HITACHI PROGRAMMABLE AUTOMATION CONTROLLER

HX Series

APPLICATION MANUAL (Software)

(SERVICE MANUAL)

NJI-638(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 PAC only, any damage caused to third party equipment by malfunction of the PAC is not covered by the warranty.

○ Repair

Any investigation or repair after the warranty period cannot be covered as free of charge. Also any faults caused by above (1) to (4), will be charged for its repair (or for its investigation), even if the product is within the warranty period. In case of any contact, please ask your supplier or local Hitachi distributor. (Depending on failure part, investigation may not be possible to apply)

○ 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 PAC
・PAC system engineer
・Person handling PAC
・Manager after installing PAC

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 : Cases where if handled incorrectly a dangerous circumstance may be created, resulting in possible death or severe injury.

CAUTION : Cases where if handled incorrectly a dangerous circumstance may be created, resulting in possible minor to medium injury to the body, or only mechanical damage

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

In any case, they both contain important information, so please follow them closely.

Icons for prohibited items and required items are shown blow:

: Indicates prohibited items (items that may not be performed). For example, when open flames are prohibited, is shown.

: Indicates required items (items that must be performed). For example, when grounding must be performed, is shown.

1. About installation

⚠ CAUTION
• Use this product in an environment as described in the catalog and 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. • Perform installation according to this manual. If installation is not performed adequately, it may result in dropping, malfunction or an operational error in the unit. • Do not allow foreign objects such as wire chips to enter the unit. They may become the cause of fire, malfunction or failure.

2. About wiring

REQUIRED

- Always perform grounding (FE terminal).
  If grounding is not performed, there is a risk of electric shocks and malfunctions.

⚠ CAUTION

- Connect power supply that meets rating.
  If a power supply that does not meet rating is connected, fire may be caused.
- The wiring operation should be performed by a qualified personnel.
  If wiring is performed incorrectly, it may 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.
- Structure the emergency stop circuit, interlock circuit, etc. outside the PAC.
  Damage to the equipment or accidents may occur due to failure of the PAC.
  However, do not interlock the unit to external load via relay drive power supply of the relay output module.

⚠ CAUTION

- When performing program change, forced output, RUN, STOP, etc., while the unit is running, be sure to verify safety.
  Damage to the equipment or accidents may occur due to operation error.
- Supply power according to the power–up order.
  Damage to the equipment or accidents may occur due to malfunctions.

⚠ CAUTION

- Use power supply unit of EH-PS series for supplying electric power.

⚠ CAUTION

- Do not connect DC power supply module EH-PSD to a master power circuit. Supply a power to EH-PSD through an appropriate isolation transformer less than up to 150VA by all means.

4. About preventive maintenance

DANGER
• Do not connect the +, - of the battery in reverse. Also, do not charge, disassemble, heat, place in fire, or short circuit the battery. There is a risk of explosion or fire.

PROHIBITED
• Do not disassemble or modify the unit. Electric shock, malfunction or failure may result.

CAUTION
• Turn off the power supply before removing or attaching module/unit. Electric shock, malfunction or failure may result.

Revision History

No.	Description of revision	Date of revision	Manual number
1	The first edition	2016.12	NJI-638(X)

Table of contents

Chapter 1 Introduction

1.1 Installation

1.1.1 System Requirements

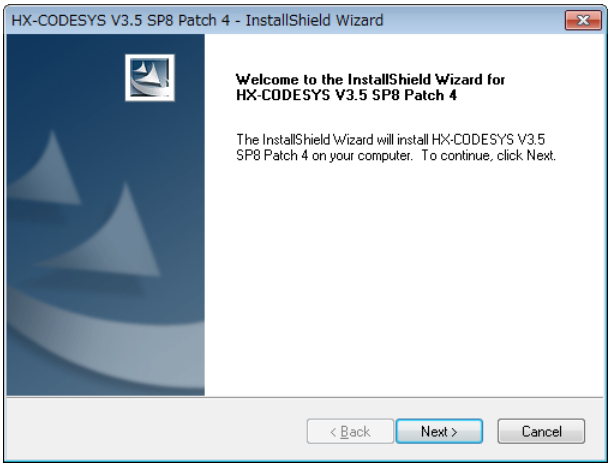
Recommended CPU : Pentium 1 GHz or higher
Memory : 1 GB or more RAM
Free disc space : 3 GB or more
Resolution : 1024 x 768 (XGA) or higher
Operating System : Windows XP SP3 / Vista / 7 (32 / 64 bit) / 8 / 8.1 / 10

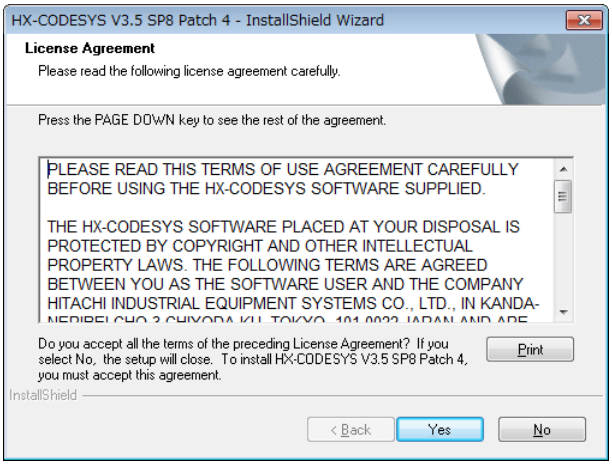
1.1.2 Installation of HX-CODESYS

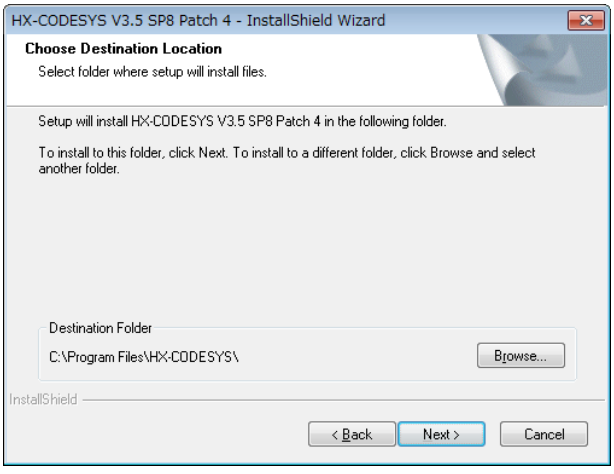
Before installation, shut down all other windows applications. If not, the installation may not be finished correctly.

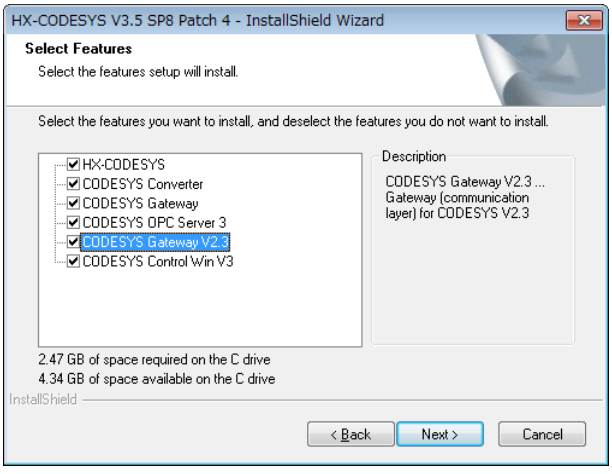
1. The installation wizard starts up automatically by double click [Setup_HXCODESYSV35SP8P4.exe] on HX-CODESYS installation DVD.

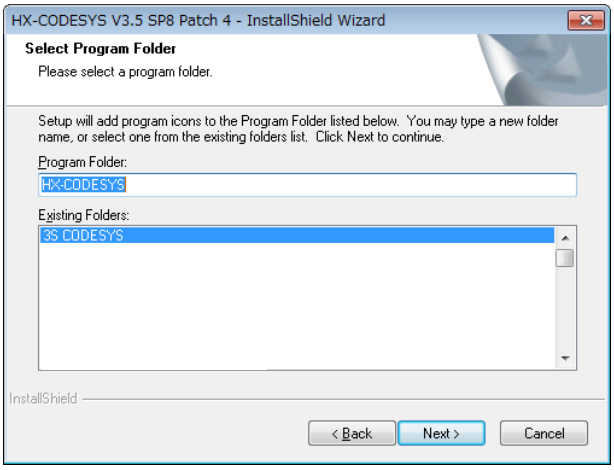
2. Follow the instructions

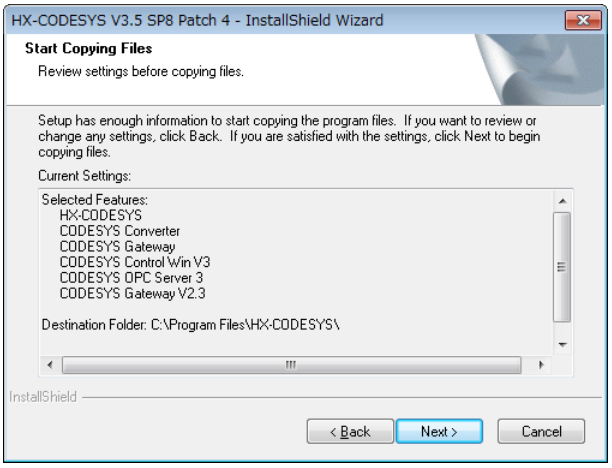


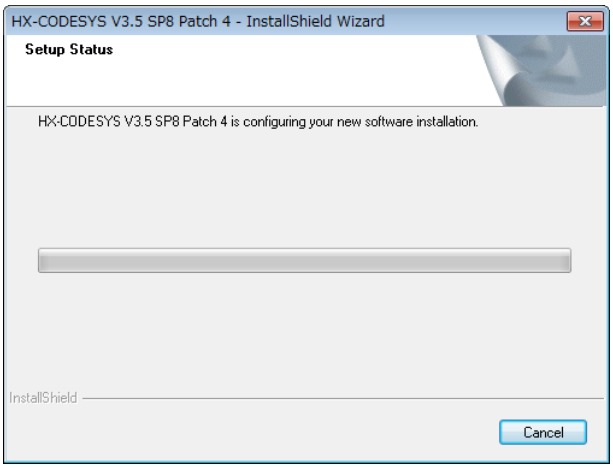


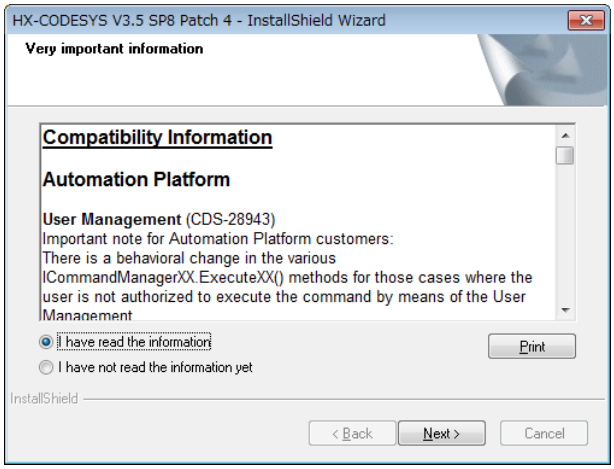


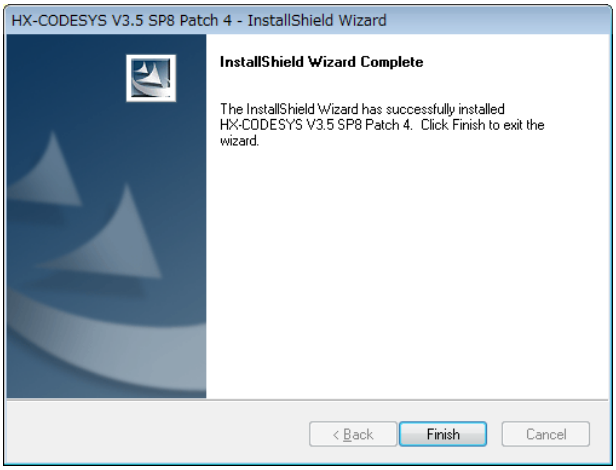








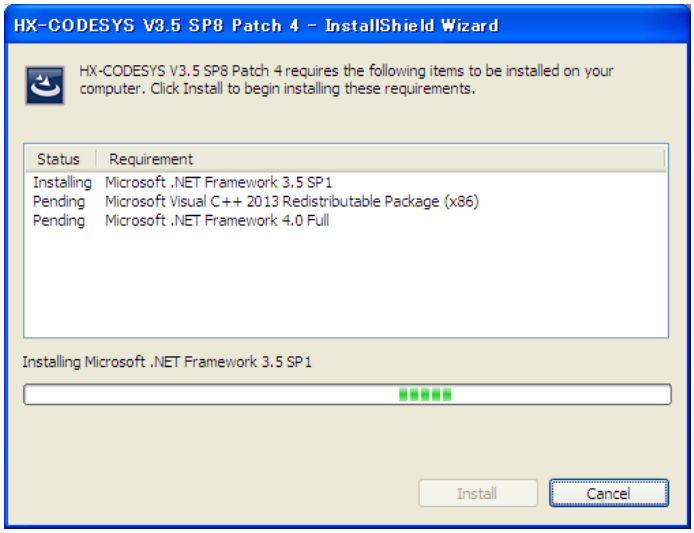




It takes about 30 minutes to 2 hours to finish installation depending on the specifications of PC.

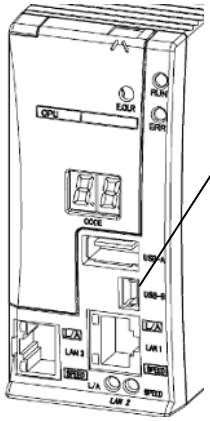
Note

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



1.1.3 Installation of USB Driver

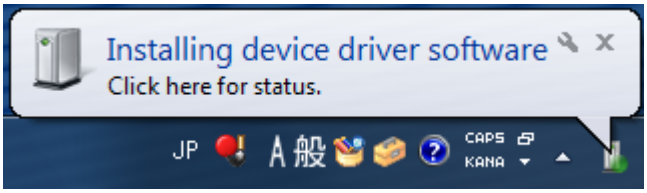
1. Plug USB cable to CPU module.



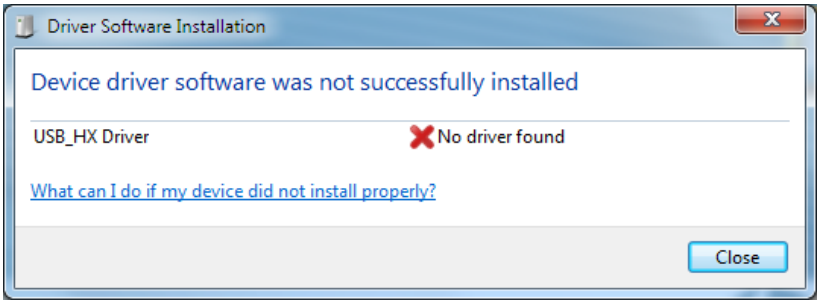
USB device port
(Type:mini B)

USB cable is not included in the product.
To prevent communication error by noise, prepare USB cable with ferrite core (A – mini B).

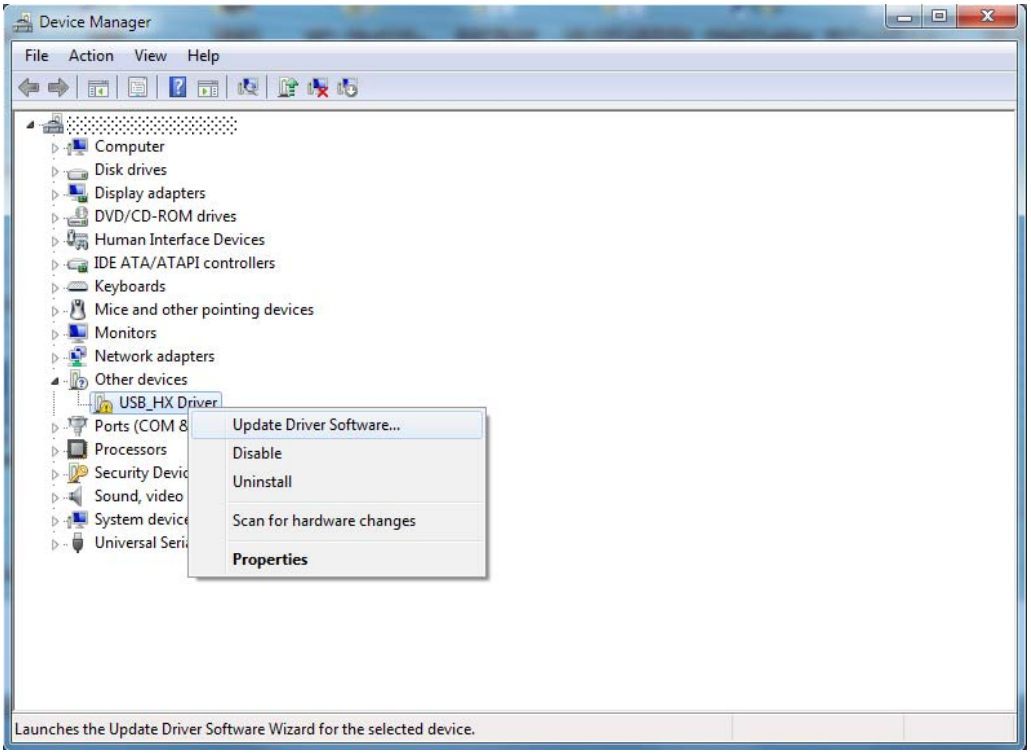
2. Popup window appears at right-bottom of screen. Click the popup window.



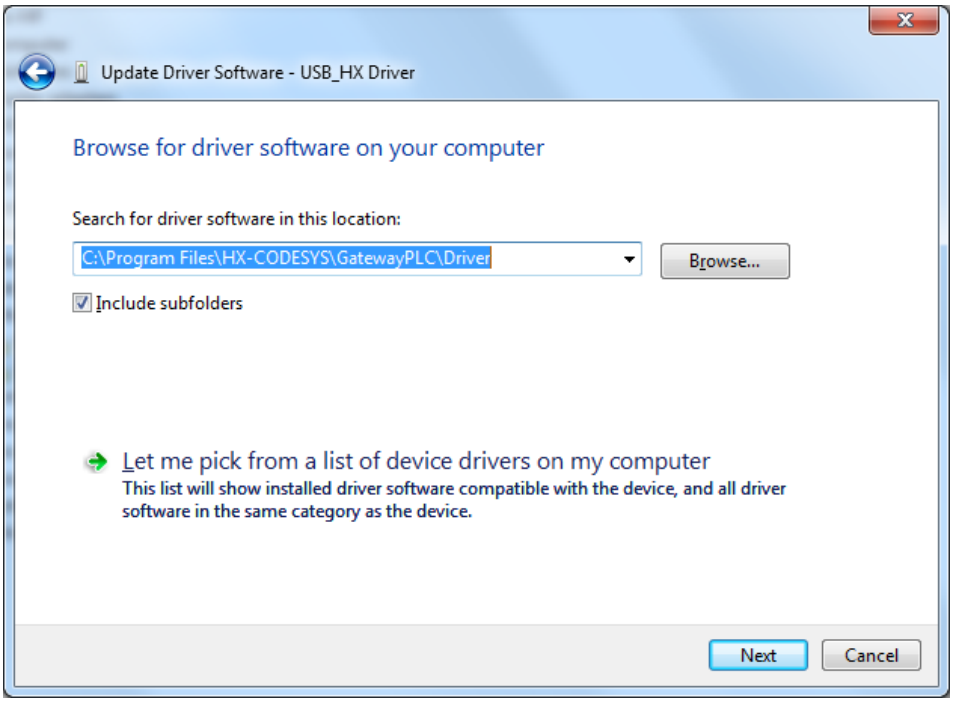
3. Click [Close] in this dialog (skip to search the driver in PC).



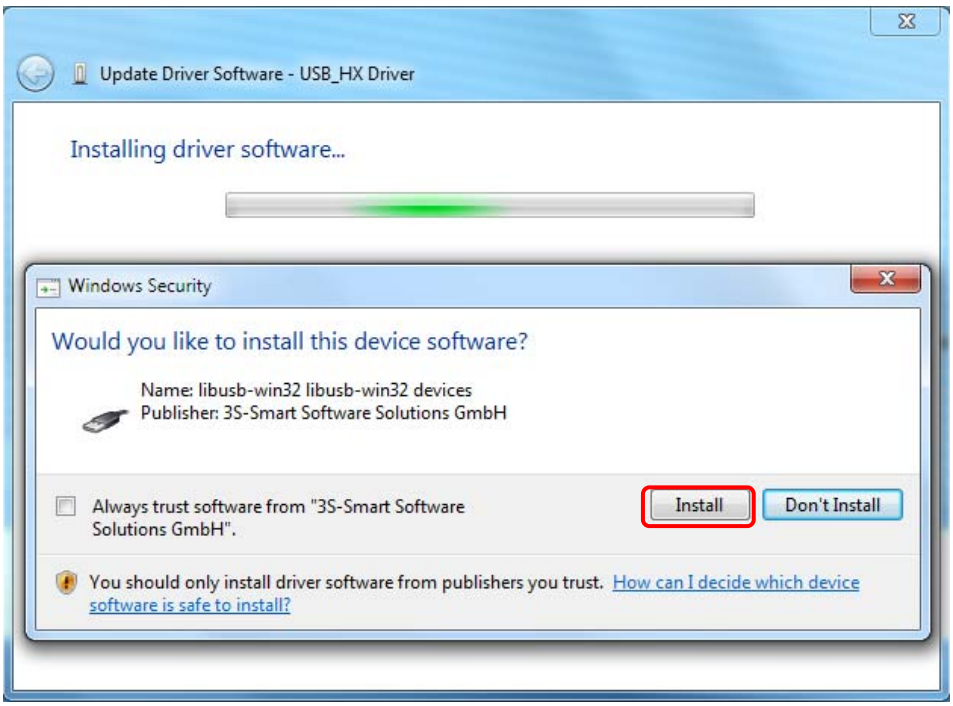
4. Open [Device Manager], right-click on [USB_HX Driver] in [Other devices], and choose [Update Driver Software…].



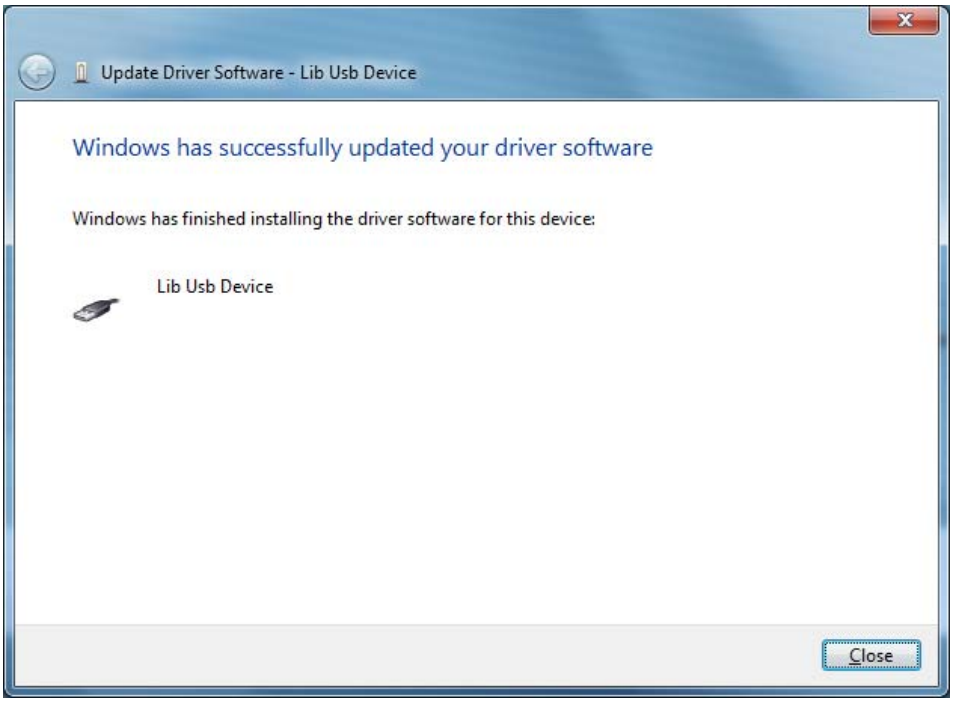
5. Enter the below path and click [Browse] to install the USB driver.

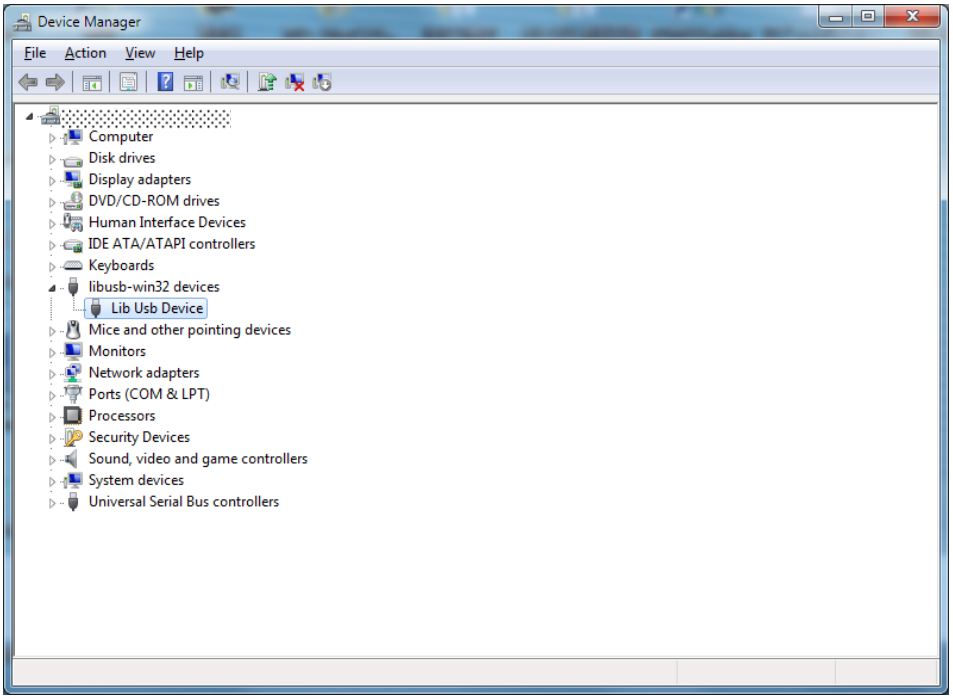


6. Click [Install].



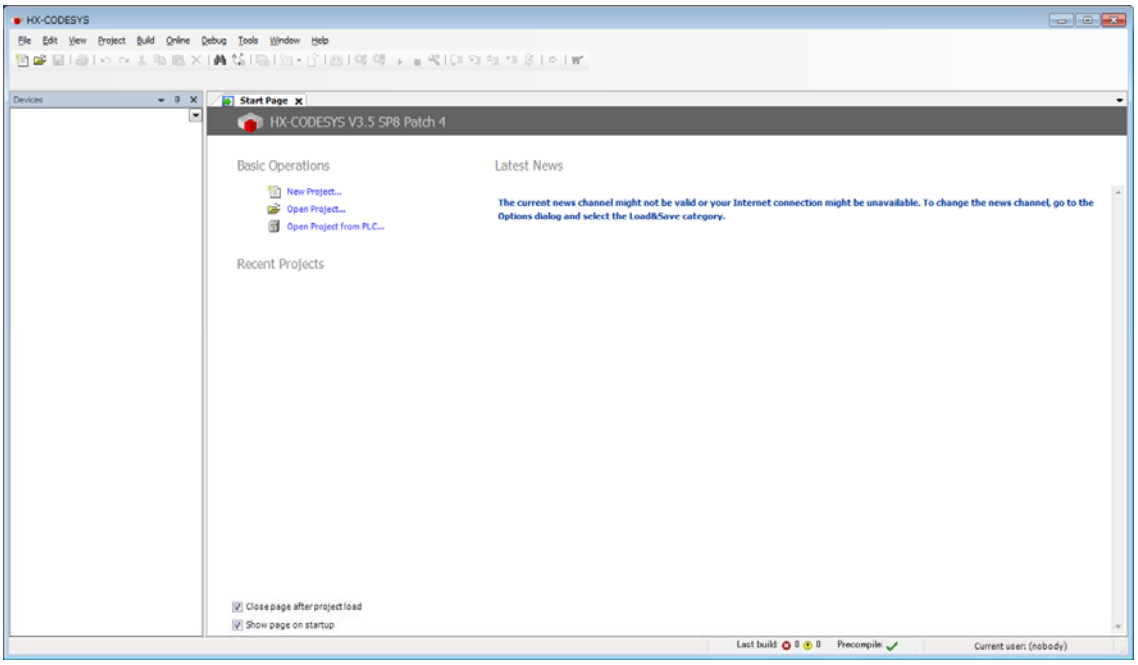
7. If the installation is finished successfully, below dialog appears.



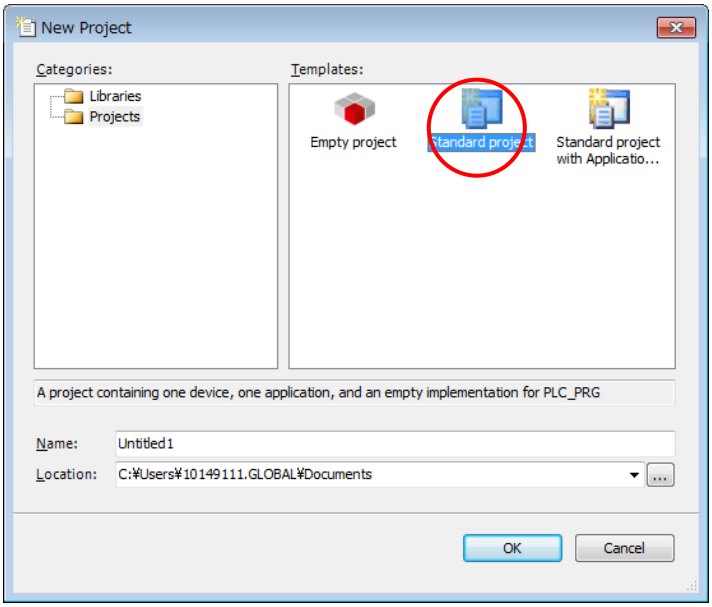


1.2 Startup

Choose [Start menu]-[All programs]-[HX-CODESYS]-[HX-CODESYS V3.5 SP8 Patch 4], then the start page is displayed.

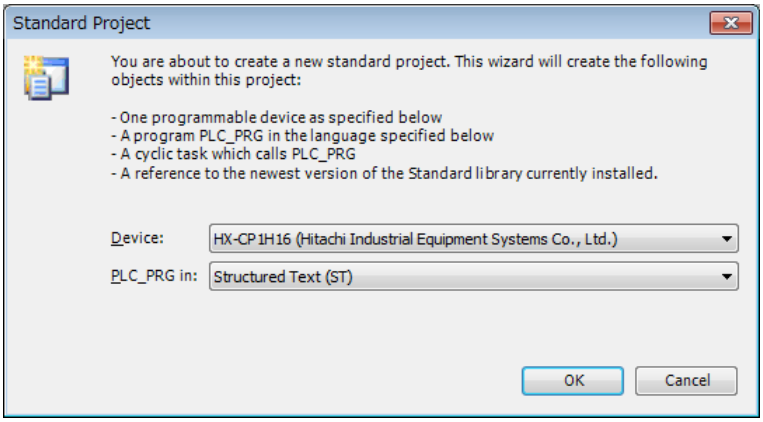


Click  icon or choose [File]-[New Project...] to create a new project file. Then New Project dialog box appears. Choose [Standard project], enter new file name, specify location and click [OK].

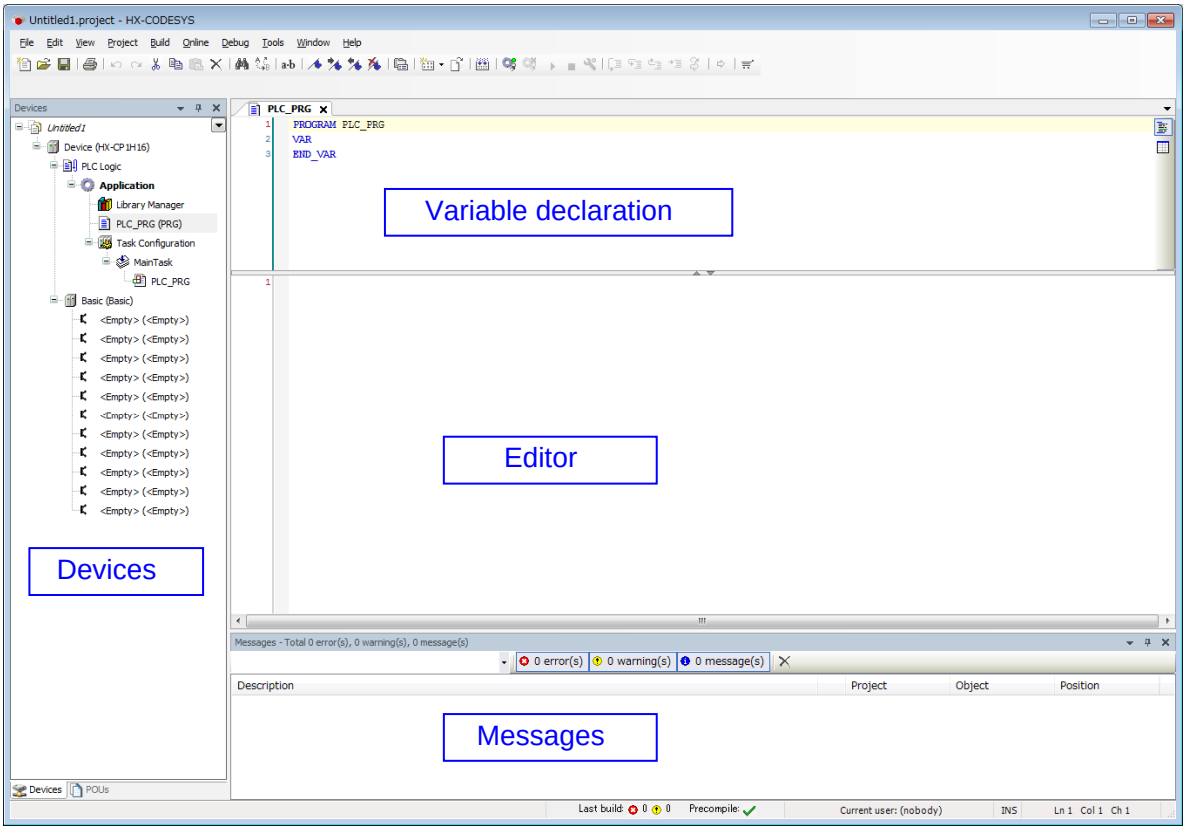


Choose CPU type and programming language and click [OK].
Available languages are as follows.

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



Initial layout of the project is shown like this.



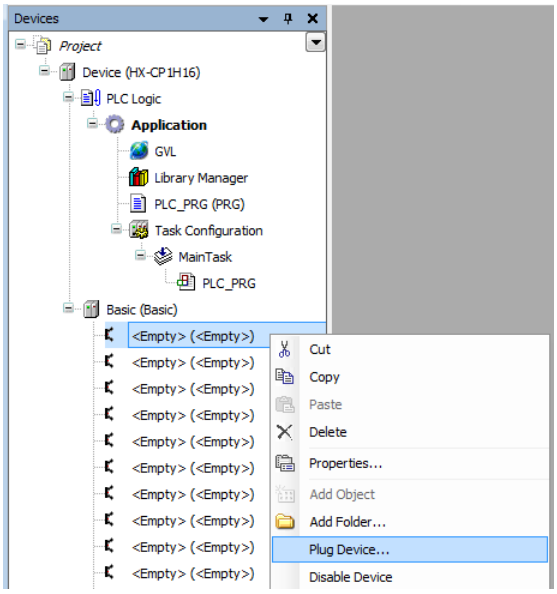
[Devices] and [Massages] window may not be shown at the first startup. They can be viewed with the menu [View]. If [Devices] (device tree) is behind the [POUs] tab, click [Devices] tab to show it. Double-click on POU (PLC_PRG) to open [Editor] and [Variable declaration].

Chapter 2 Programming

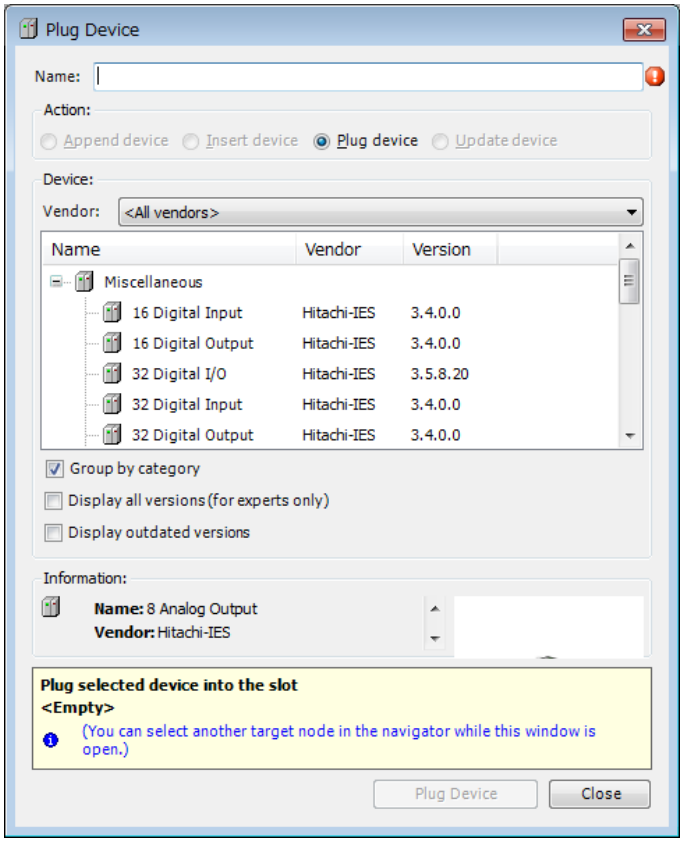
2.1 Plug Device (I/O Configuration)

2.1.1 Plug Device (I/O Configuration)

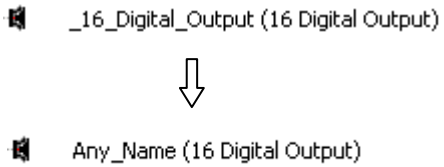
Right click on <Empty> slot and choose [Plug Device...].



Choose I/O module for each slot. The next slot can be configured by clicking next empty slot without closing the Plug Device window every time.



Module name can be renamed at [Properties] in right mouse click menu.



Configure I/O modules according to the list below.

Model names	Device Names	Slot position
EH-XD8, XD16, XDL16	16 Digital input	No restriction
EH-XA16, XAH16		
EH-XD32, XDL32, XD32E, XDL32E, XD32H, XD32S	32 Digital input	
EH-XD64	64 Digital input	
EH-YR8B, YR12, YR16, YR16D	16 Digital output	
EH-YT8, YT16		
EH-YTP8, YTP16, YTP16S		
EH-YS16		
EH-YT32, YT32E, YT32H	32 Digital output	
EH-YTP32, YTP32E		
EH-YT64	64 Digital output	
EH-YTP64		
EH-PT4	4 Analog input	
EH-AX44, AX8V, AX8H, AX8I, AX8IO	8 Analog input	
EH-AXH8M		
EH-AXG5M *1		
EH-TC8		
EH-RTD8		
EH-AY22, AY2H, AY4V, AY4H, AY4I *2	8 Analog output	
EH-AYH8M		
EH-AYG4M *2		
EH-CU, CUE	EH-CU/E	
EH-POS	EH-POS/4	
EH-LNK, OLNK, OLNKE, OLNKG	EH-LNK	Only 0 to 7 slot on basic base *3
EH-RMP2	EH-LNK	
EH-FLN2, FLN3	EH-FLN2/3	

*1 Set [8 Analog input] although this is 5-channel module.

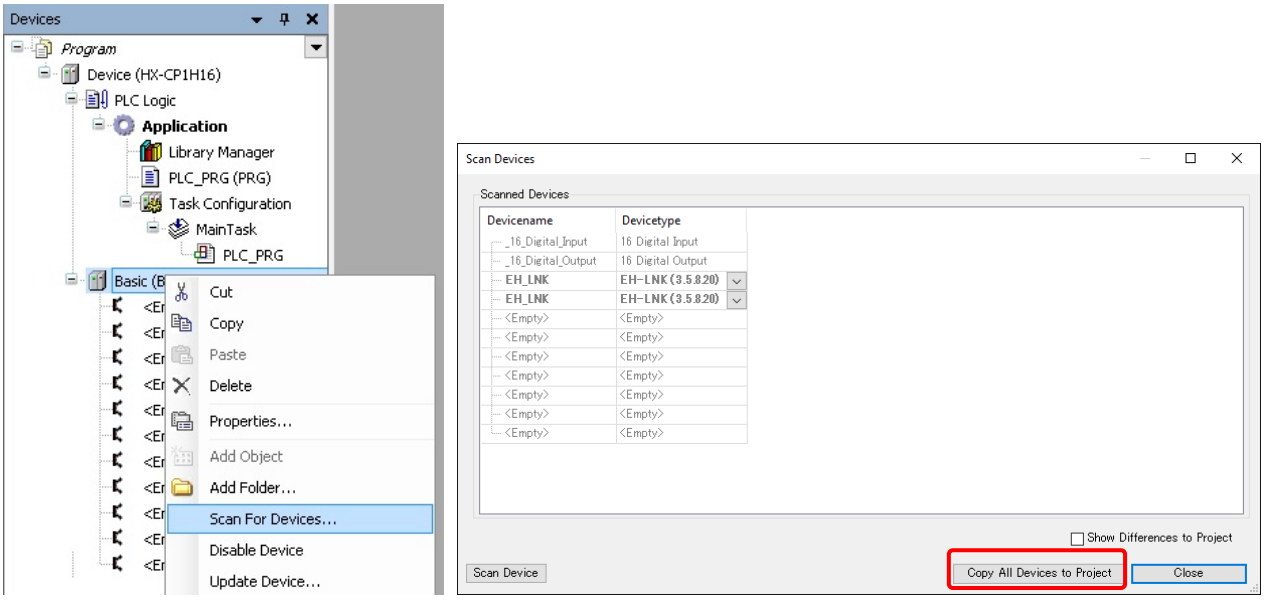
*2 Set [8 Analog output] although this is 2 or 4-channel module.

*3 If the module is mounted on another slot than slot 0 to 7, following error message appears in CPU log.

- EH-LNK is mounted on slot *. Allowed position for EH-LNK is slot 0 to 7.
- EH-FLN2/3 is mounted on slot *. Allowed position for EH-FLN2/3 is slot 0 to 7.

2.1.2 Scan for Devices

Instead of plugging I/O modules one by one, actual I/O module information can be read out from connected CPU.
Right click on basic or expansion base and choose [Scan For Devices...]. Then [Scan Devices] dialog appears. Click [Copy All Devices to Project]. This function works for chosen base only. If you have several expansion bases, repeat [Scan For Device] for each base.

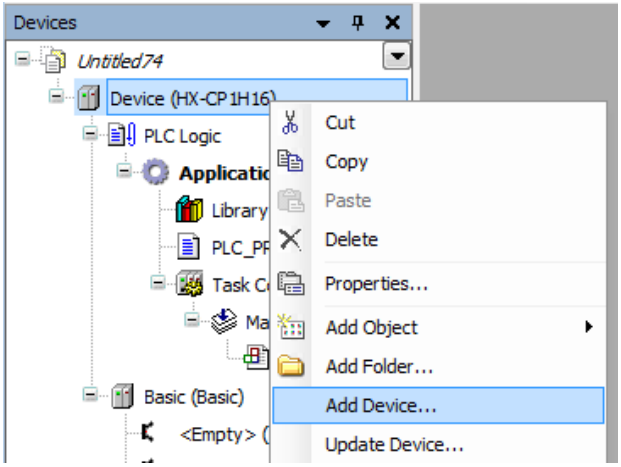


Note

This function is available only when logout (offline). But it is necessary to login once.
If you have never login, this function will not work properly.

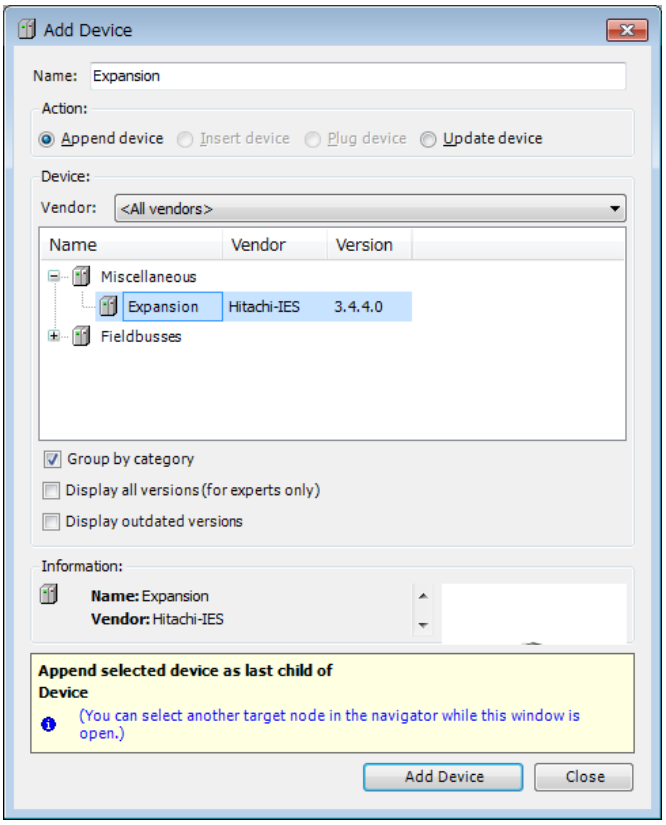
2.1.3 Expansion Unit

Instead of [Plug Device], choose [Add Device] to configure expansion units.



Select [Expansion] under Miscellaneous. HX-CPU allows to expand up to 5 expansion bases.

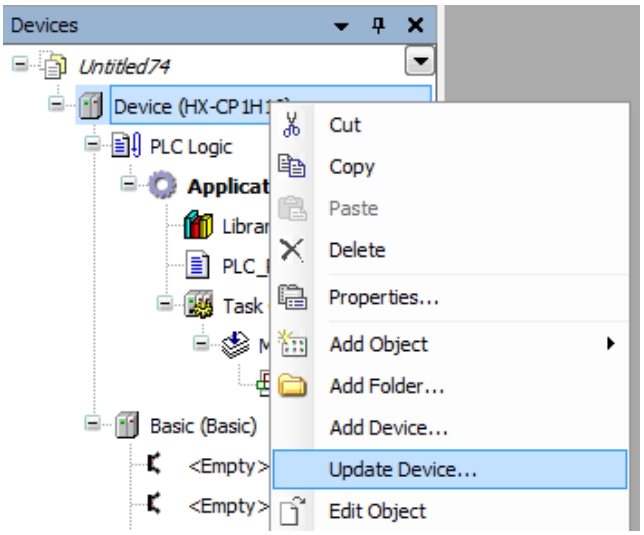
Set station number with ascending order from 1 to the nearest EH-IOCH2 to CPU.



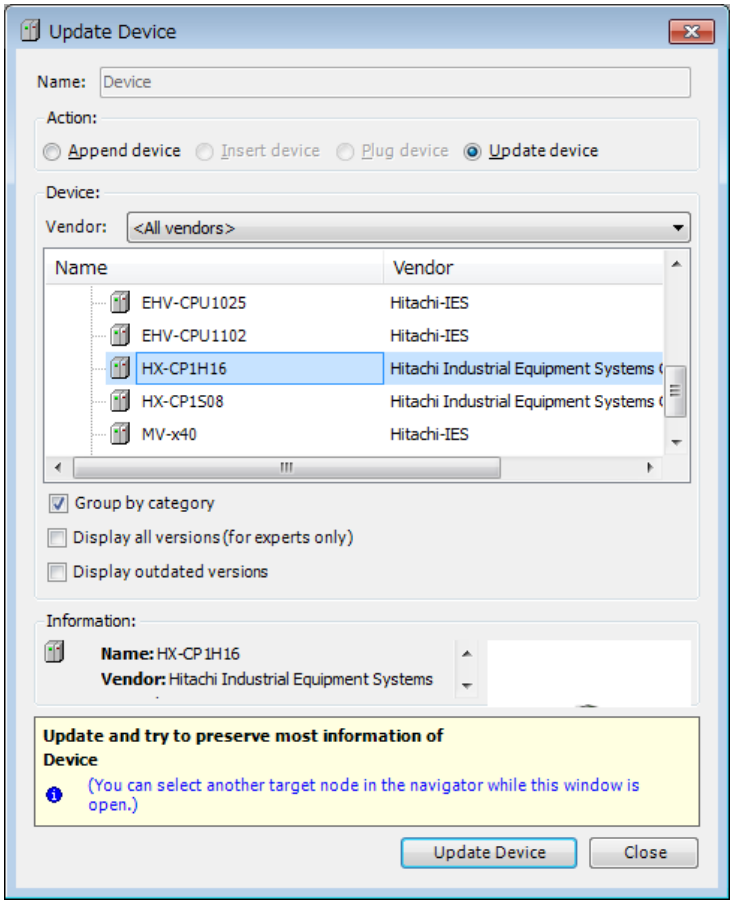
2.1.4 Update Devices

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.



Select CPU type, and click [Update Device].

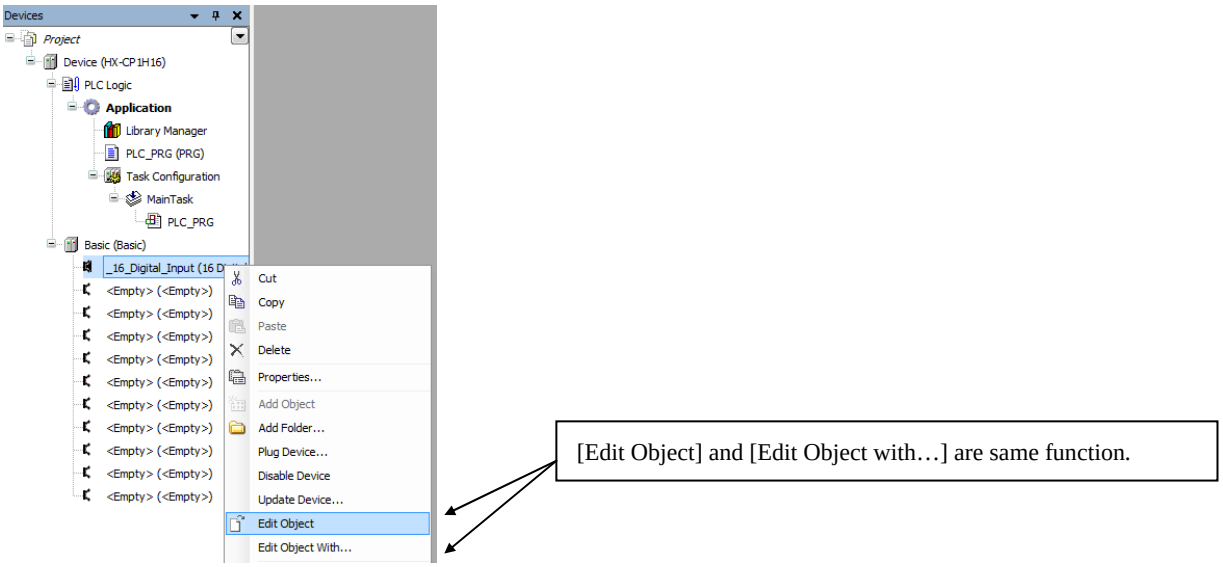


2.1.5 I/O Address

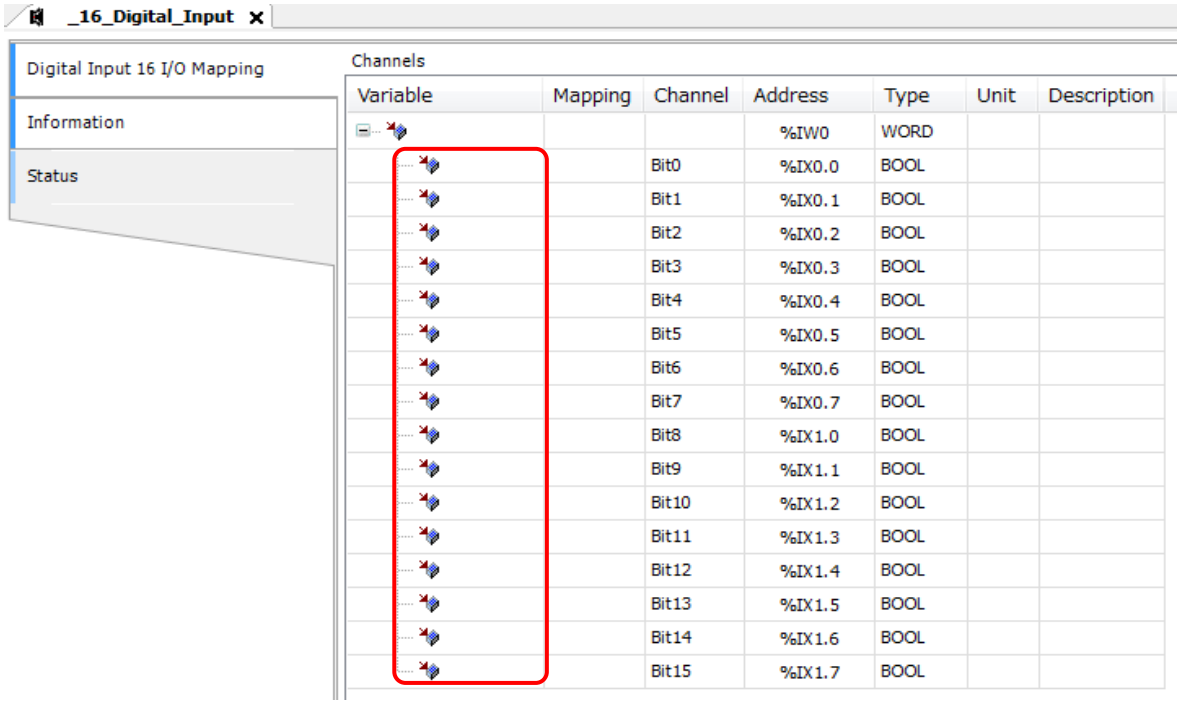
External I/O can be defined as Global variable or Local variable. Be sure to use global variable in normal use.

[Global variable]

Double click on plugged I/O module or right click and choose [Edit Object].



I/O-Bus Mapping window appears as below. Input variable name at the mapping table. These variables are used in the programming.

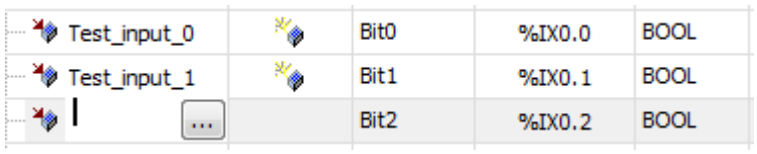


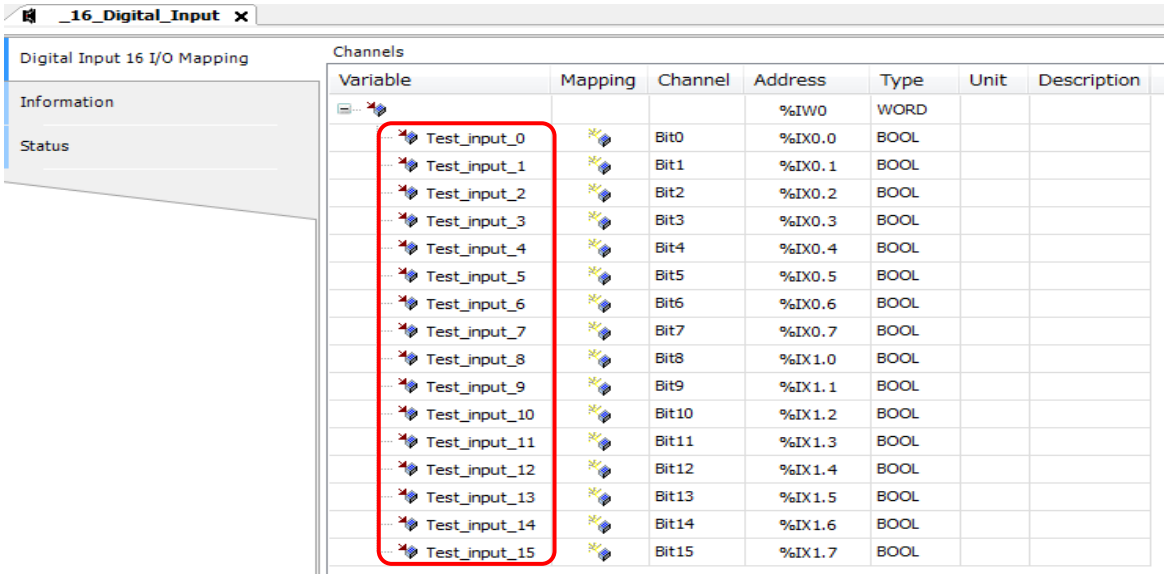
Variable	Mapping	Channel	Address	Type	Unit	Description
			%IW0	WORD		
		Bit0	%IX0.0	BOOL		
		Bit1	%IX0.1	BOOL		
		Bit2	%IX0.2	BOOL		
		Bit3	%IX0.3	BOOL		
		Bit4	%IX0.4	BOOL		
		Bit5	%IX0.5	BOOL		
		Bit6	%IX0.6	BOOL		
		Bit7	%IX0.7	BOOL		
		Bit8	%IX1.0	BOOL		
		Bit9	%IX1.1	BOOL		
		Bit10	%IX1.2	BOOL		
		Bit11	%IX1.3	BOOL		
		Bit12	%IX1.4	BOOL		
		Bit13	%IX1.5	BOOL		
		Bit14	%IX1.6	BOOL		
		Bit15	%IX1.7	BOOL		

Note

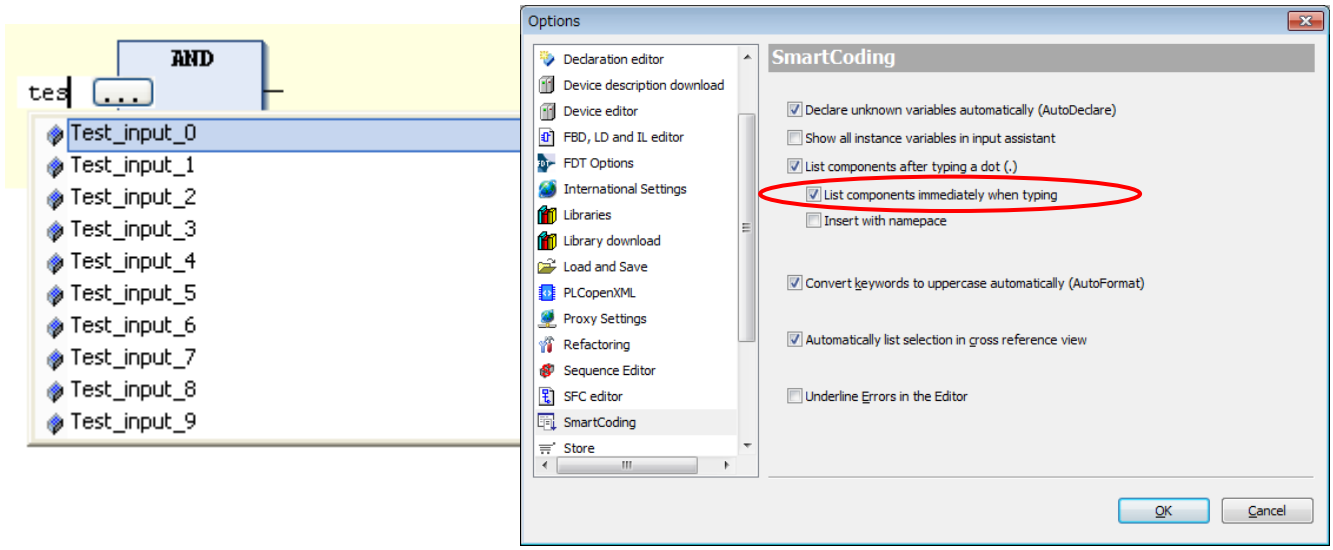
Available characters for variable names are only alphabet a to z, A to Z and number 0 to 9 and _ (underscore). The first character must not be numeric characters. Several words like BOOL, WORD, IF, FOR etc. are reserved.

Input any variable names in the field [Variable] according to your system.

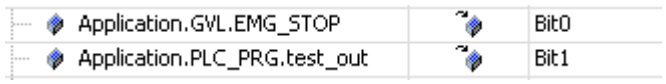




If [List components immediately when typing] in the menu [Tools]-[Options]-[Smart coding] is enabled, defined variable names will be automatically listed up when it is used in all POU with assist of auto-complete.



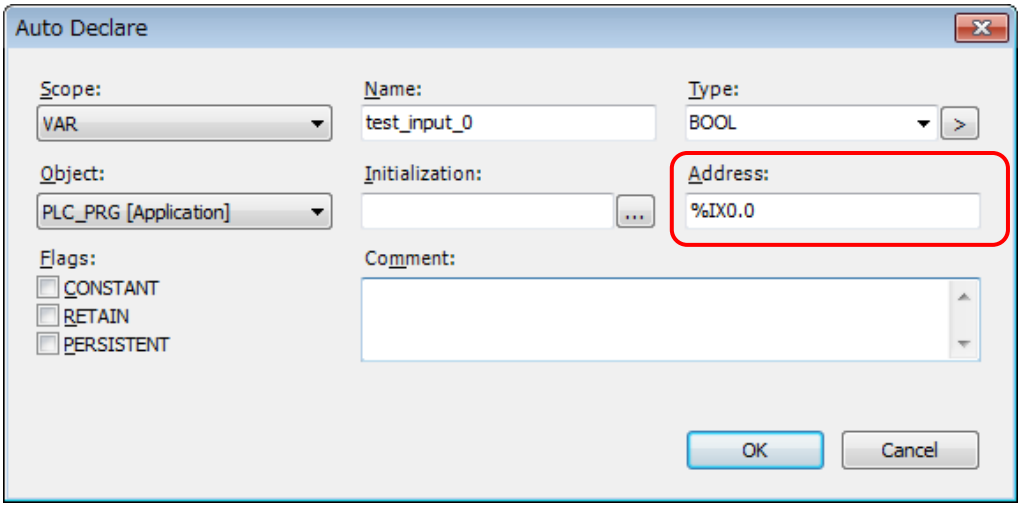
If a variable is already used (declared) in POU or global variable list, it can be taken by clicking ... icon in I/O mapping window. (... icon appears by clicking empty field.)



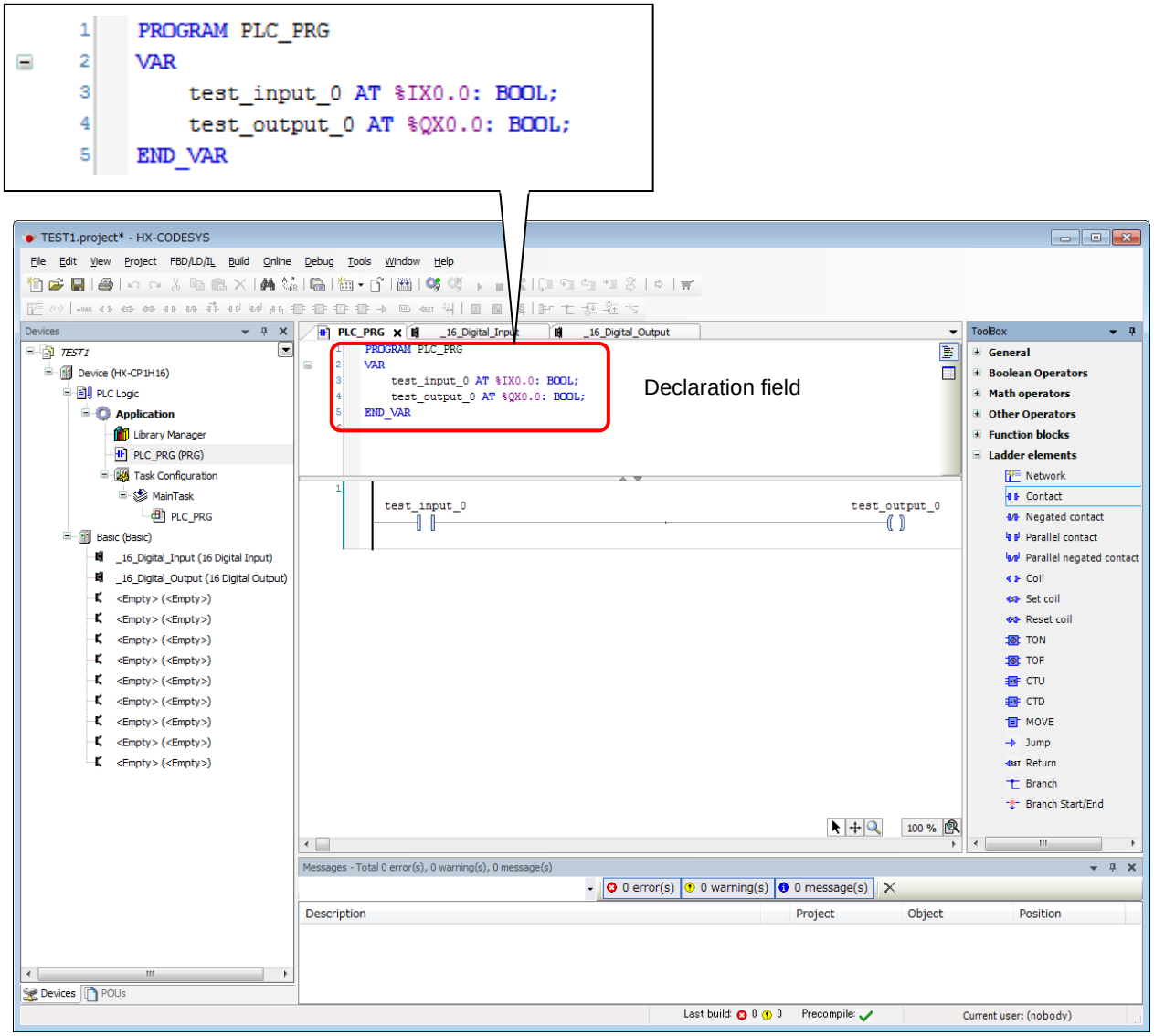
[Local variable]

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], declared information is added automatically as below.



I/O address example of 64 points output module

Bit number	BOOL	BYTE	WORD	DWORD	LWORD	
Bit 0	%QX0.0	%QB0	%QW0	%QD0	%QL0	LSB
Bit 1	%QX0.1					
Bit 2	%QX0.2					
Bit 3	%QX0.3					
Bit 4	%QX0.4					
Bit 5	%QX0.5					
Bit 6	%QX0.6					
Bit 7	%QX0.7					
Bit 8	%QX1.0	%QB1				
Bit 9	%QX1.1					
Bit 10	%QX1.2					
Bit 11	%QX1.3					
Bit 12	%QX1.4					
Bit 13	%QX1.5					
Bit 14	%QX1.6					
Bit 15	%QX1.7					
Bit 16	%QX2.0	%QB2	%QW1			
Bit 17	%QX2.1					
Bit 18	%QX2.2					
Bit 19	%QX2.3					
Bit 20	%QX2.4					
Bit 21	%QX2.5					
Bit 22	%QX2.6					
Bit 23	%QX2.7					
Bit 24	%QX3.0	%QB3				
Bit 25	%QX3.1					
Bit 26	%QX3.2					
Bit 27	%QX3.3					
Bit 28	%QX3.4					
Bit 29	%QX3.5					
Bit 30	%QX3.6					
Bit 31	%QX3.7					
Bit 32	%QX4.0	%QB4	%QW2	%QD1		
⎮	⎮					
Bit 39	%QX4.7					
Bit 40	%QX5.0	%QB5				
⎮	⎮					
Bit 47	%QX5.7					
Bit 48	%QX6.0	%QB6	%QW3			
⎮	⎮					
Bit 55	%QX6.7					
Bit 56	%QX7.0	%QB7				
⎮	⎮					MSB
Bit 63	%QX7.7					

Following 5 different codes access the same bit.

```
%QX0.0:=1;
%QB0  :=1;
%QW0  :=1;
%QD0  :=1;
%QL0  :=1;
```

Note

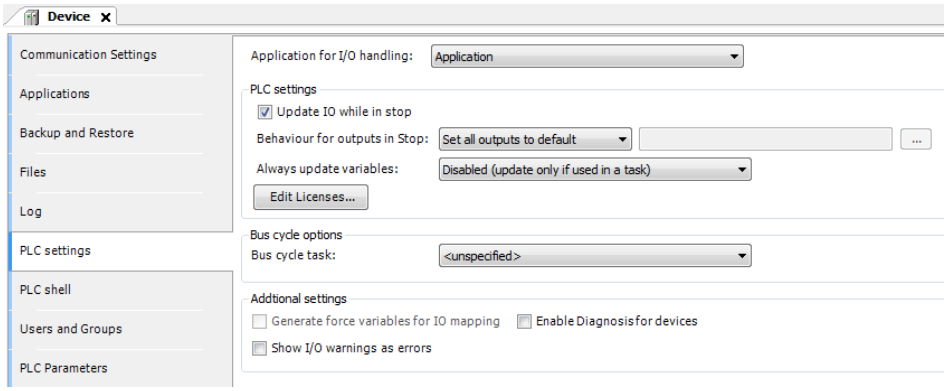
If application program of EHV+ series is reused in HX-series and direct IEC addresses are used, although it is unusual, be noted that I/O addresses are byte-swapped. If variables are defined in I/O mapping tables, then it is no problem.

The deference of I/O address of 64 points output module between EHV+ series and HX series.

	EHV+ series IEC address					HX series IEC address				
Bit	BOOL	BYTE	WORD	DWORD	LWORD	BOOL	BYTE	WORD	DWORD	LWORD
Bit 0	%QX7.0	%QB7	%QW3	%QD1	%QL0	%QX0.0	%QB0	%QW0	%QD0	%QL0
Bit 1	%QX7.1					%QX0.1				
Bit 2	%QX7.2					%QX0.2				
Bit 3	%QX7.3					%QX0.3				
Bit 4	%QX7.4					%QX0.4				
Bit 5	%QX7.5					%QX0.5				
Bit 6	%QX7.6					%QX0.6				
Bit 7	%QX7.7					%QX0.7				
Bit 8	%QX6.0	%QB6				%QX1.0	%QB1			
Bit 9	%QX6.1					%QX1.1				
Bit 10	%QX6.2					%QX1.2				
Bit 11	%QX6.3					%QX1.3				
Bit 12	%QX6.4					%QX1.4				
Bit 13	%QX6.5					%QX1.5				
Bit 14	%QX6.6					%QX1.6				
Bit 15	%QX6.7					%QX1.7				
Bit 16	%QX5.0	%QB5	%QW2			%QX2.0	%QB2	%QW1		
Bit 17	%QX5.1					%QX2.1				
Bit 18	%QX5.2					%QX2.2				
Bit 19	%QX5.3					%QX2.3				
Bit 20	%QX5.4					%QX2.4				
Bit 21	%QX5.5					%QX2.5				
Bit 22	%QX5.6					%QX2.6				
Bit 23	%QX5.7					%QX2.7				
Bit 24	%QX4.0	%QB4				%QX3.0	%QB3			
Bit 25	%QX4.1					%QX3.1				
Bit 26	%QX4.2					%QX3.2				
Bit 27	%QX4.3					%QX3.3				
Bit 28	%QX4.4					%QX3.4				
Bit 29	%QX4.5					%QX3.5				
Bit 30	%QX4.6					%QX3.6				
Bit 31	%QX4.7					%QX3.7				
Bit 32	%QX3.0	%QB3	%QW1	%QD0		%QX4.0	%QB4	%QW2	%QD1	
⋮	⋮					⋮				
Bit 39	%QX3.7					%QX4.7				
Bit 40	%QX2.0	%QB2				%QX5.0	%QB5			
⋮	⋮					⋮				
Bit 47	%QX2.7					%QX5.7				
Bit 48	%QX1.0	%QB1	%QW0			%QX6.0	%QB6	%QW3		
⋮	⋮					⋮				
Bit 55	%QX1.7					%QX6.7				
Bit 56	%QX0.0	%QB0				%QX7.0	%QB7			
⋮	⋮					⋮				
Bit 63	%QX0.7					%QX7.7				

2.2 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, unused I/Os are not 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]-[PLC Parameters] 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.

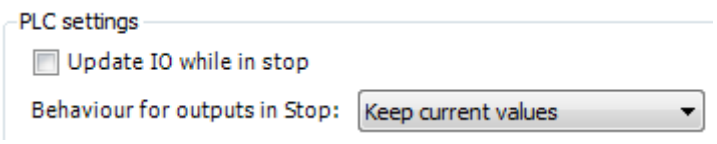
Always update variables:

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.

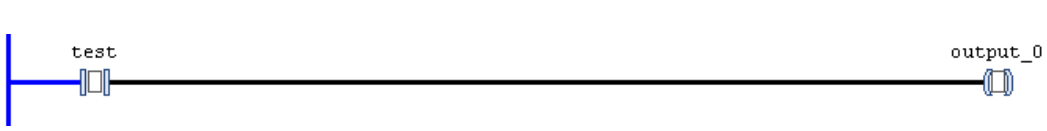
Note

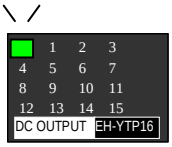
If all the following conditions are fulfilled and reset warm/cold is operated, the last status of output module remains although monitored output status of HX-CODESYS is reset.

- Update IO while in stop in PLC settings: Disabled
- Behavior for outputs in Stop in PLC settings: Keep current values
- Reset all outputs in STOP in Device Configuration: No
- Variable of output module is mapped to existing variable that declared in POU or GVL.



Actual output remains after Reset warm/cold





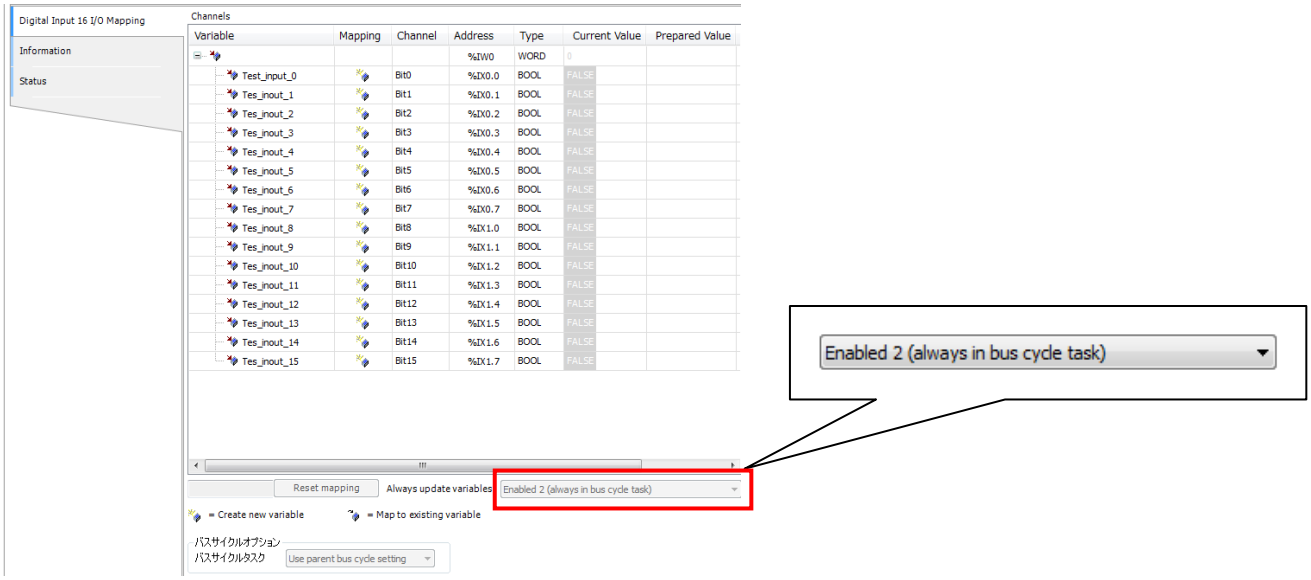
This is expected behavior. If this setting combination is required, keep in mind this mismatching and be careful to use.

Reset all outputs in STOP

This setting is in [Device]-[PLC Parameters]. If [Reset all outputs in STOP] is [Yes] (default), all the PLC outputs including counter outputs and pulse train output of positioning module are reset because it is reset by a certain hardware signal running on the back plane bus. If default value in configured as TRUE in I/O mapping table, it is momentary reset (FALSE) at run or stop timing. If default values should be kept, set [Reset all outputs in STOP] as [No]. In this case, you must be aware following limitation.

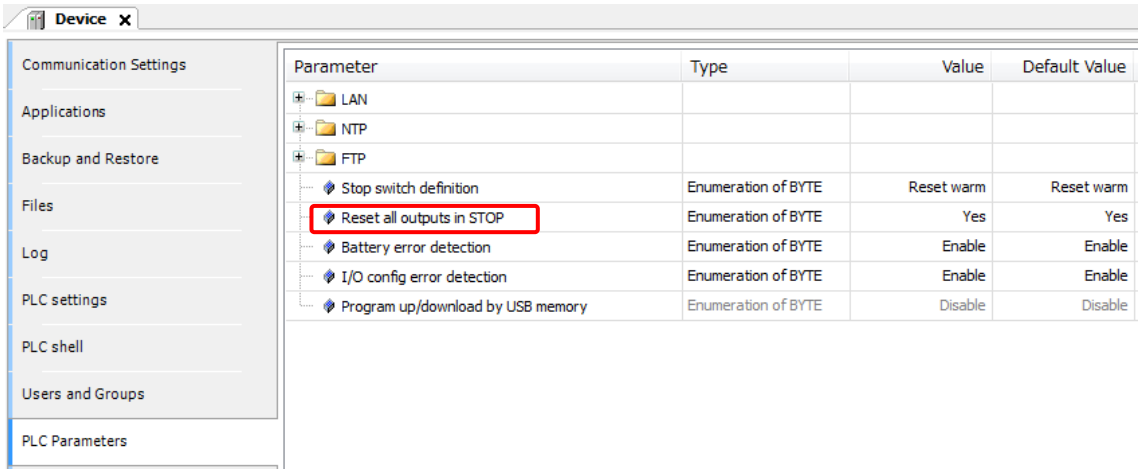
Note

I/O that is not used in program does not refresh. Therefore, you are not seen in TRUE when you monitored a mapping table in online even if the outside input that you don't use set to ON.
When you want to monitor unused I/O, please set [Enabled1] or [Enabled2] to [Always updates variables] of the mapping table of the lower right. I/O is refreshed regardless of use / unused.



Note

If [Reset all outputs in STOP] is [No], PLC outputs without IEC address, such as counter outputs or pulse train outputs, are NOT reset when CPU stops. We recommend you to set [Yes] when using counter or positioning modules.

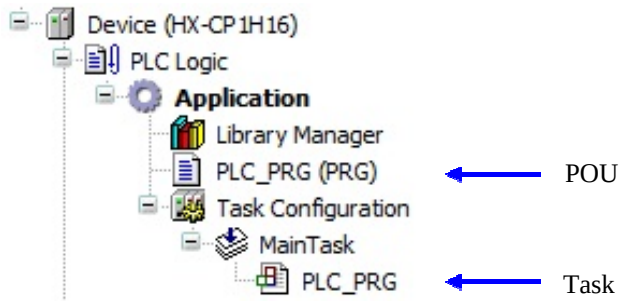


Note

If [Reset all outputs in STOP] is [Yes] (default), default value of [TRUE] in I/O mapping table is momentary reset (FALSE) at run or stop timing.

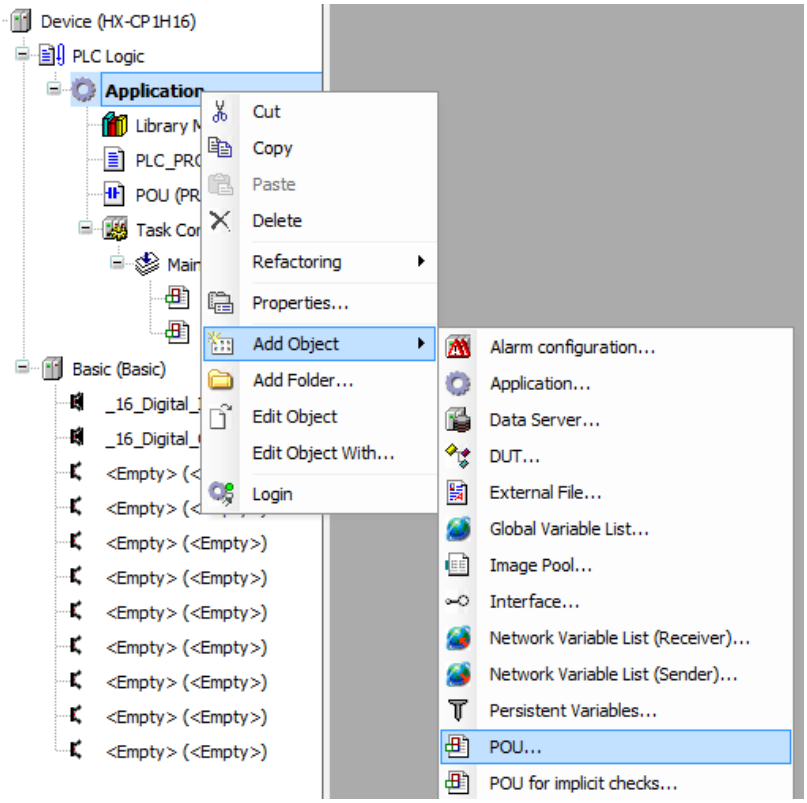
2.3 POU and Task

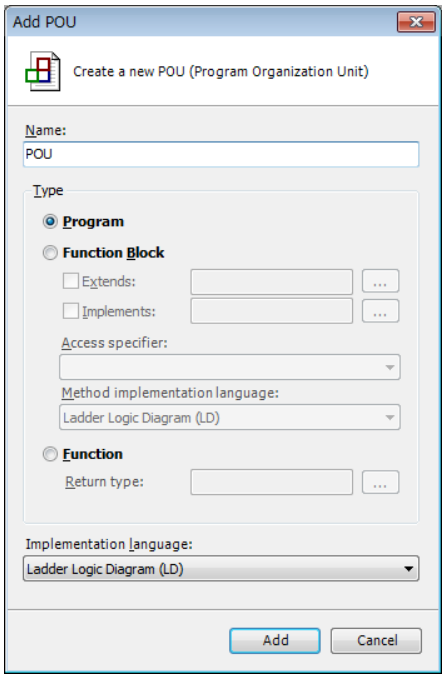
One application has at least one POU and one task as shown below.



POU

POU stands for Program Organization Unit and smallest unit of 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.





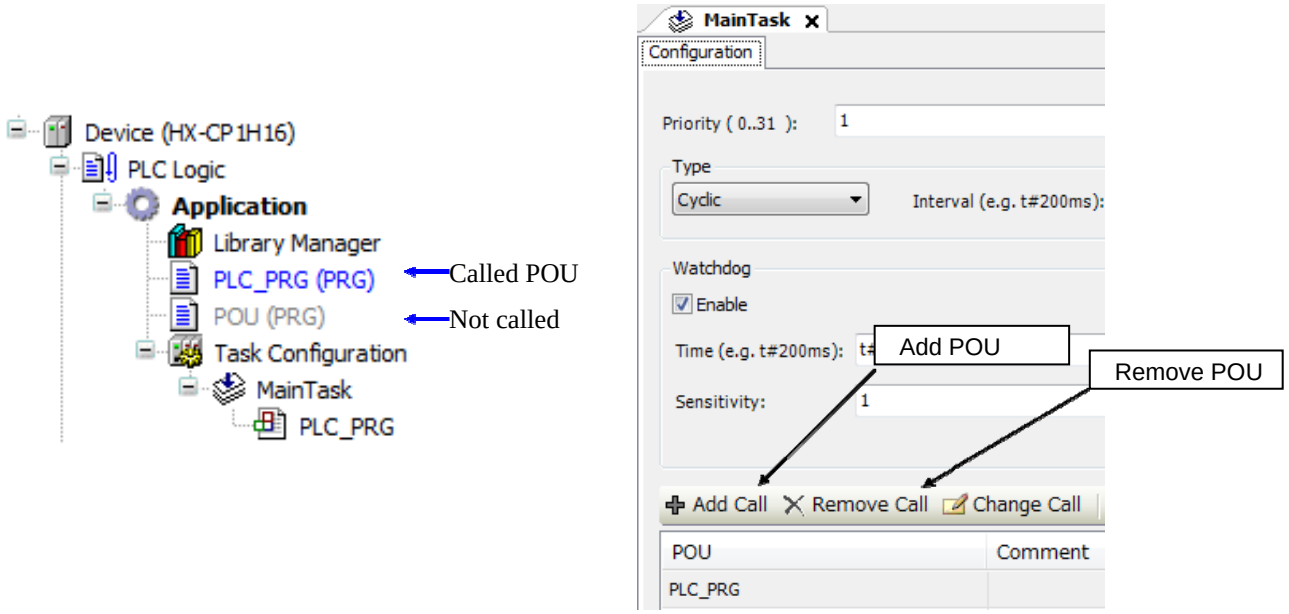
There are three kinds in POU, and each POU are different in how to use.

	Called by task	How to call	I/O limitation	Store data
Program (PRG)	Yes	Direct name	None	Yes
Function Block (FB)	No	Instance (copy)	None	Yes
Function (FUN)	No	Direct name	1 output only	No

Task

POU does not have information how it is executed. This information is handled by task.

Put priority, choose type of task and add or remove POU accordingly. Instead of clicking [Add Call], POU can be dragged into a task. After a project compiled, the color of POU will be changed. If POU is called by a task, it will be blue. If the POU is not called, the color will be gray.



Priority (0..31)

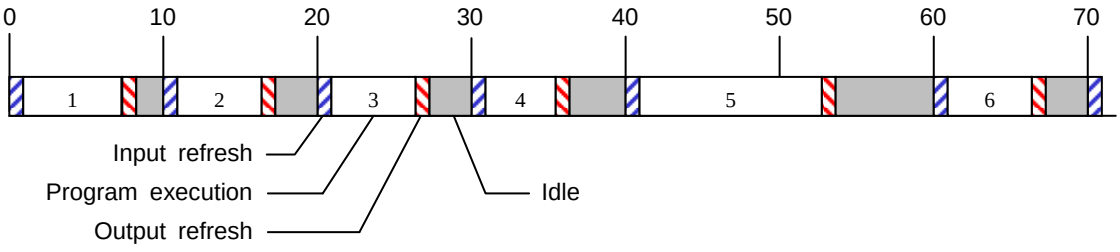
0 is the highest priority, 31 is the lowest.

Type

Choose type of the task. Inputs are read at the beginning of each task, and outputs are written at the end of the program execution.

Cyclic task

The task will be processed cyclic according to the time definition given in the field [Interval]. If actual execution time exceeds the cycle time, the next cycle does not start immediately but wait until the next fixed cycle time. For example, cycle time is set as 10 ms and 5th scan starting from 40 ms takes 12 ms, then 6th cycle starts at 60 ms as below figure.



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 3 ms of sleeping time is added at the end of each cycle for other tasks to be executed properly.

Status task

The task will be started when selected variable is TRUE.

Watchdog

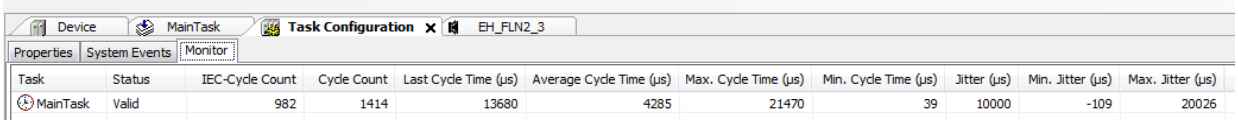
When it is enabled, watchdog function is activated. If program execution time exceeds watchdog time, CPU stops program execution with [24] error code displayed at 7 segment LED.

There are two different conditions to detect watchdog error as follows.

Example: Time: #5 ms, Sensitivity: 3.

- Detect condition 1 : one cycle exceeds 15 ms
- Detect condition 2 : scan cycle exceeds 5 ms in 3 times consecutive

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



Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (µs)	Average Cycle Time (µs)	Max. Cycle Time (µs)	Min. Cycle Time (µs)	Jitter (µs)	Min. Jitter (µs)	Max. Jitter (µs)
MainTask	Valid	982	1414	13680	4285	21470	39	10000	-109	20026

2.4 Available Characters for Variable Names

Variable name / POU name

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

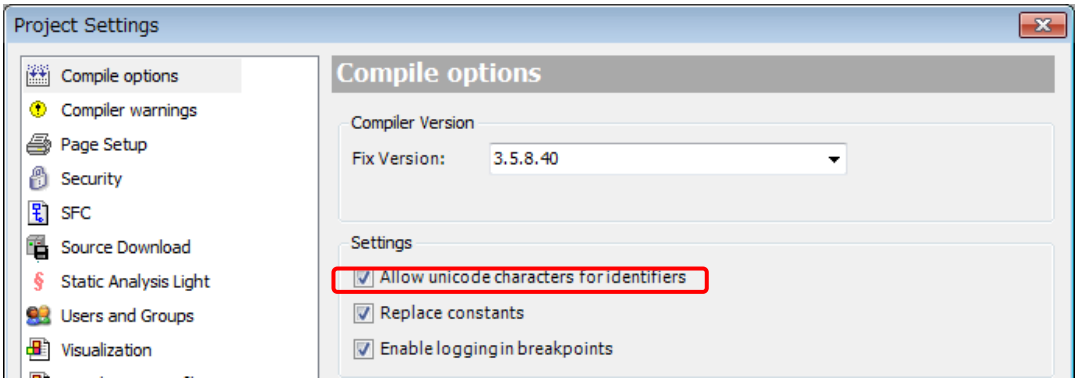
Supported characters

Types	Supported	Remarks
Numerical	0 to 9	Not allowed to begin with numeric characters.
Alphabetical	a to z, A to Z	There is no difference of the small and capital letter.
Symbol	_	Trailing underscores are not allowed.

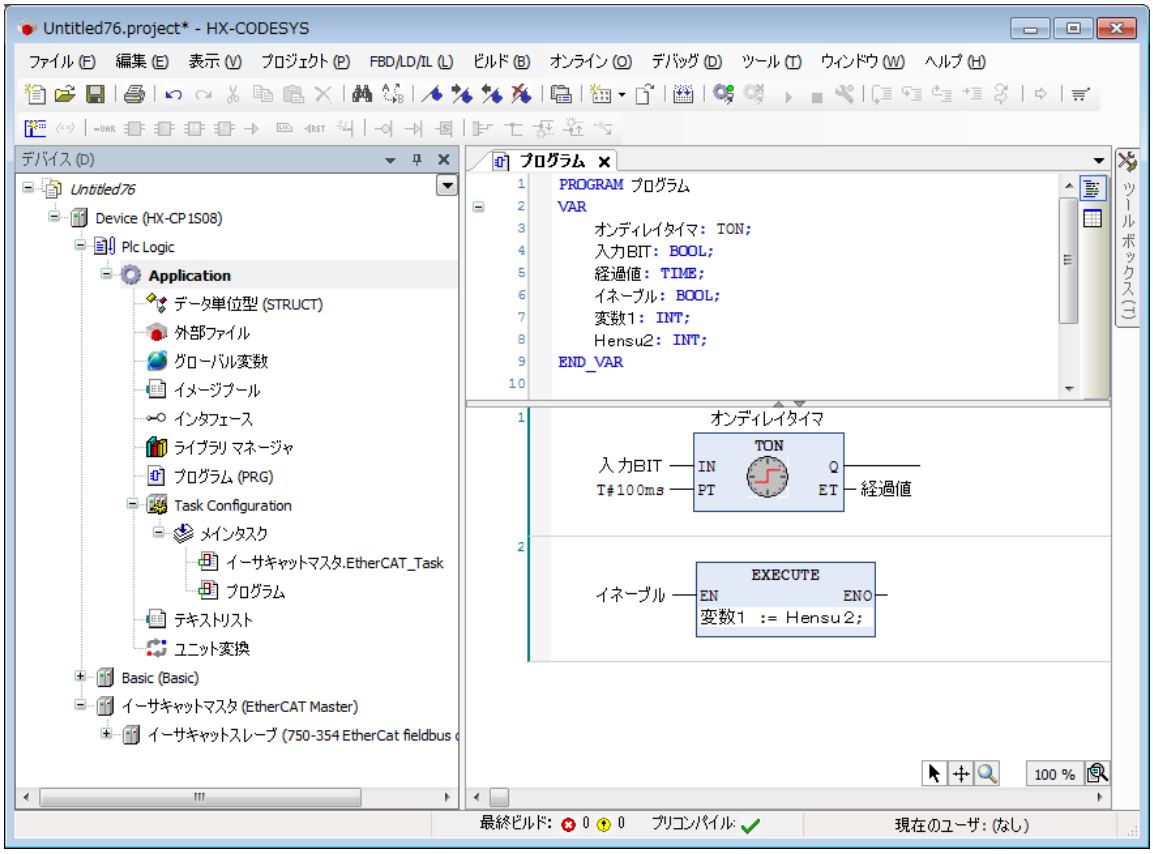
Examples for variable names

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.

If you use unicode characters, click [Project] - [Project setting], and check [Allow unicode characters for identifiers] in [Compile options].

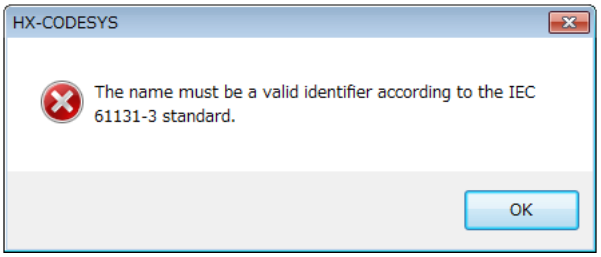


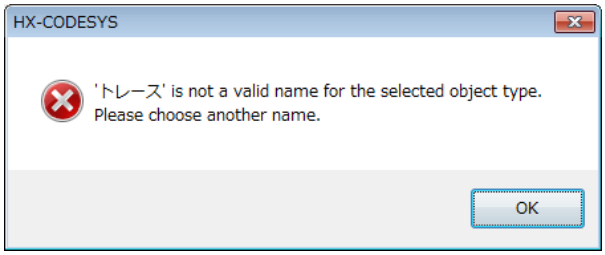
Example: Japanese



Note

If unicode characters are used improperly, warning message is displayed as follows.



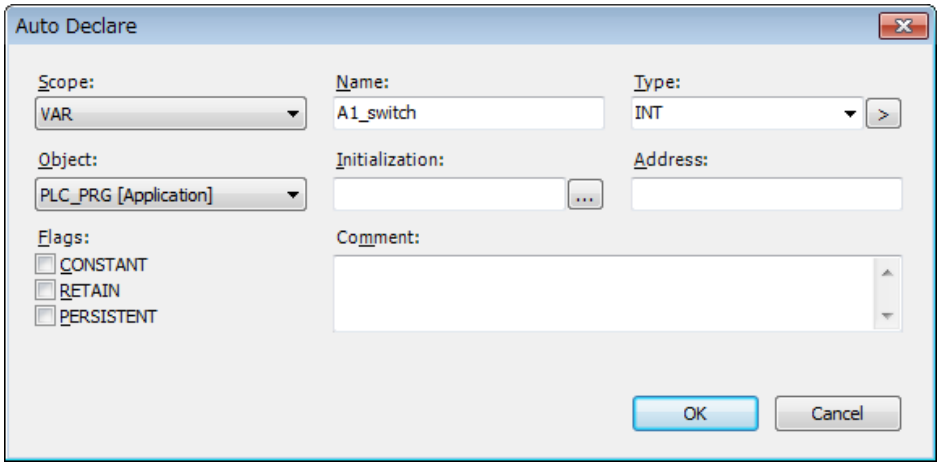


Variables using multibyte characters (Japanese, Chinese, umlaut in German, etc.) cannot be used for HMI or OPC communication.

2.5 Variables

2.5.1 Data Memory

In HX-CODESYS programming, external I/O 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 2.5.8 for further information.
Name		Variable name is defined. (available characters are described in section 2.4.)
Type		Data type is defined. Refer to section 2.5.5.
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 retained in nonvolatile RAM. It is not initialized by reset warm. But it is initialized by reset cold or program downloading. (Refer to the section 2.12)
	PERSISTENT	The value is retained in nonvolatile RAM. It is not initialized by reset warm, reset cold, and program downloading. (Refer to the section 2.12)

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;
```

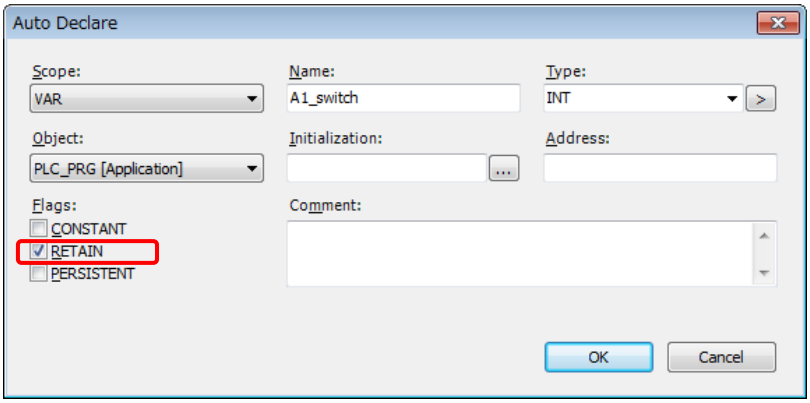
Login display

```
wTest 16#0005 :=5;
wTest 16#0005 .0 TRUE;
wTest 16#0005 .1 FALSE;
wTest 16#0005 .2 TRUE;
wTest 16#0005 .3 FALSE;
```

2.5.2 Retentive Data Memory

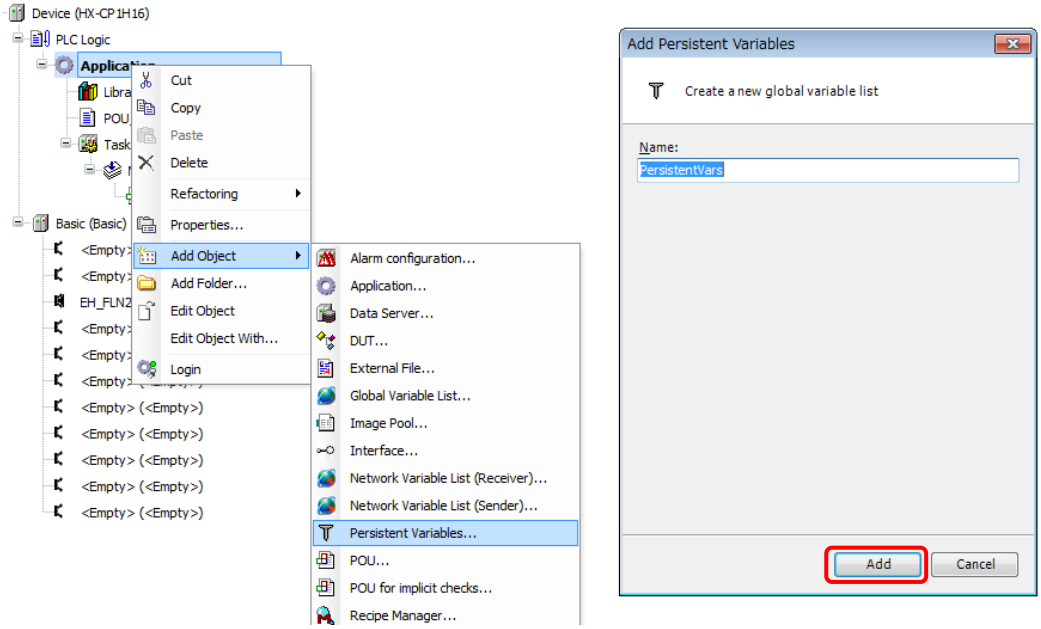
[RETAIN]

Activate [RETAIN] to set retain variable. The value is retained while power failure.

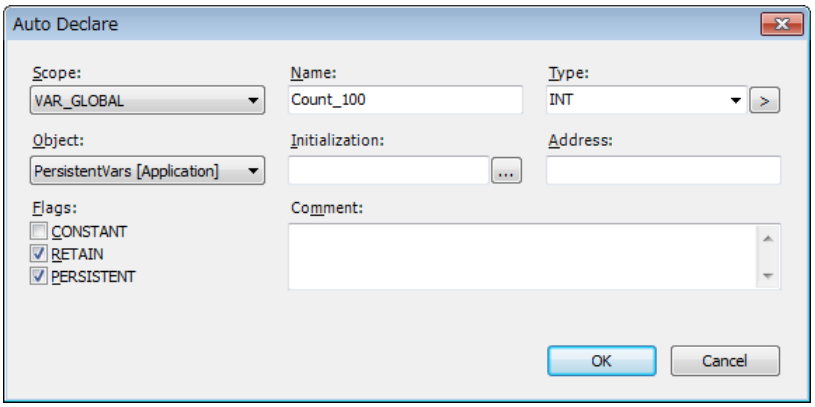


[PERSISTENT]

Add [Persistent Variables] by right mouse click menu on [Application]-[Add Object]-[Persistent Variables].



Activate [RETAIN] and [PERSISTENT] both and choose [VAR_GLOBAL] at [Scope] and [PersistentVars] at [Object] to set persistent variable. The value is retained while power failure. Unlike [RETAIN], it is not initialaized even when reset cold or program downloading.

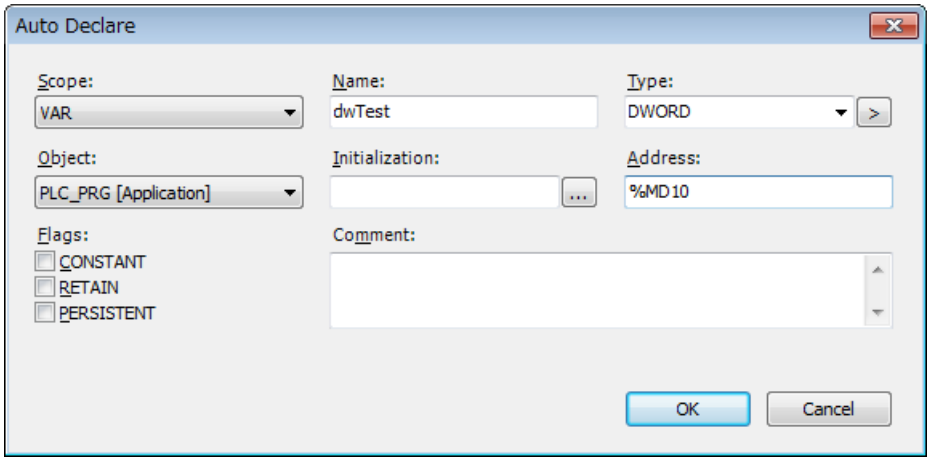


When it is used in POU, add prefix "`PersistentVars.`" to variable names.

```
PersistentVars.Count_100;
```

2.5.3 Marker Memory

Normally users do not have to take care about internal address of data memory.
However if it is necessary to use I/O address consciously, such as accessing only the upper word of DWORD data, use marker memory. 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 2-9. Just replace “Q” with “M”.
The marker memory does not support RETAIN nor PERSISTENT flags.

Variable declaration

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

END_VAR
```

Login display

Expression	Type	Value	Prepared value	Address
dwTest	DWORD	16#12345678		%MD10
wTest_L	WORD	16#5678		%MW20
wTest_H	WORD	16#1234		%MW21

The max. size of marker memory is 48KB. Supported address range is shown below.

Data type	Address range
BOOL	%MX0.0 to %MX49151.7
BYTE	%MB0 to %MB49151
WORD	%MW0 to %MW24575
DWORD	%MD0 to %MD12287
LWORD	%ML0 to %ML6143

Note

If CPU Link module or FL-net module are used, the marker memory is used for their shared memory. Be sure use free address if the marker memory is used as general register.

2.5.4 Numeric Literals

The constant inputs in the following formats.

Types	Examples	Remarks
Binary	`FALSE, TRUE, 0, 1, 2#1010_1111`	Underscore is ignored
Decimal	`-12 0 123_456 +986 10#1234`	Underscore is ignored
Hexadecimal	`16#1234, 16#FF00_F000`	Underscore is ignored
Real	`-12.0 0.0 0.4560 3.14159_26`	Underscore is ignored
Time	`T#100ms, T#5.5s`	Timer (TON, etc.)
Date	`DT#2012-12-31-12:34:56`	RTC (Realtime clock)

2.5.5 Elementary Data Types

HX-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} to $2^{63}-1$
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`
22	`WSTRING`	Variable-length double-byte character string	16× n	-

2.5.6 User Defined Data Types

HX-CODESYS supports below data types.

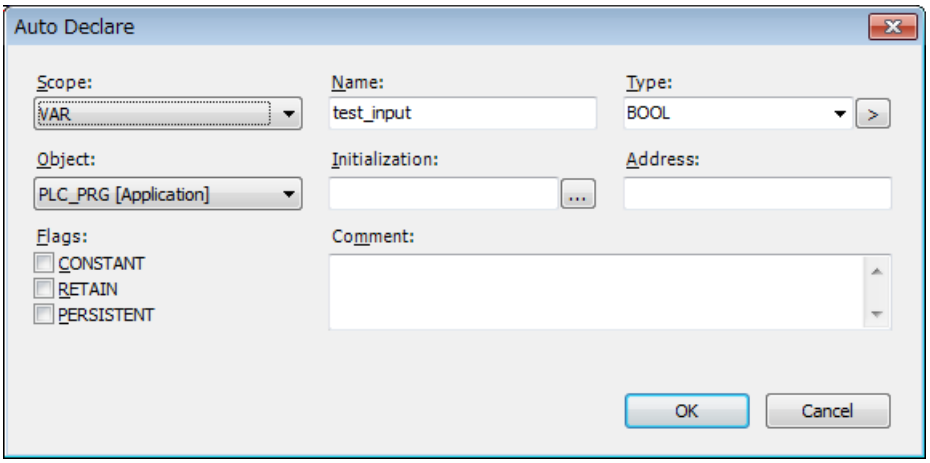
No.	Name	Examples
1	Array	`ARRAY [0..100] OF WORD;` `ARRAY [0..11, 0..22, 0..33] OF REAL;`
2	Subrange	`WORD(0..4095)`

3	Enumeration	`TYPE COLOR:`
		`(Red, Yellow, Green)`
		`:=Green;`
		`END_TYPE`

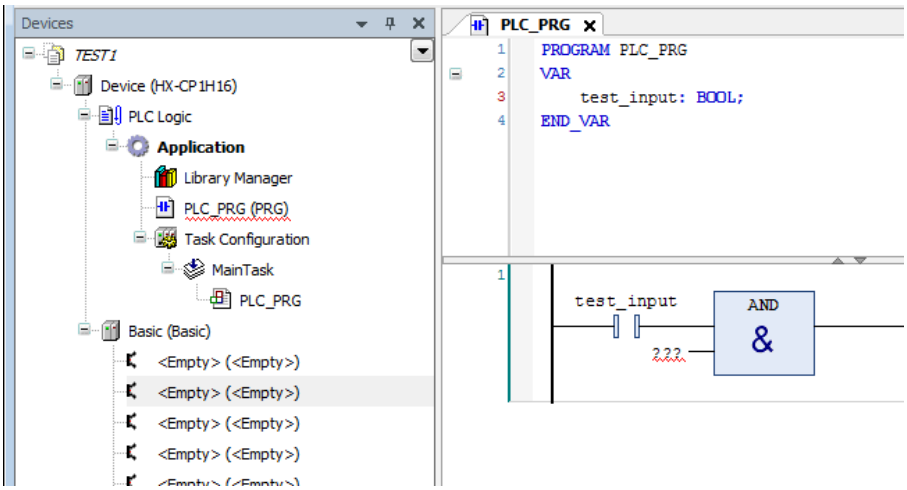
4	Structure	`TYPE STRUCT_sample`
		`STRUCT`
		`ID : WORD;`
		`Flag : BOOL;`
		`Weight : REAL;`
		`END_STRUCT`
		`END_TYPE`

2.5.7 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], this variable is registered in declaration part of POU as below.

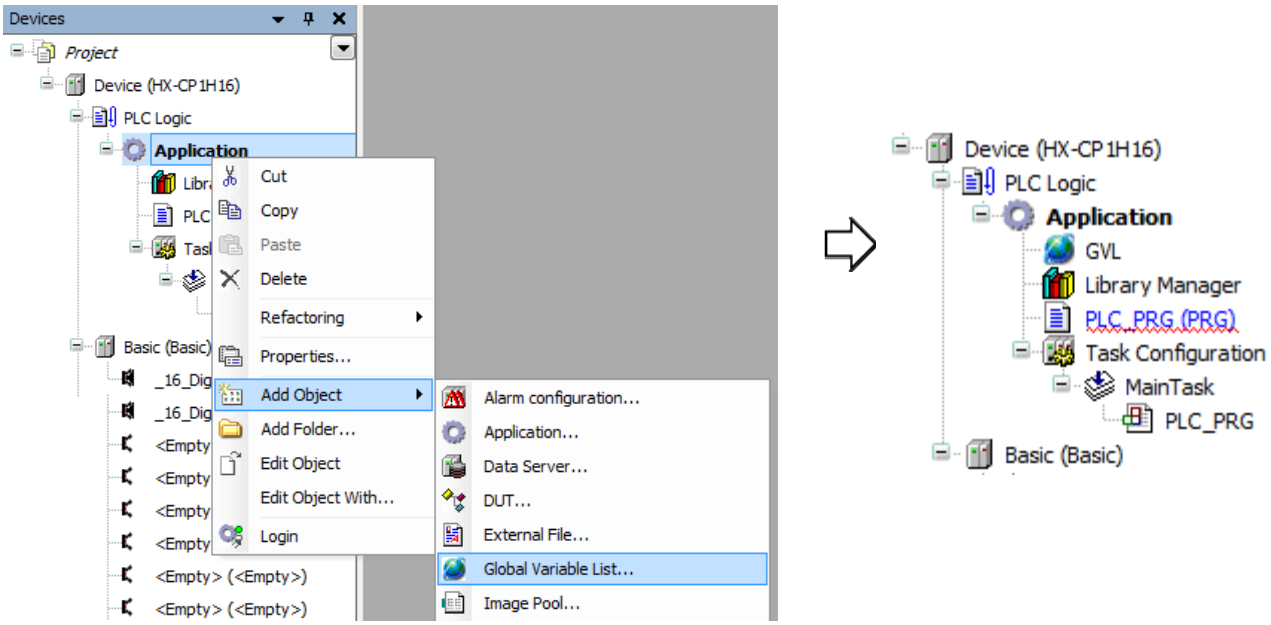


This variable is valid only in the POU. Even if same variable name is used in another POU, [Auto Declare] window will appear and it will be assigned in another memory location and handled as different variable.

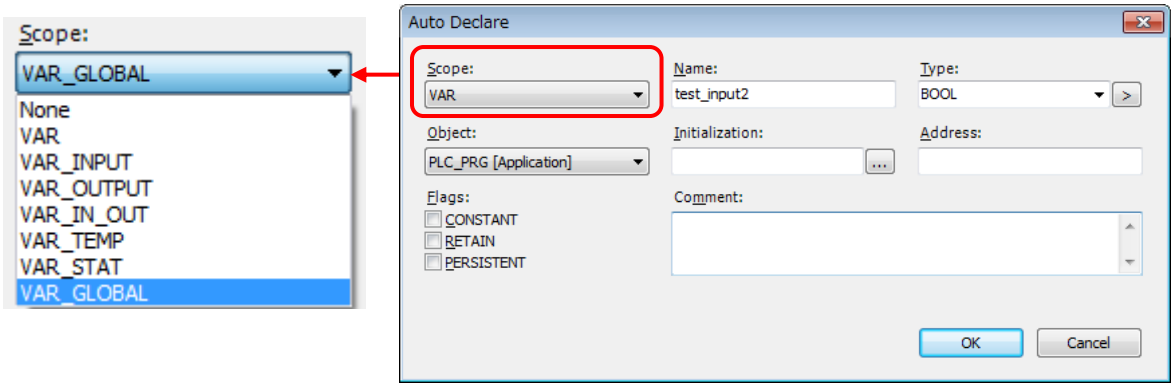
To display the [Auto Declare] window again after defining the variable, place the cursor in the variable and then select [Edit] - [Auto Declare...].

2.5.8 Global Variable

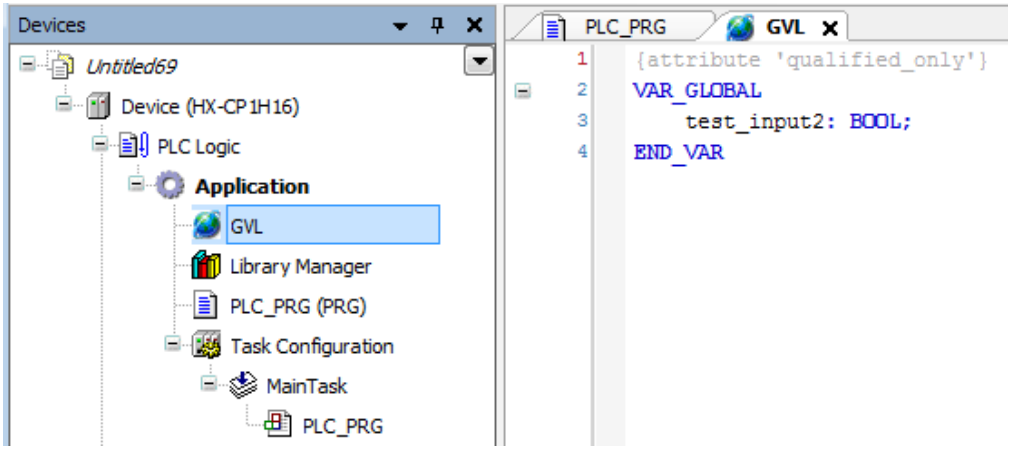
If variables need to be commonly used in all POUs, [Global Variable List] must be created by right click on Application as below.



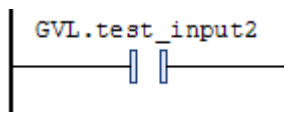
If new variable name is used in POU, [Auto Declare] window appears as shown in local variables. Choose [VAR_GLOBAL] at [Scope] as below.



New variable name [test_input2] is registered in GVL as below instead of POU.



Add a prefix "GVL." to global variable when using in POU.



2.6 Configuration

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

Parameter		Description	
LAN/ ETH1, ETH2, ETH3	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		
	Ethernet port Link speed / Duplex mode		
Default Gateway			
Change IP information		Yes	IP information is downloaded together with application.
		No (default)	IP information is not downloaded when application downloading.
NTP	NTP function	Enable / Disable (default)	Setting Use(Enable) calendar clock from NTP server or not(Disable)
	Port number	ETH1 / ETH2 / ETH3	Setting port used calendar clock
	Logical port number	123	The number is fixed.
	Specified by	IP address	How to specify NTP server (fixed)
	IP address or Host name	Setting IP address or host name	
	Access cycle	Setting access cycle to NTP server (Unit: minute default: 60 min.)	
	Timeout	Timeout is 10 s. (fixed)	
	TimeZone	Setting time zone	
FTP	FTP server	Setting parameters regarding FTP server Refer to chapter 4.2	
	Port number		
	Access Media		
	User Name		
	Password		
Stop switch definition		Reset warm (default)	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.
Reset all outputs in STOP		Yes (default)	all outputs are reset by hardware signal on the backplane when switching to stop mode. (Refer to chapter 2.2)
		No	all outputs are controlled by IEC program (software)
Battery error detection		Enable (default)	Detect battery error
		Disable	Not detect battery error
I/O config error detection		Enable (default)	Detect I/O configuration error
		Disable	Not I/O configuration error
Program up/download by USB memory		Enable / Disable (default)	Program up/download function is enabled/disabled.

Note

- Do not set network address (host parts 0) nor broadcast address (host parts 255). It is possible to set, but CPU will detect an error and message will be stored in the log.
- Do not set illegal subnet mask such as 255.255.253.0. It is possible to set, but CPU will detect an error and message will be stored in the log.
- When using default gateway, be sure to use correct IP addresses not to duplicate network address of ETH1, 2, 3 with network address of the gateway.

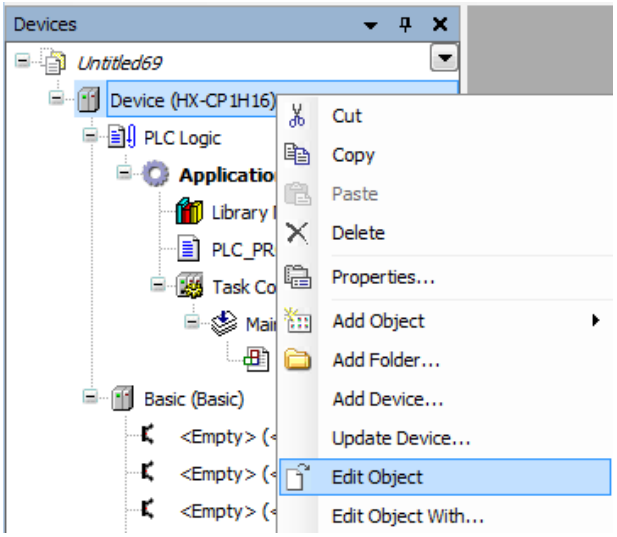
2.7 Communication Settings

How to configure

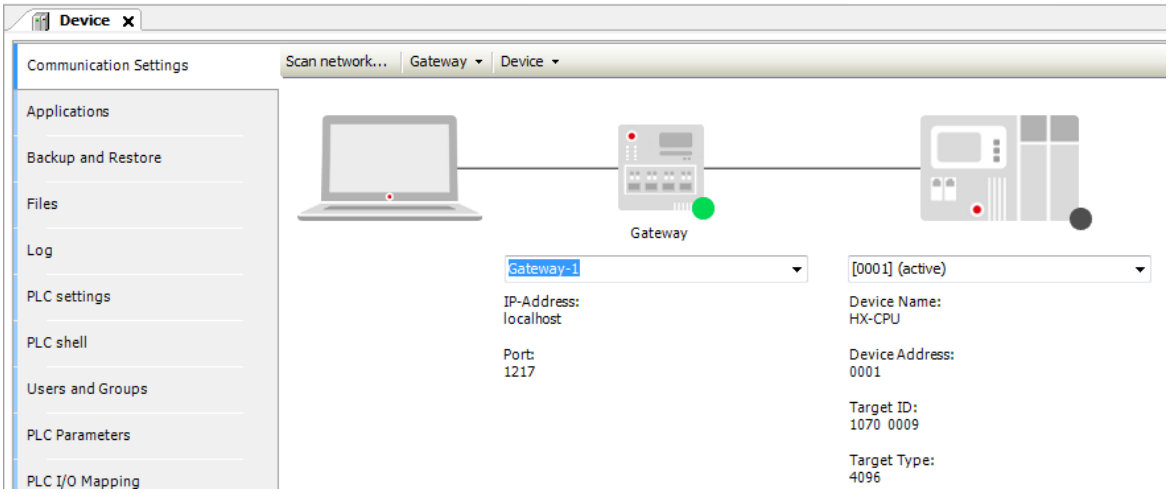
Connect PC to HX-CPU with USB cable (Mini-B) or LAN cable. Either crossover cable or straight cable is available for Ethernet ports of HX-CPU. Be sure to set IP addresses of PC and HX-CPU in the same segment. If not, login with USB and change IP address of HX-CPU accordingly. The default IP addresses of HX-CPU are shown below.

Port	IP address	Subnet mask
ETH1	192.168.0.1	255.255.255.0
ETH2	192.168.1.1	255.255.255.0
ETH3	192.168.2.1	255.255.255.0

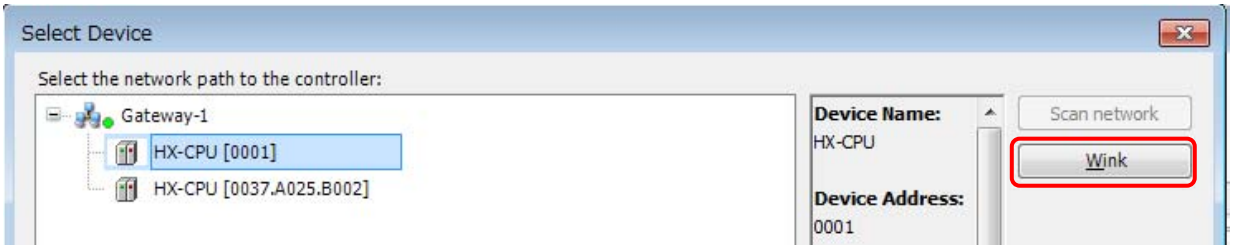
Double click on [Device (HX-CPxxxx)] or right click and choose [Edit Object].



[Device] window will appear as below.



Choose [Communication Settings] tab and click [Scan network]. If multiple CPUs exist in the network, all the CPUs are displayed as below.

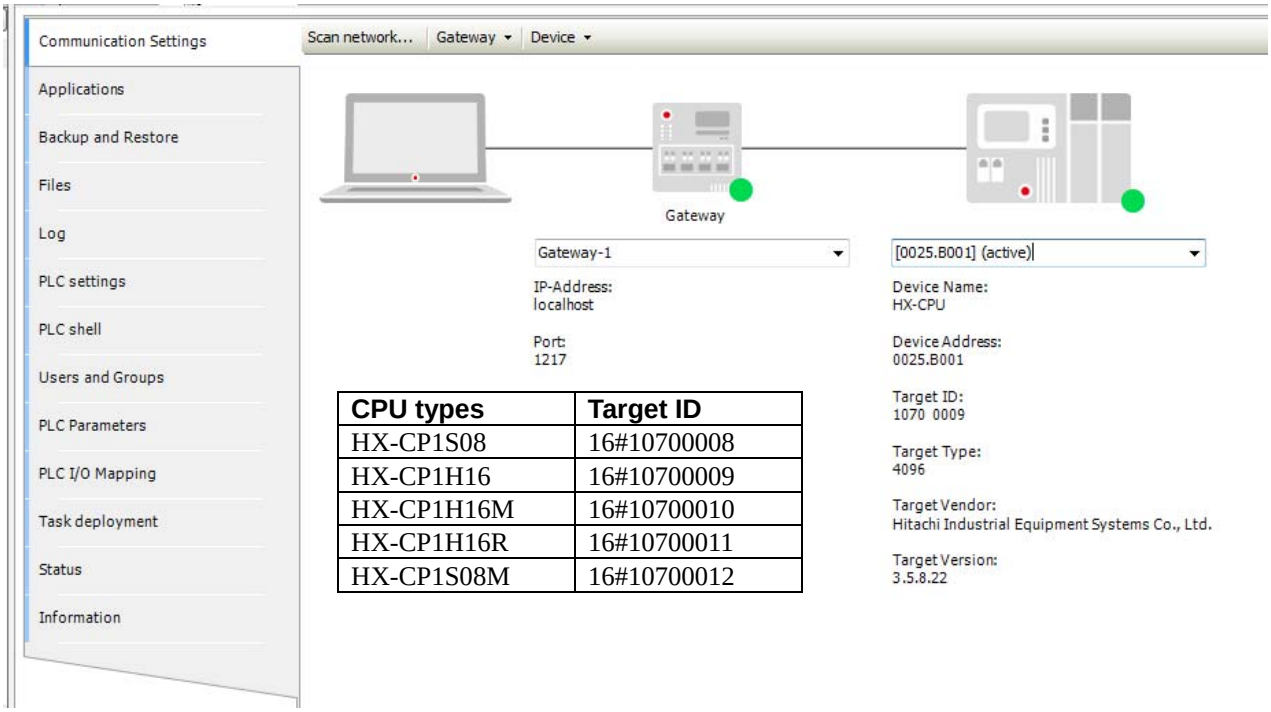


If you choose one of the CPU and click [Wink], RUN LED on the CPU will blink 2 times in case of STOP status or 3 times in case of RUN status.

Note

If you choose [Open Project from PLC…] in the Start Page and click [Wink], CPU does not react.

Click [OK], then the CPU is configured as target device.



CPU types	Target ID
HX-CP1S08	16#10700008
HX-CP1H16	16#10700009
HX-CP1H16M	16#10700010
HX-CP1H16R	16#10700011
HX-CP1S08M	16#10700012

If CPU is not displayed correctly, please check following points.

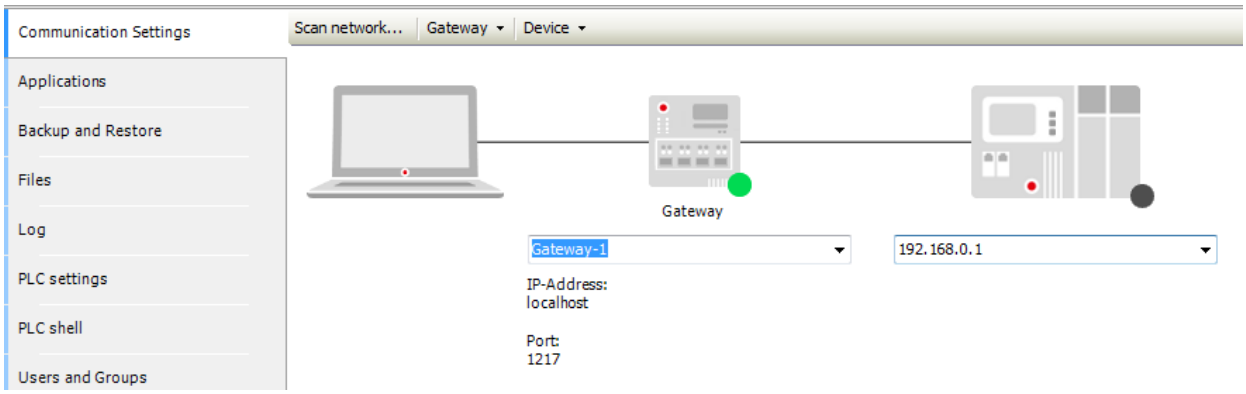
Possible cause	Action
Bad connection of LAN cable	Check the connection of cable and the power of network switches. Check if the LED of L/A at Ethernet connector is lighting in green.
No USB driver installed	Install USB driver (refer to 1.1.3).
Wrong IP address	Check IP address and subnet mask of PC and CPU both. If IP address is unknown, login with USB and set IP address accordingly.
Wrong CPU type	Set right CPU type by right mouse clicking on device and choosing [Update device]. If [Device]-[Filter network scans by target ID] is disabled, other types of CPUs are displayed.

Note

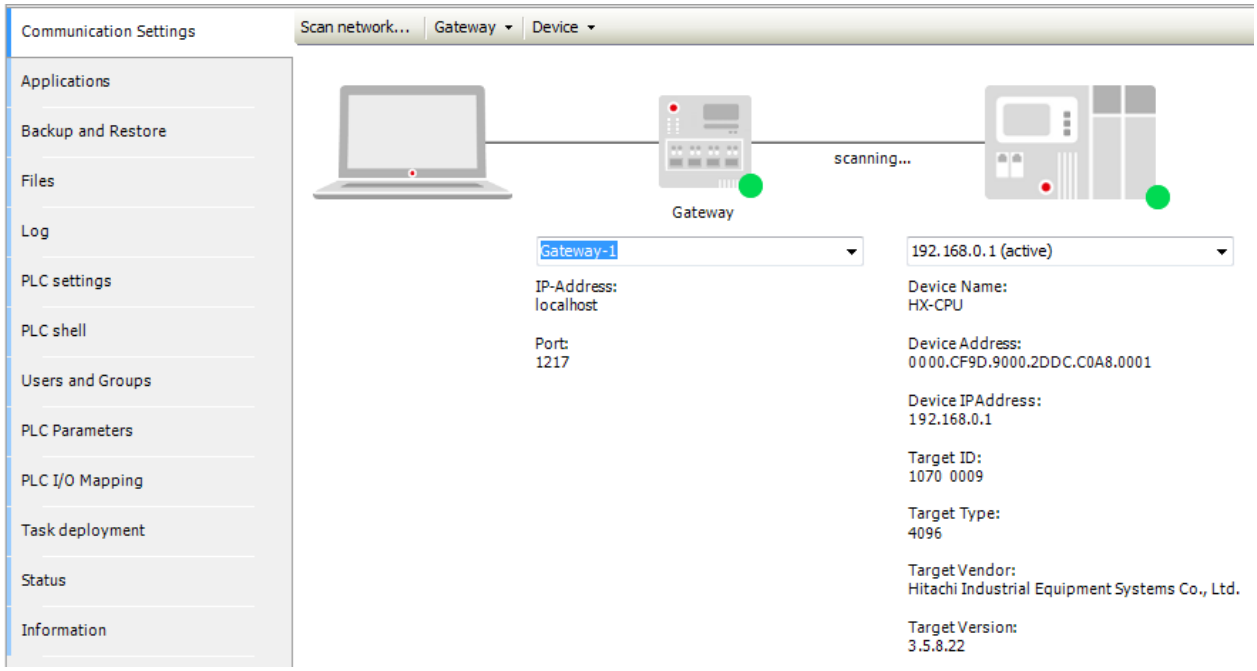
If both USB cable and LAN cable are connected to the same CPU, only one network detected first is displayed.

TCP/IP

The protocol of above mentioned Ethernet communication is UDP/IP. If TCP/IP is required instead of UDP/IP, enter IP address of CPU directly in the address field.

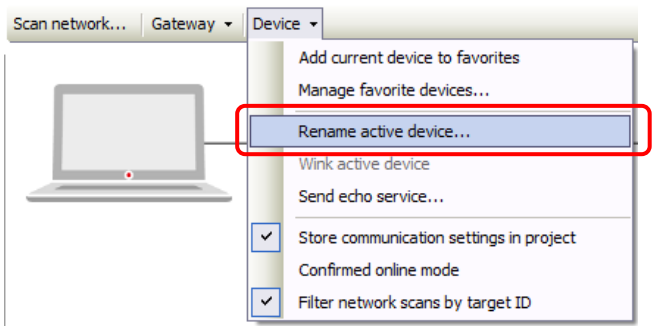


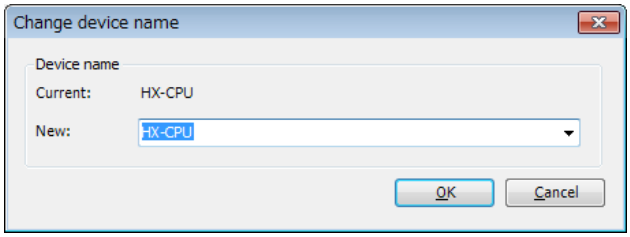
If entered IP is found, information of CPU is displayed.



Rename device

After communication opened, device name of connected CPU can be changed in the menu [Device]-[Rename active device…].



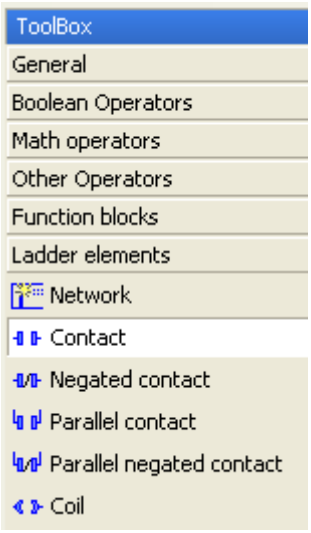


2.8 Programming

Ladder programming

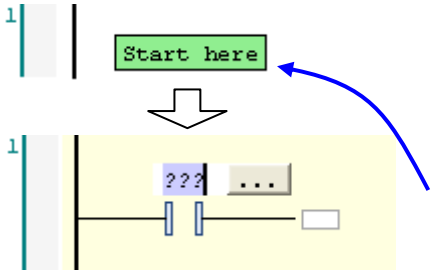
Basic ladder programming is shown below as a first step. Please refer to online-help of HX-CODESYS for further information about programming.

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



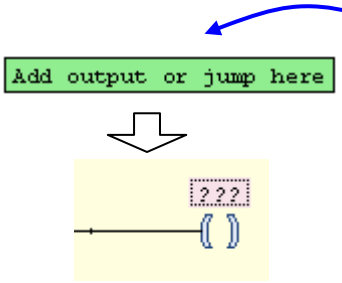
Contact

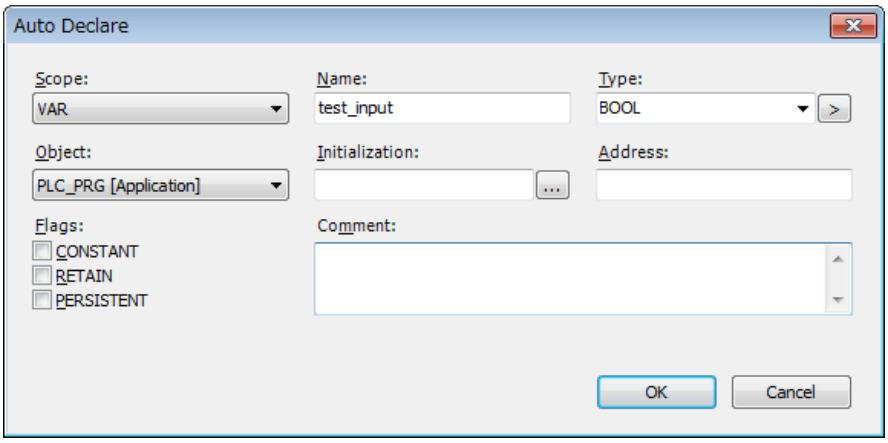
- Drag from ToolBox to [Start here].
- Menu [FBD/LD/IL]-[Insert Contact]
- Right mouse click [Insert Contact]
- Shortcut key [Ctrl + K]



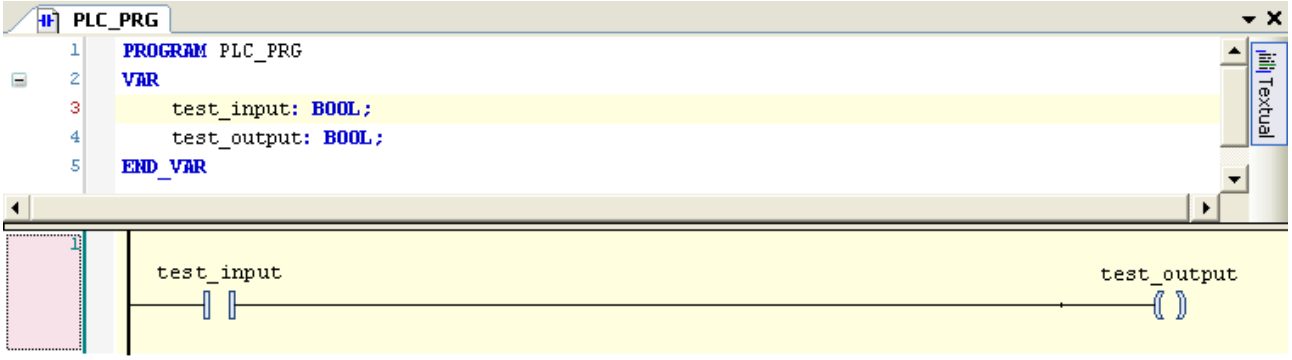
Coil

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



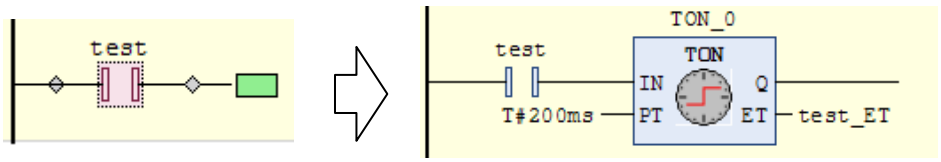


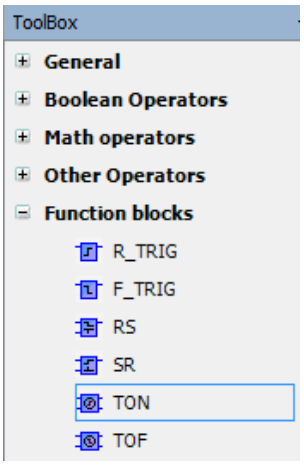
If new variable name is used, [Auto Declare] window appears automatically. Edit each input field and check-boxes if necessary, and Click [OK]. The variable is declared in declaration window as below.



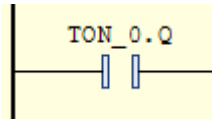
Timer (TON)

Drag TON in Function Blocks to the position to put and drop it.





Specify the name of TON (TON_0 in the figure) and set PT (preset timer value) and the variable for ET (elapsed time). PT is mandatory, but ET is optional. If ET is left as open, delete variable name. New symbol can be added to the output of TON (Q), but this output can be left as open.

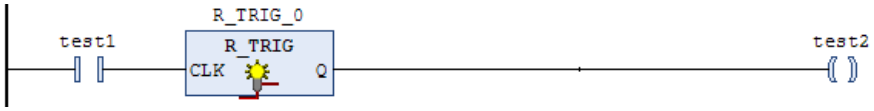


If timer output is used as a contact, put suffix ".Q" to the name of TON. Elapsed time is used as suffix ".ET".

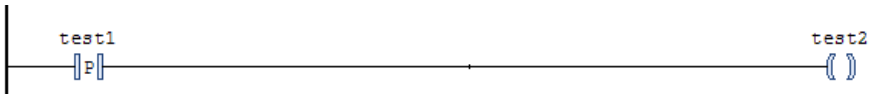
Edge detection

3 different samples with same behavior are shown for rising edge detection.

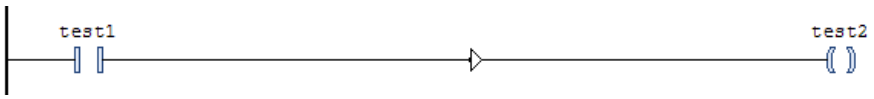
Function block R_TRIG
(F_TRIG for falling edge)



Right mouse click and choose [Edge detection] once.
(—|N|— for falling edge)



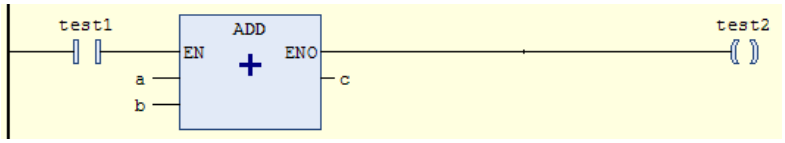
Right mouse click and choose [Edge detection] once.
(—◁— for falling edge)



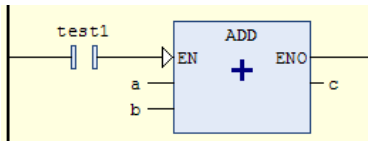
Box with EN/ENO

If EN input is TRUE, the box is executed. By right mouse clicking and choosing [Edge detection] or [Negation], the condition to execute can be changed. The ENO output has the same value as the EN input.

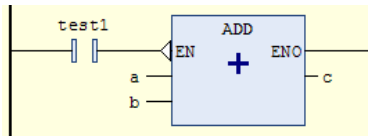
Example 1: The box is executed while [test1] is TRUE.



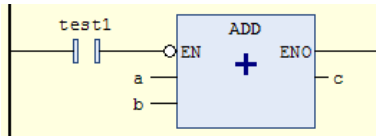
Example 2: Rising edge detection



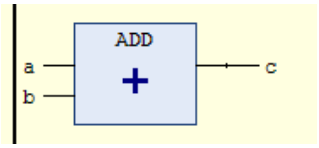
Example 3: Falling edge detection



Example 4: Negation

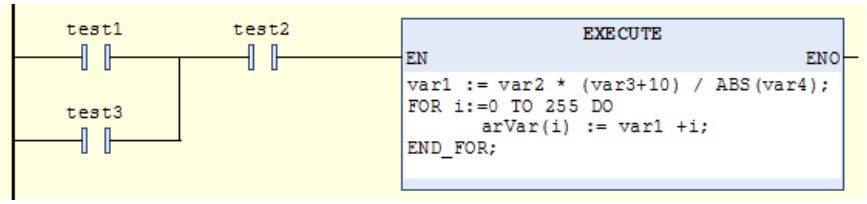


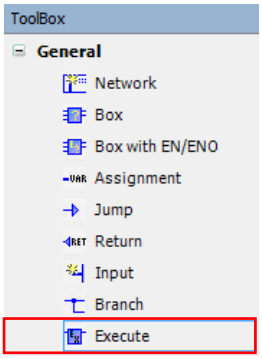
Example 5: Always executed (box without EN/ENO)



Execute Box

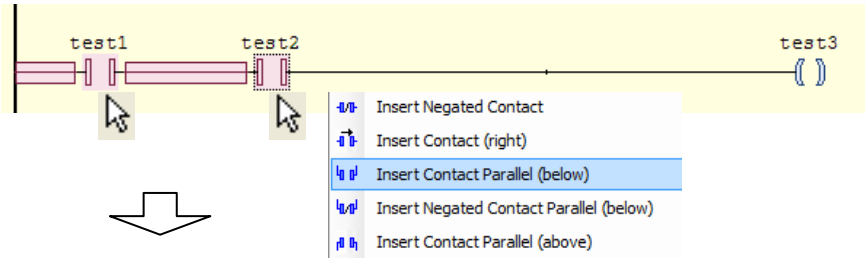
The element is a box that enables you directly enter ST code. LD language is suitable for bitwise operation and ST language is suitable for math calculation. This execute box brings advantages of ST to LD/FBD editor.

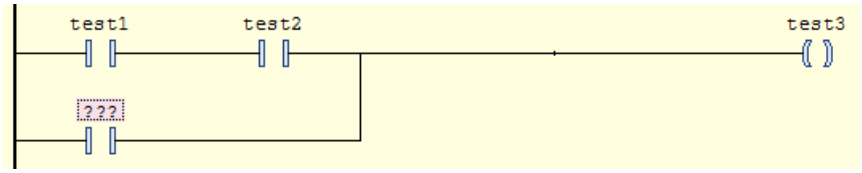




Parallel contact across multiple contacts

If you need to insert a contact in parallel to multiple contacts, choose multiple contacts with pressing [Shift] key, then choose [Insert Contact Parallel (below)] in the right mouse click menu or press [Ctrl+R] key.

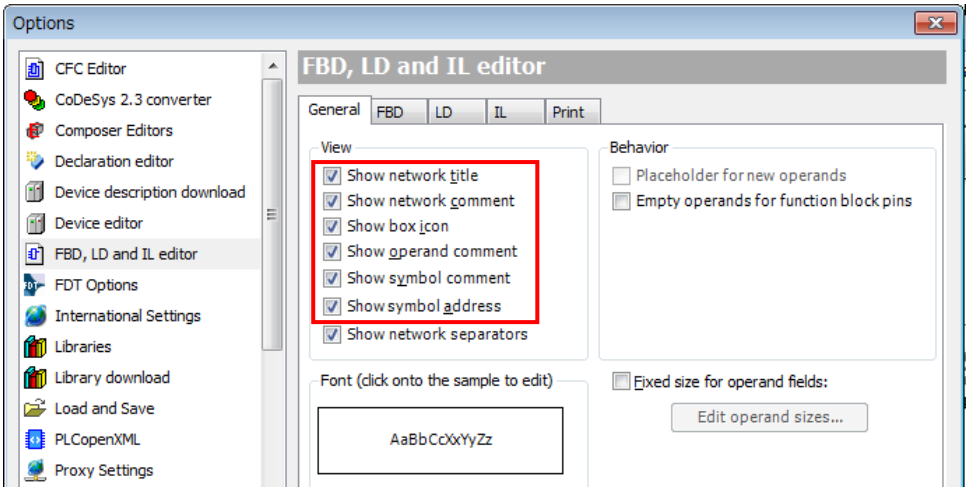


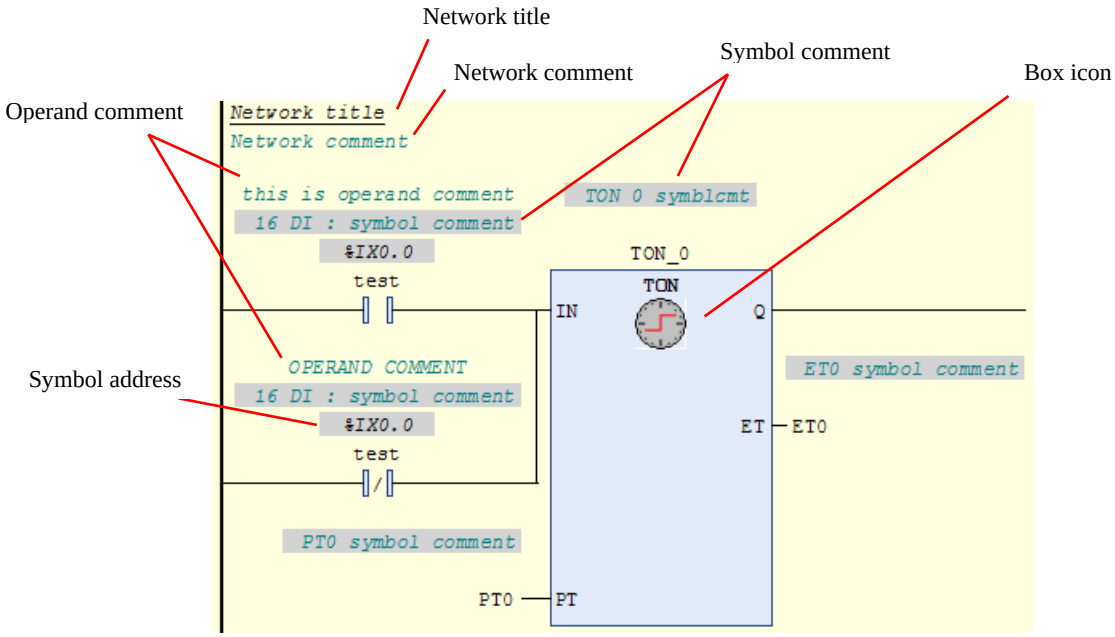


Comment

Several different comments are enabled or disabled in the menu [Tools]-[Options]-[FBD, LD and IL editor].

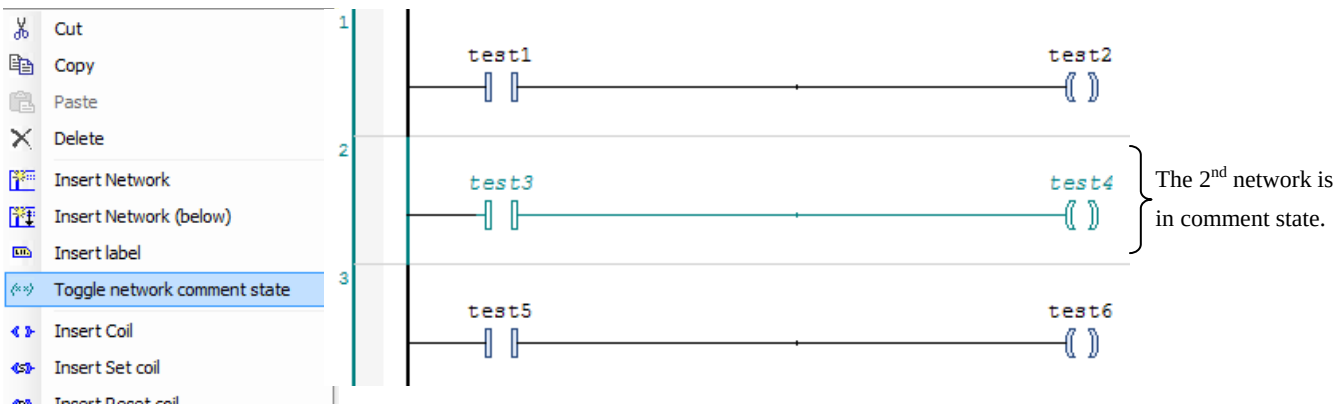
Since symbol comment is linked to variable, if same variables are used in different location, the same symbol comment is used. Operand comment is not linked to variables. If same variables are used, different operand comments can be set.





Toggle network comment state

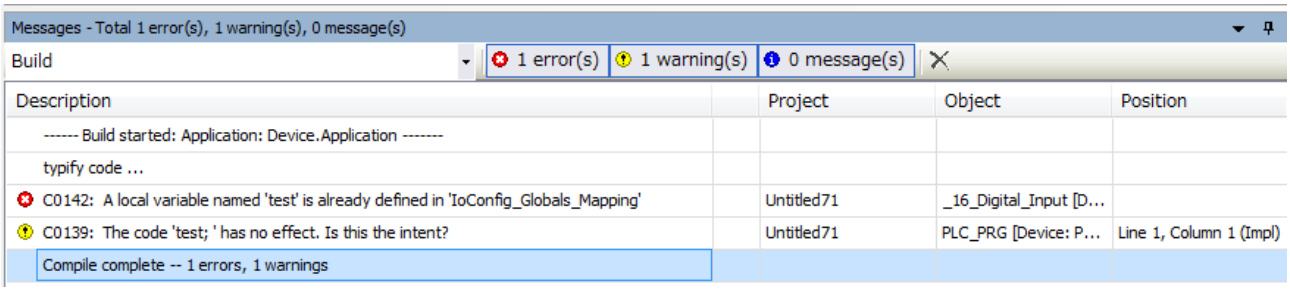
A network can be disabled with visible temporary. Choose [Toggle network comment state] in right mouse click menu. The color of this network will be changed and ignored. Perform the same operation to restore.



2.9 Login / Logout

Login

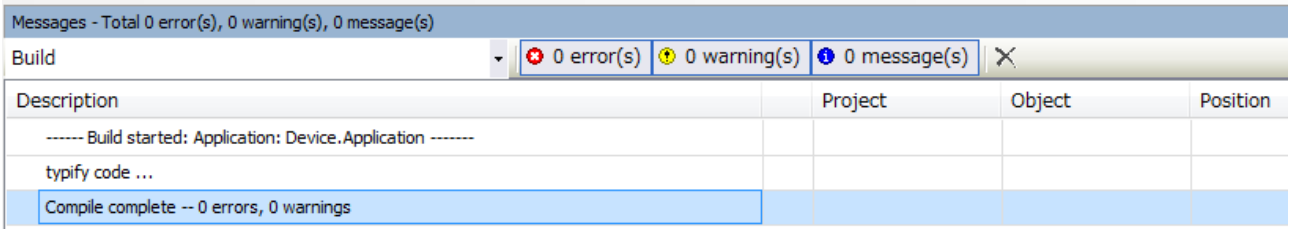
After programming, click  or choose [Build] in Build menu. If compiling fails, error information is shown at [Description] field as follows. Double click the message to jump to the part to be corrected.



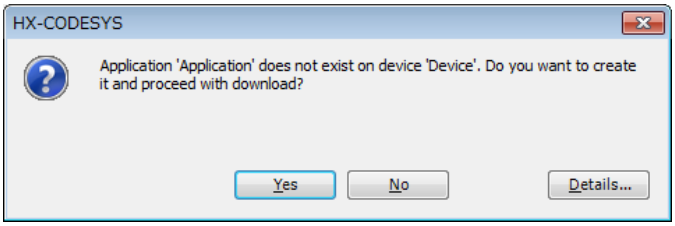
Note

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

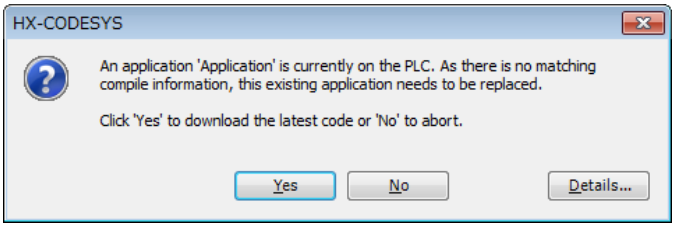
When all errors are removed as below, click  or choose [Login] in Online menu to download the program to CPU.



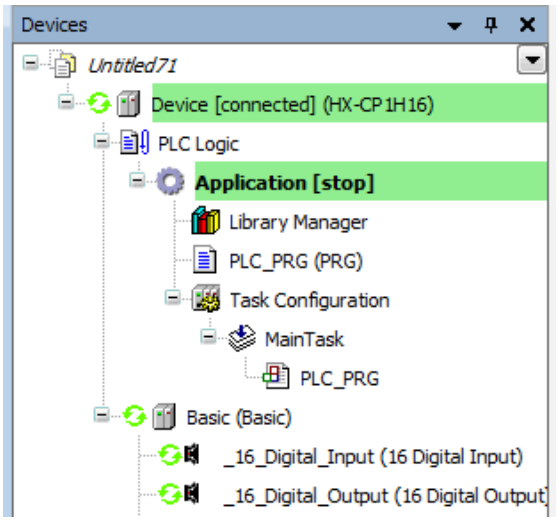
If no application is in the CPU, this message appears. Click [Yes] to download.



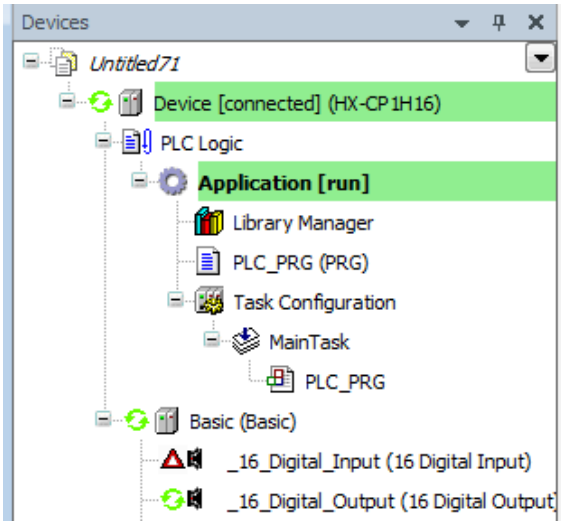
If unknown version of application is in the CPU, this message appears. Click [Yes] to download.



When logging in successfully, green circle icon is displayed at [Device]. If mounted I/O modules are matched with configured ones, green icon is displayed at each I/O module also.



If any mounted I/O module is mismatched, red triangle icon is displayed at mismatched module as below.



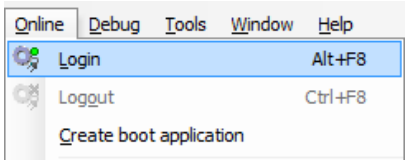
Online change

Online change is a function to change application program while CPU is running without stopping program execution. When online change is performed, only changed program is downloaded to the HX-CPU.

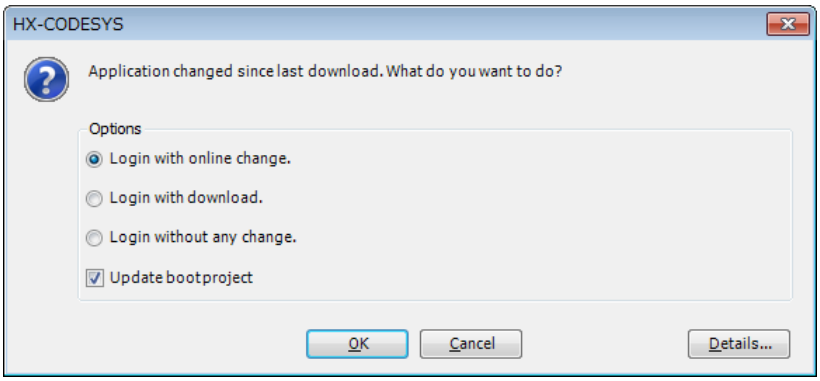
⚠ CAUTION
Since running program is changed by online change operation, the machine might not work as expected, which could cause personal injury or damage. Before performing online change, check new program code carefully.

Operation of 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.



If [Clean] or [Clean all] or changing of I/O configuration, Task configuration or so on is performed before online change, online change is not possible, and download is required.

Note

Pointer variables retain their values from the last cycle. If a pointer refers to a variable whose value was changed in an online change, then the variable no longer yields the correct value. Make sure that pointer variables are reassigned in each cycle.

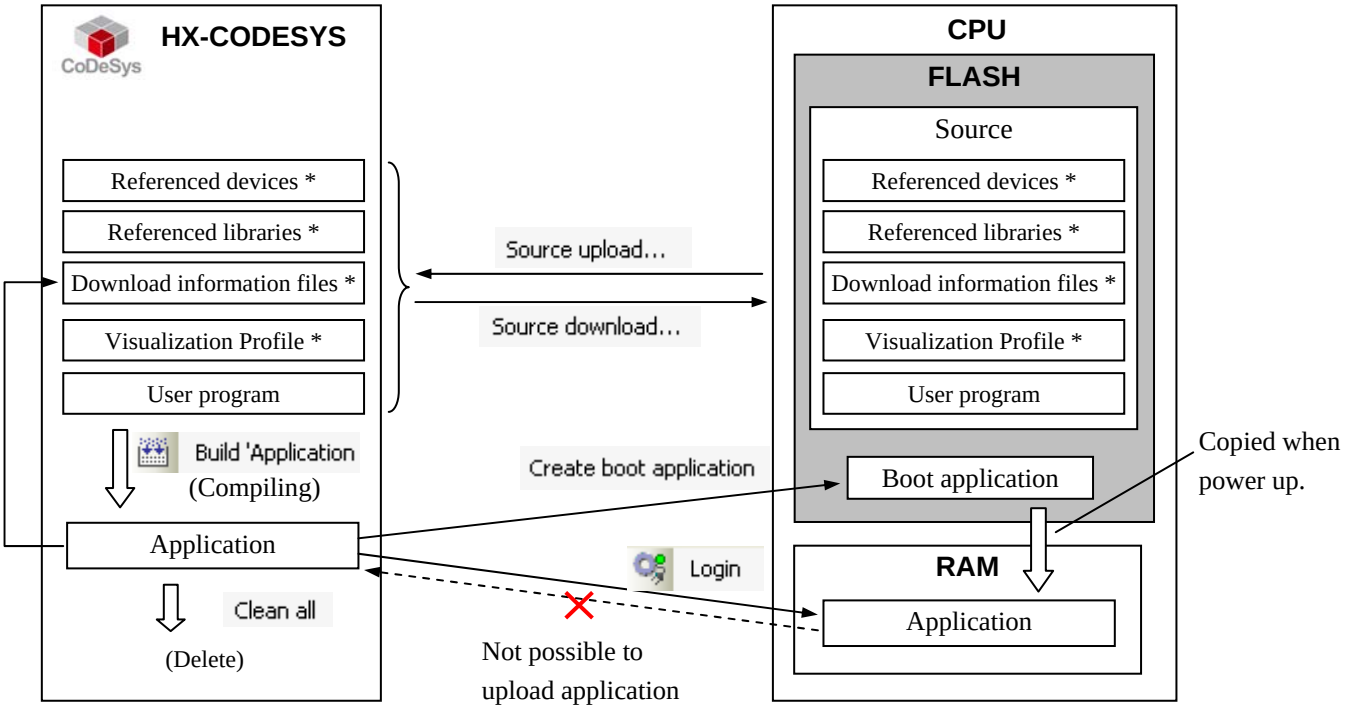
Logout

Choose the menu [Online]-[Logout] or click  icon to logout.

If the total number of variables in opened POU exceeds about 30,000, it takes long time to logout. If 30,000 or more variables are used in a POU, it is recommended to split it 2 or more POUs and not to open all the POUs when login.

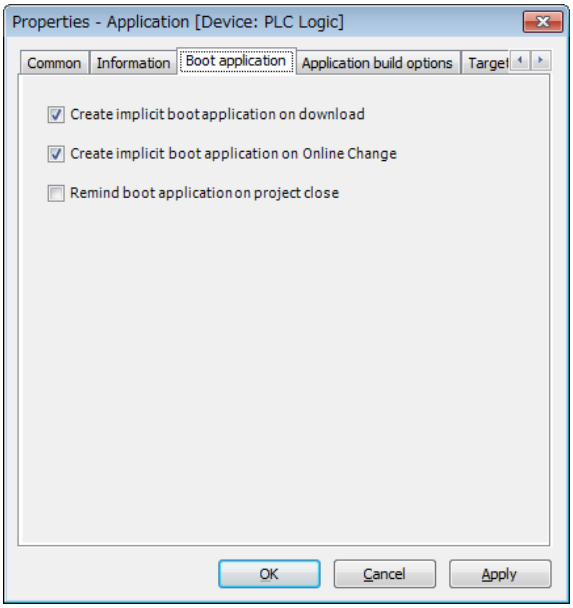
2.10 Boot Application

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.

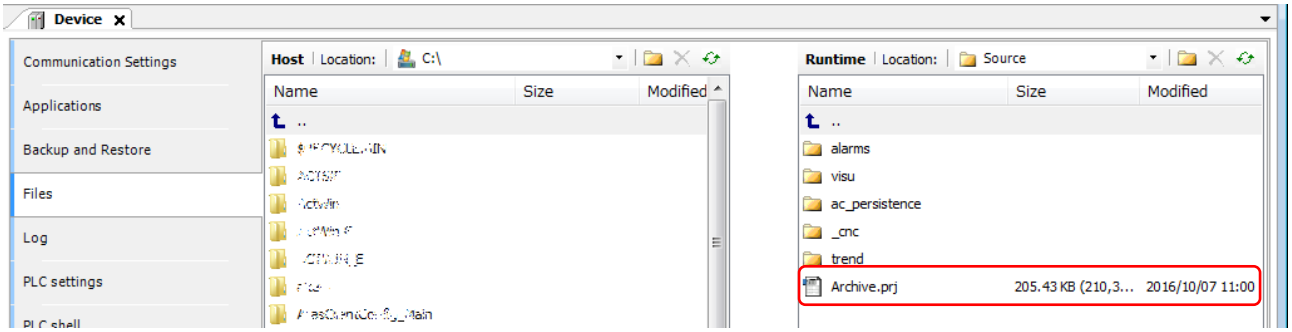


2.11 Source Download / Upload

Path

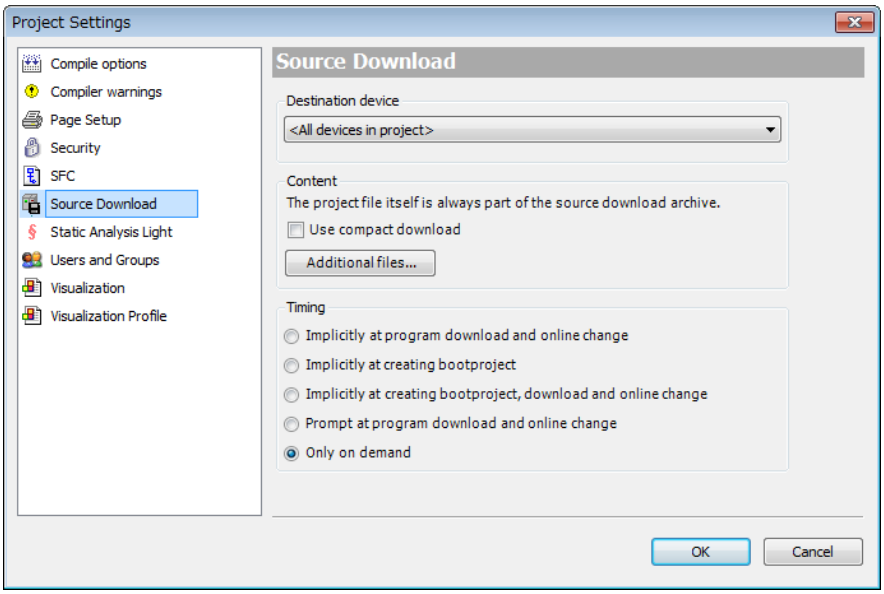
Source file is a file including all the information of project such as programs, variable names, comment, programming language settings and other related information. Source file is downloaded by choosing the menu [Online]-[Source download to connected device] or [File]-[Source download]. Uploading is done by choosing the menu [File]-[Source upload].

If a source file is downloaded to a CPU, it is saved as “Archive.prj” in the folder [Source]. Since executable files such as application file and boot application file are binary format, it is not possible to open as a visible program file or to reverse compile to source file. Be sure to save source files in PC or CPU, otherwise it is not possible to edit or change running program.



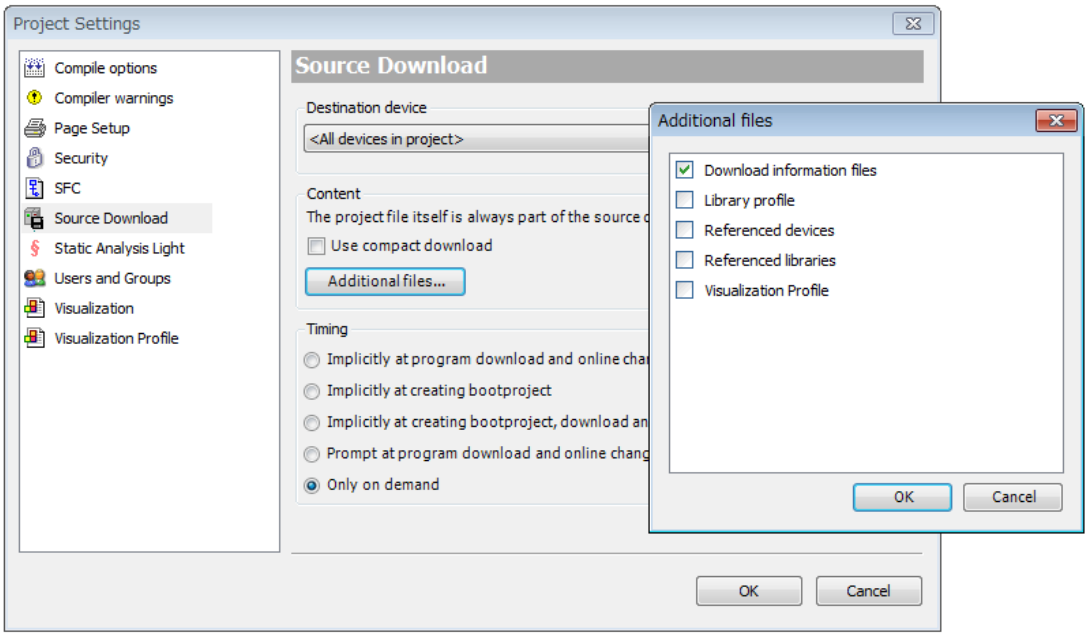
Downloaded timing

Downloaded timing of source file can be set in the menu [Project]-[Project Settings]-[Source Download]. The default setting is [Only on demand]. If source file is to be updated together with application, choose [Implicitly at creating bootproject, download and online change].



Downloaded information files

Besides boot application, source file can be saved in the CPU module, 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.



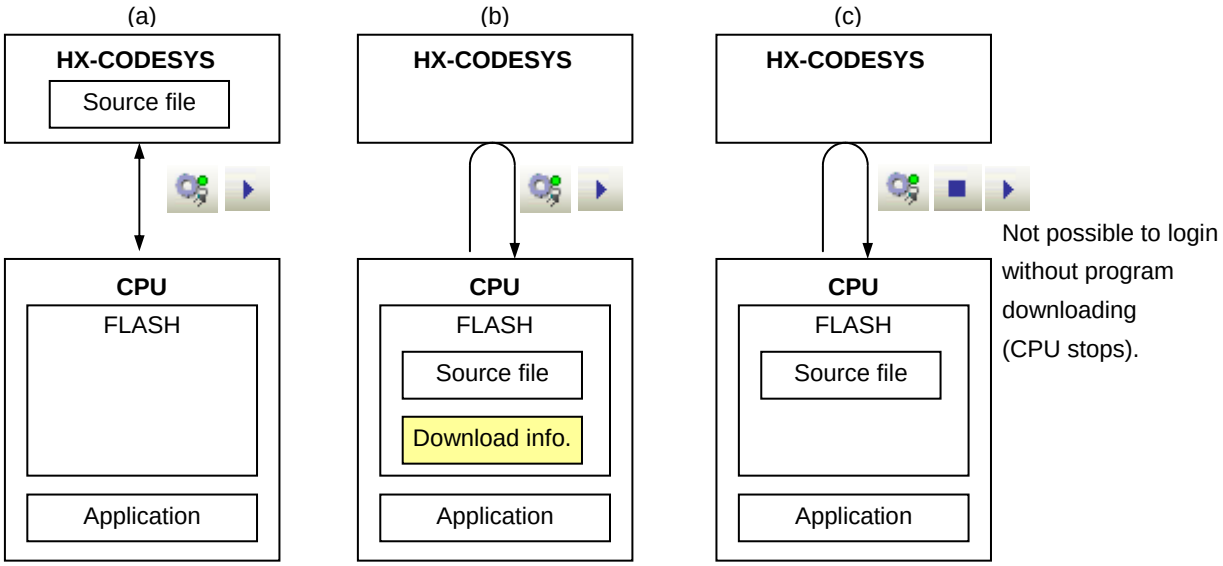
Download information files

[Download information files] in [additional files setting] is not always necessary, 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. ➔ just Login for onlinechange

(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 HX-CODESYS is not able to verify if application and source files are identical. It is possible to login after downloading, but CPU must stop at that time.



2.12 Run / Stop / Reset / Initialize

Run/Stop

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

Switch position / User operations	STOP	RUN
Stop with HX-CODESYS	Stop (ignored)	Stop
Run with HX-CODESYS	Stop (ignored)	Run
Reboot PLC (Cycle power)	Stop	Run *

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

HX-CPU resets all the I/O at starting by hardware signal. For this reason, outputs are reset in one task cycle time at the timing from stop to run independent from PLC settings.

Reset

When CPU detects a serious error called “exception”, such as watchdog error, program execution stops. If HX-CODESYS is connected, “Exception” indication blinks until this status is cleared. This exception status is cleared only by [Reset] operation. HX-CODESYS has 2 different types of [Reset] operation: [Reset warm] and [Reset cold]. All of them can initialize exception status, but behaviours 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.

Initialize

Initializing operation is to reset not only exception status and data memory but also non-volatile memory. HX-CODESYS has two different initializing operation, [Reset origin] and [Reset origin device]. Differences are listed in the table below.

Data / Operation	VAR	VAR RETAIN	VAR PERSISTENT	Application (volatile memory)	Boot application (non-volatile)	Source file (non-volatile)	WebVisu Data, Online user info.	IP address, Realtime Clock data
STOP	X	X	X	X	X	X	X	X
Reset warm	-	X	X	X	X	X	X	X
Reset cold	-	-	X	X	X	X	X	X
Download	-	-	X	(updated)	(updated)	(updated)	(updated)	X*1
Online Change	X	X	X	(updated)	(updated)	(updated)	(updated)	X
Reboot PLC	-	X	X	-	X	X	X	X
Reset origin （Initialize PLC）	-	-	-	-	-	X	X	X
Reset origin device [Device]	-	-	-	-	-	-	-	X

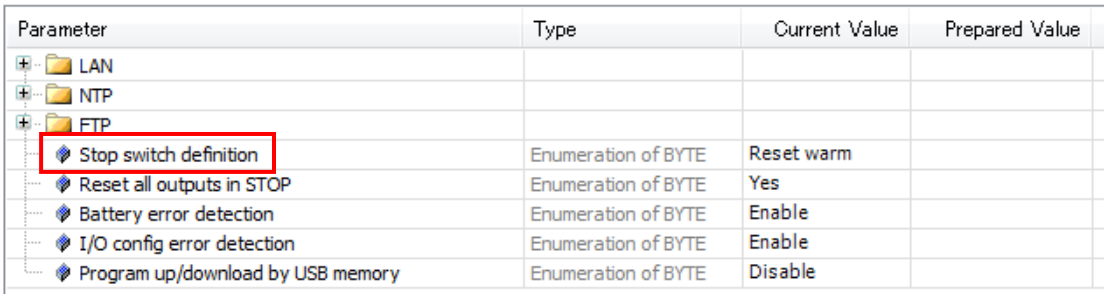
X = maintained, - = initialized, *1: Updated if [Change IP information] is set as YES.

Note

If downloaded application is renamed in [Device]-[File] window of HX-CODESYS, in which the application name in CPU is mismatched with HX-CODESYS, [Reset origin] and [Reset origin device] do not work properly.

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 behaviour as [Stop] of existing Hitachi PLC.



2.13 Global Network Variables

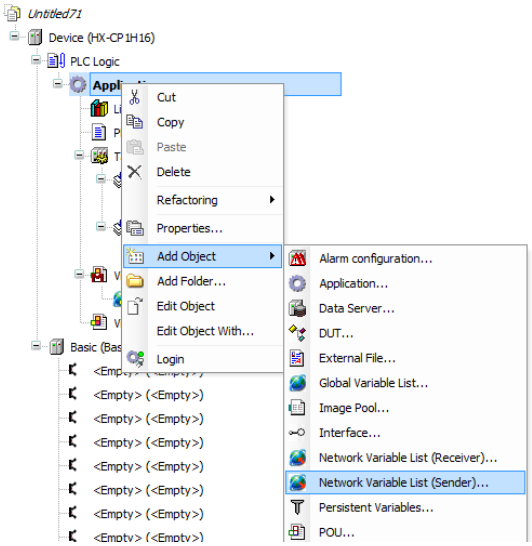
Any variables can be listed in global network variable list, which are sent to all other CPUs (max. 32 CPUs) in the network with broadcast address of UDP/IP. Either port ETH 1/2/3 can be used.

How to configure?

Setting steps for sender CPU and receiver CPU are shown as follows.

[CPU Sender]

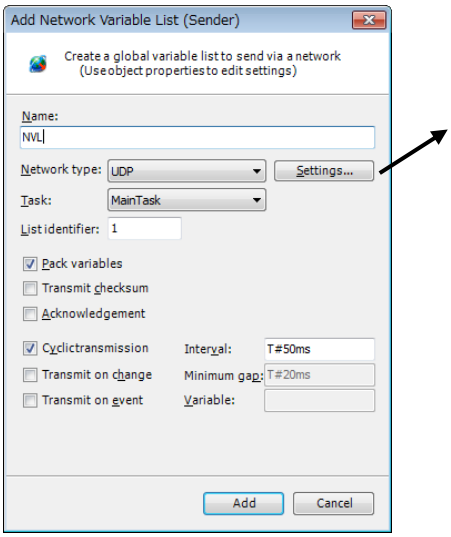
Right click on [Application] of send-CPU and choose [Network Variable List (Sender)].

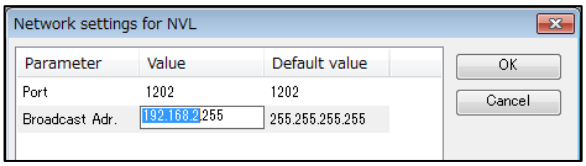


Network type: Choose [UDP].

Settings : Set port number and broadcast address.

Task: Choose any one task. The variables are sent at the end of a task cycle.





Example of Broadcast Adr.
IP address : 192.168.0.1
Subnet mask : 255.255.255.0
Broadcast address in this case is 192.168.0.255

Note

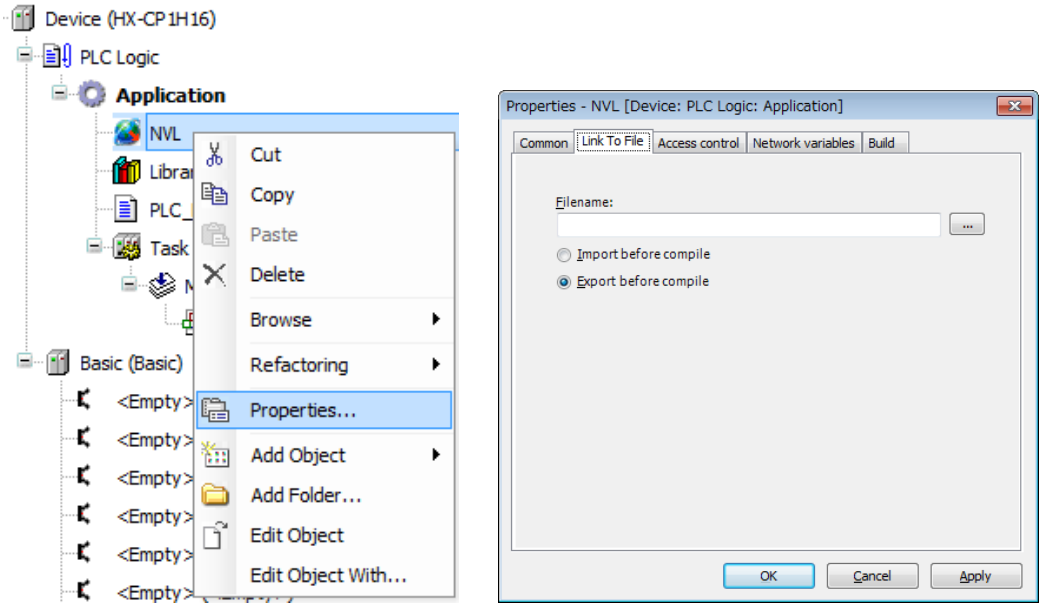
If existing old project is reused for HX and the port number is set as other number than 1202, be sure to set 1202. The port number for network variable is fixed as 1202 in HX-CPU.

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 HX-CODESYS for further information.

After parameter settings of sender CPU are completed, create a file to export to receiver CPU by right mouse clicking and choosing [Properties]-[Link to File]. Be noted that 255 bytes or more of STRING/WSTRING variable are not available in network variables. Parameters of sender CPU can be edited in the tab [Network properties].

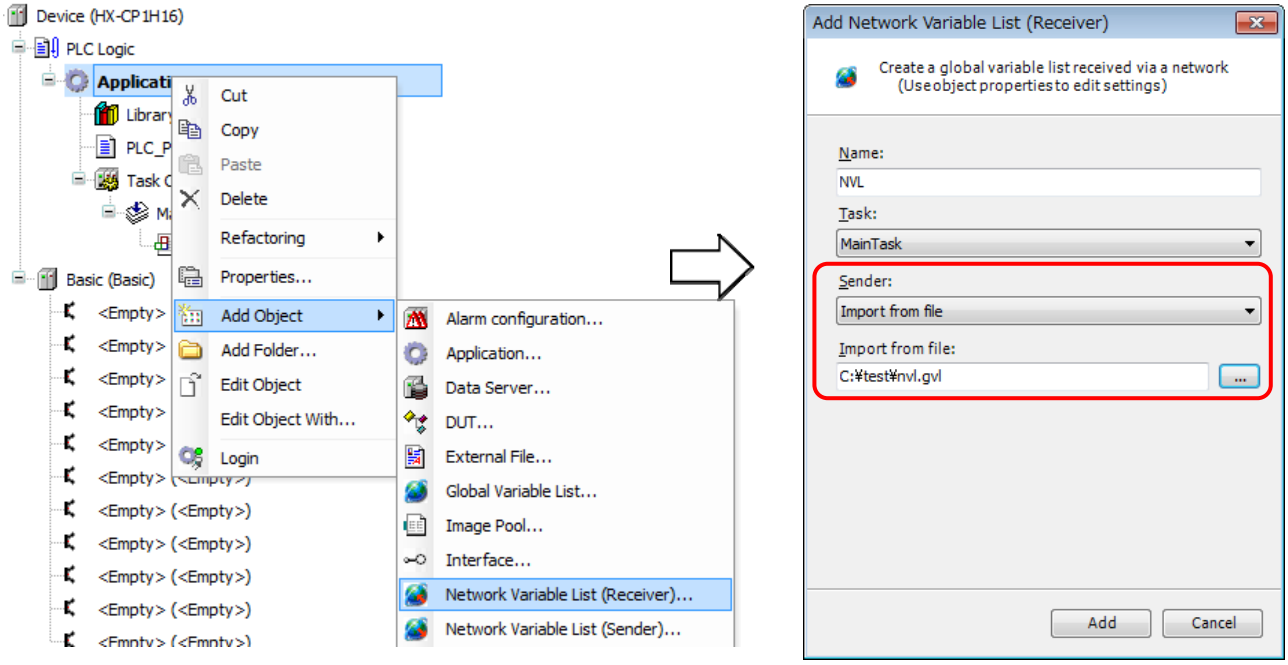
Note

Parameters in properties can be modified in online mode, but new information is not downloaded to CPU when changed. It is recommended to change the parameters in offline mode and then to download.



[CPU Receiver]

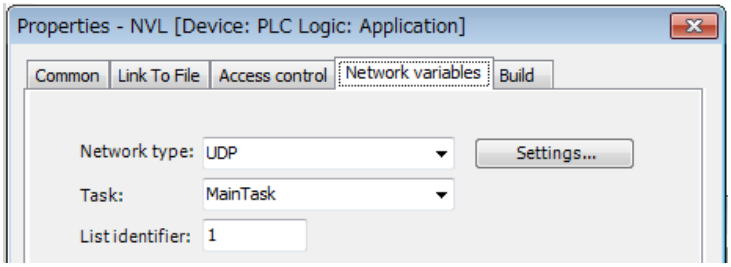
The next step is configuration for receiving CPU. Right click on [Application] of Receive-CPU and choose [Global Network Variable List...]. Be sure to check if Sender is properly set as configured list above.



Configuration is completed for both send and receive-CPU.

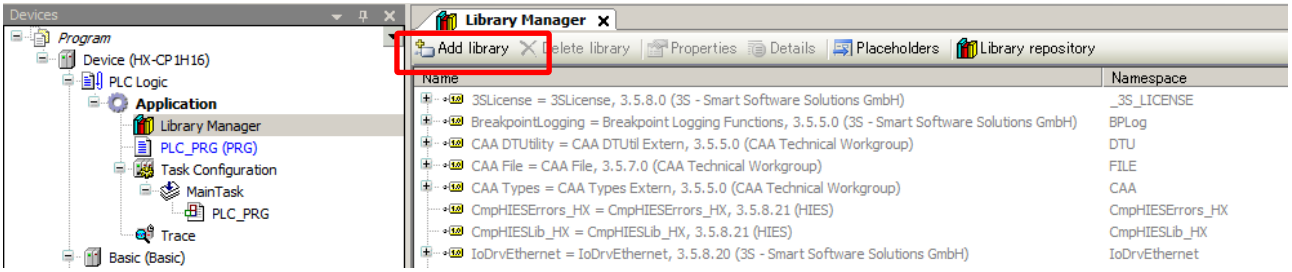
Note

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



2.14 Library

A library is a package including several function or function blocks. The libraries are stored in the Library repository. If you need to use libraries in your project, you must load it from the library repository by [Add library]. In case of HX-CPU, following libraries are preloaded automatically to new project.



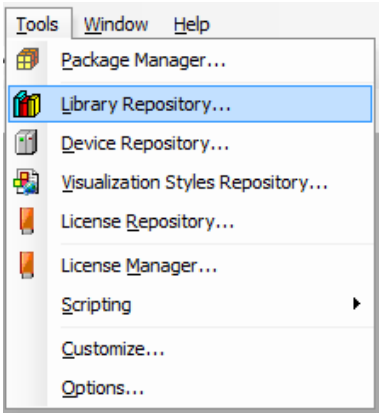
Libraries as shown below are loaded automatically when new project is opened.

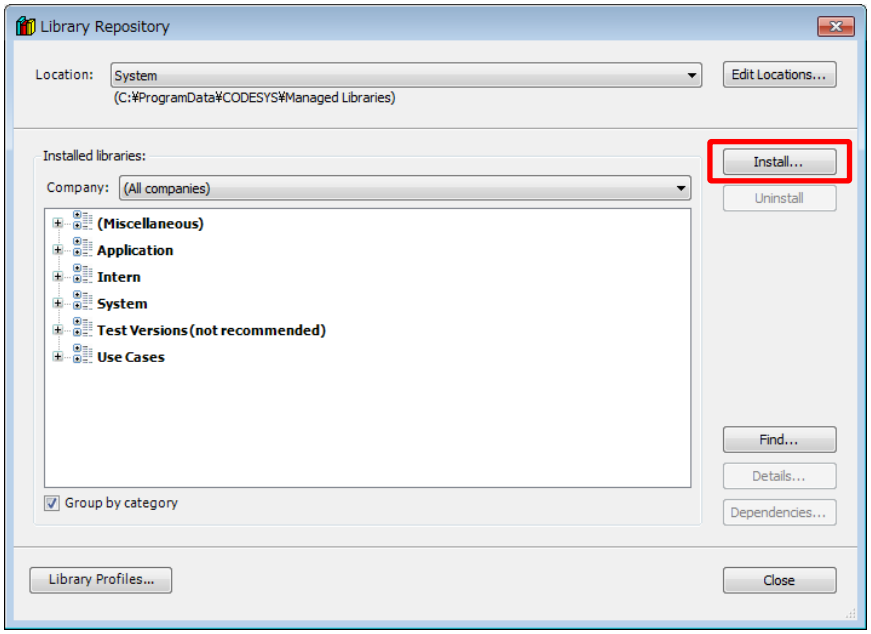
Loaded libraries automatically

Name of Library	Note
IoStandard	System library for I/O control
3SLincense	System library
Standard	IEC61131-3 compliant standard library
Util	PID, BCD and other utility instructions included
CAA DTUtility	RTC (realtime clock) data reading/writing
CAA File	File access library
SysCom	Serial communication library
CAA Types	Sub library for CAA File
CmpHIESLib_HX *1	HX specific instruction library
CmpHIESErrors_HX *1	HX specific error handling library

*1 Use libraries with suffix "_HX". The libraries without "_HX" are for EHV+ series.

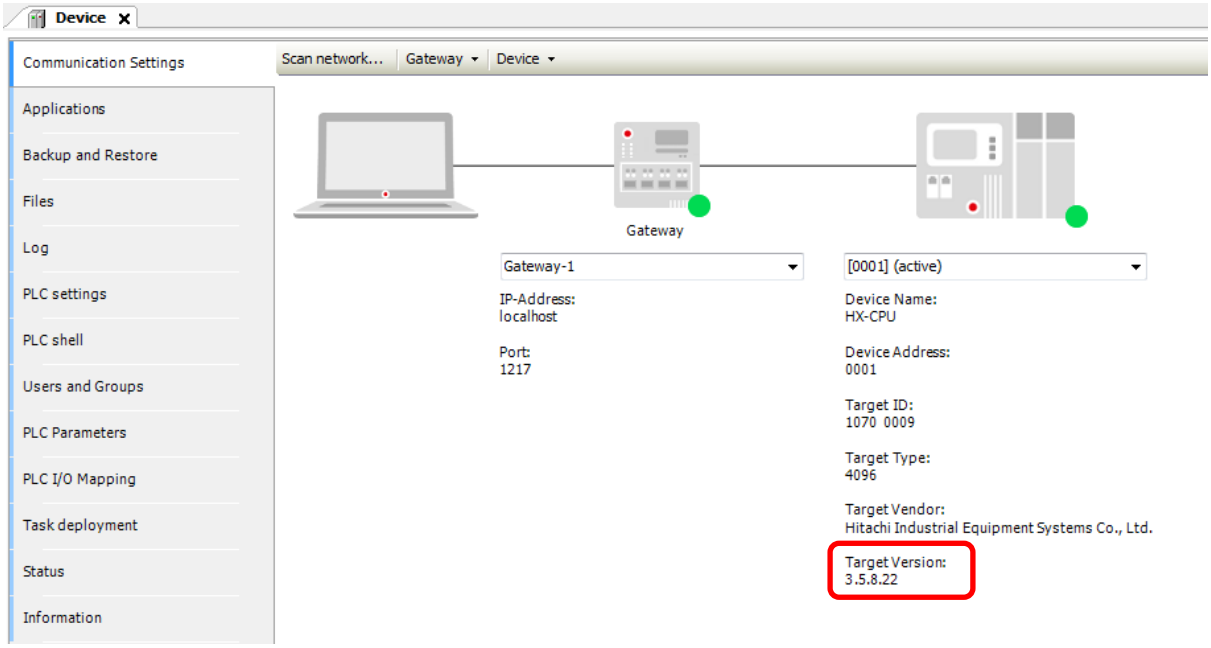
If these libraries are not found in the library manager, install libraries by choosing [Tools]-[Library Repository] and then clicking [Install].





2.15 Version

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



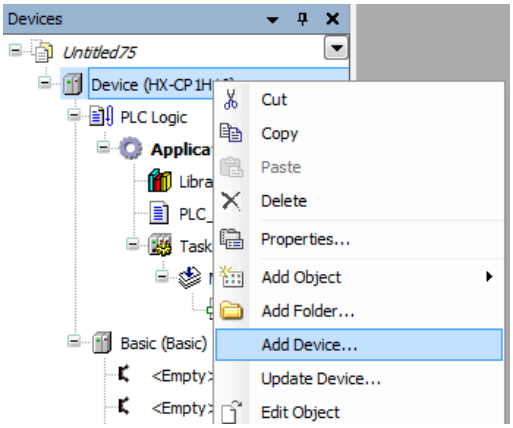
MEMO

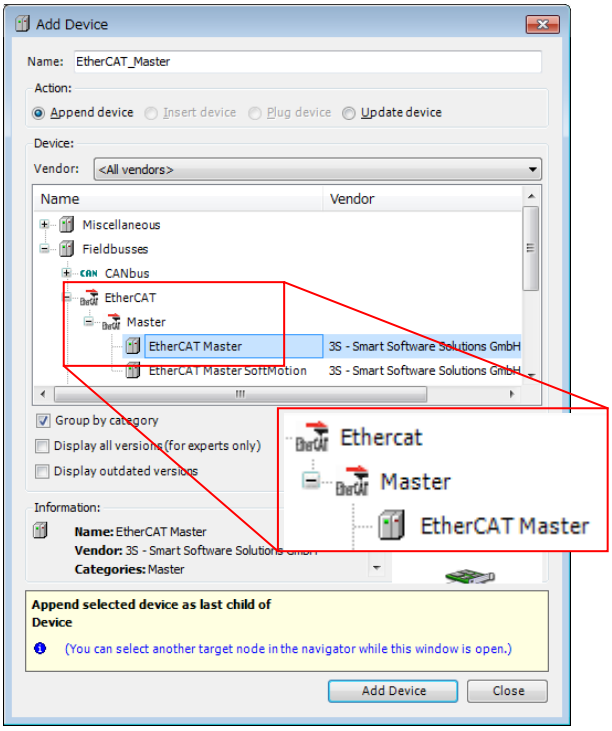
Chapter 3 Communication Function

3.1 EtherCAT Master

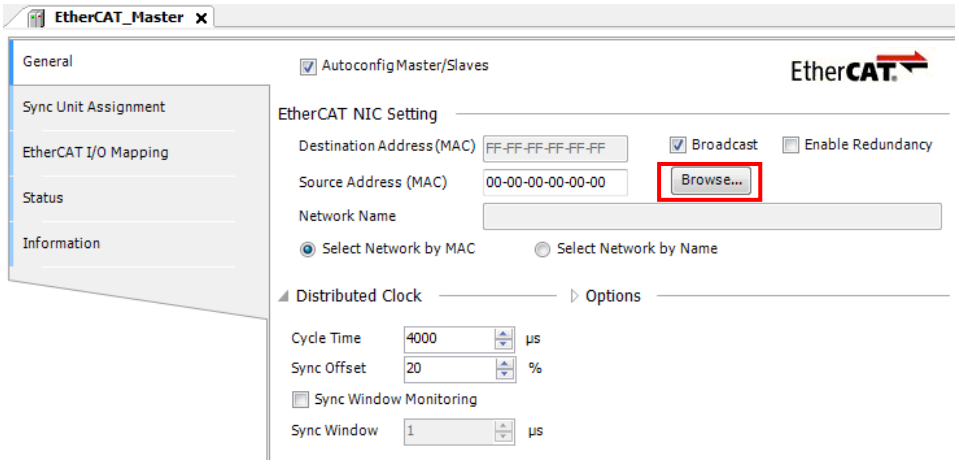
3.1.1 Configuration

Right-click on [Device] and choose [Add Device…].
[Add Device] window appears.
Click [EtherCAT Master] and [Add Device] button.

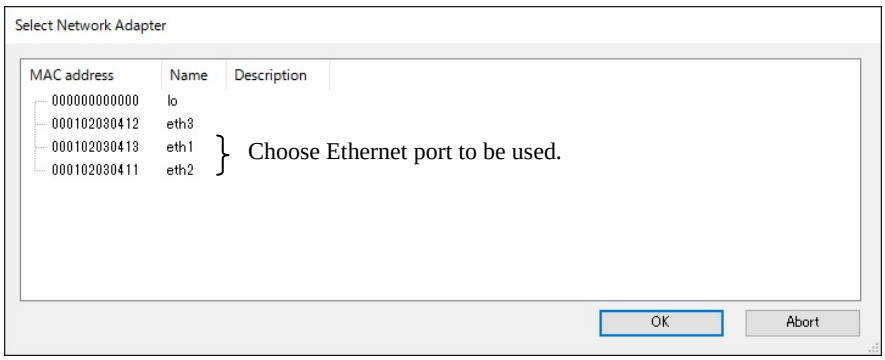




Double click [EtherCAT Master (EtherCAT Master)] to configure Ethernet port. After communication between PC and HX-CPU configured, click [Browse…] button and choose Ethernet port for EtherCAT master.

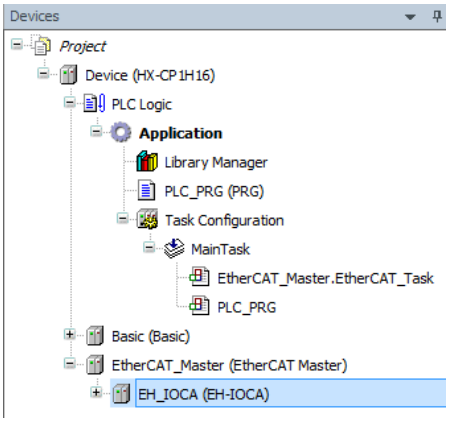


* [Enable Redundancy] is not supported.

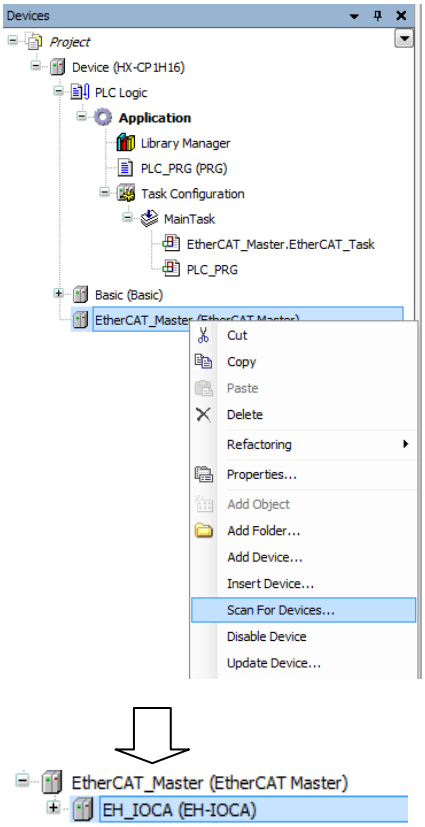


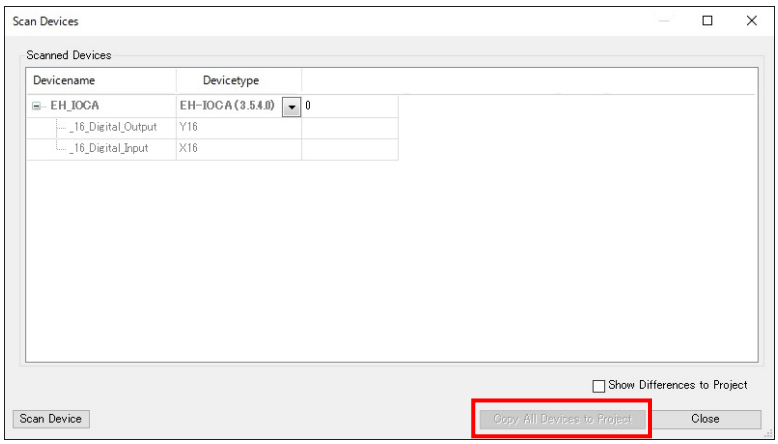
Be sure to use [eth1] or [eth2] for EtherCAT master.

Right mouse click on [EtherCAT master] and choose [Add Device]. The available devices are shown in [Add Device] window. Choose slave units according to your system configuration and click [Add Device] button.



If actual slaves are physically connected, it is possible to search the devices from the network if communication between PC and HX-CPU has been already configured according to chapter 2.7. Right mouse click on [EtherCAT_Master (EtherCAT Master)] and choose [Scan For devices]. Click [Copy All Devices to Project] to complete.

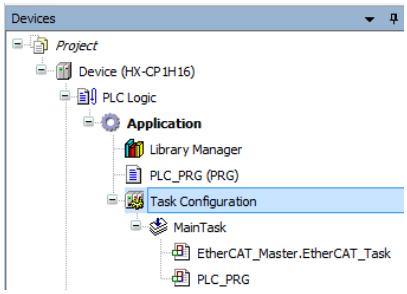




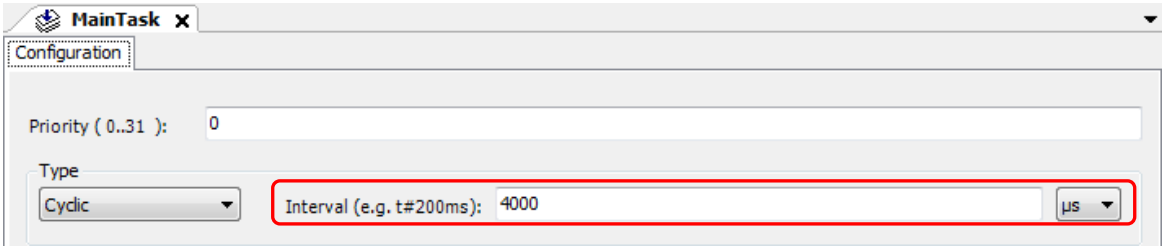
If EtherCAT Slave Information (ESI) file of the slave is not installed, right device name is not shown in the Scan Device dialog. Be sure to obtain the ESI file and install it to HX-CODESYS in advance in the menu [Tools]-[Device Repository].

3.1.2 Cycle of EtherCAT Task

If EtherCAT master is added, EtherCAT Master task is created automatically under MainTask.

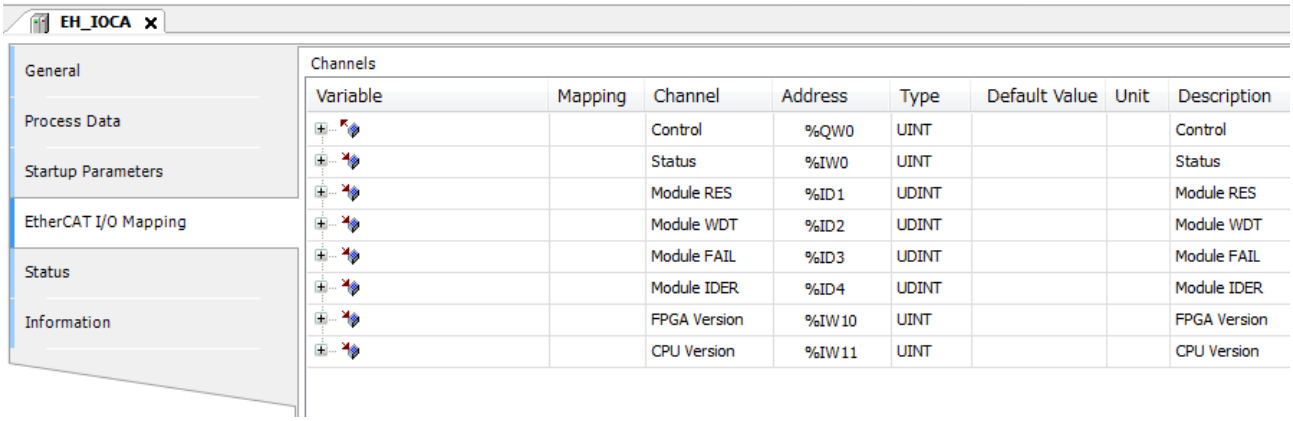


The minimum Cycle time for EtherCAT master is 1 ms because a single microprocessor handles all the tasks in HX-CPU. If cycle time is too small, 25 error (Microprocessor Overload) will be detected. This cycle time depends on user program size and the number of I/O modules however, do not set a value less than 1 ms even if program size and the number of I/O modules are small.



3.1.3 Programming

The mapping of slave's I/O are shown in [EtherCAT I/O mapping] tab as follows. Put variables in this map as same as other I/O modules.



Caution

If PLC is powered up with RUN/STOP switch in RUN position, I/O refresh of EtherCAT slaves may start approximately 1 second later than refresh of standard external I/O due to configuration process of EtherCAT master for slave units. If this delay gives impact your system operation, use xConfigFinished bit. This bit is set when configuration of EtherCAT master is finished. Sample program in ST language is shown as follows.

Sample program

Be sure to use the same instance name as EtherCAT master. Default instance of EtherCAT master is [EtherCAT_Master].

```
IF EtherCAT_Master.xConfigFinished=FALSE THEN
    RETURN;
END_IF;
```

Below program is not executed while EtherCAT_Master.xConfigFinished is FALSE (OFF).

3.1.4 Wiring

(1) Cable

Use category 5 or higher category of STP (Twisted pair with shield) cable.

(2) Hub (switch)

Standard hub (switch) is not allowed to use in EtherCAT network. Special switch for EtherCAT is required if branch topology is required. (Ex. Model CU1128 by Beckhoff)

Caution

Please note that using various Ethernet based communication (EtherCAT master, Modbus-TCP, NVL, Gateway) at the same time will limit the communication performance.

3.2 Modbus-TCP / RTU

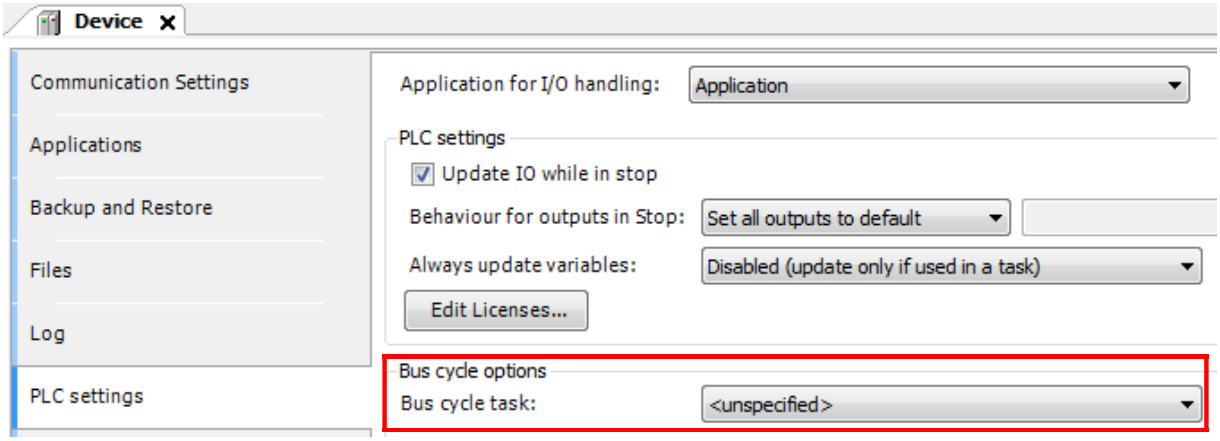
3.2.1 Overview

HX-CPU supports following function codes.

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

X: Supported, -: Not supported

Modbus communication processing is executed by [Bus Cycle Task] specified in [PLC settings] tab of Device. Any task can be assigned for [Bus Cycle Task]. If <unspecified> is chosen, the shortest cycle task is assigned automatically.

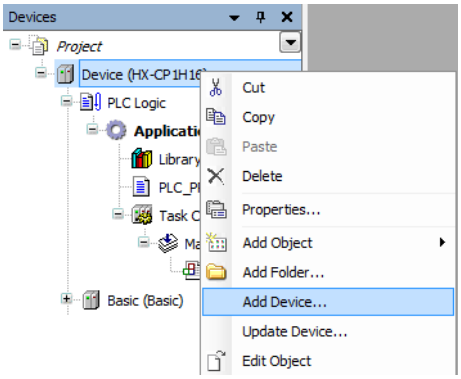


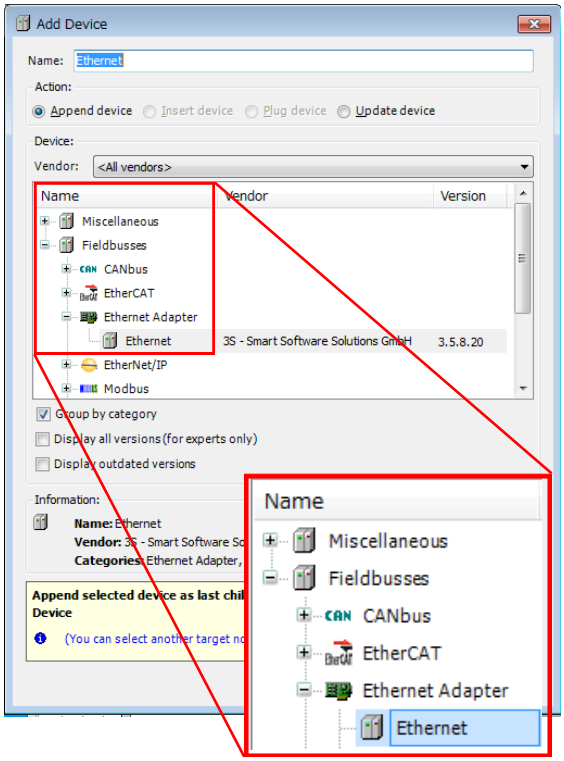
Caution

Modbus-TCP master and Modbus-RTU master do not support broadcast query.

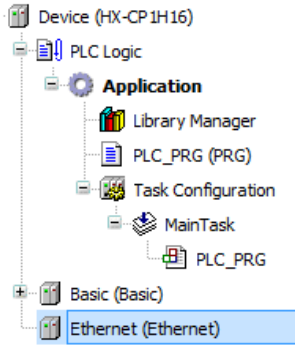
3.2.2 Modbus-TCP Master (Client)

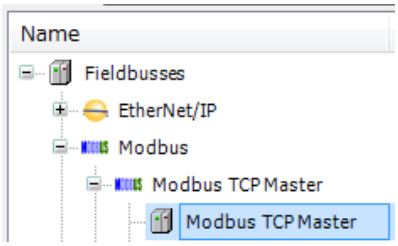
Right-click on [Device] and choose [Add Device…].
[Add Device] window appears.
Click [Ethernet] and [Add Device].



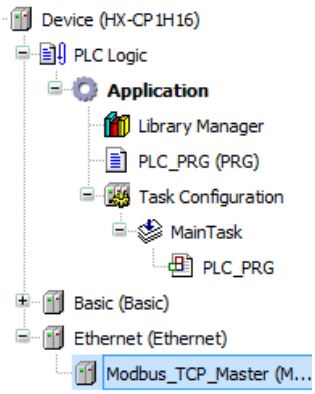


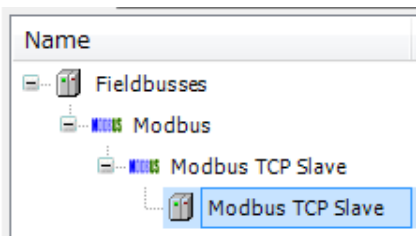
With [Add Device] window opened, click [Ethernet] in the device tree. Then available devices will be shown in the [Add Device] window. Click [Modbus TCP Master] and [Add Device].

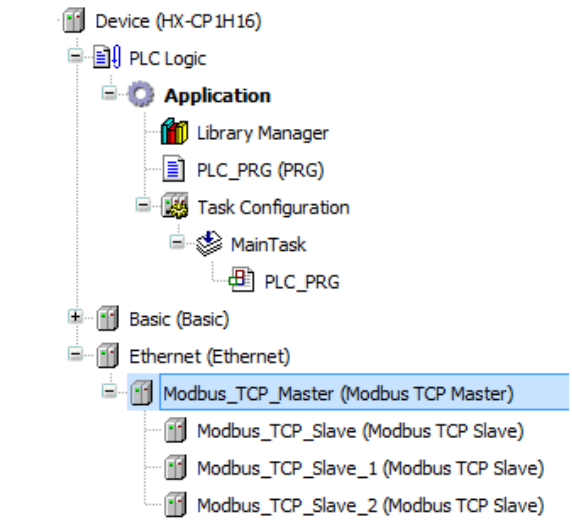




With [Add Device] window opened, click [Modbus_TCP_Master] in the device tree. Then [Modbus TCP Slave] is shown in the [Add Device] window. Click [Modbus TCP Slave] and [Add Device] button according to your Modbus system configuration. If three slave units are used, add three times of slave devices.

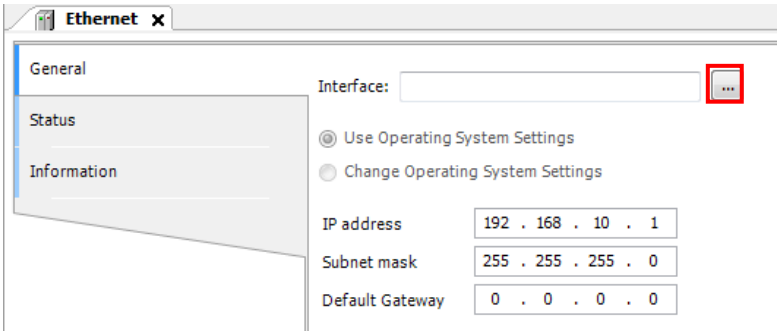


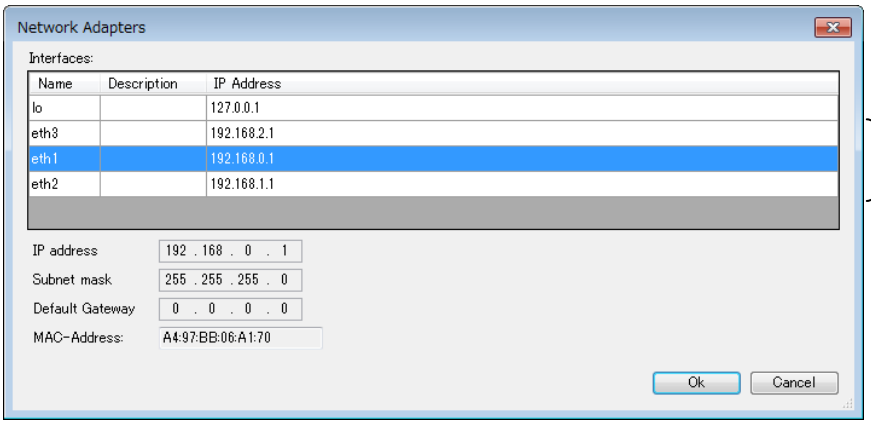




Be sure to configure all slaves to be controlled.

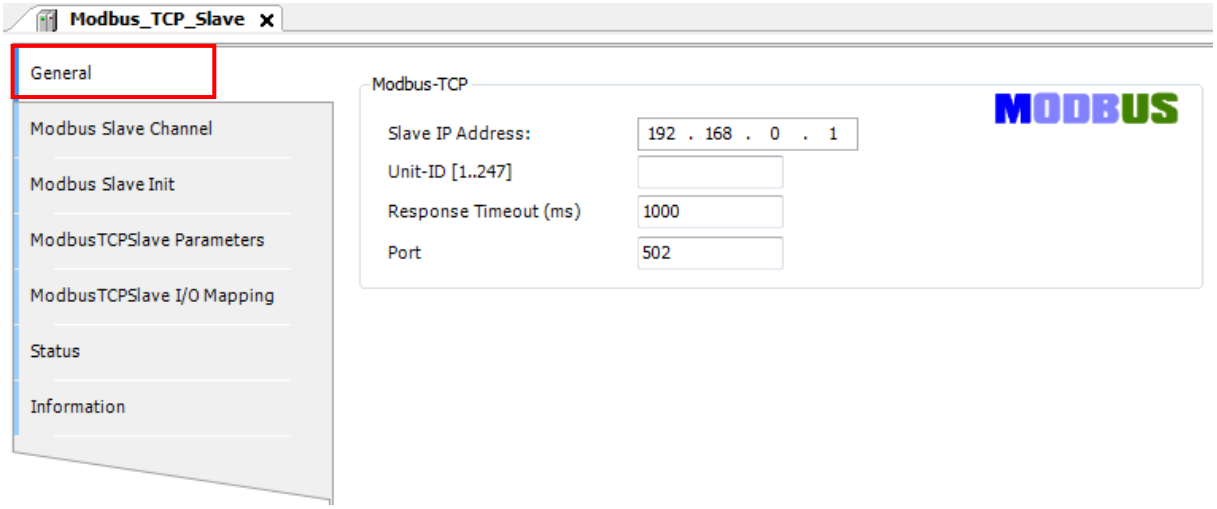
Choose Ethernet port for Modbus-TCP. After communication between PC and HX-CPU configured, click […] button and choose Ethernet port for Modbus-TCP.



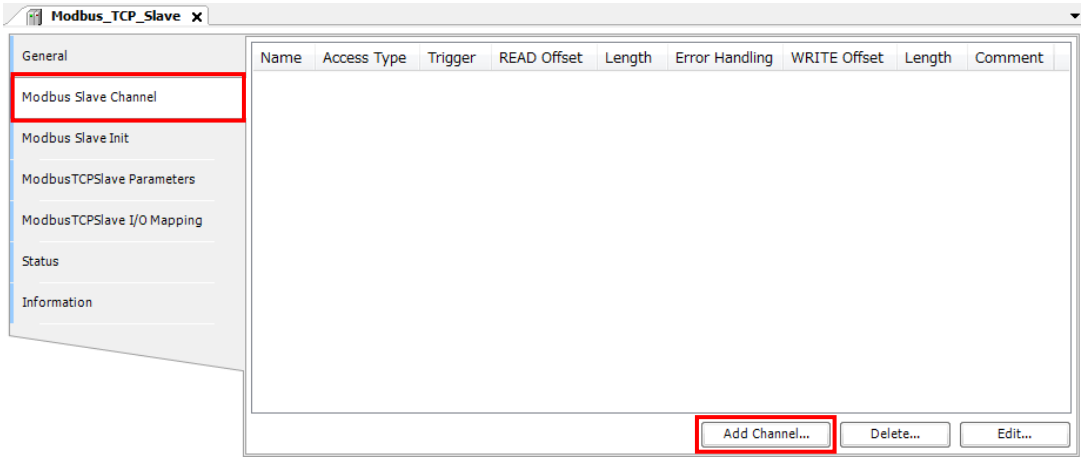


Choose Ethernet port to be used.

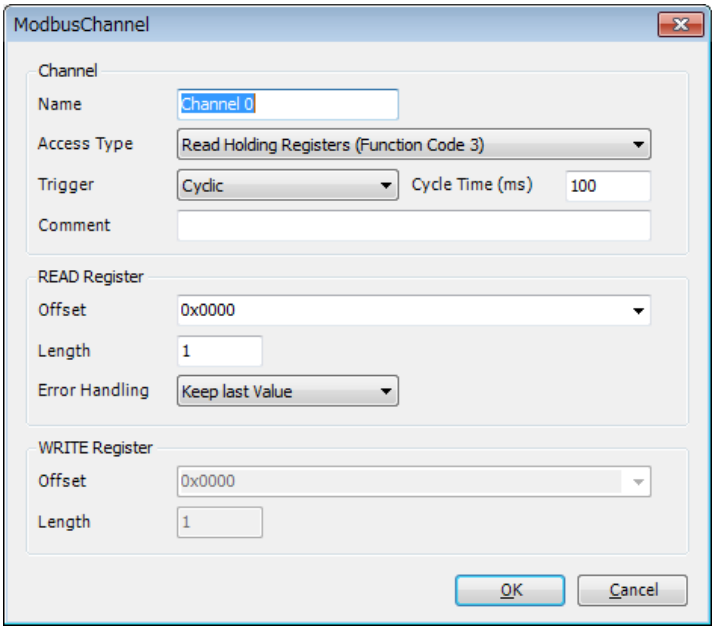
Function codes to be sent must be configured in each slave. Double-click a slave unit to open configuration window. Set IP address, response timeout and port number as follows. Unit-ID is required only when Modbus gateway (Ethernet to serial) device is used.

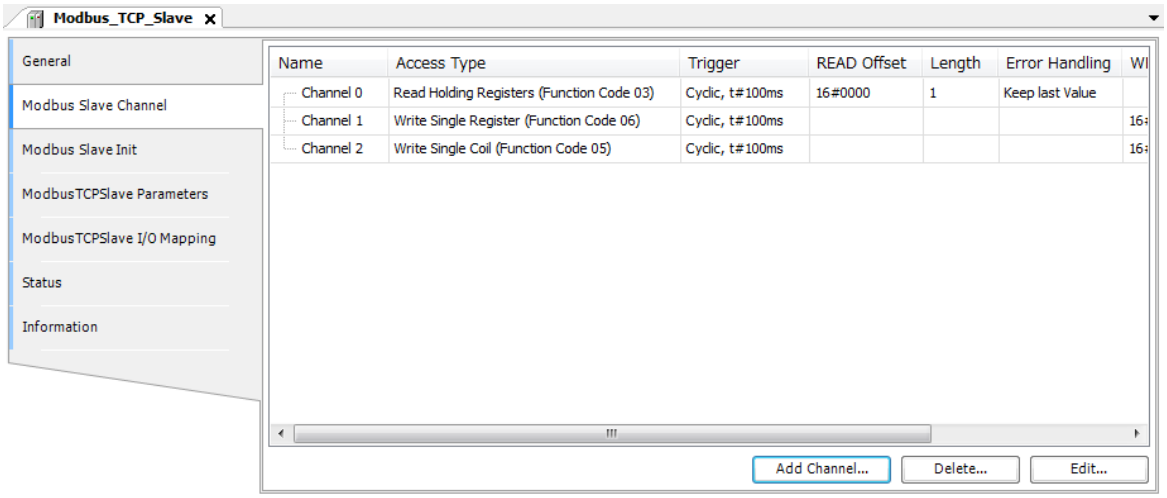


Open [Modbus Slave Channel] tab and click [Add Channel…] to add function codes.

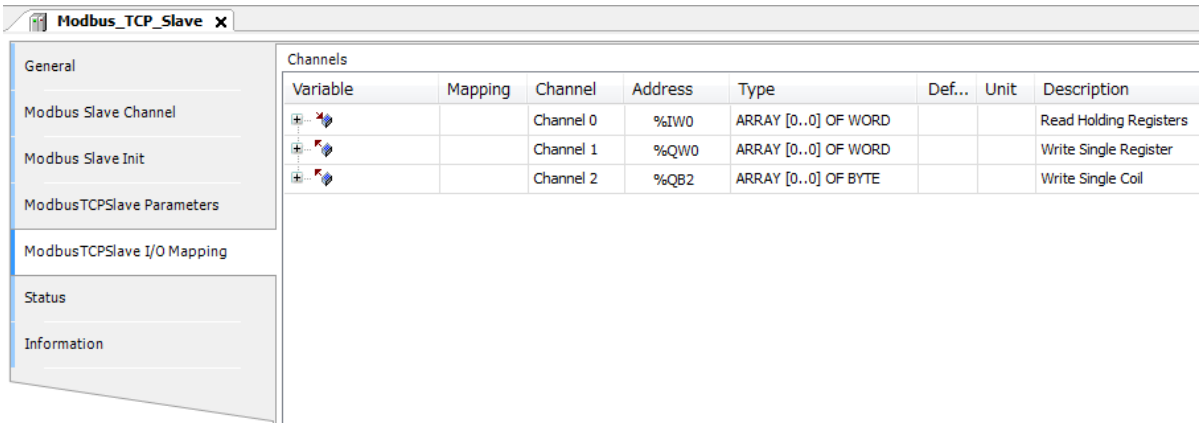


Configure each parameter as below. If the [Trigger] setting is [Rising edge], trigger variable (BOOL) will be automatically assigned in %QX address.





Read and written data from/to slaves is assigned to %IW or %QW as seen in [ModbusTCPSlave 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).

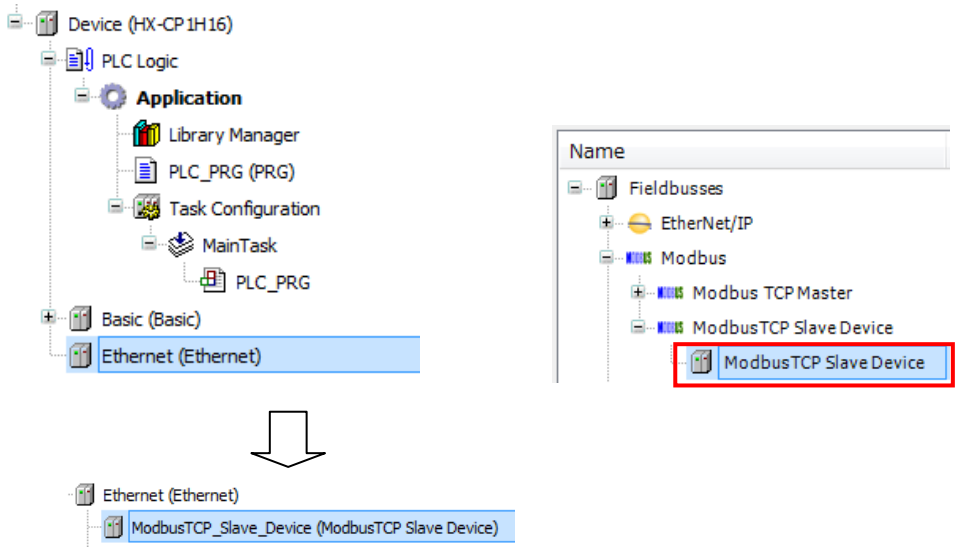


Caution

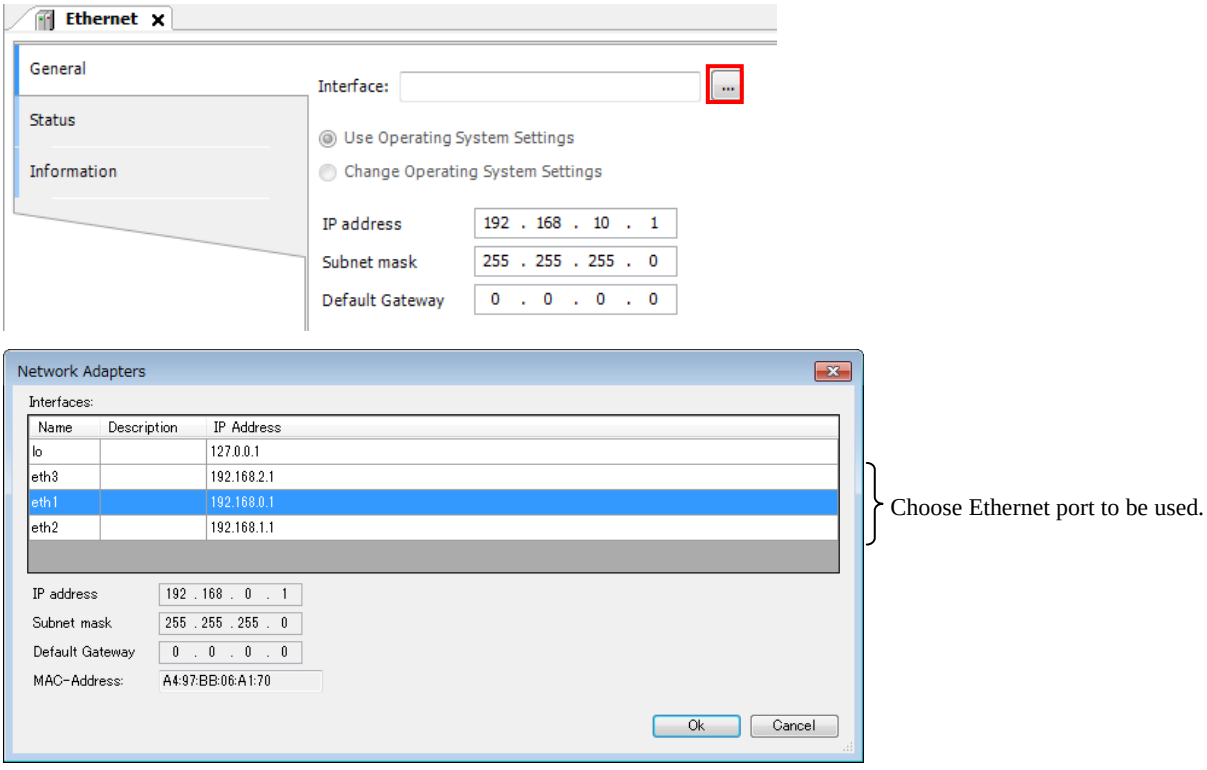
- Modbus master works in the bus cycle task configured in Modbus_TCP_Master. If several channels are configured, single bus cycle can handle only either sending or receiving of one channel.
- If trigger is set as [Rising edge] and this bit is set and reset frequently, command sending and receiving may not work properly. When T1 is defined as time from starting of request command to end of response for channel 1, and T2 is defined as time from starting of request command to end of response for channel 2, do not set or reset trigger variable during T1+T2+...+Tn.

3.2.3 Modbus-TCP Slave (Server)

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



Choose Ethernet port for Modbus-TCP. After communication between PC and HX-CPU configured, click […] button and choose Ethernet port for Modbus-TCP.

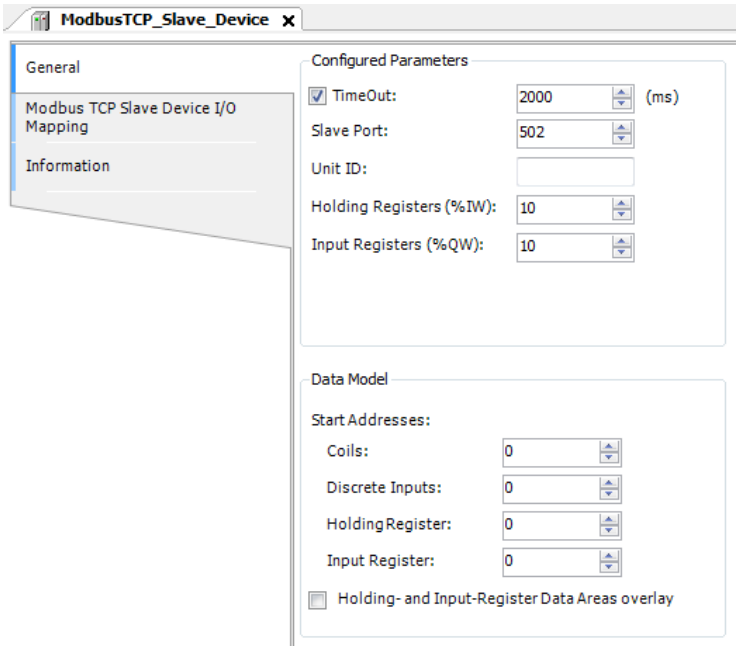


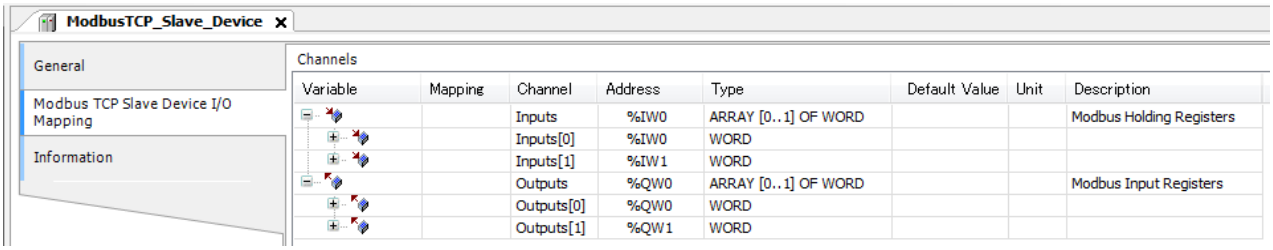
Caution

- Also other Ethernet ports than chosen one work as Modbus-TCP slave however, it is recommended to use configured Ethernet port only.
- Modbus-TCP slave can be added to eth1 to eth3 each, but there is only one Modbus map in the CPU. In this sense, do not configure 2 or more Modbus-TCP slaves. If the same port number is used in several Modbus-TCP slaves, Modbus-TCP slave gets error.
- If warm reset is performed while TCP connection is opened, it takes approximately 1 minute to open the connection again.

Double-click [Modbus TCP Slave Device] and set parameters. Port number for Modbus-TCP is fixed as 502. Do not set other number but 502.

Mapping table will be created in [Modbus TCP Slave Device I/O Mapping] tab according to configured size for Holding Registers (%IW) and Input Registers (%QW).

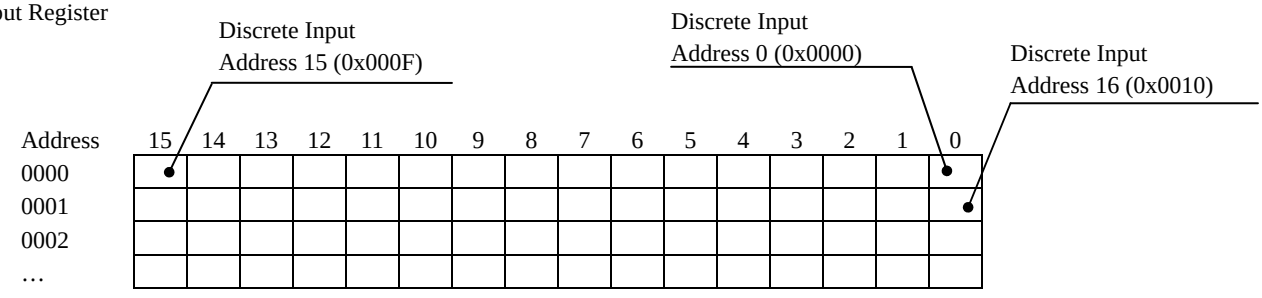




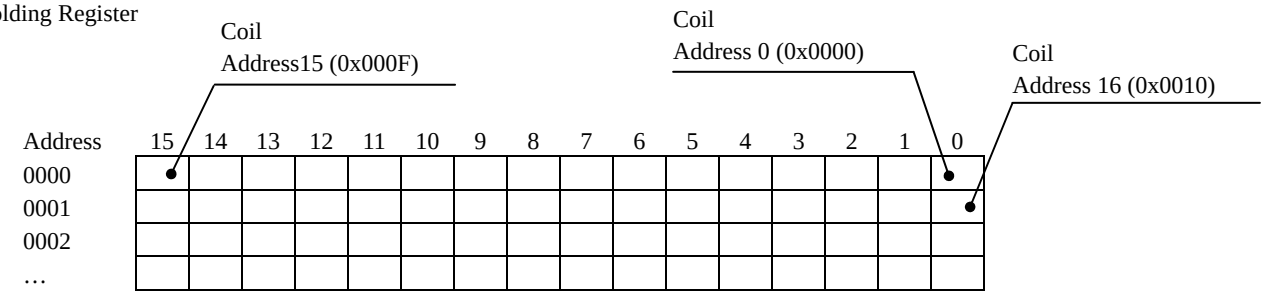
WORD registers and BOOL registers are physically in the same memory as below.

- Input register (WORD) and discrete input (BOOL)
- Holding register (WORD) and Coil (BOOL)

Input Register

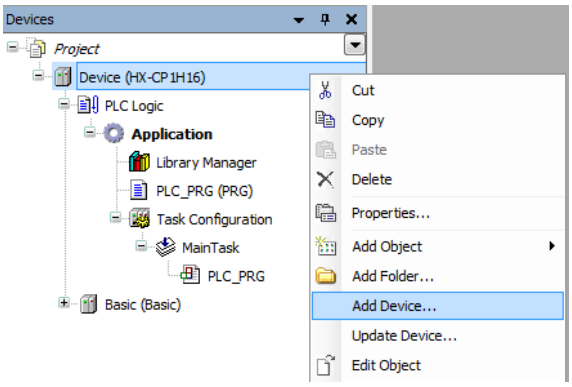


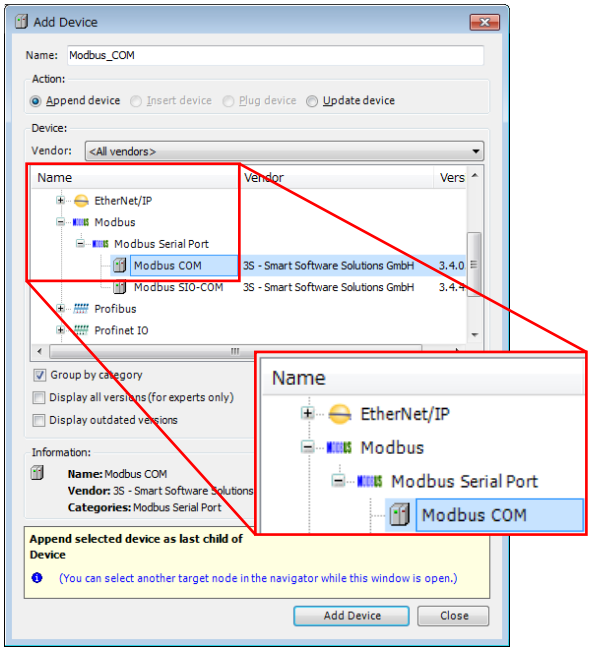
Holding Register



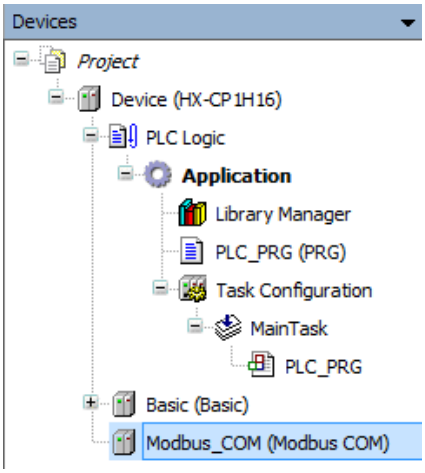
3.2.4 Modbus-RTU Master

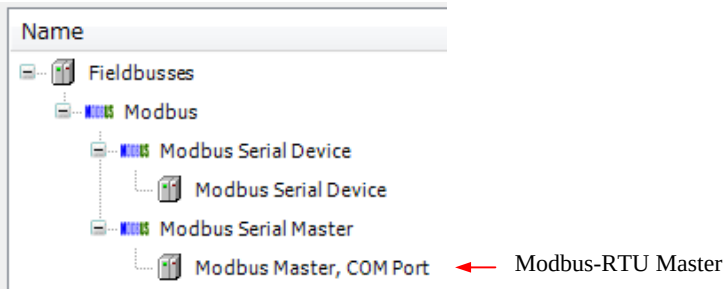
Right-click on [Device] and choose [Add Device…].
[Add Device] window appears.
Click [Modbus COM] and [Add Device] button.



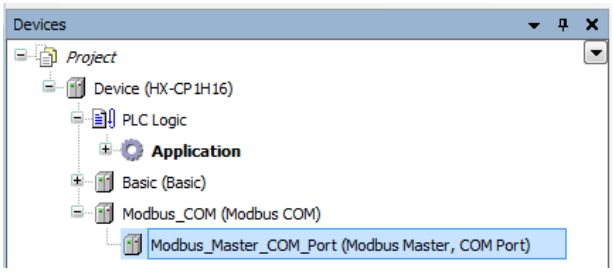


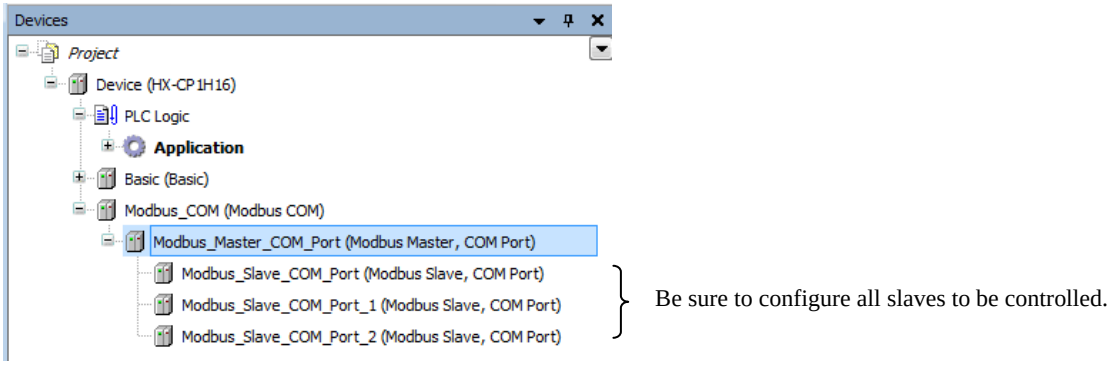
With [Add Device] window opened, click [Modbus_COM] in the device tree. Then available devices will be shown in the [Add Device] window. Click [Modbus Master, COM Port] and [Add Device] button.





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. If three slave units are used, add three times of [Modbus Slave, COM port].

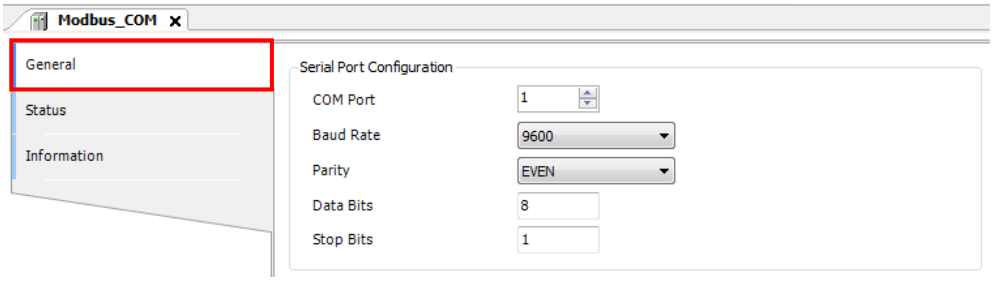




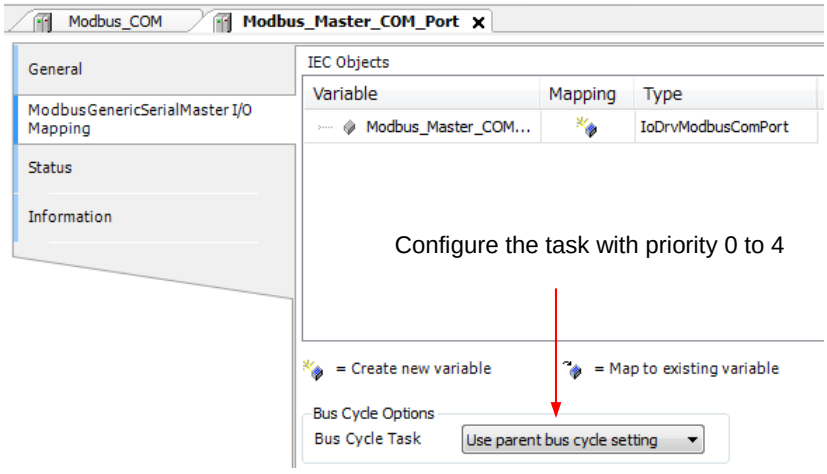
Caution

Be sure to use devices according to Modbus standard. If CPU receives illegal data format, CPU may fail operation.

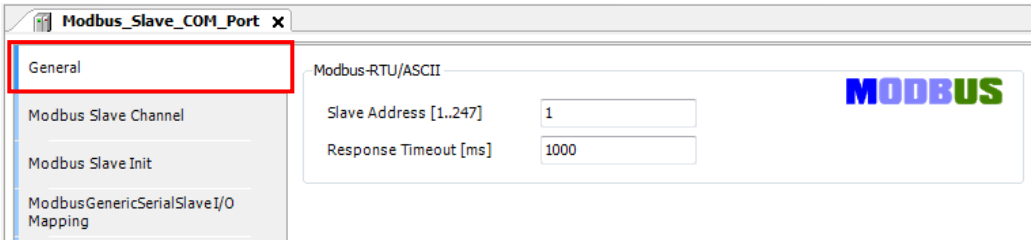
Configure the serial port. Double-click [Modbus_COM (Modbus COM)] to open Modbus_COM window. This configuration must be same as slaves' configuration. COM Port number of HX-CPU is [1]. Available baud rate is from 9600 up to 115200.



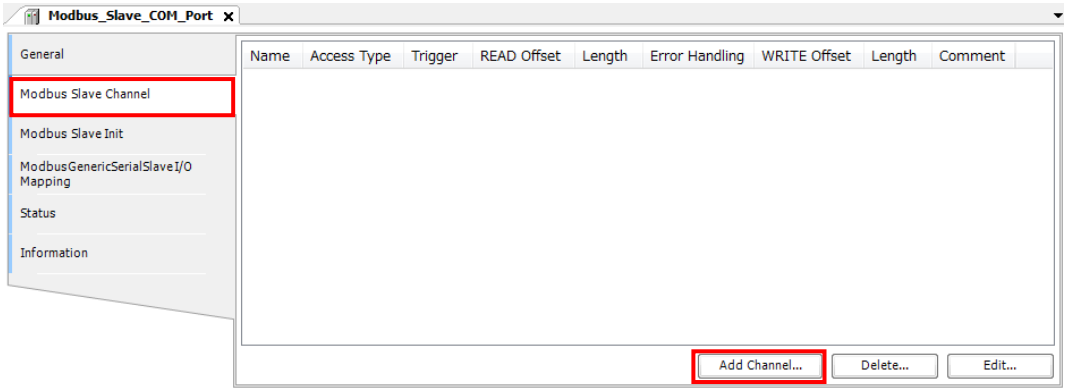
Set the Bus Cycle Task in [Modbus Generic Serial Master I/O Mapping] tab. The priority of assigned task must be from 0 to 4. If 5 or higher priority is set, Modbus may not work properly.



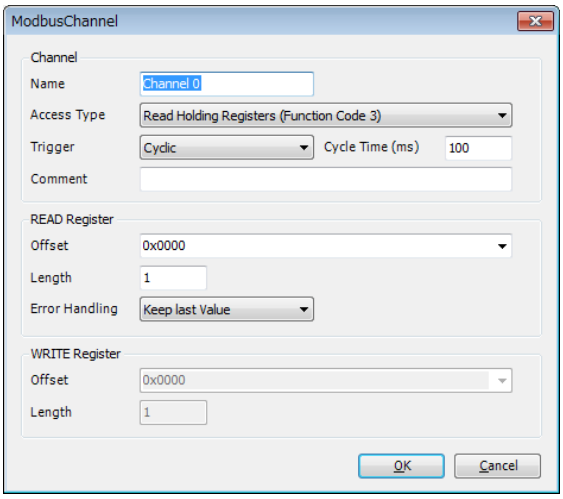
Function codes to be sent must be configured in each slave. Double-click a slave unit to open configuration window. Set slave address and response timeout as follows. Response timeout can be set also in slave individually. If it is set in master and slave both, the value in slave is applied. If response timeout in slave is deleted, then the value in master is applied.

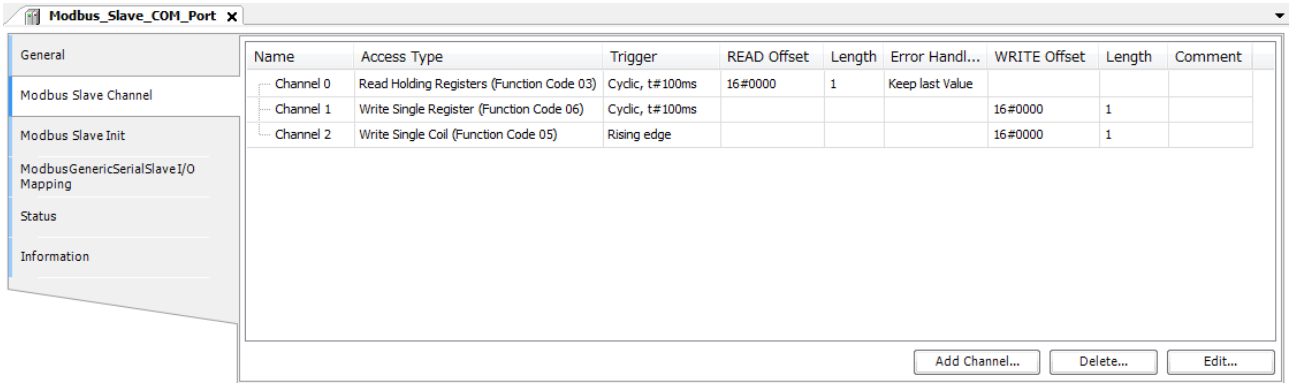


Open [Modbus Slave Channel] tab and click [Add Channel…] to add function codes.

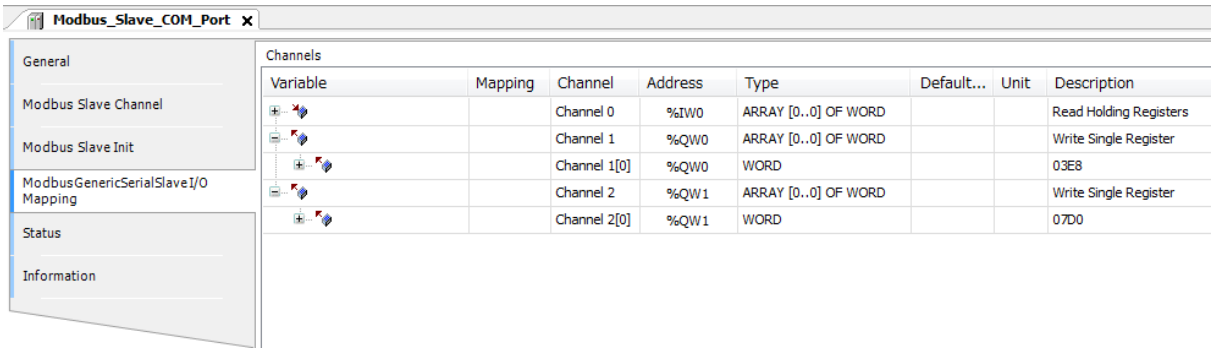


Configure each parameter as below. If the [Trigger] setting is [Rising edge], trigger variable (BOOL) will be automatically assigned in %QX address.





Read and written data from/to slaves is assigned to %IW or %QW as seen in [ModbusTCPSlave 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). [Default Value] is written once when the status changes from RUN to STOP.

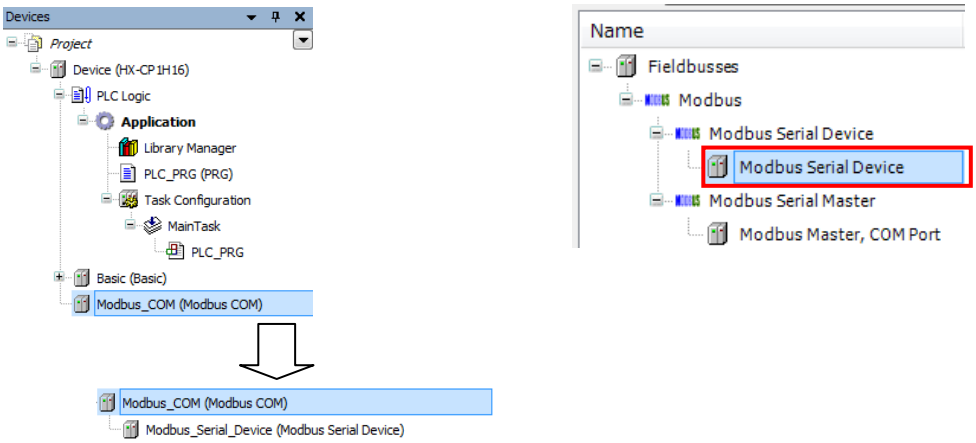


Caution

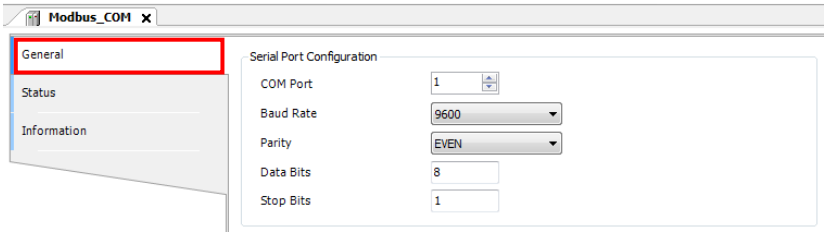
- Modbus master works in the bus cycle task configured in Modbus_Master_COM_Port. If several channels are configured, single bus cycle can handle only either sending or receiving of one channel.
- If trigger is set as [Rising edge] and this bit is set and reset frequently, command sending and receiving may not work properly. When T1 is defined as time from starting of request command to end of response for channel 1, and T2 is defined as time from starting of request command to end of response for channel 2, do not set or reset trigger variable during T1+T2+...+Tn.

3.2.5 Modbus-RTU Slave

Right-click on [Modbus COM] and choose [Add Device…]. Click [Modbus Serial Device] in the [Add Device] window and [Add Device] button.

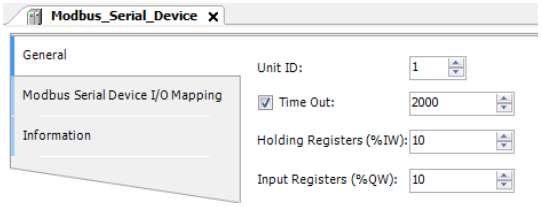


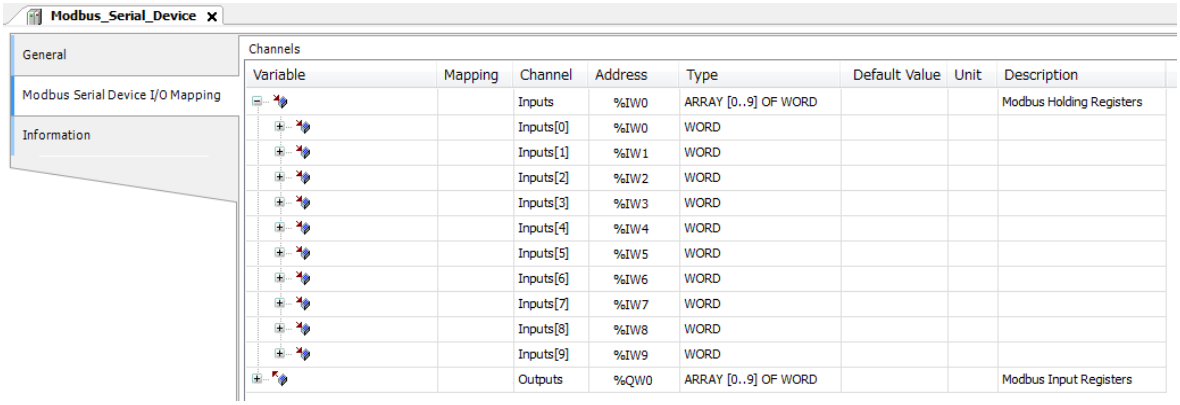
Configure the serial port. Double-click [Modbus_COM (Modbus COM)] to open Modbus_COM window. This configuration must be same as master and other slaves' configuration. COM Port number of HX-CPU is [1].



Double-click on [Modbus Serial Device] and set parameters.

Mapping table will be created in [Modbus Serial Device I/O Mapping] tab according to configured size for Holding Registers (%IW) and Input Registers (%QW).





Caution

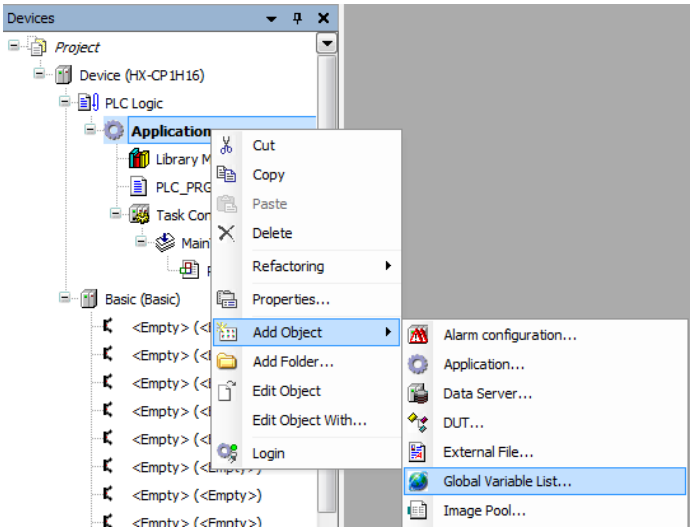
Query from master must be according to Modbus standard. If unsupported function codes, illegal address, data or the number of data are sent, HX-CPU may not send back exception responce properly.

3.3 CPU Link

3.3.1 Overview

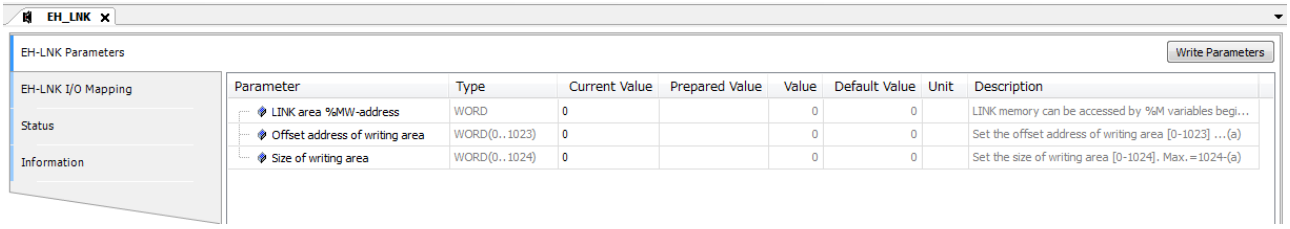
CPU Link System is the network of CPU Link modules connected in loop topology. Common memory called link area is read or written by each CPU in the network. Link area in each CPU is defined as Global Variable List (GVL) with %M address and shared by all the CPUs.

GVL is added by right-clicking on [Application] shown below.



3.3.2 Configuration Link Parameter

Add [EH-LNK] device to the device tree in advance. Double-click [EH-LNK] and set [Offset address of writing area] (Writing area %MW offset address) and [Size of writing area].



Caution

Allowed range of values are shown below. HX-CODESYS does not detect error if the value is out of the range.

- Offset address of writing area : 0 to 1023
- Size of writing area : 0 to 1024

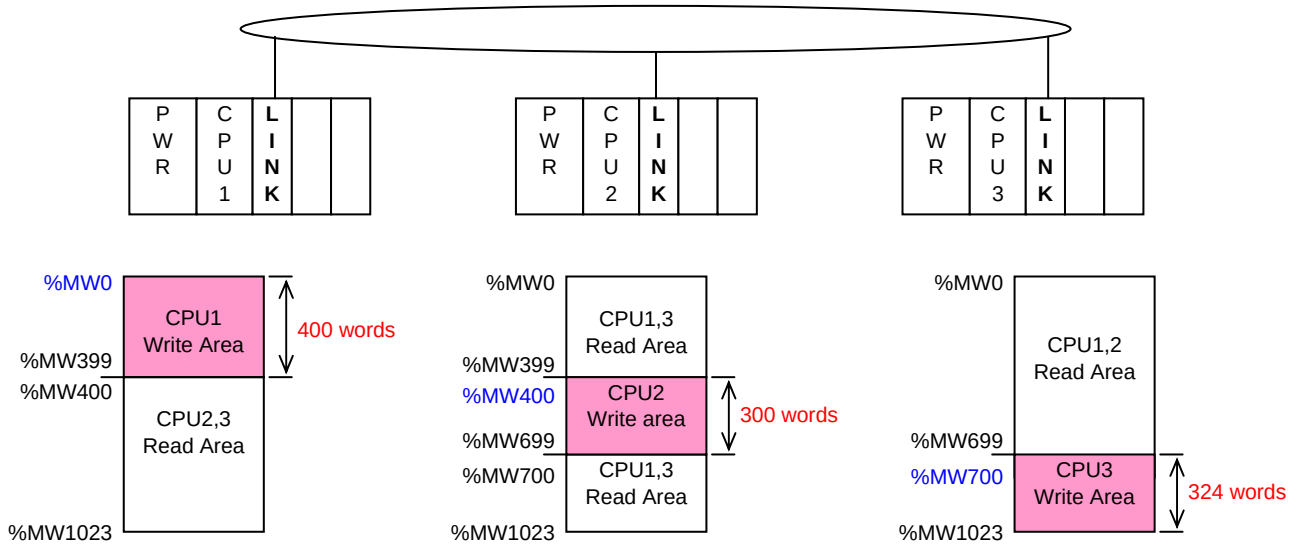
(Since the consistency between above two parameters cannot be checked by HX-CODESYS, invalid values, for example [offset address of writing area] is 1023 and [Size of writing area] is 1024, can be set. In that case, CPU detects an error and outputs error message in the CPU log.)

If PLC settings are set as follows, received link data might not be correct because I/O refresh timing and link refresh timing are not synchronized when CPU is in stop mode.

- Update IO while in stop : Disable
- Behaviour for outputs in Stop : Execute program

Configuration example

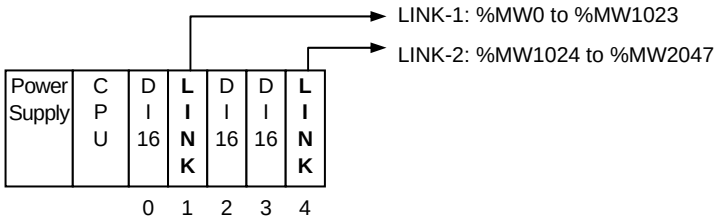
Below figure is an example of 3 CPUs. Each CPU has own writing area in the LINK area, which must not be overlapped. Writing area for a CPU is reading area for the other CPUs.



	CPU1	CPU2	CPU3
Writing area %MW-address (Write area %MW address)	0	400	700
Writing area size (Write area size)	400	300	324

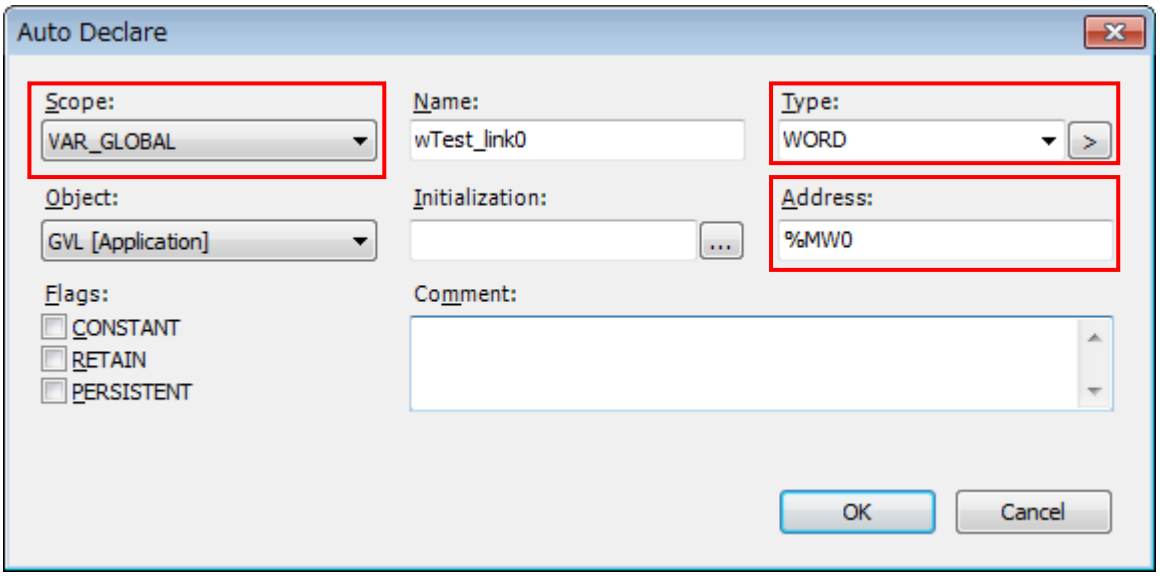
In one CPU Link module is used per CPU, the address of LINK area is from %MW0 to %MW1023. Addresses when several LINK modules used are shown below table. The address is NOT depending on mounted slot number but the number of LINK modules.

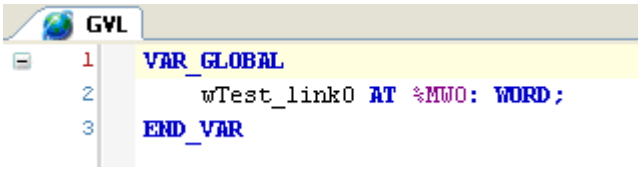
	LINK-1	LINK-2	LINK-3	LINK-4	LINK-5	LINK-6	LINK-7	LINK-8
From	%MW0	%MW1024	%MW2048	%MW3072	%MW4096	%MW5120	%MW6144	%MW7168
To	%MW1023	%MW2047	%MW3071	%MW4095	%MW5119	%MW6143	%MW7167	%MW8191



3.3.3 Declaration of Link Variable

If new variable is declared, [Auto Declare] dialog appears. For variables for LINK, put %MW address at address field and choose [VAR_GLOBAL] in scope field. Then this declaration will be added in GVL.





In POU, prefix "`GVL.`" is necessary for example "`GVL.wTest_link0`".

Note

If CPU Link module is not used, variables with %M addresses can be used as general registers.

Byte order

If EH/EHV-series, EHV+ series and HX series are used in one Link network, byte data is swapped because byte order and addressing rule are different in each series of CPU.

If 4-word data is stored in the link area as below mapping, BYTE, DWORD and LWORD data are seen as follows.

Link area mapping

Word address	Data
0	0x1234
1	0x5678
2	0xAABB
3	0xCCDD
4	…
5	…
…	…
1022	…
1023	…

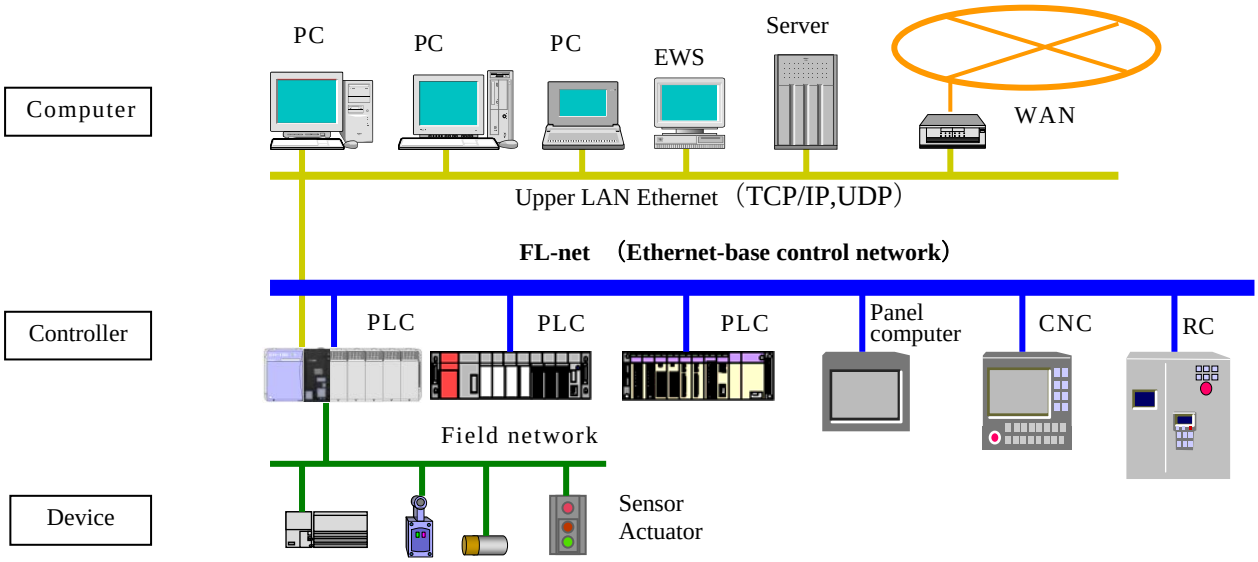
Data format

		EH/EHV			EHV+			HX		
		Address	Data		Address	Data		Address	Data	
			Binary	Dec.		Binary	Dec.		Binary	Dec.
X	bit 0 of word 0	L0	0	4	%MX1.0	0	4	%MX0.0	0	4
	bit 1 of word 0	L1	0		%MX1.1	0		%MX0.1	0	
	bit 2 of word 0	L2	1		%MX1.2	1		%MX0.2	1	
	bit 3 of word 0	L3	0		%MX1.3	0		%MX0.3	0	
	bit 4 of word 0	L4	1	3	%MX1.4	1	3	%MX0.4	1	3
	bit 5 of word 0	L5	1		%MX1.5	1		%MX0.5	1	
	bit 6 of word 0	L6	0		%MX1.6	0		%MX0.6	0	
	bit 7 of word 0	L7	0		%MX1.7	0		%MX0.7	0	
	bit 8 of word 0	L8	0	2	%MX0.0	0	2	%MX1.0	0	2
	bit 9 of word 0	L9	1		%MX0.1	1		%MX1.1	1	
	bit 10 of word 0	LA	0		%MX0.2	0		%MX1.2	0	
	bit 11 of word 0	LB	0		%MX0.3	0		%MX1.3	0	
	bit 12 of word 0	LC	1	1	%MX0.4	1	1	%MX1.4	1	1
	bit 13 of word 0	LD	0		%MX0.5	0		%MX1.5	0	
	bit 14 of word 0	LE	0		%MX0.6	0		%MX1.6	0	
	bit 15 of word 0	LF	0		%MX0.7	0		%MX1.7	0	
B	Low byte of word 0	-	-		%MB1	0x34		%MB0	0x34	
	High byte of word 0	-	-		%MB0	0x12		%MB1	0x12	
	Low byte of word 1	-	-		%MB3	0x78		%MB2	0x78	
	High byte of word 1	-	-		%MB2	0x56		%MB3	0x56	
W	Word 0	WL0	0x1234		%MW0	0x1234		%MW0	0x1234	
	Word 1	WL1	0x5678		%MW1	0x5678		%MW1	0x5678	
	Word 2	WL2	0xAABB		%MW2	0xAABB		%MW2	0xAABB	
	Word 3	WL3	0xCCDD		%MW3	0xCCDD		%MW3	0xCCDD	
D	1st DWORD	DL0	0x56781234		%MD0	0x12345678		%MD0	0x56781234	
	2nd DWORD	DL1	0xCCDDAABB		%MD1	0xAABBCCDD		%MD1	0xCCDDAABB	
L	1st LWORD	—	—		%ML0	0x12345678AABBCCDD		%ML0	0xCCDDAABB56781234	

3.4 FL-net Interface

3.4.1 Overview

FL-net is open FA network based on Ethernet using shared memory accessed by multi-vendors' CPUs. A virtual shared memory called common memory is accessed by each CPU in cyclic.



Common memory 1 and Common memory 2 are defined in FL-net. HX-CPU uses %M address for the common memory same as CPU Link. The address of the Common memory 1 is same as CPU Link module.

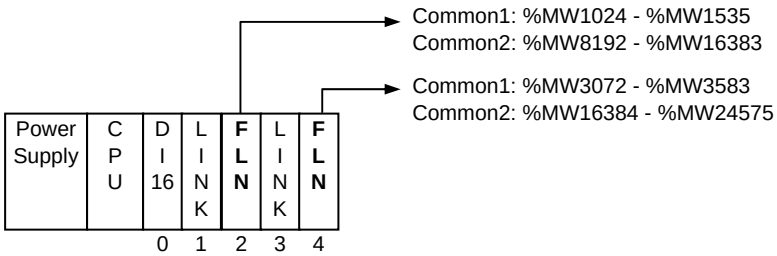
LINK No.	LINK-1	LINK-2	LINK-3	LINK-4	LINK-5	LINK-6	LINK-7	LINK-8
Start of common 1	%MW0	%MW1024	%MW2048	%MW3072	%MW4096	%MW5120	%MW6144	%MW7168
End of common 1	%MW511	%MW1535	%MW2559	%MW3583	%MW4607	%MW5631	%MW6655	%MW7679
Un-used addresses	%MW512 to %MW1023	%MW1536 to %MW2047	%MW2560 to %MW3071	%MW3584 to %MW4095	%MW4608 to %MW5119	%MW5632 to %MW6143	%MW6656 to %MW7167	%MW7680 to %MW8191

Since the size of common memory 1 is 512 words, only the 1st half of Link area is used for FL-net. The 2nd half cannot be used if FL-net module is used.

The address of the Common memory 2 is shown as below.

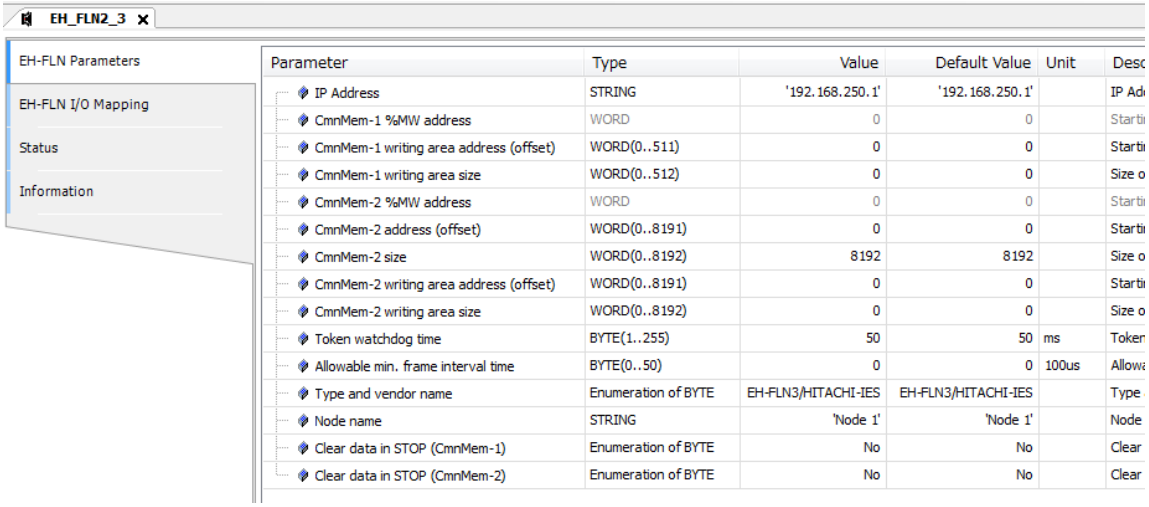
FL-net No.	FL-net 1	FL-net 2
Start of common 2	%MW8192	%MW16384
End of common 2	%MW16383	%MW24575

The size of common memory 2 is 8192 words.



3.4.2 FL-net Parameter Configuration

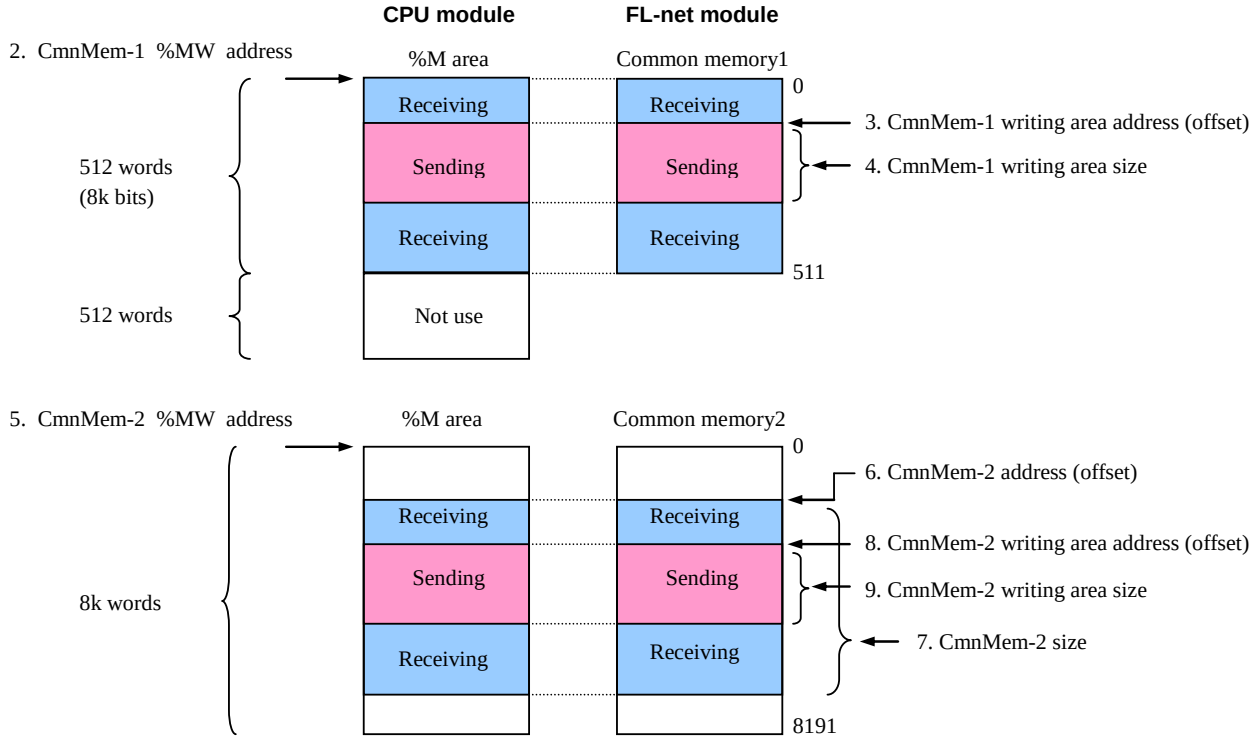
Add FL-net module to device tree in advance. The device name is EH-FLN2/3. Double click [EH-FLN2/3] and set parameters as follows.



No.	Name	Description	Default
1	IP address	IP address of EH-FLN2/3	192.168.250.1
2	CmnMem-1 %MW address	Starting address of common memory 1 is displayed in online mode.	0
3	CmnMem-1 writing area address (offset)	Set starting address (offset) of sending area of common memory 1.	0
4	CmnMem-1 writing area size	Set size of sending area of common memory1.	0
5	CmnMem-2 %MW address	Starting address of common memory 2 is displayed in online mode.	0
6	CmnMem-2 address (offset)	Set starting address (offset) of common memory 2.	0
7	CmnMem-2 size	Set size of common memory 2.	8192
8	CmnMem-2 writing area address (offset)	Set starting address (offset) of sending area of common memory 2.	0
9	CmnMem-2 writing area size	Set size of sending area of common memory 2.	0
10	Token watchdog time	Set monitoring time between token receiving and sending it out the next node.	50
11	Allowable min. frame interval time	Set waiting time between token receiving and sending out data frame to the next node.	0
12	Type and vendor name	Choose vendor name and model name. Functionality is same in both choices.	EH-FLN3/ HITACHIIES
13	Node Name	Set node name within 10 characters.	Node1
14	Clear data in STOP (CmnMem-1)	Set Yes if common memory 1 is to be cleared when CPU stops.	No
15	Clear data in STOP (CmnMem-2)	Set Yes if common memory 2 is to be cleared when CPU stops.	No

Configuration example

Configuration example is shown in below figure.



3.4.3 Cyclic Transfer

If right configuration parameters are downloaded to FL-net module together with application program, FL-net module will automatically take part in the network and start cyclic transmission. Refer to Section 3.4.5 for status of completing initialize.

Caution

If PLC settings in HX-CODESYS are configured as below and receiving area is written by the executed program in stop, the data may be overwritten by FL-net refresh cycle depending on the timing of CPU stop and FL-net refresh cycle.

- Update IO while in stop : Disable
- Behaviour for outputs in Stop : Execute program

3.4.4 Message Transmission

HX-CPU does not support sending of user message transmission for FL-net module.

Responding is limited in a part of commands from another node. If unsupported command is received, timeout is detected by the command sender.

No.	Message	Command	Response
1	Byte block read	NA	NA
2	Byte block write	NA	NA
3	Word block read	NA	NA
4	Word block write	NA	NA
5	Network parameter read	NA	X *1
6	Network parameter write	NA	NA
7	RUN / STOP direction	NA	NA
8	Profile read	NA	NA
9	Communication log data read	NA	X *1
10	Communication log data clear	NA	X *1
11	Return received message	NA	X *1
12	Pass through type message	NA	NA

*1 Response message is handled by FL-net module.

3.4.5 Status Monitor Library

The status information of EH-FLN2/3 can be read by special function block GetFLInfo.

Set FL-net number and xEnable to TRUE, then status is read out in the STRUCT FLInfo when xDone is TRUE.

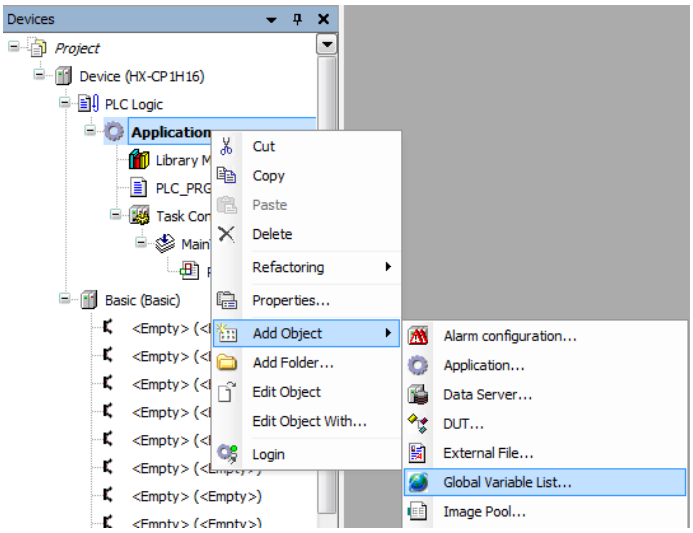
Detail of FLInfo structure

Member name	Description		Type	Remark
xInitDone	Node initialization	0: Not yet 1: Done	BOOL	-
xParamError	Parameter error	0: No error 1: Error	BOOL	-
xAdrDuplicated1	Address duplication area1	0: No error 1: Error	BOOL	-
xAdrDuplicated2	Address duplication area2	0: No error 1: Error	BOOL	-
xWaitforRcv	Wait for receiving status	0: Normal 1: Error (Wait for receiving)	BOOL	-
xTokenWatchdogTime	Timeout of token	0: No error 1: Error	BOOL	-
xNodeDuplicated	Node number duplication	0: No error 1: Error	BOOL	-
xTokenTimeoutOwnNode	Timeout of own node token	0: No error 1: Error	BOOL	-
xTBN_CBN_BSIZE	TBN, CBN or BSIZE error	0: No error 1: Error	BOOL	-
xCableDisconnect	Disconnect Cable	0: No error 1: Error	BOOL	-
xTokenModeUnmatch	Un-match token mode	0: No error 1: Error	BOOL	-
axLinkFlag	Link Node	0: Not join 1: Join	ARRAY[1..254] OF BOOL	-
axRunStatus	Status flag upper layer (Run Status)	0: STOP 1: RUN	ARRAY[1..254] OF BOOL	-
abErrStatus	Status flag upper layer (Error Status)	0: NORMAL 1: WARNING 2: ALARM	ARRAY[1..254] OF BYTE	-
sMACID	MAC address		STRING (12)	Valid FLN3 only
wRefCycleAllowed	Allowed refresh cycle time [ms]		WORD	-
wRefCycleCurrent	Current refresh cycle time (current value) [ms]		WORD	-
wRefCycleMax	Maximum refresh cycle time (maximum value) [ms]		WORD	-
wRefCycleMin	Minimum refresh cycle time (minimum value) [ms]		WORD	-
wMinFrameIntvl	Minimum frame interval [x 100us]		WORD	-
xEthernetStatus	Ethernet status flag	0: No link 1: Link-up	BOOL	Valid FLN3 only
xLinkSpeed	Link speed flag	0: 10Mbps 1: 100Mbps	BOOL	Valid FLN3 only
xDuplexMode	Duplex mode flag	0: Half duplex 1: Full duplex	BOOL	Valid FLN3 only
xSDRAMError	SDRAM error	0: No error 1: Error	BOOL	-
xEEPROMError	EEPROM error	0: No error 1: Error	BOOL	-
xSystemError	System error	0: No error 1: Error	BOOL	-
xFlashError	FLASH error	0: No error 1: Error	BOOL	-
xMPUError	MPU error	0: No error 1: Error	BOOL	-
xSystemRAMError	System RAM error	0: No error 1: Error	BOOL	-
xNodeNumberError	Node number error	0: No error 1: Error	BOOL	-
xLinkAddressError	Link address error	0: No error 1: Error	BOOL	-

3.5 Profibus Master

3.5.1 Overview

Profibus master module EH-RMP/EH-RMP2 is available with HX-CPU. Variables for Profibus master module are declared in Global Variable List (GVL) with %M address. GVL is added by right-clicking on [Application] shown below.



3.5.2 Configuration of Link Parameter

Add [EH-LNK] device to the device tree in advance. Double-click [EH-LNK] and set [Offset address of writing area] (Writing area %MW offset address) and [Size of writing area].

Offset address of writing area

Configure start address of Link area. Set 0 (zero) for EH-RMP/EH-RMP2.

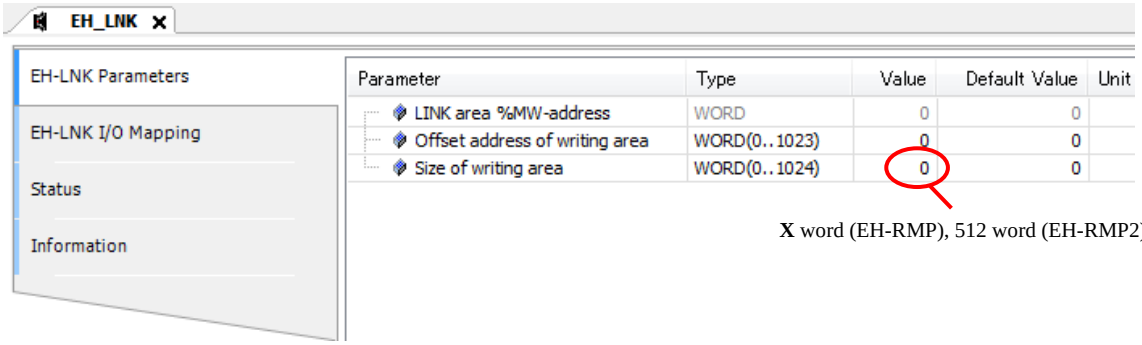
Size of writing area

EH-RMP: Set the actual size configured by Sycon.

EH-RMP2: Set 512 (as fixed value).

Caution

Do not set 0 (zero) for Size of writing area. Even if no output module is used and all slaves are input module only, configure dummy value except 0. If 0 is configured, STATUS LED blinks 4 times in EH-RMP. If other value than 512 is set in EH-RMP2, STATUS LED blinks 4 times.



X word (EH-RMP), 512 word (EH-RMP2)

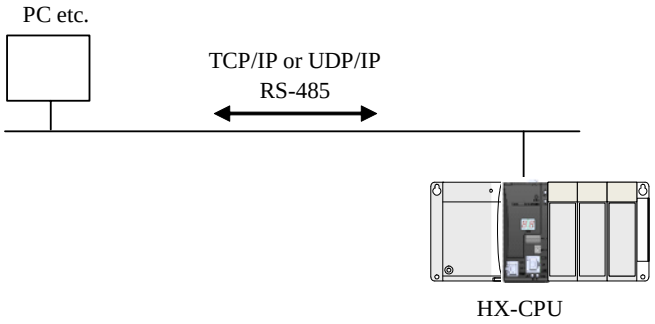
Address of reading area **(**Configuration is not needed.)

Read area is assigned automatically shown in below table.

Word address	Bit address	Hitachi address	Description (EH-RMP)	Description (EH-RMP2)
%MW0	%MX0.0-7,1.0-7	WL0	[**X**] word Write area	Write area （512word）
%MW1	%MX2.0-7,3.0-7	WL1		
%MW2	%MX4.0-7,5.0-7	WL2		
...				
%MW[**X**]		WL[X]		
...			Write area Possible to specify	
%MW255	%MX510.0-7, 511.0-7	WLFF		
%MW256	%MX512.0-7, 513.0-7	WL100	Not supported by EH-RMP (256 word)	
...				
%MW511	%MX1022.0-7, 1023.0-7	WL1FF		
%MW512	%MX1024.0-7, 1025.0-7	WL200	Read area	Read area （512word）
...				
%MW767	%MX1534.0-7, 1535.0-7	WL2FF		
%MW768	%MX1536.0-7, 1537.0-7	WL300	Not supported by EH-RMP (256word)	
...				
%MW1023	%MX2046.0-7, 2047.0-7	WL3FF		

3.6 General Purpose Communication

General purpose communication is available in Ethernet port and serial port of HX-CPU.



3.6.1 General Purpose Communication Over Ethernet

Several function blocks are available in NetBaseService library shown in below table.
Refer CAA_NetBaseService.pdf in NetBaseServices library for further information.

Table of NetBaseServices library

Protocol	Command	Description
TCP/IP	TCP_Server	TCP server set-up
	TCP_Connection	TCP server connection establish
	TCP_Client	TCP client set-up
	TCP_Write	Write sending data
	TCP_WriteBuffer	Write buffered sending data
	TCP_Read	Read receiving data
	TCP_ReadBuffer	Read buffered receiving data
UDP/IP	UDP_Peer	Peer set-up
	UDP_Send	Send UDP data
	UDP_SendBuffer	Send buffered UDP data
	UDP_Receive	Receive UDP data
	UDP_ReceiveBuffer	Receive buffered UDP data

Caution

In general, network byte order is big-endian, which is sending from higher byte, however, it is not always big-endian because it depends on data type such as WORD, DWORD, STRING. If necessary, use the following FUNCTION in SysSocket library to swap data.

- SysSockHtonl (Network byte order conversion from UDINT)
- SysSockHtons (Network byte order conversion from WORD)

Description example in ST language

```
test_out1 16#78563412 :=NBS.SysSocket.SysSockHtonl(test_in1 16#12345678 );
test_out2 16#3412 :=NBS.SysSocket.SysSockHtons(test_in2 16#1234 );
```

3.6.2 General Purpose Communication Over Serial

Function blocks are available in SysCom library shown in below table.
Refer to Application manual (Command reference) for further information.

Table of SysCom library

Commands	Function
SysComOpen	Serial port open
SysComOpen2	Serial port open / configuration
SysComClose	Serial port close
SysComSetSettings	Serial port configuration
SysComPurge	Serial port internal buffer clear
SysComRead	Receiving data (Read)
SysComWrite	Sending data (Write)

Caution

Below functions in SysCom library are not supported by HX-CPU.

- SysComGetSettings
- SysComSetTimeout

Sample program

A sample program for serial port is shown below. It is recommended to use ST language for communication programming. When 1 is set in the variable test, then string data [02 31 32 33 0D] (STX 123 CR) in the variable message is sent out from the serial port.

Variable declaration

```
PROGRAM PLC_PRG
VAR
        COM_sample:COM_Settings;
        COM_sampleEX:COM_SettingsEX;
        message:STRING:='123';
        Result:DWORD;
        Result1:DWORD;
        write_out:UDINT;
        test:INT;
        Status: INT;
        uchwyt: DWORD;
END_VAR
```

Program

```
COM_sample.sPort:=COM_Ports.SYS_COMPORT1;
COM_sample.byParity:=COM_Parity.SYS_NOPARITY;
COM_sample.byStopBits:=COM_Stopbits.SYS_ONESTOPBIT;
COM_sample.ulBaudrate:=COM_Baudrate.SYS_BR_19200;
COM_sample.ulBufferSize:=100;
COM_sample.ulTimeout:=10;
COM_sampleEX.byByteSize:=8;

CASE Status OF
0:
        uchwyt := SysComOpen(SYS_COMPORT1,ADR(Result));
        IF Result = 0 THEN
                Status := Status + 1;
        END_IF
1:
        Result1 := SysComSetSettings(uchwyt,ADR(COM_sample),ADR(COM_sampleEX));
        IF Result1 = 0 THEN
                Status := Status + 1;
        END_IF
2:
        IF test = 1 THEN
                message:=CONCAT('$02',message);
                message:=CONCAT(message,'$0d');
                write_out:=SysComWrite(uchwyt,ADR(message),LEN(message),1000,ADR(Result));
        test:=0;
        END_IF
END_CASE
```

MEMO

Chapter 4 Other Functions

4.1 OPC UA Server

HX-CPU supports OPC UA server function. OPC UA (Unified Architecture) is the newest specification of OPC based on the technology used for Web service and this is data exchanging opened standard between each softwares does not depend on vendors, programming language, operation systems or region. Adaptable scope of OPC UA is expanding not only PLC, SCADA and HMI but also MES or ERP positioned as upper layer.
Client application established by using OPC UA standardized interface, it will be possible to reuse user software system even for different controller vendors of several equipment without a lot of modification.

HX-CPU supports following functionalities as OPC UA server.

Table 4.1 OPC UA Server function

No.	Type	Support
1	supported profile	Micro Embedded Device Server
2	supported information model	PLCOpen Information Model

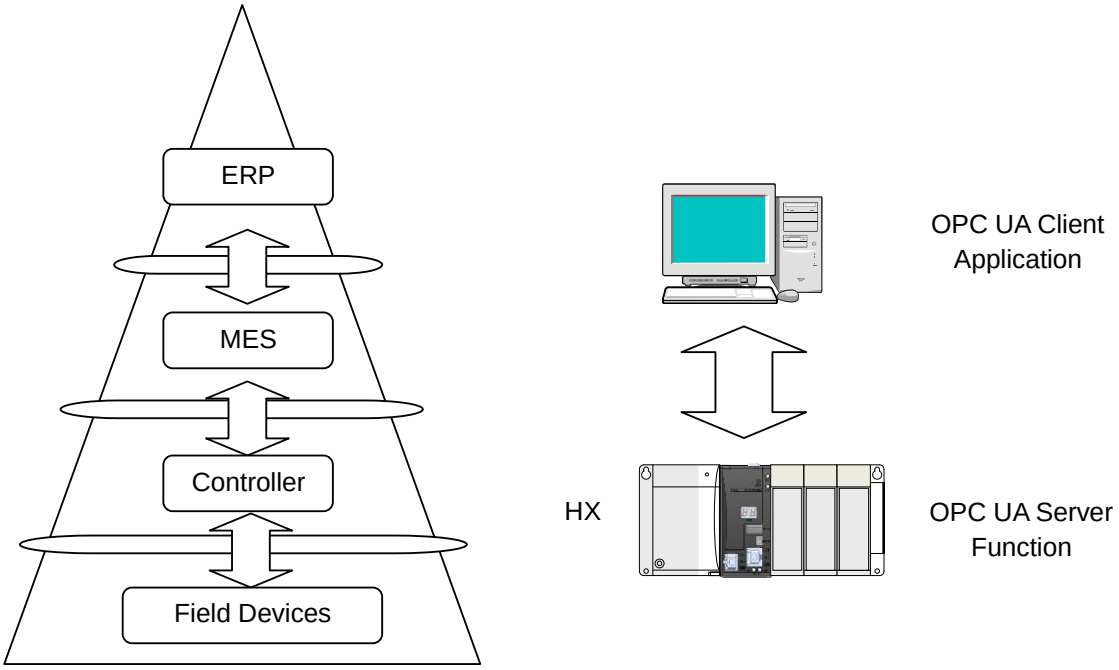


Figure 4.1 Information Models and OPC UA Server Functionality

Several technical documentation are available from OPC Foundation who is Spread promotion group and it can be possible to get them from the following address.
https://opcfoundation.org/.

(1) Configuration of HX-CPU side

Symbol configuration editor

Configure variable accessed by remotely from OPC UA client application using OPC UA server function on [Symbol Configuration] editor.

If [Symbol Configuration] is not on device tree, select [Add Object]-[Symbol Configuration] by clicking [Application].

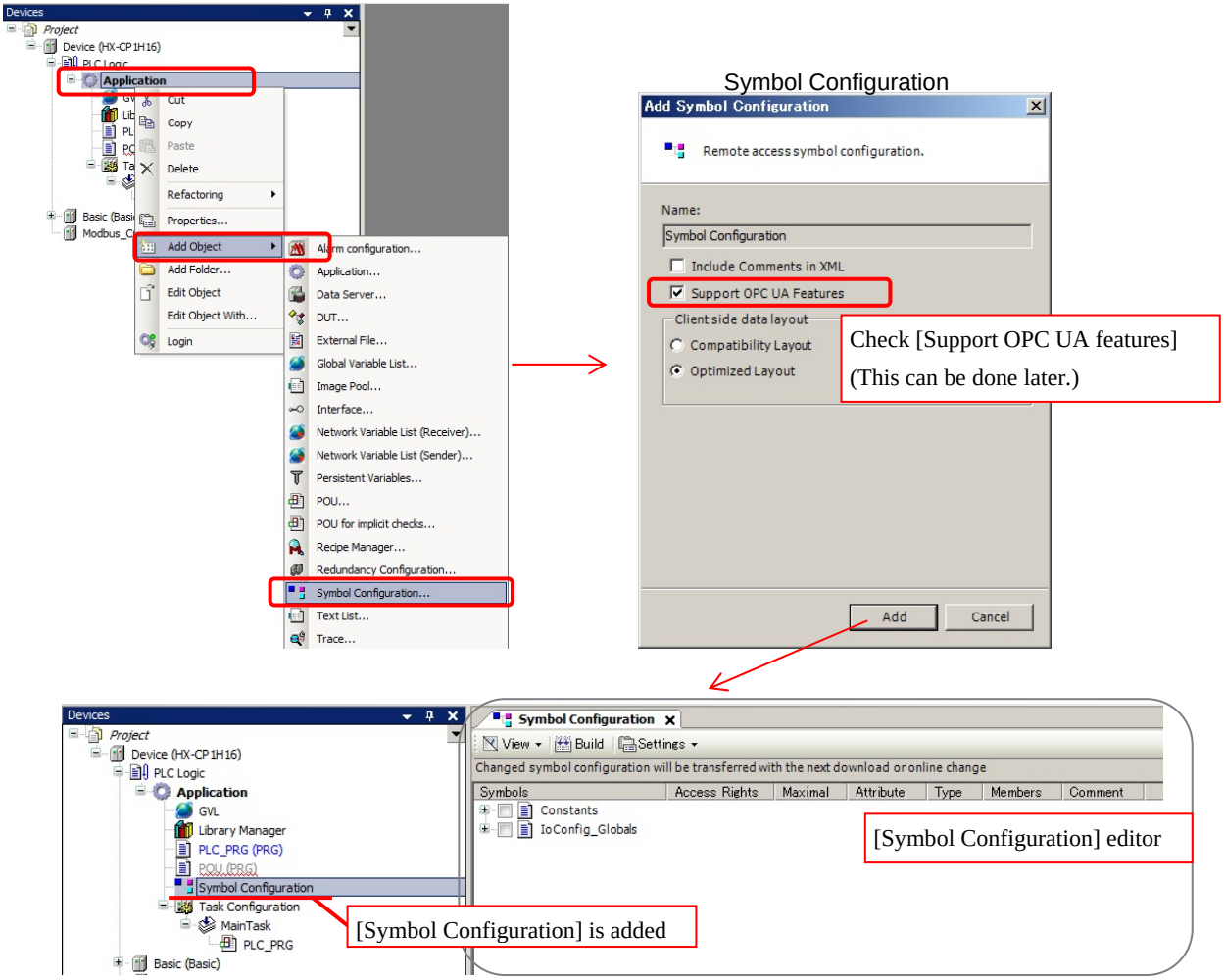


Figure 4.2 [Symbol Configuration] editor

Configuration of Remote accessing variable

List of variable included in Application are shown at [Symbol Configuration]. Specify variable can be accessed remotely.

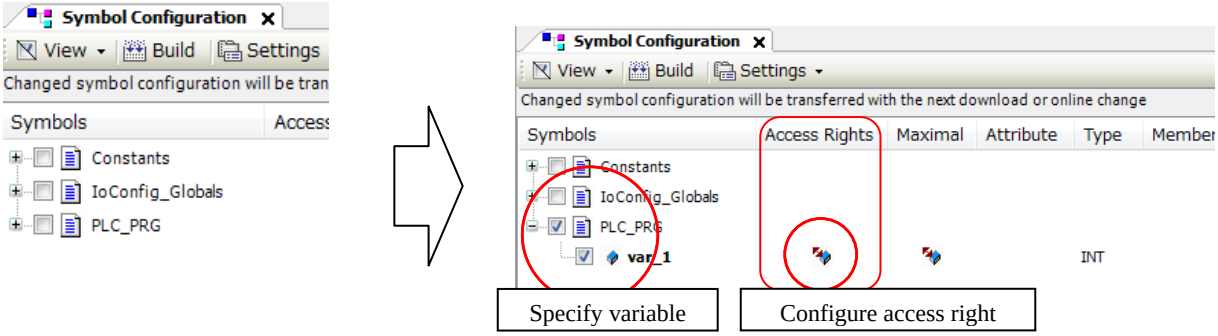


Figure 4.3 Specifying Remote access variable

It is possible to change access right of variable can be accessed remotely. Access right can be changed by each click action. Default setting is read & write.

 Read only  Write only*  Read & write

Figure 4.4 Access right of variable

*Write only is not supported

Enabling OPC UA server function

Check [Support OPC UA Features] by double clicking [Setting] tab of [Symbol Configuration].
(It is check status already when it configured during adding [Symbol Configuration].)

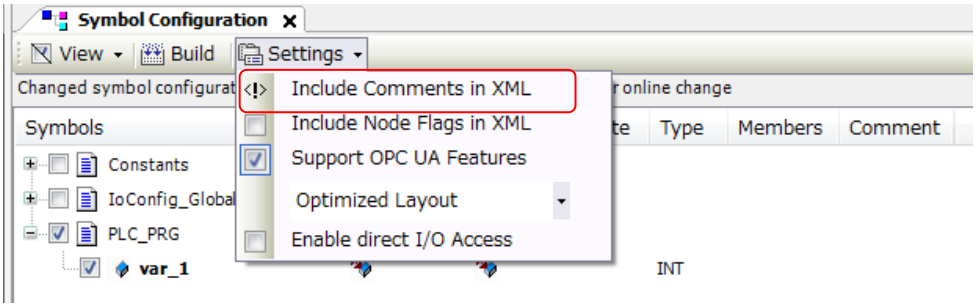


Figure 4.5 Enabling OPC UA server function

It will be possible to access specified variable from OPC UA Client Application by transferring the project to HX-CPU after above configuration and project build operation.

(2) Connecting from OPC UA

Following example shows connecting OPC UA server of HX-CPU from Client Application.
Regarding to the operation, follow client application specification.

Figure 4.6 Connecting OPC UA server

Caution

If HX-CPU is executed reset warm while OPC UA clients monitor data of HX-CPU, OPC UA clients need to re-configuration because monitoring data of HX-CPU is stop.

Make sure what is object of security protecting and take countermeasure for system configuration and operation mentioned security protection as an example by user responsibility.

- Usage of certification function and regular review for program and data should be protected.
- Usage of security function for devices used in network system.
- Connecting protection with unspecified target by usage of specifying function for connecting target.
- Operation management protection by making limitation of key lock of device setting place or user limitation.

4.2 FTP

4.2.1 FTP Server

File read or write access (upload / download) of SD card or USB memory mounted on HX-CPU from PC connected Ethernet, due to built-in FTP server function of HX-CPU. Active mode is necessary for FTP client.

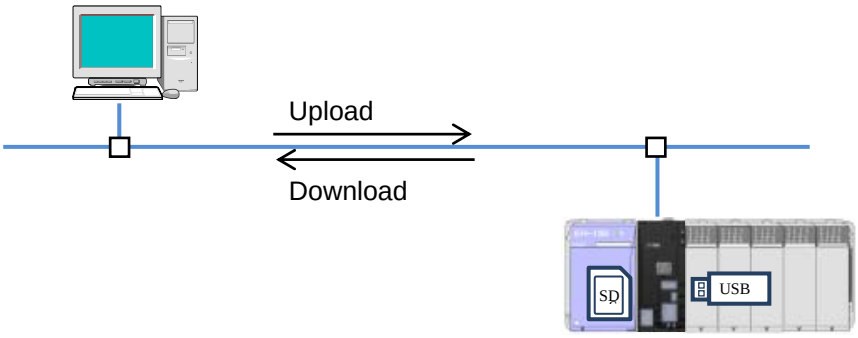


Figure 4.7 FTP server function

4.2.2 FTP Server Configuration

Configure several parameters related FTP server on [PLC Parameters] of [Device] Configuration window.

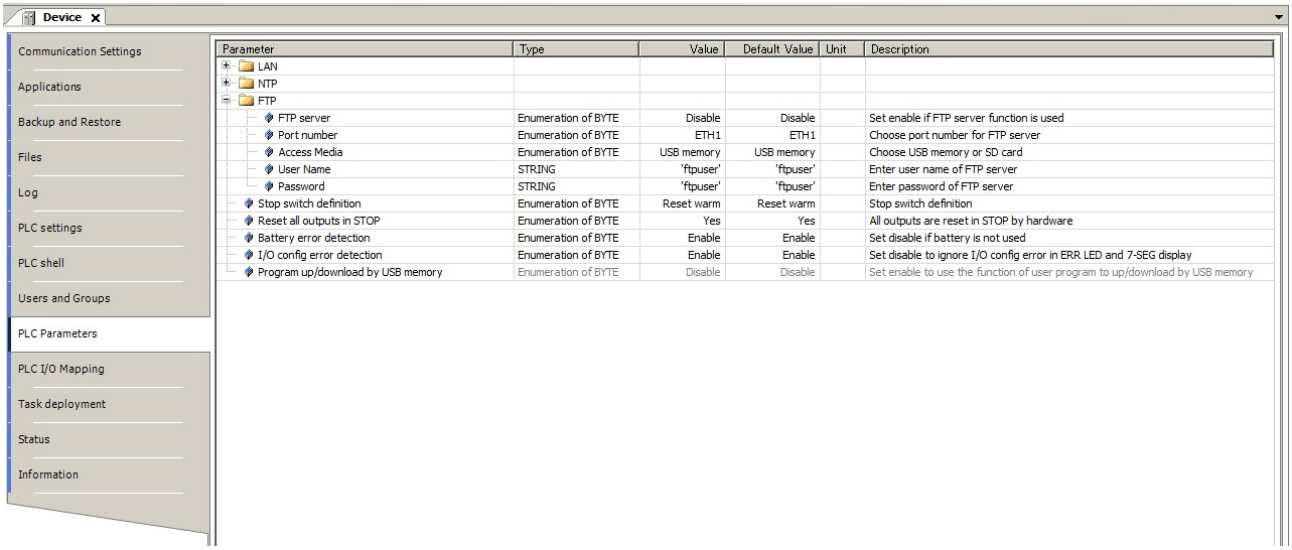


Figure 4.8 FTP server configuration

FTP server：Configure Enable when to use FTP.
Port number：Select communication port using for FTP. (ETH1/ETH2/ETH3)
Access Media：Select access target device. (USB memory/SD card)
If media is not mounted specified access target, login will not be accepted.
User Name：Configure user name for login. (Byte character alphabet or number less than equal 32 characters)
Small alpha character, number and _(under score bar) can be accepted, first character can't be number.
Password：Configure password for login. (Byte alpha character or number 4 to 32 characters)
Usable character is byte character or special character. However, " [¥ $ can't be used.
Number of connections：1 connection.
Timeout： HX-CPU logout from FTP server as timeout when 5 minutes has passed while HX-CPU is in an idle state during login.

4.2.3 List of FTP Commands

Usable command list is shown in below.

Table 4.2 Usable command list

Command	Function
ascii	Set file transfer mode to ASCII.
binary	Set file transfer mode to binary.
bye	Disconnect connection between server and exit client.
cd	Change working directory of server.
close	Disconnect connection between server.
delete	Delete specified file of server.
dir	Show detail list of server files and directories.
get	Transfer file of server into local host. [download]
lcd	Change local working directory.
ls	Show list of server files and directory.
mdelete	Delete multiple files of server.
mdir	Transfer detail of multiple files and directories into local file.
mget	Transfer multiple files of server into local host. [download]
mkdir	Make working directory onto server.
mls	Transfer several files in the several files and directory into local file.
mput	Transfer specified multiple local file into server. [upload]
open	Connect specified server.
prompt	Switch interactive mode. Toggling mode every sending command.
put	Transfer specified local file into server. [upload]
pwd	Display current working directory of server.
quit	(same as bye)
rename	Rename file name of server.
rmdir	Delete working directory of server.
type	Display current file transfer mode.

4.2.4 FTP Command Detail

【ascii】

Format: ftp> ascii
Function: Set file transfer mode to ASCII.

【binary】

Format: ftp> binary
Function: Set file transfer mode to Binary.

【bye】

Format: ftp> bye
Function: Exit ftp.

【cd】

Format: ftp> cd [directory]
Function: Change working directory to specified directory by [directory].
It is not possible upper directory from logged in directory.

【close】

Format: ftp> close
Function: Disconnect connection between FTP server.

【delete】

Format: ftp> delete [file]
Function: Delete specified file.

【dir】

Format: ftp> dir (([directory]) (local file))
Function: Display detail list of server file and directory.
Save this list into file by specified [(local file)].

【get】

Format: ftp> get [file] ([local file])
Function: Transfer file of server to local. [download]
It is possible to specify transferring local file name.

【lcd】

Format: ftp> lcd [local directory]
Function: Change local working directory.

【ls】

Format: ftp> ls
Function: List all file in current directory.

【mdelete】

Format: ftp> mdelete [file 1] ([file 2] …)
Function: Delete multiple files of server.
If interactive mode set off by prompt command, all specified files can be deleted without confirmation.

【mdir】

Format: ftp> mdir [file 1] ([file 2] …) [local file]

Function: Transfer multiple files and directories to local file.

【mget】

Format: ftp> mget [file 1] ([file 2] …)

Function: Transfer multiple files of server to local host.
If interactive mode set off by prompt command, all specified files are transferred without confirmation.

【mkdir】

Format: ftp> mkdir [directory]

Function: Make directory on server.

【mls】

Format: ftp> mls [file 1] ([file 2] …) [local file]

Function: Transfer multiple files and directory list to local file.

【mput】

Format: ftp> mput [file 1] ([file 2] …)

Function: Transfer specified files to server.
If interactive mode set off by prompt command, all specified files are transferred without confirmation.

【open】

Format: ftp> open [host]

Function: Connect server specified IP address or host name.

【prompt】

Format: ftp> prompt

Function: Change interactive mode. Toggling mode every sending command.

【put】

Format: ftp> put [local file] ([server file])

Function: Transfer specified file to server.
If server file is specified, transfer file with specified file name.

【quit】

Format: ftp> quit

Function: Exit ftp.

【rename】

Format: ftp> rename [file] [new file]

Function: Change file name of server.

【rmdir】

Format: ftp> rmdir [directory]

Function: Delete directory of server.

【type】

Format: ftp> type [type]

Function: Display current file transfer mode. It is possible to change file transfer mode by specifying.

4.2.5 Exclusive Control of File Access

Exclusive control by POU is necessary if conflicting file access is occurred between FTP server function and POU. In that case, control the file exclusively with indicating file access status by POU.

Table 4.3 Exclusive file access control for each file

Exclusive file access control for each file		File access by POU	
		Read	Write
FTP command	Read	Not necessary	Necessary
	Write	Necessary	Necessary

Make sure what is object of security protecting and take countermeasure for system configuration and operation mentioned security protection as an example by user responsibility.

- Usage of certification function and regular review for program and data should be protected.
- Usage of security function for devices used in network system.
- Connecting protection with unspecified target by usage of specifying function for connecting target.
- Operation management protection by making limitation of key lock of device setting place or user limitation.

4.3 Visualization

Visualization function supporting program debugging and monitoring by using visual displaying on Integrated Development Environment HX-CODESYS and Web visualization function (available with HX-CP1H16) by using general Web browser are available on HX-CPU.

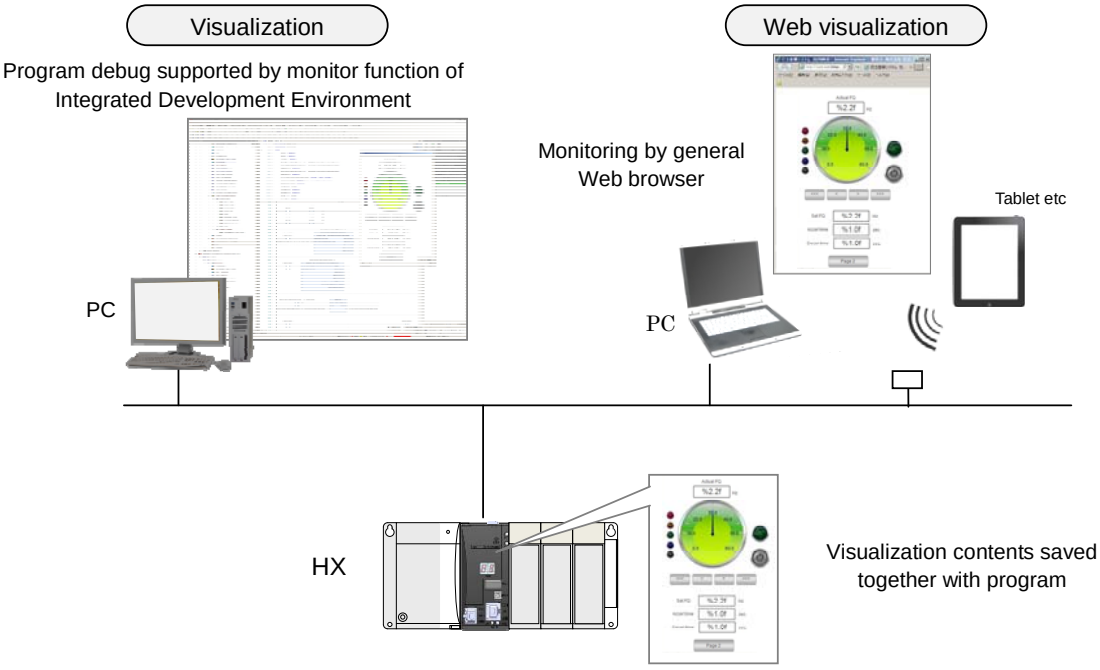


Figure 4.9 Visualization function

Create visualization object

It is needed to add visualization object onto application to enable visualization.

Select [Application]-[Add Object]-[Visualization]. Related visualization object are added device tree.

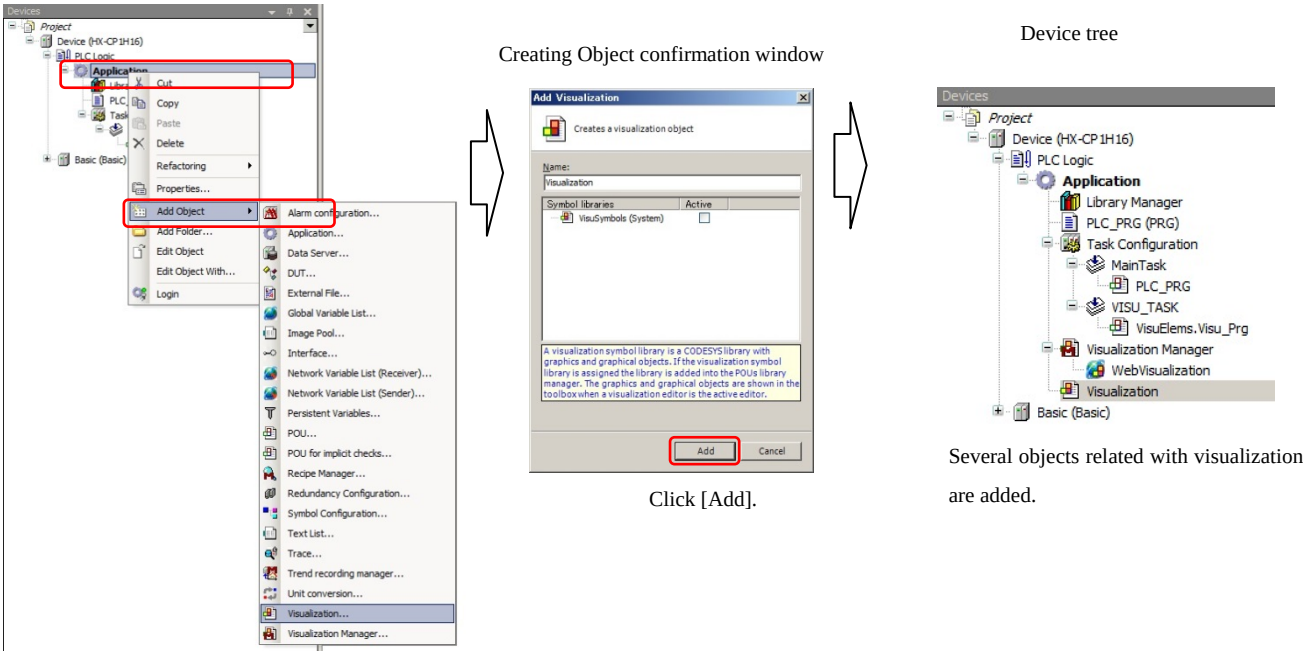


Figure 4.10 Adding Visualization object

Visualization editor

Double click [Visualization] on Device tree. Visualization editor appears.

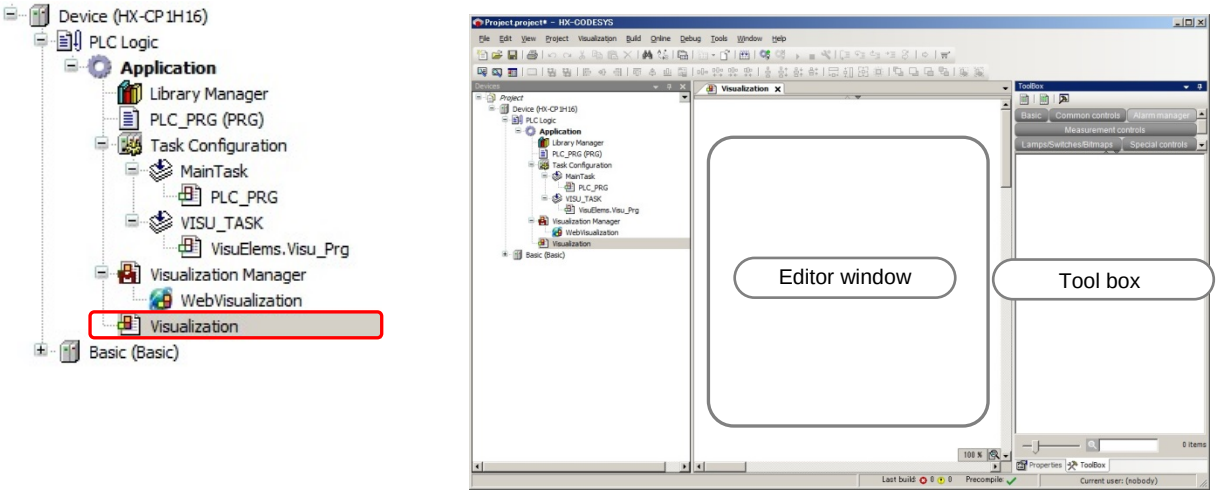


Figure 4.11 Visualization editor

Usage of elements

There are several parts (this after [elements]) available to create graphical display in Tool box. Select element to use from the tool box and drag it into the editor window. The element is placed on the editor window and the property window appears.

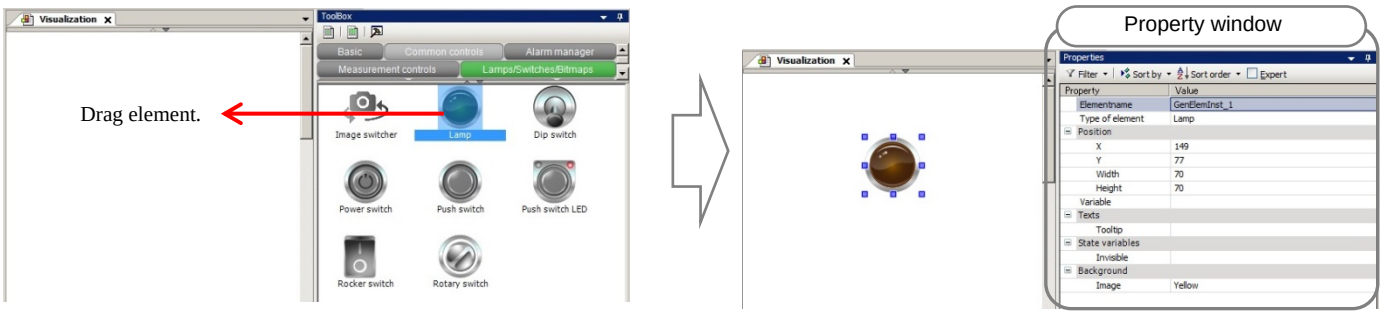


Figure 4.12 Usage of elements

Variable Assignment

Assign variable for element by specifying on “Variable” field of “Property”.

It is also possible to assign variable using in application by Input Assistant.

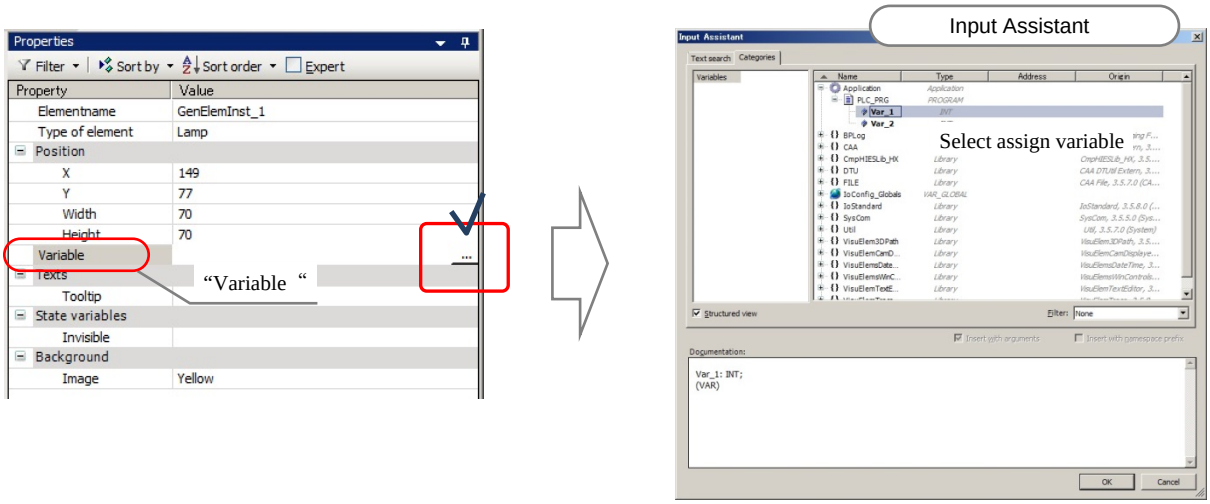


Figure 4.13 Assignment variable for element

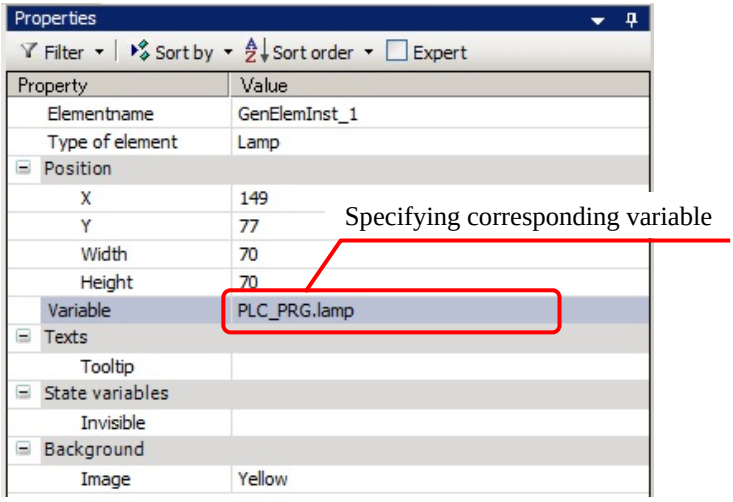
Commonly used elements

Display BOOL type variable (Read)

Display example

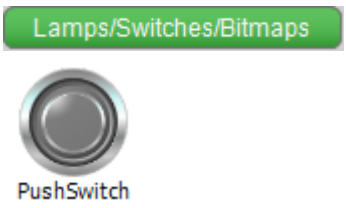
In case of [FALSE]

In case of [TRUE]



- It is possible to specify tooltip when mouse indicates close object by specifying [Texts]-[Tooltip] of Property. (This is same as other elements.)
- It is possible to set Display / Non-display by specifying [State variables]-[Invisible] of Property. (This is same as other elements.)

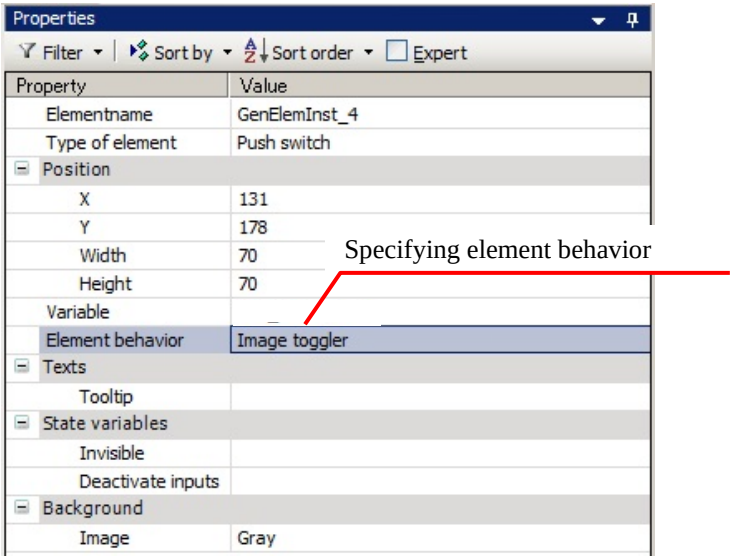
Switching BOOL type variable (Write)



Display example

In case of [FALSE]

In case of [TRUE]



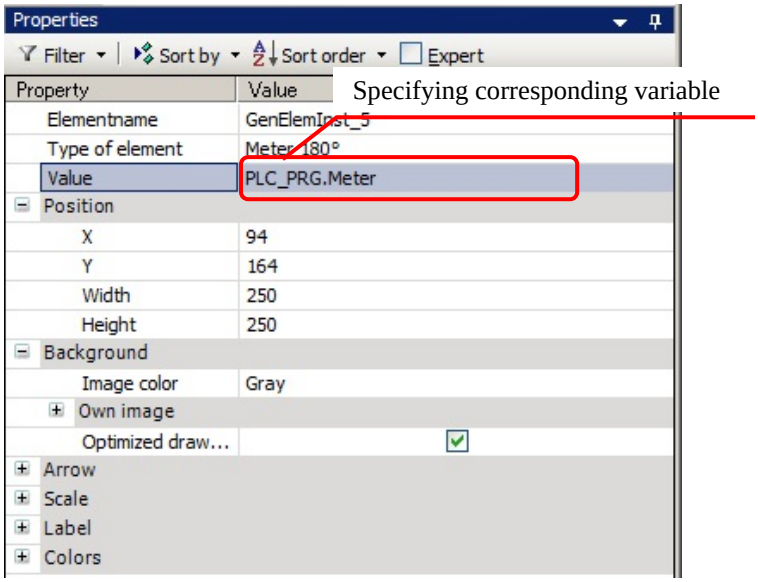
- It is possible to specify element behavior at [Element behavior] of Property.
  Image toggler: Toggle switches behavior. It turns ON by click once and Turns OFF by one more click.
  Image tapper: Push switches behavior. It turns ON during click only.
- It is also possible to be unable to operate by specified variable value by setting [State variables]-[Deactivate inputs] of Property.

Refer online help for further detail of each element specifications.

Graphical display of Variable value

Meter180

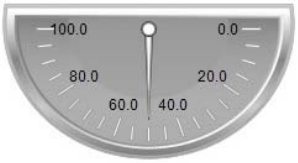
Display example



Specifying corresponding variable

Changing example by Property specifying

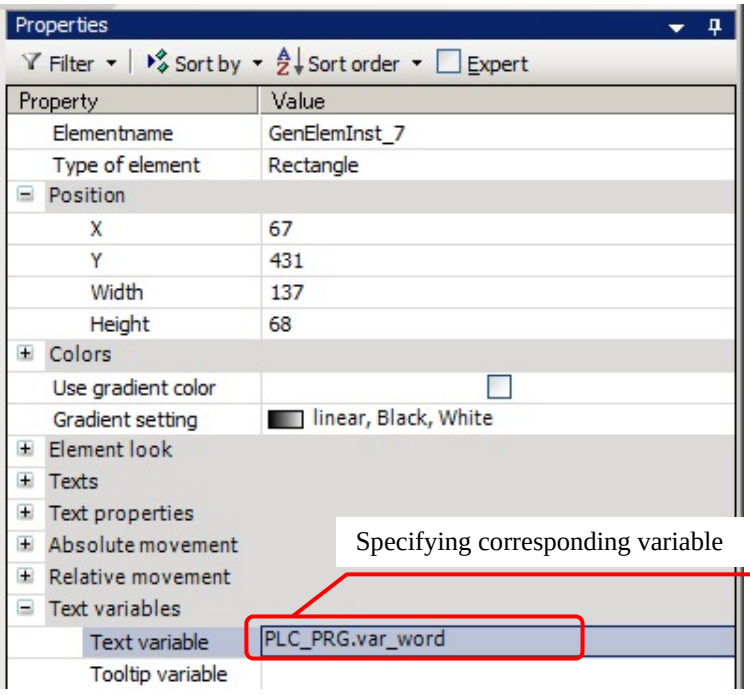
Direction



Color, meter indicator, scale of value, display position etc

Display variable value by text (string)

Rectangle



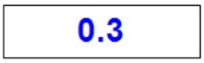
Specifying corresponding variable

%d

Specify type of display	
Signed decimal display	%d
Hexadecimal display	%x
Unsigned decimal display	%u
String display	%s
Floating display	%f

*Changing number of digits is possible

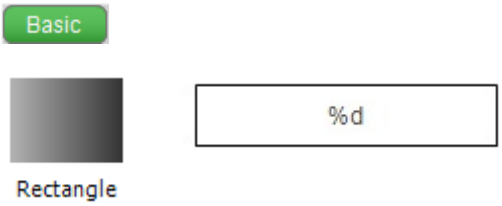
%.1f



%.3f

Refer online help for further detail of each element specifications.

Input variable value by text (string)

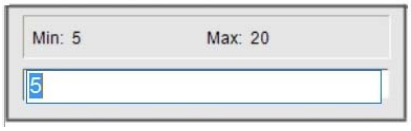


Display example

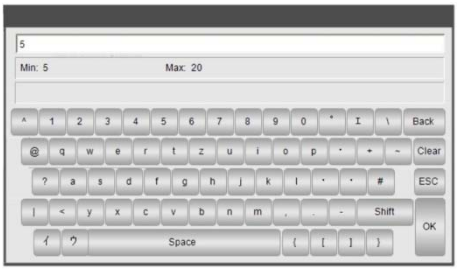
Input types : Text input

In case of PasswordField checked

Input types : Text input with limits

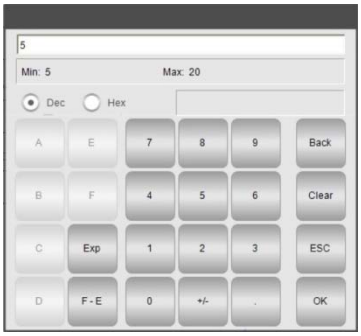


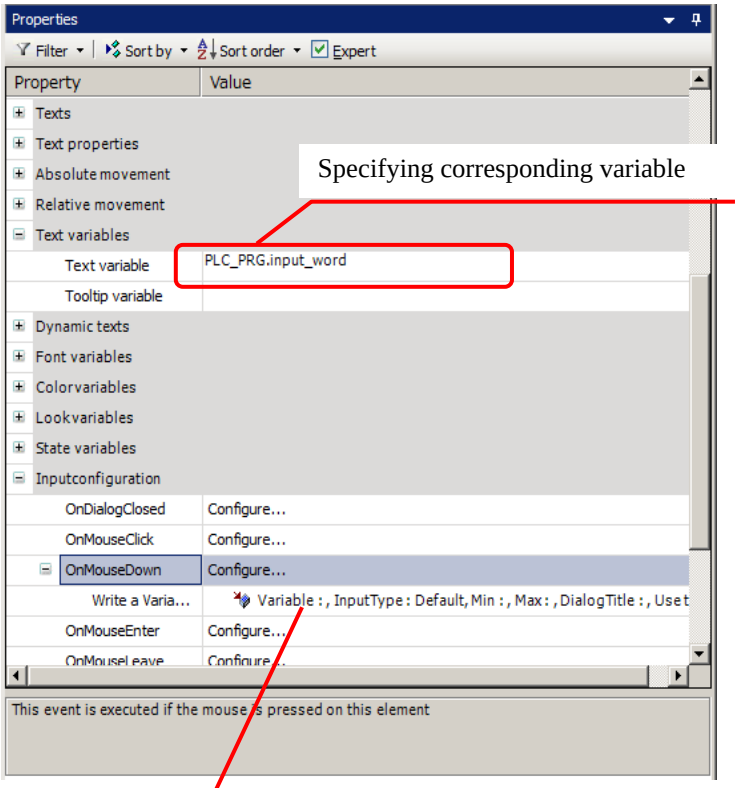
Input types : VisuDialogs Keypad



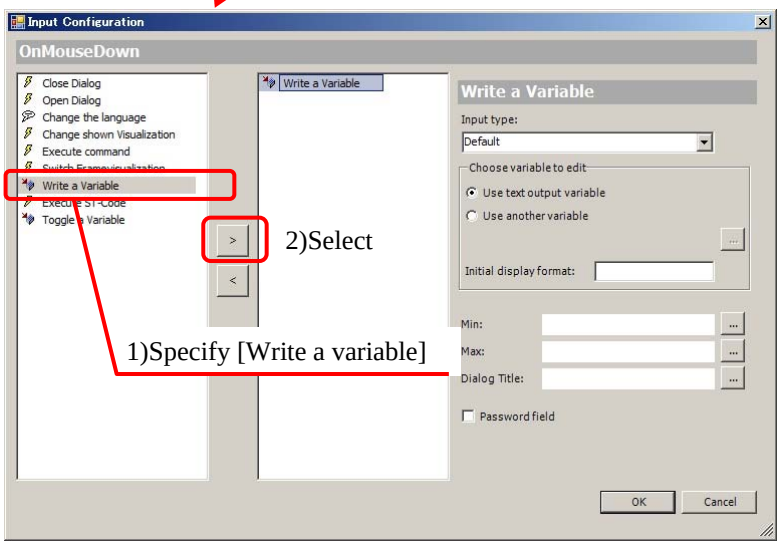
Input types : VisuDialogs Numpad

Input types : VisuDialogs NumpadExtended





Specify input structure by clicking



Input types	Default(Text input) Text input Text input with limits VisuDialogs Keypad VisuDialogs Numpad VisuDialogs NumpadExtended
Choose variable to edit	Use text output variable Pass input value to specified variable at this element. Use another variable Pass input value to other variable. Initial display format Use comment input of Min/Max for VisuDialogs.
Min	Specify minimum value of input.
Max	Specify maximum value of input.
Passwordfield	Display [*(asterisk)] during input.

Refer online help for further detail of each element specifications.

Task configuration of Visualization

"VISU_TASK" is added automatically by adding Visualization object.

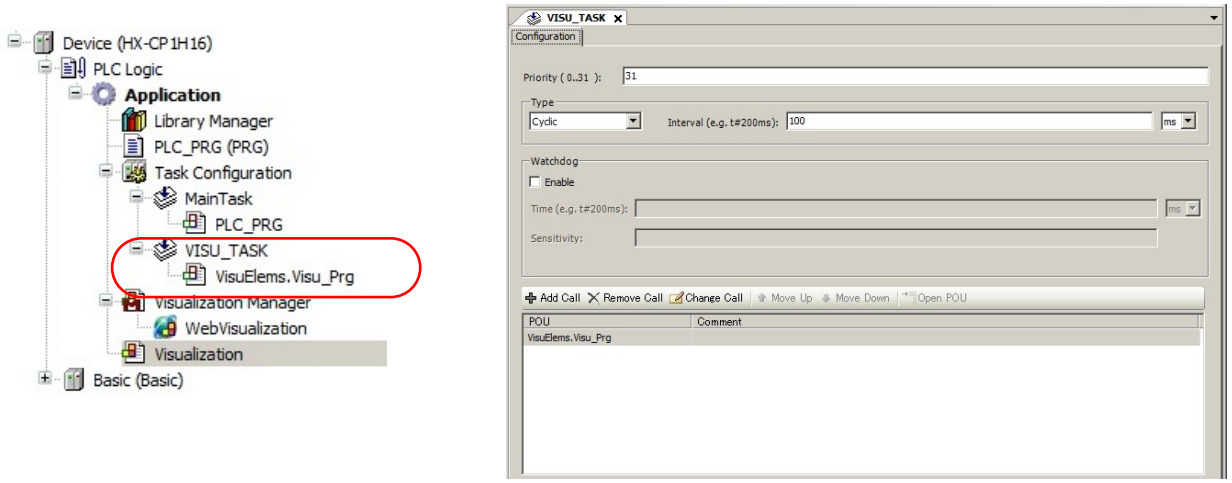


Figure 4.14 Task configuration of Visualization

Priority (0..31)

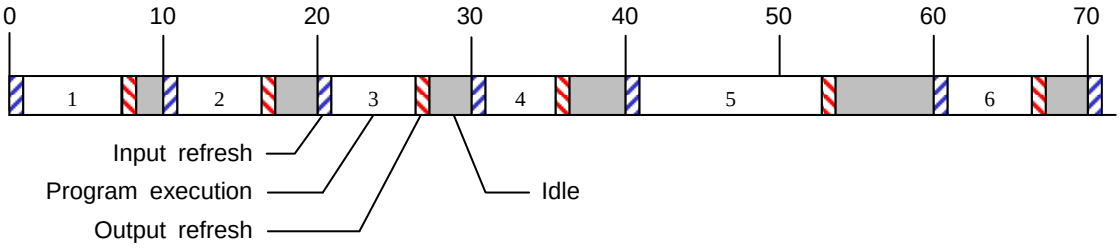
0 is the highest priority, 31 is the lowest.

Type

Choose type of the task. Inputs are read at the beginning of each task, and outputs are written at the end of the program execution.

Cyclic task

The task will be processed cyclic according to the time definition given in the field [Interval]. If actual execution time exceeds the cycle time, the next cycle does not start immediately but wait until the next fixed cycle time. For example, cycle time is set as 10 ms and 5th scan starting from 40 ms takes 12 ms, then 6th cycle starts at 60 ms as below figure.



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 3 ms of sleeping time is added at the end of each cycle for other tasks to be executed properly.

Status task

The task will be started when selected variable is TRUE.

Watchdog

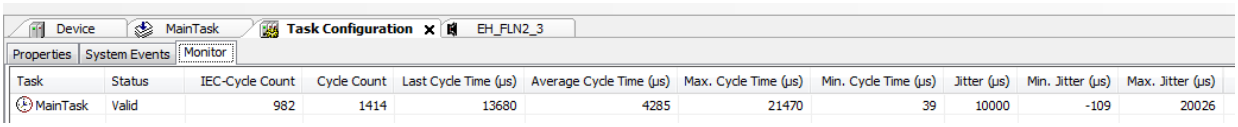
When it is enabled, watchdog function is activated. If program execution time exceeds watchdog time, CPU stops program execution with [24] error code displayed at 7 segment LED.

There are two different conditions to detect watchdog error as follows.

Example: Time: #5 ms, Sensitivity: 3.

- Detect condition 1 : one cycle exceeds 15 ms
- Detect condition 2 : scan cycle exceeds 5 ms in 3 times consecutive

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



Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (µs)	Average Cycle Time (µs)	Max. Cycle Time (µs)	Min. Cycle Time (µs)	Jitter (µs)	Min. Jitter (µs)	Max. Jitter (µs)
MainTask	Valid	982	1414	13680	4285	21470	39	10000	-109	20026

Visualization manager

This is common configuration items for Visualization and Web Visualization.

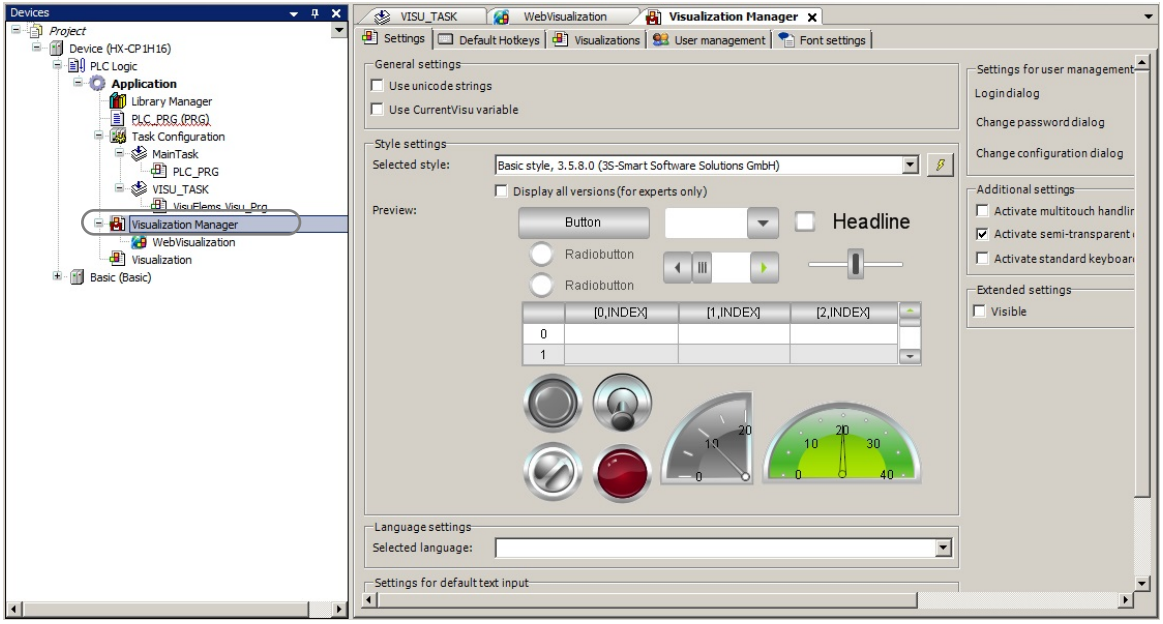
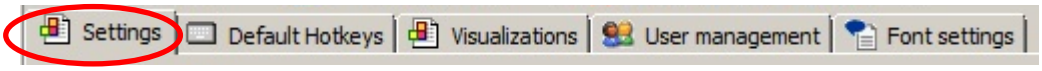
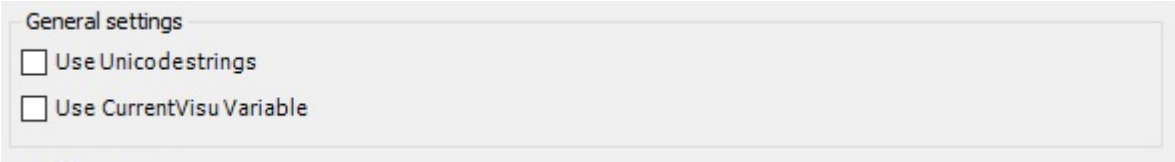


Figure 4.15 Configuration of Visualization manager

Configuration

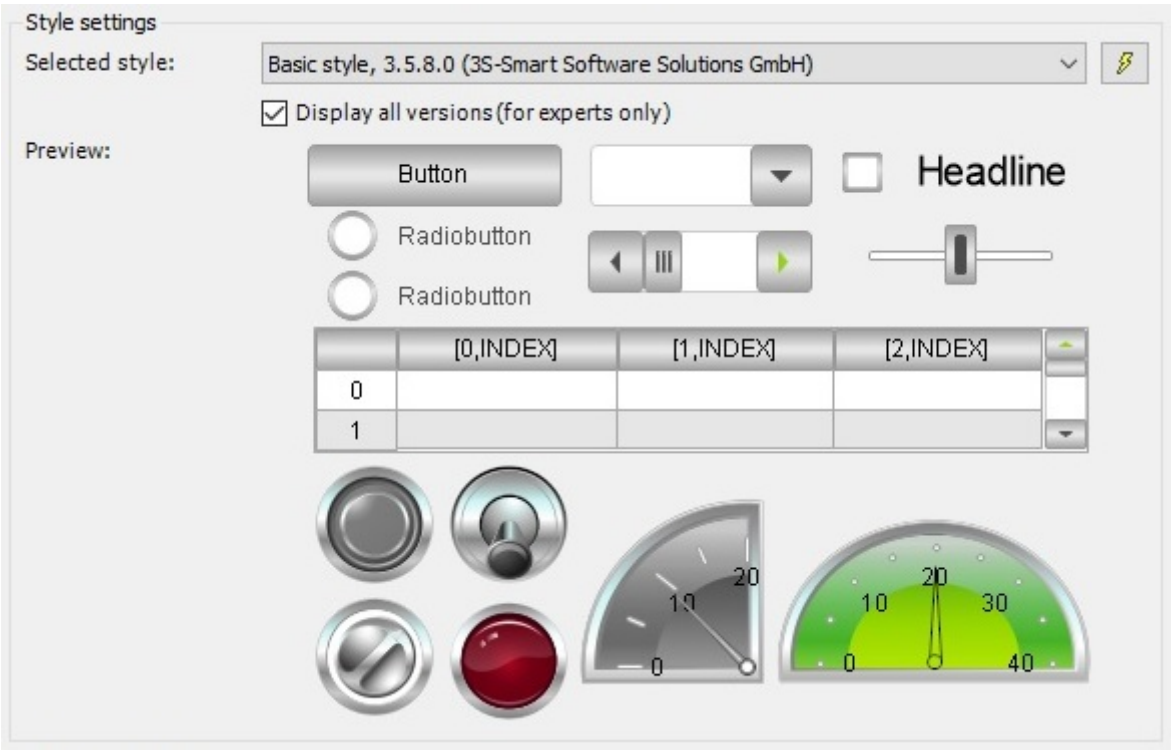


General setting



Use Unicodestrings	All string is executed by Unicode format used in Visualization by this option. If Japanese character is displayed for Web visualization, please check this.
User CurrentVisu Variable	Pass name of current display visualization to Global string variable [CurrentVisu].

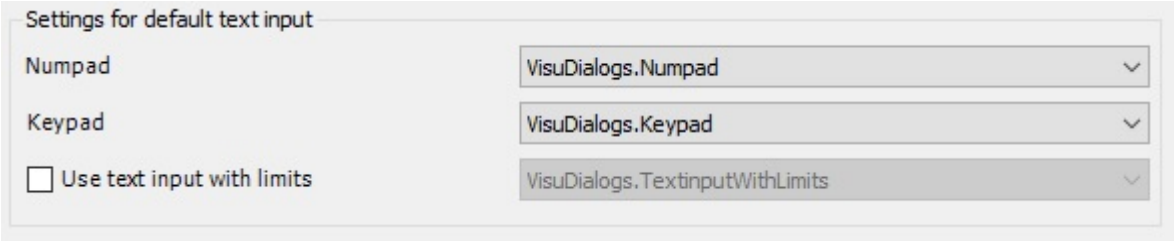
Style settings：(Use this default value without change)



Language settings

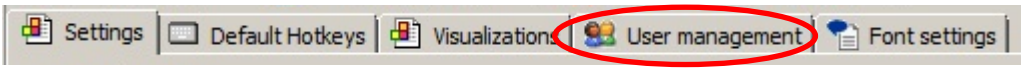
Configured language will be used at the time of start.

Setting for default test input



Use text input with limits	Default of text input is dialog with the minimum and maximum value range.

User Management of Visualization



Configure access limitation for security protection is possible for Visualization.
Refer [4.7 Support function of security protection] for additional information.

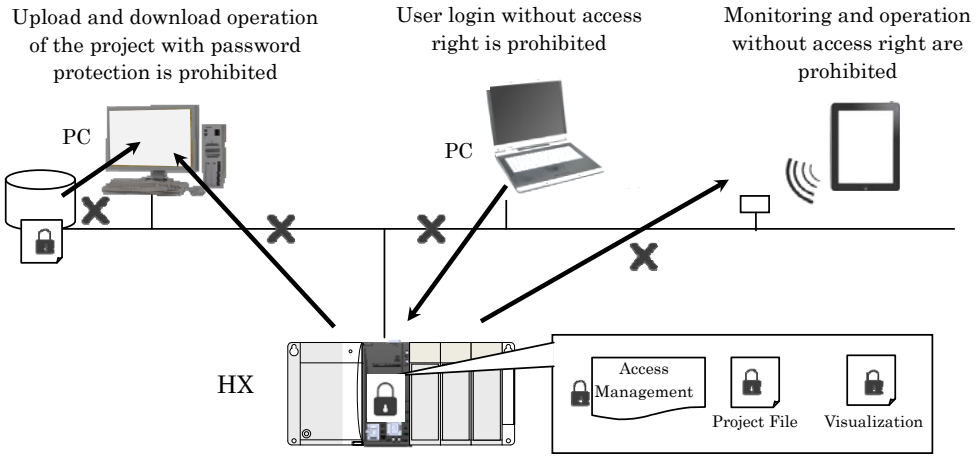
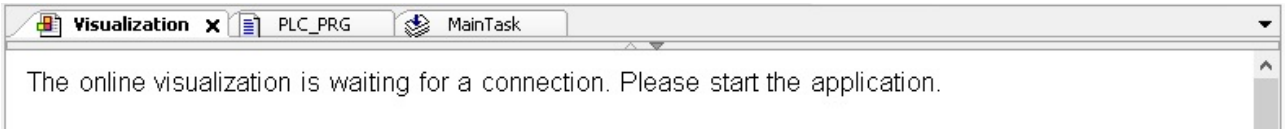


Figure 4.16 Support function of security protection

Start of Visualization

Visualization can start by transferring built project HX-CPU after configuration of visualization is completed.
Following display appears during HX-CPU is stop.



Visualization starts automatically when status of HX-CPU is set to RUN.

Web Visualization (HX-CP1H16 only)

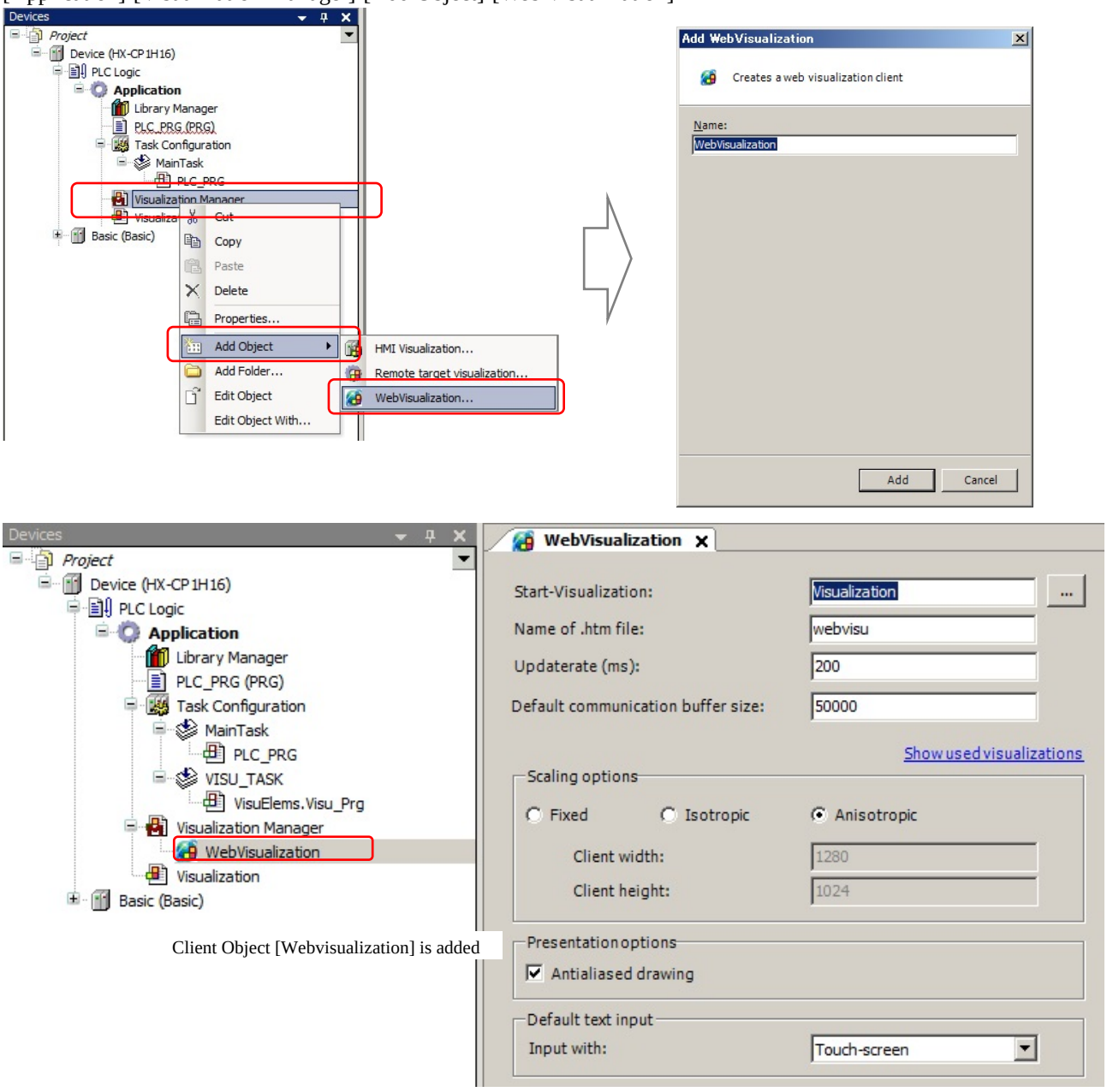
It is possible to access Visualization stored in HX-CPU from general Web browser of PC or tablet. This function is called Web Visualization.
Use Web browser following HTML5 . (Example: Firefox, Safari, Chrome, IE9 or later, etc)
Web Visualization can be used during HX-CPU is in RUN status.

Caution

Web Visualization of HX-CP1S08 operates. However this is only demonstration purpose. Therefore please do not use Web Visualization of HX-CP1S08. Display is limited within 30 minutes.

Add client object of Web Visualization under [Visualization Manager] on Device tree.

[Application]-[Visualization Manager]-[Add Object]-[Web Visualization]



Client Object [Webvisualization] is added

Figure 4.17 Web Visualization Object

If Webvisualization Object configuration display is not displayed on right side, it will appears by double clicking “Webvisualization” object.

Configuration items of Webvisualization object

Start Visualization	Input Visualization name when starts Visualization at the time of start automatically.
Name of .htm file:	Specify HTML file name of Visualization. Usable character for file name is byte alpha or number. File name including Japanese character can't be monitored by browser.
Updaterate (ms)	Specify refresh cycle in ms units for Web browser. (Default is 200 ms)
Default communication buffer size	Specify communication buffer size between browser.

Scaling options

Fixed / Isotropic / Anisotropic	Fixed: Display original size when it is made. Isotropic: Ratio of image is kept and displayed. Anisotropic: Display according to Browser display size after adjusted automatically.
Client width	Specify window width size of browser (Pixel)
Client height	Specify window height size of browser (Pixel)

Presentation options

Antialiased drawing	Check when it needed to display drawing smoothly.

Default text input

Touch-screen	Select when Touch screen is used for Web client.
Keyboard	Select when Keyboard is used for Web client.

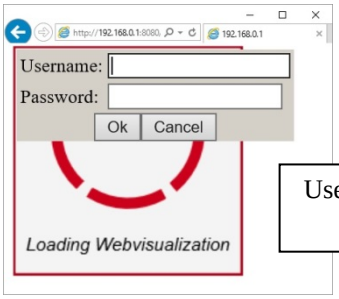
Access method from Web browser

Describe URL following below description to access downloaded Visualization of HX-CPU from Web browser.
http://[IP address]:[8080]/[Web file name]　　Example：http://192.168.0.1:8080/webvisu.htm

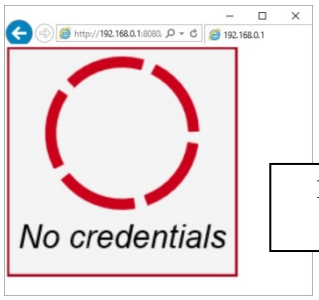
Following display appears after completing access to HX-CPU, and Display of Visualization.

This display appears when HX-CPU is stopping or Visualization is under preparation. Specified Visualization appears after HX-CPU is in running and preparation is completed.

User name and password are required when online user is registered HX-CPU.
Regarding to user name and password, please contact management person of your HX-CPU.



Username and Password are required.

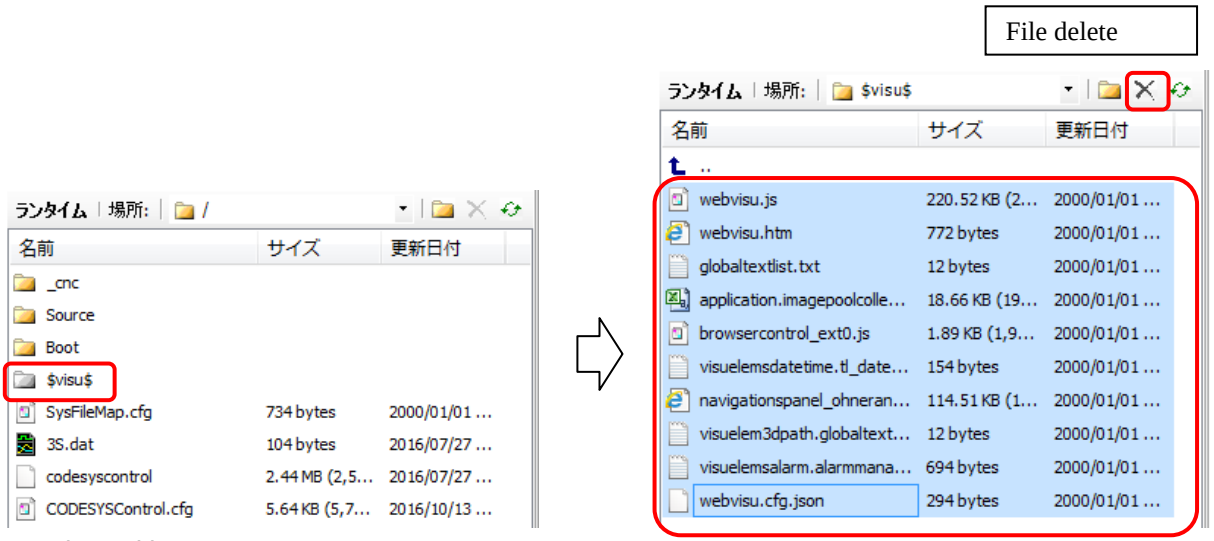


Left display appears by press [Cancel]

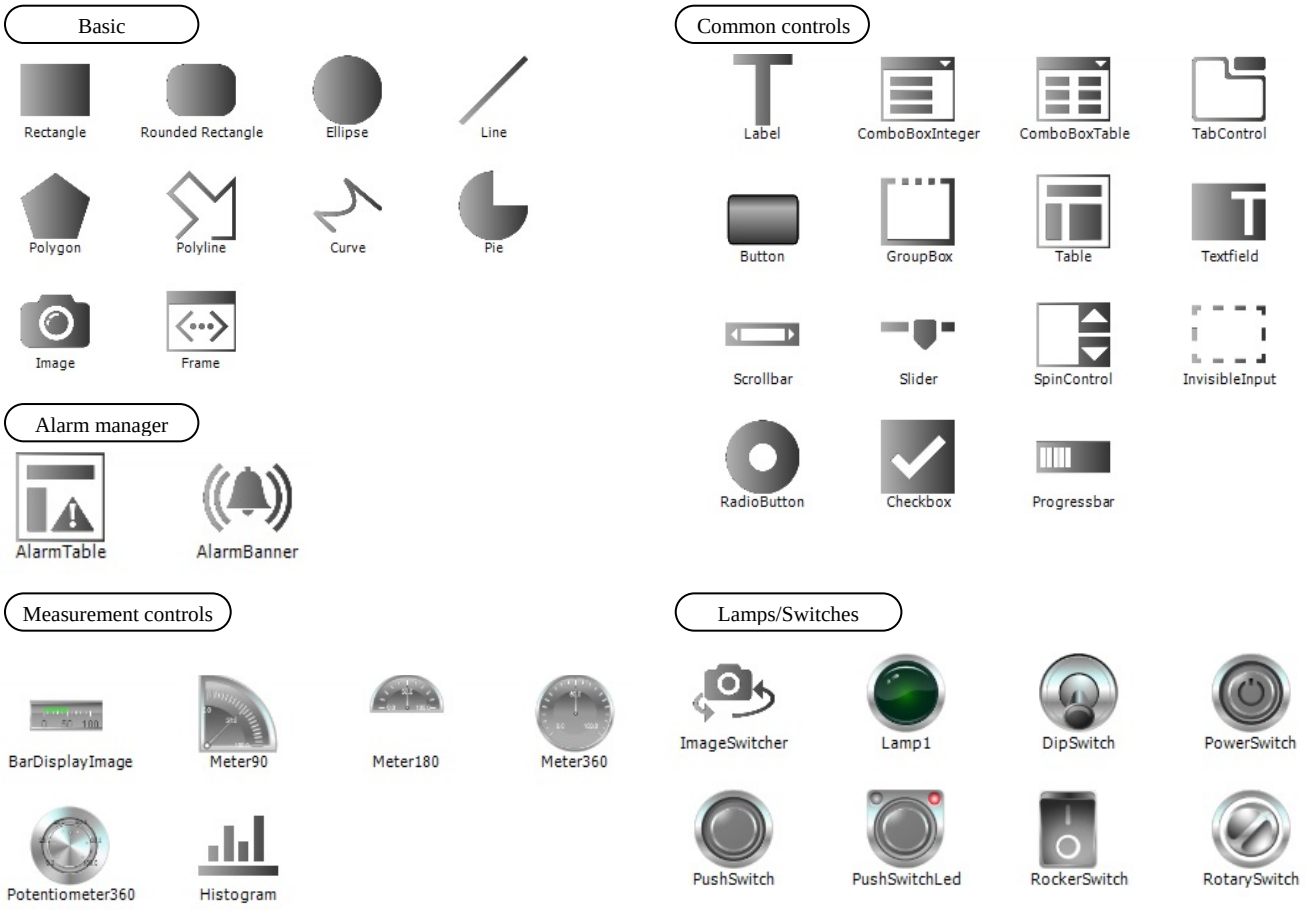
Method of Visualization files deleting

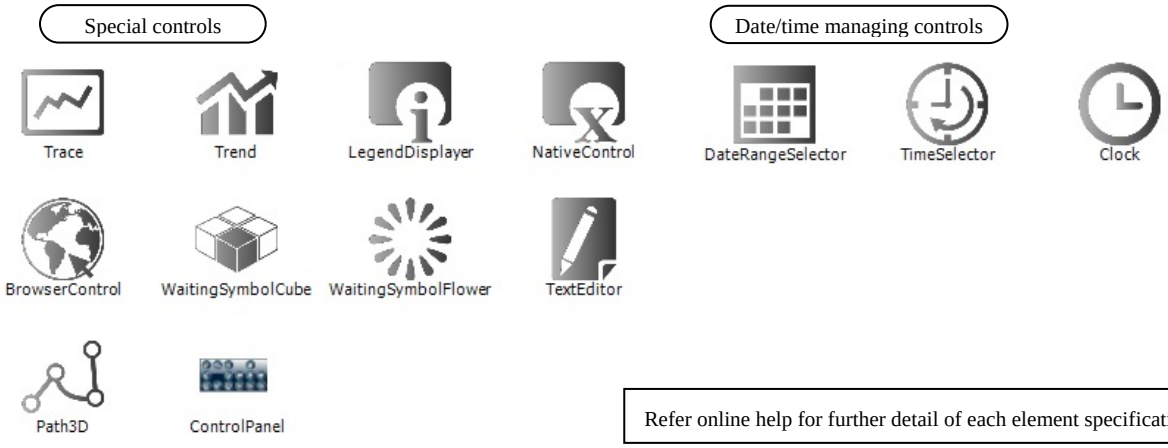
Visualization file registered in HX-CPU, this file is registered dedicated Visu folder inside of HX-CPU.

If Visualization file name is changed, the file having previous name is still remain and remaining free area of folder is shorted , [File Transfer Error] will appear. It is possible to clear inside of [$visu$] folder by file delete in the right click menu of device tree. Do not delete files other than inside of [$visu$] folder.



Element in Tool box





Refer online help for further detail of each element specifications.

Make sure what is object of security protecting and take countermeasure for system configuration and operation mentioned security protection as an example by user responsibility.

- Usage of certification function and regular review for program and data should be protected.
- Usage of security function for devices used in network system.
- Connecting protection with unspecified target by usage of specifying function for connecting target.
- Operation management protection by making limitation of key lock of device setting place or user limitation.

4.4 Calendar Clock Function

HX-CPU includes Calendar clock IC, Clock data can be used in program as system clock. Use this function by setting system clock and time zone information. Configure clock data by dedicated function block (CAA DTUtil library). Refer “System clock command” of HX Application Manual [Command reference edition] for further detail.

Table 4.4 List of CAA DTUtil library

Command	Function
GetDateAndTime	Get system clock
SetDateAndTime	Set system clock
GetTimeZoneInformation	Get time zone information
SetTimeZoneInformation	Set time zone information

Time zone

There are two type of time zone in HX-CPU. Use Clock function by setting same information for both two time zone.

Table 4.5 List of time zone

Time zone	Configure method	Target
Time zone 1	Function block （SetTimeZoneInformation）	GetTimeZoneInformation
Time zone 2	Configuration （PLC Parameters）	GetNTPStatus

Caution

Maximum system clock data of HX-CPU is 2038 January 19th 03:14:17. Configure and manage of clock data not to exceed maximum data due to exceeding maximum value operation may not be correct.
Here is example program to restart from 2000 January 1st 00:00:00 automatically if clock data exceed 2037 December 31st 23:59:59. Use this example program by modifying according to the system specification. Configure NTP Client function invalid, if following example program.

Declaration part of variable.

```
PROGRAM POU
VAR
          BLINK_0: BLINK;
          GetDateAndTime_0: DTU.GetDateAndTime;
          SetDateAndTime_0: DTU.SetDateAndTime;
          GET_TIME_ERROR: BOOL;
          SET_TIME_ERROR: BOOL;
END_VAR
```

Program

```
BLINK_0(ENABLE:=TRUE, TIMELOW:=T#719M, TIMEHIGH:=T#1M);
GetDateAndTime_0(xExecute:=BLINK_0.OUT,
                 xError=>GET_TIME_ERROR);
IF (GetDateAndTime_0.dtDateAndTime >=DT#2037-12-31-23:59:59) THEN
           SetDateAndTime_0(xExecute:=TRUE,
                            xError=>SET_TIME_ERROR,
                            dtDateAndTime:=DT#2000-01-01-0:00:00);
END_IF
```

4.5 NTP Client Function

NTP client function getting clock information from NTP (Network Time Protocol) of network is available with HX-CPU.
It is also possible to set clock information of HX-CPU Calendar Clock IC by getting clock information from NTP server. Get clock information when start RUN, it can be possible to set 1 minute to 1440 minute (24 hours) interval and to get clock information by specified cycle.

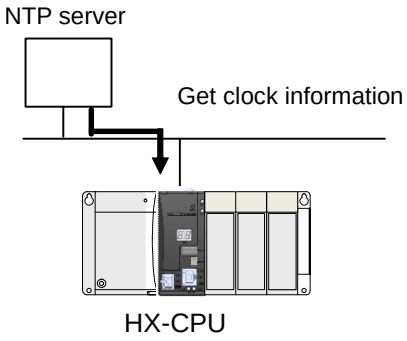


Figure 4.18 NTP Client function

Table 4.6 Specification of NTP client

Items	Specification
Communication protocol	SNTP（Simple Network Time Protocol）
Getting cycle	Start RUN timing, User configuration (00:01-24:00)
Collected clock data	Year / Month and date / Day / Hour / minute / second (data type: DATE_AND_TIME)
Refresh getting interval	Refresh by calendar time clock IC

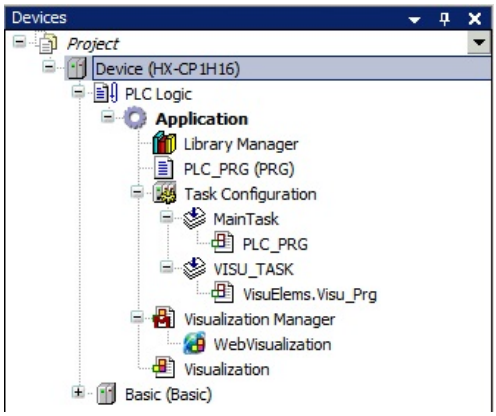
It is possible to get NTP status by using dedicated function block (GetNTPstatus). ExecNormal of GetNTPstatus turn TRUE by getting clock data correctly, therefore if clock data is used in user program, use it after confirming ExecNormal of GetNTPstatus turns TRUE.

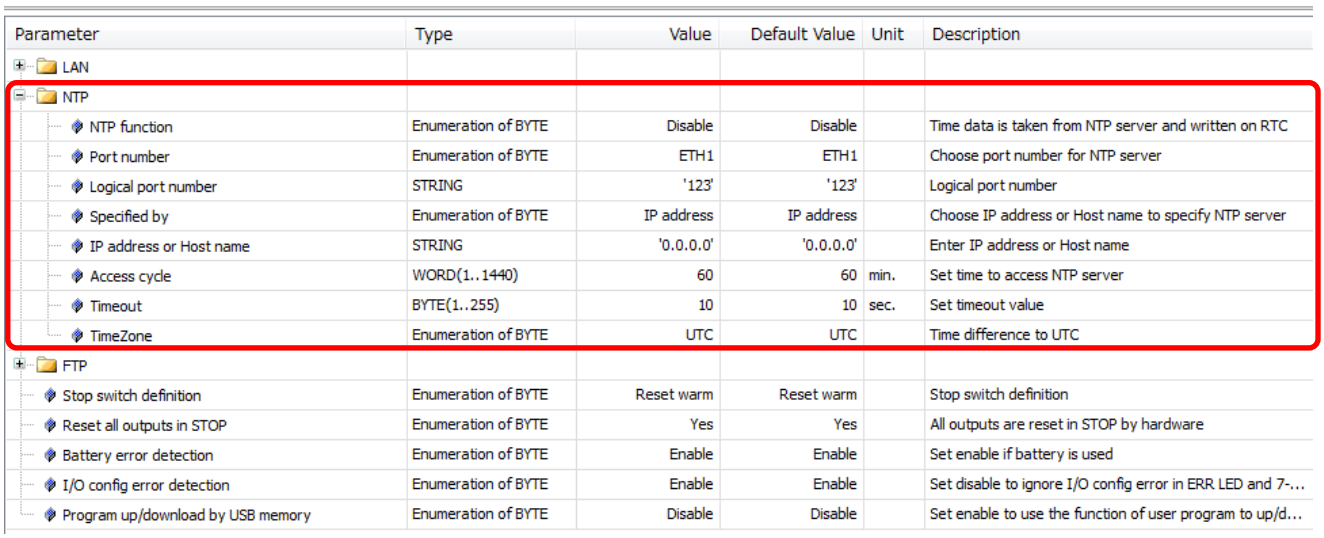
Caution

If NTP client function is used, use it after setting time zone. Configuration is done by dedicated function block (SetTimeZoneInformation). Refer “System clock command” of HX Application Manual [Command reference edition] for further detail information.

Configuration Method

[Device] window appears after selecting [Edit Object] by double clicking or right clicking [Device (HX-CP…)] on Device tree. Select items of [NTP] by double clicking [PLC Parameter] tab.





Parameter	Type	Value	Default Value	Unit	Description
LAN					
NTP					
NTP function	Enumeration of BYTE	Disable	Disable		Time data is taken from NTP server and written on RTC
Port number	Enumeration of BYTE	ETH1	ETH1		Choose port number for NTP server
Logical port number	STRING	'123'	'123'		Logical port number
Specified by	Enumeration of BYTE	IP address	IP address		Choose IP address or Host name to specify NTP server
IP address or Host name	STRING	'0.0.0.0'	'0.0.0.0'		Enter IP address or Host name
Access cycle	WORD(1..1440)	60	60	min.	Set time to access NTP server
Timeout	BYTE(1..255)	10	10	sec.	Set timeout value
TimeZone	Enumeration of BYTE	UTC	UTC		Time difference to UTC
FTP					
Stop switch definition	Enumeration of BYTE	Reset warm	Reset warm		Stop switch definition
Reset all outputs in STOP	Enumeration of BYTE	Yes	Yes		All outputs are reset in STOP by hardware
Battery error detection	Enumeration of BYTE	Enable	Enable		Set enable if battery is used
I/O config error detection	Enumeration of BYTE	Enable	Enable		Set disable to ignore I/O config error in ERR LED and 7-...
Program up/download by USB memory	Enumeration of BYTE	Disable	Disable		Set enable to use the function of user program to up/d...

Table 4.7 Configuration Items of NTP client

Item name	Contents	Setting range
NTP function	Select valid or invalid getting clock information from NTP server.	Disable / Enable
Port number	Select communication port getting clock information.	ETH1 / ETH2 / ETH3
Logical port number	Set port number using for NTP server connection.	123 (Not changeable)
Specified by	Select specifying method of NTP server.	IP address (Fixed IP address)
IP address or Host name	Specify NTP server.	xxx.xxx.xxx.xxx
Access cycle	Set time interval of clock information.	1 to 1,440 (unit: minute)
Timeout	Set detecting time of timeout.	10 (unit: minute (fixed))
TimeZone	Specify time zone.	UTC-12:00 to UTC+12:00

4.6 Removable Media

HX-CPU supports USB memory and SD card (HX-CP1H16) as removable media and file access is possible. Here is sample program description executing data logging on removable media by using CAA File.
This sample program is 3 kinds data (time stamping, dummy data, text) logging making CSV file (File name: LoggingSample.csv) on USB memory. New data is added every 10 seconds automatically. Modify data writing timing according system usage.

	A	B	C
1			
2	DT#2016-05-17-21:25:22	1	This is Test !
3	DT#2016-05-17-21:25:32	2	This is Test !
4	DT#2016-05-17-21:25:42	3	This is Test !
5	DT#2016-05-17-21:25:52	4	This is Test !
6	DT#2016-05-17-21:26:02	5	This is Test !
7	DT#2016-05-17-21:26:13	6	This is Test !

Declaration of variable

```
PROGRAM PLC_PRG
VAR
        USBMountSts     : BOOL;
        SDMountSts              : BOOL;
        sDirSD          : STRING :='/media/sd-mmcblk0p1'; // SD card
        sDirUSB         : STRING :='/media/usb-sda1';    // USB memory
        sFileName               : CAA.FILENAME;
        FileOpen        : File.Open;
        FileClose               : File.Close;
        FileWrite               : File.Write;
        FileFlush               : FILE.Flush;
        sMedia          : STRING;
        iState          : UINT :=1;
        hfile           : CAA.HANDLE;
        sWriteLine              : STRING(128);
        GetRTC          : DTU.GetDateAndTime;
        xRDRTC          : BOOL;
        xReadDone               : BOOL;
        dtTemp          : DATE_AND_TIME;
        sDT             : STRING;
        T1              : TON;
        FileOpenDone    : BOOL;
        FileOpenErr     : BOOL;
        FileWriteDone   : BOOL;
        FileWriteErr    : BOOL;
        FileFlushDone   : BOOL;
        FileFlushErr    : BOOL;
        Err             : BOOL;
        Exclsv          : BOOL;
        iNum            : INT;
        sNum            : STRING;
END_VAR
```

Program

```
USBMountSTS:=UsbMountStatus();
IF USBMountSTS=FALSE THEN
   RETURN;
END_IF

//SDMountSts:=SDMountStatus();
//IF SDMountSTS=FALSE THEN
//       RETURN;
//END_IF

CASE iState OF

         1:
//       sMedia := sDirSD; // SD card
         sMedia := sDirUSB; // USB memory
         sFileName :='/LoggingSample.csv';
         sFileName := CONCAT(sMedia, sFileName);
         iState := 2;

         2: // FileOpen: Mode.MWRITE
                  FileOpen(xExecute:=TRUE, sFileName:=sFileName, xExclusive:=Exclsv, eFileMode:=
File.MODE.MWRITE);
                  IF FileOpen.xDone = TRUE THEN
                           FileOpenDone:=TRUE;
                           hfile:=FileOpen.hFile;
                           FileOpen(xExecute:=FALSE);
                           xRDRTC := TRUE;
                           iState:=10;
                           ELSIF FileOpen.xError=TRUE THEN
                           FileOpenErr:=TRUE;
                           FileOpen(xExecute:=FALSE);
                           iState:=90;
                  END_IF

         3: // FileOpen Mode.MAPPD
                  FileOpen(xExecute:=TRUE, sFileName:=sFileName, xExclusive:=Exclsv, eFileMode:=
File.MODE.MAPPD);
                  IF FileOpen.xDone = TRUE THEN
                           iState:=10;
                           hfile:=FileOpen.hFile;
                           FileOpen(xExecute:=FALSE);
                           xRDRTC := TRUE;
                  ELSIF FileOpen.xError=TRUE THEN
                           FileOpen(xExecute:=FALSE);
                           iState:=90;
                  END_IF

         10: // Get RTC data
                  GetRTC(xExecute:=xRDRTC, xDone=>xReadDone, dtDateAndTime=>dtTemp);
                  IF xReadDone=TRUE THEN
                           sDT:=DT_TO_STRING (dtTemp);
                           GetRTC(xExecute:=FALSE);
                           iState:=11;
                  END_IF

         11: // Create & Combine the data
                  iNum := iNum +1;
                  sNum := INT_TO_STRING(iNum);

                  sWriteLine := CONCAT('$r$n', sDT);
                  sWriteLine := CONCAT(sWriteLine, ',');
                  sWriteLine := CONCAT(sWriteLine, sNum);
                  sWriteLine := CONCAT(sWriteLine, ',');
                  sWriteLine := CONCAT(sWriteLine, 'This is Test ! ');
                  iState:=12;
```

Enable these when logging data on SD card.

```
        12: // FileWrite
                FileWrite(xExecute:=TRUE,hFile:=hfile,pBuffer:=ADR(sWriteLine),
                        szSize:=INT_TO_UDINT(LEN(sWriteLine)));
                IF FileWrite.xDone = TRUE THEN
                   FileWrite(xExecute:= FALSE);
                   FileWriteDone:=TRUE;
                   iState:=20;    // To Flush
                ELSIF FileWrite.xError = TRUE THEN
                        FileWriteErr:=TRUE;
                        FileWrite(xExecute:= FALSE);
                        iState:=90;
                END_IF
                        xRDRTC := FALSE;

        20: // FileFlush
                FileFlush(xExecute:= TRUE, hFile:= hfile);
                IF FileFlush.xDone = TRUE THEN
                   FileFlush(xExecute:= FALSE);
                   FileFlushDone:=TRUE;
                   iState:=30;    // Close that file
                ELSIF FileWrite.xError = TRUE THEN
                   FileFlushErr:=TRUE;
                   FileFlush(xExecute:= FALSE);
                   iState:=90;
                END_IF

        30: // Close that file.
                FileClose(xExecute:= TRUE, hFile:= hfile);
                IF FileClose.xDone = TRUE THEN
                   iState := 40;
                   FileClose(xExecute:= FALSE);
                ELSIF FileClose.xError = TRUE THEN
                   FileClose(xExecute:= FALSE);
                   IState := 90;
                END_IF

        40: // wait 10 seconds
                T1(IN:=TRUE, PT:=T#10S);
                IF t1.Q THEN
                        iState:=3;
                        T1(IN:=FALSE);
                END_IF

        90: // Error
                Err:= TRUE; // Error

END_CASE;
```

Caution

- Access files after confirming USB memory mounting status or SD card mounting status by program when access files of USB memory or SD card. SdMountStatus command and UsbMountStatus command are prepared to get mounting status.
- Don’t remove USB memory or SD card during accessing file or directory. It will be cause can’t access again. If USB memory or SD card needed to be removed during PLC is in RUN status, removing action like SD card switch pressing and execution of UsbUnmount command required after executing Flush/close command. It is convenient to execute by prepared input variable for invoking UsbUnmount command.
- It may take long time for file accessing time depends on USB memory or SD card type. It is recommended to separate file access task and general I/O access task.
- There are some possibilities to access file of USB memory or SD card not only from CAA File but also from FTP client etc at same timing. To avoid this situation, exclusive file access control is required.

4.7 Supporting Function for Security Protection

Supporting function for Security protection protecting illegal access is available on HX-CPU to prepare external security risk via network accessing. Use this supporting function as one of method to keep needed security level for system.

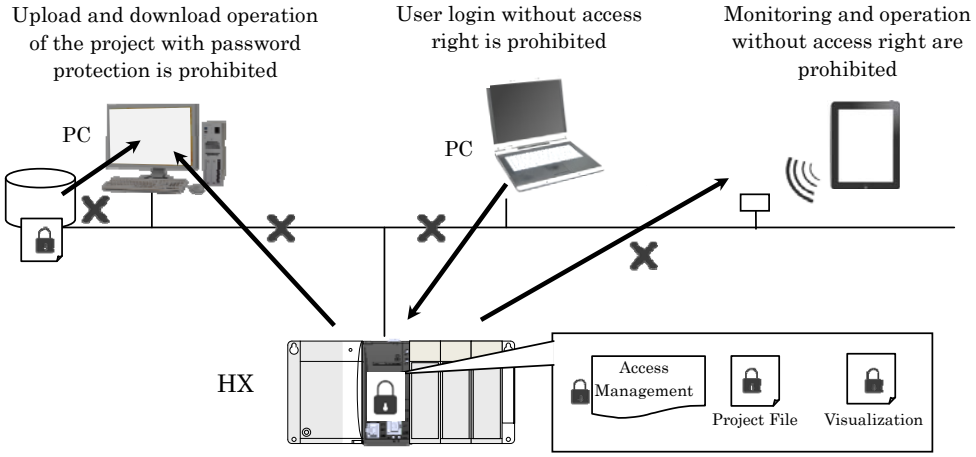
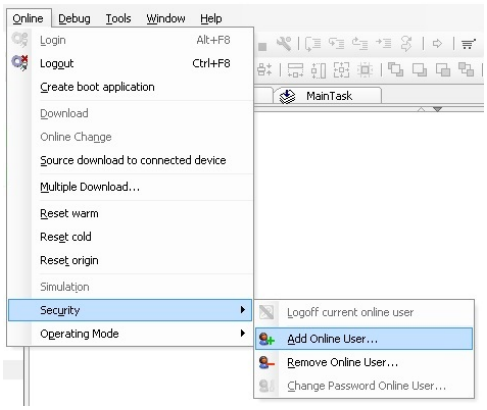


Figure 4.19 Supporting function for security protection via network access

Limitation of online user

It is possible to limit user online connecting HX-CPU. Only the registered user can login HX-CPU by registering name and password. This is same as for Visualization or Web Visualization.

[Online]-[Security]-[Add Online user]



Dialog of Add Online user

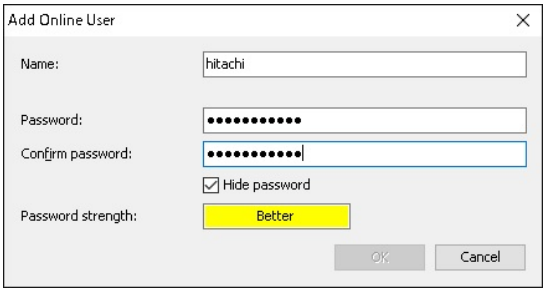
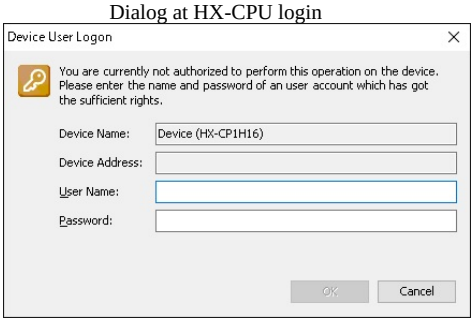


Figure 4.20 Add online user

Following dialog appears at login after registering online user. Input registered name and password.

Dialog at HX-CPU login



At Web visualization access

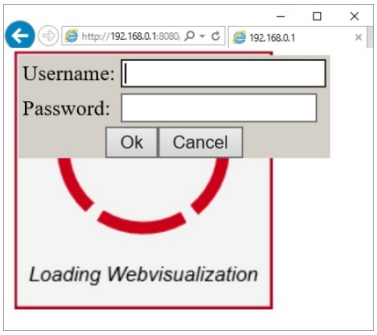


Figure 4.21 Access of HX-CPU registered online user

Dialog at login appears again when user name or password is different from registered. Confirm user name and password with manager registered online user when login can’t be possible.

It is recommended like in below not to image easily.

Length of password more than equal 8 characters (Most suitable is 12 characters or more)
Mixture capital letter and small letter
Mixture number
Mixture special character
Avoid existing name or easily guessed phrase for password (“123”, “abc” or “qwerty” etc)

Caution

Login with empty user name and password can’t be used after adding new user with this procedure.
Please make note registered password without missing.

Registered user can delete online user registration after login. If not to use dialog at login, add new user with name “Everyone” and empty password. It is possible to login without login dialog from next login.

Caution

Be careful, online user is deleted by [Reset origin device] operation.

Password protection of Project

Password protection is possible for project file.

This protection is valid when to open stored file or to open uploaded stored source program in HX-CPU by HX-CODESYS.

Open dialog of [Project]-[Project Settings]-[Security].

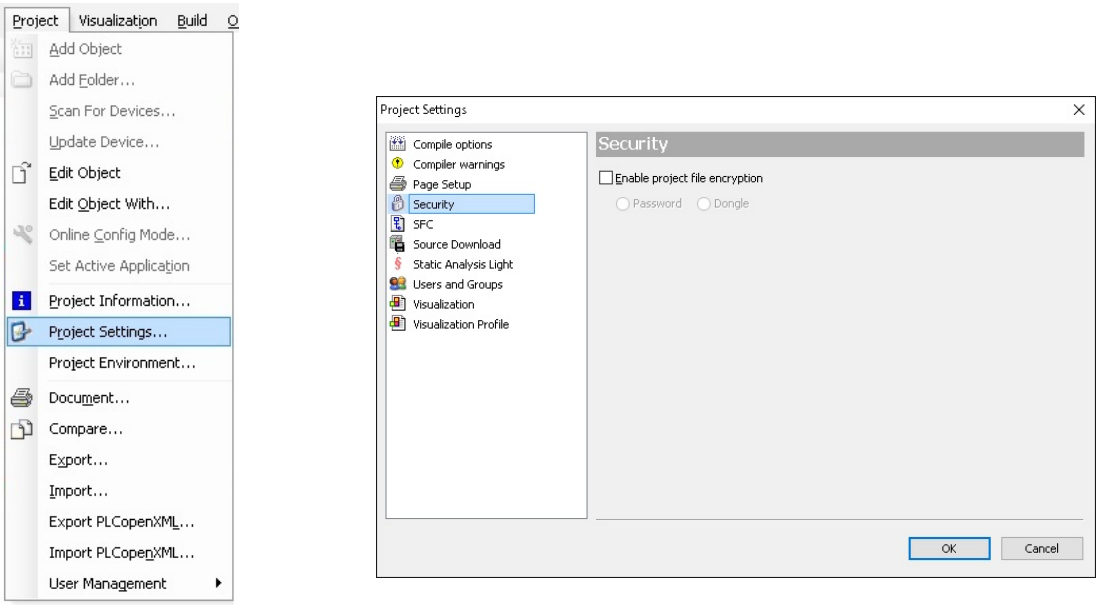


Figure 4.22 Configuration of Project password

Select [Password] by check on [Enable project file encryption].

Input current password, new password and new password confirming.

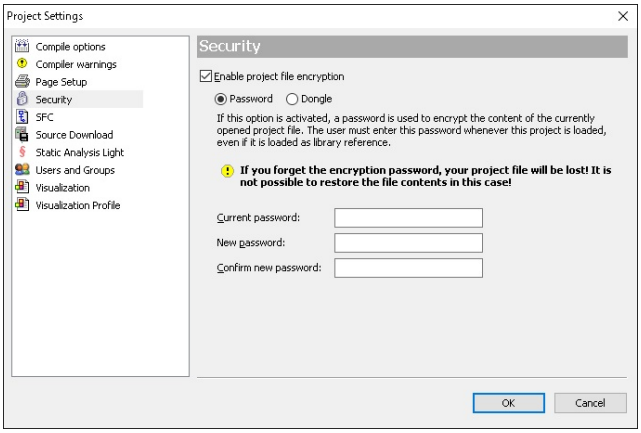
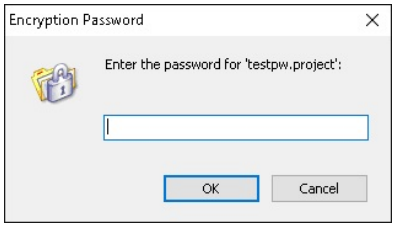


Figure 4.23 Input Project password

Then protection is enabled when to open stored file or to open uploaded source program stored in HX-CPU.

At the time to open project or uploading



In case of password is not correct

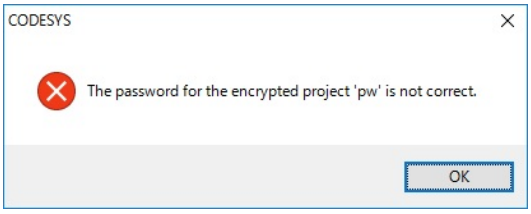


Figure 4.24 Input Project password

Caution

Project can’t be opened if forgetting coding password. Please be careful not to forget password and manage it.

Access limitation of Visualization

It is possible to make access limitation for Visualization page or Display element.
Configure access right for each group by registering user and group belonging its user.

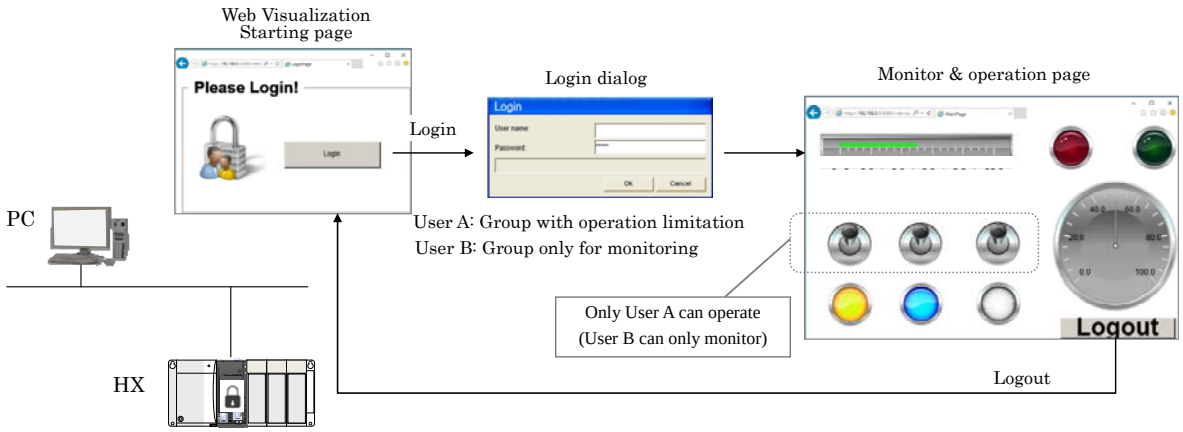


Figure 4.25 Visualization example with access limitation

User management of Visualization

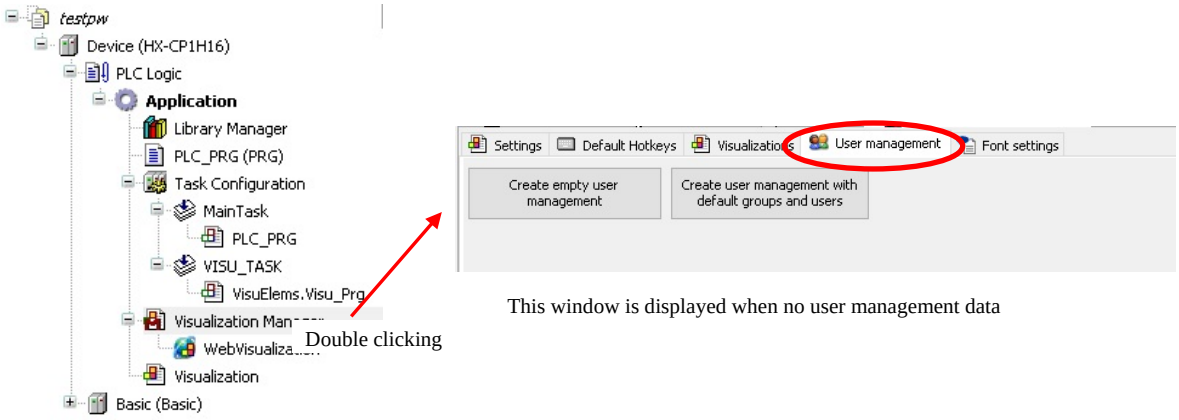
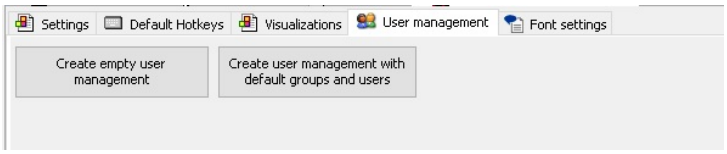
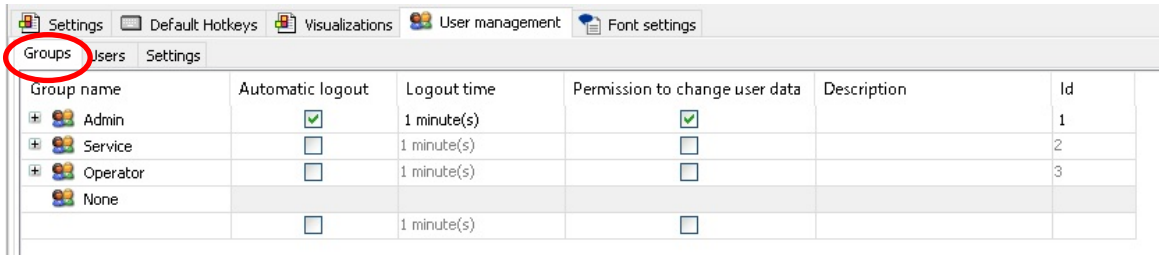


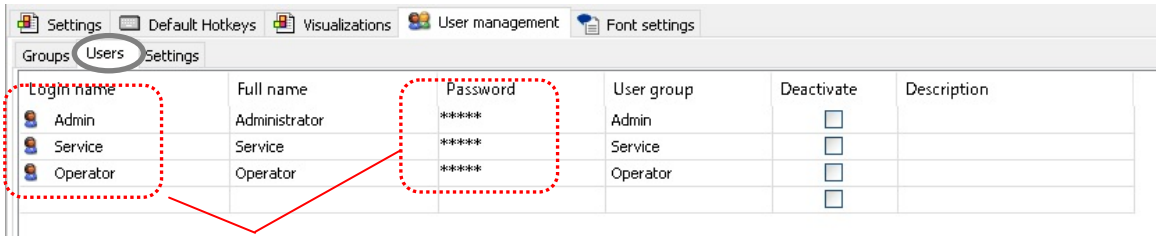
Figure 4.26 Visualization manager

There is no user management data as default setting. Click [Create user management with default groups and users] in order to register user management group and user.



Configuration of Group and User is default setting in below. At this default setting, user belongs to Group [Admin] are configured having right of data changing.



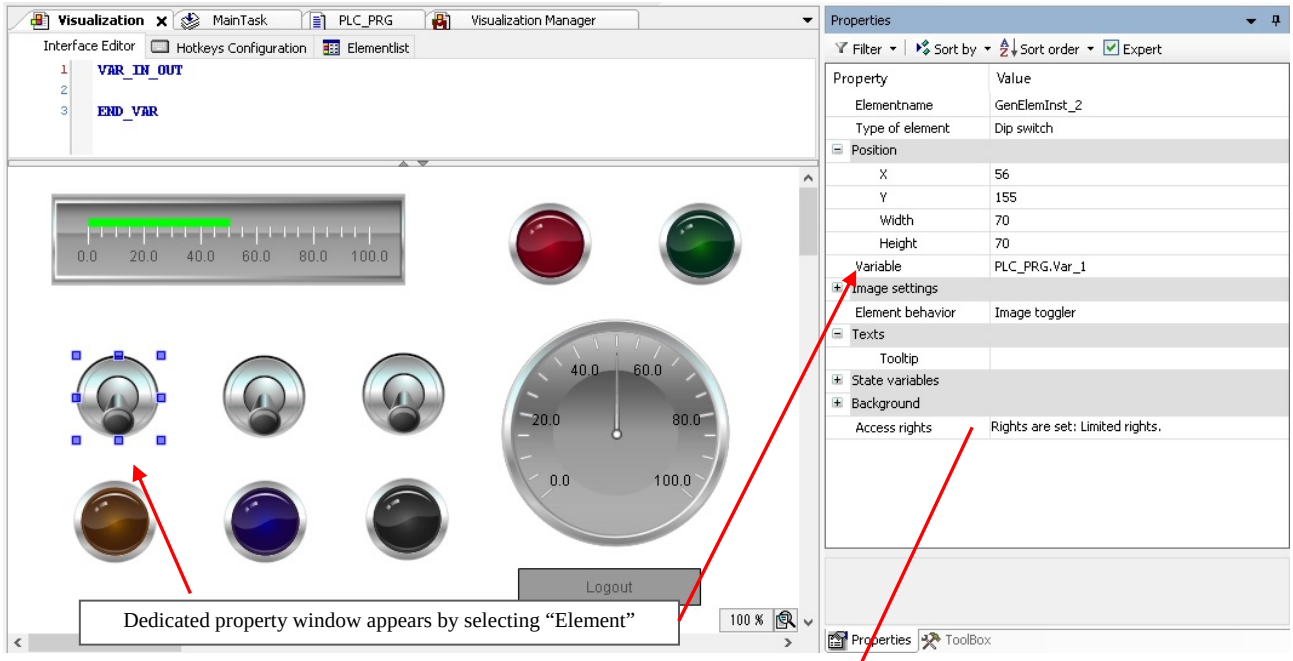


User name to password are same at default setting

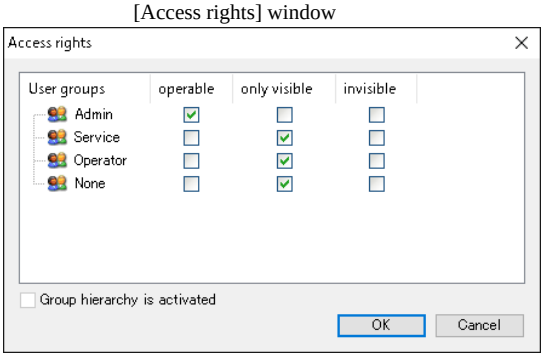
Figure 4.27 User management Visualization manager

Next is explanation of Element configuration of Visualization.

Configure [Access rights] on the property window of element.



[Access rights] window



Double clicking

Configure other than user of Group [Admin] can’t operate.

Figure 4.28 Access right configuration for element of Visualization

⚠ Caution

In the control system, recently, the connection and cooperation with the information communication system progress and information security risks including cyber attacks are growing. In a system applying this product, physical security measures mainly in the installation location and security measures in use via network are needed.

[Security risk example via the network]
- Abnormal operation, performance degradation, information leakage and data tampering by attacks from outside
- Malfunction, harm and damage occurrence due to programs and/or data tampering from outside
- It is used as an attacking step for the-other systems

Hitachi Group is striving security improvement of control systems by establishing prerequisite protecting target defined for each product and equipping security protection functions under the own provision security design procedure.

In order to deal with the security risks from the outside via the network, this product is equipped with a security protection support functions for the purpose of prevention of unauthorized access. However, the security level to be determined by the control system. In addition, the assumed security risk is not fixed, it will be something to change on a daily basis.

Not only in our products, individual security protection support functions of each product configuring the system is one means to ensure the security level required for the system, it does not completely prevent the security risk growing daily.

The construction of the security level required for the control systems are responsible by the system and customer. In addition, for the maintenance of the security level will require continuous improvement measures.

In a system using this product, regardless of the presence or absence of the use of security protection support functions, trouble, accident or damages caused by unauthorized external access, Hitachi Group will not be able to bear any responsibility.

It is required for the customer side to clarify the target of the security protection of the system, following the conduct measures example to a representative, please refer to the construction and operation of the system.

-Utilization and regular review of the authentication function for the program and the data to be protected

-Utilize the security functions of the device configuring the network

-Prevention of the unspecified connection by the use of a particular function to identify connection

-Measures in the operational management, such as to lock the location of devices or limit the operator

MEMO

Chapter 5 Debug Function

Several debugging functions are available in HX-CODESYS. In this chapter, following debug functions are described.

Table 5.1 List of Debug function

No.	Functions	Refer to
1	Monitor	5.2
2	Flow control	5.3
3	Break point	5.4
4	Single cycle and Step over / into / out	5.5
5	Force values and Write values	5.6
6	Trace	5.7

There are two methods, one is online debug done by connecting with HX-CPU, the other is offline debug executing simulation on HX-CODESYS.

5.1 How to Start

Even it is same operation of debug operation between online debug and offline debug, starting operation is different. Each method how to start is described in below.

How to start offline debug

Login and start execution.

How to start simulation

Click [Online]- [Simulation]. Login and start execution.

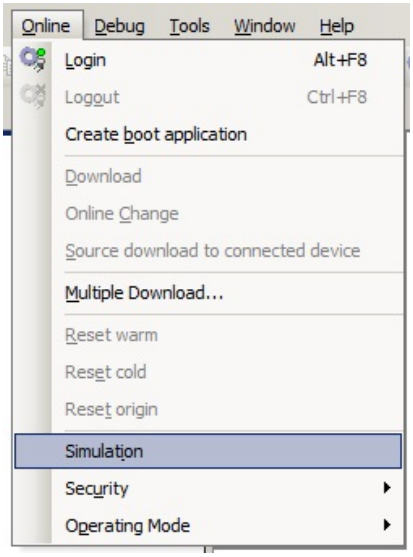


Figure 5.1 Simulation debug

5.2 Monitor

Monitor function is to monitor contact, coil and current value of variable and it is possible to monitor variable declaration part, LD (Ladder logic diagram) display part, FDB (Function Block Diagram) part, ST (Structured Text) display part etc.

Declaration variable part

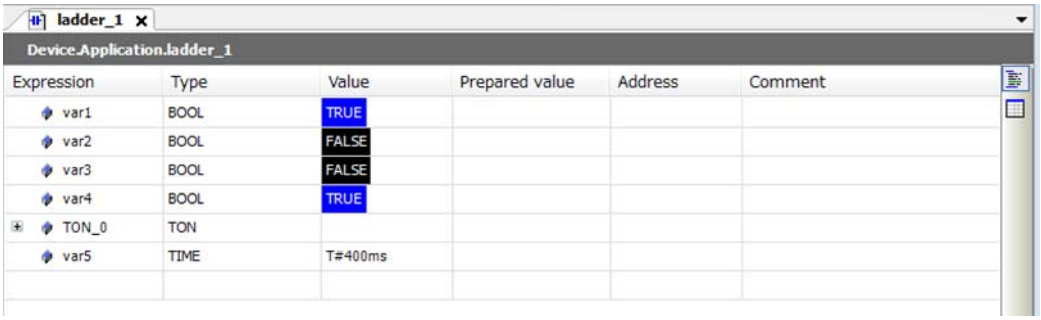


Figure 5.2 Monitor value of variable

Table 5.2 Monitor value of variable

Display	Contents
TRUE	Variable is TRUE.
FALSE	Variable is FALSE.
T#400ms	Value is 400 ms.

Ladder logic diagram

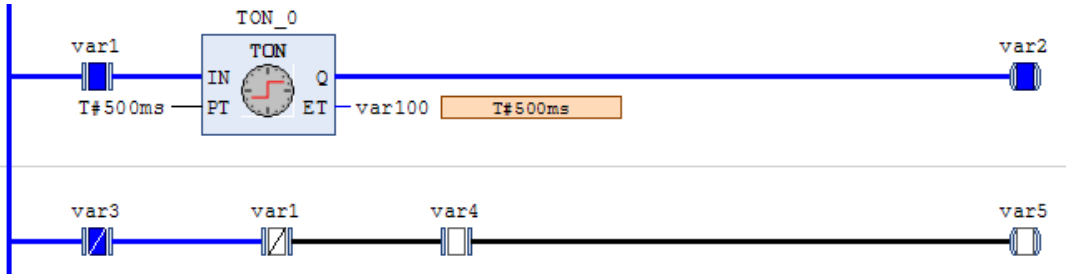


Figure 5.3 Monitor ladder logic diagram

Table 5.3 Monitor ladder logic diagram

Symbol	Description
var1	Contact is TRUE.
var3	Negative contact is TRUE.
var2	Contact is FALSE.
var1	Negative contact is FALSE.
var2	Coil is ON.
var5	Coil is OFF.
T#500ms	Value is 500 ms.

Function Block Diagram

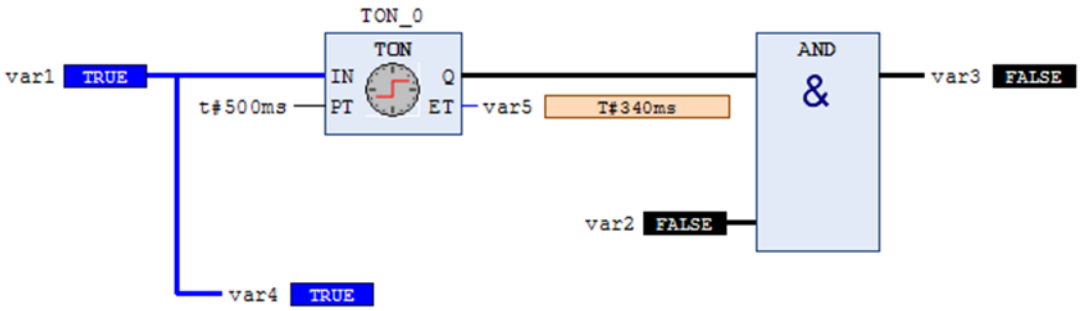


Figure 5.4 Monitor function block diagram

Table 5.4 Monitor function block diagram

Display	Contents
TRUE	Value is ON.
FALSE	Value is OFF.
T#340ms	Value is 340 ms.

Structured text

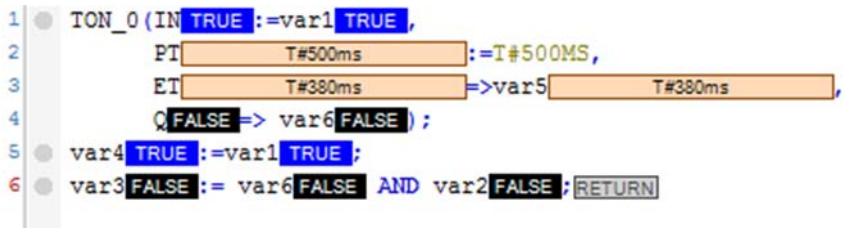


Figure 5.5 Monitor structured text

Table 5.5 Monitor structured text

Display	Contents
TRUE	Value is ON.
FALSE	Value is OFF.
T#500ms	Value is 500 ms.

Change display mode

It is possible to change display for variable monitor with binary, decimal or hexadecimal. Select [Debug]-[Display mode].

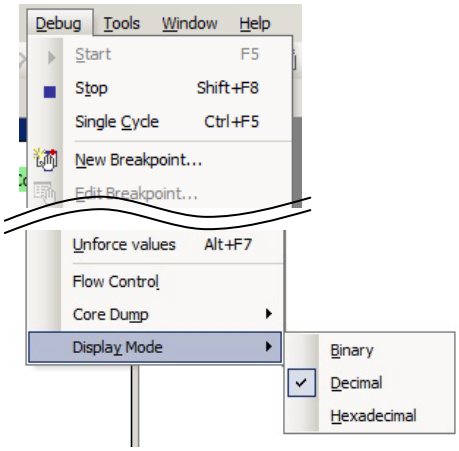


Figure 5.6 Change display mode

Binary display

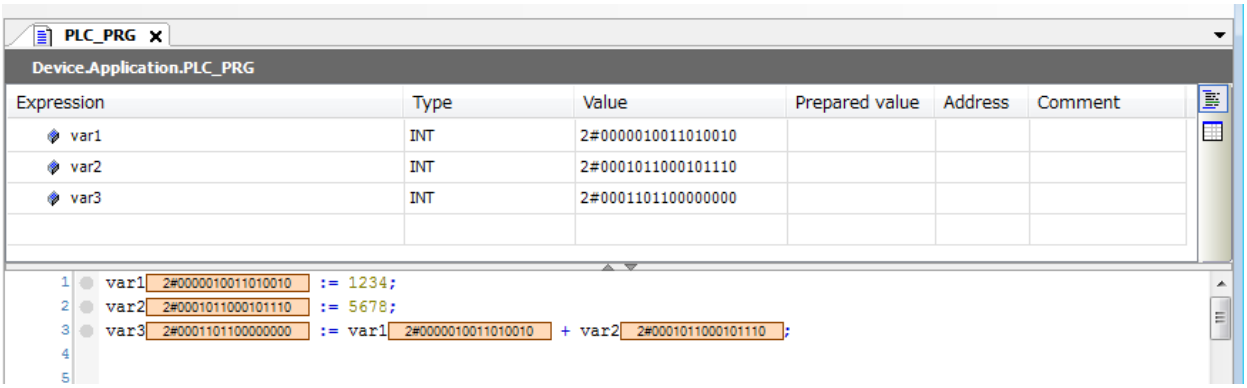


Figure 5.7 Binary display

Decimal display

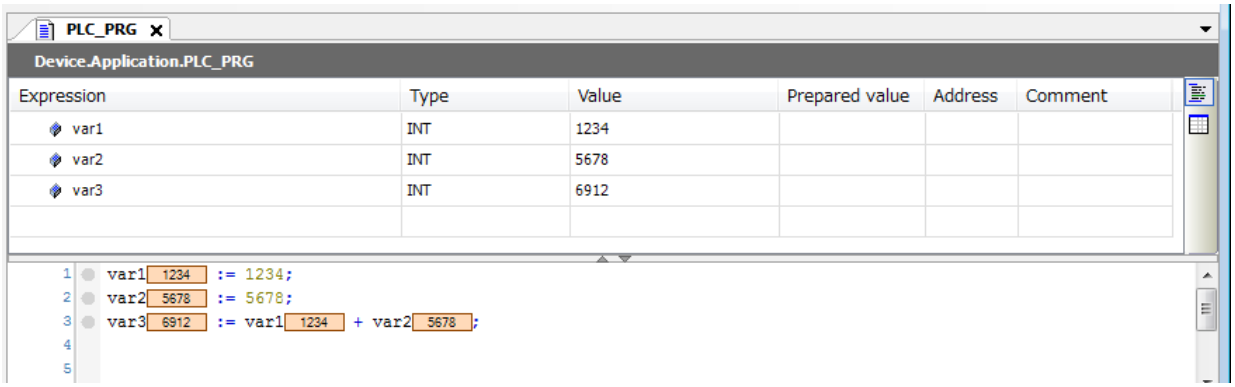


Figure 5.8 Decimal display

Hexadecimal display

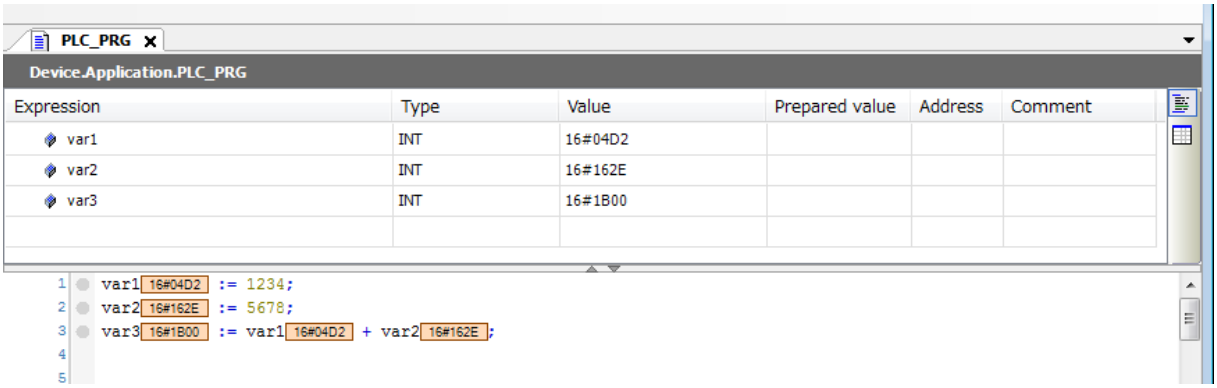


Figure 5.9 Hexadecimal display

Array variable monitor

It can't be display with array variable more than 1,000 if array declaration exceeds 1,000 at the default status.

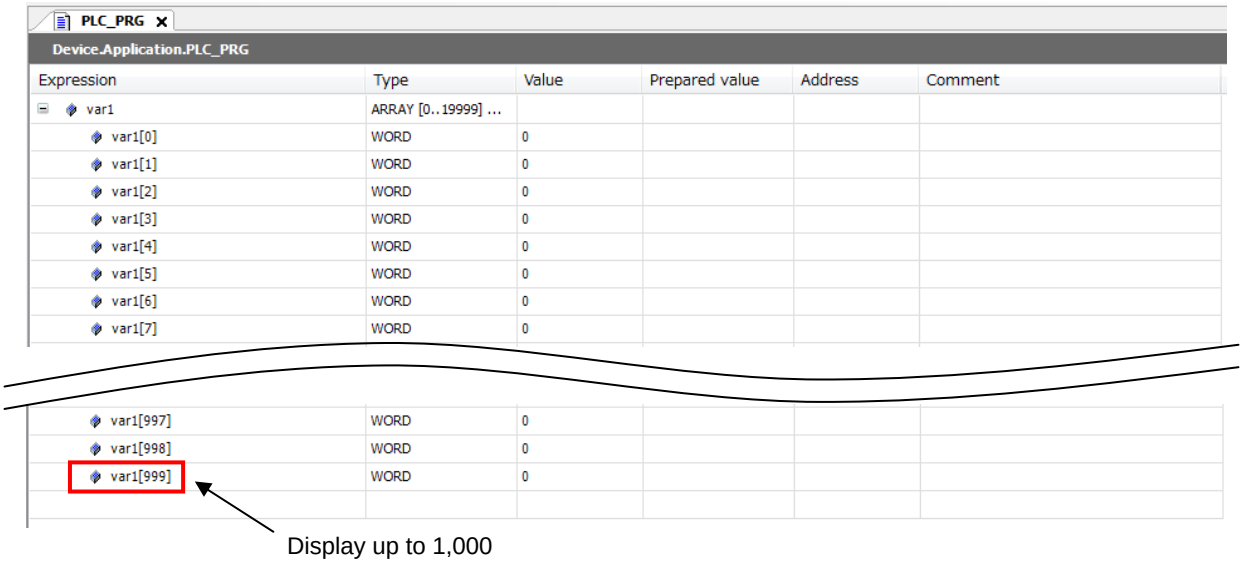


Figure 5.10 Array variable monitor 1

Change range of list display by double clicking [ARRAY[*..***]O..] to monitor array variable more than 1,000. List display is extended up to 20,000.

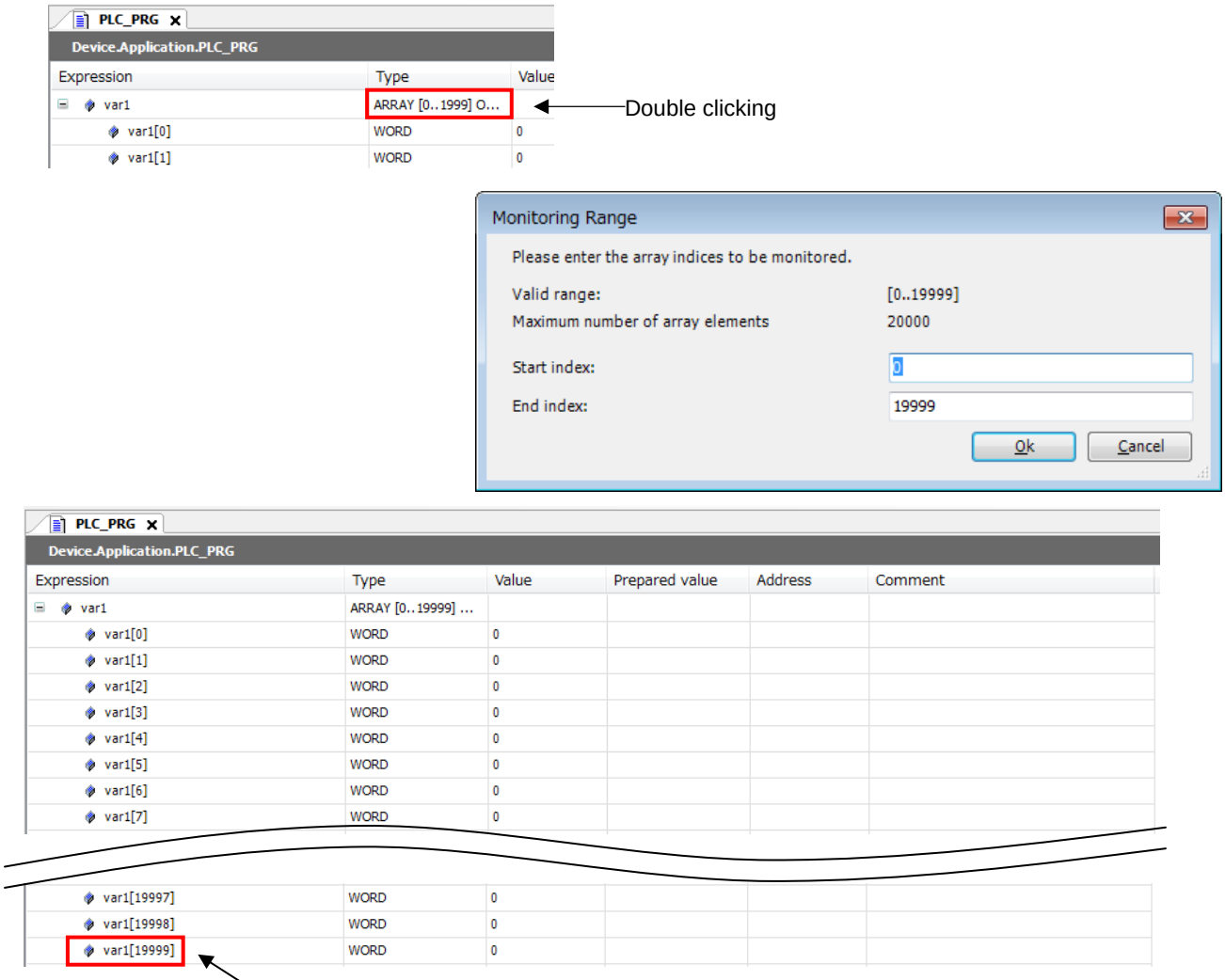


Figure 5.11 Array variable monitor 2

5.3 Flow Control

Flow control function is possible to confirm executing part of program by indicating green color on the part of execution.

Click [Debug]-[Flow Control] after login.

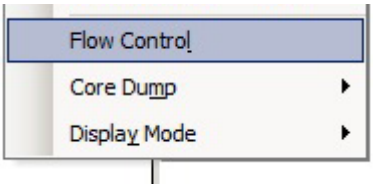


Figure 5.12 Enable flow control

Display only executing part colored green.

```
1  IF x 70 > 100 THEN
2      x 70 :=0;
3  ELSIF x 70 < 50 THEN
4      a 444 := 111; b 555 := 222; c 666 := 333;
5  ELSE
6      a 444 := 444; b 555 := 555; c 666 := 666;
7  END_IF
```

Executing program part.

Figure 5.13 Display flow control

In the above example, there are three condition [x>100], [x<50] and [other] divided by condition of x value. Display executing part with green color in the case of [x=70] shows [other].

5.4 Break Point

Break point function is possible to stop program at the position specified stopping position of application program for debugging.
The possible break point position is the position can be changed value of variable or the position to call program after branch.

Configuration method of Break point

Show example of Break point with structured text.
Grey colored circle appears where break point can be set after login.

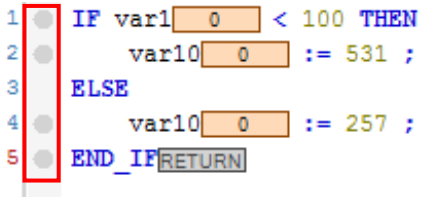


Figure 5.14 Break point position

After click [New Breakpoint] of [Debug] menu, property screen of Break point appears and then click [Location] tab. Specify POU to set Break point at [POU]. Specify the line to set Break point at [Position]. Check [Enable breakpoint immediately] to enable break point immediately after specifying.

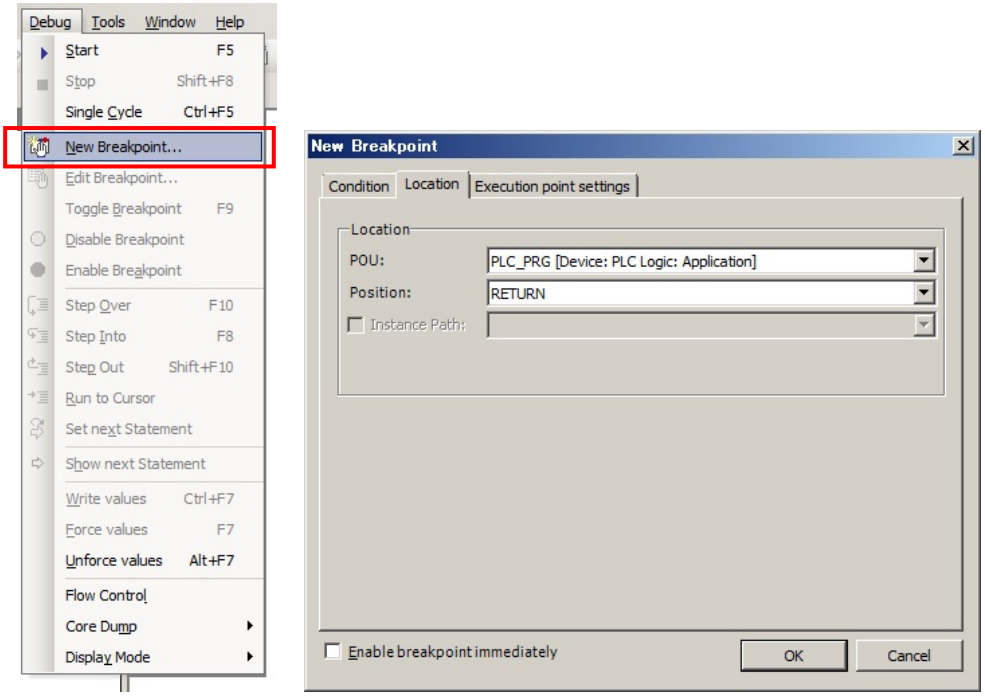
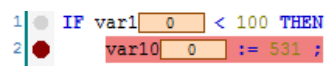
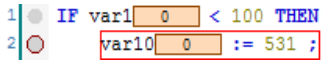


Figure 5.15 Break point settings

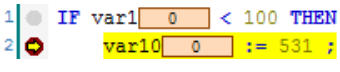
Display red colored circle at valid Break point.



Display grey colored circle at invalid Break point.



Display yellow colored allow at the position program stopped.



5.5 Single Cycle and Step Over / Into / Out

5.5.1 Single Cycle Execution

Single cycle execution function is possible to execute only 1 cycle.

Execute only 1 cycle of program by clicking [Debug]-[Single cycle]. If there are several tasks, all tasks are executed only 1 cycle instead of task cycle.

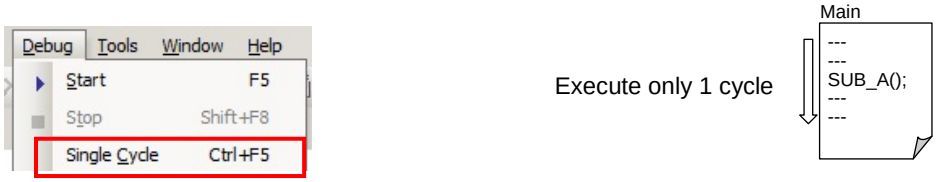


Figure 5.16 Single cycle step function

5.5.2 Step Execution

There are four kinds of step execution function in HX-CODESYS.

Set some Break point is needed due to Step execution function execute program after stopping temporaly.

(1) Step Into function

Step into function execute every one step of function or function block. If this function is to step into at the SUB_A (the line that will be executed next), this function move to the first line of SUB_A, stepping into the call.

Click [Debug]-[Step Into] menu to execute Step in.

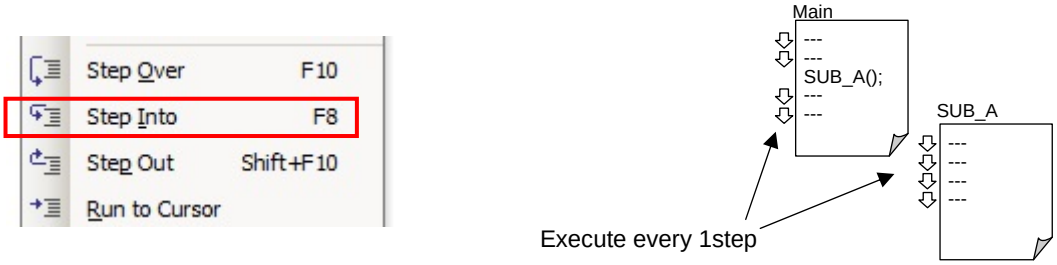


Figure 5.17 Step into function

(2) Step Over function

Step over function execute function or function block at once. If this function is to step into at the SUB_A (the line that will be executed next), this function move to the next SUB_A, stepping over the call.

Click [Debug]-[Step Over] to execute Step over.

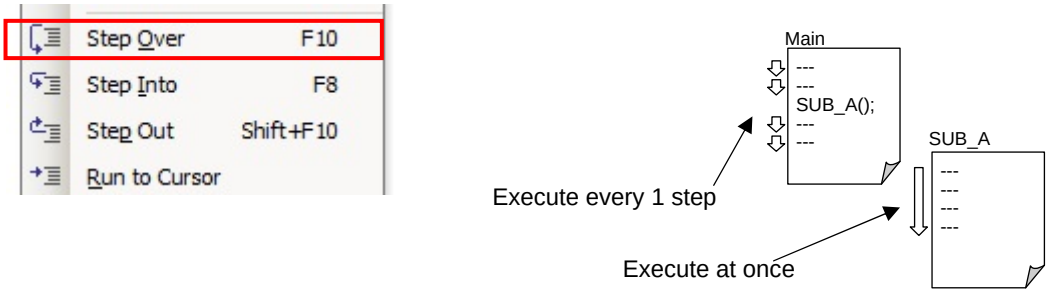


Figure 5.18 Step over function

(3) Step Out function

If Step out function execute during the execution of function or function block by Step into function, execute all program of function or function block and go through original program from function or function block.
Execute Step-in by clicking [Debug]-[Step Out].

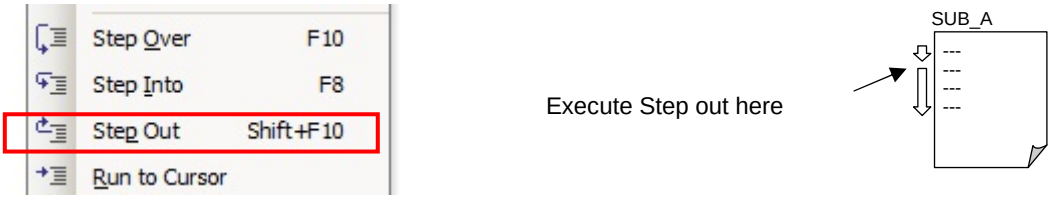


Figure 5.19 Step out function

(4) Run to cursor

Run to Cursor execute program until position of cursor on the line.

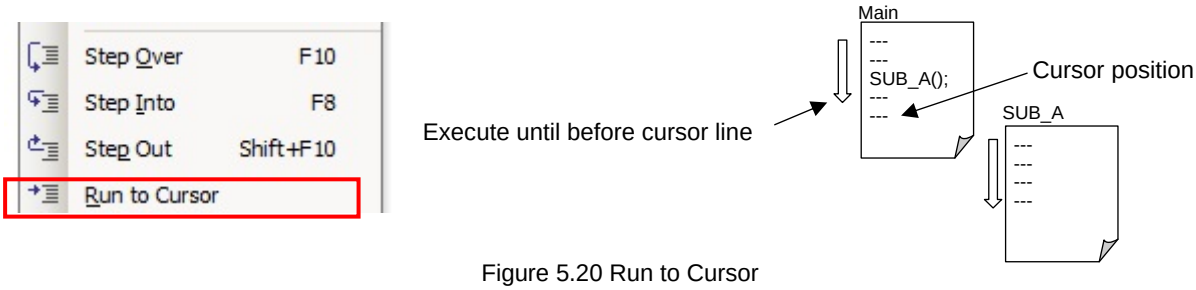


Figure 5.20 Run to Cursor

5.6 Force Values and Write Values

Variable value can be forced to a certain value. In force values, the value is set every cycle. In write values, the value is set once.

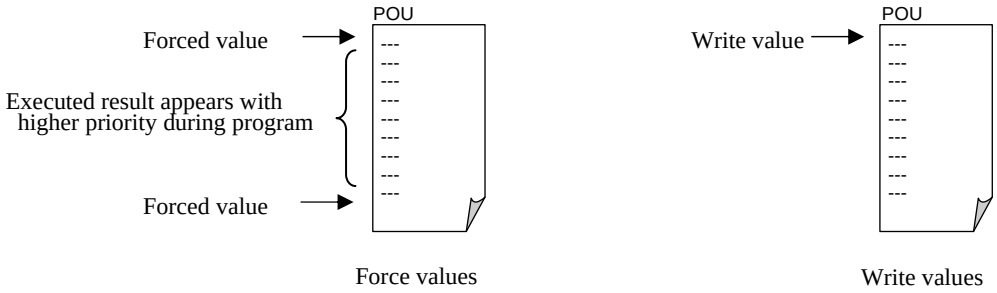
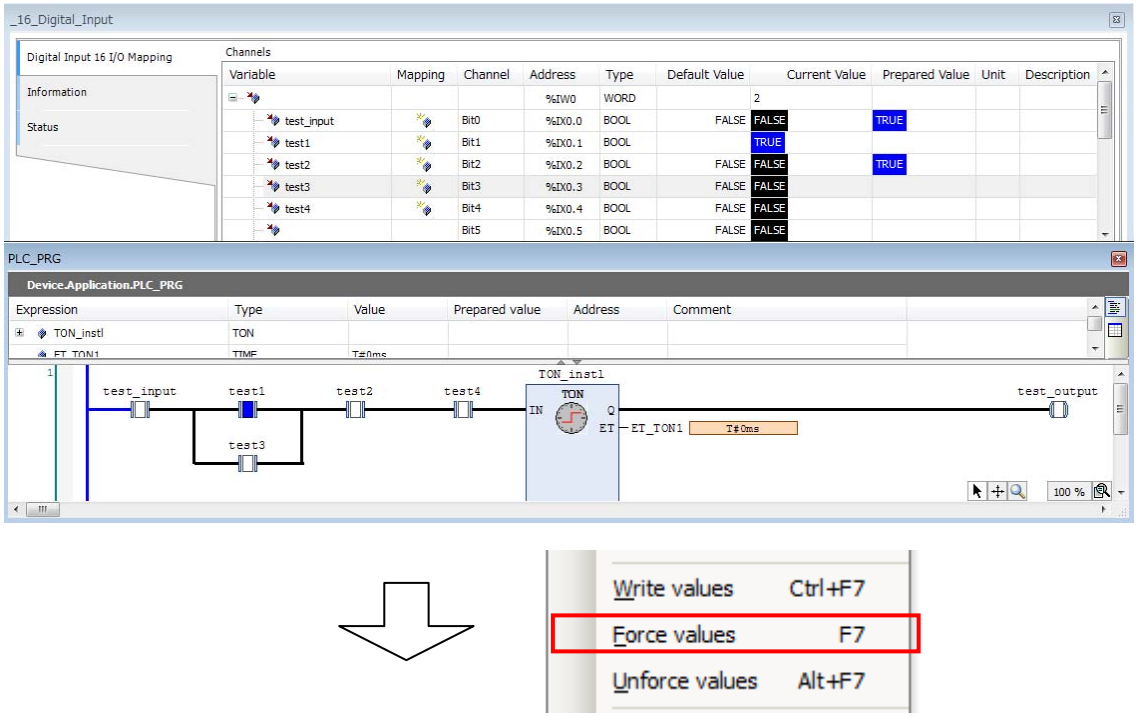


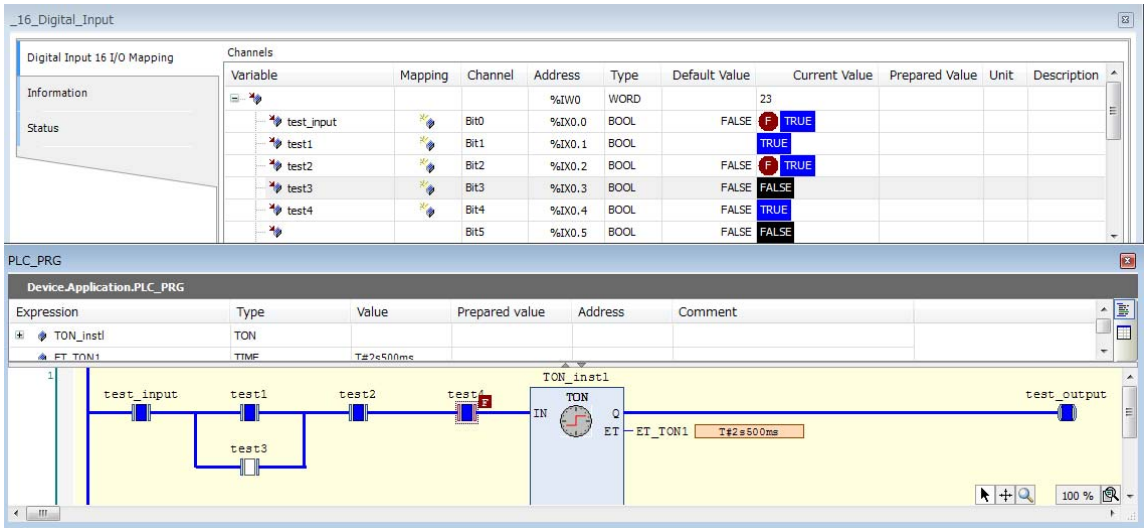
Figure 5.21 Force values and Write values function

⚠ Caution
There is some risk like malfunction of device, danger of human depends on target system caused by use of Value force set and write value function. Please test enough and confirm stable operation of target system with value of variable to use for force values and write values.

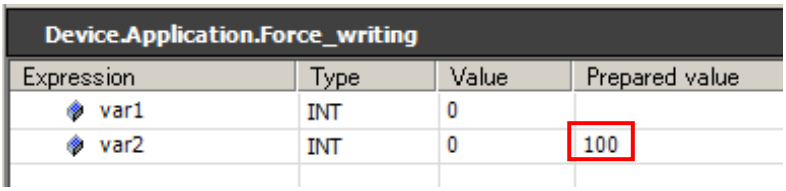
Configuration method of force function

Set write value of variable by double clicking [Prepared value] of variable declaration part.





It is possible to set value for variable similar way.



Display value with “<>” at value monitor part of editor part by setting value.

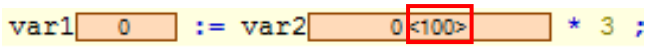


Figure 5.22 method of force function

Configuration method of Write value function

Click [Debug]-[Write value] after setting write value of variable with similar way of Value force set function.

The value is set only once when starting user program execution.

5.7 Trace

Trace function samples variable without dedicated program.

Configuration method

Create trace object

Add trace object by clicking [Add Object]-[Trace] after right clicking [Application].

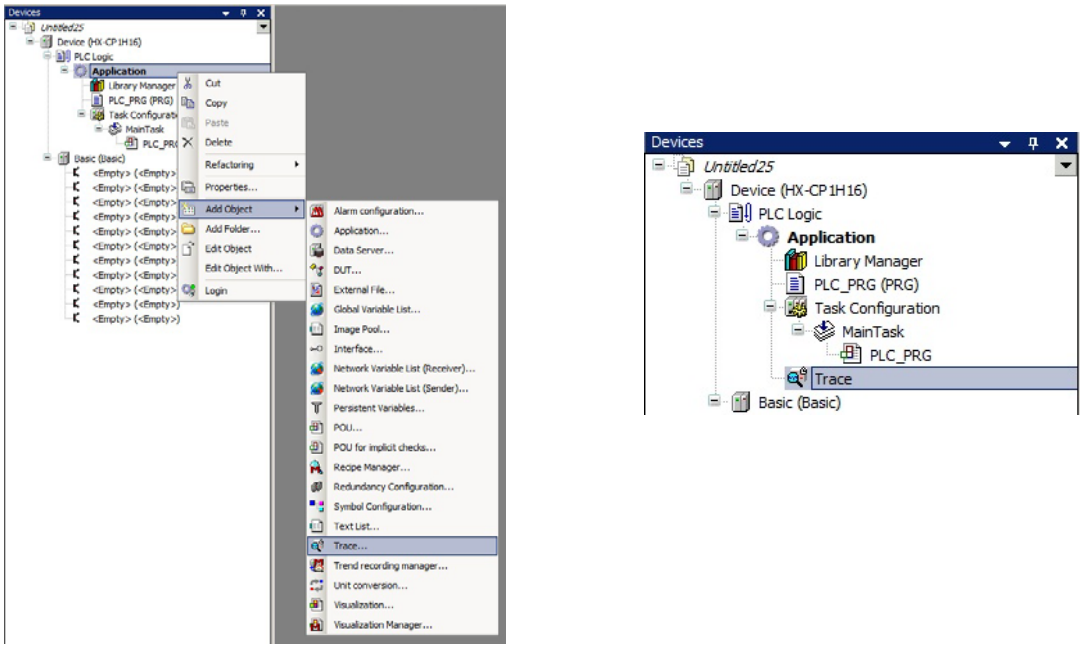


Figure 5.23 Create trace object

Trace editor appears by double clicking [Trace] object.

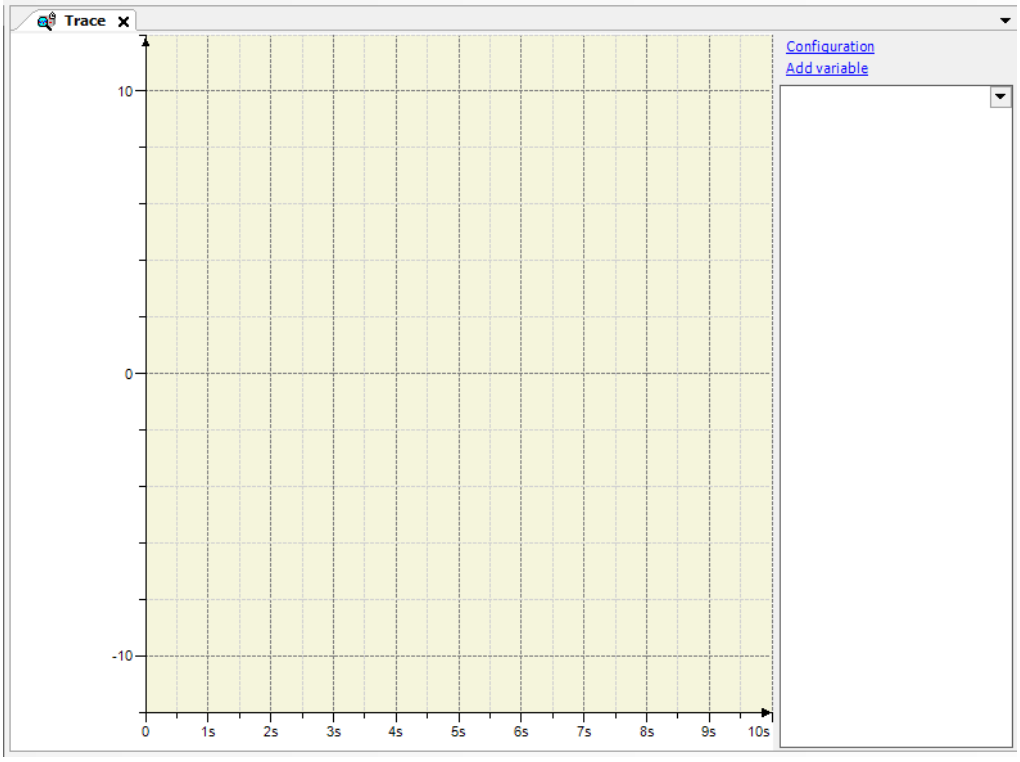


Figure 5.24 Trace editor

Configure [Task] by clicking [Configuration]. Trace sampling cycle is defined by this task configuration. Configure other items if it is needed. It is recommended the trace task priority is configured lower due to avoid impacting the other process of user program when trace task load is heavy.

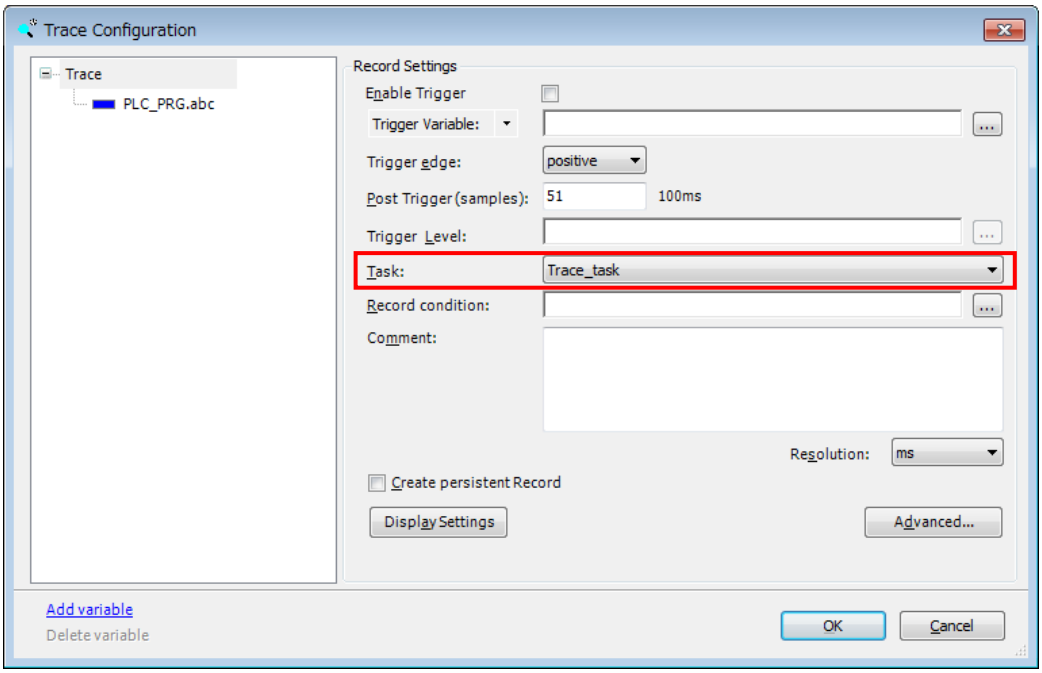


Figure 5.25 Configure trace task

Configure trace variable at [Variable] by clicking [Add variable]. Configure other items if it is needed. In case of adding new variable, configure by clicking [Add variable] with similar way.

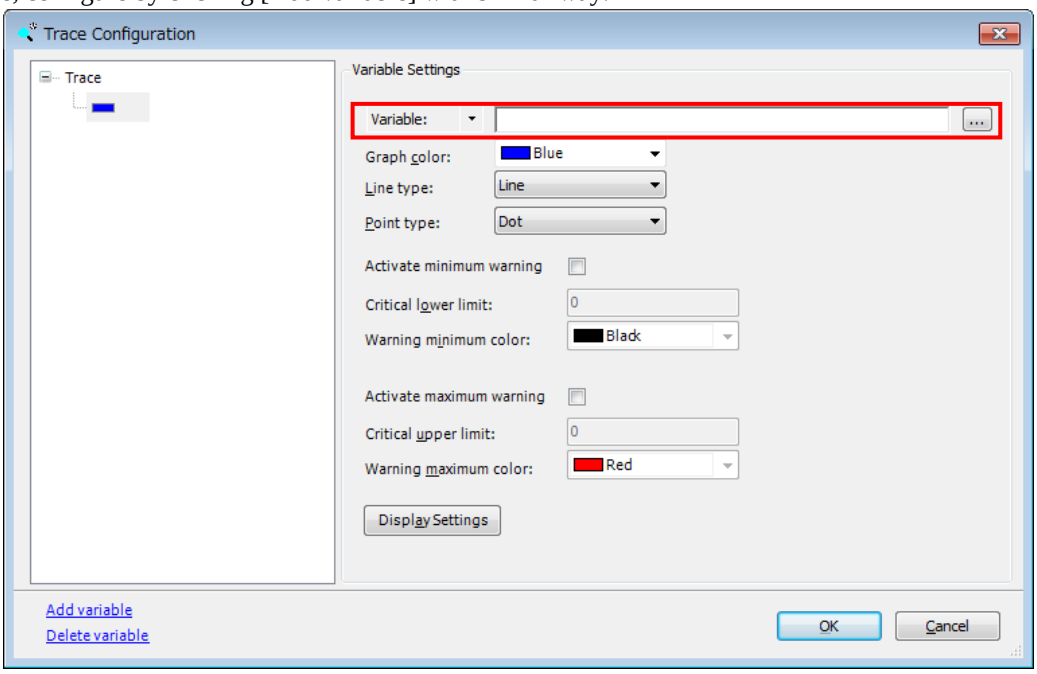


Figure 5.26 Add variable

Download trace data to HX-CPU by clicking [Trace]-[Download Trace] menu after login.

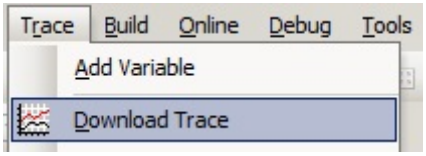


Figure 5.27 Download Trace

Display starts immediately according to the configured contents.

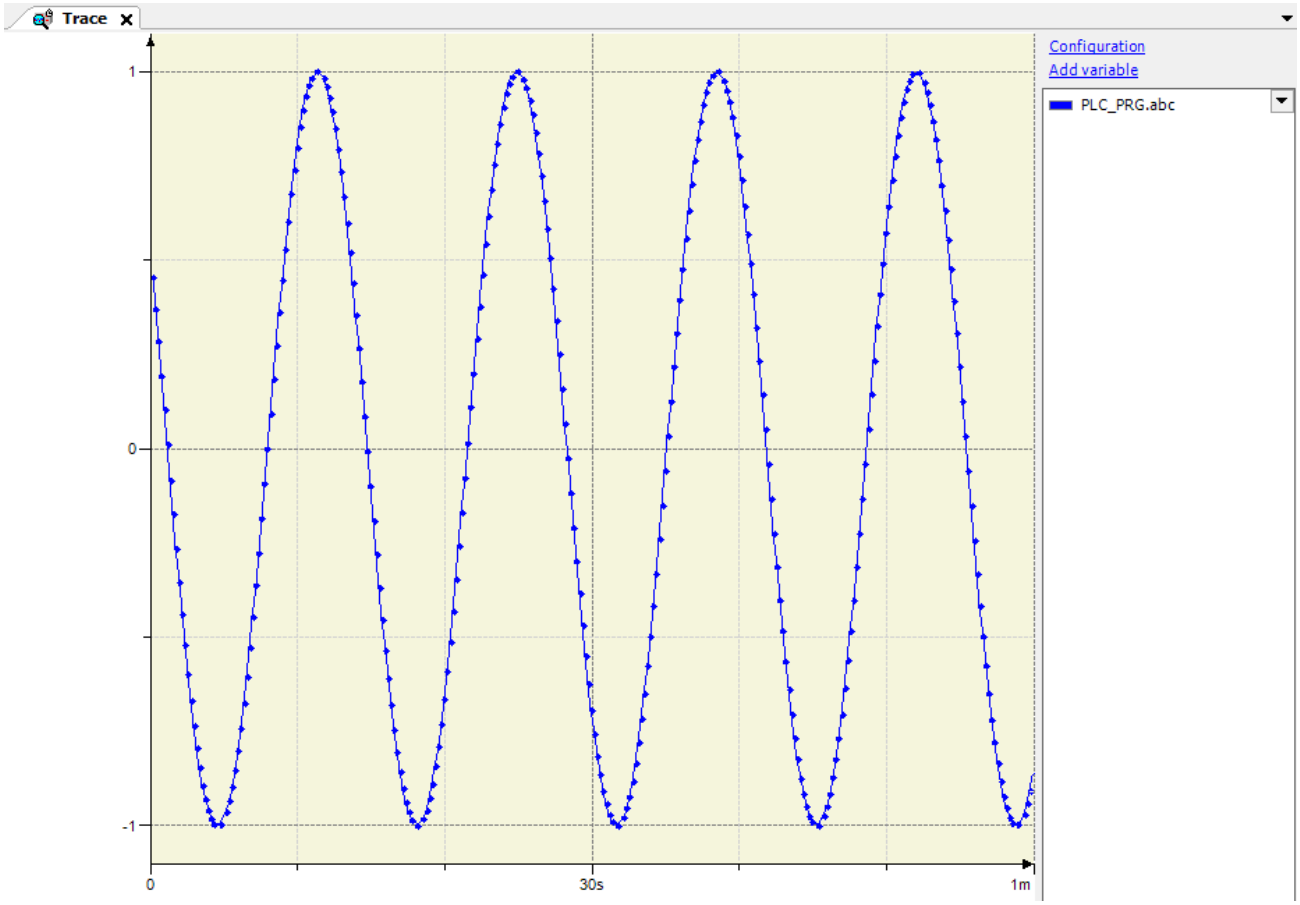


Figure 5.28 Trace data display

Following operation is possible after displaying trace data on [Trace] menu. Please refer online help for further detail information.

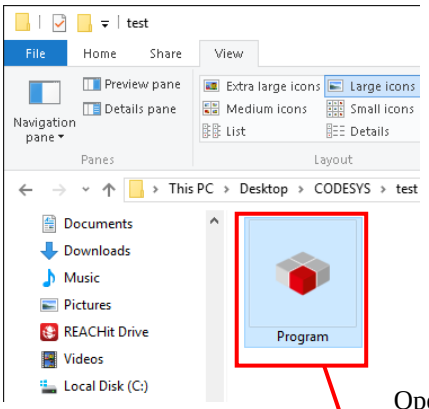
-Start/Stop of Trace data
-Customize display graph
-Access Trace data of HX-CPU
-Save/Read of Trace data

Appendix 1 Known Restrictions

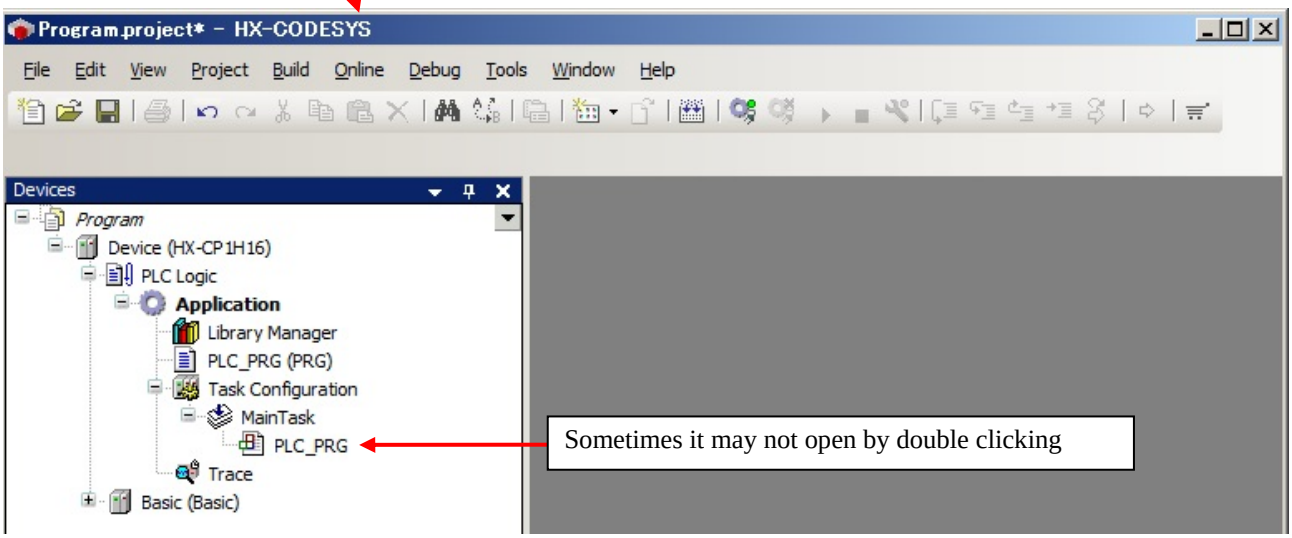
Following restrictions has been recognized with HX-CODESYS V3.5 SP8 Patch4. These are depend on the based software CODESYS V3.5 SP8 Patch4 from 3S-Smart Software Solutions, and will be improved later version.

[Open project]

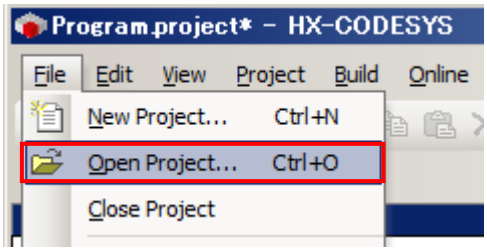
When open project file by specifying file from saved project file, sometimes POU can not be opened.



Open



This can be avoided and Project can be opened by [File]-[Open Project] after invoking HX-CODESYS.



[Leap year]

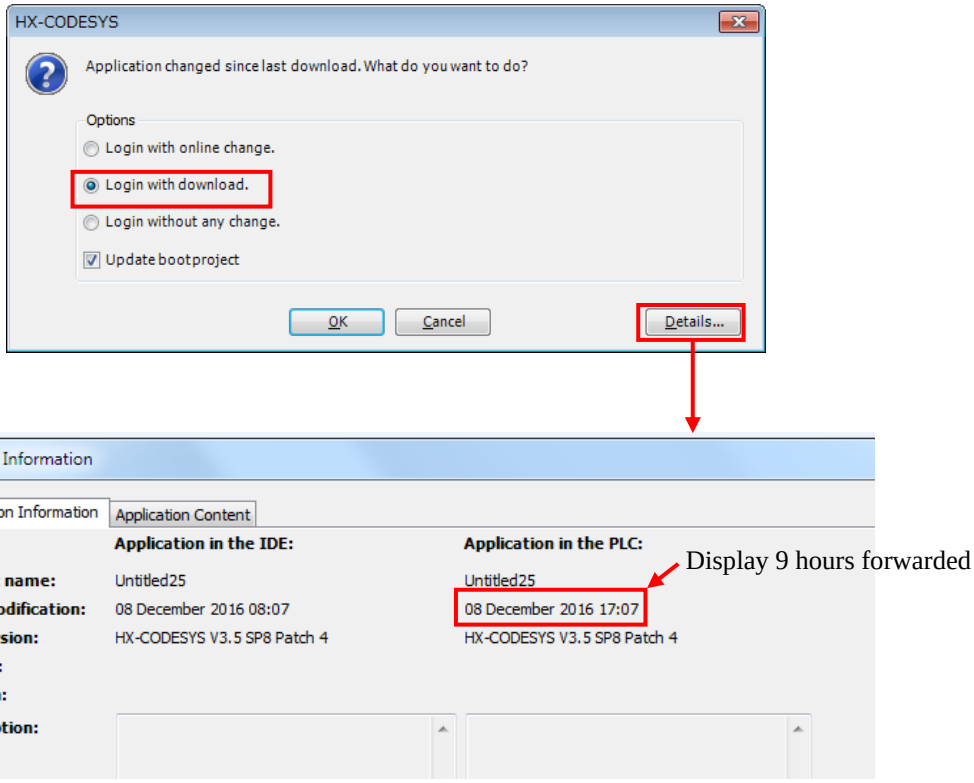
Getting day information command DTU.GetDayOfWeek() can't get correct information on February 29th (leap year) and day information is [7].

```
DOW1  SUNDAY    :=DTU.GetDayOfWeek(dtDate:=D#2016-2-28, peE
DOW2  7         :=DTU.GetDayOfWeek(dtDate:=D#2016-2-29, peE
DOW3  TUESDAY   :=DTU.GetDayOfWeek(dtDate:=D#2016-3-1, peEr
DOW4  WEDNESDAY :=DTU.GetDayOfWeek(dtDate:=D#2016-3-2, peEr
DOW5  THURSDAY  :=DTU.GetDayOfWeek(dtDate:=D#2016-3-3, peEr
DOW6  FRIDAY    :=DTU.GetDayOfWeek(dtDate:=D#2016-3-4, peEr
DOW7  SATURDAY  :=DTU.GetDayOfWeek(dtDate:=D#2016-3-5, peEr
DOW8  SUNDAY    :=DTU.GetDayOfWeek(dtDate:=D#2016-3-6, peEr
DOW9  MONDAY    :=DTU.GetDayOfWeek(dtDate:=D#2016-3-7, peEr
```

[Application information]

It is possible to confirm PLC project and PLC project application information at login, however latest updated day and time of PLC application is forwarded 9 hours if previous download is done by [Login with download]. This is no problem on execution.

*Latest updating day is displayed correctly, when previous download done by [Login with online change].

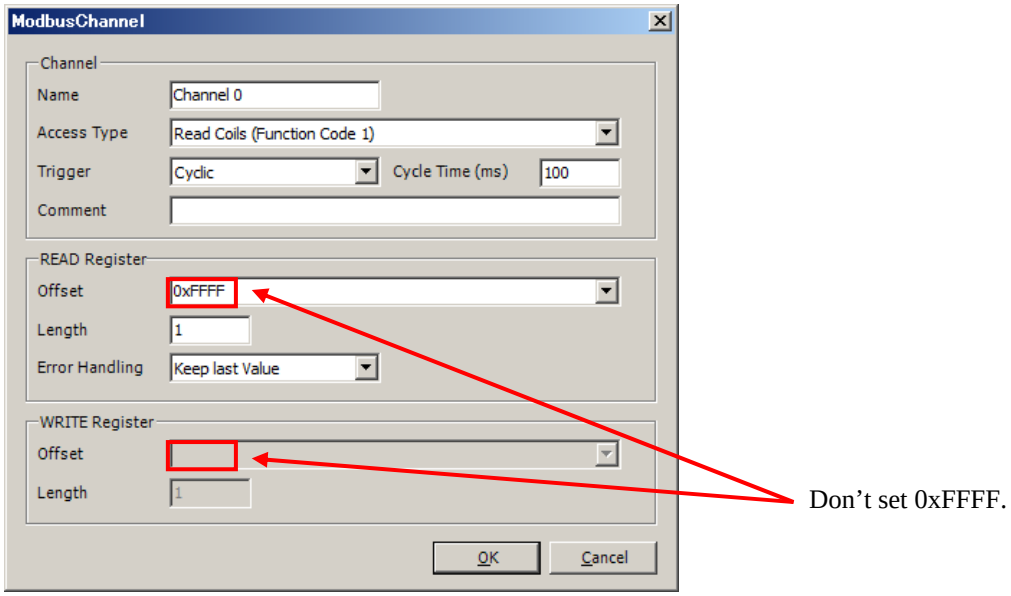


[Modbus-RTU Master]

[Modbus channel offset]

Don’t set “0xFFFF” for ofset value of Slave Modbus channel, when using Modbus-RTU master.

And don't set combination values of offset and lengths exceeding “0xFFFF” for ofset value of Slave Modbus channel.(Example Offset:0xFFFE, Length: 2)



[Modbus-RTU / TCP Slave]

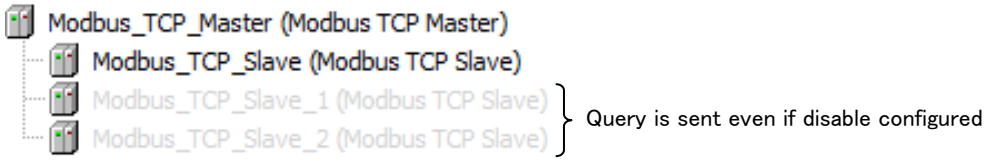
[Disable of Device]

Query will be sent even if Modbus-RTU / TCP slave device is disabled. Therefore, slave function block error(*) is detected due to receive timeout slave is occurred.

* In case of Modbus-RTU: ModbusSlaveComPort_Diag
  In case of Modbus-TCP: ModbusTCPSlave_Diag

In case of Modbus-RTU

In case of Modbus-TCP



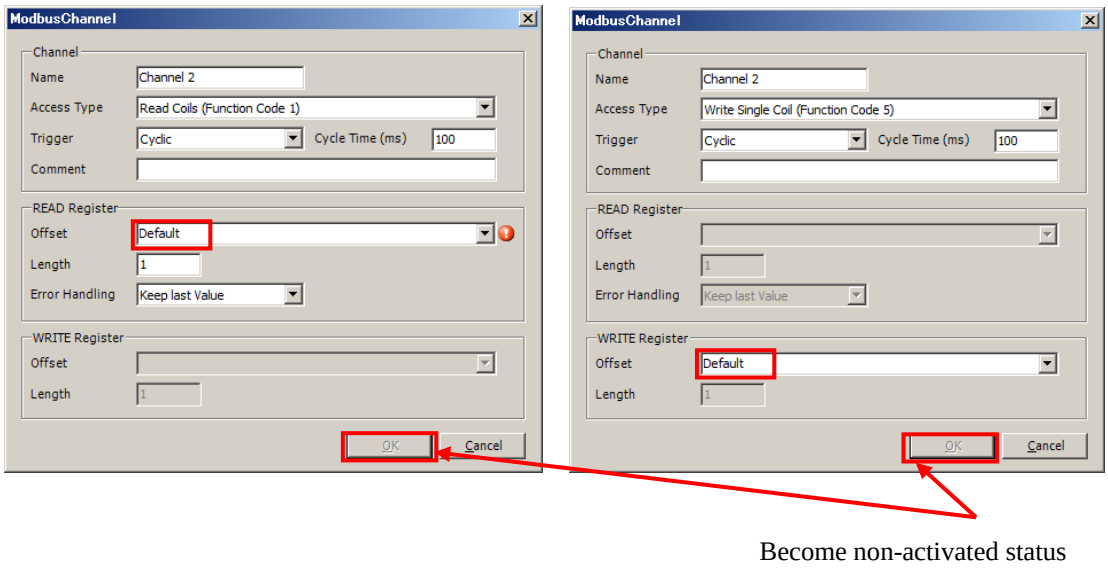
[Modbus-TCP Master]

[Channel configuration]

OK button becomes non-activated if specific offset address is used for slave channel registration at Modbus-TCP master.

Specific offset address: 0x0001 / 0x0005 / 0x03E8

This can be avoided by pressing Enter instead of OK clicking.



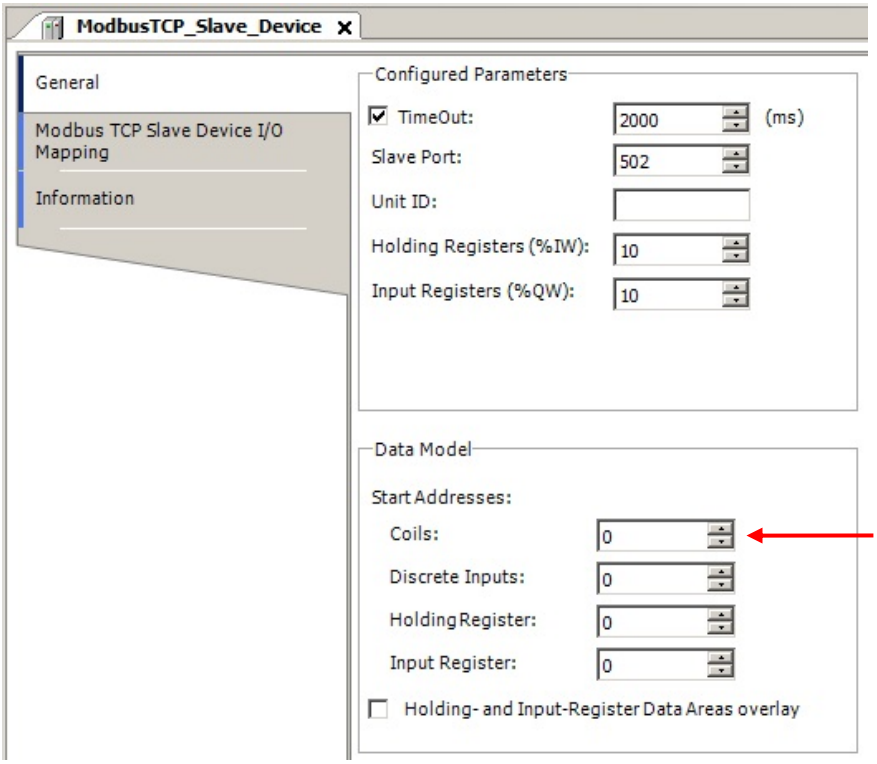
[Modbus-TCP Slave]

[Device function code 15 (Write multiple coils)]

Configure number of coil is 8 integral multiples when write data using Function code 15 (Write multiple coils) from external Modbus-TCP master and HX-CPU is used as Modbus-TCP slave. Operation is not properly if configure is not 8 integral multiples.

[Start address of Coil]

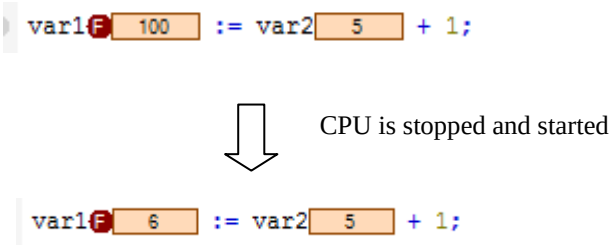
Configure start address of coil is 16 integral multiples when it is not specified 0(zero) and HX-CPU is used as Modbus-TCP slave. Operation of Function code 5 (write single coil) is not properly if other value is set.



Set 16 integral multiples value

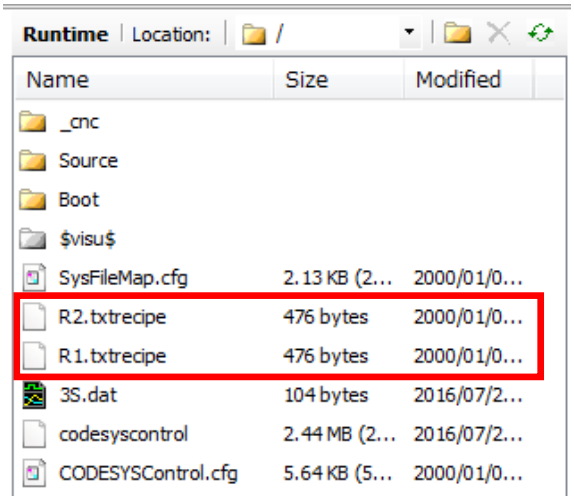
[Force values]

If CPU is stopped by RUN / STOP switch while variables are forced, the variables are automatically unforced because stop position of RUN / STOP switch is [Reset warm]. If HX-CODESYS is connected with online in this case, forced status continues to display. Choose [Debug]-[Unforce values] to release force status.



[Reset origin]

When the project included Recipe Manager is downloaded to HX-CPU, Several files about Recipe Manager are created but these files are not deleted by Reset origin. Please delete each file by [Device]-[Files] tab.



[Visualization]

[Text editor] in visualization function is not supported. Please do not use it. If you use it HX-CPU may operate incorrectly.

[Online change]

Do not make online change with changing FB body and without changing its instance.
If you do that, error dialog appears. Click [Ignore] to continue.

Example: FB is replaced, but instance is not changed as follows.

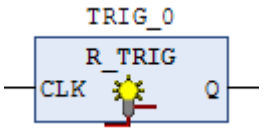
In Structured Text

Before: `TRIG_0 : R_TRIG;`

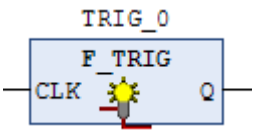
After: `TRIG_0 : F_TRIG;`

In LD/FBD

Before



After

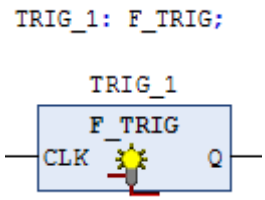


If it is necessary to change like this, choose download (CPU stops) instead of online change. If it's not possible to stop CPU, add new instance as follows.

In Structured Text

After: `TRIG_1 : F_TRIG;`

In LD/FBD



In addition to R_TRIG and F_TRIG, this restriction is applied to TON and TOF also.

Besides FB, it is applied to STRUCT too. Do not make online change with changing STRUCT name and without changing variable name of STRUCT.

Before: `test_STR : STR_A;`

After: `test_STR : STR_B;`

If it is necessary to change like this, choose download (CPU stops) instead of online change. If it's not possible to stop CPU, add new variable as follows.

Before: `test_STR : STR_A;`

After: `test_STR2 : STR_B;`

MEMO

Appendix 2　Open Source Software (OSS) List

This product is using open source software (hereinafter OSS).

Information about OSS used for this product is as follows. After checking the contents, please use product.

[GNU GPL application software]

We are using the following open source software which is an applicable object of the version after GNU General Public License Version 1, February 1989 or later version (hereinafter GPL) according to a condition of GPL. We disclose the source code of open source software included in this product. We will supply the media to the person who hopes for copy, modify and distribute open source software. However please understand the following points in advance.

- We can not answer a question about the contents of source cord at all.
- We do not guarantee program which is made by offered source cord at all.
- When requesting a media, a cost sometimes occurs.
- Supply period of source code shall be either the period specified in the license or during the production period whichever is longer.

[GPLv1 application software package list]

perl-5.22.1

The following conditions are applied to GPLv1 application software.

It's possible to refer to the following Web site.

https://www.gnu.org/licenses/gpl-1.0.html

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies.　By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.

When we speak of free software, we are referring to freedom, not price.　Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original authors' reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

   a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and
   b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).
   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General
   Public License.
   d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

   a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

   b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,

c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard
libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License.
Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public
License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free
Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your
program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

[GPLv2 application software package list]

attr-2.4.47

base-files-3.0.14

base-passwd-3.5.29

busybox-1.24.1

ethtool-4.2

eudev-3.1.5

glibc-2.23

init-ifupdown-1.0

initscripts-1.0

iptables-1.6.0

kernel-3.14.26

kmod-22+git0+42f32b8ae4

netbase-5.3

opkg-utils-0.1.8+git0+53274f0875

procos-3.3.11

sysvinit-2.88dsf

sysvinit-inittab-2.88dsf

update-rc.d-0.7

util-linux-2.27.1

vsftpd-3.0.3

The following conditions are applied to GPLv2 application software.

It's possible to refer to the following Web site.

https://www.gnu.org/licenses/gpl-2.0.html

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change. b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License. c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)
   These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on

the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or, c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution

system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
one line to give the program's name and an idea of what it does.
Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:
Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:
Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision'
(which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

[GPLv3 application software package list]

bash-4.3.30

coreutils-8.25

gcc-runtime-5.3.0

lftp-4.6.3a

libdn-1.32

libgcc-5.3.0

readline-6.3

The following conditions are applied to GPLv3 application software.

It's possible to refer to the following Web site.

https://www.gnu.org/licenses/gpl-3.0.html

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding

Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a

country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR

IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

[GNU LGPL application software]

We are using the following open source software which is an applicable object of the version after GNU LESSER General Public License Version 2.1, February 1999 or later version (hereinafter LGPL) according to a condition of LGPL. We disclose the source code of open source software included in this product. We will supply the media to the person who hopes for copy, modify and distribute open source software. However please understand the following points in advance.

- We can not answer a question about the contents of source cord at all.
- We do not guarantee program which is made by offered source cord at all.
- When requesting a media, a cost sometimes occurs.
- Supply period of source code shall be either the period specified in the license or during the production period whichever is longer.

[LGPLv2.1 application software package list]

attr-2.4.47

crackklib-2.9.5

glibc-2.23

kmod-22+git0+42f32b8ae4

libdn-1.32

procos-3.3.11

util-linux-2.27.1

The following conditions are applied to LGPLv2.1 application software.

It's possible to refer to the following Web site.

https://www.gnu.org/licenses/lgpl-2.1.html

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999
Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
   a) The modified work must itself be a software library.
   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.
   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:
   a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
   b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
one line to give the library's name and an idea of what it does.
Copyright (C) year name of author

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:
Yoyodyne, Inc., hereby disclaims all copyright interest in
the library `Frob' (a library for tweaking knobs) written
by James Random Hacker.

signature of Ty Coon, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

[LGPLv3 application software package list]

gmp-6.1.0

The following conditions are applied to LGPLv3 application software.

It's possible to refer to the following Web site.

https://www.gnu.org/licenses/lgpl-3.0.html

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
d) Do one of the following: ◦0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
◦1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

[BSD application software]

We are using the following open source software which is an applicable BSD according to a condition of copyright holder.

[BSD application software package list]

libpam-1.2.1

libcap-2.24

tcl-8.6.4

tcp-wrappers-7.6

util-linux-2.27.1

The following conditions are applied to BSD application software.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

[Artistic application software]

We are using the following open source software which is an applicable Artistic according to a condition of Artistic.

[Artistic application software package list]

perl-5.22.1

The following conditions are applied to Artistic application software.

The "Artistic License"

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

"Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

"Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

"Copyright Holder" is whoever is named in the copyright or copyrights for the package.

"You" is you, if you're thinking about copying or distributing this Package.

"Reasonable copying fee" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

"Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

    a) place your modifications in the Public Domain or otherwise make them freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

    b) use the modified Package only within your corporation or organization.

c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

b) accompany the distribution with the machine-readable source of the Package with your modifications.

c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this package. You may charge any fee you choose for support of this package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own. You may embed this Package's interpreter within an executable of yours (by linking); this shall be construed as a mere form of aggregation, provided that the complete Standard Version of the interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whoever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called "undump" or "unexec" methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. C subroutines (or comparably compiled subroutines in other languages) supplied by you and linked into this Package in order to emulate subroutines and variables of the language defined by this Package shall not be considered part of this Package, but are the equivalent of input as in Paragraph 6, provided these subroutines do not change the language in any way that would cause it to fail the regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

[About glibc-2.23]

We are using the following open source software which is an applicable glibc according to a condition of copyright holder.

The following conditions and GNU GPL/GNU LGPL are applied to glibc-2.23 application software.

This file contains the copying permission notices for various files in theGNU C Library distribution that have copyright owners other than the FreeSoftware Foundation. These notices all require that a copy of the noticebe included in the accompanying documentation and be distributed with binary distributions of the code, so be sure to include this file along with any binary distributions derived from the GNU C Library.

All code incorporated from 4.4 BSD is distributed under the followinglicense:

Copyright (C) 1991 Regents of the University of California.All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. [This condition was removed.]
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The DNS resolver code, taken from BIND 4.9.5, is copyrighted by UC Berkeley, by Digital Equipment Corporation and by Internet Software Consortium. The DEC portions are under the following license:

Portions Copyright (C) 1993 by Digital Equipment Corporation.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies, and
that the name of Digital Equipment Corporation not be used in advertising or publicity pertaining to distribution of the document or software without specific, written prior permission.

THE SOFTWARE IS PROVIDED ``AS IS'' AND DIGITAL EQUIPMENT CORP. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL DIGITAL EQUIPMENT CORPORATION BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The ISC portions are under the following license:

Portions Copyright (c) 1996-1999 by Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The Sun RPC support (from rpcsrc-4.0) is covered by the following license:

Copyright (c) 2010, Oracle America, Inc.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the "Oracle America, Inc." nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following CMU license covers some of the support code for Mach, derived from Mach 3.0:

Mach Operating System
Copyright (C) 1991,1990,1989 Carnegie Mellon University All Rights Reserved.

Permission to use, copy, modify and distribute this software and its documentation is hereby granted, provided that both the copyright notice and this permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS ``AS IS" CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Carnegie Mellon requests users of this software to return to

Software Distribution Coordinator
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3890

or Software.Distribution@CS.CMU.EDU any improvements or extensions that they make and grant Carnegie Mellon the rights to redistribute these changes.

The file if_ppp.h is under the following CMU license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY CARNEGIE MELLON UNIVERSITY AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE UNIVERSITY OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The following license covers the files from Intel's "Highly Optimized Mathematical Functions for Itanium" collection:

Intel License Agreement

Copyright (c) 2000, Intel Corporation

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* The name of Intel Corporation may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTEL OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The files inet/getnameinfo.c and sysdeps/posix/getaddrinfo.c are copyright (C) by Craig Metz and are distributed under the following license:

/* The Inner Net License, Version 2.00

The author(s) grant permission for redistribution and use in source and binary forms, with or without modification, of the software and documentation provided that the following conditions are met:

0. If you receive a version of the software that is specifically labelled as not being for redistribution
(check the version message and/or README), you are not permitted to redistribute that version of the software in any way or form.
1. All terms of the all other applicable copyrights and licenses must be followed.
2. Redistributions of source code must retain the authors' copyright notice(s), this list of
conditions, and the following disclaimer.
3. Redistributions in binary form must reproduce the authors' copyright notice(s), this list
of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
4. [The copyright holder has authorized the removal of this clause.]
5. Neither the name(s) of the author(s) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY ITS AUTHORS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

If these license terms cause you a real problem, contact the author. */

The file sunrpc/des_impl.c is copyright Eric Young:

Copyright (C) 1992 Eric Young
Collected from libdes and modified for SECURE RPC by Martin Kuck 1994
This file is distributed under the terms of the GNU Lesser General Public License, version 2.1 or later - see the file COPYING.LIB for details. If you did not receive a copy of the license with this program, please see <http://www.gnu.org/licenses/> to obtain a copy.

The libidn code is copyright Simon Josefsson, with portions copyright
The Internet Society, Tom Tromey and Red Hat, Inc.:

Copyright (C) 2002, 2003, 2004, 2011 Simon Josefsson

This file is part of GNU Libidn.

GNU Libidn is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

GNU Libidn is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with GNU Libidn; if not, see <http://www.gnu.org/licenses/>.

The following notice applies to portions of libidn/nfkc.c:

This file contains functions from GLIB, including gutf8.c and gunidecomp.c, all licensed under LGPL and copyright hold by:

Copyright (C) 1999, 2000 Tom Tromey
Copyright 2000 Red Hat, Inc.

The following applies to portions of libidn/punycode.c and libidn/punycode.h:

This file is derived from RFC 3492bis written by Adam M. Costello.

Disclaimer and license: Regarding this entire document or any portion of it (including the pseudocode and C code), the author makes no guarantees and is not responsible for any damage resultingfrom its use. The author grants irrevocable permission to anyone to use, modify, and distribute it in any way that does not diminish the rights of anyone else to use, modify, and distribute it, provided that redistributed derivative works do not contain misleading author or version information. Derivative works need not be licensed under similar terms.

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The file inet/rcmd.c is under a UCB copyright and the following:

Copyright (C) 1998 WIDE Project.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the project nor the names of its contributors
   may be used to endorse or promote products derived from this software
   without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE

FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The file posix/runtests.c is copyright Tom Lord:

Copyright 1995 by Tom Lord

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in supporting documentation, and that the name of the copyright holder not be used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission.

Tom Lord DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL TOM LORD BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The posix/rxspencer tests are copyright Henry Spencer:

Copyright 1992, 1993, 1994, 1997 Henry Spencer. All rights reserved.
This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.

2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.

3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.

The file posix/PCRE.tests is copyright University of Cambridge:

Copyright (c) 1997-2003 University of Cambridge

Permission is granted to anyone to use this software for any purpose on any computer system, and to redistribute it freely, subject to the following restrictions:

1. This software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
   without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

2. The origin of this software must not be misrepresented, either by explicit claim or by omission.
   In practice, this means that if you use PCRE in software that you distribute to others, commercially or otherwise,
   you must put a sentence like this

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England.

somewhere reasonably visible in your documentation and in any relevant files or online help data or similar. A reference to the ftp site for the source, that is, to

ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/

should also be given in the documentation. However, this condition is not intended to apply to whole chains of software. If package A includes PCRE, it must acknowledge it, but if package B is software that includes package A, the condition is not imposed on package B (unless it uses PCRE independently).

3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software.

4. If PCRE is embedded in any software that is released under the GNU General Purpose Licence (GPL), or Lesser General Purpose Licence (LGPL), then the terms of that licence shall supersede any condition above with which it is incompatible.

Files from Sun fdlibm are copyright Sun Microsystems, Inc.:

Copyright (C) 1993 by Sun Microsystems, Inc. All rights reserved.

Developed at SunPro, a Sun Microsystems, Inc. business.
Permission to use, copy, modify, and distribute this software is freely granted, provided that this notice is preserved.

Part of stdio-common/tst-printf.c is copyright C E Chew:

(C) Copyright C E Chew

Feel free to copy, use and distribute this software provided:

1. you do not pretend that you wrote it
2. you leave this copyright notice intact.

Various long double libm functions are copyright Stephen L. Moshier:

Copyright 2001 by Stephen L. Moshier <moshier@na-net.ornl.gov>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, see <http://www.gnu.org/licenses/>. */

[About openssl-1.0.2g]

We are using the following open source software which is an applicable open ssl according to a condition of copyright holder.

The following conditions are applied to openssl-1.0.2g application software.

LICENSE ISSUES
==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

```
/* ====================================================================
 * Copyright (c) 1998-2016 The OpenSSL Project.    All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *       notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *       notice, this list of conditions and the following disclaimer in
 *       the documentation and/or other materials provided with the
 *       distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *       software must display the following acknowledgment:
 *       "This product includes software developed by the OpenSSL Project
 *       for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *       endorse or promote products derived from this software without
 *       prior written permission. For written permission, please contact
 *       openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *       nor may "OpenSSL" appear in their names without prior written
 *       permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *       acknowledgment:
 *       "This product includes software developed by the OpenSSL Project
 *       for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.    IN NO EVENT SHALL THE OpenSSL PROJECT OR

 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
```

```
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com).  This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

 Original SSLeay License
 -----------------------
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
```

```
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
```

[About libgcc-5.3.0/gcc-runtime-5.3.0]

We are using the following open source software which is an applicable libgcc and gcc-runtime according to a condition of copyright holder.

The following conditions and GNU GPL are applied to libgcc-5.3.0 and gcc-runtime-5.3.0 application software.

---

insert GPL v3 text here

GCC RUNTIME LIBRARY EXCEPTION
Version 3.1, 31 March 2009

General information:
http://www.gnu.org/licenses/gcc-exception.html
Copyright (C) 2009 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
This GCC Runtime Library Exception ("Exception") is an additional permission under section 7 of the GNU General Public License, version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a notice placed by the copyright holder of the file stating that the file is governed by GPLv3 along with this Exception.
When you use GCC to compile a program, GCC may combine portions of certain GCC header files and runtime libraries with the compiled program. The purpose of this Exception is to allow compilation of non-GPL (including proprietary) programs to use, in this way, the header files and runtime libraries covered by this Exception.

0. Definitions.
A file is an "Independent Module" if it either requires the Runtime Library for execution after a Compilation Process, or makes use of an interface provided by the Runtime Library, but is not otherwise based on the Runtime Library.
"GCC" means a version of the GNU Compiler Collection, with or without modifications, governed by version 3 (or a specified later version) of the GNU General Public License (GPL) with the option of using any subsequent versions published by the FSF.
"GPL-compatible Software" is software whose conditions of propagation, modification and use would permit combination with GCC in accord with the license of GCC.
"Target Code" refers to output from any compiler for a real or virtual target processor architecture, in executable form or suitable for input to an assembler, loader, linker and/or execution phase. Notwithstanding that, Target Code does not include data in any format that is used as a compiler intermediate representation, or used for producing a compiler intermediate representation.
The "Compilation Process" transforms code entirely represented in non-intermediate languages designed for human-written code, and/or in Java Virtual Machine byte code, into Target Code. Thus, for example, use of source code generators and preprocessors need not be considered part of the Compilation Process, since the Compilation Process can be understood as starting with the output of the generators or preprocessors.
A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-compatible software, or if it is done without using any work based on GCC. For example, using non-GPL-compatible Software to optimize any GCC intermediate representations would not qualify as an Eligible Compilation Process.

1. Grant of Additional Permission.
   You have permission to propagate a work of Target Code formed by combining the Runtime Library with Independent Modules, even if such propagation would otherwise violate the terms of GPLv3, provided that all Target Code was generated by Eligible Compilation Processes. You may then convey such a combination under terms of your choice, consistent with the licensing of the Independent Modules.

2. No Weakening of GCC Copyleft.
   The availability of this Exception does not imply any general presumption that third-party software is unaffected by the copyleft requirements of the license of GCC.

[About tcl-8.6.4]

We are using the following open source software which is an applicable tcl according to a condition of copyright holder.

The following conditions and BSD are applied to tcl-8.6.4 application software.

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, ActiveState Corporation and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement,
license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause
252.227-7014 (b) (3) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

[About zlib-1.2.8]

We are using the following open source software which is an applicable zlib according to a condition of copyright holder.

The following conditions are applied to zlib-1.2.8 application software.

---

zlib License

This software is provided `as-is`, without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

We disclose the source code of open source software included in this product. We will supply the media to the person who hopes for copy, modify and distribute open source software. However please understand the following points in advance.

- We can not answer a question about the contents of source cord at all.
- We do not guarantee program which is made by offered source cord at all.
- When requesting a media, a cost sometimes occurs.
- Supply period of source code shall be either the period specified in the license or during the production period whichever is longer.