNCTION. If counter value exceeds preset 1, then 1 is stored in EventPrm.byChannelIndex.

Functions contain no internal state information. Use global variables including local I/O (external I/O) in functions. Local I/O (external I/O) used only in functions is not updated. Make sure to use in other programs or function blocks also.

Sample program

INT_CU1

```
FUNCTION INT_CU1 : DWORD
VAR_IN_OUT
    EventPrm: IoDrvHIES.EVTPARAM_CounterSet;
END_VAR
VAR
END_VAR
```

```
IF EventPrm.byChannelIndex=1 THEN
    test1:=test1+1; // Global variable
    alarm1:=TRUE;  // Local output (Digital output bit1)
    ELSE
    test2:=test2+1; // Global variable
    alarm2:=TRUE;  // Local output (Digital output bit2)
END_IF
```

3.17 Interrupt input

DC inputs of MICRO-EHV+ are configurable for interrupt inputs.
If rising or falling edge is detected, then event task will be executed immediately, which leads POU (FUNCTION) under the task called accordingly. Desired program can be written in this FUNCTION.

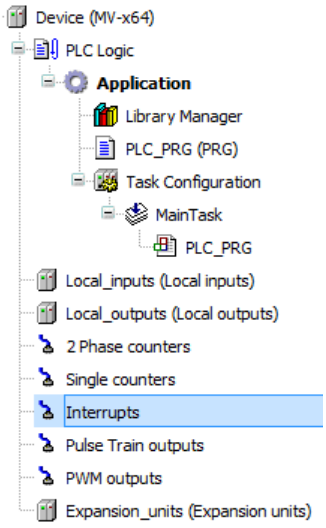
Available I/O addresses of high speed counter and interrupt input are listed as below.

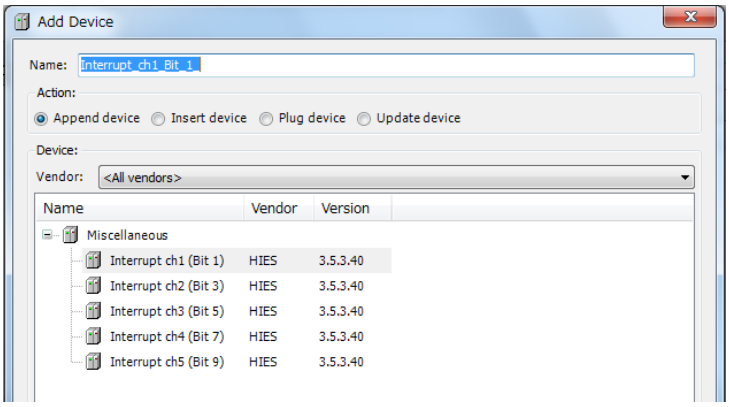
	bit 0 %IX7.0	bit 1 %IX7.1	bit 2 %IX7.2	bit 3 %IX7.3	bit 4 %IX7.4	bit 5 %IX7.5	bit 6 %IX7.6	bit 7 %IX7.7	bit 8 %IX6.0	bit 9 %IX6.1
Single phase	ch.1		ch.2		ch.3		ch.4		ch.5	
2-phase	ch.1-A		ch.1-B	ch.1-Z	ch.3-A		ch.3-B	ch.3-Z		
Interrupt		ch.1		ch.2		ch.3		ch.4		ch.5

Configuration

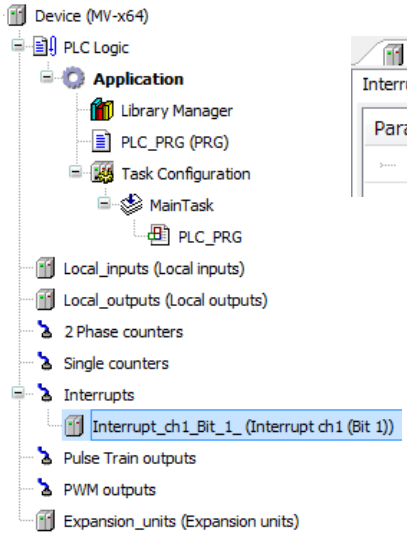
Right click on “Interrupts” and choose “Add Device…”.

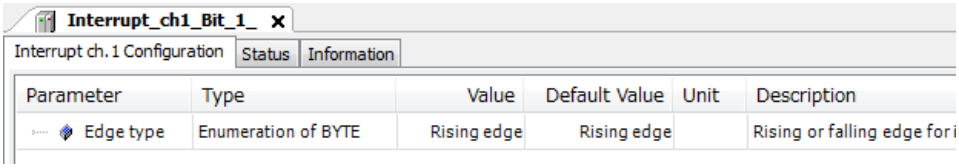
Then “Add Device” window appears. Click Interrupt chx and [Add Device] button.





Double click on “Interrupt_chx…” or right click and choose “Edit Object”. Interrupt configuration window appears.





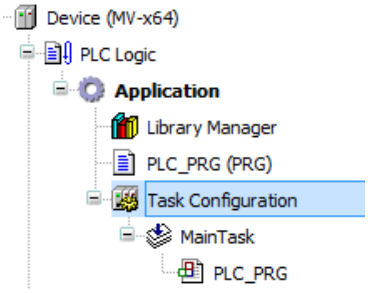
Configuration

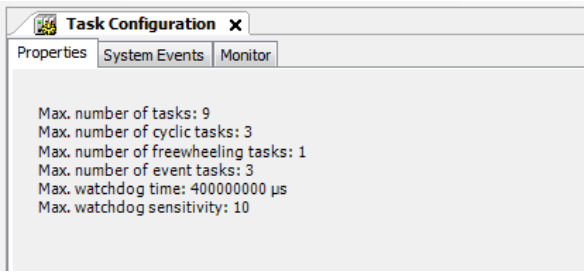
Parameter	Descriptions	Choices	Default
Edge type	Rising edge is transition from low to high.	Rising edge	Rising edge
	Falling edge is transition from high to low.	Falling edge	

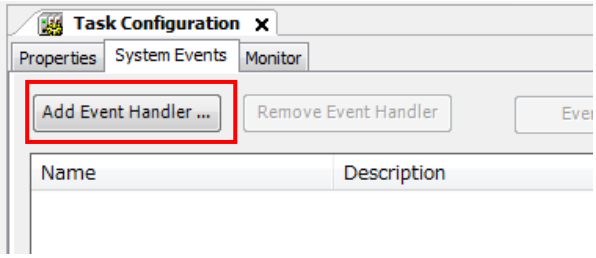
Event task

If rising or falling edge is detected, then corresponding event task will be executed immediately, which leads POU (FUNCTION) under the task called accordingly. Desired program can be written in this FUNCTION. Event task is configured as follows.

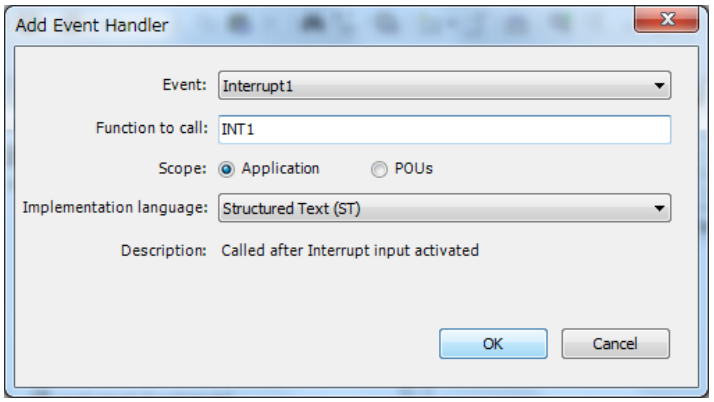
Double click on “Task Configuration” or right click and choose “Edit Object”. Task Configuration window appears. Click “System Events” tab.

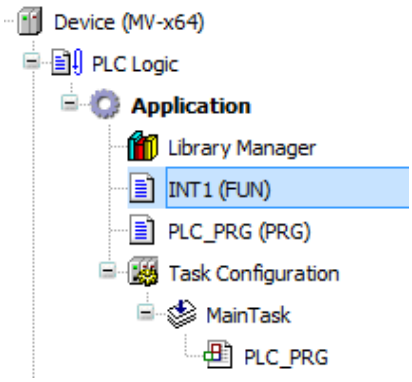






Click “Add Event Handler…” then Add Event Handler window appears. Choose an interrupt event from Interrupt 1 to 5 in “Event” and enter a name in “Function to call”. Click OK button, then FUNCTION will be created.





When new FUNCTION is created, a variable “EventPrm” (type:STRUCT EVTPARAM_Interrupt) is automatically defined in the FUNCTION. This variable contains no data.
(Library IoDrvHIES is implicitly added to Library Manager, and this STRUCT is defined in it.)

Create your program in this FUNCTION accordingly. Functions contain no internal state information. Use global variables including local I/O (external I/O) in functions.
Local I/O (external I/O) used only in functions is not updated. Make sure to use in other programs or function blocks also.

3.18 Pulse Train Outputs

DC outputs of MICRO-EHV+ are configurable for pulse train output. This can be used to control stepper motors. Two different controls, position control and speed control, are supported. The both controls have ramp feature.

Position control

Position and speed are specified as number of pulses and frequency. After specified number of pulses have been completed, pulse train operation stops.

Speed control

Only speed is specified as frequency of pulses. After started, frequency can be changed freely. Separate function block is required to stop pulse train operation.

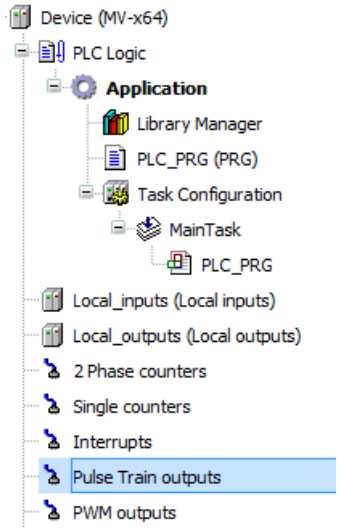
Available I/O addresses of pulse train output and PWM output are listed as below.

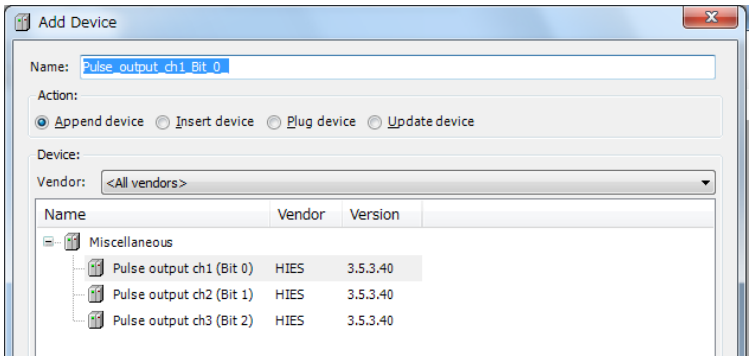
		bit 0 %QX3.0	bit 1 %QX3.1	bit 2 %QX3.2	bit 3 %QX3.3	bit 4 %QX3.4	bit 5 %QX3.5
Pulse Train output	Pulse	ch.1	ch.2	ch.3			
	Direction (optional)				ch.1	ch.2	ch.3
PWM output		ch.1	ch.2	ch.3			

Since the same output addresses are shared by pulse train and PWM outputs, either pulse train or PWM outputs can be configured in each output.

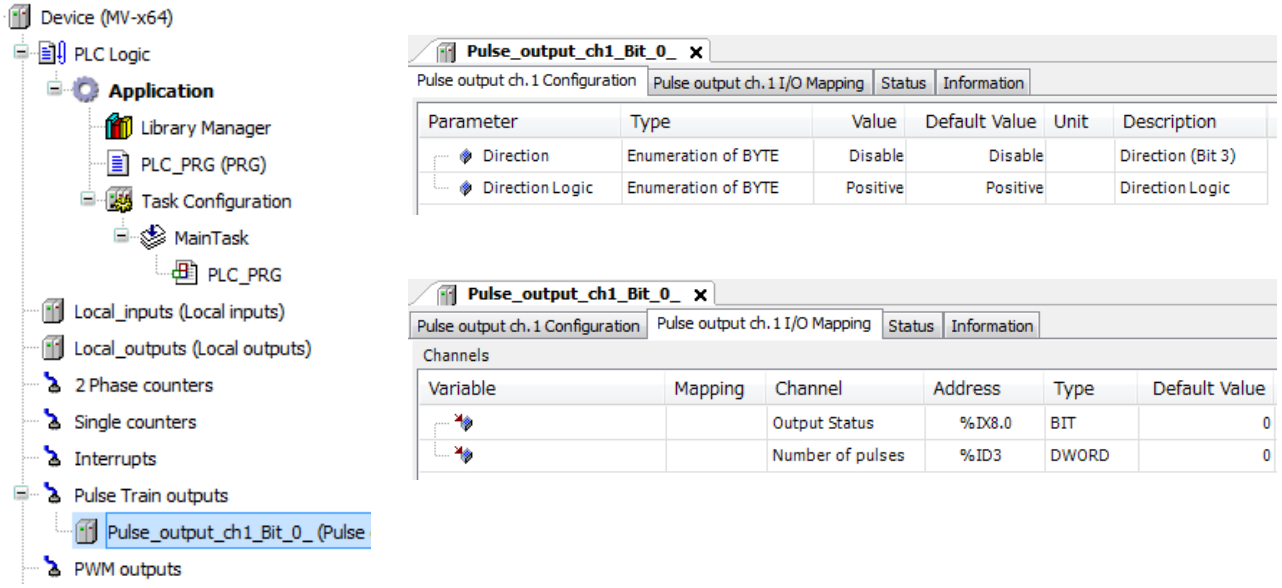
Configuration

Right click on “Pulse Train outputs” and choose “Add Device…”.
Then “Add Device” window appears. Click Pulse output chx and [Add Device] button.





Double click on “Pulse_output_chx…” or right click and choose “Edit Object”. Pulse output configuration window appears. Pulse output has Configuration tab and Mapping tab as below.



Configuration

Parameter	Descriptions	Choices	Default
Direction	Additional output besides pulse train output can be used to define direction.	Disable	Disable
		Enable	
Direction Logic	Forward (Number of pulses = positive) : ON Reverse (Number of pulses = negative) : OFF	Positive	Positive
	Forward (Number of pulses = positive) : OFF Reverse (Number of pulses = negative) : ON	Negative	

Mapping

Name	I/O	Type	Descriptions
Output status	Input (%IX)	BOOL	TRUE: Pulse train is being output. FALSE: Pulse train is not output.
Number of pulses	Input (%ID)	DWORD	Accumulated number of pulses is stored. (position control only) 0 to 4,294,967,295 (direction disabled) -2,147,483,648 to 2,147,483,647 (direction enabled)

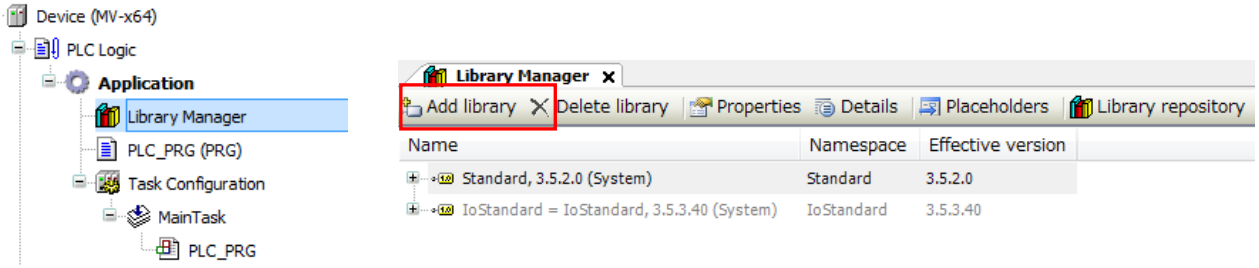
When the number of pulses exceeds the maximum or minimum value, it returns to 0 or 4,294,967,295.

Operations

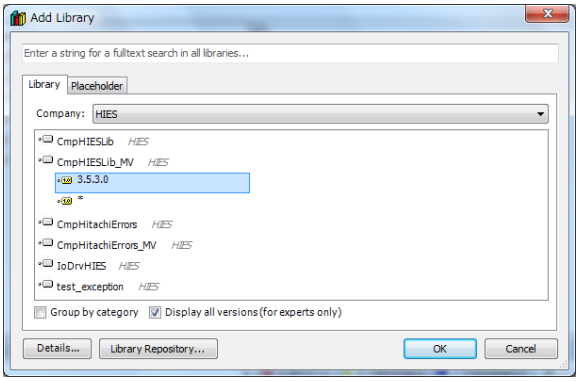
Pulse train output can be controlled by variables and function blocks as below.

Operations	Variable or FB	Name
Read output status	Variable	Output status (%IX)
Read accumulated number of pulses	Variable	Number of pulses (%ID)
Start pulse train output (position control)	Function block	MV_PLS_Start
Write current position data (position control)	Function block	MV_PLS_Write
Start pulse train output (speed control)	Function block	MV_PLS_speed
Change speed of pulse train output (speed control)	Function block	MV_PLS_Changespeed
Stop pulse train output	Function block	MV_PLS_Stop

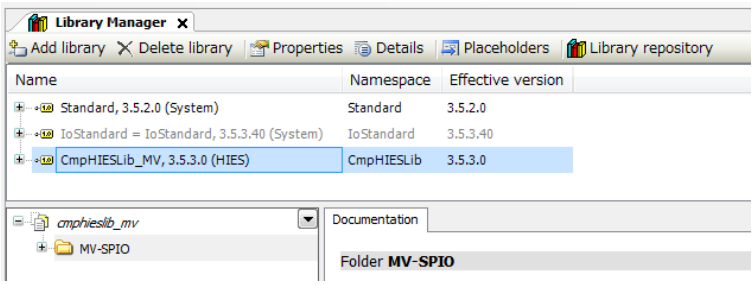
Dedicated library must be added to the project for pulse train operations.
Double click on “Library Manager” or right click and choose “Edit Object”. Library Manager window appears.



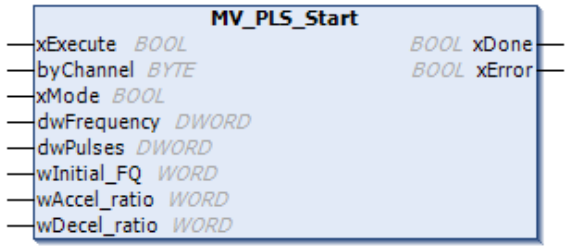
Click “Add library” and choose “CmpHIESLib_MV”.



CmpHIESLib library is added in the project.

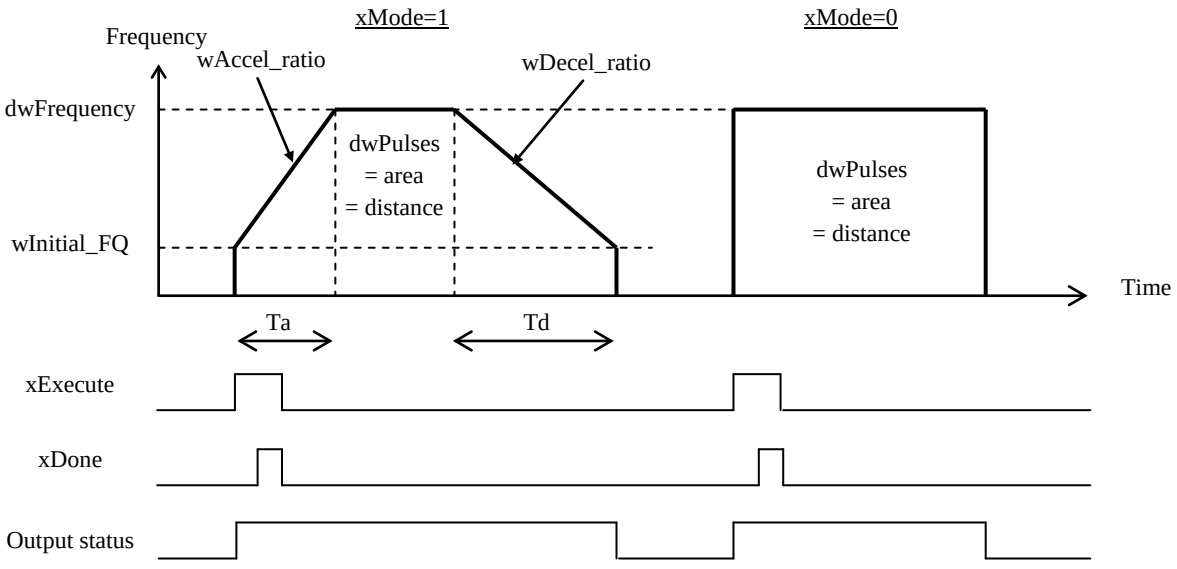


MV_PLS_Start



FB name	Description		
MV_PLS_Start	This function block is to output pulse train for position control. Pulse train stops after specified number of pulses have been completed.		
Name	**Type**	**I/O**	**Descriptions**
xExecute	BOOL	IN	A rising edge starts the execution.
byChannel	BYTE	IN	Channel number (1 to 3)
xMode	BOOL	IN	0: Without ramp 1: With ramp
dwFrequency	DWORD	IN	Frequency of pulse train outputs (20 to 100,000 [Hz])
dwPulses	DWORD	IN	The number of pulses 0 to 4,294,967,295 (direction disabled) -2,147,483,648 to 2,147,483,647 (direction enabled)
wInitial_FQ	WORD	IN	Initial frequency (20 to 65,535[Hz]) — If ramp is not used (xMode=0), set 0 or leave as no connection.
wAccel_ratio	WORD	IN	Acceleration ratio (1 to 65,535 [Hz/100ms])
wDecel_ratio	WORD	IN	Deceleration ratio (1 to 65,535 [Hz/100ms])
xDone	BOOL	OUT	Execution done successfully
xError	BOOL	OUT	Error occurred

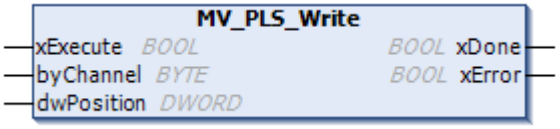
Note: If this function block is executed while pulse train output in operation, all the parameters are overwritten.



$$wAccel\_ratio = \frac{dwFrequency\ [Hz] - wInitial\_FQ\ [Hz]}{10 \times Ta\ [s]}$$

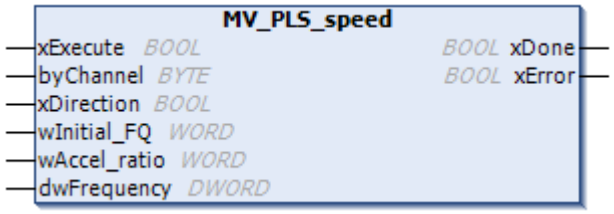
$$wDecel\_ratio = \frac{dwFrequency\ [Hz] - wInitial\_FQ\ [Hz]}{10 \times Td\ [s]}$$

MV_PLS_Write



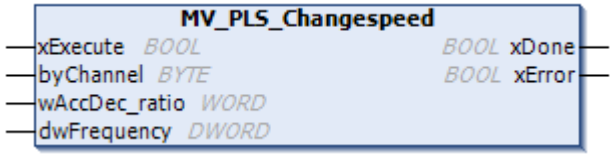
FB name	Description		
MV_PLS_Write	This function block is to overwrite current position data		
Name	**Type**	**I/O**	**Descriptions**
xExecute	BOOL	IN	A rising edge starts the execution.
byChannel	BYTE	IN	Channel number (1 to 3)
dwPosition	DWORD	IN	Position data to write 0 to 4,294,967,295 (direction disabled) -2,147,483,648 to 2,147,483,647 (direction enabled)
xDone	BOOL	OUT	Execution done successfully
xError	BOOL	OUT	Error occurred

MV_PLS_Speed



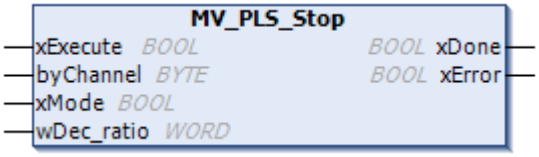
FB name	Description		
MV_PLS_Speed	This function block is to output pulse train for speed control. Pulse train operation continues until the function block MV_PLS_Stop is received.		
Name	**Type**	**I/O**	**Descriptions**
xExecute	BOOL	IN	A rising edge starts the execution.
byChannel	BYTE	IN	Channel number (1 to 3)
xDirection	BOOL	IN	0: Forward 1: Reverse
wInitial_FQ	WORD	IN	Initial frequency (20 to 65,535[Hz])
wAccel_ratio	WORD	IN	Acceleration ratio (1 to 65,535 [Hz/100ms])
dwFrequency	DWORD	IN	Frequency of pulse train outputs (20 to 100,000 [Hz])
xDone	BOOL	OUT	Execution done successfully
xError	BOOL	OUT	Error occurred

MV_PLS_Changespeed



FB name	Description		
MV_PLS_Changespeed	This function block is to change frequency of pulse train for speed control. This function block works only when pulse train is in operation by MV_PLS_Speed.		
Name	**Type**	**I/O**	**Descriptions**
xExecute	BOOL	IN	A rising edge starts the execution.
byChannel	BYTE	IN	Channel number (1 to 3)
wAccDec_ratio	WORD	IN	Acceleration ratio (1 to 65,535 [Hz/100ms])
dwFrequency	DWORD	IN	Frequency of pulse train outputs (20 to 100,000 [Hz])
xDone	BOOL	OUT	Execution done successfully
xError	BOOL	OUT	Error occurred

MV_PLS_Stop



FB name	Description		
MV_PLS_Stop	This function block is to stop pulse train outputs for position and speed control both.		
Name	**Type**	**I/O**	**Descriptions**
xExecute	BOOL	IN	A rising edge starts the execution.
byChannel	BYTE	IN	Channel number (1 to 3)
xMode	BOOL		0: Stop with ramp 1: Stop without ramp
wDec_ratio	WORD	IN	Deceleration ratio (1 to 65,535 [Hz/100ms] If ramp is not used (xMode=1), set 0 or leave as no connection.
xDone	BOOL	OUT	Execution done successfully
xError	BOOL	OUT	Error occurred

Note

If output configuration is changed from pulse train output to standard DC output, the PLC must be power-cycled.

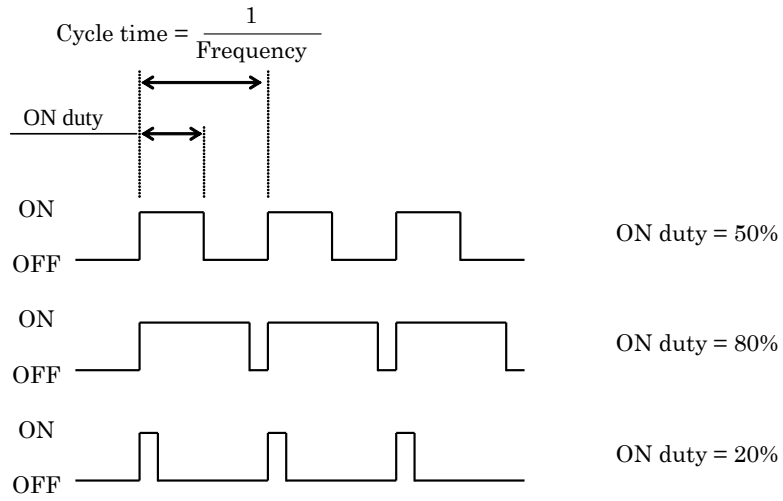
3.19 PWM Outputs

DC outputs of MICRO-EHV+ are configurable for PWM (Pulse-Width Modulation) output. This can be used to control DC and stepper motors.

Available I/O addresses of pulse train output and PWM output are listed as below.

		bit 0 %QX3.0	bit 1 %QX3.1	bit 2 %QX3.2	bit 3 %QX3.3	bit 4 %QX3.4	bit 5 %QX3.5
Pulse Train output	Pulse	ch.1	ch.2	ch.3			
	Direction (optional)				ch.1	ch.2	ch.3
PWM output		ch.1	ch.2	ch.3			

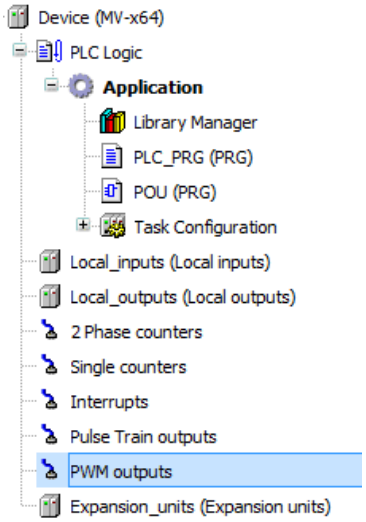
Since the same output addresses are shared by pulse train and PWM outputs, either pulse train or PWM outputs can be configured in each output.

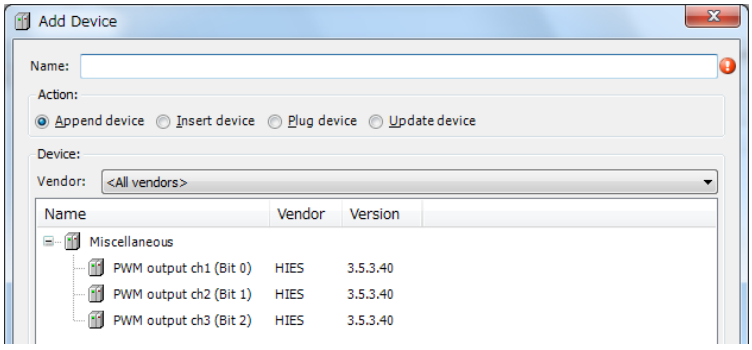


Configuration

Right click on “PWM outputs” and choose “Add Device…”.

Then “Add Device” window appears. Click PWM output chx and [Add Device] button.





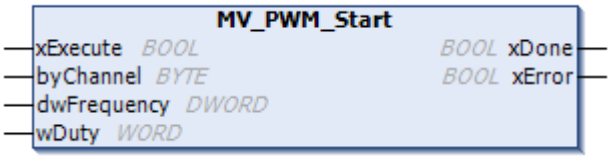
Configuration and Mapping are not necessary for PWM outputs.

Operations

PWM output can be controlled by function blocks as below.

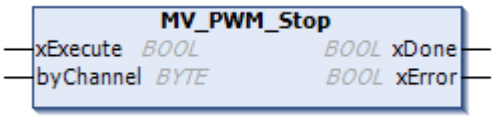
Operations	Variable or FB	Name
Start PWM output	Function block	MV_PWM_Start
Stop PWM output	Function block	MV_PWM_Stop

MV_PWM_Start



FB name	Description		
MV_PWM_Start	This function block is to start PWM output.		
Name	**Type**	**I/O**	**Descriptions**
xExecute	BOOL	IN	A rising edge starts the execution.
byChannel	BYTE	IN	Channel number (1 to 3)
dwFrequency	DWORD	IN	Frequency of pulse train outputs (20 to 100,000 [Hz])
wDuty	WORD	IN	ON duty (0 to 100 [%])
xDone	BOOL	OUT	Execution done successfully
xError	BOOL	OUT	Error occurred

MV_PWM_Stop



FB name	Description		
MV_PLS_Stop	This function block is to stop PWM output.		
Name	**Type**	**I/O**	**Descriptions**
xExecute	BOOL	IN	A rising edge starts the execution.
byChannel	BYTE	IN	Channel number (1 to 3)
xDone	BOOL	OUT	Execution done successfully
xError	BOOL	OUT	Error occurred

Note

If output configuration is changed from PWM output to standard DC output, the PLC must be power-cycled.

3.20 Option board

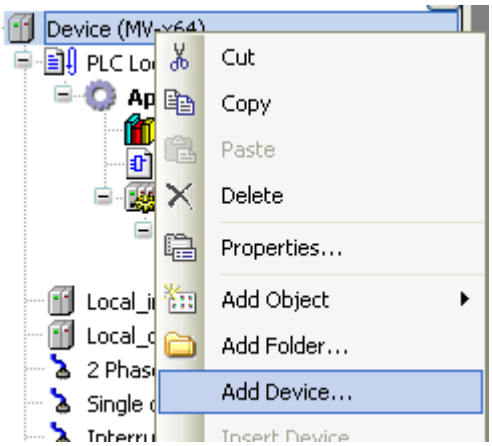
3.20.1 Supported function

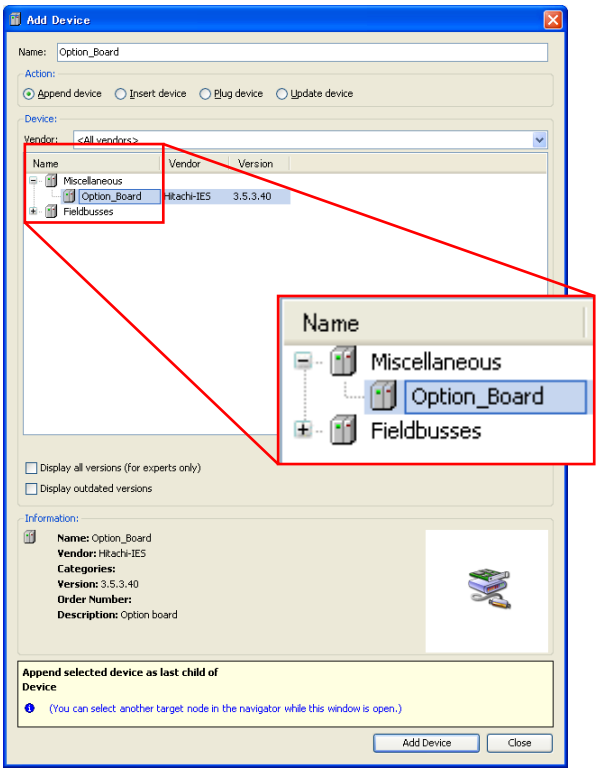
Option board supports the following communication functions and analog input.

Function		OBV-NES	OBV-485A
Modbus-RTU master	Modbus-RTU master	X	X
	Modbus-RTU slave	X	X
	General purpose communication (Free protocol)	X	X
Analog input		-	X

X = Supported, - = Not supported

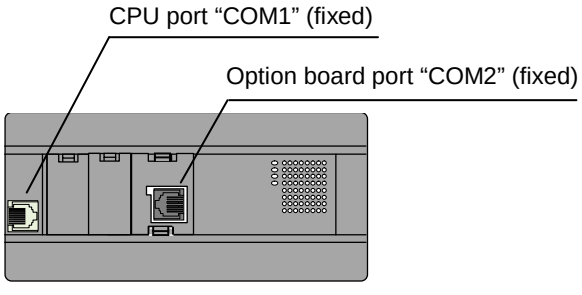
Right click on “Device” and choose “Add Device...”.
“Add Device” window appears.
Click “Option_Board” and [Add Device] button.





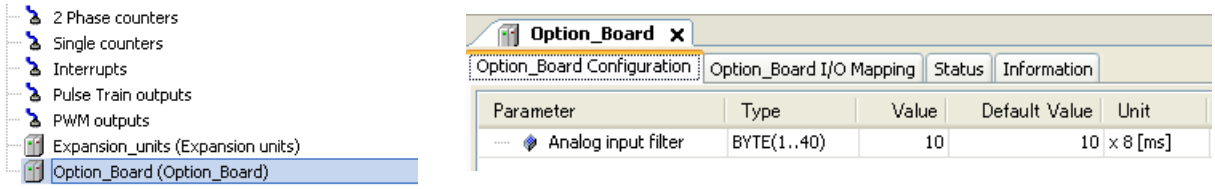
3.20.2 Port number setting

The port number setting of option board is fixed to “COM2” (“COM1” is reserved for CPU local port).

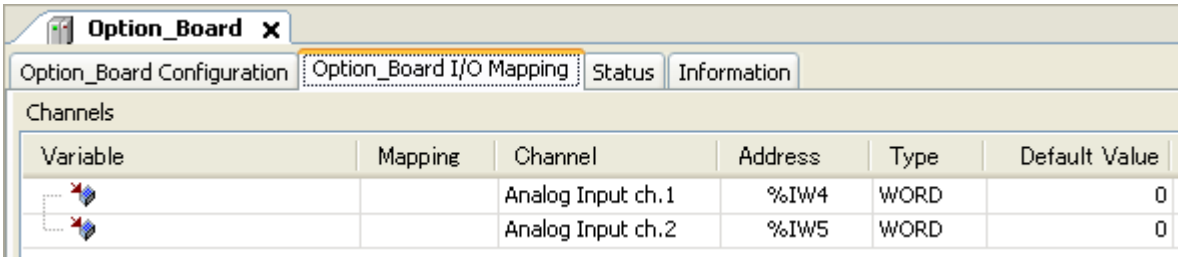


3.20.3 Analog input setting

Double click on “Option_Board” or right click and choose “Edit Object.” Option_Board Configuration window appears. Set “Analog input filter”.



Data of Analog input will be assigned to %IW as seen in “Option_Board I/O Mapping” tab.

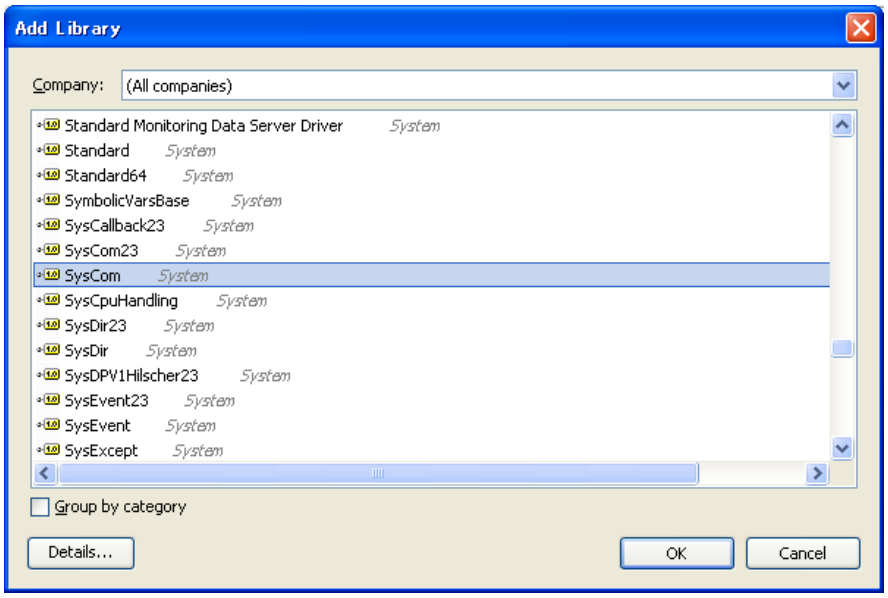


3.20.4 Modbus-RTU communication

Modbus operation is same as CPU port. Refer to section 3.15.4 and 3.15.5 for further information.

3.20.5 General purpose communication

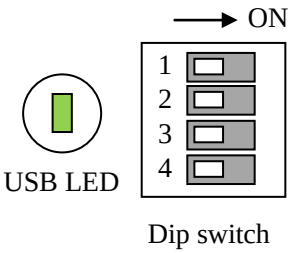
Besides Modbus-RTU communication, option board supports general purpose communication same as CPU port. Add “SysCom” library by clicking “Add library” on Library manager. Only the difference from CPU port is COM port number. Be sure to set “COM2”. Refer to section 3.24.3 Serial communication for further information.

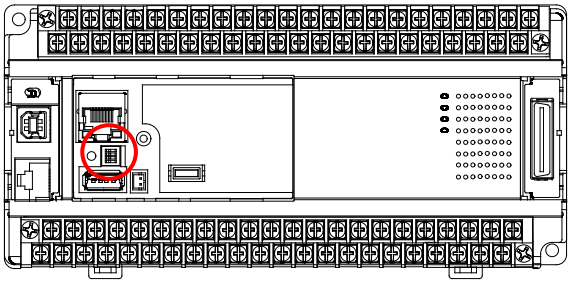


3.21 USB program transfer

This function is supported by MICRO-EHV+ version 3.5.3.41 or newer.

User program can be downloaded, uploaded or verified according to dip switch settings as below.





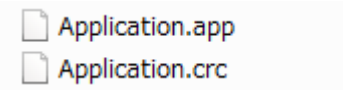
Function	SW1	SW2	SW3	SW4	Action when	USB LED
Download (USB➔PLC)	ON	OFF	OFF	OFF	Power up	1s ON / 1s OFF
Upload (USB➔PLC)	OFF	ON	OFF	OFF	Plug USB	0.5s ON / 0.5s ON
Verify	ON	ON	OFF	OFF	Plug USB	2s ON / 2s OFF
Boot from USB	OFF	OFF	ON	OFF	Power up	1s ON / 1s OFF
USB memory is plugged.						ON
USB memory is removed.						OFF

3.21.1 Download from USB to PLC

Application program is downloaded from USB to FLASH memory in the PLC. Currently stored application in the FLASH memory will be overwritten. Follow the steps as below.

(1) Create project files

Project files to be downloaded are created with EHV-CODESYS. Choose the menu [Online] - [Create boot application] in off-line mode and save them to USB memory. Then file “Application.app” and “Application.crc” will be created in the root directory of USB memory.



(2) Plug the USB memory to the USB connector of MICRO-EHV+.

(3) Set the dip switch 1-ON, 2,3,4-OFF.

(4) Turn ON power to the PLC

(5) After USB LED blinking, project files have been downloaded to the PLC.

Besides project files created by EHV-CODESYS, also uploaded project files from PLC can be downloaded.

Note

- If corrupt file is downloaded or no boot project is stored in USB and internal FLASH memory, initialization during power-up does not complete, which causes STATUS LED keeps blinking and OK LED does not light up. In this case, download a right project to recover.
- If the parameter “Retain Mismatch” in Configuration is changed, power cycling is necessary to reflect.

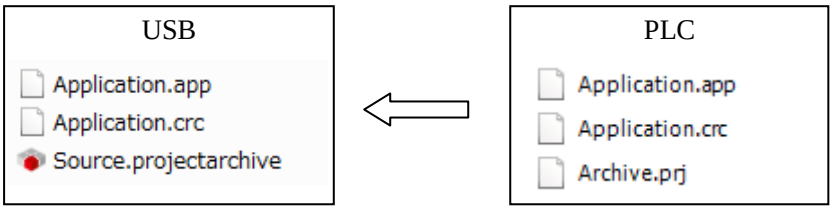
3.21.2 Upload from PLC to USB

Application program and source file, if available, is uploaded from FLASH memory in the PLC to USB memory. Follow the steps as below.

(1) Set the dip switch 2-ON, 1,3,4-OFF.

(2) Plug the USB memory to the USB connector of MICRO-EHV+.

(3) Application.app and Application.crc will be uploaded from PLC to USB. If source file (Archive.prj) is stored in the PLC, it will be uploaded too as “Source.projectarchive”, which can be opened by EHV-CODESYS.



3.21.3 Verify between PLC and USB

Application programs between PLC and USB are verified. Follow the steps as below.

(1) Set the dip switch 1,2-ON, 3,4-OFF.

(2) Store Application.app and Application.crc in the USB memory.

(3) Plug the USB memory to the USB connector of MICRO-EHV+.

(4) USB LED shows verifying result as below.

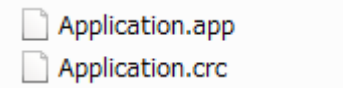
Verifying result	USB LED
Match	ON
Mismatch	Blinking in 5 sec. (250ms ON / 250ms OFF)

3.21.4 Boot from USB

Application program is downloaded from USB to RAM memory in the PLC. Currently stored application in the FLASH memory will NOT be overwritten. When USB memory is removed and power is cycled, original application program in the FLASH memory will be loaded to RAM as a boot project. Follow the steps below.

(1) Create project files

Project files to be downloaded are created with EHV-CODESYS. Choose the menu [Online] - [Create boot application] in off-line mode and save them to USB memory. Then file “Application.app” and “Application.crc” will be created in the root directory of USB memory.



(2) Plug the USB memory to the USB connector of MICRO-EHV+.

(3) Set the dip switch 3-ON, 1,2,4-OFF.

(4) Turn ON power to the PLC

(5) After USB LED blinking, project files have been downloaded to RAM memory of the PLC.

Note

- If USB memory device is not plugged or no application file is stored in USB memory, exception “load boot project failed” (error code 31) is detected.
- IP information is not changed in this operation mode. Do not set “Yes” at “Change IP information” in Configuration parameter.

3.22 USB data logging (File system)

This function is supported by MICRO-EHV+ version 3.5.3.41 or newer.
This function works only when USB memory is plugged in MICRO-EHV+.

Since USB memory supports file system, the library CAA File is available with USB memory. One of the useful functions of CAA File is data logging. A sample program of data logging is introduced as below.

This sample program will create a CSV file (file name: Datalog.csv) with 3 data (time stamp, dummy data, text) as follows. New log data is additionally written to the same file every 10 seconds automatically. Modify data format and writing timing according to your system requirements.

	A	B	C
1			
2	DT#2014-02-07-01:25:08	32	This is Test !
3	DT#2014-02-07-01:25:17	42	This is Test !
4	DT#2014-02-07-01:25:27	56	This is Test !
5	DT#2014-02-07-01:25:38	61	This is Test !
6	DT#2014-02-07-01:25:48	71	This is Test !
7	DT#2014-02-07-01:25:58	86	This is Test !

Variable declaration

```
PROGRAM Logging
VAR
    sFileName       : CAA.FILENAME;
    FileOpen        : File.Open;
    FileClose       : File.Close;
    FileWrite       : File.Write;
    sMainDir        : STRING := '/usbHd/00'; // Base directory can be adapted
    iState          : UINT;
    hfile           : CAA.HANDLE;
    sWriteLine      : STRING(128);
    xError          : BOOL := FALSE;
    xInit           : BOOL := FALSE;
    xTestDone       : BOOL := FALSE;
    xStartWr        : BOOL := FALSE; (* Request bit to write new log *)
    GetRTCData      : RTClk.GetDateAndTime;
    xRDRTC          : BOOL;
    xReadDone       : BOOL;
    dtTemp          : DATE_AND_TIME;
    sDT             : STRING;
    iNum            : INT;
    sNum            : STRING;
    T1              : TON;
END_VAR
```

Program

```
IF xStartWr THEN  // Request bit to write new log data
    sFileName := '/Datalog.csv'; // File name
    sFileName := CONCAT(sMainDir, sFileName);
    iState := 0;
    xInit := TRUE;
    xTestDone := FALSE;
    xError := FALSE;
    xStartWr := FALSE;
END_IF

IF xInit THEN
    CASE iState OF
    0: //Open the specified file with write access.
       FileOpen(xExecute:= TRUE, sFileName:= sFileName, xExclusive:= FALSE,
        eFileMode:= File.MODE.MAPPD);
       IF FileOpen.xDone = TRUE THEN
           iState := 1;
           hfile := FileOpen.hFile;
           FileOpen(xExecute:= FALSE);
           xRDRTC := TRUE;
       ELSIF FileOpen.xError = TRUE THEN
           iState:= 10;
           FileOpen(xExecute:= FALSE);
       END_IF

    1:  // Get RTC data
       GetRTCData(xExecute:= xRDRTC, xDone => xReadDone, dtDateAndTime=> dtTemp,);
       IF xReadDone = TRUE THEN
           sDT := DT_TO_STRING (dtTemp); // 1st logging data (RTC)
           GetRTCData(xExecute := FALSE);
           iState := 2;
       END_IF

    2: // Write a string to the opened file.
       // Update the number
       iNum := iNum +1;
       sNum := INT_TO_STRING(iNum);  // 2nd logging data (incremental number in STRING)
       // Combine the data
       sWriteLine := CONCAT('$r$n', sDT);
       sWriteLine := CONCAT(sWriteLine, ',');
       sWriteLine := CONCAT(sWriteLine, sNum);
       sWriteLine := CONCAT(sWriteLine, ',');
       sWriteLine := CONCAT(sWriteLine, 'This is Test ! ');  // 3rd logging data (sample text)
       FileWrite(xExecute:= TRUE,hFile:= hfile, pBuffer:= ADR(sWriteLine),
        szSize:= INT_TO_UDINT(LEN(sWriteLine)));
       IF FileWrite.xDone = TRUE THEN
           iState := 3;
           FileWrite(xExecute:= FALSE);
       ELSIF FileWrite.xError = TRUE THEN
           iState:= 11;
           FileWrite(xExecute:= FALSE);
       END_IF
       xRDRTC := FALSE;

    3: // Close that file.
       FileClose(xExecute:= TRUE, hFile:= hfile);
       IF FileClose.xDone = TRUE THEN
           iState := 4;
           FileClose(xExecute:= FALSE);
       ELSIF FileClose.xError = TRUE THEN
           iState:= 12;
           FileClose(xExecute:= FALSE);
       END_IF
```

```
    4: // The test is done.
        xTestDone:= TRUE;
        xError := FALSE;
        xInit := FALSE;
    10:
        xTestDone := FALSE;
        xError := TRUE; (* File open error *)
        xInit := FALSE;
    11:
        xTestDone := FALSE;
        xError := TRUE; (* File write error *)
        xInit := FALSE;
    12:
        xTestDone := FALSE;
        xError := TRUE; (* File close error *)
        xInit := FALSE;
END_CASE
END_IF

T1(IN:=NOT(T1.Q), PT:=T#10S);
IF T1.Q THEN
    xStartWr:=TRUE;  // TRUE every 10 sec.
END_IF;
```

Note

Not all the function blocks of CAA File libraries are supported as below.

Function block	Supported	Function block	Supported
Close	Y	Read	Y
Copy	Y	Rename	Y *1*2
Delete	Y	SetPos	Y
EOF	Y	Write	Y
Flush	-	DirClose	Y
GetAttribute	-	DirCreate (incl. sysDirCreate)	*3
GetPos	Y	DirList	Y
GetSize	Y	DirOpen	Y
GetTime	Y	DirRemove	Y
Open	Y	DirRename	Y *1*2

Y : Supported

- : Not supported

*1 : If target file does not exist, no error is returned due to a bug in the CAA File library 3.5.3.0.

*2 : Use the same path for old and new file/directory. If they are different, rename function does not work properly.

*3 : It is not recommended to use DirCreate and sysDirCreate since it takes long time to execute, which could result in processor overload exception. Instead of this, create new directory manually with PC in advance.

Note

Do not plug out USB memory while accessing file/directory, otherwise it is not properly recognized when plugged in the next time. If it is necessary to plug out USB memory while PLC is in RUN status and USB memory is always accessed, be sure to close file/directory in user program before plugging out, for example adding an input to stop accessing and close file/directory.

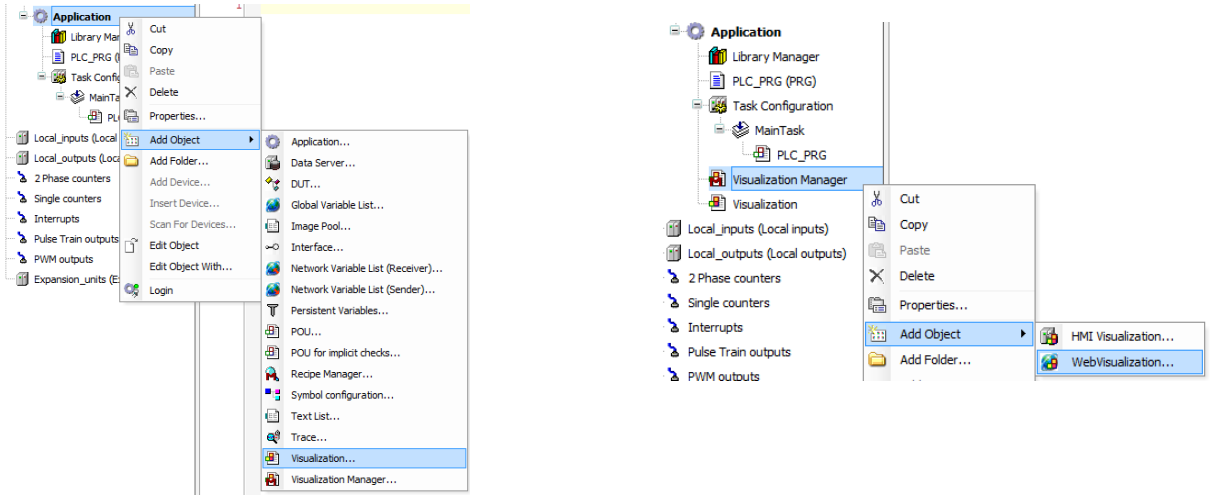
3.23 Web visualization

This function is supported by MICRO-EHV+ version 3.5.3.41 or newer.
This function works only when USB memory is plugged in MICRO-EHV+.

Besides standard visualization, MICRO-EHV+ supports Web visualization. Once visualization application is created and downloaded together with user program, PLC works as a web server and any web browser can access, read and write variables in the PLC.

Configuration

Right click on “Application” and choose “Add Object” - “Visualization”. Then “Visualization Manager” and “Visualization” inserted under the Application.
Right click on “Visualization” and choose “Add Object” and “WebVisualization”.



Creating web page

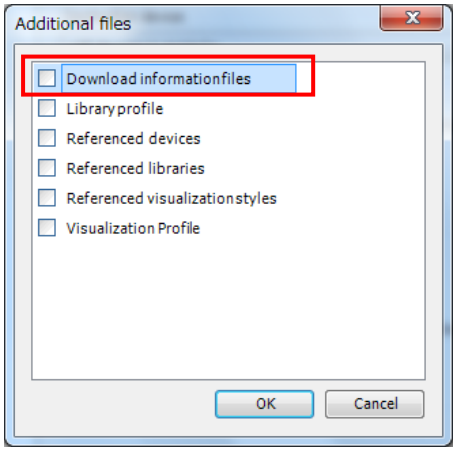
Refer to online-help of EHV-CODESYS for further information to create visualization pages.

Access from web browser

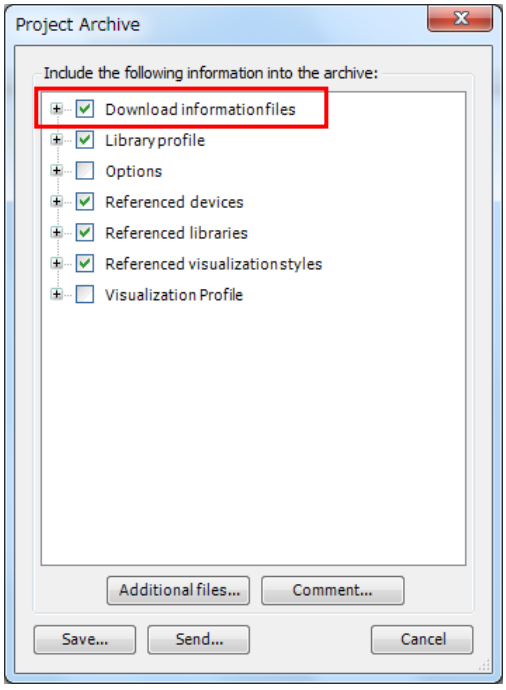
Created web visualization application is downloaded together with application program to MICRO-EHV+, which can be accessed by any web browsers. When IP address of the target PLC is 192.168.0.1, enter URL as follows.
http://192.168.0.1/webvisu.htm/

Note

- Web browser must support HTML5 with JavaScript enabled.
- Webvisualization works only when PLC is in RUN status.
- Local I/O (external I/O) is not updated in web visualization and standard visualization. If it is necessary to read or write local I/O directly, use additional variables in web visualization and copy them to/from local I/O in the IEC program.
- Webvisualization requires processor load. Normally up to 80% of cycle time of the task is available for user program execution without Webvisualization however, 15 to 20% of cycle time is for program execution with Webvisualization. Be sure to set longer cycle time for the task of user program.
- Since Webvisualization requires a lot of memory, “Source download” fails because the flash memory size in MICRO-EHV+ is not enough. In order to download source file, disable “Download information files” in Additional files in [Project Settings]-[Source Download].
  Without “Download information files”, online change is not possible by uploaded source file. (It is possible to download, but CPU must be stopped.) If online change is necessary, be sure to keep project archive file in your PC with Download information files enabled.



Additional files for Source download

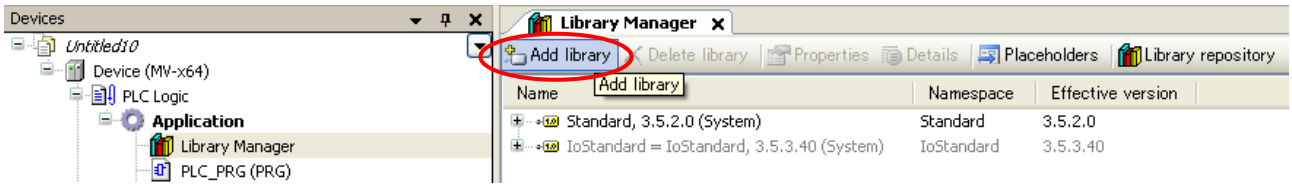


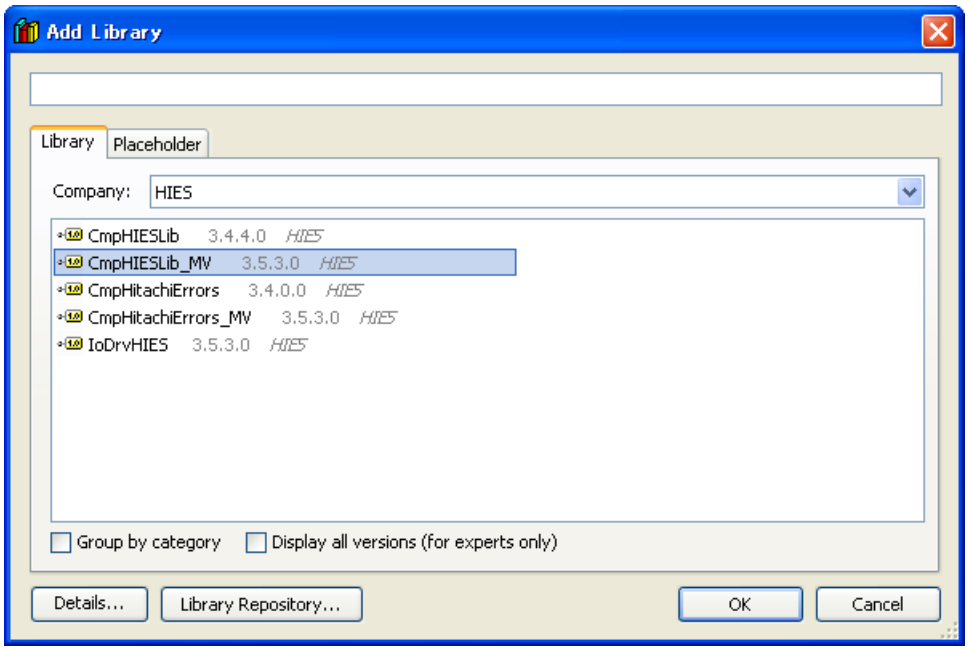
Setting for Project Archive

3.24 Libraries

3.24.1 How to install

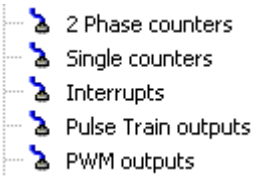
In order to read/write MICRO-EHV+ series PLC's specific information, the following libraries are available. Add necessary CmpHIESLib_MV by choosing "Add library" as shown below.



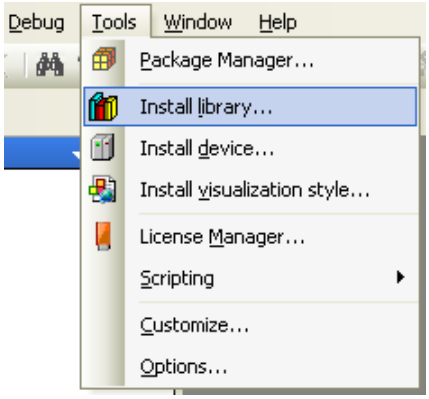


CmpHIESLib_MV is Hitachi-IES's special library including;

2 Phase counters
Single counters
Interrupts
Pulse Trans outputs
PWM outputs

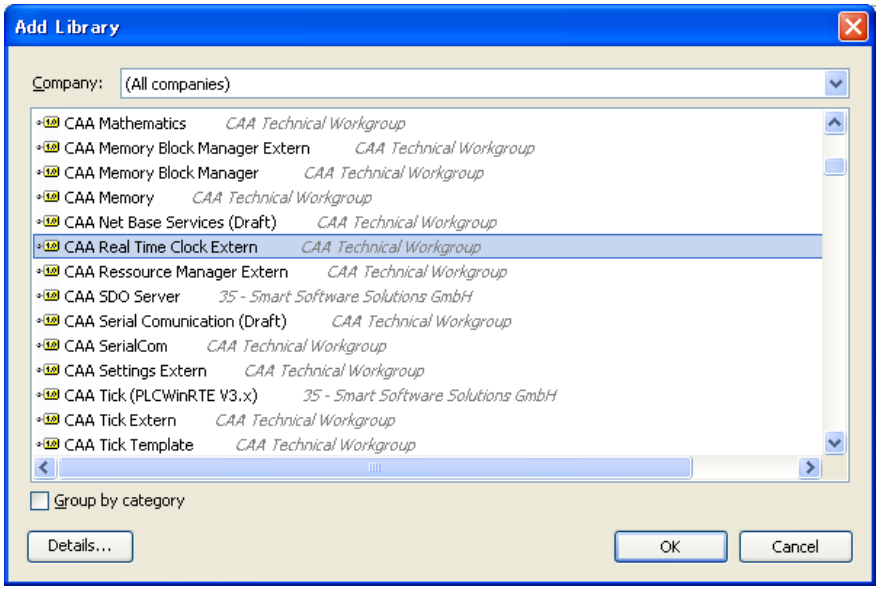


If these libraries are not found in the library list as above, install library by choosing [Tools]-[Install library...].

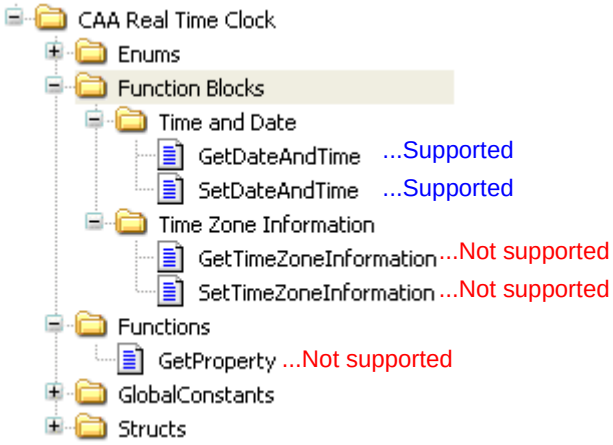


3.24.2 Real time clock

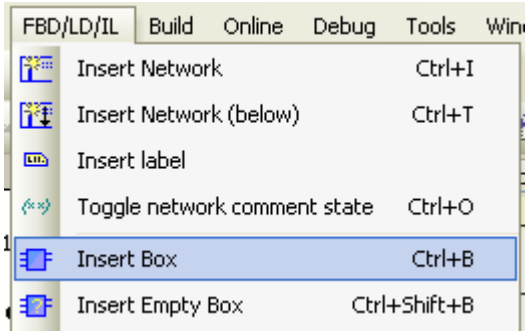
Add “CAA Real Time Clock Extern” library by clicking “Add library” on Library manager.

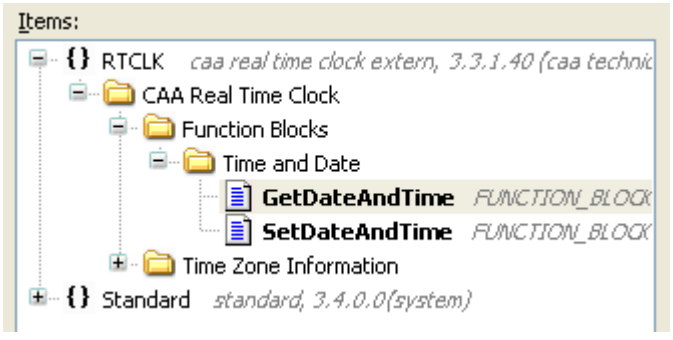


In the CAA Real Time Clock Extern libraries, GetDateAndTime and SetDateAndTime are supported. Be noted that the others are not supported.



Following example is in FBD language. Choose [Insert Box] in [FBD/LD/IL] menu and GetDateAndTime or SetDateAndTime as shown below.

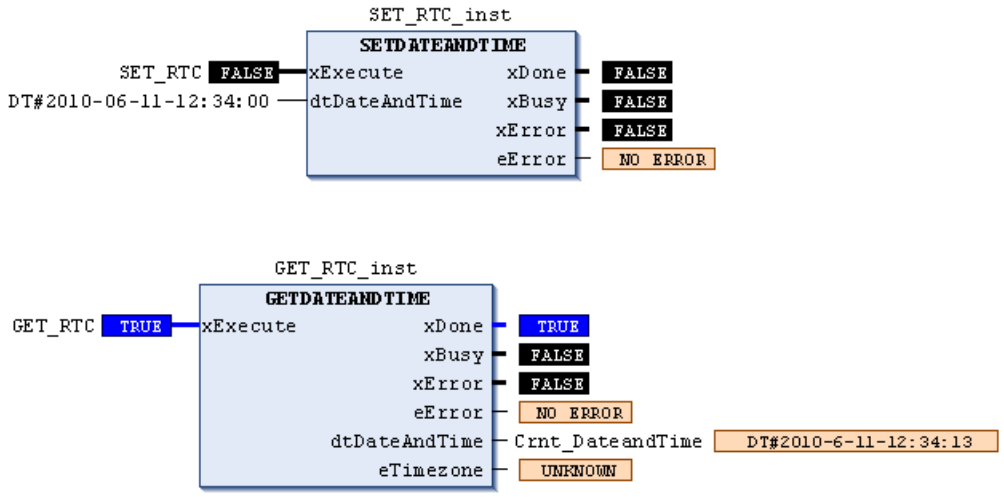




Declare instance of the function blocks and necessary variables as below.

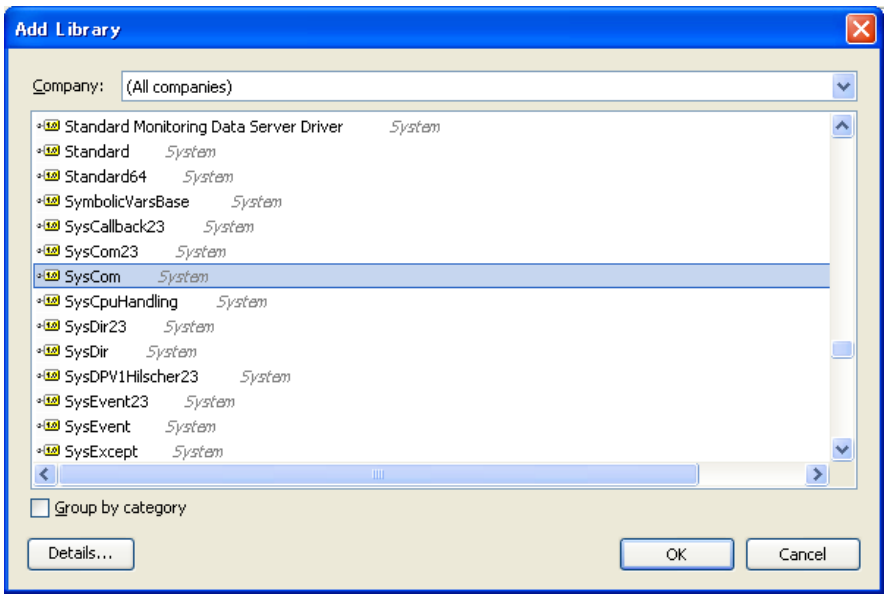
```
PROGRAM PLC_PRG
VAR
    SET_RTC_inst: SETDATEANDTIME;
    SET_RTC: BOOL;
    GET_RTC_inst: GETDATEANDTIME;
    GET_RTC: BOOL;
    Crnt_DateandTime: DATE_AND_TIME;
END_VAR
```

By rising edge of xExecute bit of SETDATEANDTIME, data in dtDateAndTime is written to the RTC device.
By rising edge of xExecute bit of GETDATEANDTIME, current date and time is read out to the variable connected to dtDateAndTime as shown below. When xExecute bit of GETDATEANDTIME is FALSE, dtDateAndTime is default value 1970-1-1-0:0:0.



3.24.3 Serial communication

Add “SysCom” library by clicking “Add library” on Library manager. Instead of SysCom library, CAA SerialCom can be used, but a sample program of SysCom library is shown in this manual.

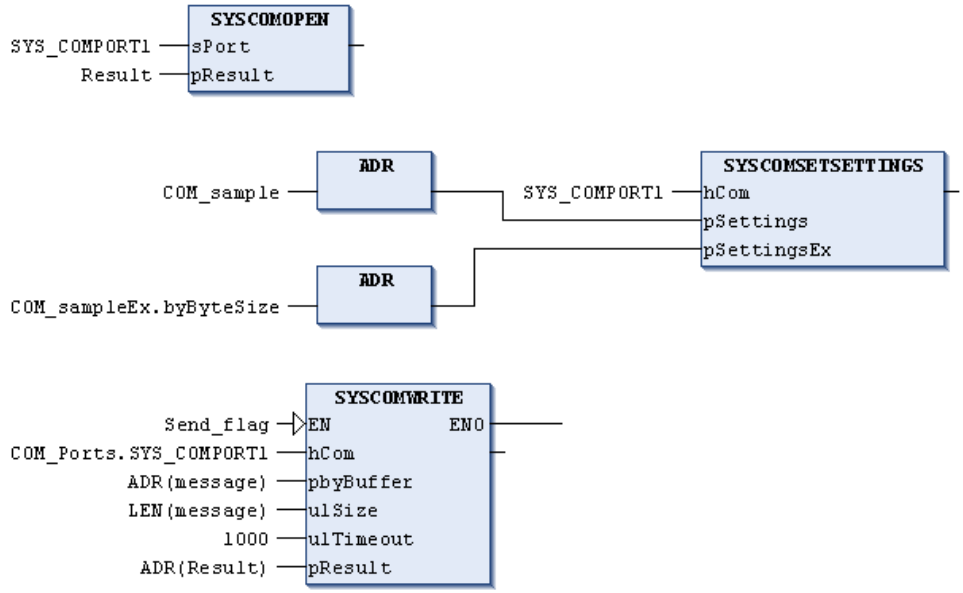


It is recommended to use ST language for serial communication settings since it is more flexible.

```
1  PROGRAM serial_sample2
2  VAR
3      COM_sample:COM_Settings;
4      COM_sampleEx:COM_SettingsEx;
5      message: STRING:='123';
6      Result: DWORD;
7      write_out: UDINT;
8      test: INT;
9  END_VAR
10
```

```
1  COM_sample.sPort:=COM_Ports.SYS_COMPORT1;            ...COM1
2  COM_sample.byParity:=COM_Parity.SYS_NOPARITY;        ...Non parity
3  COM_sample.byStopBits:=COM_StopBits.SYS_ONESTOPBIT;...1 stop bit
4  COM_sample.ulBaudrate:=COM_Baudrate.SYS_BR_19200;   ...baudrate 19,200bps
5  COM_sample.ulBufferSize:=100;                        ...buffer size 100 bytes
6  COM_sample.ulTimeout:=10;                            ...Timeout 10ms
7  COM_sampleEx.byByteSize:=8;                          ...8 bit / frame
8
9  SysComOpen(SYS_COMPORT1, ADR(Result));
10
11 SysComSetSettings(SYS_COMPORT1, ADR(COM_sample), ADR(COM_sampleEx));
12
13 IF test= 1 THEN;
14 message := CONCAT('$02', message);    ...Connect  02 + "123"
15 message := CONCAT(message, '$0d');    ...Connect  02 "123" + 0d
16 write_out:=SysComWrite( SYS_COMPORT1, ADR(message), LEN(message), 1000, ADR(Result));
17 test:=0;
18 END_IF;
```

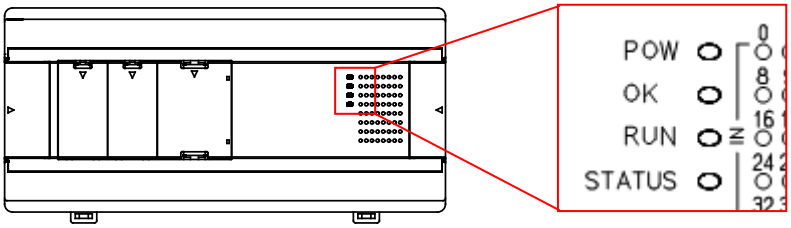
If the variable "test" is 1, then string data "02 31 32 33 0D" (STX 123 CR) will be sent out.
It is possible to write in FBD language as below, but it is necessary to set parameters in COM_Settings and COM_SettingsEx as same as above program line 1 to 7.



3.25 Troubleshooting

Error indication

MICRO-EHV+ indicates the error by the lighting pattern (ON / blink / OFF) of OK LED. If two or more errors are detected at the same time, smaller error code has higher priority to be displayed. If error is detected, read the description following countermeasures depending on error level.



Error code

Error code	Error level	Countermeasure
88, 11 to 19	Serious error	Cycle power. If it does not solve, contact your local supplier.
20 to 31	Exception	Exception status is cleared only by Reset operation. Execute Reset cold/warm/origin by EHV-CODESYS
70 to 78	Warning	User program execution does not stop by warning. If you need to activate alarm or any action by warning, use CmpHitachi_MV library.

Err. code	Error name [Detected when]	Description	PLC system	Applica-tion	OK LED
88	Microprocessor failure [Power on]	Overflow of internal watchdog timer due to system program error.	Stop	Stop	●
11	System program error (FLASH) [Power on]	Checksum value of system program (Runtime) in FLASH does not match the checksum calculated.	Stop	Stop	●
12	System RAM failure [Power on]	Read/write check for system RAM has failed.	Stop	Stop	●
13	Misalignment / Illegal instruction / Privileged instruction [Always]	Microprocessor has detected an exception processing in system program. (*1)	Stop	Stop	●
16	System program error (system RAM) [Always]	Checksum value of system RAM does not match the checksum calculated.	Stop	Stop	●
18	MAC address error [Power on]	MAC address is missing or wrong value.	Stop	Stop	✳
19	Data memory failure [Power on]	Read/write check for data memory (RAM) has failed.	Stop	Stop	✳

: ON : 250ms ON / 250ms OFF : 500ms ON / 500ms OFF : 1s ON / 1s OFF : 2s ON / 2s OFF : OFF

Err. code	Error name [Detected when]	Description	PLC system	Applica-tion	OK LED
20	Misalignment / Illegal instruction / Privileged instruction [Always]	Microprocessor has detected an exception processing in application.	Run	Stop	
21	Retain identity mismatch [Power on]	Retain data memory is undefined status due to battery empty. (*2)	Run	Stop	
23	Unresolved external references [Always]	Unresolved external references are detected.	Run	Stop	
24	Software watchdog of IEC task expired [Always]	Actual cycle time has exceeded watchdog time. Set longer watchdog time.	Run	Stop	
25	Processor load watchdog [Always]	Microprocessor load watchdog of all IEC task has been exceeded. Set longer interval time of task.	Run	Stop	
26	IEC task configuration failed [Always]	IEC task configuration has failed.	Run	Stop	
27	Division by zero [DIV executed]	The divisor of division command is 0 in IEC program.	Run	Stop	
31	Load boot project failed [Power on]	Checksum value of application (user program) in FLASH does not match the checksum calculated.	Run	Stop	
70	I/O configuration error [Always]	I/O configuration does not match with actual I/O modules.	Run	Run	
71	Battery error [Always]	Battery voltage is low or battery is disconnected.	Run	Run	
75	Option board invalid ID [Power on]	Hardware error is detected in option board. Or MICRO-EHV+ does not support the target option board.	Run	Run	
77	FLASH writing failure [FLASH writing]	Failure has been detected in writing FLASH memory or the number of writing times has been exceeded.	Run	Run	
78	Parameters in FLASH check sum error [Power on]	Checksum value of parameters in FLASH (IP address, etc.) does not match the checksum calculated.	Run	Run	

: ON : 250ms ON / 250ms OFF : 500ms ON / 500ms OFF : 1s ON / 1s OFF : 2s ON / 2s OFF : OFF

CAUTION

If error cause is removed, error code remains except for error code 71 (battery error).
71 Error and OK LED blinking automatically disappear if battery is replaced to new one.

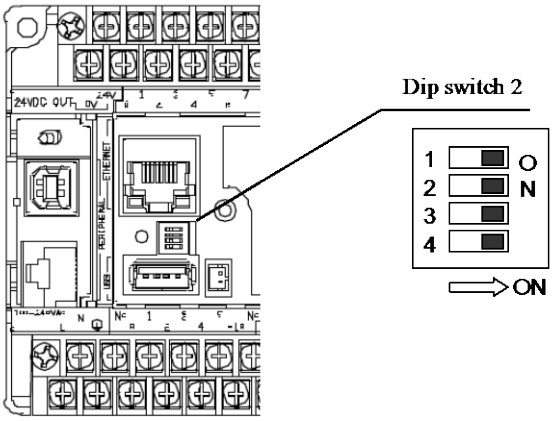
(*1) OK LED is not lighting up

If error code 13 is detected, OK LED is not lighting up and it is not possible to communicate with EHV-CODESYS because the system program or boot project is failure. If cycling power does not solve the problem, boot project could be failure. In this case, it is possible not to load boot project from flash memory.

< Unloading boot project >

(1) Remove power from the PLC

(2) Slide No.1 to 4 of the dip switch 2 to the right side (ON).



(3) Supply power to the PLC and check if OK LED is lighting up.

(4) Login and create boot project to restore your system.

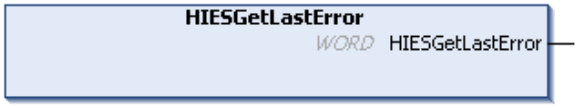
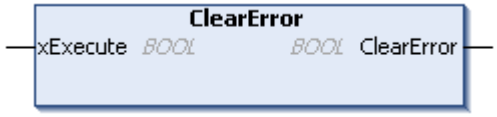
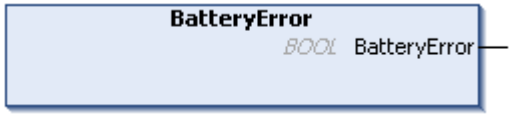
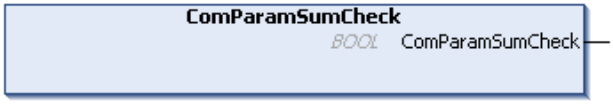
(*2) 21 Error

If power failure time of the PLC exceeds 12 hours, data in variables configured as RETAIN (backup by optional battery or capacitor in the unit) are not retained (undefined values). When PLC is powered up with RETAIN data undefined values, behavior of PLC can be selected in the setting [Retain mismatch] of EHV-CODESYS. If this setting is RUN (default), PLC starts with all RETAIN data initialized. (If RUN/STOP switch is in STOP position, PLC does not start regardless of the setting.)

Parameter	Type	Value	Default Value
IP Address	STRING	'192.168.0.1'	'192.168.0.1'
Subnet Mask	STRING	'255.255.255.0'	'255.255.255.0'
Default Gateway	STRING	'0.0.0.0'	'0.0.0.0'
Ethernet port Link speed / Duplex mode	Enumeration of BYTE	Auto Negotiation	Auto Negotiation
Change IP information	Enumeration of BYTE	No	No
Stop switch definition	Enumeration of BYTE	Reset warm	Reset warm
Digital Filter	BYTE(1..40)	4	4
OK LED blinking while battery error	Enumeration of BYTE	Enable	Enable
Retain mismatch	Enumeration of BYTE	Run	Run

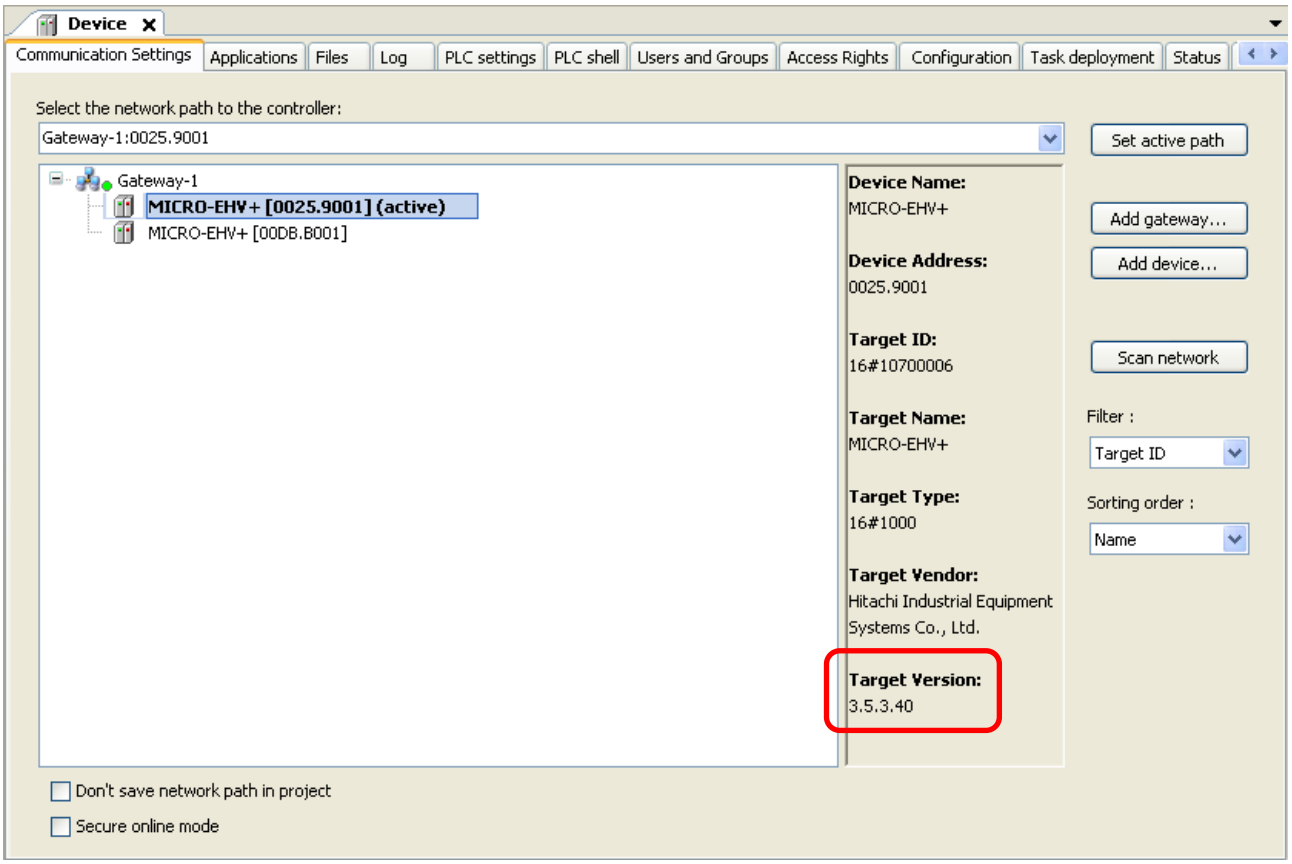
Error libraries

As for warnings (error code 70 to 78), special libraries called “CmpHitachiErrors_MV” are available as below. Use them in your application program if necessary. If it is not registered in your library repository, install CmpHitachiErrors_MV.library by choosing [Tools]-[Install library...].

Error code	Libraries (CmpHitachiErrors_MV.library)	Input	Output
all		-	Last detected error code (WORD)
all		Execution bit to clear error code (BOOL)	Result (BOOL)
70	 (FB)	-	70 Error bit (BOOL) Unit number (WORD) Slot number (WORD)
71		-	71 Error bit (BOOL)
77		-	77 Error bit (BOOL)
78		-	78 Error bit (BOOL)

3.26 Version

Firmware version (Target-Version) of your CPU is monitored in communication settings of Device as below.



The key functions supported by MICRO-EHV+ are listed below.

Category	Function	Firmware version	
		3.5.3.40	3.5.3.41
Communication	Modbus-TCP Master	-	-
	Modbus-TCP Slave	X	X
	Modbus-RTU Master	X	X
	Modbus-RTU Slave	X	X
	EtherCAT Master	-	-
	CAN	-	-
	Profibus	-	-
	Profinet	-	-
USB File System	Logging	-	X
	Web visualization	-	X
	Program upload / download	-	X
Expansion unit	Positioning Expansion Unit	-	-

X: Supported -: Not supported

MEMO

Chapter 4 Installation

For use in safety, avoid installing the PLC in the following locations.

- Excessive dusts, salty air, and/or conductive materials (iron powder, etc.)
- Direct sunlight
- Temperature less than 0°C or more than 55°C
- Dew condensation
- Humidity less than 5% or more than 95%
- Direct vibration and/or impact to the unit
- Corrosive, explosive and/or combustible gasses
- Water, chemicals and/or oil splashing on the PLC
- Close to noise emission devices

4.1 Installation

(1) Installing location and environment

(a) Install the PLC in Use the environment specified in the “2.1 General Specifications”.

(b) Mount the PLC onto the metal plate.

(c) Install the PLC in a suitable enclosure such as a cabinet which opens with a key, tool, etc.

(2) Installation of a unit

(a) Precaution when installing the unit

1] Fix the unit securely with screws in 2 places (M4, length 20mm (0.79in.) or longer) or DIN rail.

2] In order to keep within allowable ambient temperature range,

a) Ensure sufficient space for air circulation. (50mm (1.97in.) or more at top and bottom, 10mm (0.39in.) or more at right and left)

b) Do not install close to equipment that generates a lot of heat (heater, transformer, large-capacity resistance, etc.).

c) If ambient temperature is more than 55°C, install a fan or a cooler so that the ambient temperature becomes below 55°C.

3] Do not install inside a cabinet with high-voltage equipments installed.

4] Install 200mm (7.87in.) or more away from high-voltage wires or power wires.

5] Do not install the PLC upside down in vertical nor in horizontal.

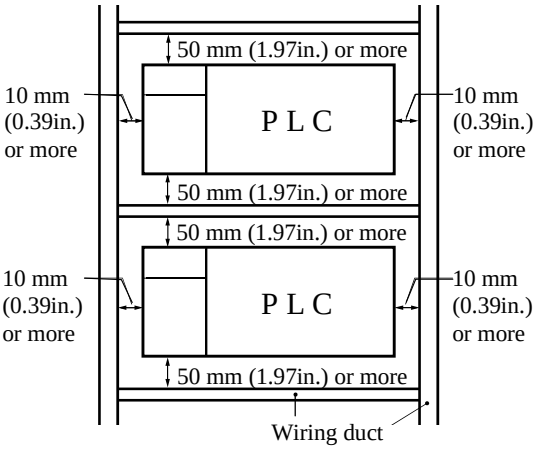


Figure 4.1 Amount of installation

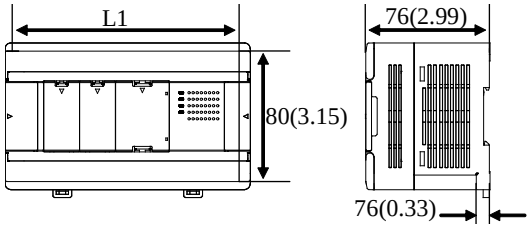


Figure 4.2 External dimensions

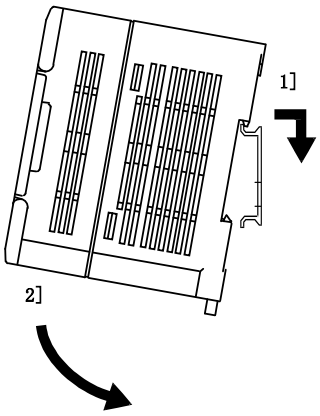
Dimensional table

Unit	L1
8/14/16 Points, Analog Expansion	85 (3.35)
40 Points Basic, 28 Points Expansion	140 (5.51)
64 Points Basic, Expansion	185 (7.28)

Unit: mm (in.)

(b) Mounting to a DIN rail

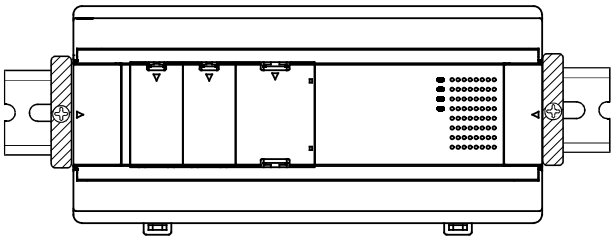
Attaching to a DIN rail



1] Hook the base unit to a DIN rail as shown left.
2] Press the base unit into the DIN rail until it clicks.

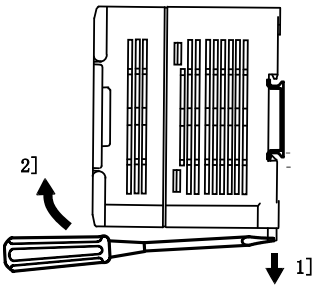
Note: Make sure the base unit is securely fixed after installation.

Fixing the unit



Install DIN rail clamps from both sides. (The unit could slide without clamps.)

Removing the unit from the DIN rail



1] Pull down the retaining clip on the bottom of the base unit.

2] Pull the unit away from the DIN rail.

4.2 Wiring

(1) Separation of power system

Several different power sources are used with PLC, such as main power of PLC, power for I/O signal and power for external devices. These power sources should be separated as much as possible.

If these power sources come from one power source, install transformers or noise filters to separate those power lines as much as possible.

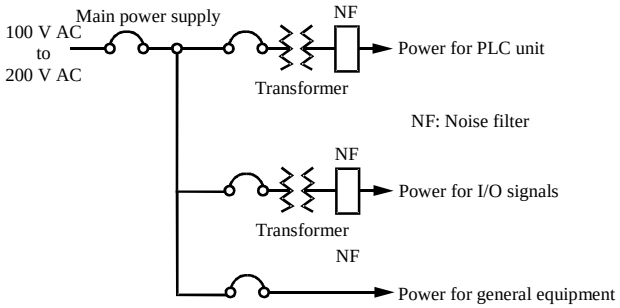


Figure 4.3 Example of power system diagram

(2) Fail safe

1] Construct an interlock circuit outside the PLC.

When the PLC power supply is turned ON/OFF, the lag time and the difference in the startup time between the basic unit's power and the expansion unit's power or the external power (particular DC power supply) for the PLC I/O signals may temporarily cause the I/O not to operate normally.

For this reason, apply the power to the expansion unit before the basic unit or apply the power to the basic unit and the expansion unit at the same time. In addition, the external power (particular DC power supply) for the PLC I/O signals should be applied before the PLC units.

Also, it is conceivable that a fault in the external power and a failure in the PLC unit lead to abnormal actions. To prevent such actions from causing abnormal operation the entire system, and from a point of view of creating a fail safe mechanism, construct circuit such as an emergency stop circuit, the protect circuit, and the interlock circuit, for the sections that lead to a mechanical breakdown and accident from abnormal actions outside the PLC.

2] Install a lightning arrester

To prevent damage to equipment as a result of being struck by lightning, we recommend setting up a lightning arrester for each PLC power supply circuit.

MICRO-EHV+ series PLC detects power failures from a voltage drop of the internal 3.3 V DC power supply. For this reason, the load in the 3.3 V DC power of the unit is light, the 3.3 V DC is retained for a long time and operations may continue for more than 100ms. Therefore, when using the AC input unit, an OFF delay timer for coordinating with the internal 3.3 V DC is needed because the AC input signal turns off more quickly than the internal 3.3 V DC.

(3) Wiring to the power supply terminal

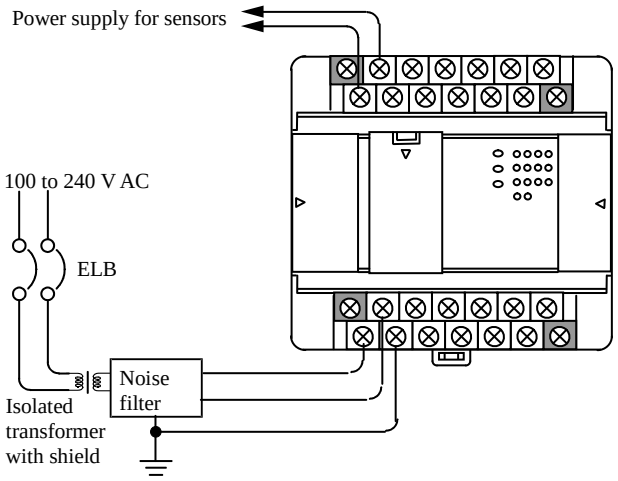


Figure 4.4 Wiring power diagram

(a) For power supply wiring, use a cable of 2 mm^2 ($0.0031in^2$.) or more to prevent a voltage drop from occurring.

(b) The function ground terminal (FE terminal) should use a cable of 2 mm^2 ($0.0031in^2$) or more and Class D grounding (100 Ω or less). The appropriate distance for ground cable is within 20m (65.62ft.).

1] Shared with instrumentation panel, relay panel grounding.

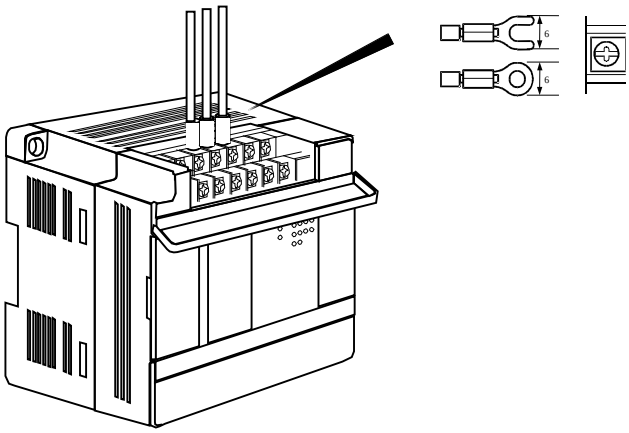
2] Avoid joint grounding with equipment that can generate noise such as high-frequency heating furnace, large power panel (several kW or more), thyristor exchanger, electric welders, etc.

3] Be sure to connect a noise filter (NF) to the power cable.

(c) The terminal screw size is M3. Recommended torque is from 0.5 to 0.6 N·m (4.4 to 5.3 in.-lbs).

(d) Use the same power supply system for the basic and expansion units.

(4) Wiring cable for I/O signals



The terminal screw size is M3.

Recommended torque is from 0.5 to 0.6 N·m (4.4 to 5.3 in.-lbs).

Use a crimp terminal with an outer diameter of 6mm (0.24in.) or less when using it.

Use only up to 2 crimp terminals in the same terminal. Avoid claming down more than 3 at the same time.

The terminal block supports 0.36 to 2.1 mm^2 (AWG22 to 14). If 2 crimping terminals are connected to one terminal screw, use 0.36 to 1.3 mm^2 (AWG20 to 16) cable.

Note: Use shielded cable for the relay output when corresponding to CE marking EMC command is necessary.

(5) Input wiring for the input terminal

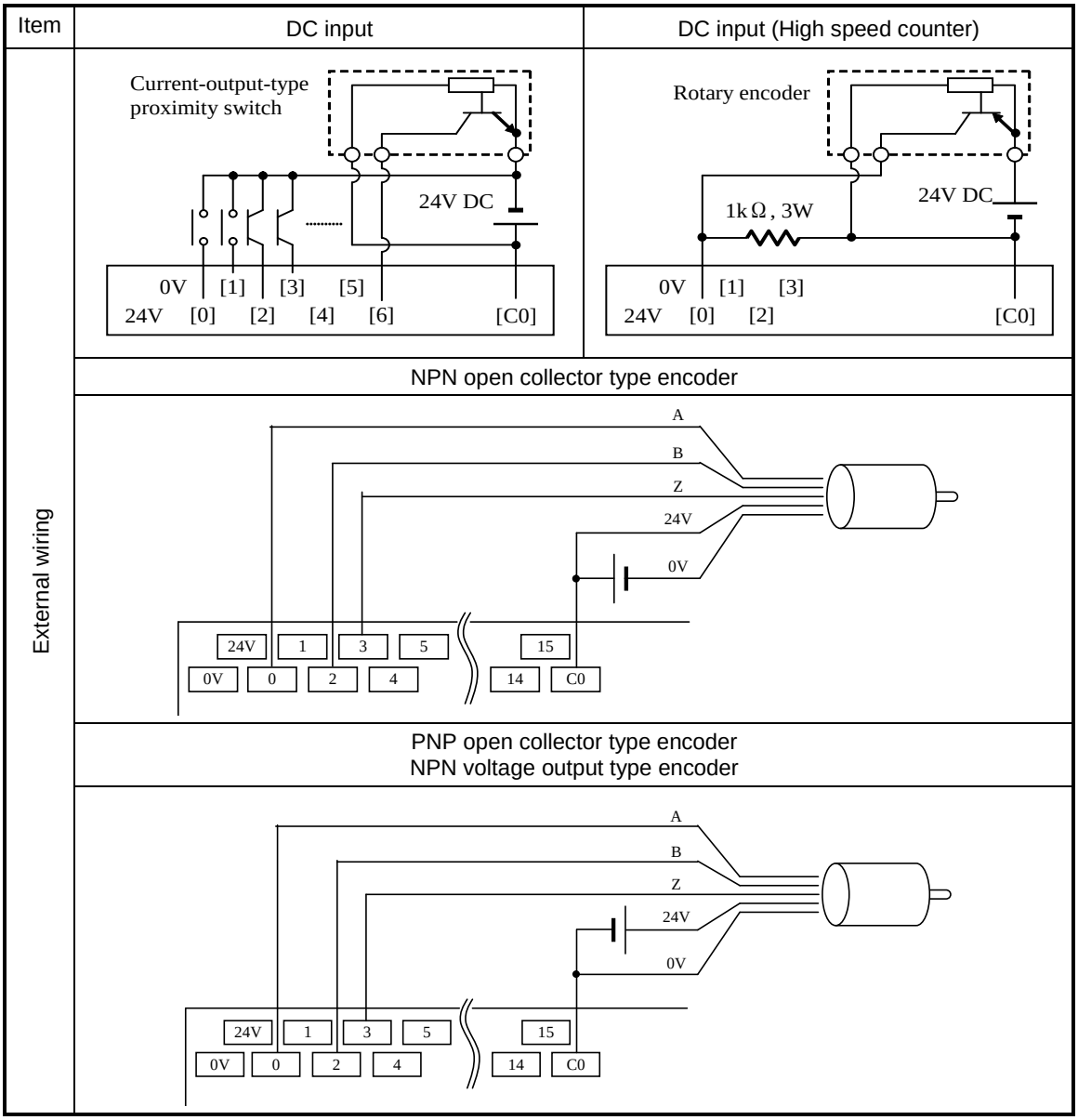


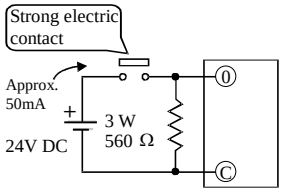
Figure 4.5 Input wiring

(a) DC input

1] When all input terminal (0, 1, …) and the common terminal (C) are loaded with 24 V DC, the input changes to ON, and approximately 8 mA current flow to the external input contacts.

2] For sensors such as a proximity switch and photoelectric switch, current-output-type (transistor open collector) can be directly connected. For voltage-output-type sensors, connect them to the input terminal after first going through the transistor.

3] Measures to prevent faulty contact in a strong electric contact



The current that flows to a contact when external contacts are closed is approximately 8mA. If the use of a strong electric contact cannot be avoided, add resistance as shown in the diagram at left and supply sufficient current to the contact to prevent a faulty contact.

4] Limit the wiring length within 30 m (98.43ft.).

5] Each common on the input terminal block is independent of each other. Make an external connection as needed.

(6) Output wiring for the output terminal

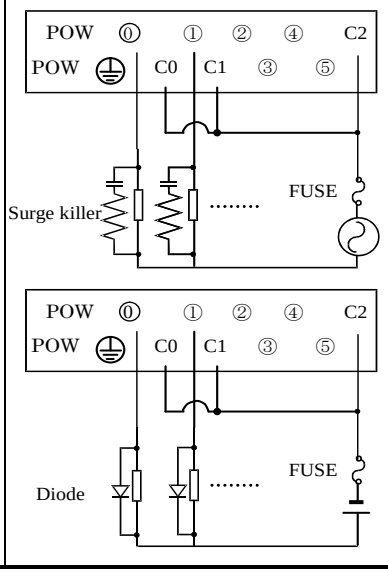
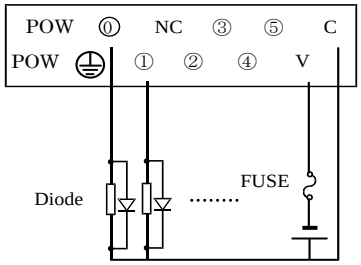
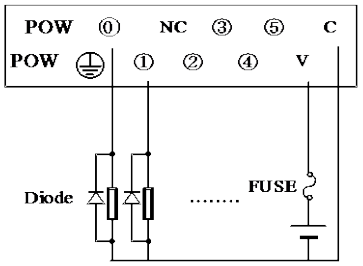
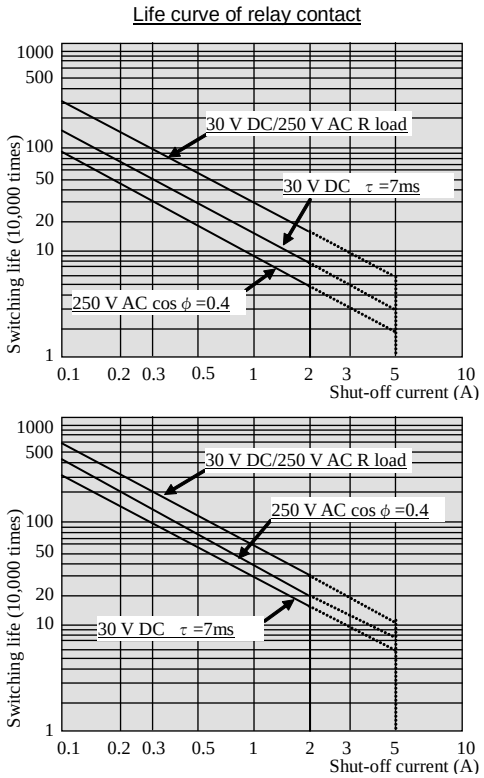
Item	Relay output (MV-***DR)	Transistor output (sink type) (MV*-**DT)	Transistor output (source type) (MV*-***DTP(S))
External wiring			

Figure 4.6 Output wiring

(a) Wiring for the relay output module

1] Life of relay contact



Life of the contact is also in squared reverse proportion to the current, so be aware that interrupting rush current or directly driving the capacitor load will drastically reduce the life of the relay.
When switching is done with high frequency, use a transistor output unit.

Above: 40/64 points basic unit
40/64 points expansion unit
16 points expansion unit

Below: other than listed above

2] Surge killer

For inductive load, connect a surge killer (capacitor 0.1μF, + resistance of around 100 Ω) in parallel to the load. Also, for DC load, connect a flywheel diode.

3] Fuse

A fuse is not built in this module. Install a 6A fuse in the common to prevent the external wiring from burning out. Install a 2A fuse in each independent contact output circuit.

(b) Wiring for the transistor output terminal

1] Flywheel diode

For inductive load, connect a flywheel diode in parallel.

2] V and C terminals

Always connect a V terminal and C (common) terminal. If the module is used without connecting these terminals, the internal flywheel diode does not function and there is a risk that the module will malfunction or breakdown.

3] Fuse

A fuse to prevent the external wiring from burning out is not built. So it is recommended to install a fuse for preventing the external wiring from burning out, but this does not protect internal transistor elements. Therefore, note that these elements are destroyed when the external load is short-circuited. Please contact us for repair if the external load short-circuits.

(7) Wiring for the analog I/O terminal

- Do not apply excess voltage to the analog input terminal beyond the rated input voltage. Similarly, do not subject the terminal to current that exceeds the rated input current. Connecting the analog input terminal to a power supply other than the specified types may cause damage to the product or burning or its internal components.
- For unused channels of the analog input, short the input terminals before use.
- When wiring the external lines of the analog terminal, route then through the shield cables while separating them form other power lines or signal lines subject to differential voltage. Shield cables must be grounded on one side. However, whether it is more effective to ground on one side or leave both sides open, depends on the noise environment condition in the actual use. Provide appropriate grounding based on the noise environment.
- Use separate piping for the AC power supply line and the signal/data lines.
- Wire the signal lines and data lines as close as possible to the grounded surface of the cabinet or a metal bar.

(8) Wiring to the unit terminal

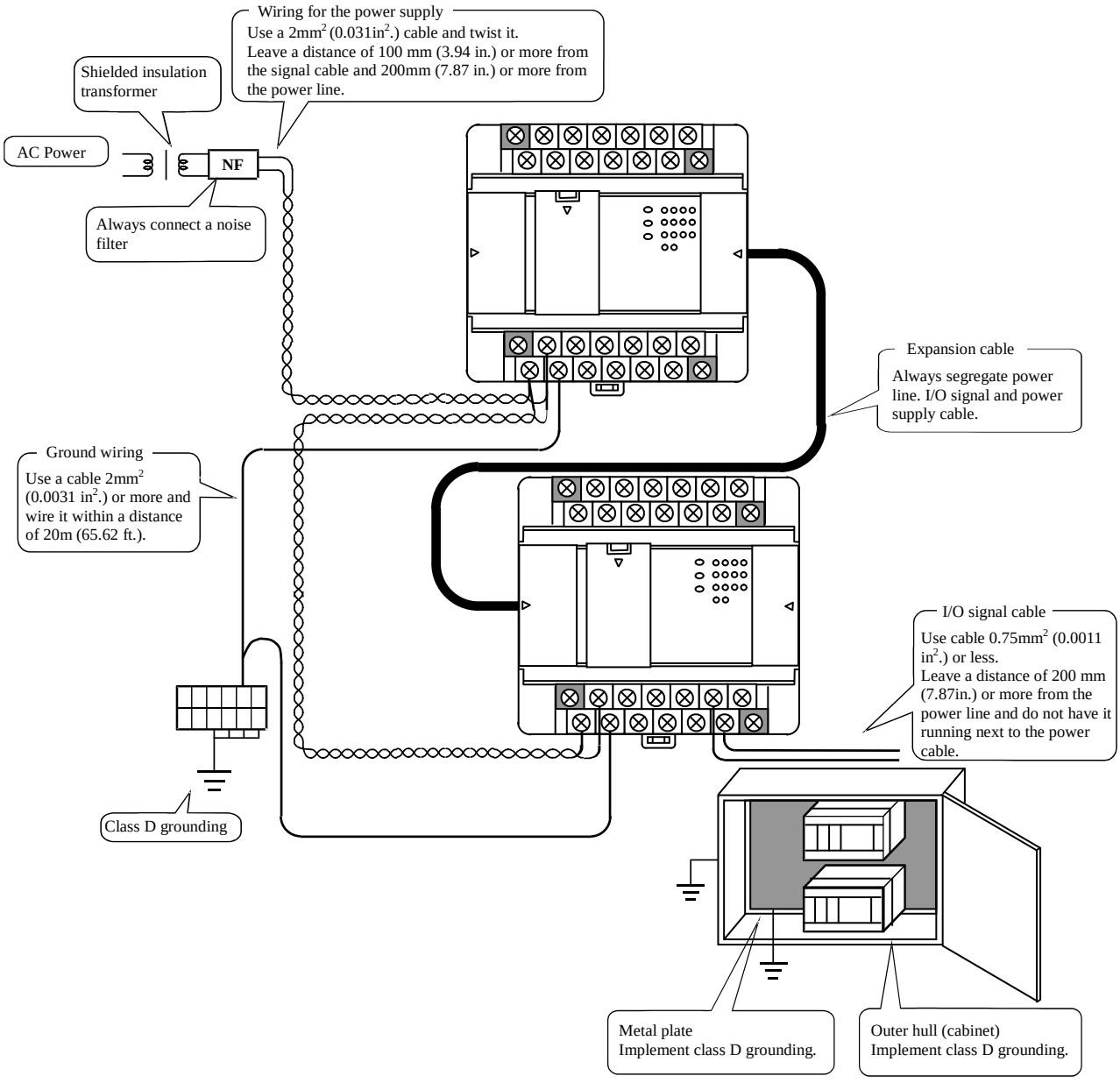


Figure 4.7 Example of wiring

Chapter 5 Maintenance

In order to use the PLC in the best condition and maintain the system to operate properly, it is necessary to conduct daily and periodic inspections.

5.1 Daily and Periodic Inspection

(1) Daily inspection

Verify the following items while the system is running.

Table 5.1 Items for daily inspection

Item	LED display	Inspection method	Normal status	Main cause of error
Unit LED display	POW	Visual check	ON	Power supply error, etc.
	RUN	Visual check	ON (Running)	OFF: Microprocessor error, memory error, etc. Refer to chapter 3 for further information.
	OK *1	Visual check	ON	OFF: Serious errors such as microprocessor error or memory error, etc. Refer to chapter 3. Blink: Battery error (71 error) *2

*1 MICRO-EHV+ indicates the error by the lighting pattern (ON / blink / OFF) of OK LED.

*2 If the power isn't supplied without replacing the battery after battery error detected (OK LED blinking), retain data and realtime clock data could be lost due to battery empty. If power off time is long enough, it is possible that a battery becomes empty while this power failure. In that case, retain data and realtime clock data would be already lost in the next power up.

(2) Periodic inspection

Turn off the power for the external I/O circuit, and check the following items once every six months.

Table 5.2 Items for periodic inspection

Part	Item	Check criteria	Remarks
Programming device to CPU	Check the operation of the programming device	All switch and display lamps work properly.	
Power supply	Check for the voltage fluctuations	85 to 264 V AC	Tester
I/O	Output relay life	Electrical life 200,000 times Mechanical life 20 million times	Refer to the relay contact life curve in the section 4.1 Installation.
	LED	Turns ON/OFF correctly	
	External power voltage	Within the specification for each I/O.	Refer to the Chapter 2 Specifications
Battery (Lithium battery)	Check voltage and life	OK LED blinks. Within 2 years after replacement.	
Installation and connecting areas	(1) All units are securely fixed. (2) All connectors fit snugly. (3) All screws are tight. (4) All cables are normal.	No defects	Tighten Check insertion Tighten Visual check
Ambient environment	(1) Temperature (2) Humidity (3) Others	0 to 55 °C 5 to 95 % RH (no condensation) No dust, foreign matter, vibration	Visual check
Spare part	Check the number of parts, the storage condition	No defects	Visual check
Program	Check program contents	Compare the contents of the latest program saved and CPU contents, and make sure they are the same.	Check both master and backup.

5.2 Product Life

The lifetime of electrolytic capacitors used in the power module is limited. If the lifetime is exceeded, performance of product is not guaranteed. Be sure to conduct inspection and maintenance as follows.

(1) Power module

Many electrolytic capacitors are used in the power module. It is said that lifetime of electrolytic capacitor would be half when ambient temperature increases 10 °C.

If lifetime of electrolytic capacitor is exceeded, output power becomes unstable especially when output current is high due to many point of outputs are activated for example.

Prepare spare units with considering 5 years lifetime in case ambient temperature is 30°C. For longer lifetime, take account of installation location in terms of temperature and air circulation around the unit.

(2) Battery

A battery to maintain realtime clock data and retain memory is available. Be noted following points about lifetime of battery.

- The battery life as shown below is total power failure time of PLC.
- When OK LED blinks, replace a battery to new one.

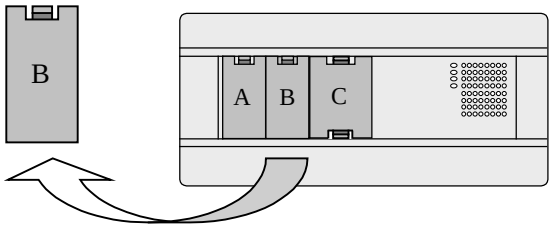
As a guideline, replace a battery every two years even when the total power failure time is less than the guaranteed value shown in the table.

Battery life (Total power failure time) [year]	
Guaranteed value (MIN) @55°C	Actual value (MAX) @25°C
5	10

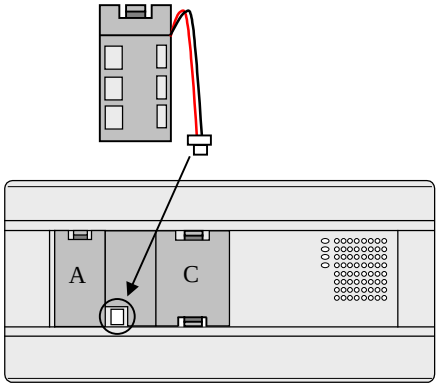
How to install the battery

1] Prepare a new battery (MV-BAT).

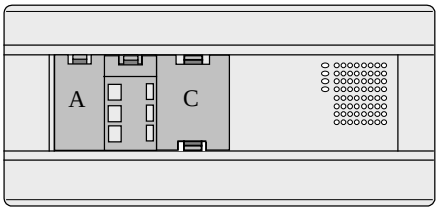
2] Remove the cover B.



3] Insert the battery connector into the connector on the unit.



4] Attach the battery cover together with battery to the unit.



*: If replacing the battery without power supplied, power off time should be less than 30 minute.

DANGER

Precaution when handling the battery.

Use MV-BAT for the new battery. Be careful because a false replacement may cause the battery to explode.

Do not connect + and – of the battery reversely, charge them , take them apart, heat them, throw them into the fire, short them.

CAUTION

Disposal (collection) of the battery

Old battery should be individually put in plastic bag or similar (to prevent short circuit and a disposal company should be requested to dispose of them.

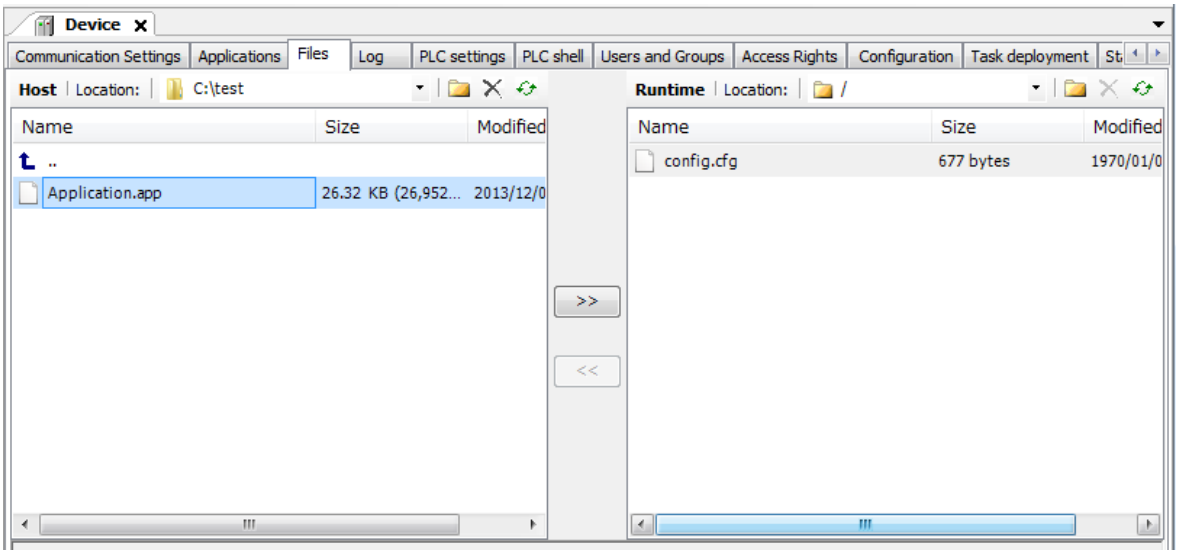
MEMO

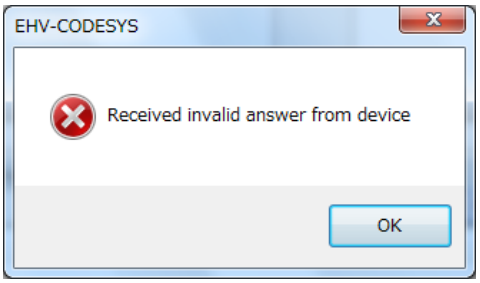
Appendix　Known Restrictions

Below restrictions are known in EHV-CODESYS V3.5 SP3 Patch6 and MICRO-EHV+ V3.5.3.41 or older.

1. Copying a file in [Files] window

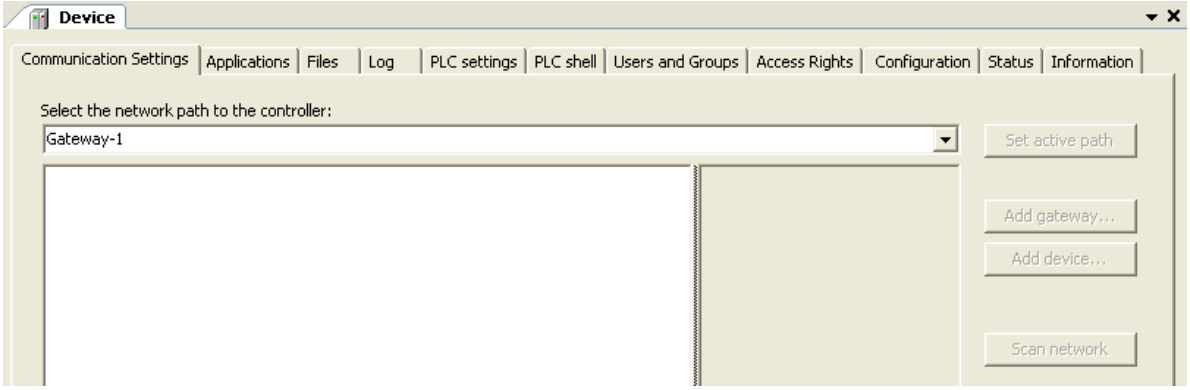
As an optional function, it is possible to send files in [Files] tab in Device window as below, but it does not work properly with MICRO-EHV+. If attempting to copy a file from PC to PLC in [Files] tab of [Device] window, it fails with an error message as below. Instead of this function, use “Login” or “Create boot project”.





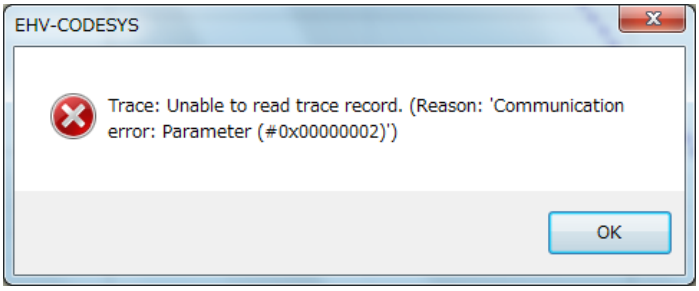
2. Add gateway button

If you click [Scan network] and delete the gateway before scanning completed, warning dialog box appears. After clicking [OK], [Add gateway] button will be deactivated. Choose [Add gateway] in right mouse click menu or close and open the Device window to enable [Add gateway] button again.



3. Trace

If RUN/STOP switch is toggled from RUN to STOP while trace monitoring, following error message appears although it is no problem practically. Click OK and choose “Download Trace” to restart. This message appears only when the Stop switch definition is configured as Reset warm.



4. Cable disconnection

Do not disconnect communication cable while program file or config.cfg file is being downloaded. Otherwise it could fail to establish online communication. In this case, cycle power and login again.

5. Power cycling

If MICRO-EHV+ without backup capacitor charged (more than 12 hours of power failure) is powered up and cycled power just a few seconds after powered up, the PLC could fail to start the system program correctly (OK LED is not lighting up). Login and execute “Reset origin” to reset the PLC.

6. Error code 20

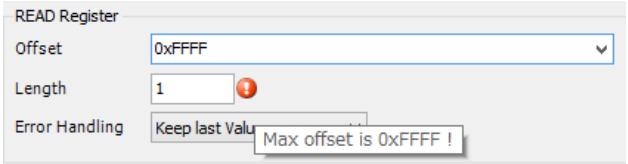
Error code 20 is not supported in MICRO-EHV+ V3.5.3.41 or older.

7. Modbus-TCP/RTU slave

If unsupported function code is sent from a master to MICRO-EHV+ as a slave, MICRO-EHV+ does not respond any data although an exception response must be sent back according to Modbus protocol.

8. Modbus register address 0xFFFF

When Modbus-RTU master is used, available register address must be from 0x0000 up to 0xFFFF however, register address 0xFFFF is not allowed to enter.



9. Modbus-RTU master FC 05 and FC 15

When function code 05 (Write Single Coil) or 15 (Write Multiple Coils) is used in Modbus-RTU master, default value in I/O mapping table does not work. Be sure to write value (TRUE or FALSE) in user program.

10. Modbus-TCP slave

If MICRO-EHV+ is reset (Reset warm/cold operation) during Modbus-TCP communication, it takes about 60 seconds to restart communication due to limitation of TCP protocol stack.

If Modbus client does not respond properly to closing command from MICRO-EHV+ with reset operation, status indication in EHV-CODESYS shows green circle although communication stops. The indication turns to red triangle 60 seconds after the reset operation, but actual communication status is ready-to-start.

MEMO