HITACHI PROGRAMMABLE AUTOMATION CONTROLLER

HX Series

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

○ Repair

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

○ Ordering parts or asking questions

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

(1) Model

(2) Manufacturing number (MFG.NO.)

(3) Details of the malfunction

○ Reader of this manual

This manual is described for the following person.

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

Warning

(1) This manual may not be reproduced in its entirety or ant portion thereof without prior consent.

(2) The content of this document may be changed without notice.

(3) This document has been created with utmost care. However, if errors or questionable areas are found, please contact us.

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 programmable controller (hereinafter referred to as PLC).
  Damage to the equipment or accidents may occur due to failure of the PLC.
  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	2016.07	NJI-638(X)

Table of contents

MEMO

Chapter 1　Prepared HX-CODESYS

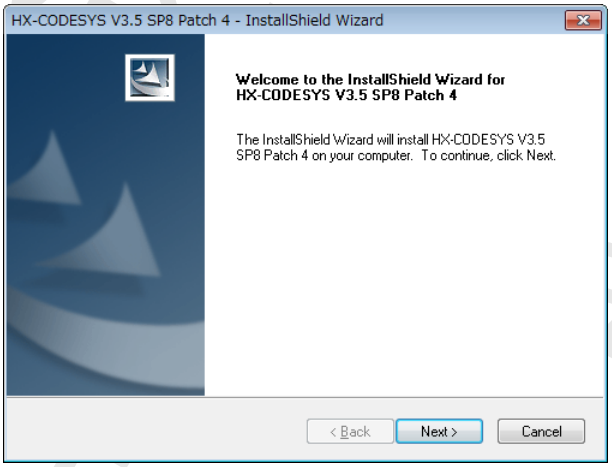
1.1 Installation

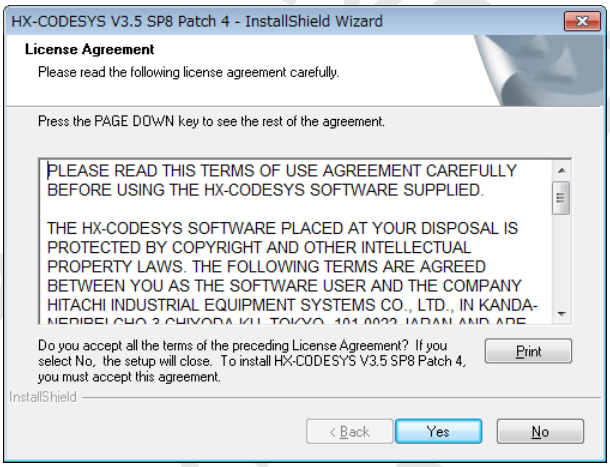
1.1.1 Installation of HX-CODESYS

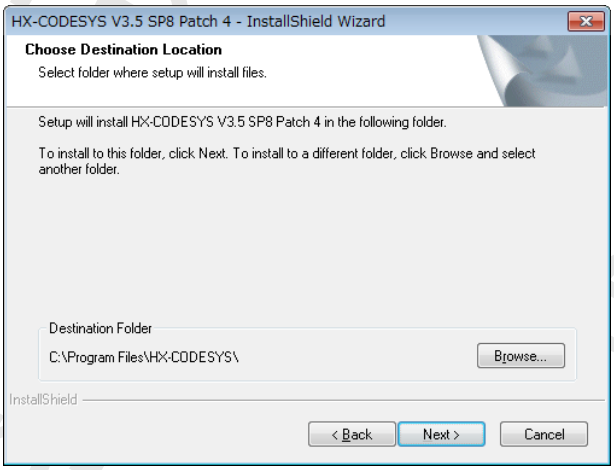
CODESYS のインストールを開始する前に、他の Windows アプリケーションを実行している場合は終了させてからインストールを開始してください。終了させない場合は正常にインストールできない場合があります。

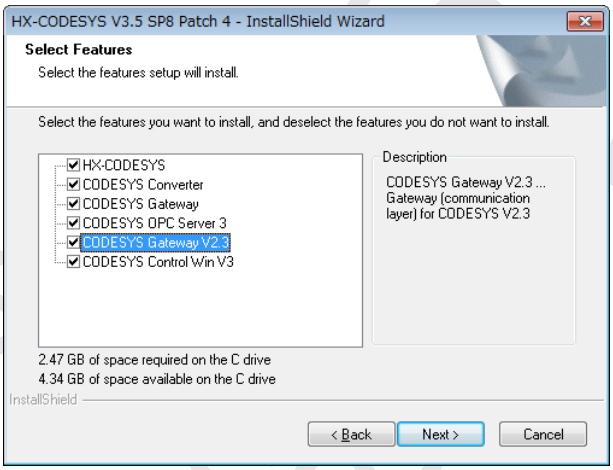
1. The installation wizard starts up automatically by double click “Setup_HX-CODESYSV35SP8Patch4.exe” on EHV-CODESYS installation CD.

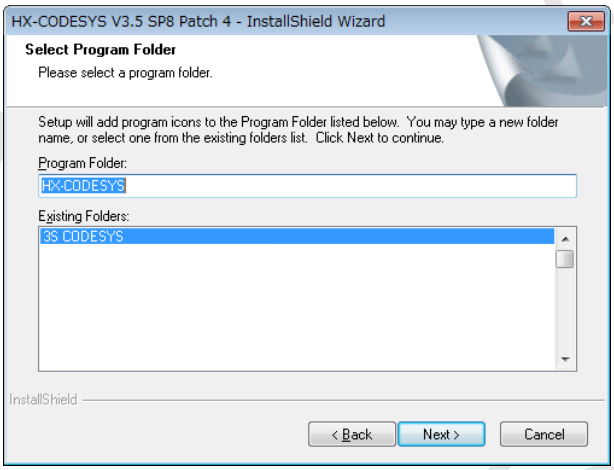
2. Follow the instructions

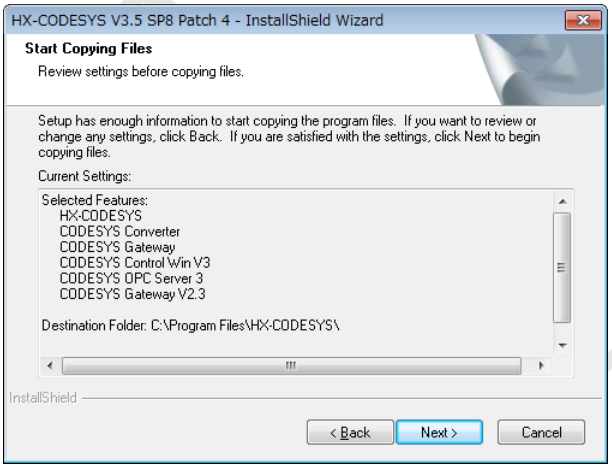


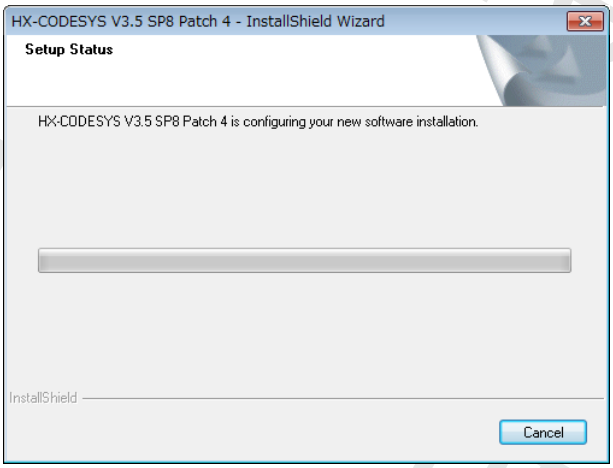


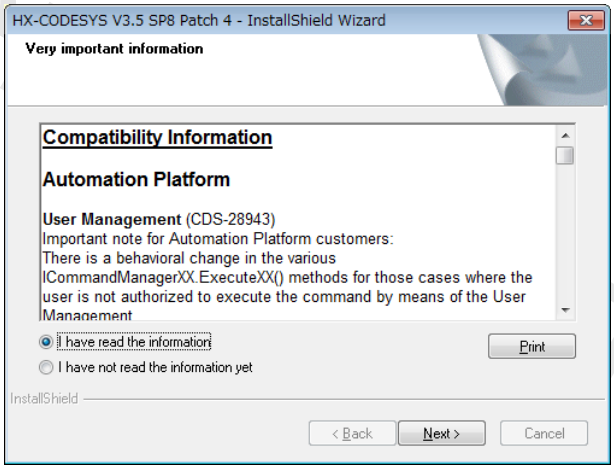


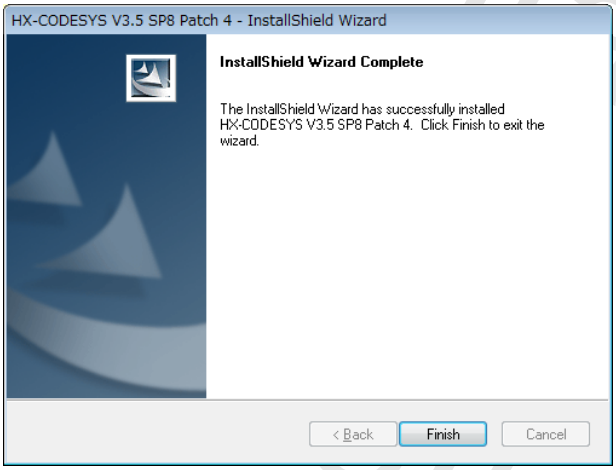








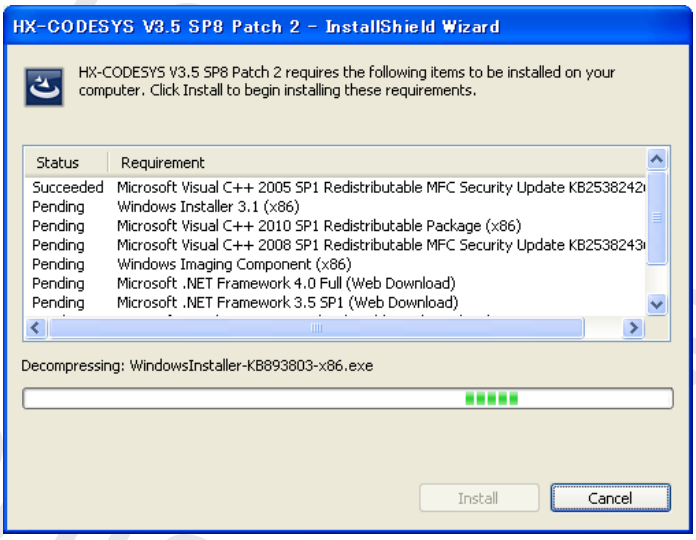




インストール完了まで 30 分程度の時間を要します。

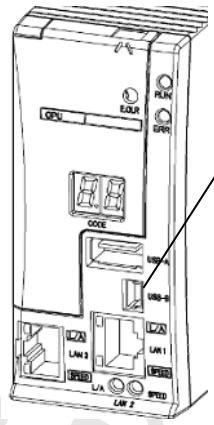
注 意

インストール先のパソコンには.NETFramework 4.0 が必要です（WindowsXP SP3 の場合）。これがインストールされていない場合には HX-CODESYS のインストールは停止し、右図の画面が表示されます。[Install]をクリックすると、.NET フレームワークをインターネットからダウンロードすることができます。パソコンがインターネットに接続していない場合は[Cancel]をクリックして、HX-CODESYS の CD-R からインストールしてください。



1.1.2 USB ドライバのインストール

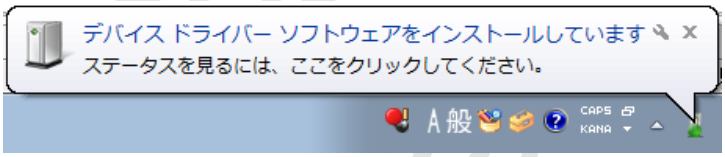
1. Plug in USB cable to CPU module.



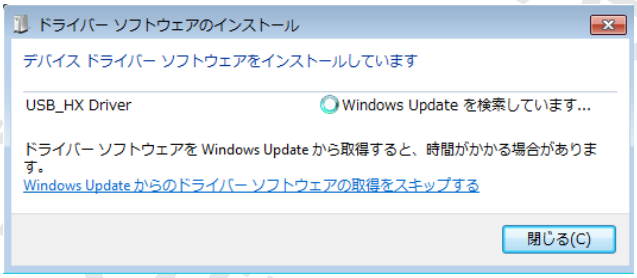
USB device port
(Type:miniB)

USB ケーブルは製品に同梱されておりません。
ノイズによる通信エラーを防止するため、フェライトコア付の
USB ケーブル(A – miniB)をお求め、ご使用ください。

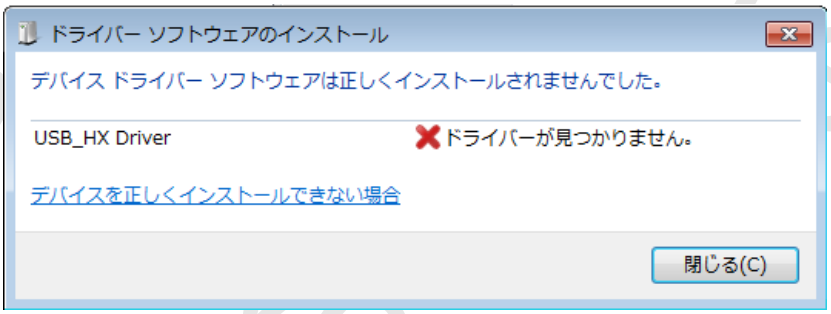
2. Popup window appears at right-bottom of screen. Click the popup window.



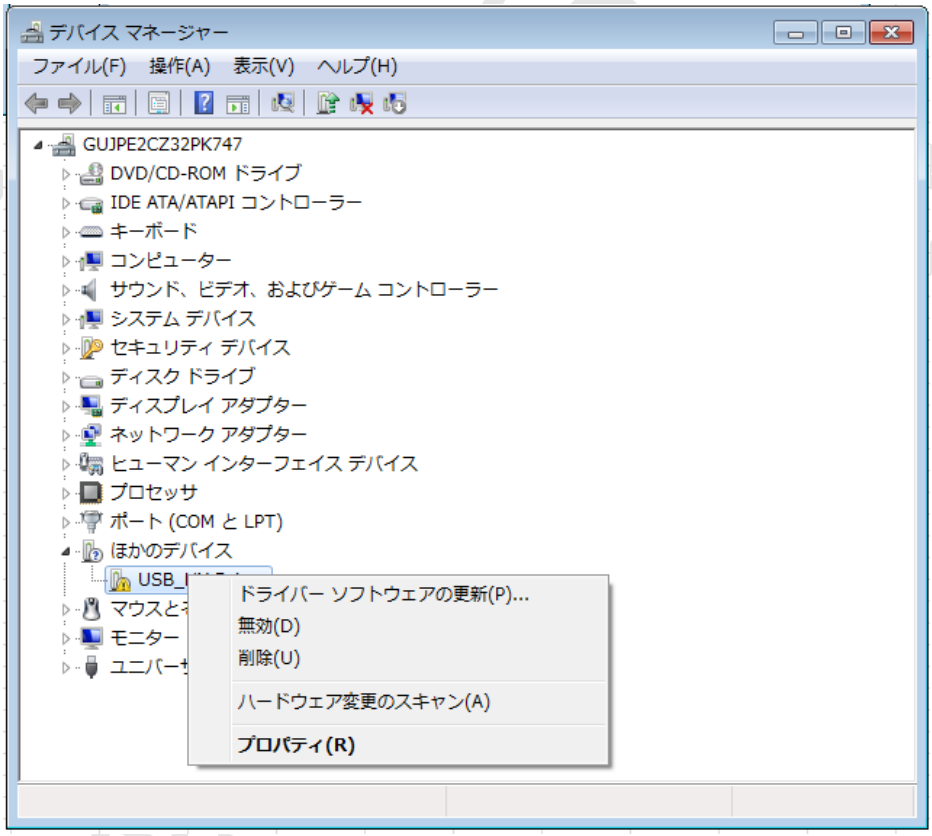
3. Windows Update からのドライバーソフトウェアの取得をスキップします。



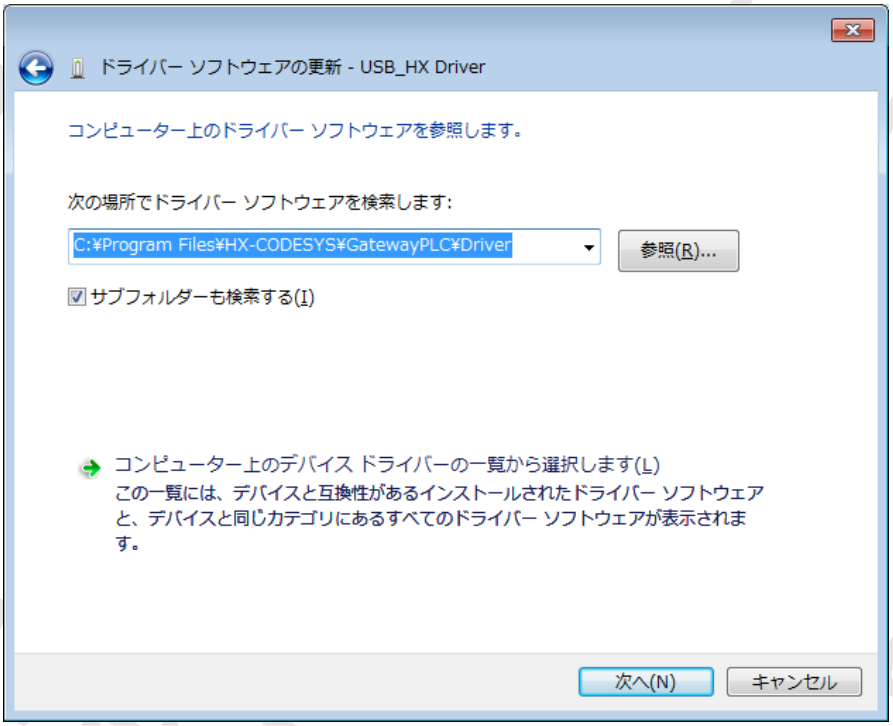
4. ドライバソフトウェアのインストールを一旦終了します。



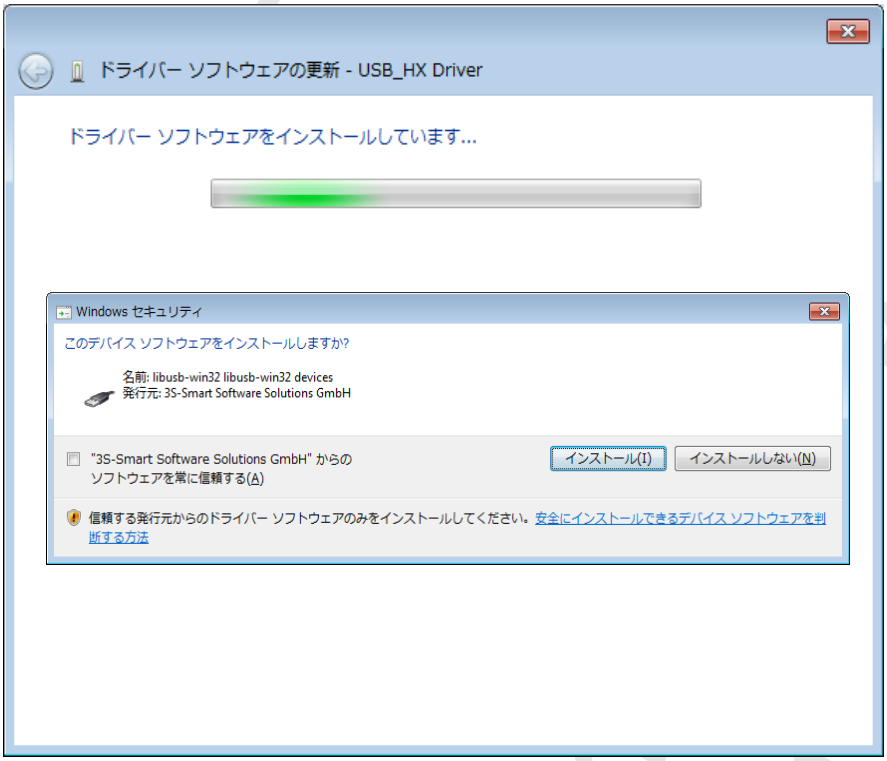
5. デバイスマネージャを開き、他のデバイスとして表示されている USB_HX_Driver を右クリックし、ドライバーソフトウェアの更新をクリックします。



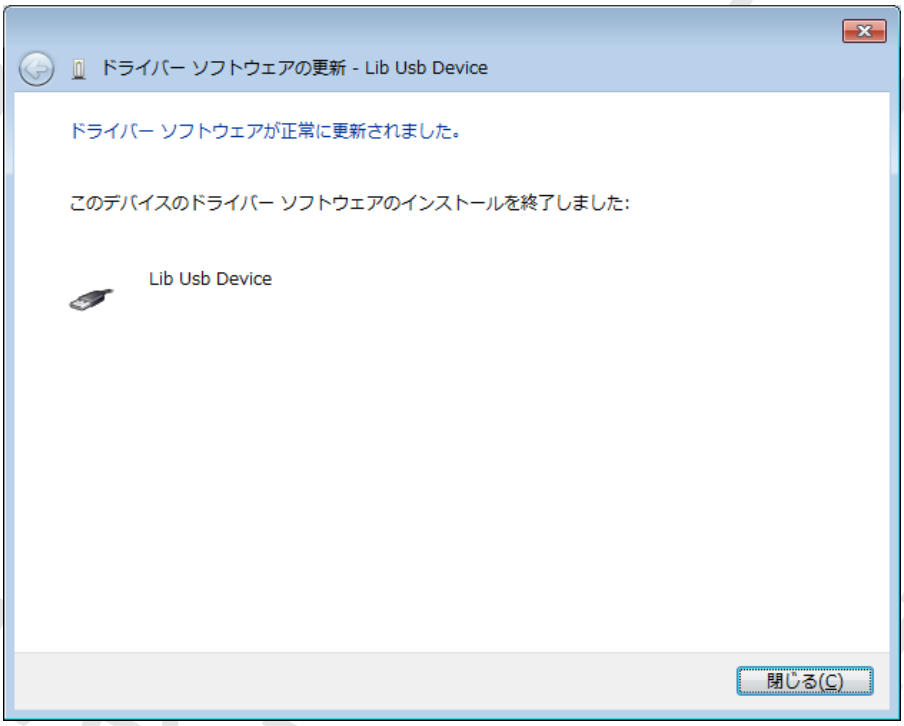
6. ドライバーを手動で検索を選択し、下記のフォルダを指定します。

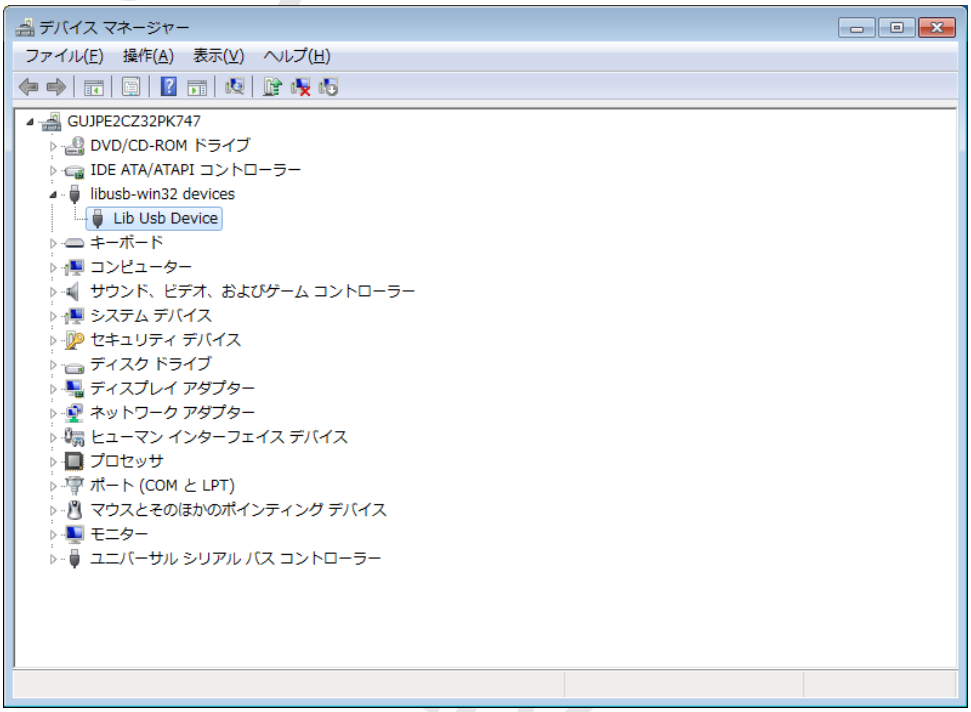


7. ドライバソフトウェアをインストールしてください。



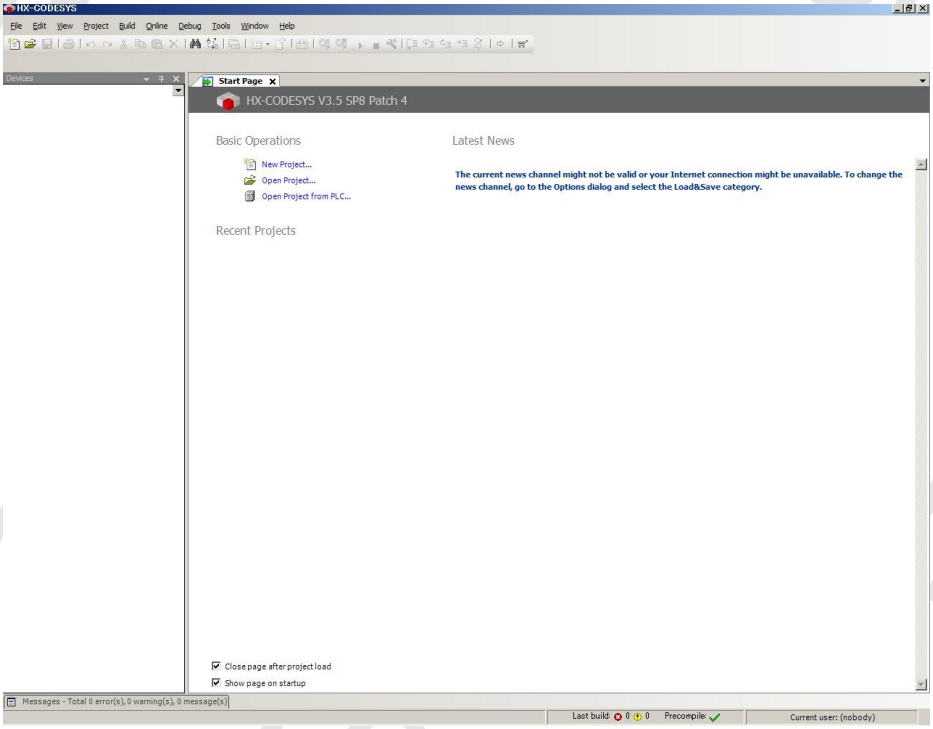
8. 正常に終了したらデバイスマネージャには、下記の様に表示されます。



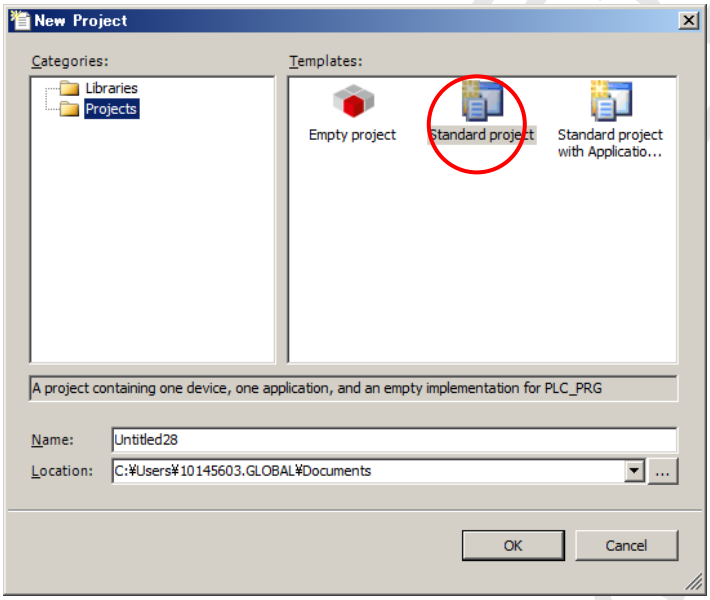


1.2 Startup

[スタートメニュー]-[全てのプログラム]-[HX-CODESYS]-[HX-CODESYS V3.5 SP8 Patch 4]をクリックすると、下記のスタートページが表示されます。

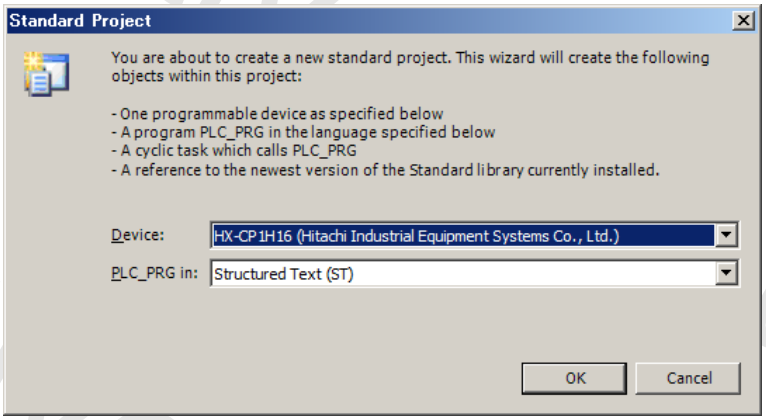


Click  icon or choose [File]-[New Project...] to create a new project file. Then New Project dialog box appears as below. Choose “Standard project”, enter new file name, specify location and click [OK].

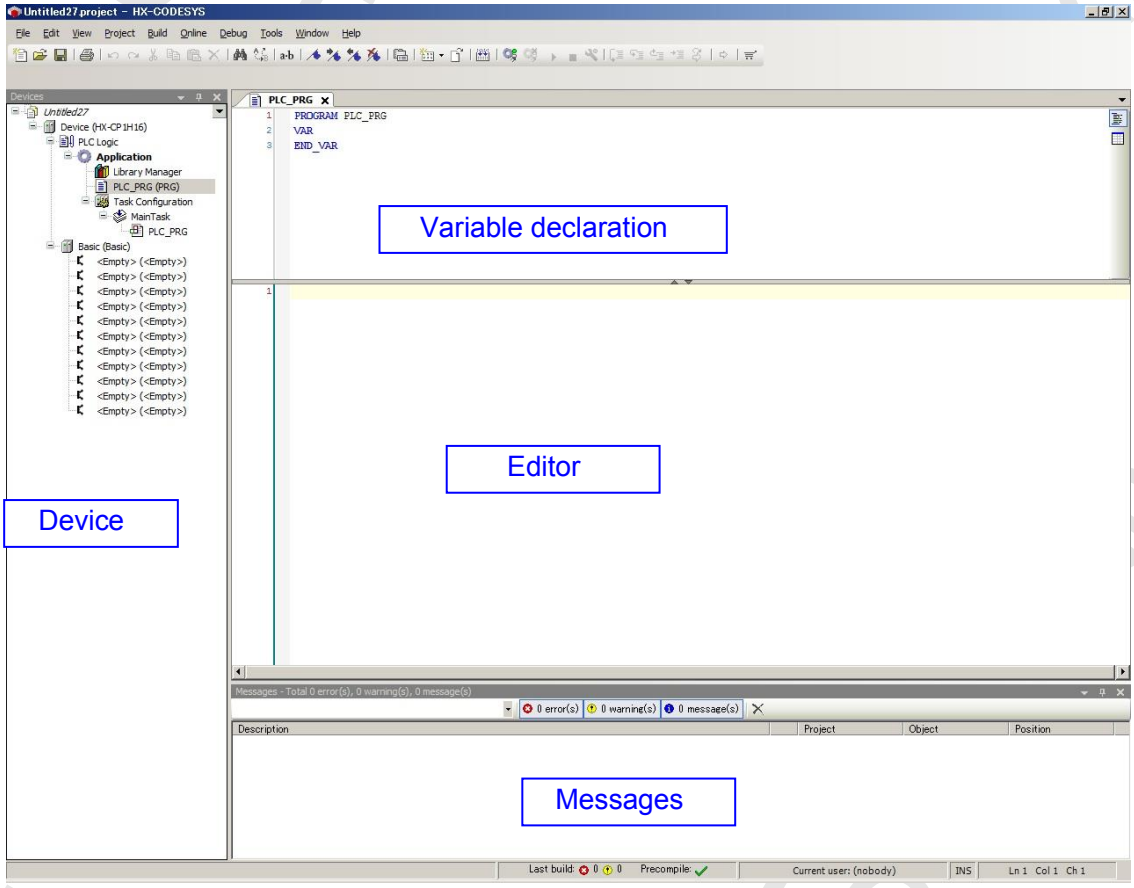


Choose CPU type and programming language and click [OK].
Available languages are as follows.

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



以下のような画面が基本構成となります。



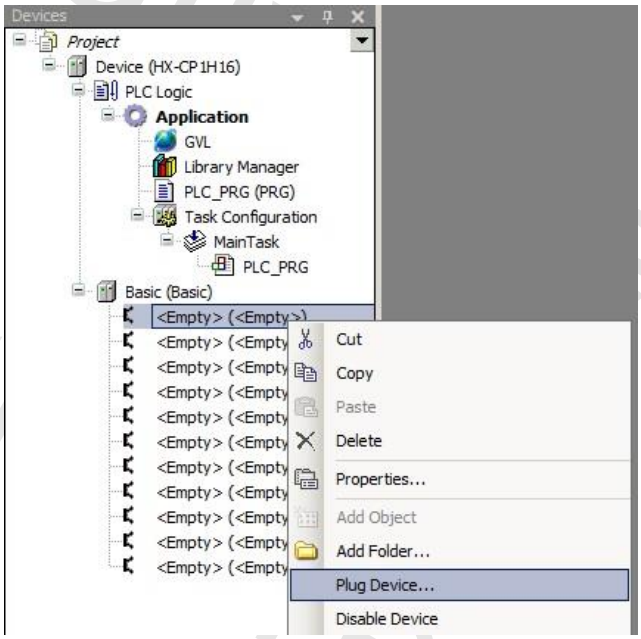
In the default setting, Device tree is behind the POU window. Click Devices tab to show it. “Toolbox” and “Properties” windows can be shown by [View] menu.

Chapter 2 Programming

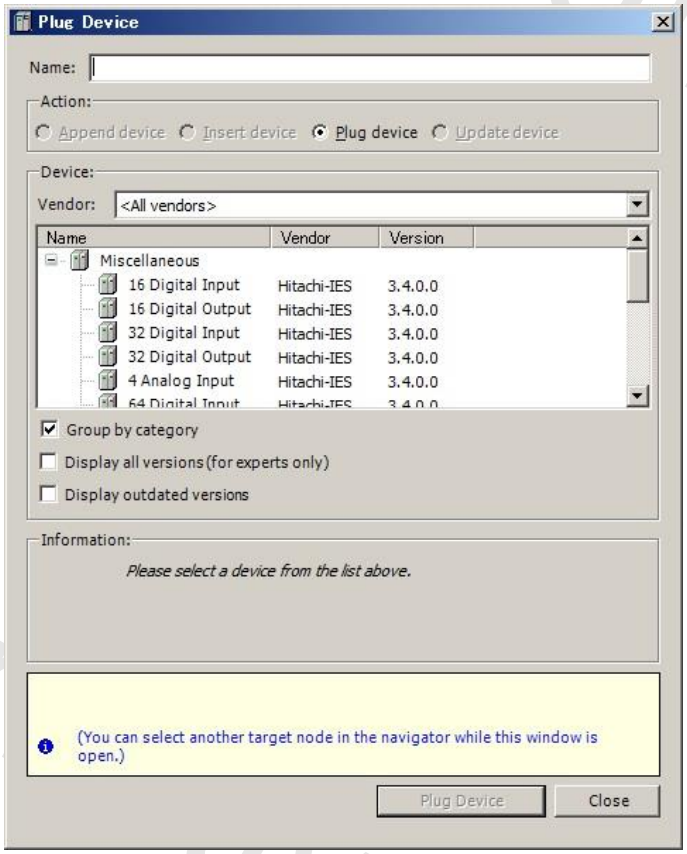
2.1 Plug Device (I/O configuration)

2.1.1 Plug Device (I/O configuration)

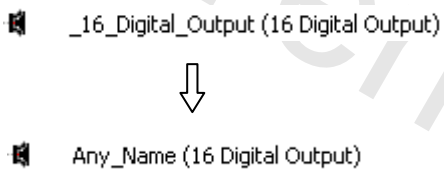
Right click on <Empty> slot and choose “Plug Device...”.



Choose I/O module for each slot. The next slot can be configured by clicking next empty slot <u>without</u> 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, H16		
EH-XD32, XDL32, XD32E, XDL32E, XD32H	32 Digital input	
EH-XD64	64 Digital input	
EH-YR8B, YR12, YR16, YR16D	16 Digital output	
EH-YT8, YT16		
EH-YTP8, YTP16, YTP16S		
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 *1	8 Analog output	
EH-AYH8M		
EH-AYG4M *1		
EH-CU, CUE	EH-CU/E	
EH-POS	EH-POS/4	
EH-LNK, OLNK, OLNKE, OLNKG	EH-LNK	Only 0-7 slot on basic base *2
EH-RMP2	EH-LNK	
EH-FLN2, FLN3	EH-FLN2/3	

*****1 Although the number of channel is not 8, configure “8 ch. Analog ”.

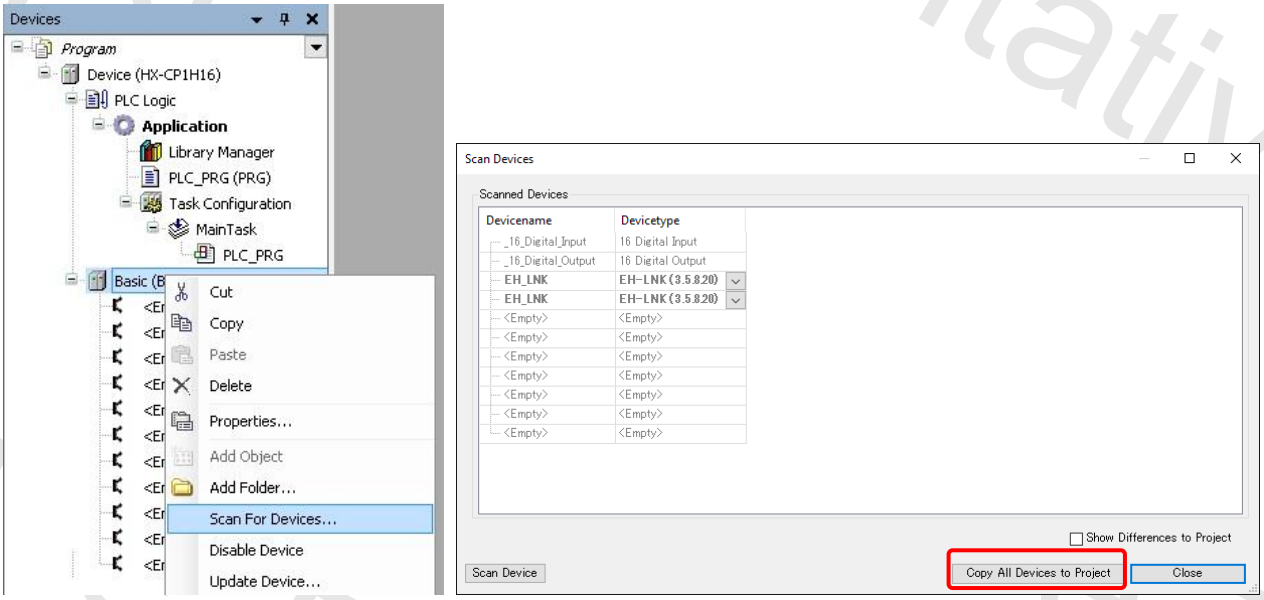
*2 If these module are mounted at 0-7 slot on basic bas, the errors blow are logged when download application to HX-CPU.

・In case of EH-LNK : EH-LNK is mounted on slot *. Allowed position for EH-LNK is slot 0 to 7.

・In case of EH-FLN2/3 : 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.

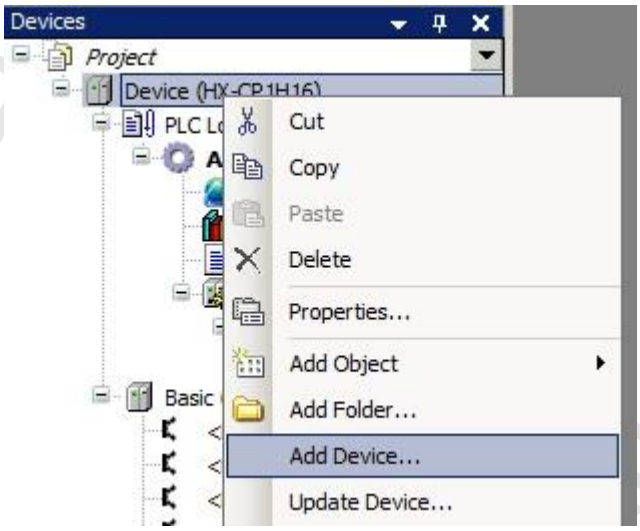


Note

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

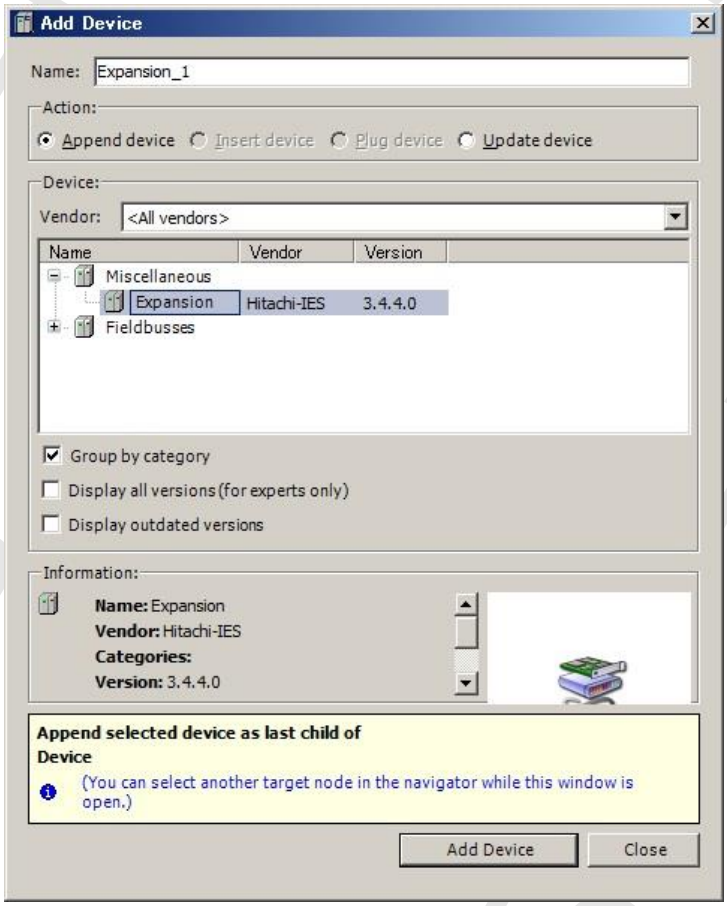
2.1.3 Expansion unit

Instead of “Plug Device”, choose “Add Device” to configure expansion units.



Select “Expansion” under Miscellaneous. HX-CP1S08, HX-CP1H16 allows to expand up to 5 expansion bases.

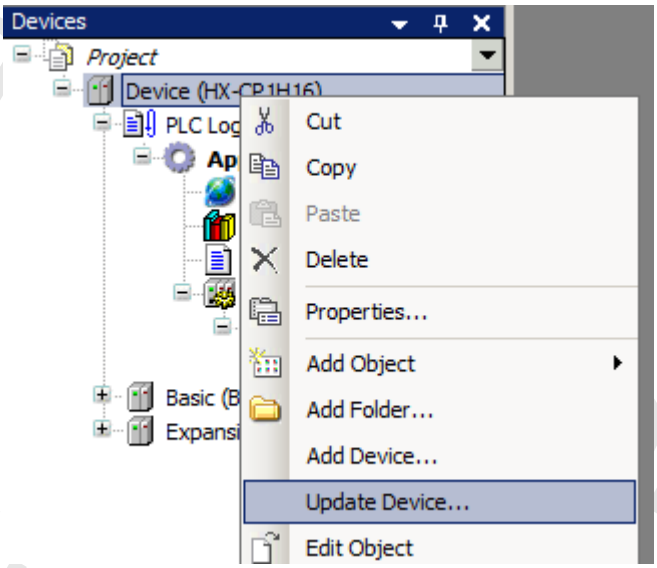
EH-IOCH2 側にて設定するユニット No.は、CPU から近い方から Unit1 として順番に設定してください。



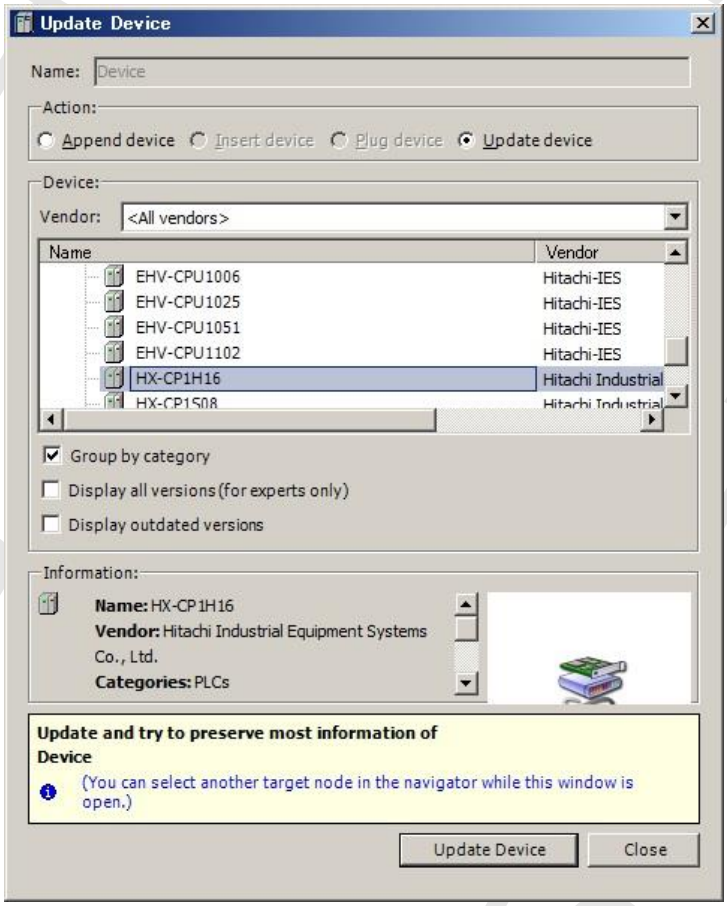
2.1.4 Update Device

Although device (CPU) type is required to set when creating new project, it can be changed later.

Right mouse click on the device and choose “Update Device”. Then “Update Device” windows appears.



Select CPU type , and click “Update Device” button.

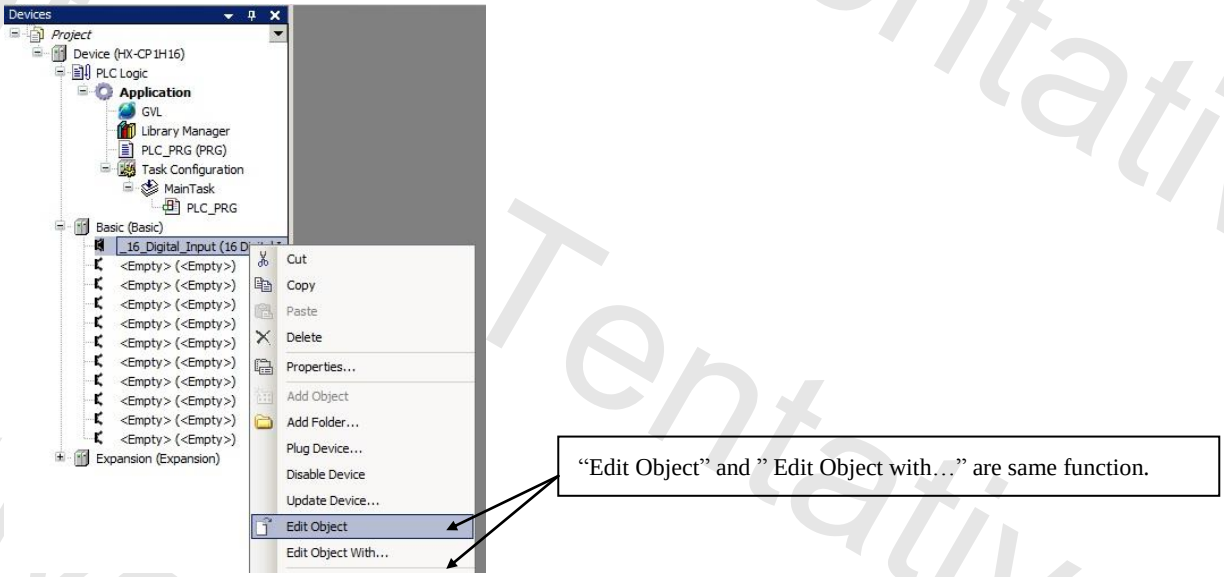


2.1.5 I/O address

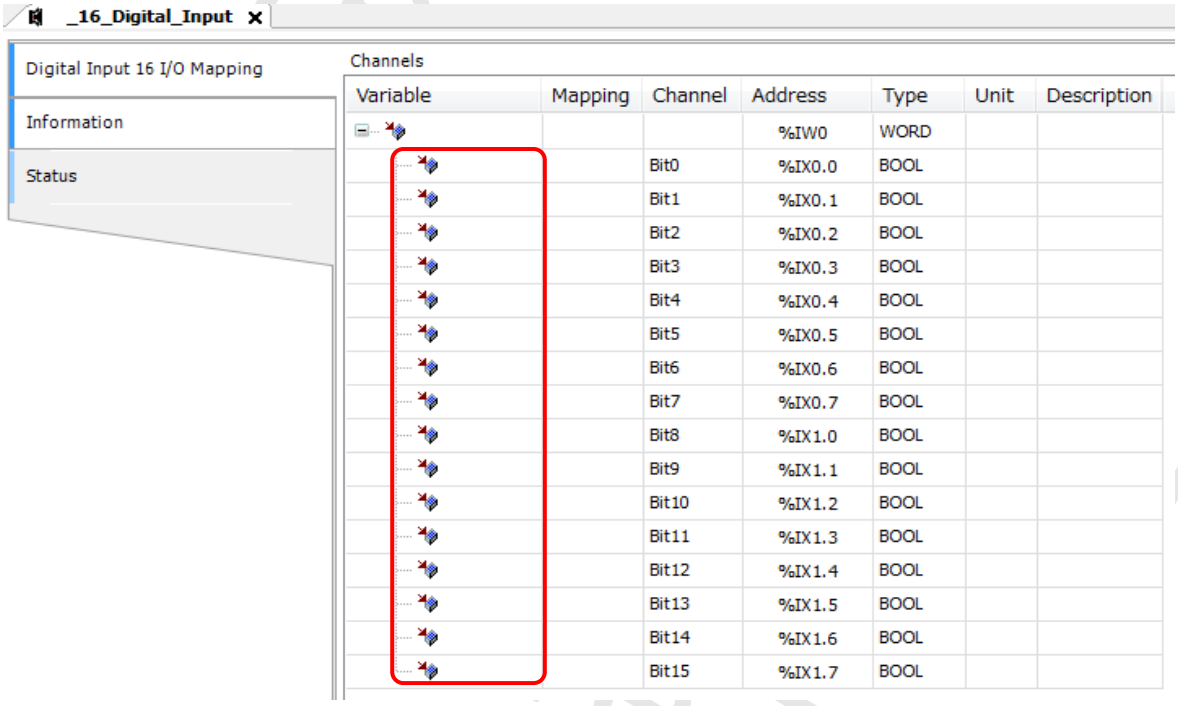
I/O addresses and variable names can be linked in two different ways: Global variable or Local variable as below.

[Global variable]

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



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

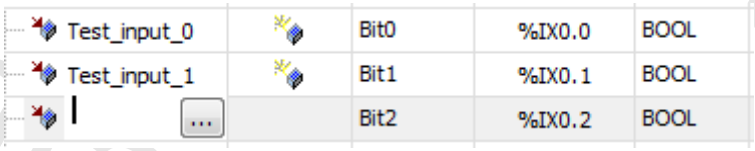


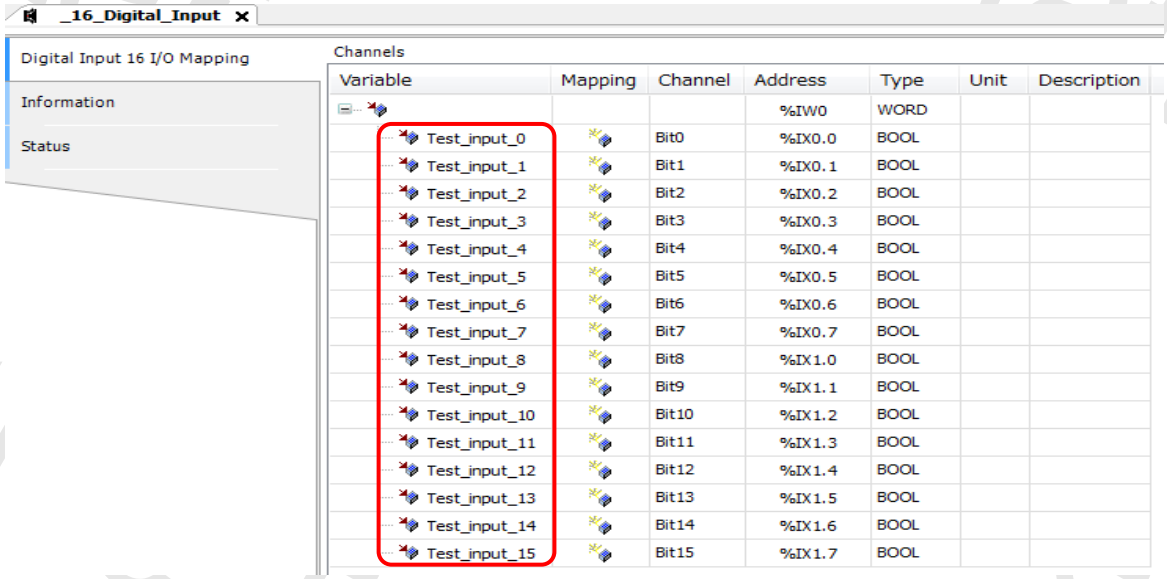
Variable	Mapping	Channel	Address	Type	Unit	Description
			%IW0	WORD		
		Bit0	%IX0.0	BOOL		
		Bit1	%IX0.1	BOOL		
		Bit2	%IX0.2	BOOL		
		Bit3	%IX0.3	BOOL		
		Bit4	%IX0.4	BOOL		
		Bit5	%IX0.5	BOOL		
		Bit6	%IX0.6	BOOL		
		Bit7	%IX0.7	BOOL		
		Bit8	%IX1.0	BOOL		
		Bit9	%IX1.1	BOOL		
		Bit10	%IX1.2	BOOL	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

Note

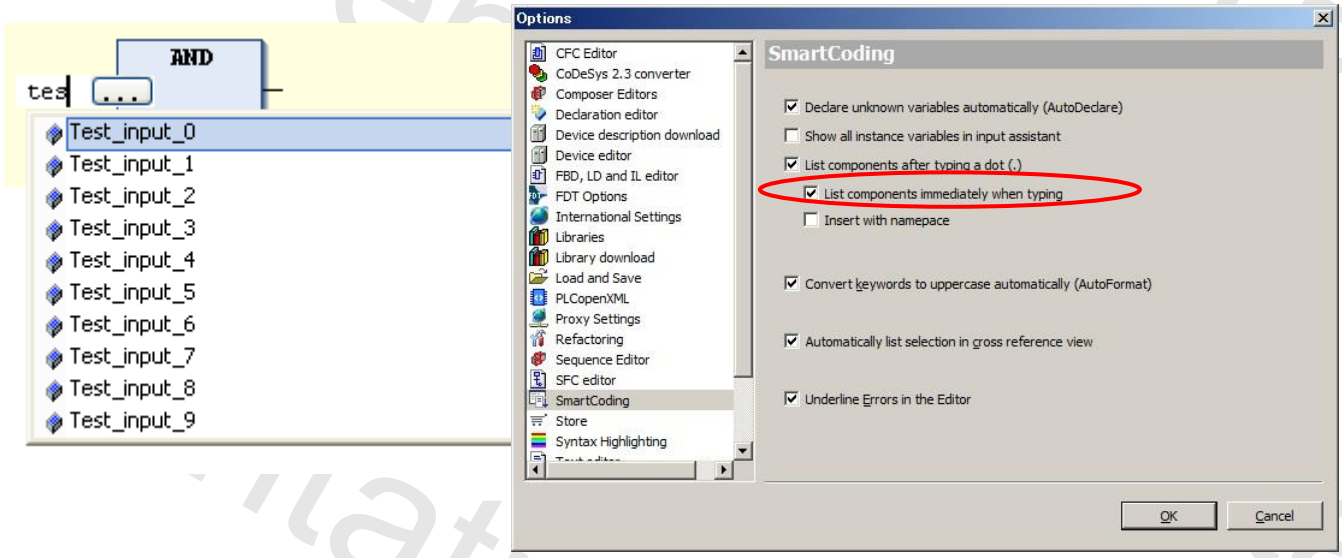
Available characters for variable names are only alphabet a to z, A to Z and number 0 to 9 and _ (underscore). The first character must not be numeric characters. Several words like BOOL, WORD, IF, FOR etc. are reserved.

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

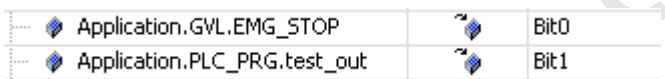




After defining variable names, they will be automatically listed up when it is used in all POU with assist of auto-complete that are displayed in case of selecting “List components immediately when typing”.



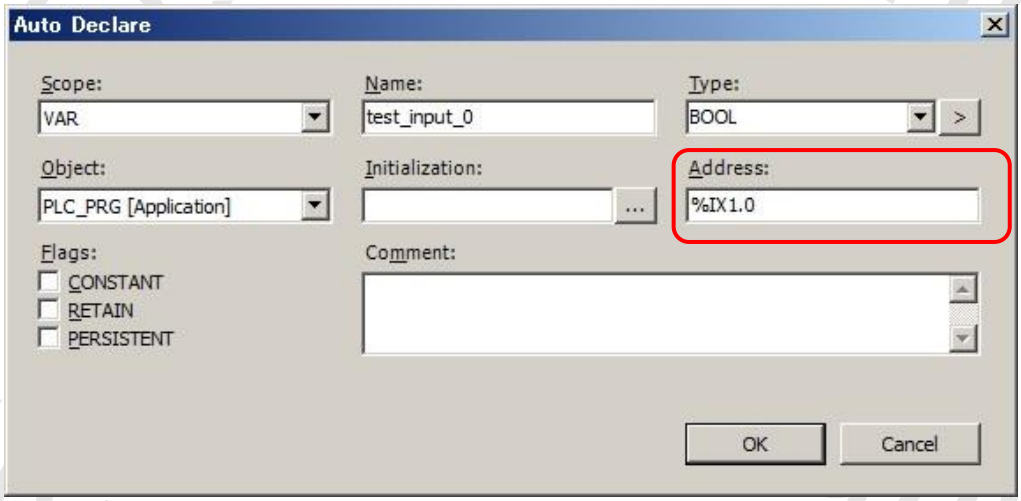
If a variable is already used (declared) in POU or global variable list, it can be taken by clicking … icon in I/O mapping window. (… icon appears by clicking empty field.)



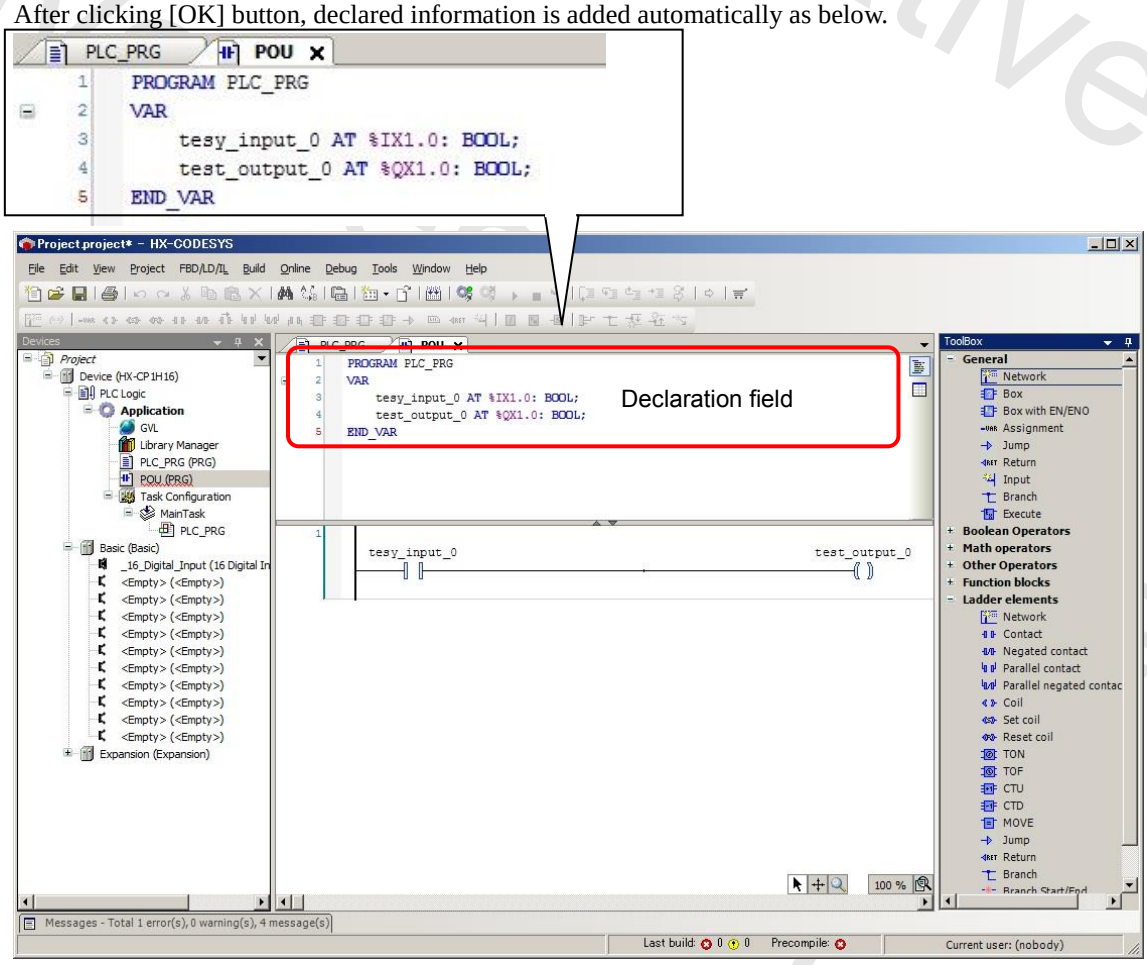
[Local variable]

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

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



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



I/O address example of 64 points 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

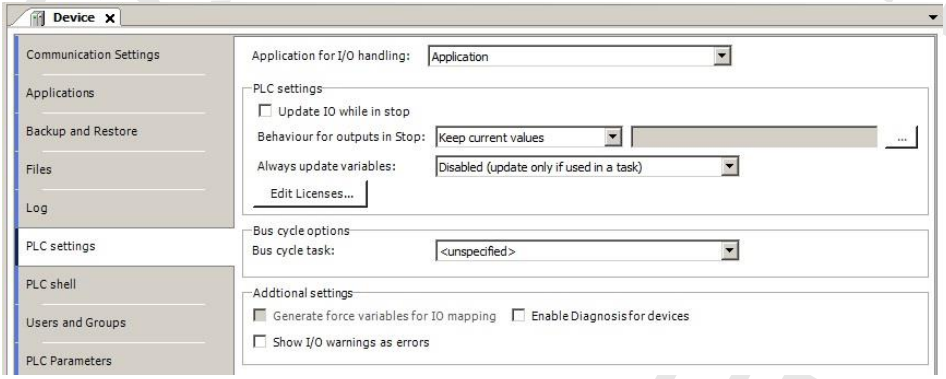
If you use the application of EHV+ series to HX series, be noted that direct IEC address data is swapped. In case of using variable name, data is not swapped.

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

	EHV+ series IEC address					HX series IEC adress				
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, Not used I/Os are updated.



Update IO while in STOP

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

Behaviour for outputs in STOP

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

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

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

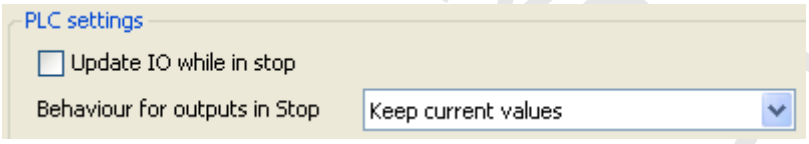
Always update variables:

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.

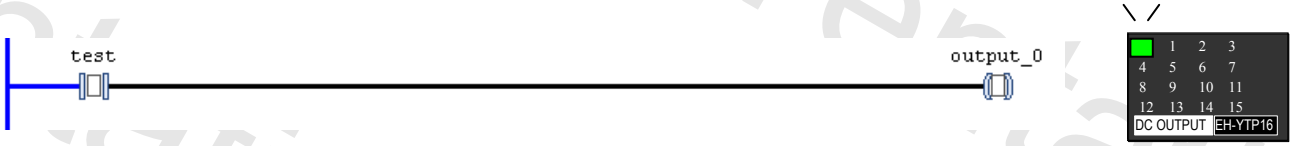
Note

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



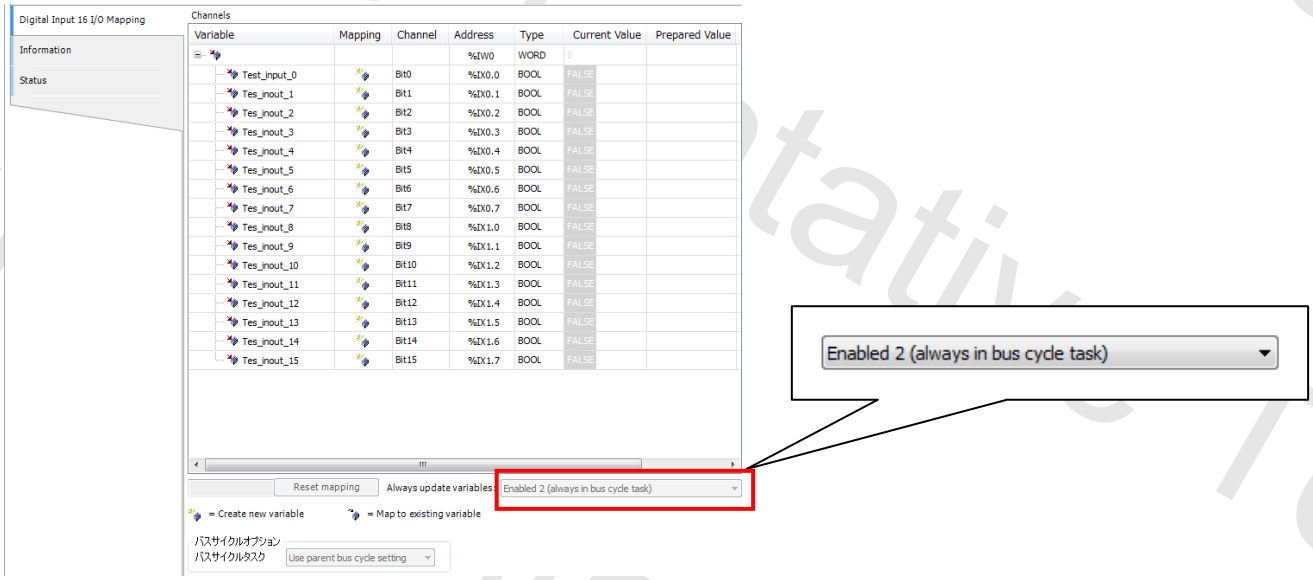
This is expected behavior. If this setting combination is required, keep in mind this mismatching and be careful to use.

Reset all outputs in STOP

This setting is in [Device]-[Configuration]. 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.

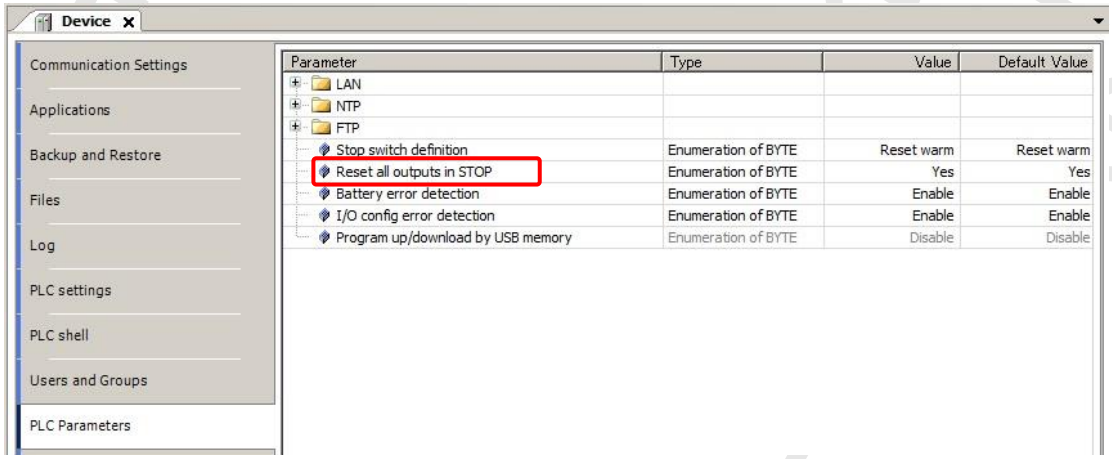
Note

プログラムで使用していない I/O はリフレッシュされません。従って、例えば使用していない外部入力を ON してもオンライン中にマッピングテーブルをモニタした時に TRUE に見えません。マッピングテーブル右下の「常に変数を更新」を有効にすると、使用/未使用に関わらずリフレッシュされます。



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.

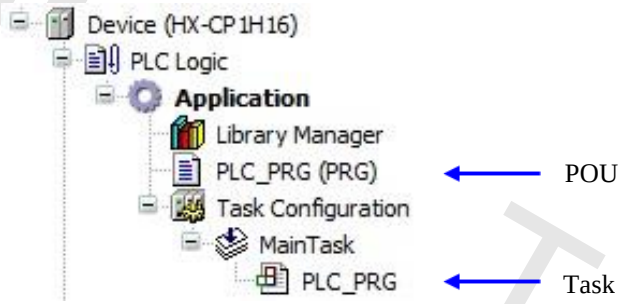


Note

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.

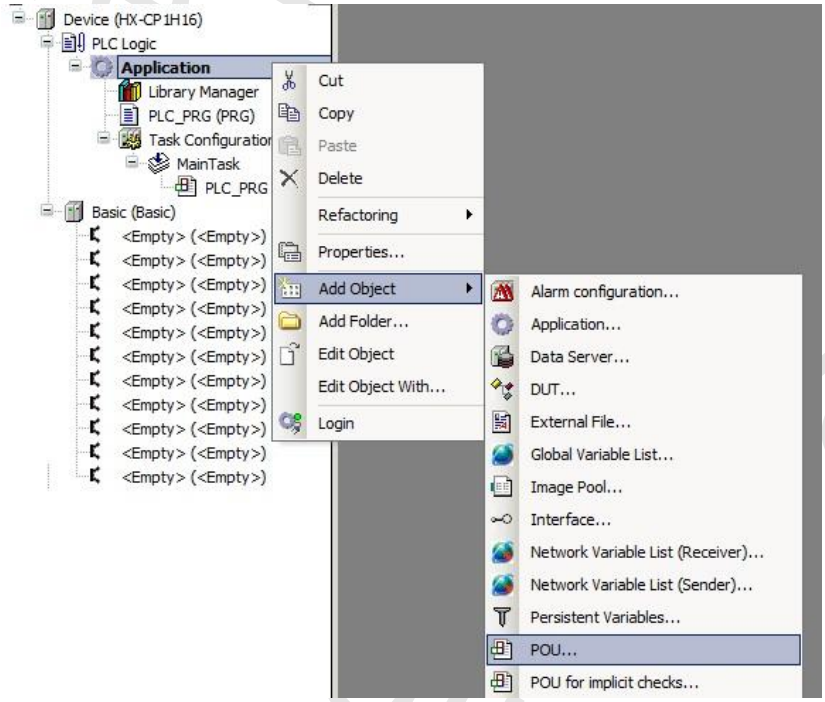
2.3 POU and task

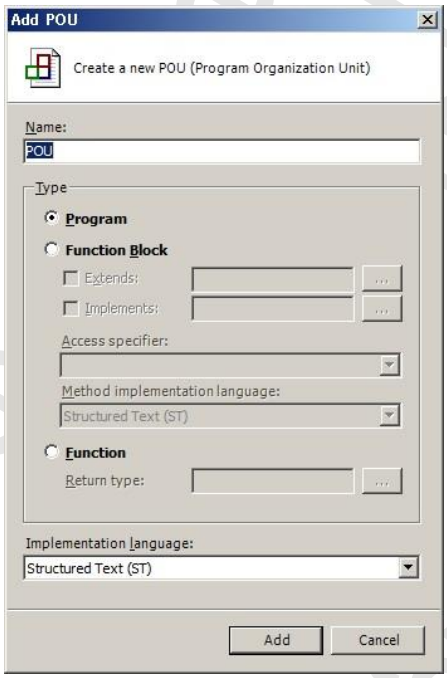
One application has at least one POU and one task as shown below.



POU

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

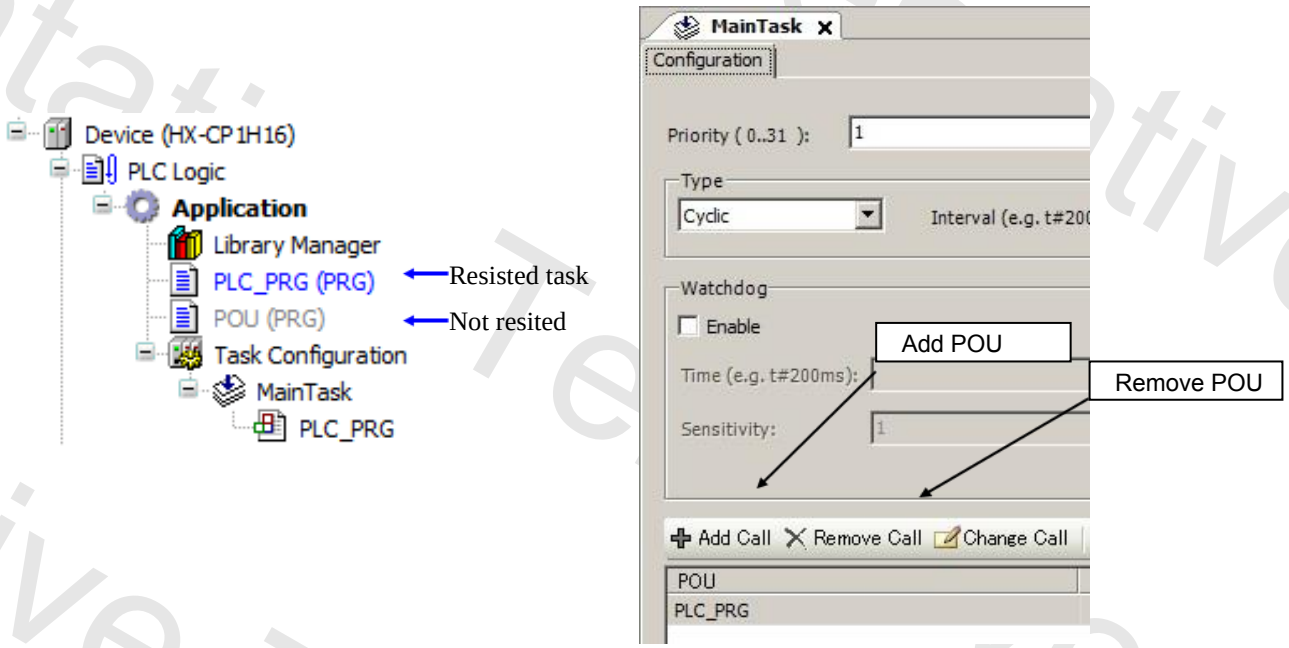




Task

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

Put priority, choose type of task and add or remove POU accordingly.



Priority (0-31)

0 is the highest priority, 31 is the lowest.

Type

Cyclic task

The task will be processed cyclic according to the time definition given in the field “Interval”.

Event task

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

Freewheeling task

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

Status task

The task will be started when selected variable is TRUE.

Watchdog

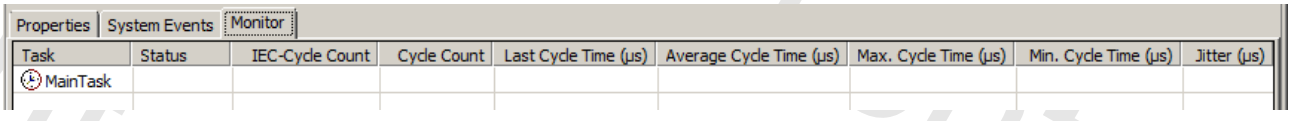
When it is enabled, watchdog function is activated. If program execution time exceeds watchdog time, CPU stops program execution with “24” error code displayed at 7 segment LED.

Example: In case of “Time:#5ms” and “Sensitivity:10, there are two detect conditions as below.

・Detect condition 1 : over 10 times continuously

・Detect condition 2 : over 50ms（5ms×10）

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



2.4 Available characters for variable names

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

Supported characters

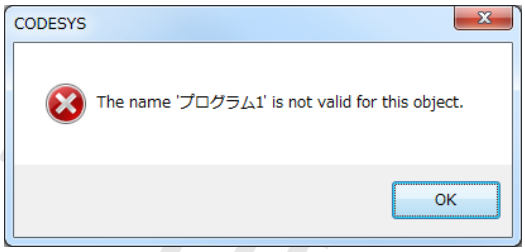
Types	Supported	Remarks
Numerical	0 to 9	Not allowed to begin with numeric characters.
Alphabetical	a to z, A to Z	
Symbol	_	Trailing underscores are not allowed.

Examples for variable names

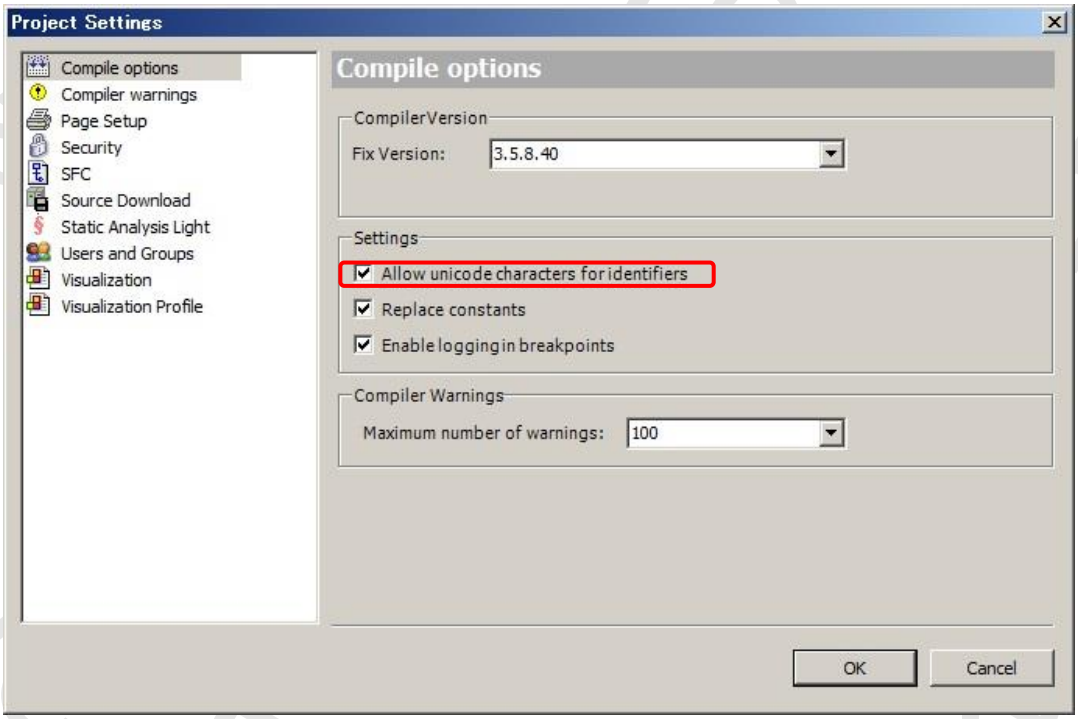
Allowed or not	Examples	Descriptions
Allowed	Test_200	
	TEST	
	Test55	
	_Test	
Not allowed	2test	Starting with numeric character.
	test__200	Trailing underscores are not allowed
	test-5	Minus sign is not allowed.
	test#3	other signs than underscore are not allowed.
	test 3	Space is not allowed.
	IF	Reserved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

注　意

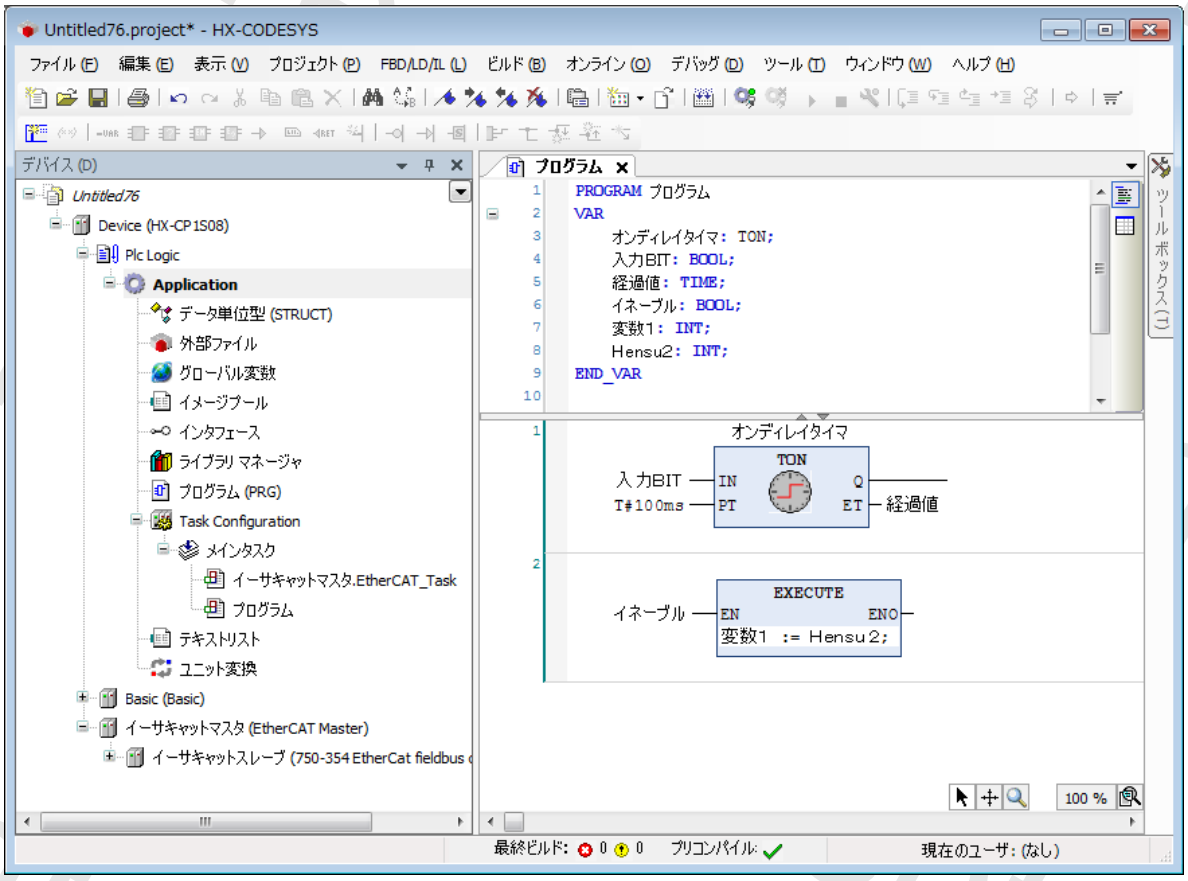
Warning message is displayed at the points that the Unicode characters are not allowed.



If you use unicode characters, click “Project”-“Project setting”, and check “Allow unicode characters for identifiers” in “Compile option”.



Example: Japanese



2.5 Variables

2.5.1 Data memory

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

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

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 Maker memory

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

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

Variable declaration

```
VAR
    dwTest AT %MD10: DWORD;
    wTest_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 ～ %MX49152.7
BYTE	%MB0 ～ %MB49152
WORD	%MW0 ～ %MW24575
DWORD	%MD0 ～ %MD12287
LWORD	%ML0 ～ %ML6143

2.5.3 Numeric literals

Numeric literals are specified as follows.

Types	Examples	Remarks
Integer	`-12 0 123_456 +986 10#1234`	Underscore is ignored
Real	`-12.0 0.0 0.4560 3.14159_26`	Underscore is ignored
Real with exponents	`-1.34E-12 1.0E+6 1.23E6`	
Base 2	`2#1111_1111 2#1110_0000`	Underscore is ignored
Base 8	`8#377 8#340`	
Base 16	`16#FF 16#ff 16#1234_ABCD`	Underscore is ignored
Boolean zero and one	`0 1 FALSE TRUE`	FALSE=0, TRUE=1
Time	`T#100ms, T#5.5s`	Timer (TON, etc.)
Date	`DT#2012-12-31-12:34:56`	RTC (Realtime clock)

2.5.4 Data types

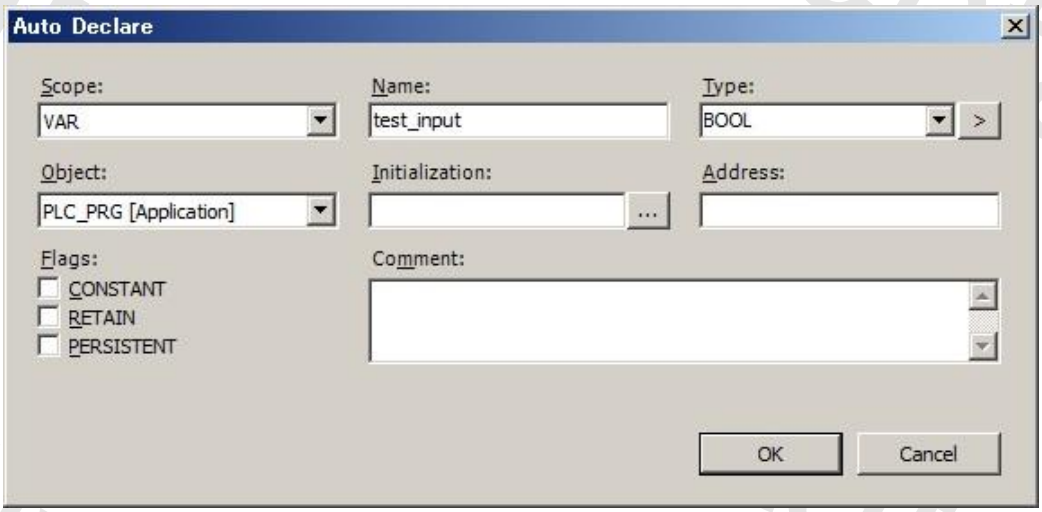
HX-CODESYS supports below data types.

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} ～ $2^{63}-1$
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	

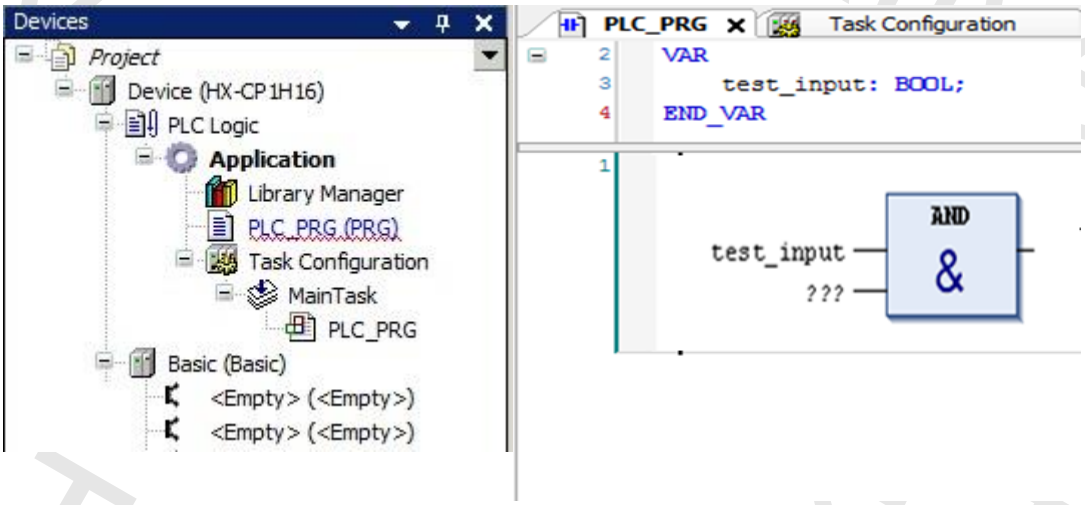
23	`ARRAY`	Array	—	Ex. in variable declaration
				`test: ARRAY[0..100] OF WORD;`
				in user program
				`test[5]:=20;`

2.5.5 Local variable

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



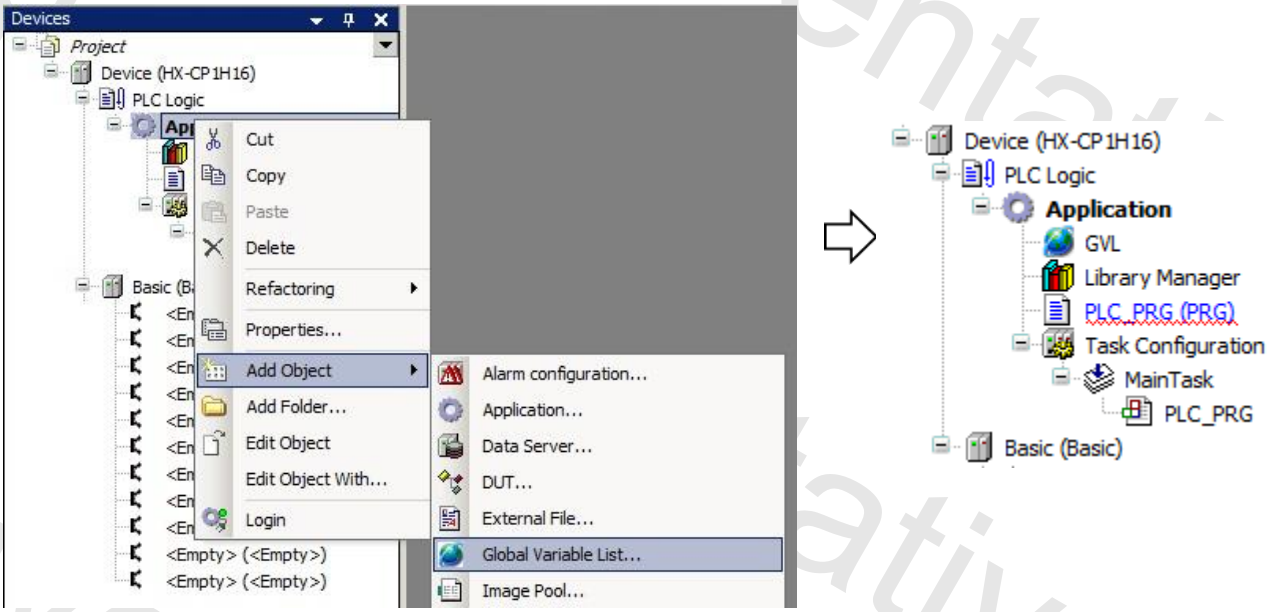
Click [OK] button, this variable is registered in declaration part of POU as below.



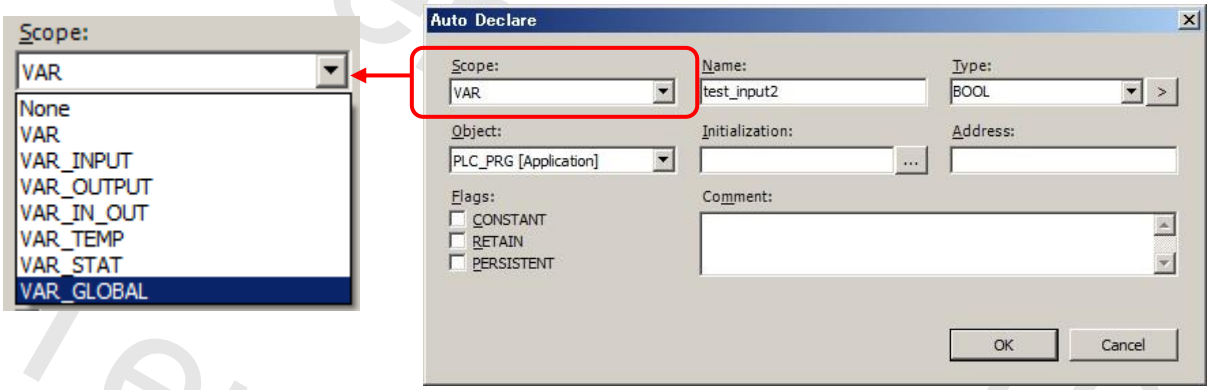
This variable is valid only in the POU. Even if same variable name is used in another POU, Auto Declare window will appear and it will be assigned in another memory location and handled as different variable.

2.5.6 Global variable

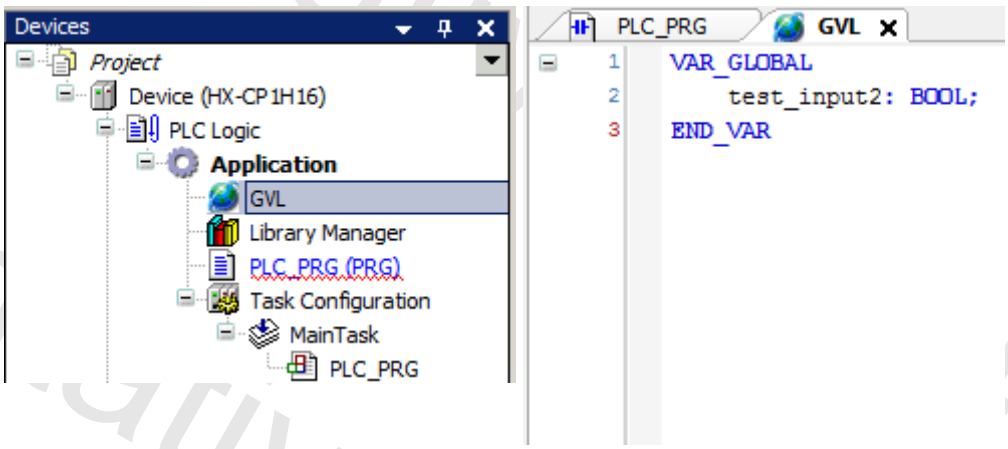
If variables need to be commonly used in all POUs, “Global Variable List” must be created by right click on Application as below.



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.



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	
	Specified by	IP address (default) / Host name	Setting how to set NTP server
	IP address or Host name	Setting IP address or host name	
	Access cycle	Setting access cycle to NTP server	
	Timeout	Setting timeout time	
	TimeZone	Setting time zone	
FTP	FTP server	Setting parameters regarding FTP server Refer to page 2.19	
	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.
Serial port term. resistor (RS-485)		No (default)	終端抵抗なし
		Yes	[1]TX+，[3]RX+ と[2]TX-，[4]RX-端子間に終端抵抗 120Ω
Reset all outputs in STOP		Yes (default)	all outputs are reset by hardware signal on the backplane when switching to stop mode. (Refer to page 3-16)
		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 eoor
Program up/download by USB memory		Enable / Disable (default)	USB メモリによるプログラムのアップロード/ダウンロード機能を有効にするか(Enable)、しないか(Disable)を設定します。

Note

-ETH1,2,3 の IP Address へネットワークアドレスの入力が可能ですが、設定不可となります。(例：10.0.0.0) この場合、ログに設定できない旨のエラーが表示されます。

-ETH1,2,3 の Subnet Mask へ不正なサブネットマスクの入力が可能ですが、設定不可となります。(例：

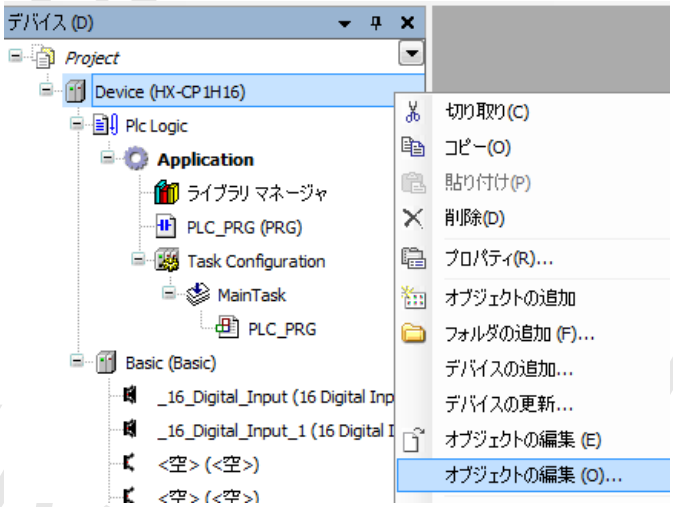
255.255.254.0）この場合、ログに設定できない旨のエラーが表示されます。

2.7 Communication settings

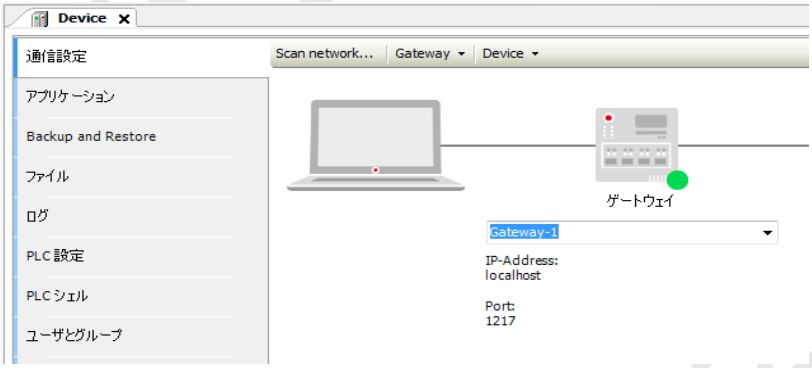
設定方法

How to configure

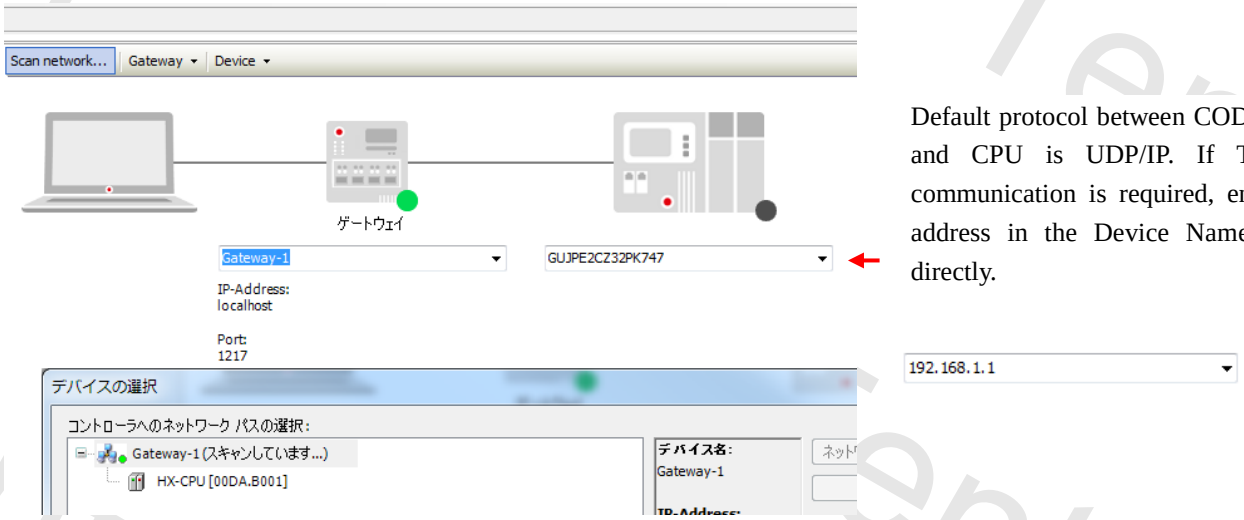
Double click on “Device (HX-CPxxxx)” or right click and choose “Edit Object”.



“Device” window will appear as below. Choose “Communication Settings” tab and click “Scan network”.

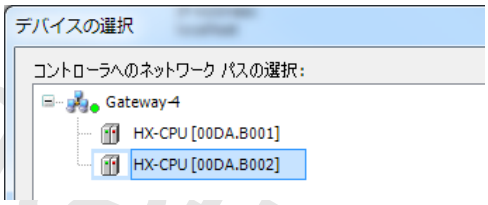


「Scan network」をクリックして、使用可能なデバイスを検索します。



Default protocol between CODESYS and CPU is UDP/IP. If TCP/IP communication is required, enter IP address in the Device Name field directly.

Scan network を行った際に、複数のノードが検索される場合があります。

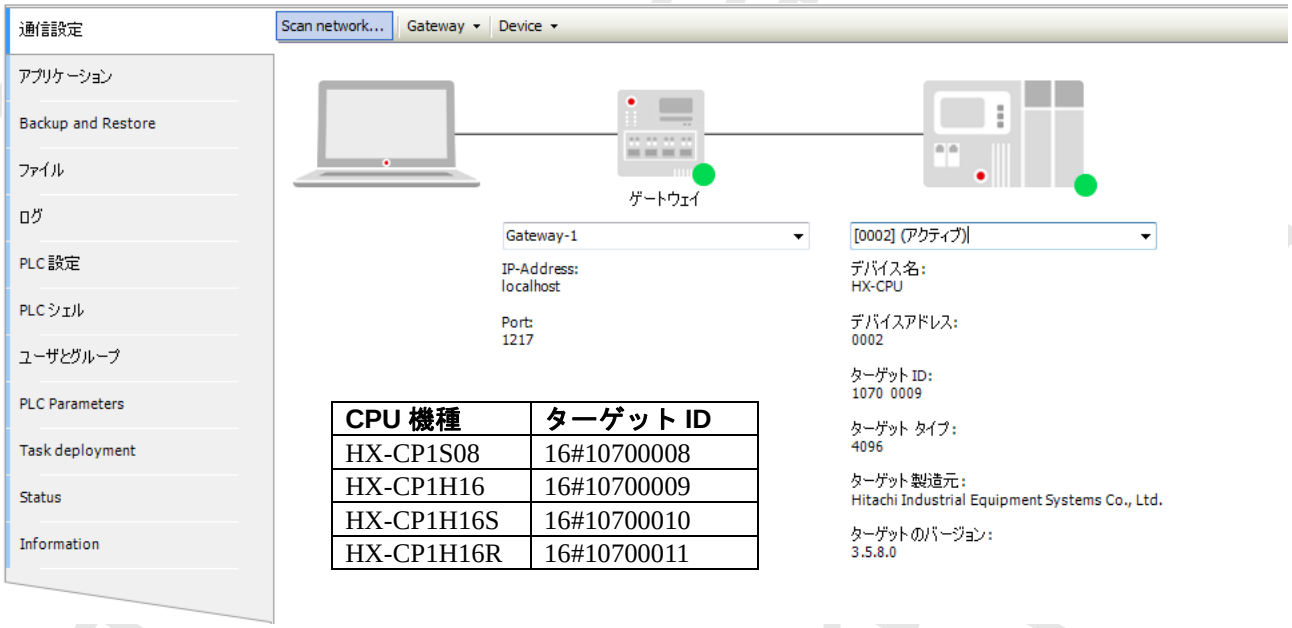


この場合は、ノードを選択した状態で[Wink]ボタンをクリックすると、そのノードの RUN ランプが点滅します。CPU が STOP 状態の場合 2 回点滅します。CPU が RUN 状態の場合 3 回点滅します。

注　意

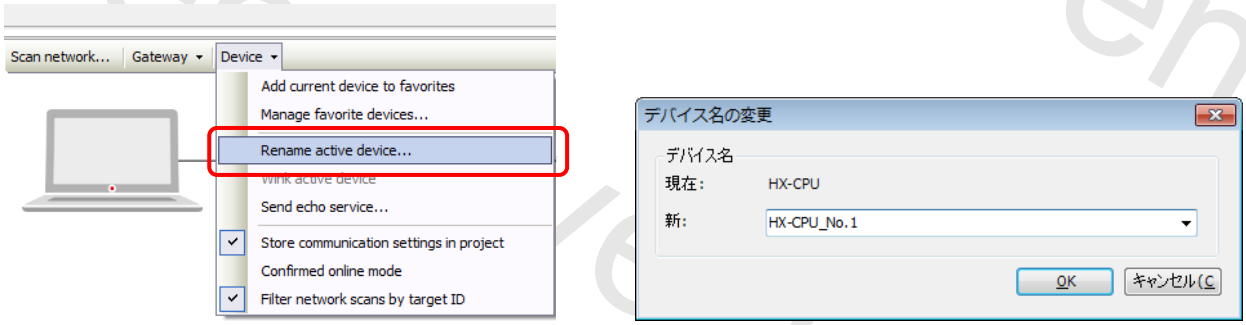
スタートページの「PLC からプロジェクトを開く…」よりデバイスを検索した際は、[Wink]ボタンにより点滅しません。

接続したい CPU をクリックしてから[OK」ボタンをクリックすると、接続対象に指定されます。
これで通信の設定は完了です。



CPU 機種	ターゲット ID
HX-CP1S08	16#10700008
HX-CP1H16	16#10700009
HX-CP1H16S	16#10700010
HX-CP1H16R	16#10700011

接続対象に指定後は、「Device」メニューの「Rename Active Device…」をクリックすると、デバイス名を変更することが出来ます。



注　意

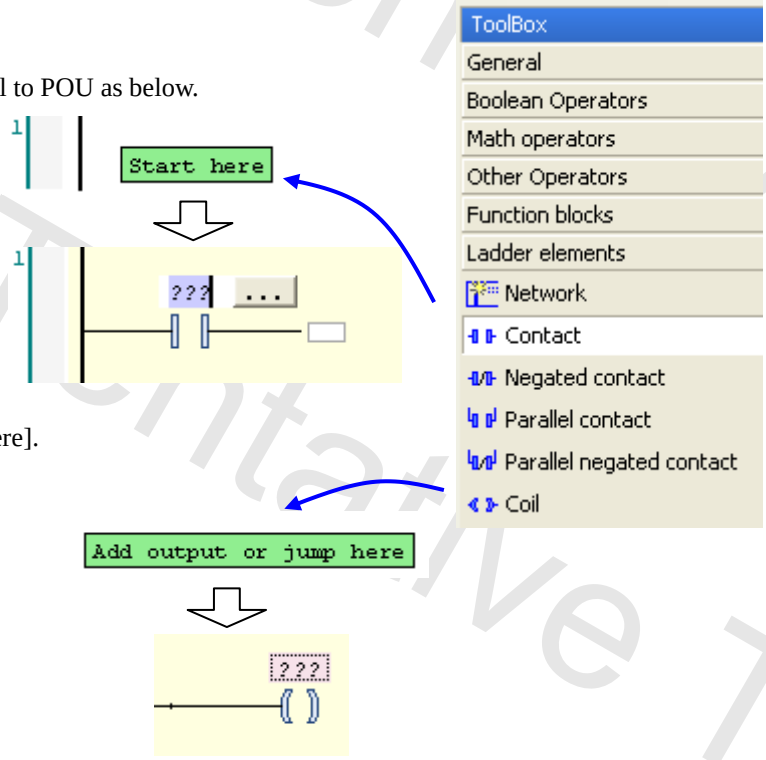
・CPU と PC 間がイーサネットケーブル、USB ケーブルすべてで接続されていたとしても、「ネットワークスキャン」では最も早く検索された通信種別ひとつだけしか表示されません。また、「フィルタ」を「ターゲット ID」から「なし」に変更すると、ネットワーク上のすべてのデバイスが検索されます。

2.8 Programming

Ladder programming

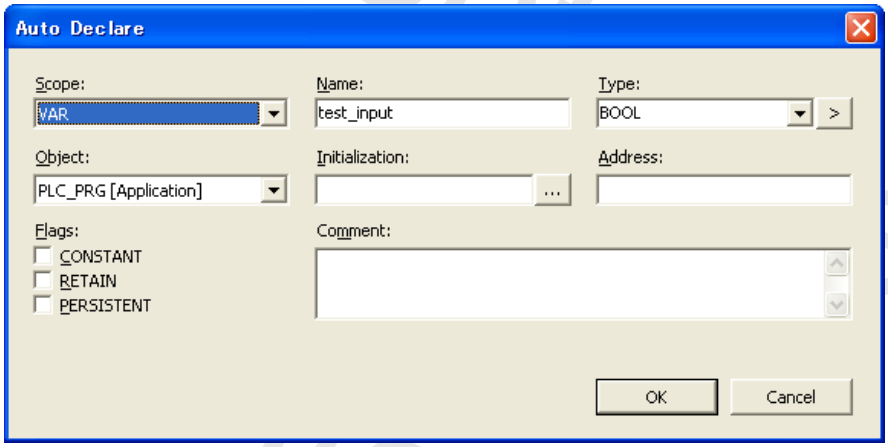
Basic ladder programming is shown below as a first step. Please refer to online-help of EHV-CODESYS for further information about programming.

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

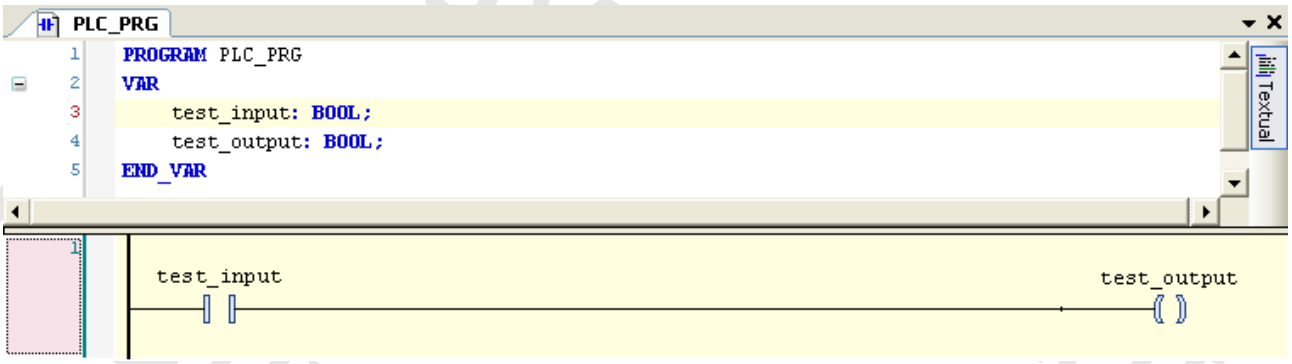


Contact
- Drag from ToolBox to [Start here].
- Menu [FBD/LD/IL]-[Insert Contact]
- Right mouse click [Insert Contact]
- Shortcut key [Ctrl + K]

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

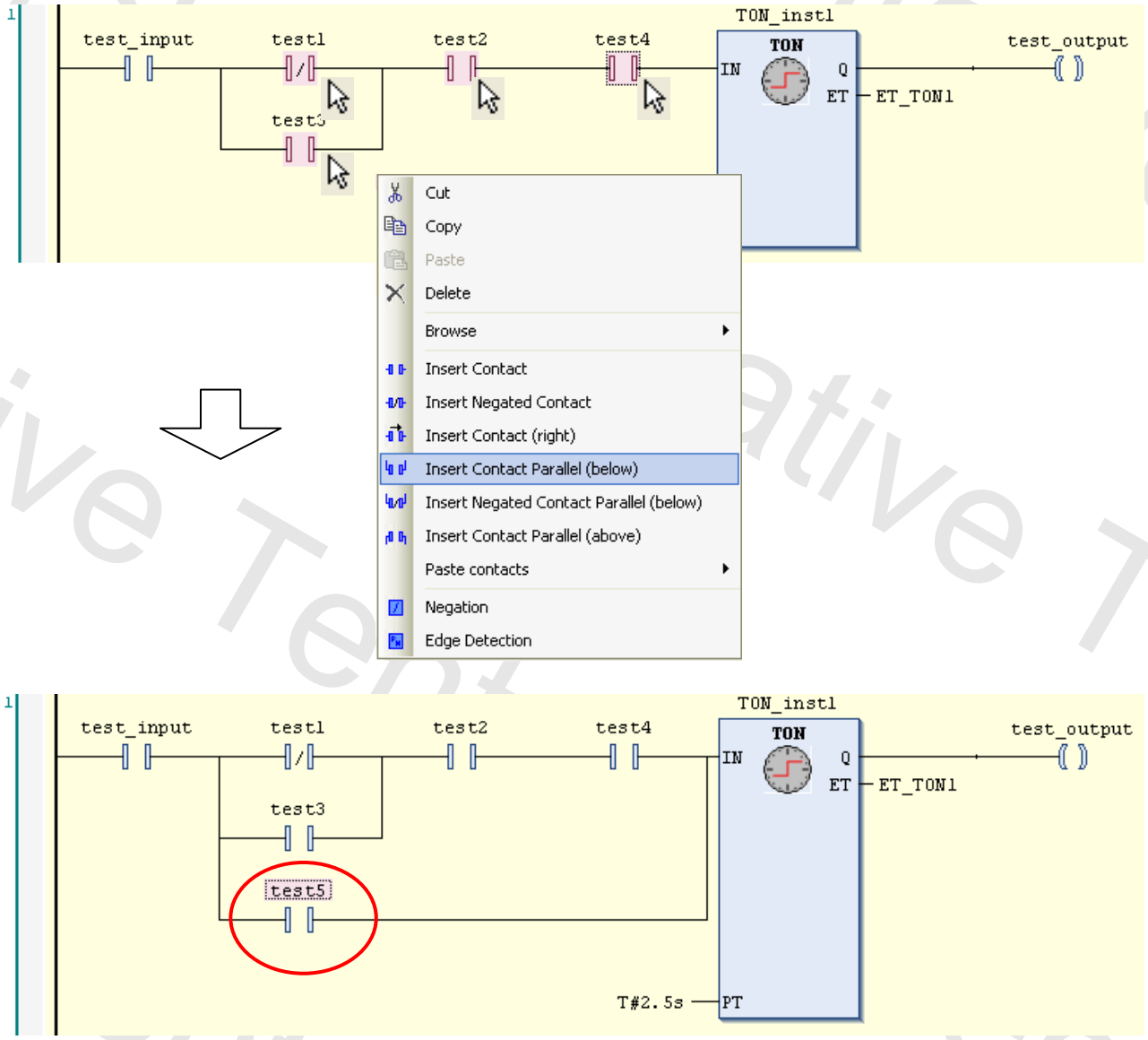


If new variable name is used, Auto Declare window appears automatically. Edit each input field and check-boxes if necessary, and Click [OK]. The variable is declared in declaration window as below.



Parallel contact across several contacts

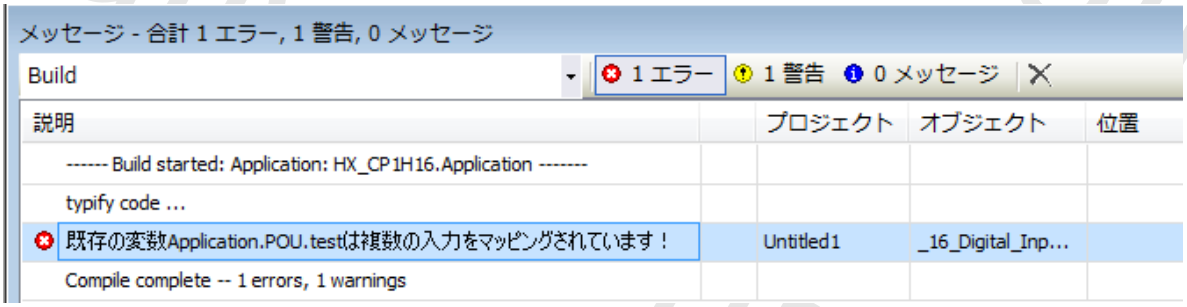
Click several contacts with shift key and choose [Insert Contact Parallel] in right-mouse click menu or press [Ctrl + R] key.



2.9 Login / Logout

Login

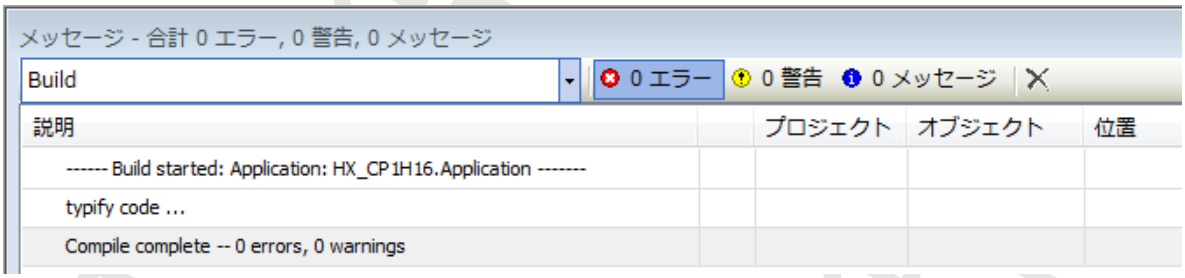
After programming, click  or choose [Build] in Build menu. If compiling fails, error information is shown at "Description" field as follows. Double click the message to jump to the part to be corrected.



Note

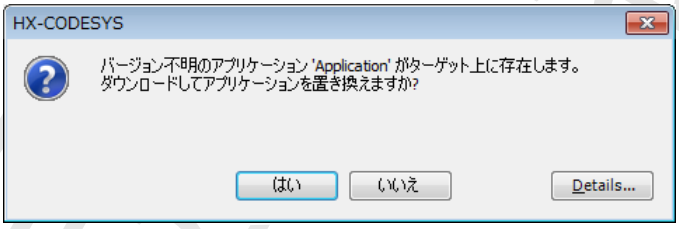
If unknown message appears, it is recommended to [Clean all] in Build menu. All compile information is deleted by this operation.

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

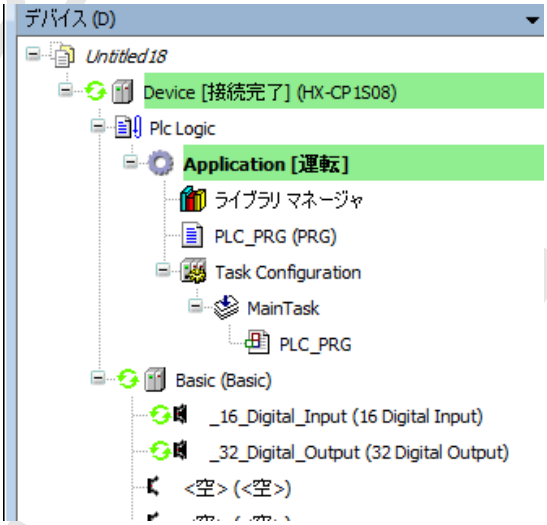


If no application is in the CPU, this message appears. Click [Yes] to download.

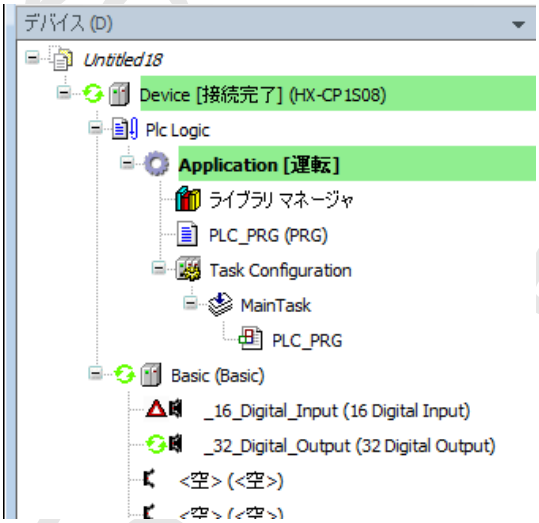
If unknown version of application is in the CPU, this message appears. Click [Yes] to download.



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.



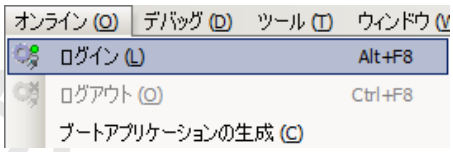
Online change（RUN 中変更）

オンライン変更は、ユーザプログラムを運転中（RUN 中）に、運転中のままプログラムの変更を行うことが出来る機能です。オンライン変更は、変更したプログラム部分の HX-CPU へダウンロードを行います。

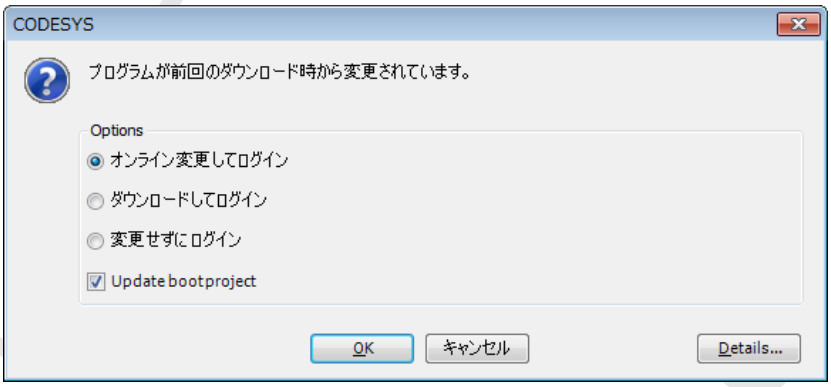
> **⚠ 注 意**
>
> オンライン変更は、実行中のユーザプログラム変更と再起動を行います。対象のシステムによっては、機器が誤動作し、人体に危険が及ぶ可能性があります。新しいユーザプログラムが対象のシステムで正常に動作することを十分に検証してから行ってください。

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.



オンライン変更前に「クリーン」もしくは「全てをクリーン」を行った場合、新しいプログラムをダウンロードしないとログインが出来なくなり、オンライン変更が出来なくなりますのでご注意ください。

注 意

ポインタ変数は、前回実行された最終サイクルでの値を保持します。オンライン変更によりポインタ変数が変更になる場合は、ポインタ変数は正しい値とならない可能性があるため、各サイクルで再度割り当ててください。

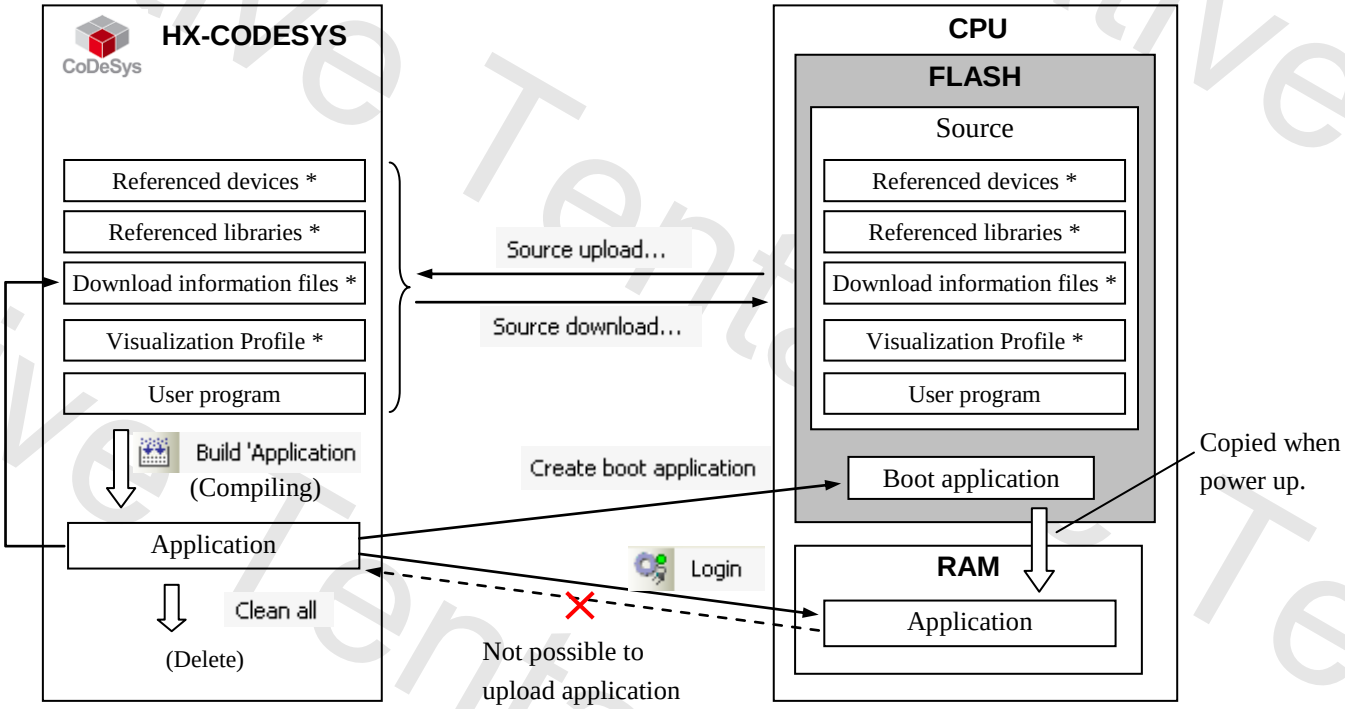
Logout


アイコンをクリックまたは「オンライン」メニューの「ログアウト」を選択してください。

ユーザプログラムの I/O 点数が約 30,000 点を超えるとログオフに時間がかかることがあります。30,000 点以上の I/O を使用する場合は POU を分け、モニタ不要な POU は閉じておくことで回避可能です。

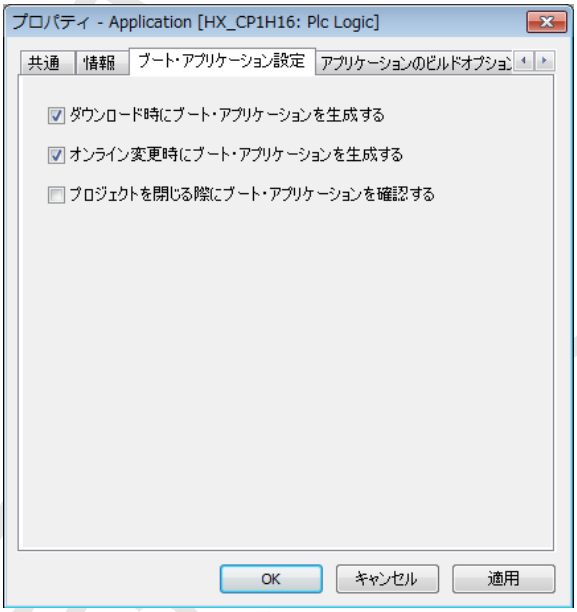
2.10 Boot application

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.



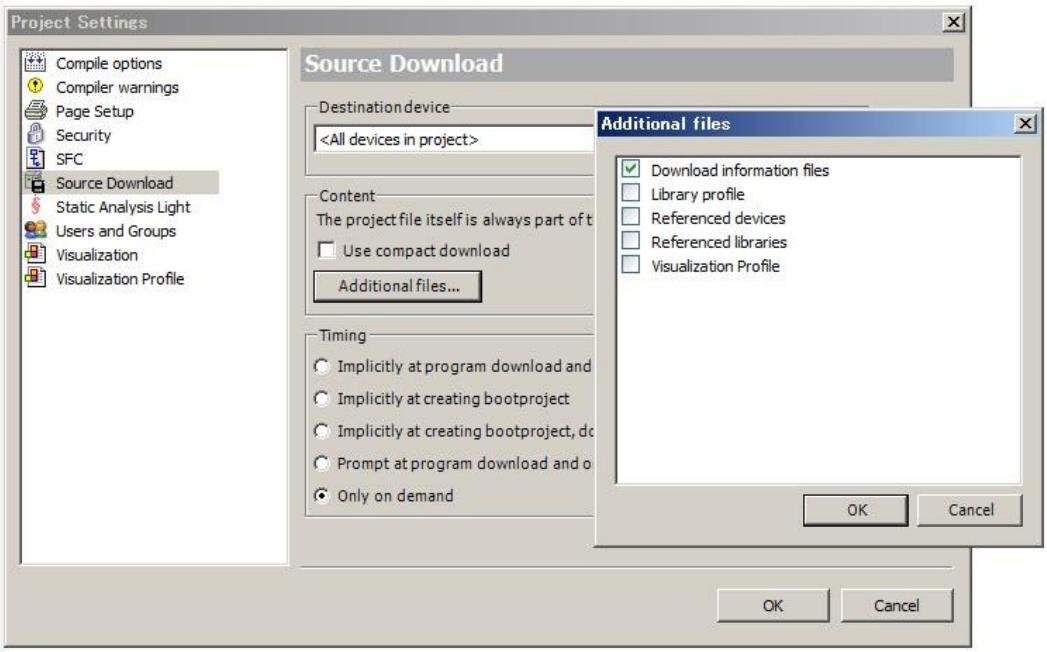
*: Optional

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

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.



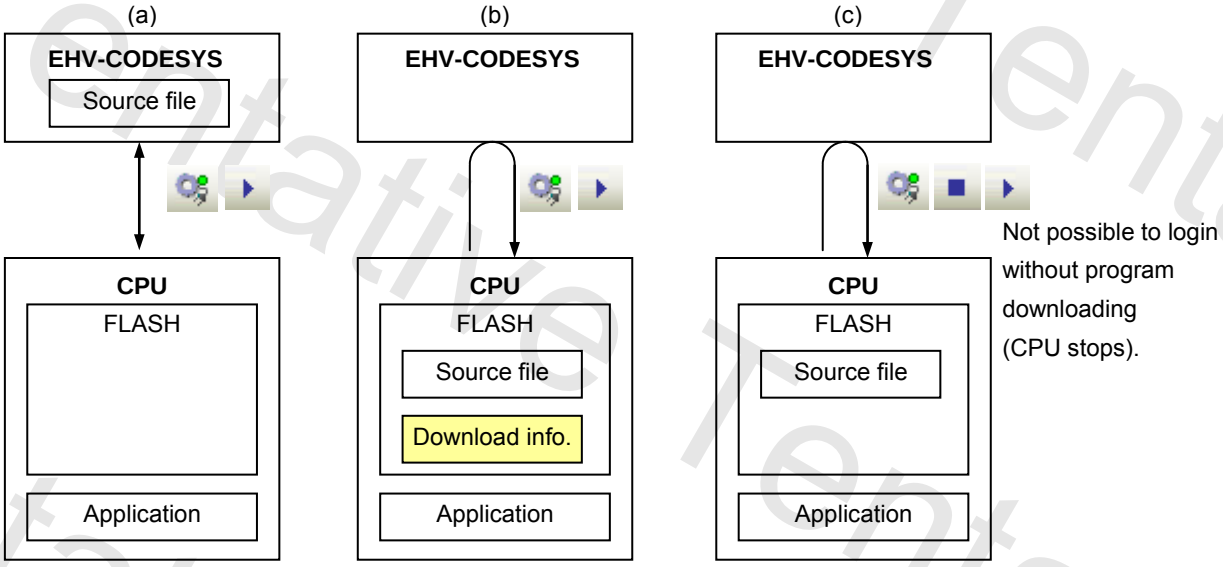
Download information files

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

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

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

(c) Online change from PC without source file to CPU with source file. ➔ Source upload and Login, then program download is required because HX-CODESYS is not able to verify program identity. 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 (no effect)	Stop
Run with HX-CODESYS	Stop (no effect)	Run
Reboot PLC (Cycle power)	Stop	Run *

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

HX-CPU は、運転開始時に基本/増設ベース上の I/O モジュールに対しハードウェアリセットを行います。このため、停止中に出力する設定/条件の場合でも運転停止→運転開始のタイミングで 1 タスク周期分、出力が OFF します。

HXM-013
RUN 開始から初期値データを出力へ反映するまでの時間が長い

Reset

When CPU detects a serious error called “exception”, such as watchdog error, program execution stops. If EHV-CODESYS is connected, “Exception” indication blinks until this status is cleared. This exception status is cleared only by “Reset” operation. 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.

初期化

例外状態の解除及び変数の初期化だけでなく不揮発性メモリの内容をリセットする場合は「初期化」操作を行います。「初期化」には「リセット（PLC 初期化）」「デバイス [Device]をリセット（PLC 初期化）」の 2 種類がありそれぞれ動作が異なります。初期化される内容は、下表を参照してください。

「リセット（PLC 初期化）」は、複数のプログラムが PLC 内に存在する場合は現在アクティブに設定されているアプリケーションプログラムのみが消去されます。複数のプログラムすべて初期化する場合にはデバイスツリーの[Device]の右クリックメニューの「デバイス [Device]をリセット（PLC 初期化）」を実行してください。この際、Visu フォルダも同時に初期化されますのでご注意ください。

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

（Initialize PLC）							
Reset origin device [Device]	-	-	-	-	-	-	-

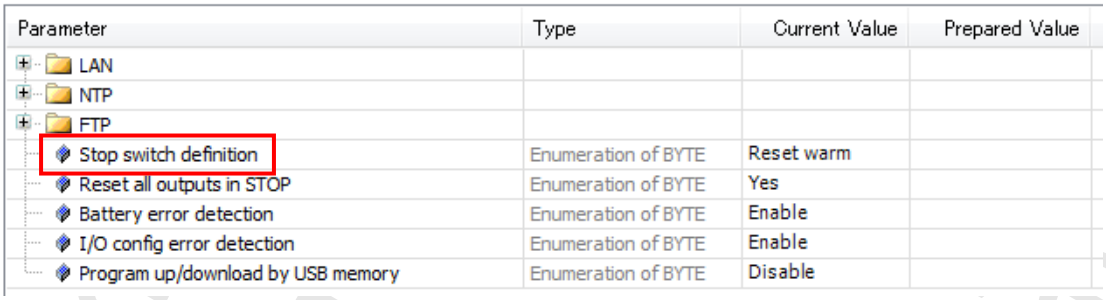
X = maintained, - = initialized

Note

転送したプログラムを HX-CODESYS の[Device]-[ファイル]の操作で名称変更しプロジェクトファイルのアプリケーション名称と不一致にした状態では「リセット（PLC 初期化）」「デバイス [Device]をリセット（PLC 初期化）」共に動作しません。

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 of original “Stop” for existing Hitachi PLC.

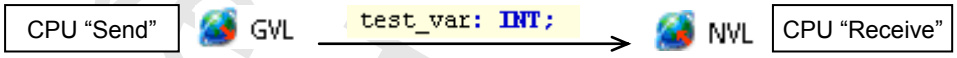


2.13 Global network variables

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

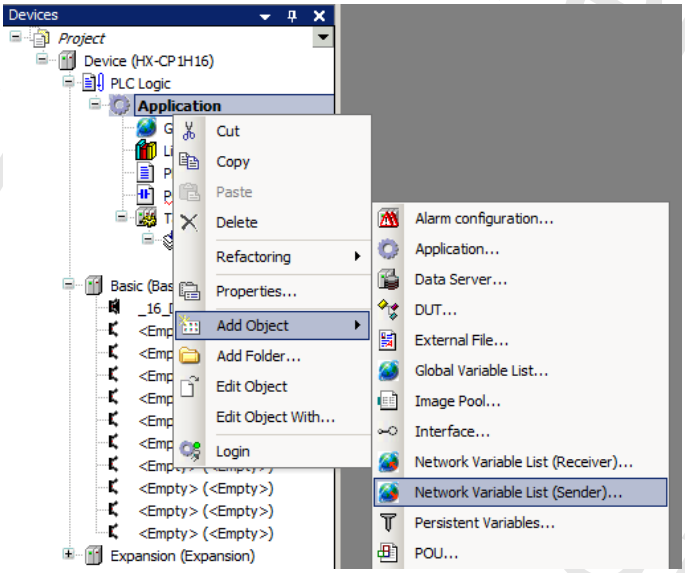
How to configure?

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

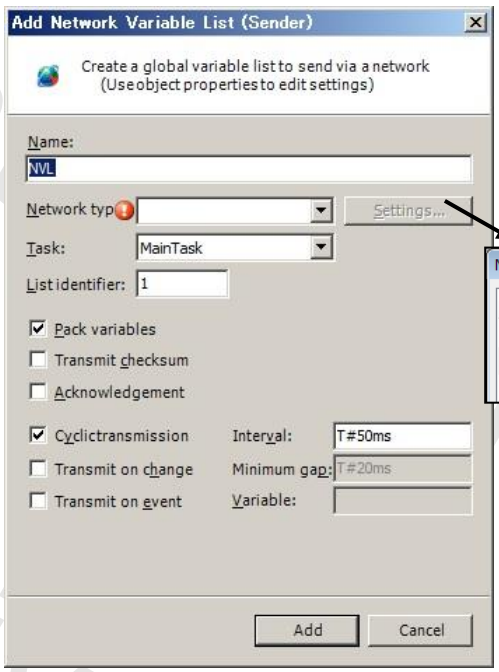


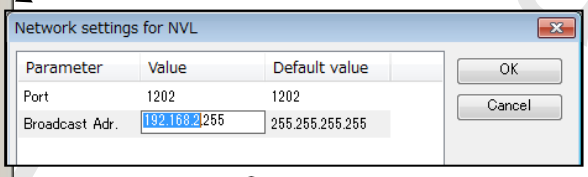
[CPU "Send"]

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



Network type: Choose "UDP".
Settings : Set broadcast address
Task: Choose any one task. The variables are sent at the end of a task cycle.

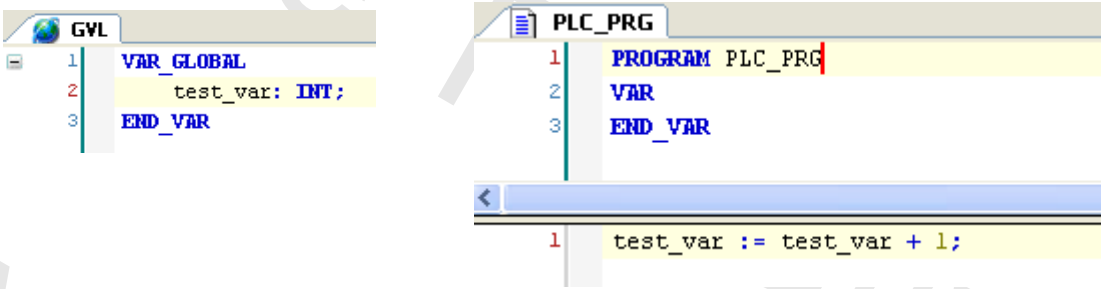




ETH1	192.168.0.255
ETH2	192.168.1.255
ETH3	192.168.2.255

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.

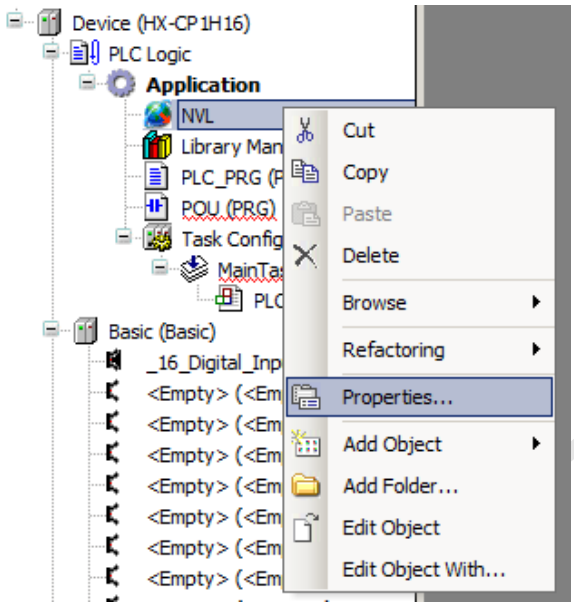
In this sample, one global variable “test_var” is defined and one-line program is written in POU as below.

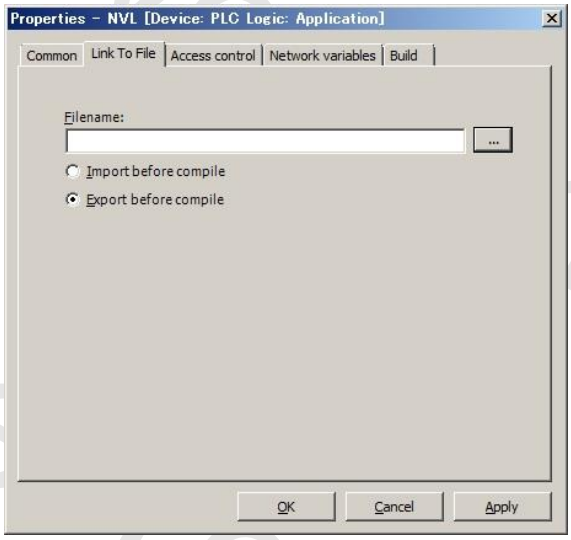


送信側のパラメータの設定が完了したら、作成した変数リスト<NVL>を右クリックして「プロパティ」を選択します。「Link To File」タブにてエクスポートするファイルを作成します。ネットワーク変数として 255 バイト以上の STRING/WSTRING 型は使用できません。尚、「Network properties」タブにて、送信側のパラメータを修正することが出来ます。

Note

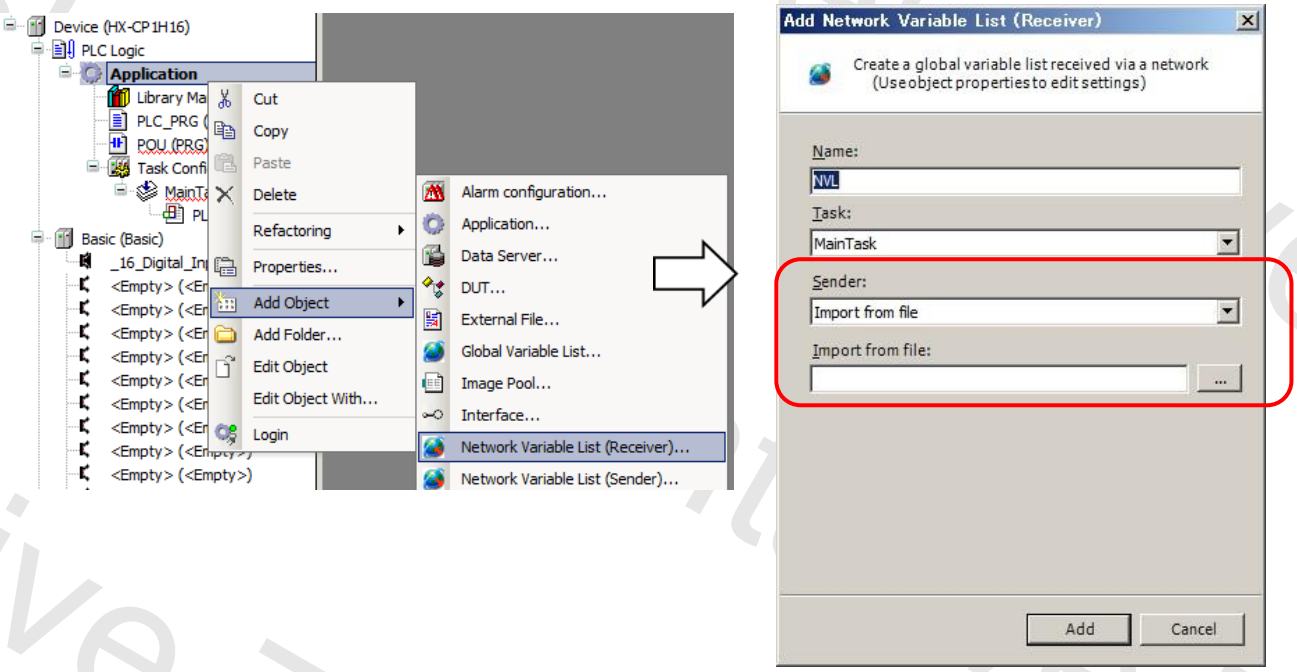
プロパティはオンライン中でも設定を変更することが可能ですが、HX-CPU に設定がダウンロードされないので、ログアウトした状態で設定を変更し、その後ログインしてダウンロードを行ってください。





[CPU “Receive”]

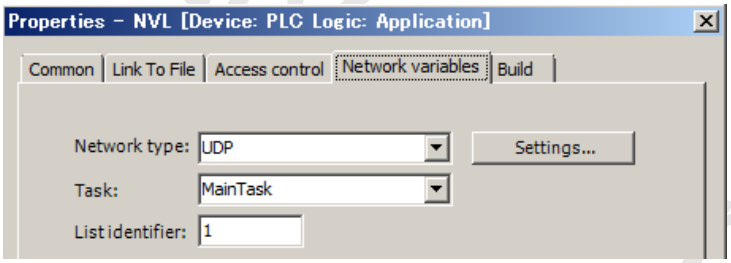
The next step is configuration for receiving CPU. Right click on “Application” of Receive-CPU and choose “Global Network Variable List...” Be sure to check if Sender is properly set as configured list above.



Configuration is completed for both send and receive-CPU.

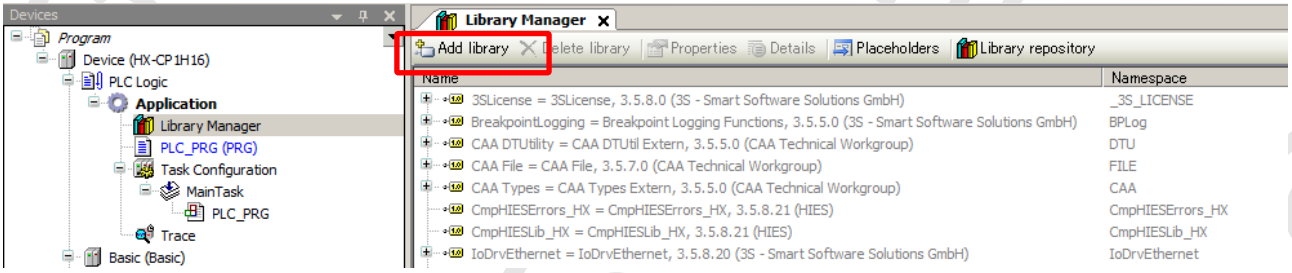
Note

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



2.14 Library

In order to read/write HX-CPU's specific information, following libraries are available. Add necessary CmpHIESLib by choosing "Add library" as shown below.



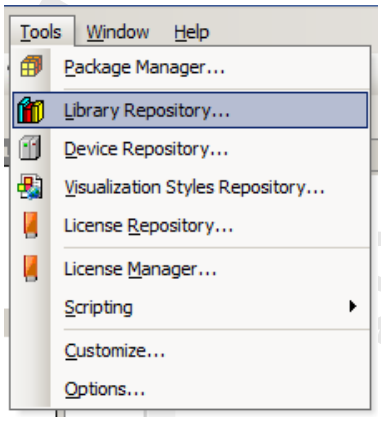
Libraries as shown below are loaded automatically when new project is opened.

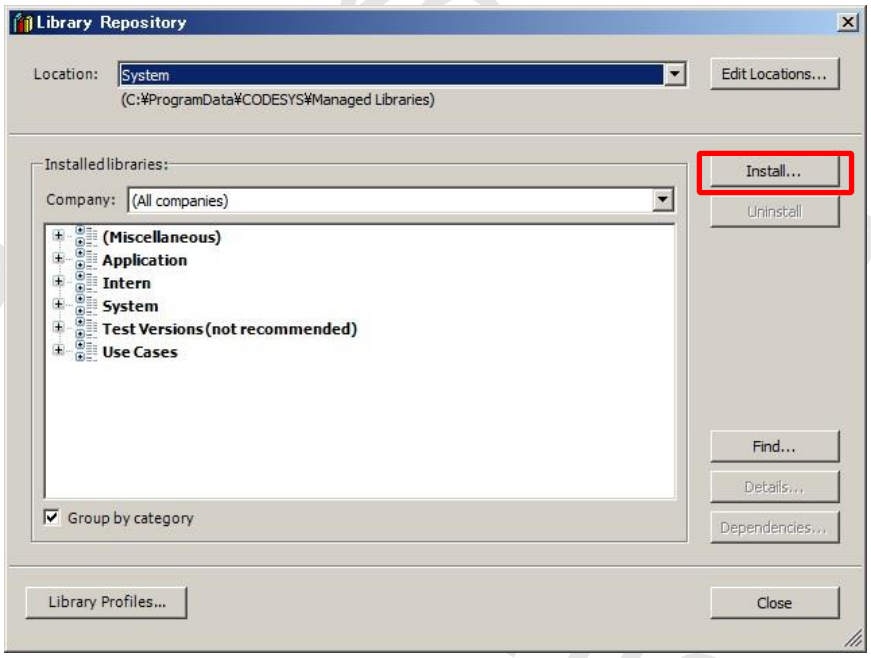
Loaded libraries automatically

Name of Library	Note
IoStandard	
3SLincense	
Util	
CAA DTUtility	
CAA File	
SysCom	
CAA Types	
CmpHIESLib_HX	Note 1
CmpHIESErrors_HX	Note 1

Note 1: Use libraries that the end characters are "HX". The other libraries are for EHV+ series.

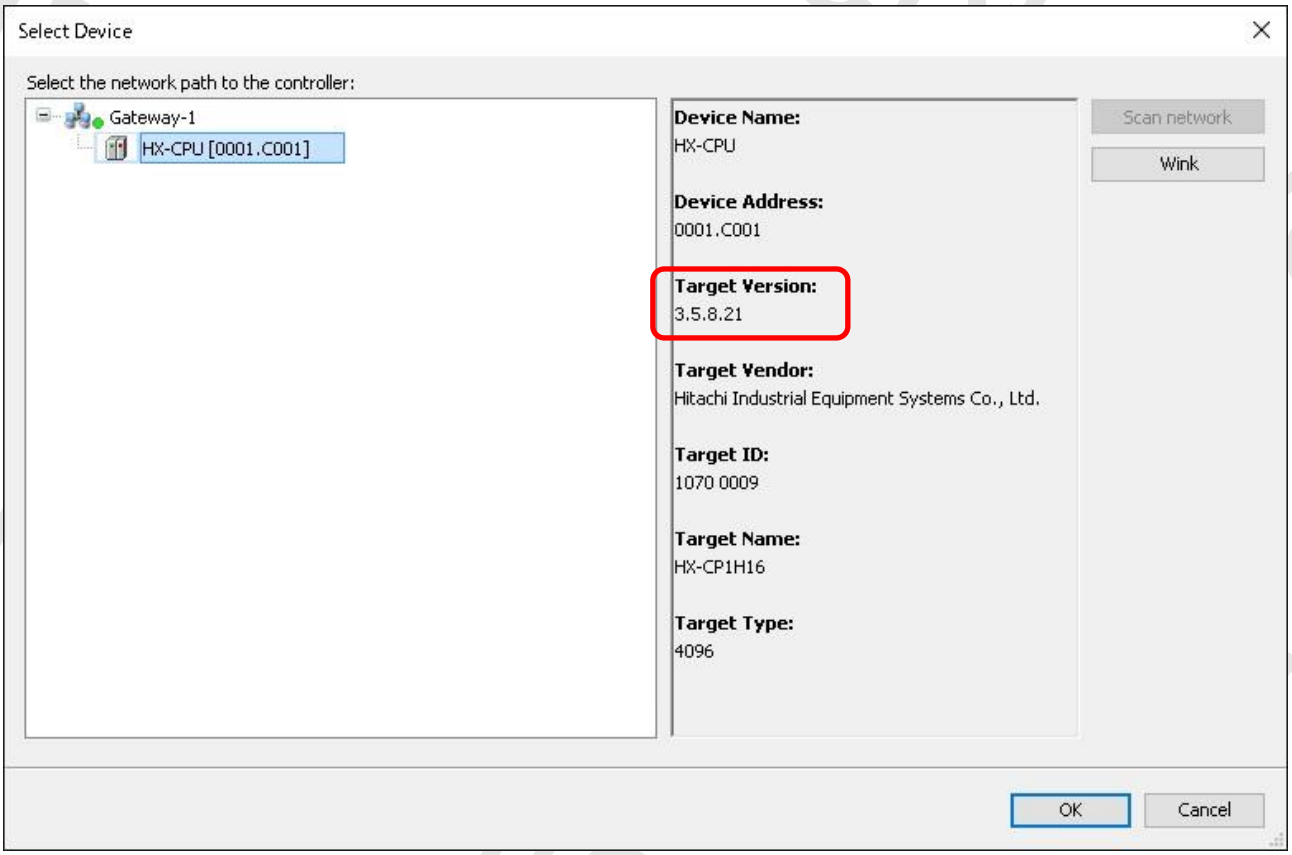
If these libraries are not found in the library list as above, install library by choosing [Tools]-[Install library...].





2.15 Version

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

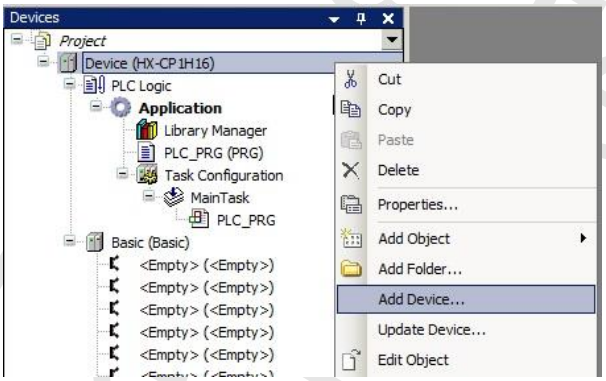


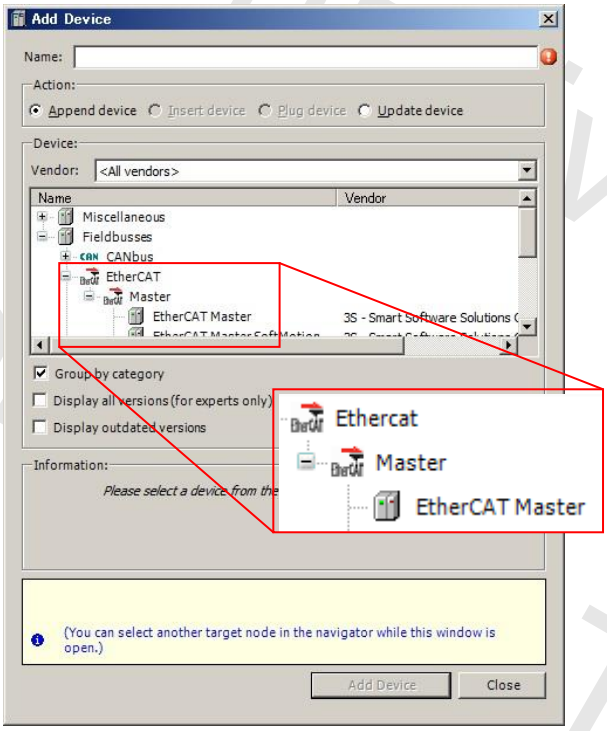
Chapter 3 Communication function

3.1 EtherCAT master

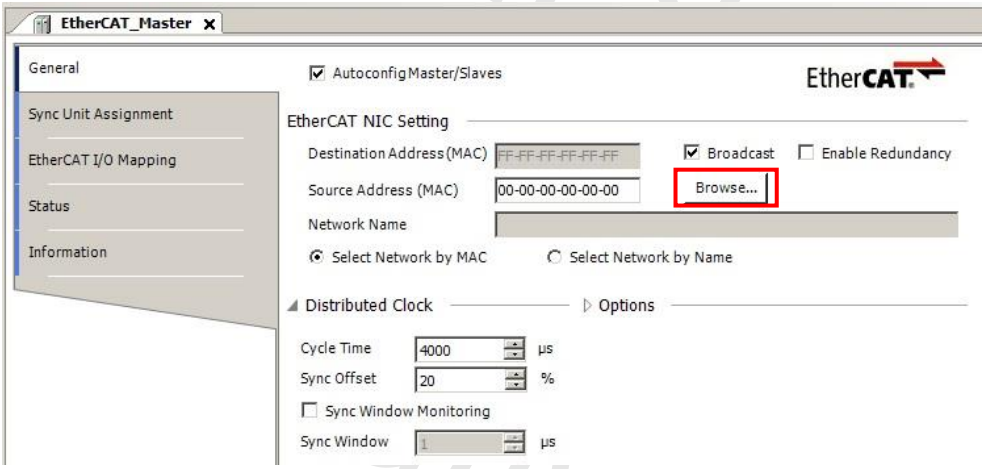
3.1.1 Configuration

Right click opens “Add Device…” on “Device”.
To click “EtherCAT Mater” on “Add Device” window and click “Add Device” button.

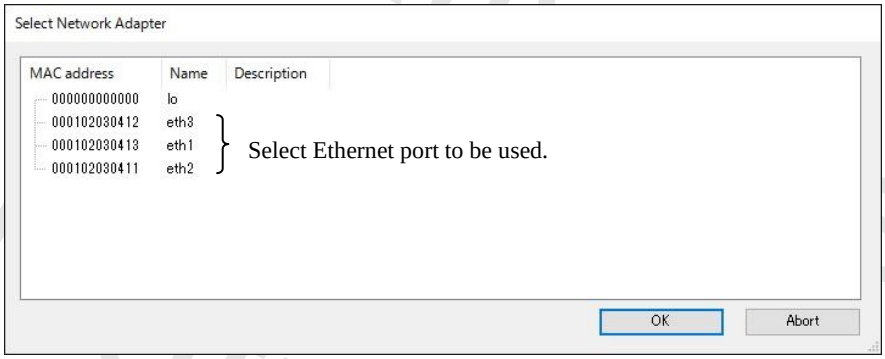




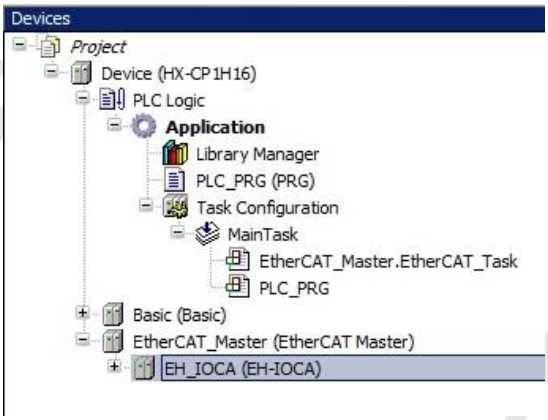
To configure Ethernet port, following window is opened by double clicking “EtherCAT Master (EtherCAT Master)”. And select Ethernet port to be used for EtherCAT Master by clicking “Browse...”.



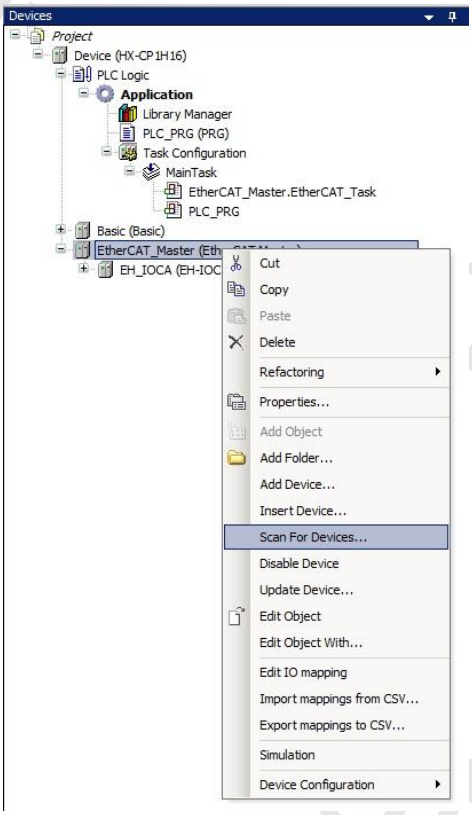
*) Enable Redundancy is not supported.

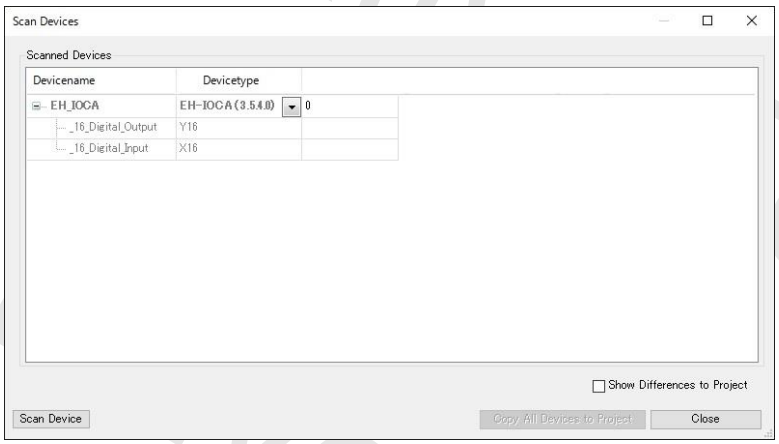


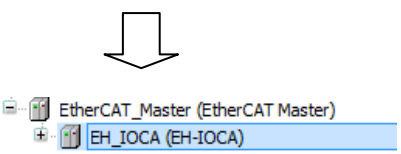
Usable devices are shown on “Add Device” window by clicking “EtherCAT Master” of Device tree. Press “Add Device” button after selecting using slave device.



If the slave is connected actually, it is possible to search the device during online after doing communication configuration according to the description Chapter 3.8. After right clicking “EtherCAT Master (EtherCAT Master)” and cliking “Search Device”, the connected device can be found. The found device can be added by clicking “Copy to Project”.



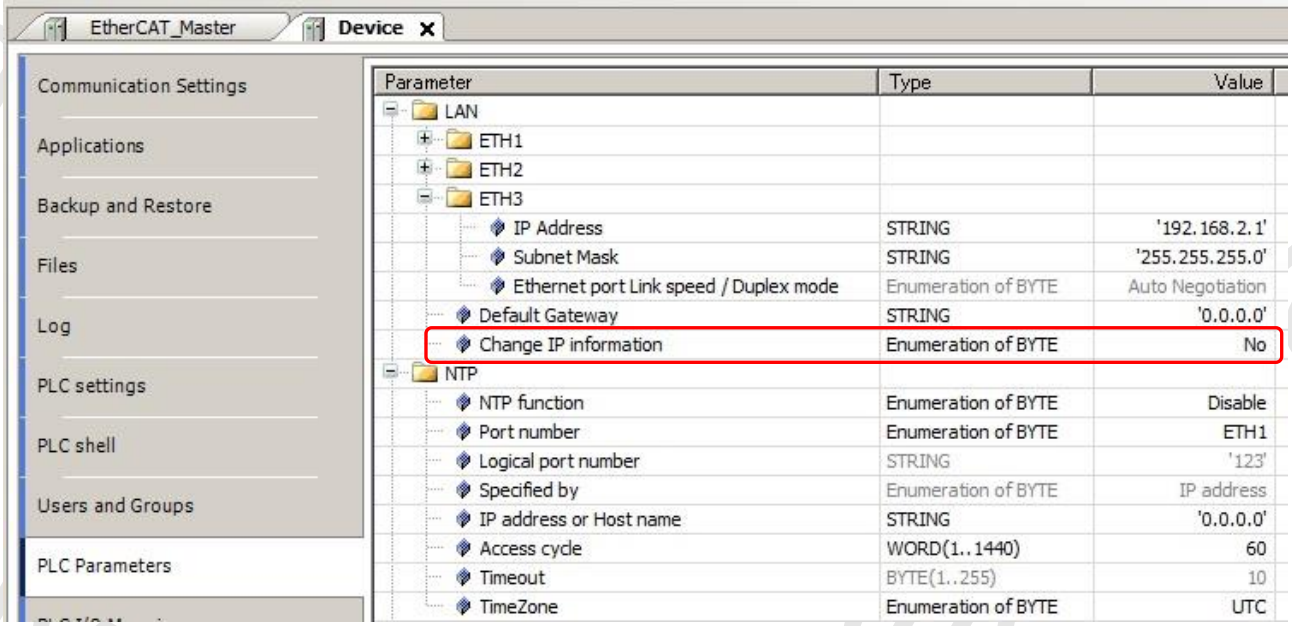




“Add Device” window will not appear if using slave device “EtherCAT slave information (ESI) file”(XML format) is not installed. If it is not shown, install ESI file at main menu [tool]-[Device Repository] after downloading ESI file from each slave vender.

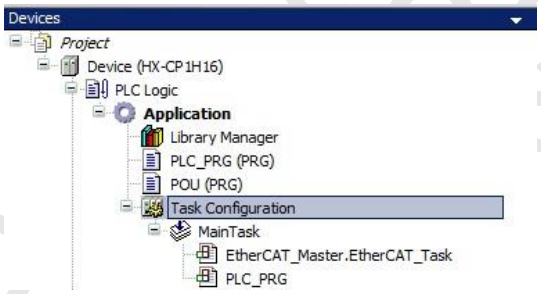
3.1.2 Ethernet communication configuration

Select “Yes” at “Change IP information”. After logging on, this configuration information is uploaded to CPU but it is required PLC power on again to valid this configuration.

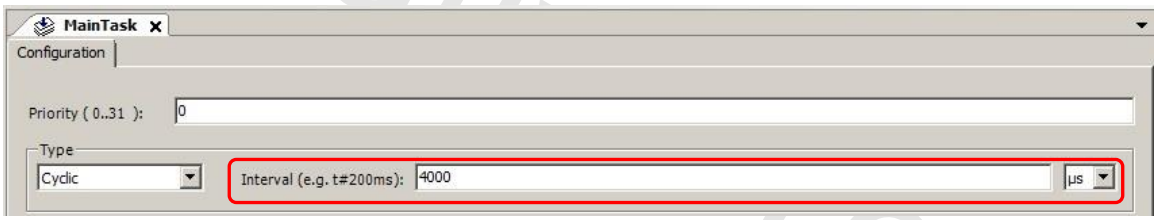


3.1.3 Cycle of EtherCAT task

EtherCAT Master task is created automatically separate from MainTask by adding EtherCAT master on Device Tree. EtherCAT_Master configuration window will open after select “Edit Object” by double clicking or right clicking on “EtherCAT_Master” task.

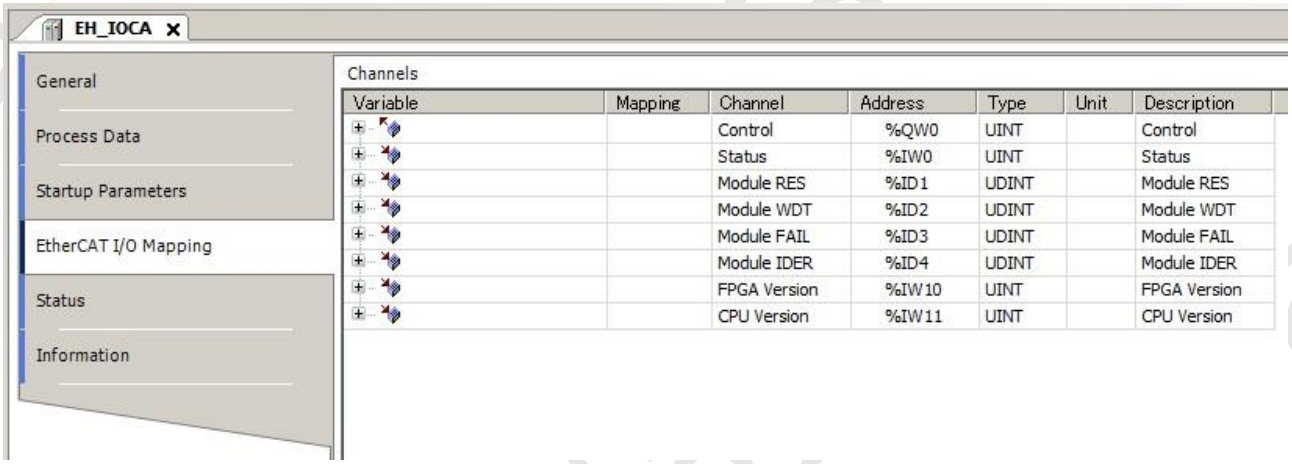


“Cycle” should be 1ms or more due to HX-CPU execute all processing with single microprocessor. If this value is too small “25 error” (Microprocessor Overload) will be detected. This cycle value depends on user program size or number of I/O module, but it is required to define more than 1ms for this value even if small program size or less number of I/O modules.



3.1.4 Programming

I/O address of slave device are shown on below “EtherCAT I/O mapping” tab. Programming can be done after setting some variable name at this variable field as same as other external I/O.



Variable	Mapping	Channel	Address	Type	Unit	Description
		Control	%QW0	UINT		Control
		Status	%IW0	UINT		Status
		Module RES	%ID1	UDINT		Module RES
		Module WDT	%ID2	UDINT		Module WDT
		Module FAIL	%ID3	UDINT		Module FAIL
		Module IDER	%ID4	UDINT		Module IDER
		FPGA Version	%IW10	UINT		FPGA Version
		CPU Version	%IW11	UINT		CPU Version

Caution

- If power PLC on keeping RUN/STOP switch position RUN, due to EtherCAT master executes configuration process for slave device, starting timing of I/O refresh of EtherCAT slave will be delayed approximately 1 second comparing other standard external I/O refresh timing. If this delay impact system operation and it is required to avoid this delay, dedicated bit of “Instance name.Finised of EtherCAT master” is recommended to use in program. This bit will turn true after completing configuration of TherCAT master. It is shown as sample program described by using ST language in below.
- EtherCAT slave refresh cycle will delay with task cycle time maxim comparing I/O refresh cycle on basic or expansion base unit, due to EtherCAT communication is executed asynchronous “EtherCAT_Master” task.

Sample program

In case of Instance Name of EtherCAT master is “EtherCAT_Master”

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

Program described this after will not be executed if during bit of EtherCAT_Master.xConfigFinished is FALSE (OFF).

3.1.5 Wiring

(1) Cable

Category 5 and more STP (Twisted pair with shield) cable is recommended.

(2) Hub

Standard hub can’t be used for EtherCAT network. EtherCAT dedicated hub is required to use if hub is needed. (Ex. Model CU1128 made by Beckhoff)

Caution

When EtherCAT master function is used, the EtherCAT port is not used for the other EtherCAT master function. If Gateway function (Connection HX-CODESYS or HMI), Modbus-TCP function, Global network variable function etc. are used together with EtherCAT master function, EtherCAT performance will be lower or stopped.

3.2 Modbus-TCP / RTU

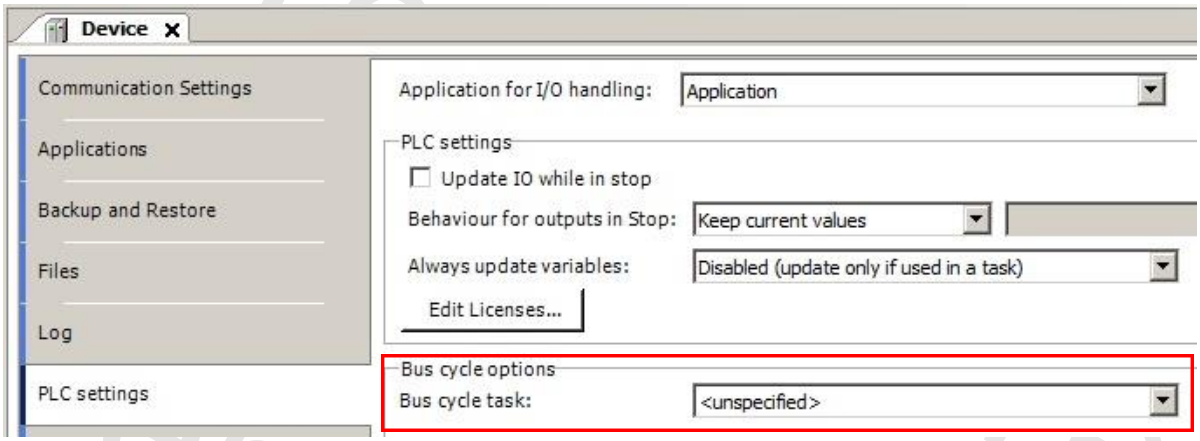
3.2.1 Overview

Supported function code by HX series are shown in below.

Hexadecimal	Decimal	Function code	Modbus-TCP		Modbus-RTU	
			Master (Client)	Slave (Server)	Master	Slave
0x01	01	Read coil status	X	X	X	NA
0x02	02	Read input relay	X	X	X	NA
0x03	03	Read keep register	X	X	X	X
0x04	04	Read input register	X	X	X	X
0x05	05	Write single coil	X	X	X	NA
0x06	06	Write single register	X	X	X	X
0x0F	15	Write multiple coils	X	X	X	NA
0x10	16	Write multiple register	X	X	X	X
0x17	23	Read/Write multiple register	X	X	X	X

X: Supported, NA: Not supported

Modbus communication processing is executed at “Bus Cycle Task” specified PLC configuration tab of Device.
Any task can be specified for “Bus Cycle Task”. If <undefined> is selected as in below screen, most shortest cycle task is “Bus Cycle Task” automatically.

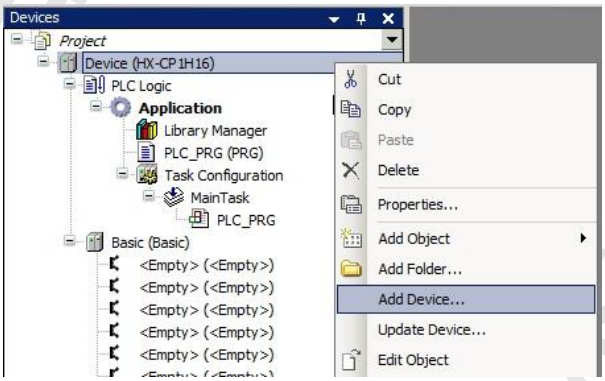


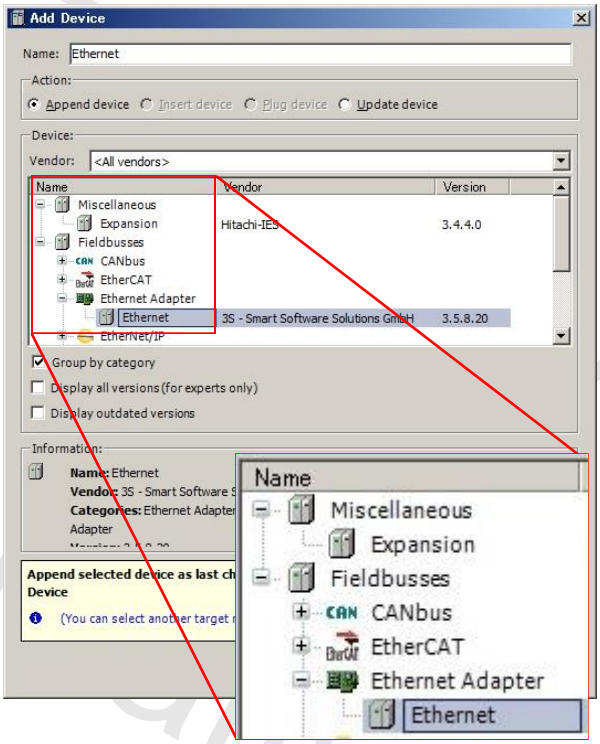
Caution

Issuing Broadcast Query of Modbus-TCP master and/or Modbus-RTU master are not supported.

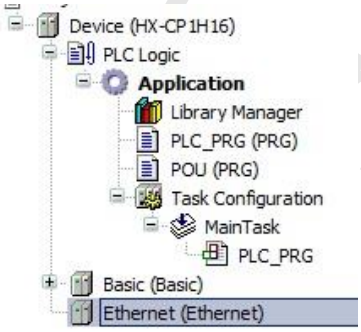
3.2.2 Modbus-TCP Master (Client)

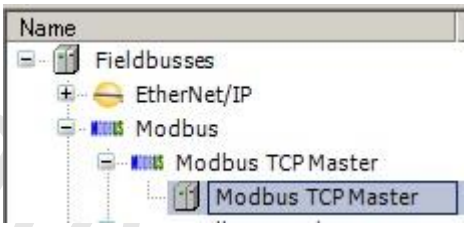
Right click opens “Add Device…” on “Device”. To click “Ethernet” on “Add Device” window and click “Add Device” button.



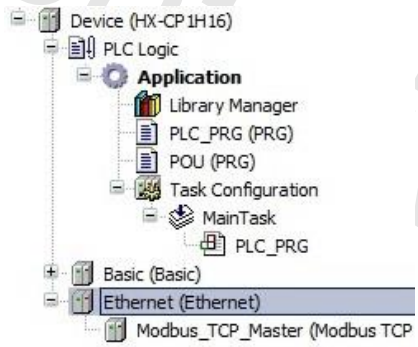


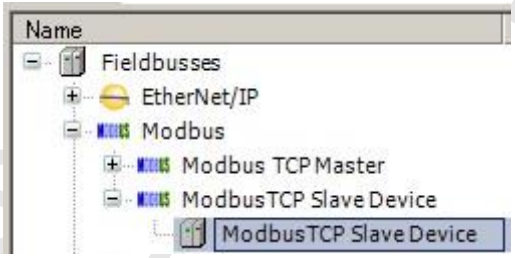
Click “Ethernet” of Device tree by keeping “Add Device” window open, usable devices are shown on “Add Device” window. Select “Modbus TCP master” and press “Add Device” button.

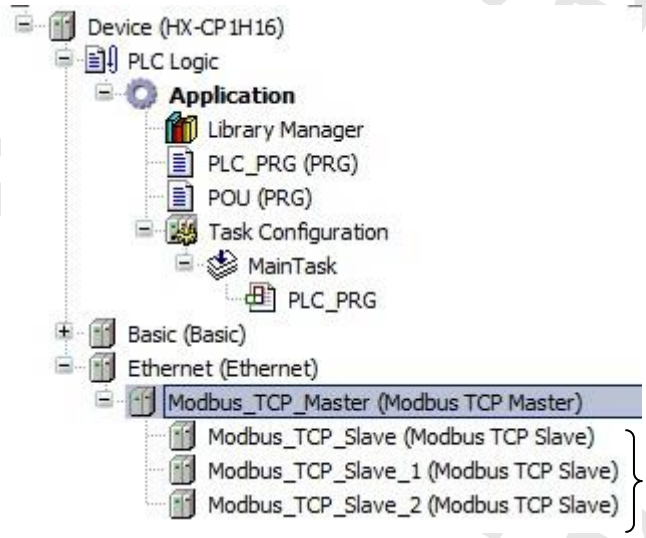




Click “Modbus_TCP_Slave Device” by keeping “Add Device” window open, “Modbus_TCP_Slave Device” is shown on “Add Device” window and select this and press “Add Device” button. Add “Modbus TCP slave” according actual system. In case of three slave units are used, three “Modbus TCP slave” are shown under master.

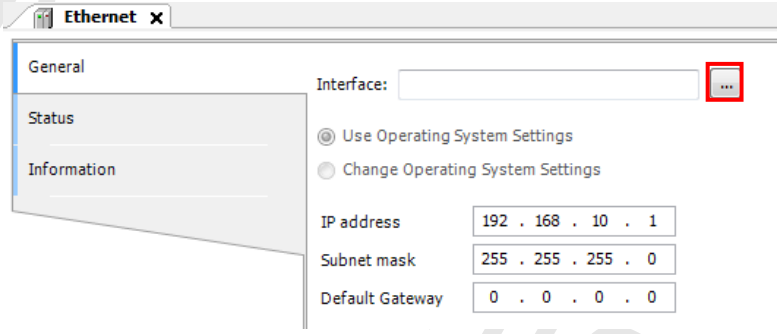


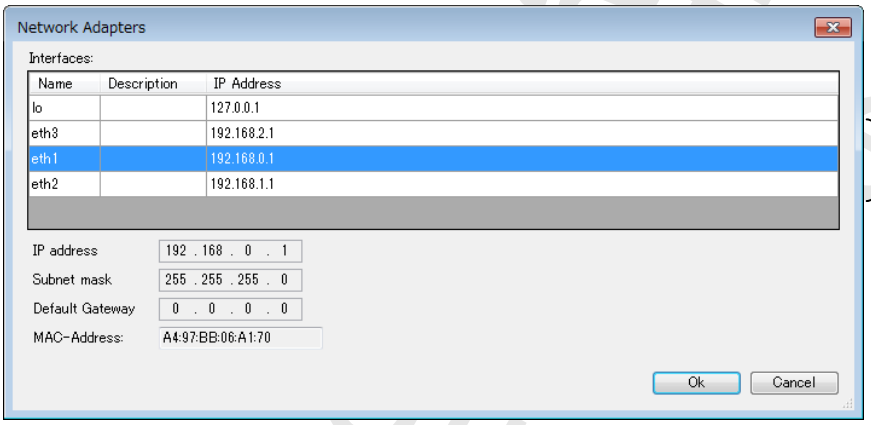




Configure these according to Control object.

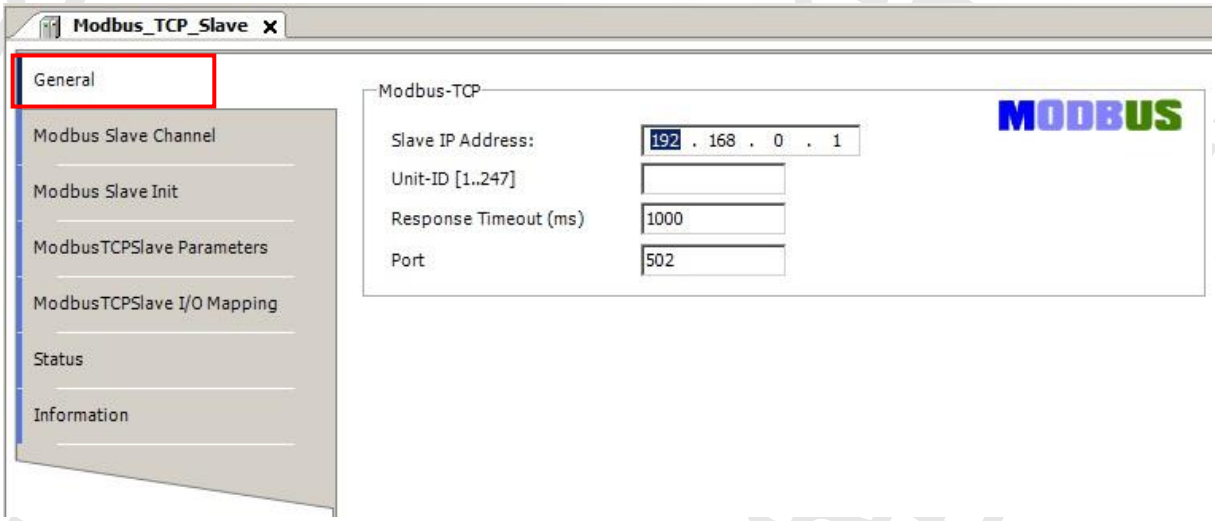
Select Ethernet port to be used. After configuring communication, double clicking "Ethernet (Ethernet)" opens following window. Click "…" to select Ethernet port for Modbus-TCP usage.



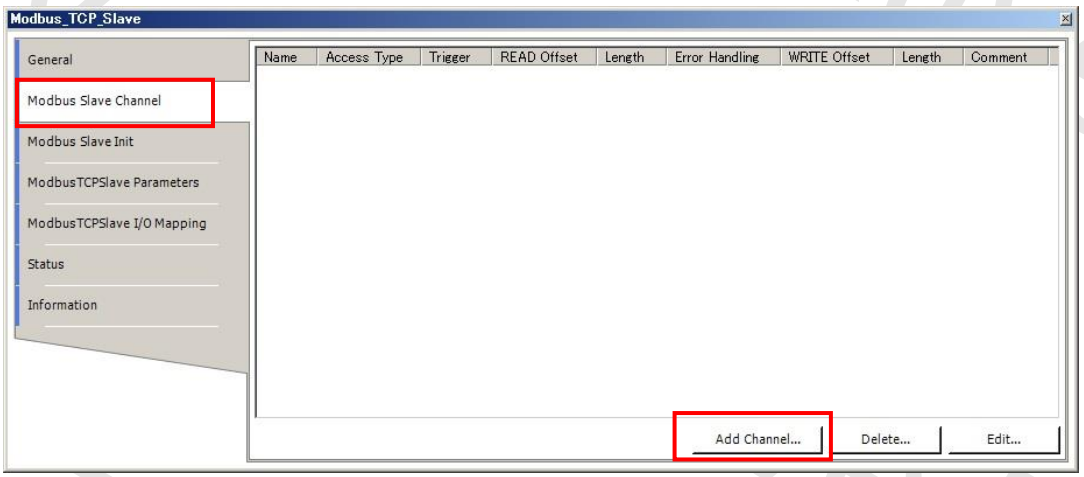


Select Ethernet port to be used.

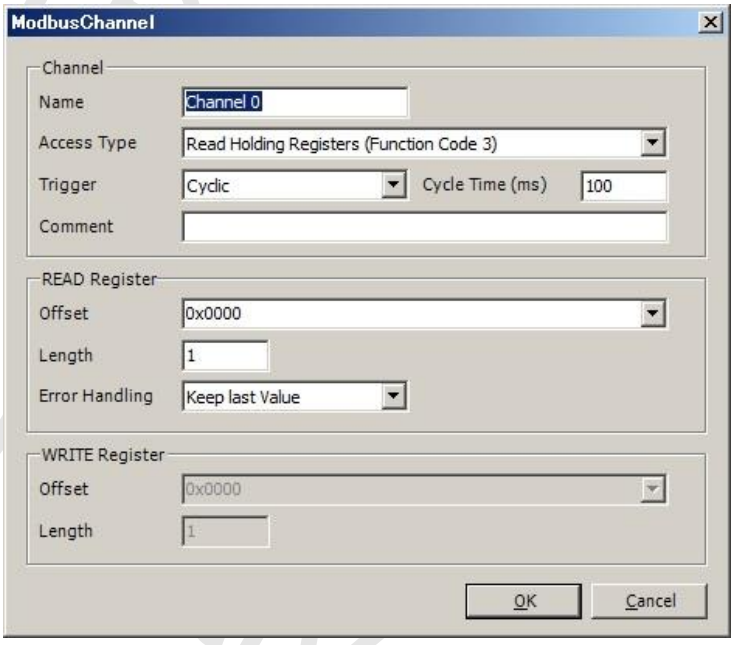
Issueing Function Code configuration are done by each slave. Following configuration windows are opened by doubleclicking for target slave. Configuration of IP address, response timeout and port number are done at this window.Unit-ID is only required to configure when Modbus gateway (Converter of Ethernet/Serial) is used.

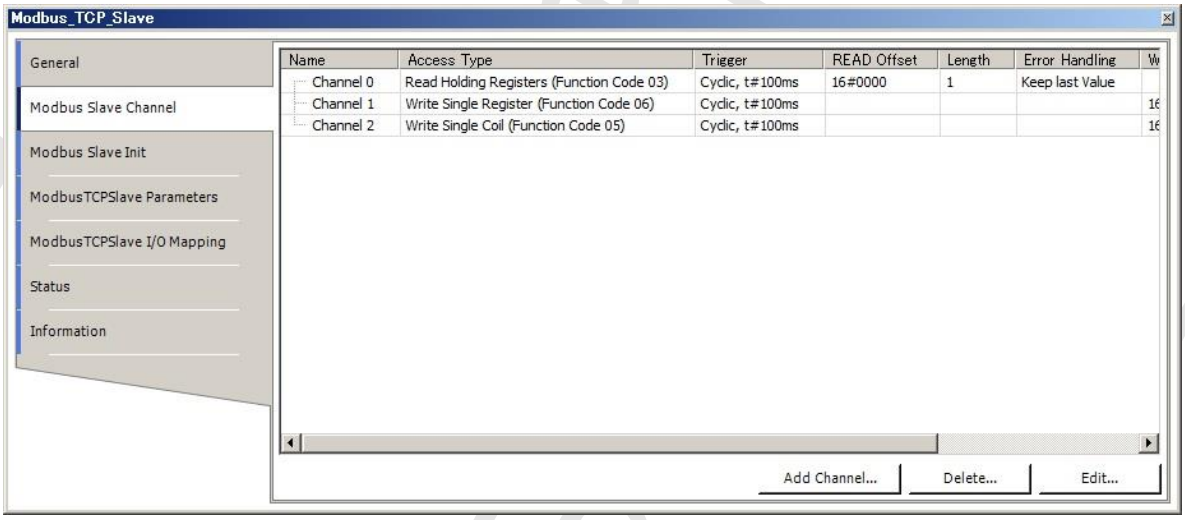


Sending Function Code are configured by “Add Chanel” button of “Modbus slave channel”.

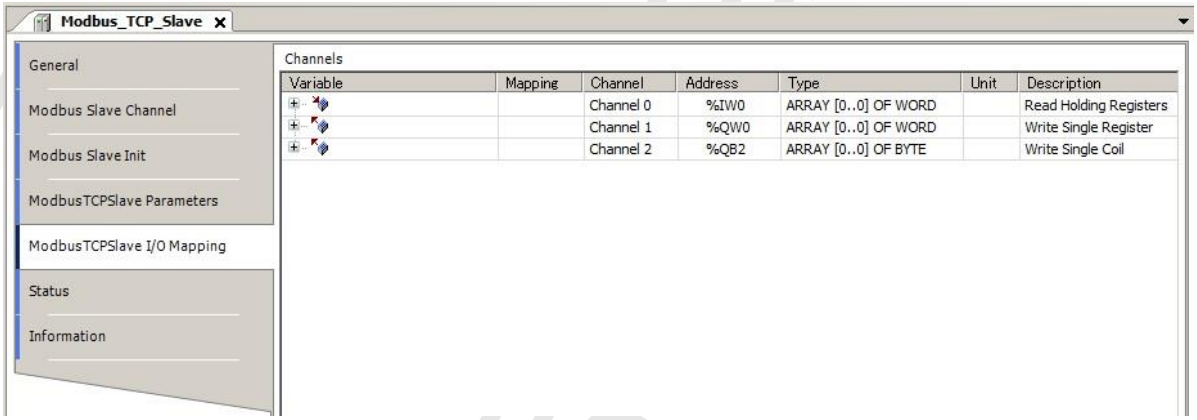


Configure each Function Code according to following procedure. When “Trigger” configuration is selected as “Rising edge (Rising Edge)”, Trigger variable (BOOL) is assigned as %QX address automatically.





Reading/writing data are assigned %IW or %IQ address shown in “ModbusTCPSlave I/O Mapping” tab. The data read from slave is Input area (%IW) and the data wrote into salve is Output area (%QW).

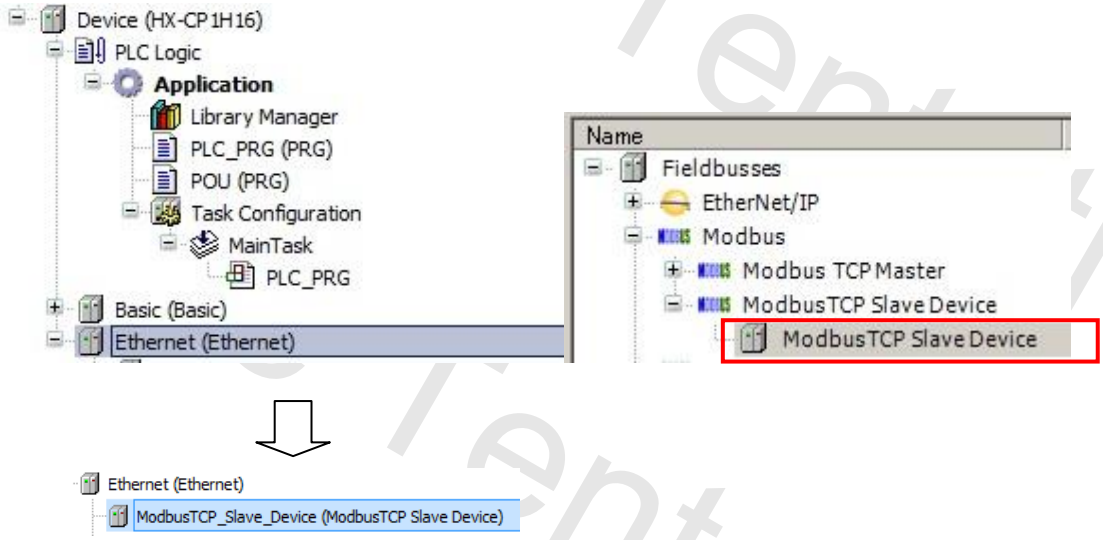


Caution

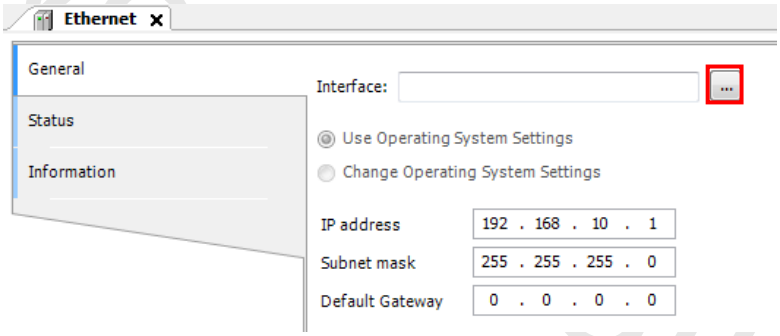
- Modbus Master sends is 1 channel Request dada every 1 cycle. Therefore in case of trigger set is "Rising edge (rising edge)" and if trigger variable changed so high-frequency, previous request data is not completed to send and it will be the timing for the next trigger and sometimes trigger may not be recognized correctly. In case of the time from the start of request data to the end of response data is T1, the rough calculation of inhibited time not to change trigger bit is described T1+T2+...+Tn.

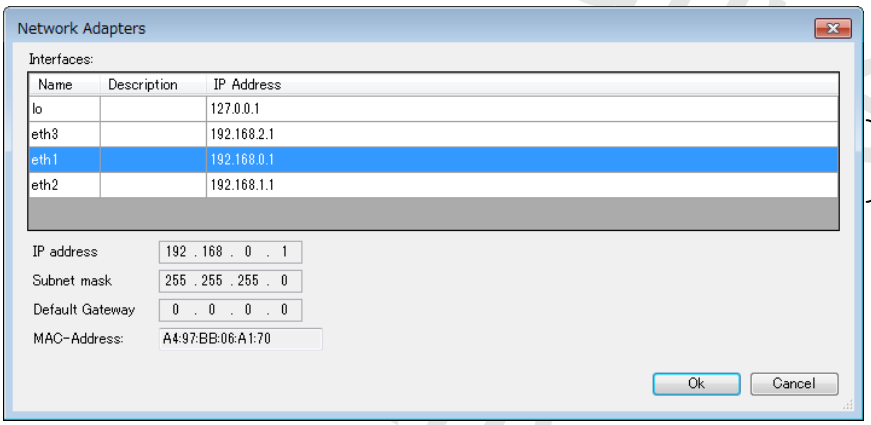
3.2.3 Modbus-TCP Slave (Server)

Add "Ethernet" same as Modbus-TCP master, add "Modbus TCP Slave Device" at "Add Device".



Select Ethernet to be used. After communication setting, following window appears by double clicking “Ethernet (Ethernet)”. Select Ethernet port for Modbus-TCP by clicking “…”.





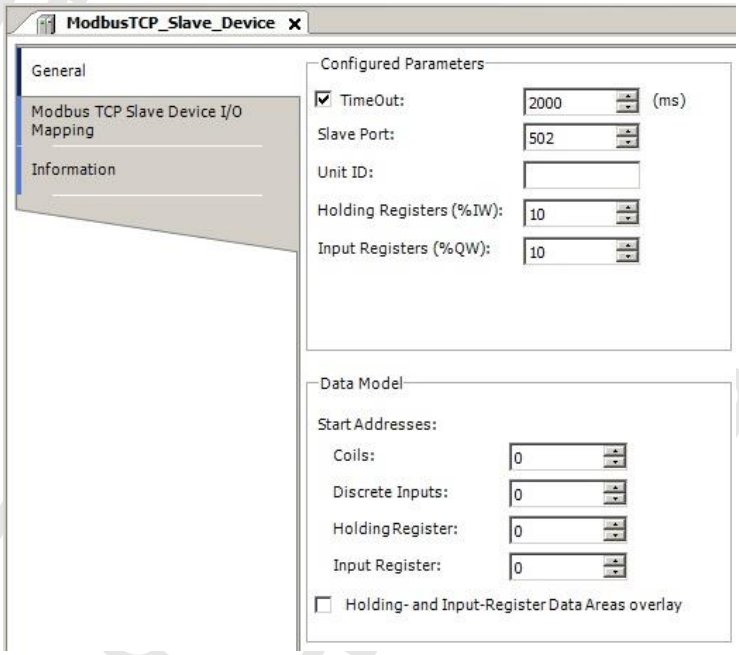
Select Ethernet port to be used.

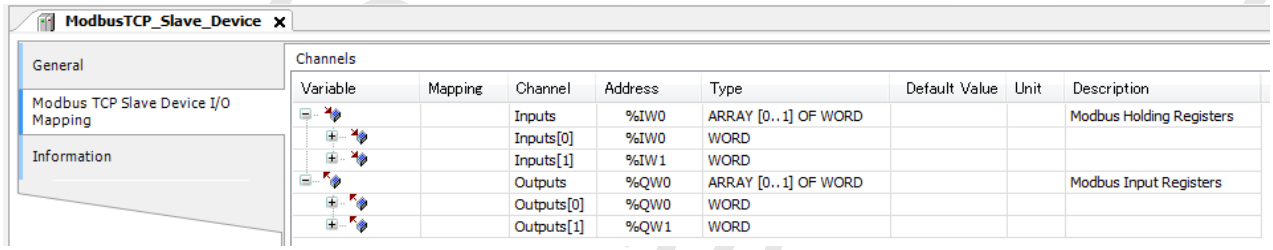
Caution

- Even it is possible to send response from other port (for example, in case of registering eth1, it means eth2 or eth3) registered as Modbus-TCP slave, it is recommended to execute communication by the port registered as Modus-TCP slave.
- Even it is possible to register eth1 – eth3 for Modbus-TCP slave, mapping table are not independent from others and are shared with others. Therefore, it is recommended to use Modbus-TCP slave with one port. Duplicated Modbus-TCP slave is detected as error, if several ports are used as Modbus-TCP slave.
- It will take approximately 1 minute to establish connection again, if warm reset is executed during establishing connection with Modbus-TCP master.

Open configuration window by double clicking “Modbus TCP Slave Device” and set needed parameters. However, do not change from 502 for slave port due to it is fixed.

Mapping table is created “ModbusTCP Slave Device I//O Mapping” tab according to configured size for Holding Registers (%IW) and Input Registers (%QW).

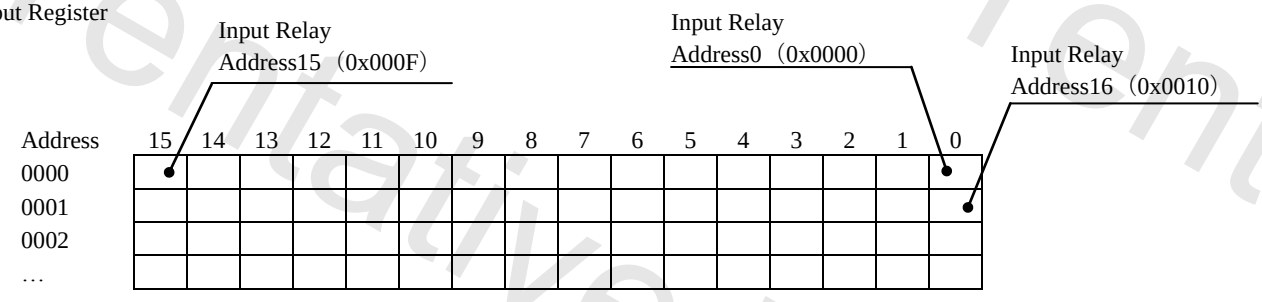




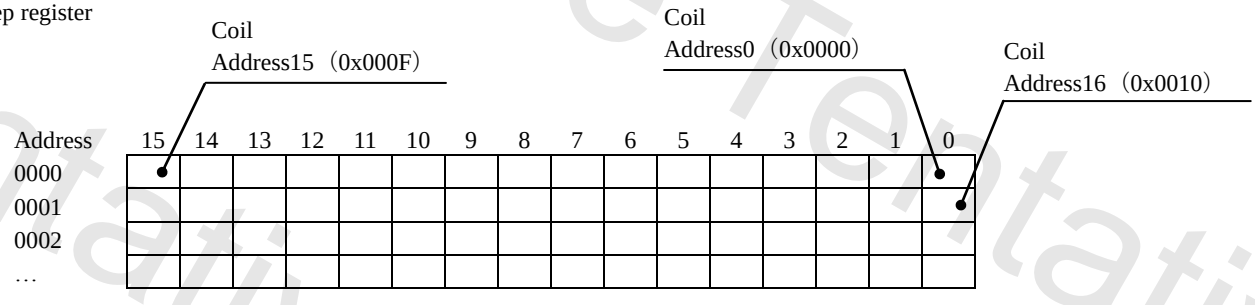
Each registers mentioned in below are physically same memory area.

- Input register (WORD) and Input relay (BOOL)
- Keep register (WOR) and Coil (BOOL)

Input Register



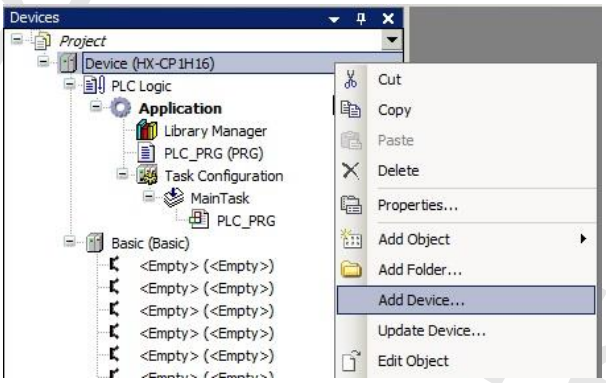
Keep register

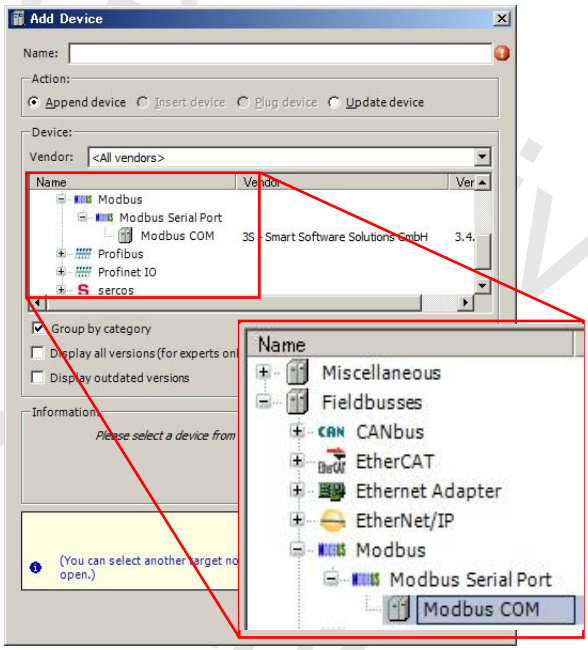


3.2.4 Modbus-RTU Master

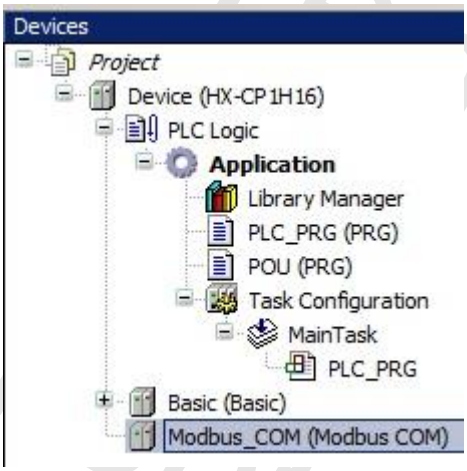
Select “Add Device” by right clicking “Device”.

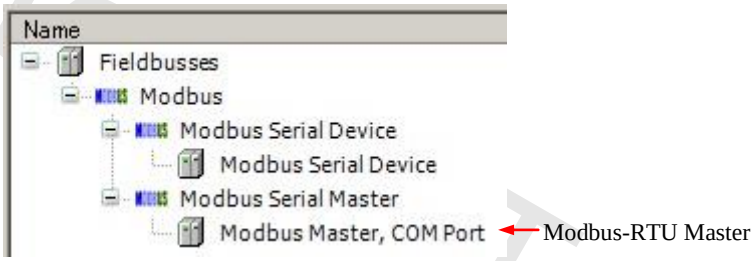
Select “Modbus COM” of “Add Device” window and press “Add Device” button.



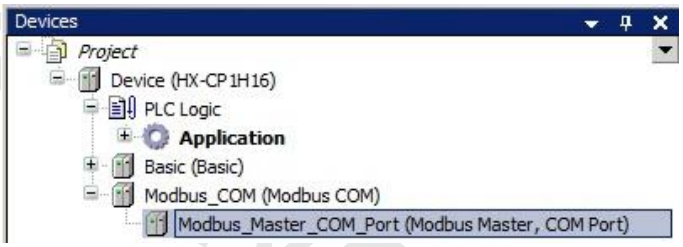


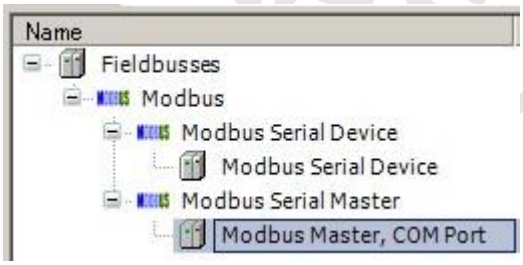
Keep “Add Device” window not to close, usable devices are shown in “Add Device” window by clicking “Modbus_COM” of Device Tree. Select “Modbus Master, COM Port” and press “Add Device” button.

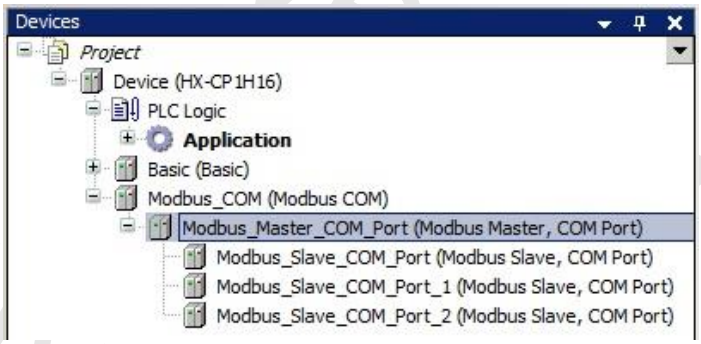




Keep “Add Device” window open and “Modbus Slave, COM Port” is shown by clicking “Modbus_Mater_COM_Port”, press “Add Device” button. Add “Modbus Slave, COM Port” according to actual system requirement. In case of three slave units are used, three “Modbus Slave, COM port” are under master.





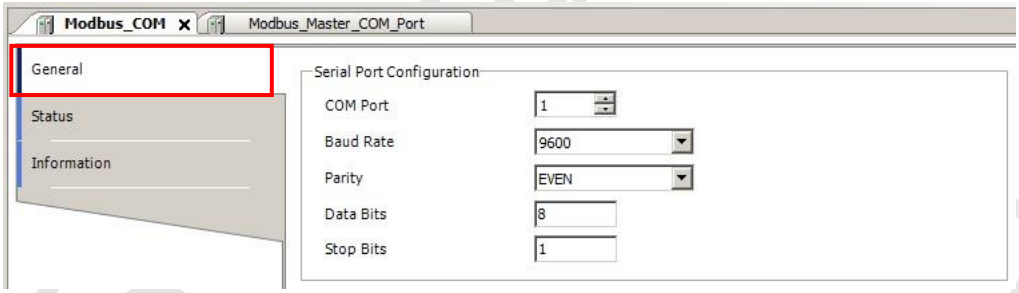


Configure these according to Control object.

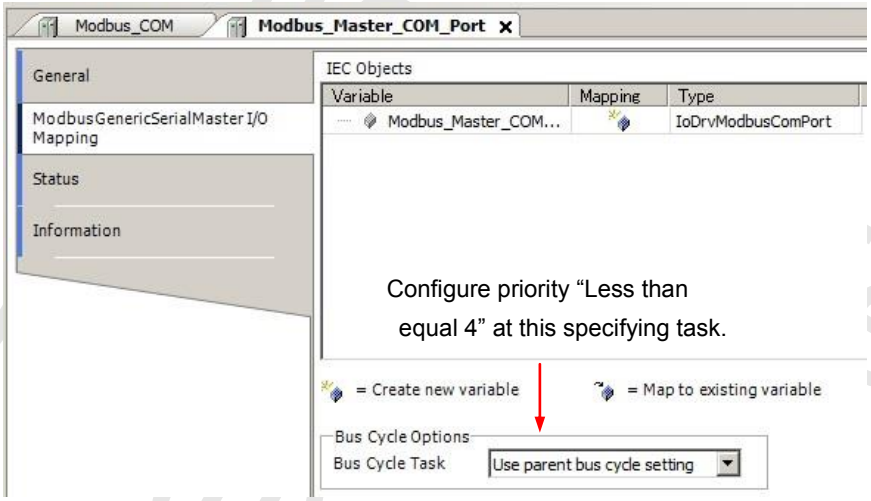
Caution

It is required correct response for the response from Modbus-RTU salve unit. If illegal response is received, there is an illegal operation risk.

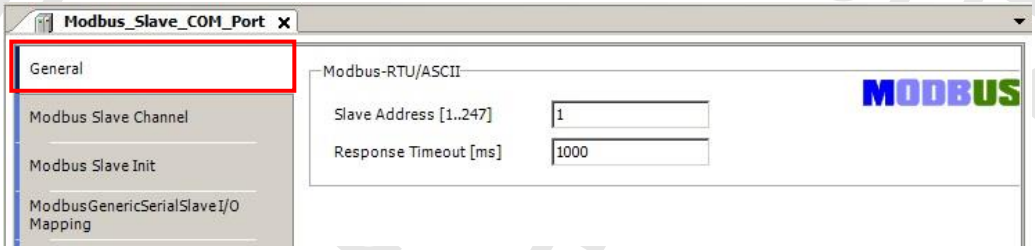
Configure port. Following window appears by double clicking “Modbus_COM (Modbus COM)”. Configure port according to slave to be used. Configure “1” for “COM Port” at HX-CPU side.



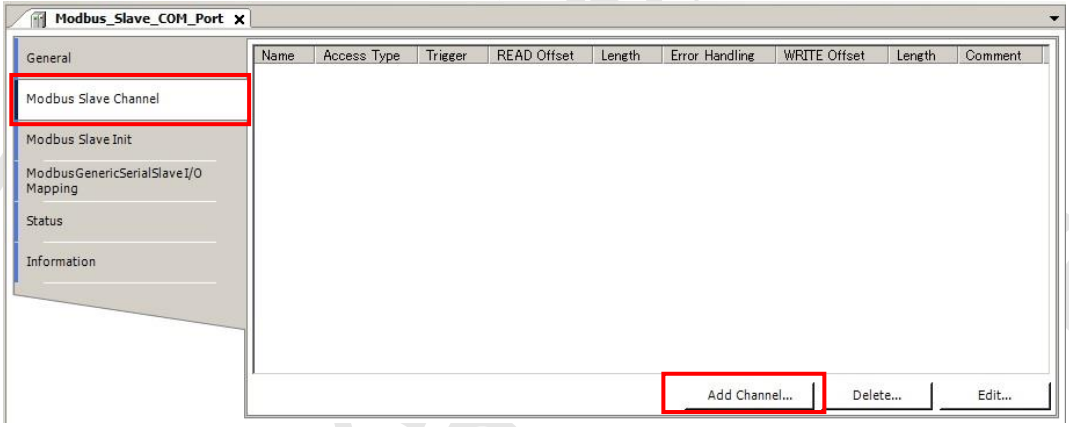
Configure Bus Cycle Task. Task configured here, configure priority “Less than equale 4” at this configuration. In caseofconfiguration more than 5, there is an illegal operation risk.



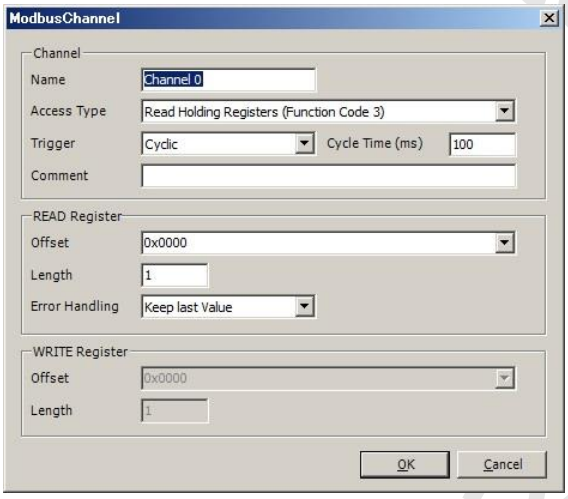
Conguration of issueing function code are done by each slave. Following configuration window appears by double clicking target slave. Configure slave adress and response timeout. Response timeout can be configured independently for each slaves and in case of both side configuration slave and master, cofigured value of slave is valid. In case of deleting default value of Slave response timeout, it is only the case to apply value of reponse timeout of master side.

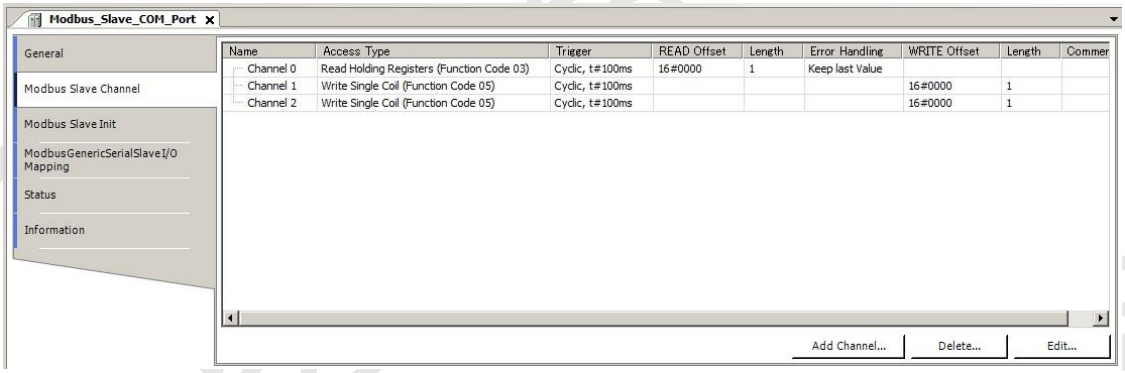


Configure function code sent by “Add Channel…” of “Modbus Slave Channel” tab.

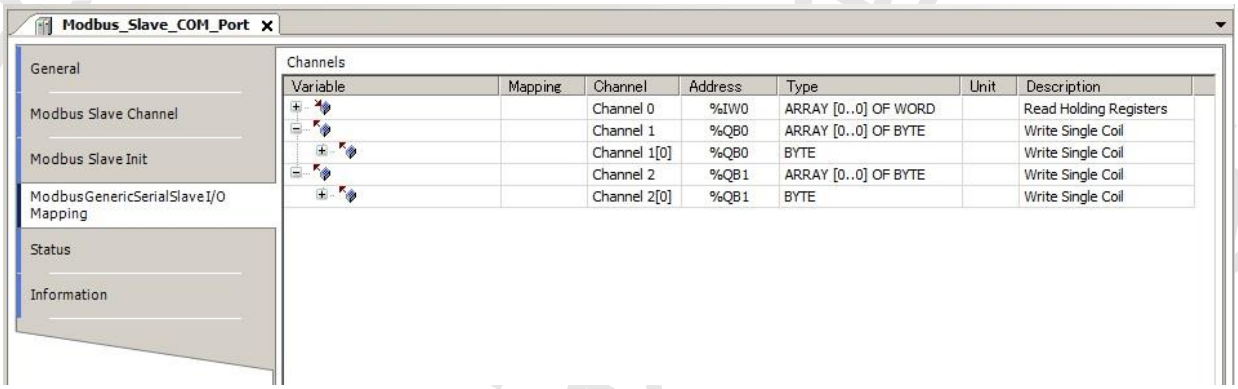


Configure following procedure for each function code. In case of selection of “Rising edge (rising edge)” for “trigger” setting, trigger variable (BOOL) is assigned %QX address automatically.





Read/Write data are assigned address of %IW or %QW shown on “ModbusGenericSerialSlaveI/OMapping” tab. Read data from slave is Input area (%IW) and Write data for slave is Output area (%QW). Value of “Default Value” is output value when the status change RUN → STOP.

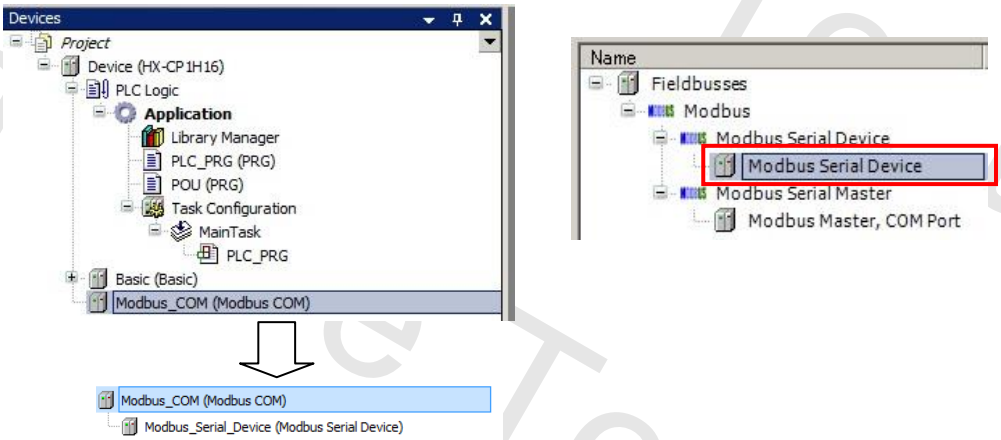


Caution

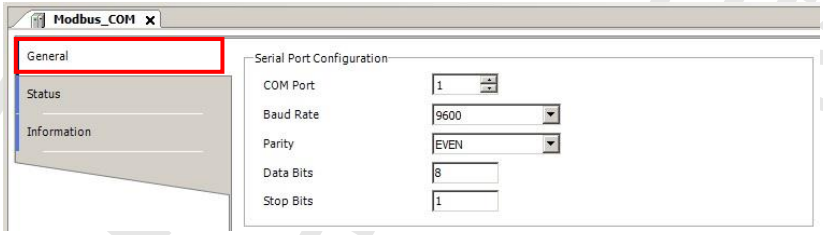
- Modbus Master sends is 1 channel Request dada every 1 cycle. Therefore in case of trigger set is "Rising edge (rising edge)" and if trigger variable changed so high-frequency, previous request data is not completed to send and it will be the timing for the next trigger and sometimes trigger may not be recognized correctly. In case of the time from the start of request data to the end of response data is T1, the rough calculation of inhibited time not to change trigger bit is described T1+T2+...+Tn.

3.2.5 Modbus-RTU Slave

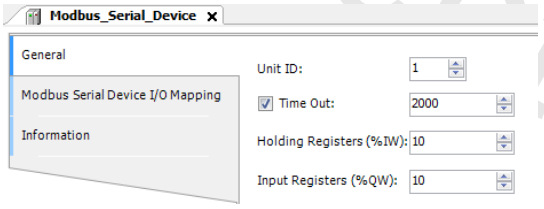
Add "Modbus COM" same as Modbus-TCP master, add "Modbus Serial Device" at "Add Device".

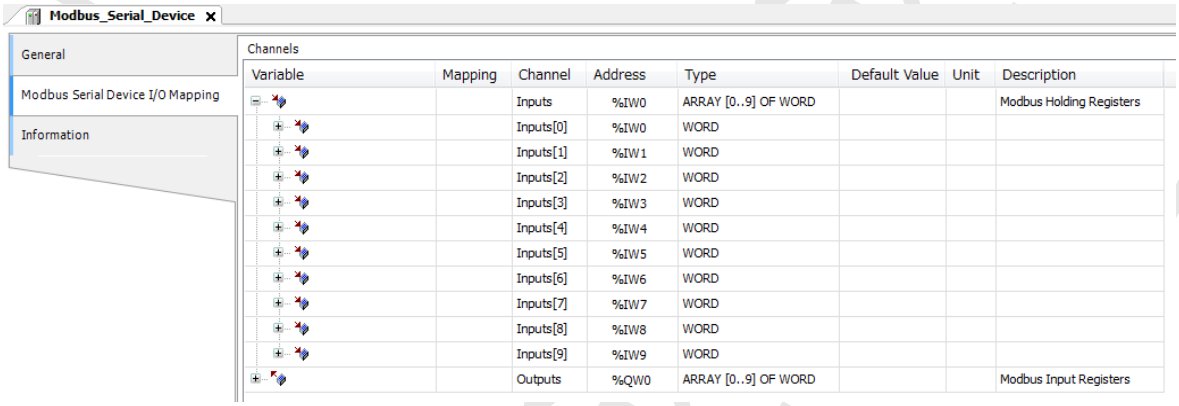


Configure port. Following window appears by double clicking “Modbus_COM (Modbus COM)”. Configure port according to using master. Configure “1” for “COM Port” at HX-CPU side.



Open configuration window by double clicking “Modbus TCP Slave Device” and configure needed parameters. Mapping table is created in “Modbus Serial Slave Device I/O Mapping” tab according to the configured size on Holding Registers (%IW) and Input Registers (%QW).





Caution

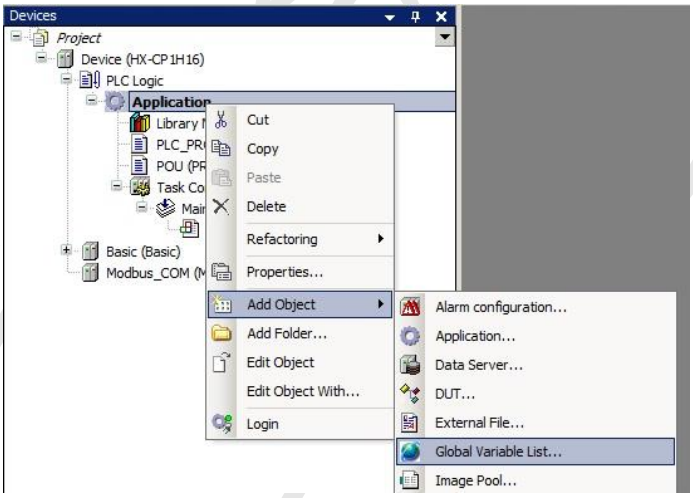
It is reqired Master query to follow Reguration of Modbus specification data format. In case of non-supported function code, out of range of address, data, data value etc, HX-CPU may not response correctlly.

3.3 CPU Link

3.3.1 Overview

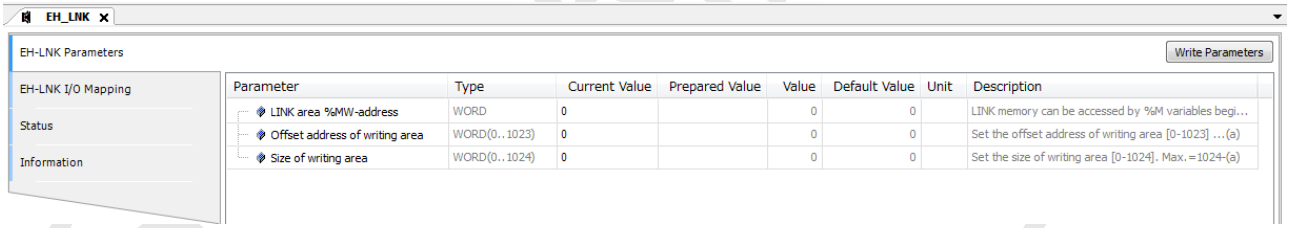
CPU Link System is Network System exchanging data via loop type connection among CPU Link Modules. Each CPU accesses shared memory called Link Area via CPU Link Module. Link Area is defined as Global Variable List (GVL) at each CPU side, %M address is used for accessing Link Area data. %M address is commonly used address among all CPUs.

GVL is added by right clicking “Application” shown in below.



3.3.2 Configuration Link parameter

Open “EH-LNK parameter” display by double clicking EH-LNK on tree and configure “Offset address of writing area (Writing area %MW offset address)” and “Size of writing area (Size of writing area)”.



Caution

Value of configuration range are shown in below. HX-CODESYS may not detect error even if configured value are beyond following range.

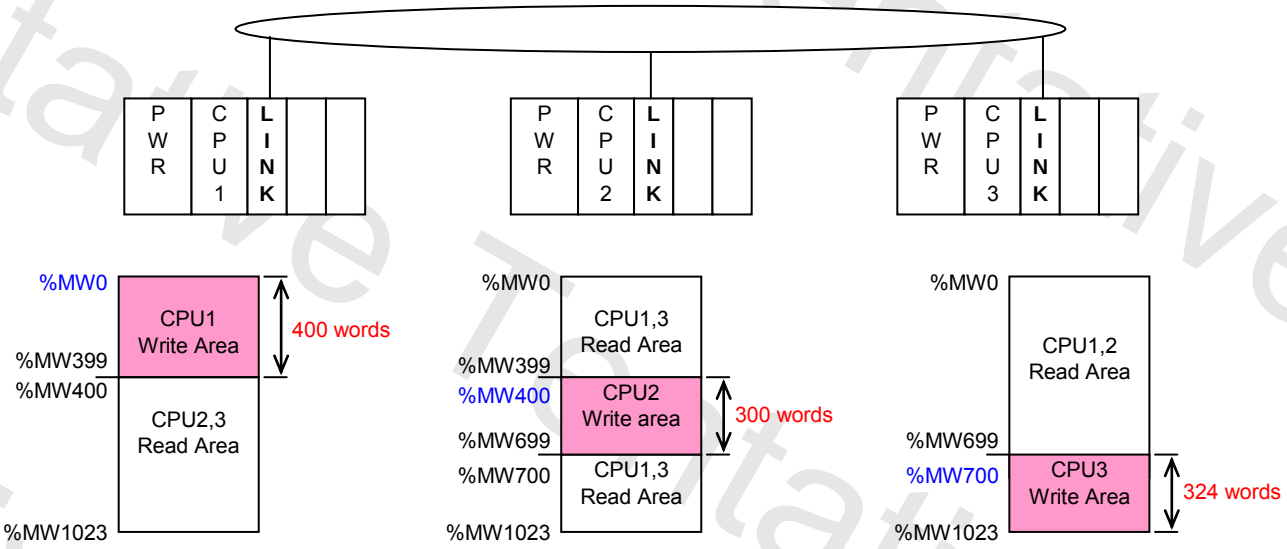
・Offset address of writing area：0～1023

・Size of writhing area　　：0～1024

(However, maximum is subtract “Offset address of writing area” value from 1024)

Configuration example

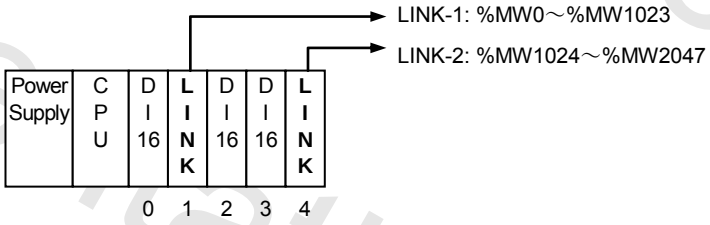
This example shows three CPUs. Each CPU has each writing area and not be duplicated. Certain writing area of CPU is reading area for the other CPUs.



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

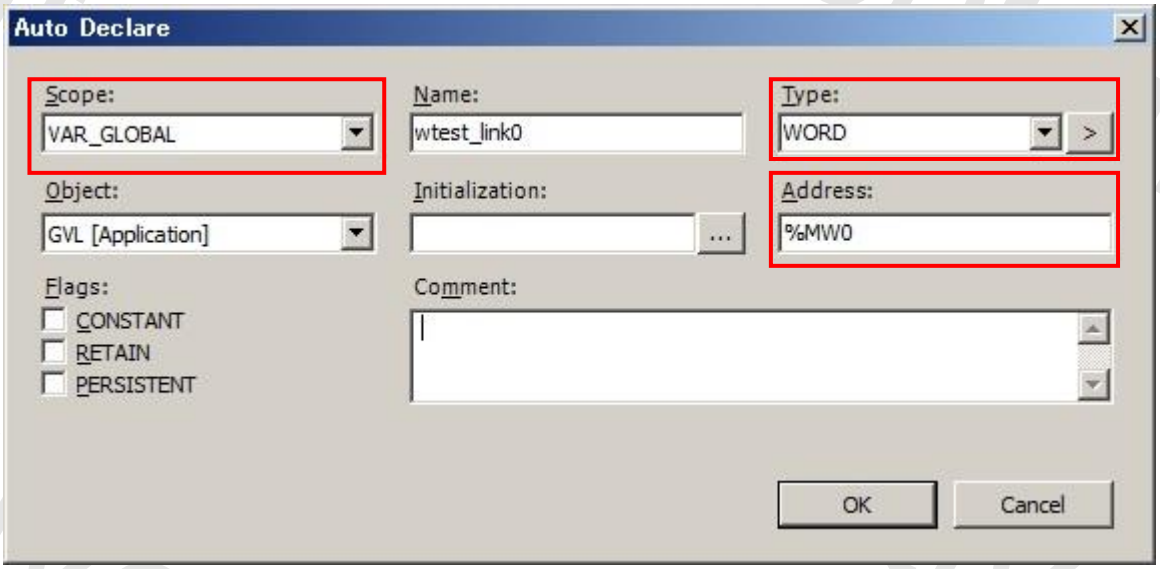
In case of CPU Link module number is one, Link area address is %MW0 - %MW1023. If several number of CPU Link modules are used in below figure, Link area address are shown in below table. This is defined by number of CPU Link modules not by mounted slot number of CPU Link modules.

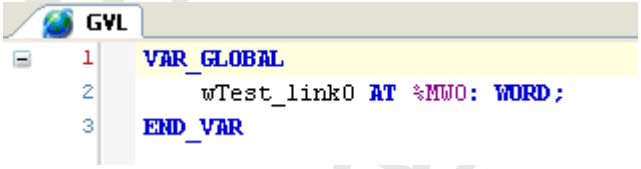
	LINK-1	LINK-2	LINK-3	LINK-4	LINK-5	LINK-6	LINK-7	LINK-8
From	%MW0	%MW1024	%MW2048	%MW3072	%MW4096	%MW5120	%MW6144	%MW7168
To	%MW1023	%MW2047	%MW3071	%MW4095	%MW5119	%MW6143	%MW7167	%MW8191



3.3.3 Declaration of Link Variable

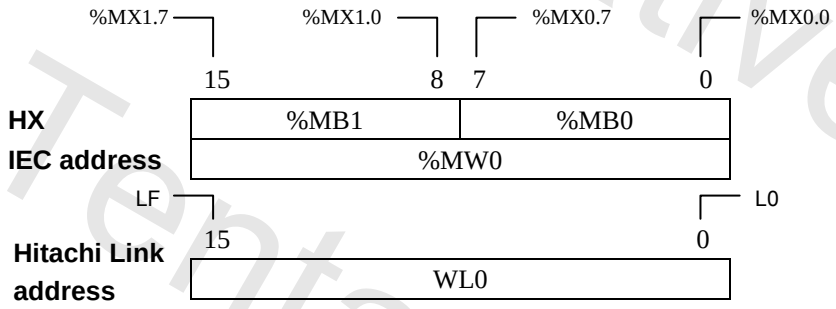
If new variable is started to use during programming, “Automatic Declaration” dialog appears. In case of Link variable, configure %MW address at address field as “VAR_GLOBAL” in scope field. This is assigned GVL automatically by pressing OK button.





Global variable declared here is used in POU. In the above case, it is “GVL.wTest_link0”.

It is also possible to use BOOL Type or BYTE Type %M address.



Caution

Take care for “Word Swap” of DWORD Type data when make communication with EHV+ series via CPU Link module.

EHV+ : %MD0 = 16#1234 5678 ➔ HX : %MD0 = 16# 5678 1234

IEC address is decimal number shown in below, however Hitachi CPU Link address is hexadecimal number.

IEC address			Hitachi CPU Link address	
bit	byte	word	bit	word
%MX0.0	%MB0	%MW0	L0000	WL000
%MX0.1			L0001	
%MX0.2			L0002	
%MX0.3			L0003	
%MX0.4			L0004	
%MX0.5			L0005	
%MX0.6			L0006	
%MX0.7			L0007	
%MX1.0	%MB1		L0008	
%MX1.1			L0009	
%MX1.2			L000A	
%MX1.3			L000B	
%MX1.4			L000C	
%MX1.5			L000D	
%MX1.6			L000E	
%MX1.7			L000F	
%MX2.0 to 2.7	%MB2	%MW1	L0010 to 001F	WL001
%MX3.0 to 3.7	%MB3			
%MX4.0 to 4.7	%MB4	%MW2	L0020 to 002F	WL002
%MX5.0 to 5.7	%MB5			
%MX6.0 to 6.7	%MB6	%MW3	L0030 to 003F	WL003
%MX7.0 to 7.7	%MB7			
...	...	...	...	...
%MX2044.0 to 2044.7	%MB2044	%MW1022	L3FE0 to 3FEF	WL3FE
%MX2045.0 to 2045.7	%MB2045			
%MX2046.0 to 2046.7	%MB2045	%MW1023	L3FF0 to 3FFF	WL3FF
%MX2047.0 to 2047.7	%MB2047			

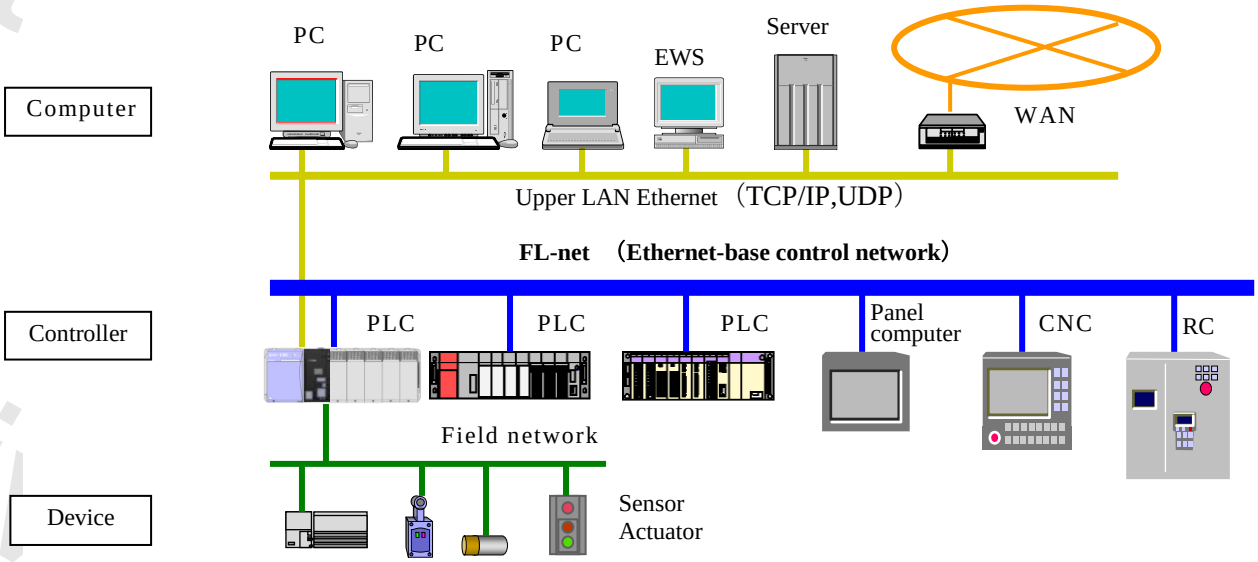
Caution

In case of no usage of CPU Link module, %M address can be used as general internal register.

3.4 FL-net interface

3.4.1 Overview

FL-net is open FA network supporting Data sharing among CPUs of multi vendors based on Ethernet. Cyclic data sharing executed by each CPU using virtual shared memory called common memory.



Common memory 1 and Common memory 2 are available for cyclic data and HX-CPU uses %M address same as CPU Link for this common memory.

Common memory 1 uses same address range with other CPU Link module.

LINK No.	LINK-1	LINK-2	LINK-3	LINK-4	LINK-5	LINK-6	LINK-7	LINK-8
Start of common1	%MW0	%MW1024	%MW2048	%MW3072	%MW4096	%MW5120	%MW6144	%MW7168
End of common1	%MW511	%MW1535	%MW2559	%MW3583	%MW4607	%MW5631	%MW6655	%MW7679
Un-used range	%MW512- %MW1023	%MW1536- %MW2047	%MW2560 %MW3071	%MW3584 %MW4095	%MW4608 %MW5119	%MW5632 %MW6143	%MW6656 %MW7167	%MW7680 %MW8191

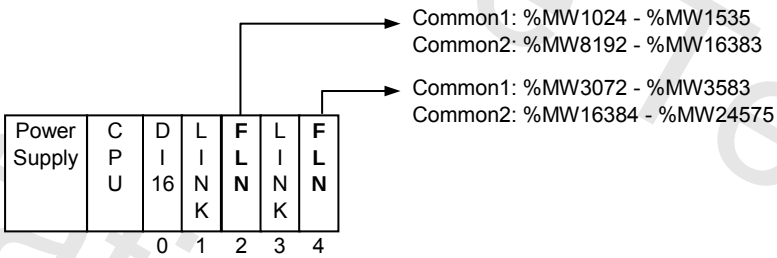
Usable range at each LINK No. is 512 words from the top due to maximum size for common memory 1 is 512 words.

When FL-net is used, it is not possible to use address of non-used range in above.

Common memory 2 uses %M address assigned for FL-net usage.

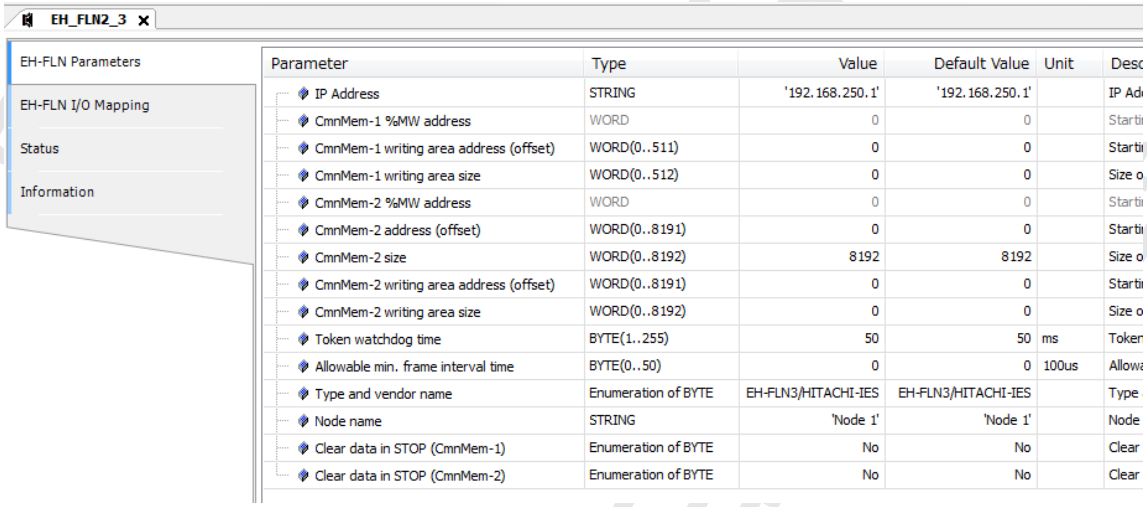
FL No.	FL-1	FL-2
Start of common2	%MW8192	%MW16384
End of common2	%MW16383	%MW24575

Maximum size of common memory2 is 8192 words.



3.4.2 FL-net parameter configuration

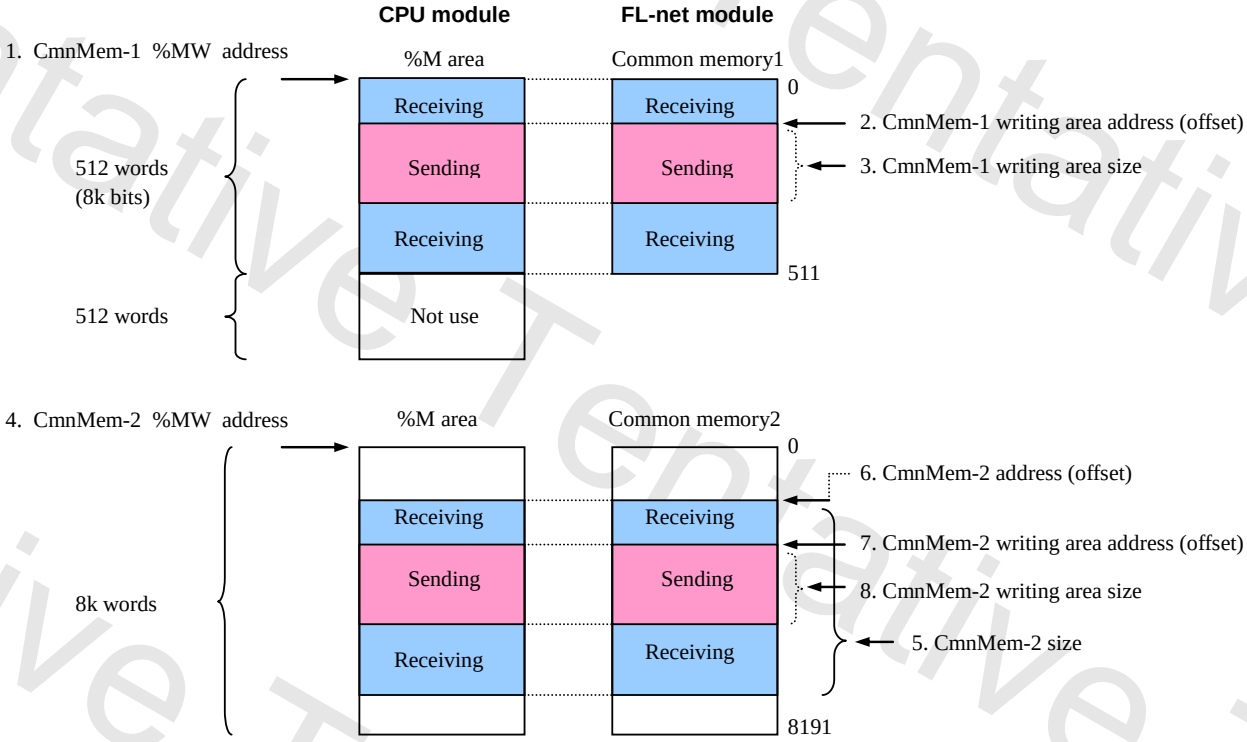
Configure each parameter after opening “EH-FLN parameters” display by double clicking EH-FLN2_3 on tree.



No.	Name	Explanation	Default
1	IP address	Set IP address of EH-FLN2/3	192.168.250.1
2	CmnMem-1 %MW address	Display starting address at On-line.	0
3	CmnMem-1 writing area address (offset)	Configure starting address (offset) of sending area of own node at common memory1.	0
4	CmnMem-1 writing area size	Configure size of sending area of own node by word unit at common memory1.	0
5	CmnMem-2 %MW address	Display starting address at On-line.	0
6	CmnMem-2 address (offset)	Configure starting address (offset) of data transfer between CPU module and EH-FLN2/3 at common memory2.	0
7	CmnMem-2 size	Configure size of data transfer between CPU module and EH-FLN2/3 by word unit. Configure added value of own node sending size and other node sending size.	0
8	CmnMem-2 writing area address (offset)	Configure starting address (offset) of sending area of own node by word unit at commonmemory2.	0
9	CmnMem-2 writing area size	Configure size of sending area of own node by word unit at common memory2.	0
10	Token watchdog time	Configure monitoring time between token receiving at own node and token passing for next node.	50
11	Allowable min. frame interval time	Time between token receiving at own node and some sending frame from own node.	0
12	Type and vendor name	Select vendor name and model name. There is no difference among these.	EH-FLN3/ HITACHIIES
13	Node Name	Configure some name with byte character or number within 10 characters.	Node1
14	Clear data in STOP (CmnMem-1)	Configure clear sending area of common memory1 or not when CPU module stops.	No
15	Clear data in STOP (CmnMem-2)	Configure clear sending area of common memory2 or not when CPU module stops.	No

Configuration Image

Each configuration parameter image shown in below figure.



3.4.3 Cyclic transfer

Start cyclic transfer and node join network automatically after correct parameter setting. Refer status monitoring library of Chapter 3.4.5 for status of completing initialize.

3.4.4 Message transfer

HX-CPU does not support function of user message transfer with FL-net module.
Response only for limited message only received inquiry message from other node.
Detect timeout at request side when requiring message can't be response from other node.

No.	Message	Require	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 FL-net module create response message.

3.4.5 Status monitor library

Function block is available to get each status information of EH-FLN2/3.
Store status information of EH-FLN2/3 into dedicated structure FLInfo at the timing of xDone turns TRUE(ON) when xEnable set ON with FL No. of FL-net.

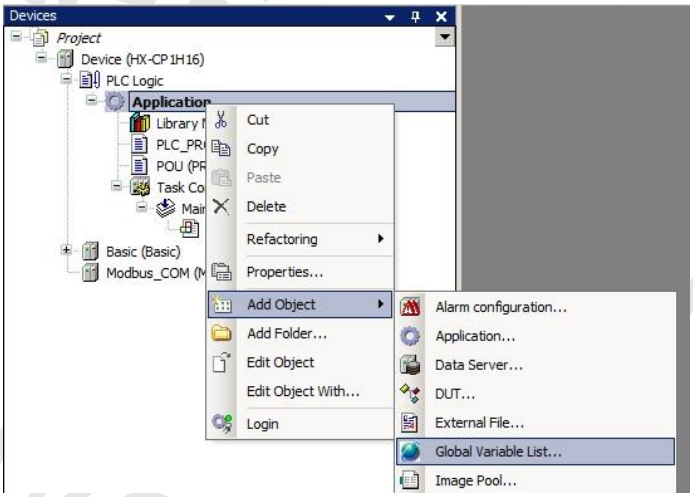
Detail of FLInfo structure

Member name	Contents		Type	Remark
xInitDone	Node initialization	1: Done 0: Not yet	BOOL	
xParamError	Parameter error	1: Error 0: Normal	BOOL	
xAdrDuplicated1	Address duplication area1	1: Error 0: Normal	BOOL	
xAdrDuplicated2	Address duplication area2	1: Error 0: Normal	BOOL	
xWaitforRcv	Wait for receiving status 1: Error(Wait for receiving) 0: Normal		BOOL	
xTokenWatchdogTime	Timeout of token	1: Error 0: Normal	BOOL	
xNodeDuplicated	Node number duplication	1: Error 0: Normal	BOOL	
xTokenTimeoutOwnNode	Timeout of own node token	1: Error 0: Normal	BOOL	
xTBN_CBN_BSIZE	TBN, CBN or BSIZE error	1: Error 0: Normal	BOOL	
xCableDisconnect	Disconnect Cable	1: Error 0: Normal	BOOL	
xTokenModeUnmatch	Un-match token mode	1: Error 0: Normal	BOOL	
axLinkFlag	Link Node	1: Joined 0: Not joined	ARRAY[1..254] OF BOOL	
axRunStatus	Status flag upper layer (RUN Status) 1: RUN 0: STOP		ARRAY[1..254] OF BOOL	
abErrStatus	Status flag upper layer (STOP 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	1: Linked 0: Not linked	BOOL	Valid FLN3 only
xLinkSpeed	Link speed flag	1: 100Mbps 0: 10Mbps	BOOL	Valid FLN3 only
xDuplexMode	Duplex mode flag 1: Full duplex 0: Half duplex		BOOL	Valid FLN3 only
xSDRAMError	SDRAM error	1: Error 0: Normal	BOOL	
xEEPROMError	EEPROM error	1: Error 0: Normal	BOOL	
xSystemError	System error	1: Error 0: Normal	BOOL	
xFlashError	FLASH error	1: Error 0: Normal	BOOL	
xMPUError	MPU error	1: Error 0: Normal	BOOL	
xSystemRAMError	System RAM error	1: Error 0: Normal	BOOL	
xNodeNumberError	Node number error	1: Error 0: Normal	BOOL	
xLinkAddressError	Link address error	1: Error 0: Normal	BOOL	

3.5. Profibus Master

3.5.1 Overview

It is possible to use EH-RMP or EH-RMP2 as Profibus master module. Variable used for Profibus master module is declared in Global Variable List (GVL) and can be accessed %M address. Add into GVL by right clicking "Application" shown in below.



3.5.2 Configuration of Link Parameter

Open "EH-LNK parameter" display by double clicking EH-LNK on tree and configure "Offset address of writing area (Writing area %MW offset address)" and "Size of writing area (Size of writing area)".

Offset address of writing area

Configure start address of Link area. Set 0(zero) for this configuration.

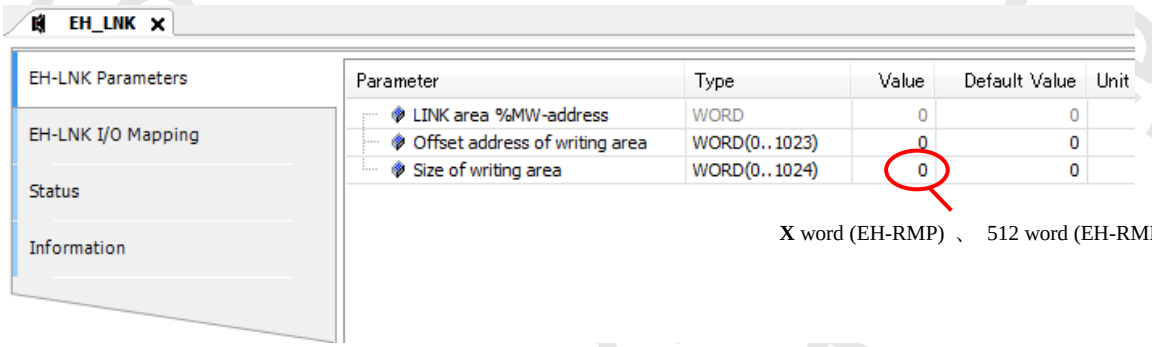
Size of writing area

EH-RMP: Configure Output size configured by Sycon

EH-RMP2: Configure 512 (as fixed value)

Caution

Do not configure 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(zero). If 0(zero) is configured, in case of EH-RMP, STATUS LED brinks 4 times. In case of EH-RMP2, configure 512. If value except 512 is configured, STATUS LED brinks 4 times.



X word (EH-RMP) 、 512 word (EH-RMP2)

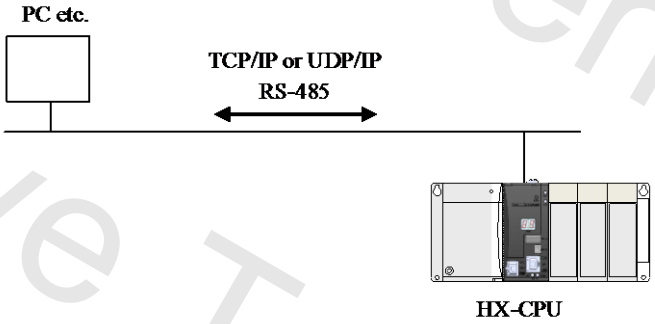
Address of reading area **(**Configuration is not needed.)

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**]				
...			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

It is possible to do general purpose communication by using Ethernet port or serial port of HX-CPU.



3.6.1 General purpose communication over Ethernet

In order to do general purpose communication over Ethernet (TCP/IP or UDP/IP), prepared Function Block library under NetBaseService library shown in below table is needed. Refer CAA_NetBaseService.pdf in NetBaseServices library for further information.

Table of NetBaseServices library

Protocol	Command	Function
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

Regarding to the byte order of networking, it is general to follow Big endian sending from most higher byte. However, it is not big-endian style due to using variable type in program. Therefore, in some case byte data conversion is needed by using dedicated function prepared in SysSocket library.

・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

In order to do general purpose communication over Serial, prepared Function Block library under SysCom library shown in below table is needed. Refer Application manual (Command reference edition) 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 of SysCom library are not supported by HX-CPU.

・SysComGetSettings
・SysComSetTimeout

Chapter 4 Other functions

4.1 OPC-UA Server

HX-CPU supports OPC-UA server functionality. OPC-UA (Unified Architecture) is the newest spcication of OPC based on the technology used for Web service and this is data exchanging opened standard between 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.

Figure OPC UA Server functionality

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

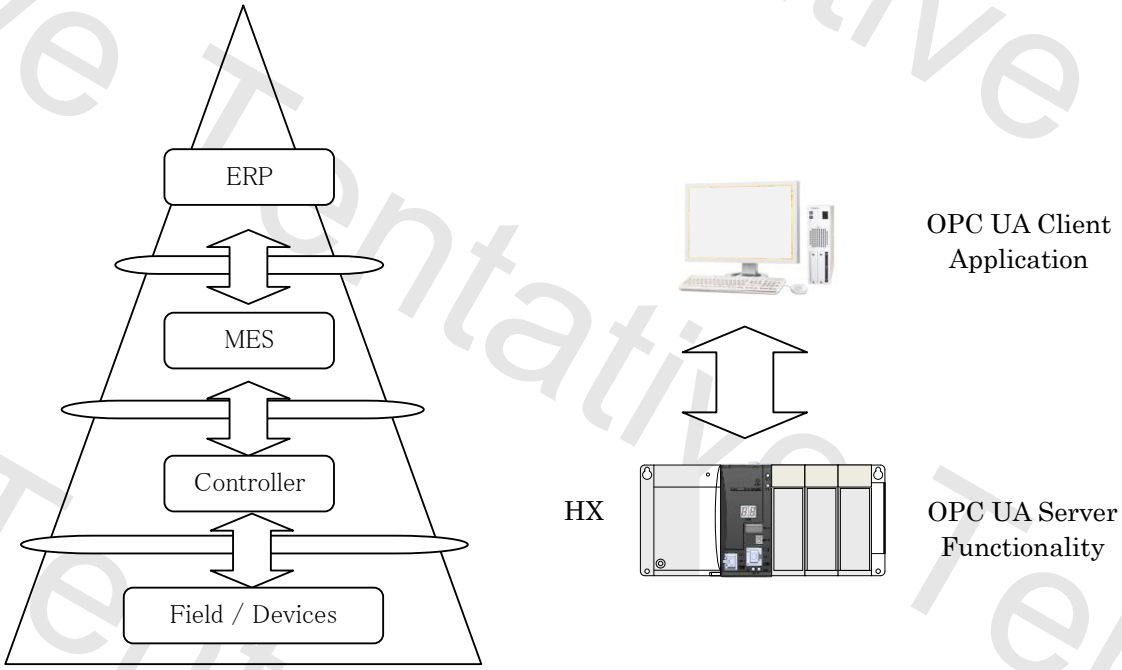


Figure Information Models and OPC UA Server Functionality

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

(1) Configuration of HX-CPU side

Symbol configuration editor

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

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

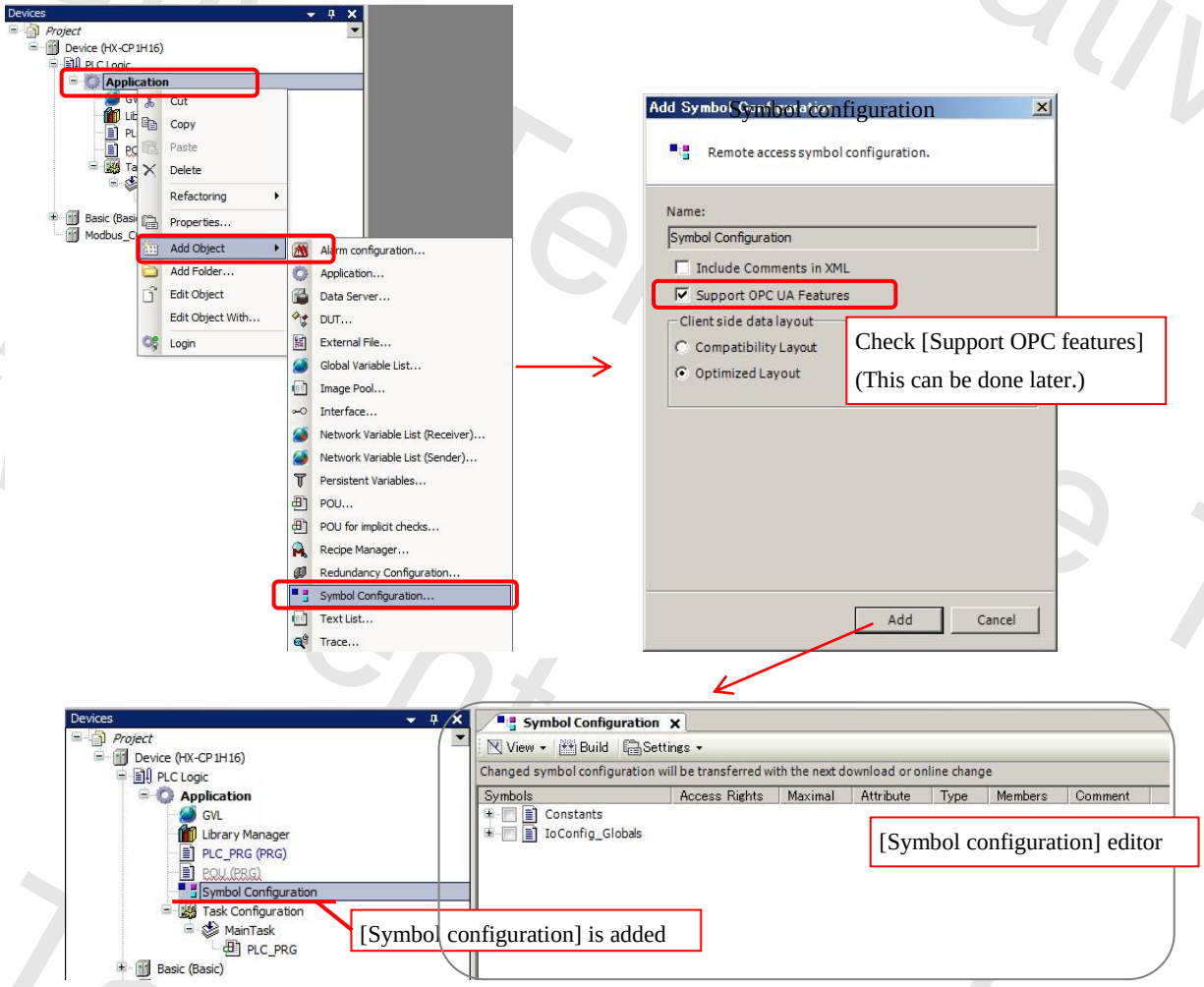


Figure [Symbol configuration] editor

Configuration of Remote accessing variable

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

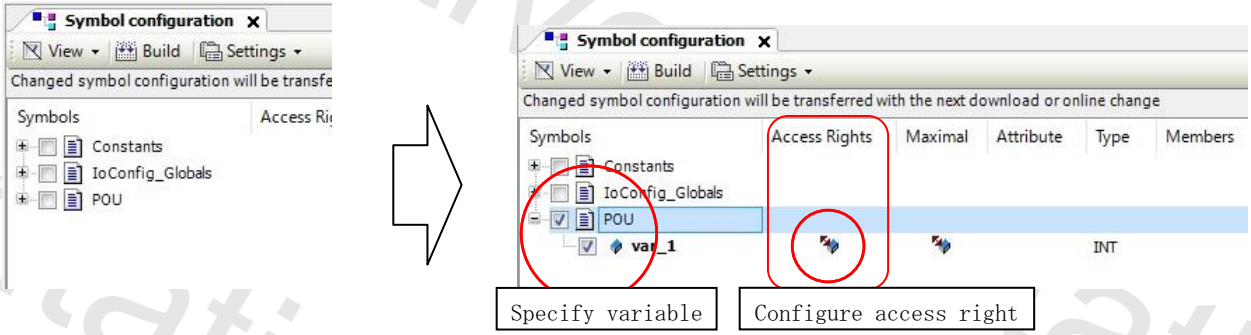


Figure Specifying Remote access variable

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 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].)

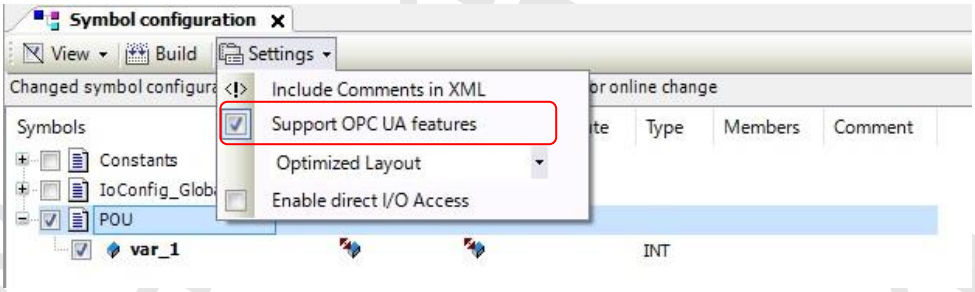


Figure Enabling OPC UA server function

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 Connecting OPC UA server

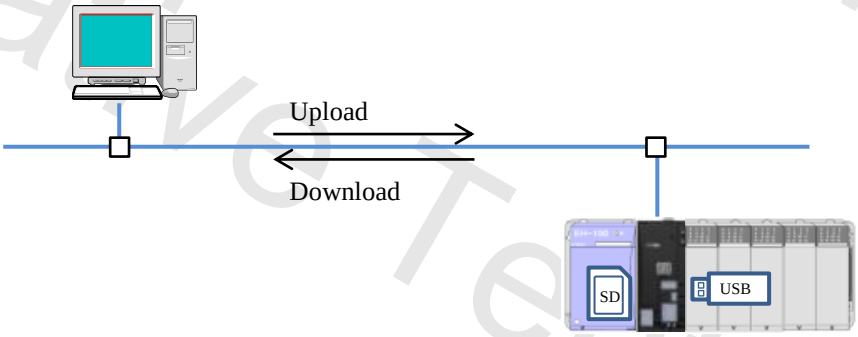
Make sure what is object of security protecting and take countermeasure for system configuration and operation mentioned security protection as an example by user responsibility.

- 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

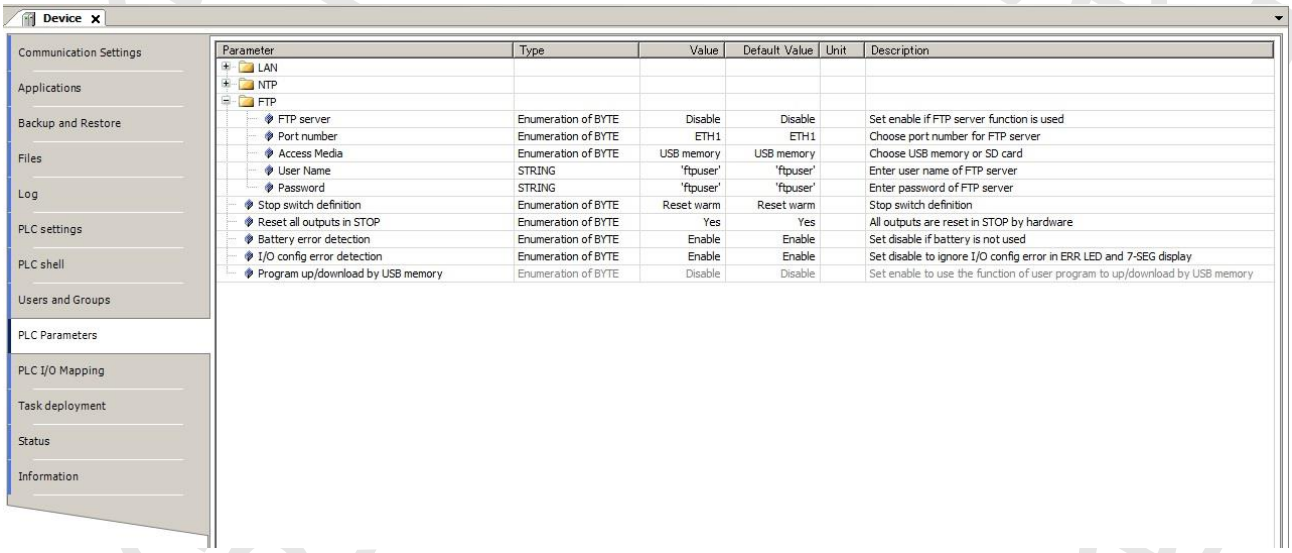
4.2.1 FTP server

File read or write access (uploading / 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 recommended for FTP client.



4.2.2 FTP server configuration

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



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.

4.2.3 List of FTP commands

Usable command list is shown in below.

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 serve.
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> asci
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: Chang remote 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 (([file]) ([local file]))

Function: Change local working 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

Function: Specify current file transfer mode.

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.

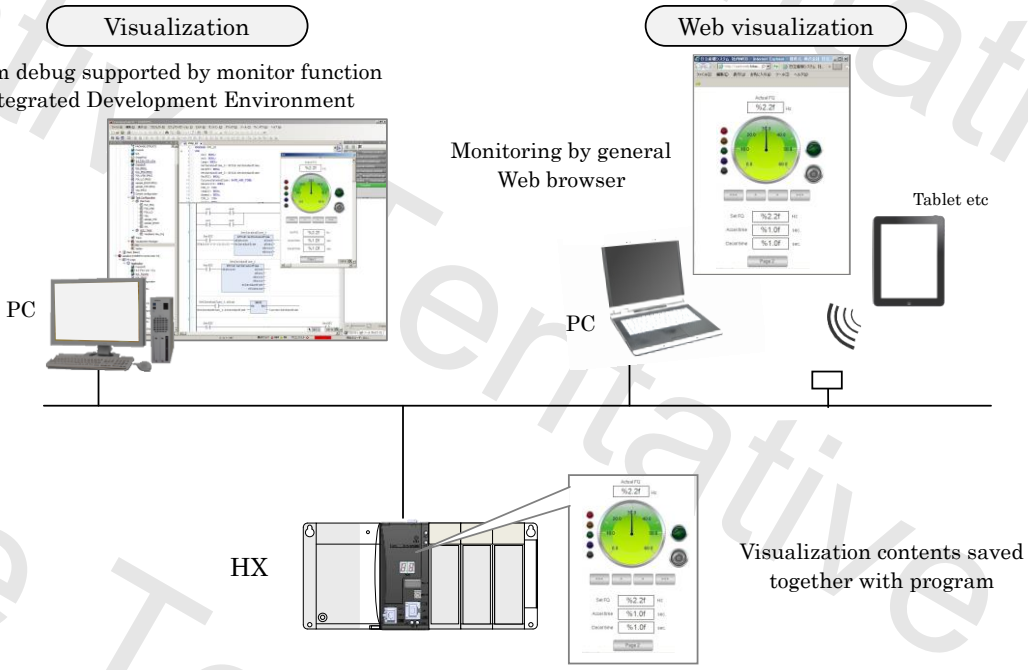


Figure Visualization functionalities

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.

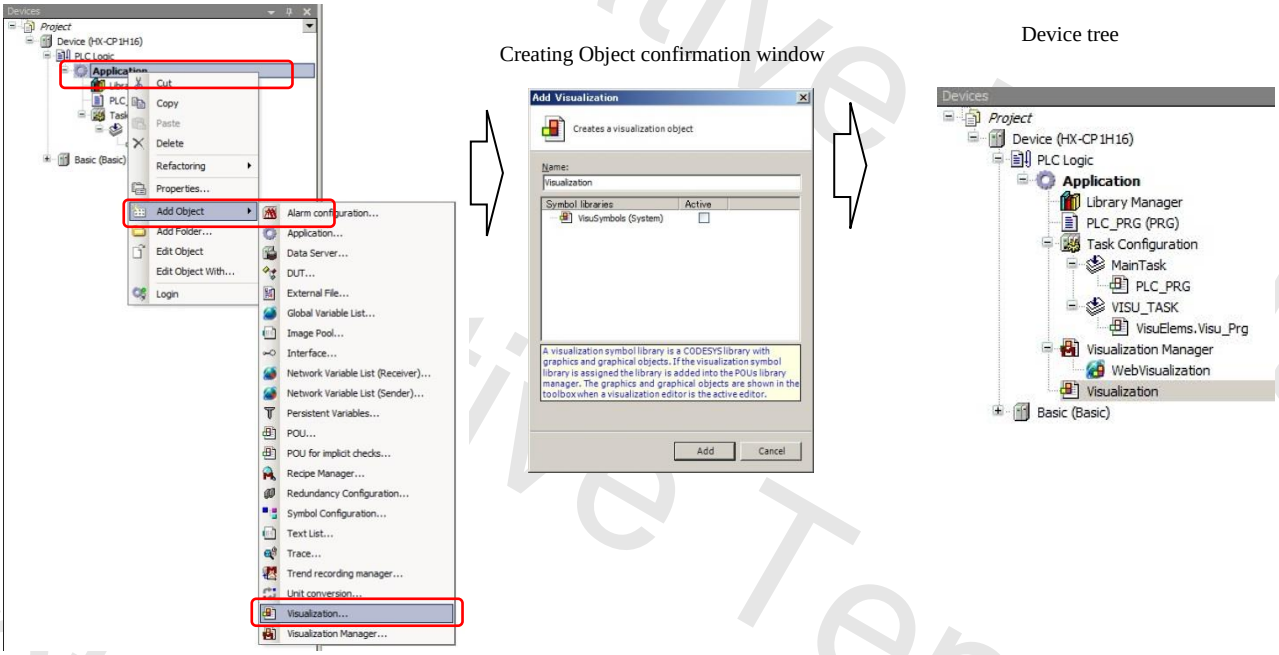


Figure Adding Visualization object

Visualization editor

Double click ”Visualization” on Device tree. Visualization editor appears.

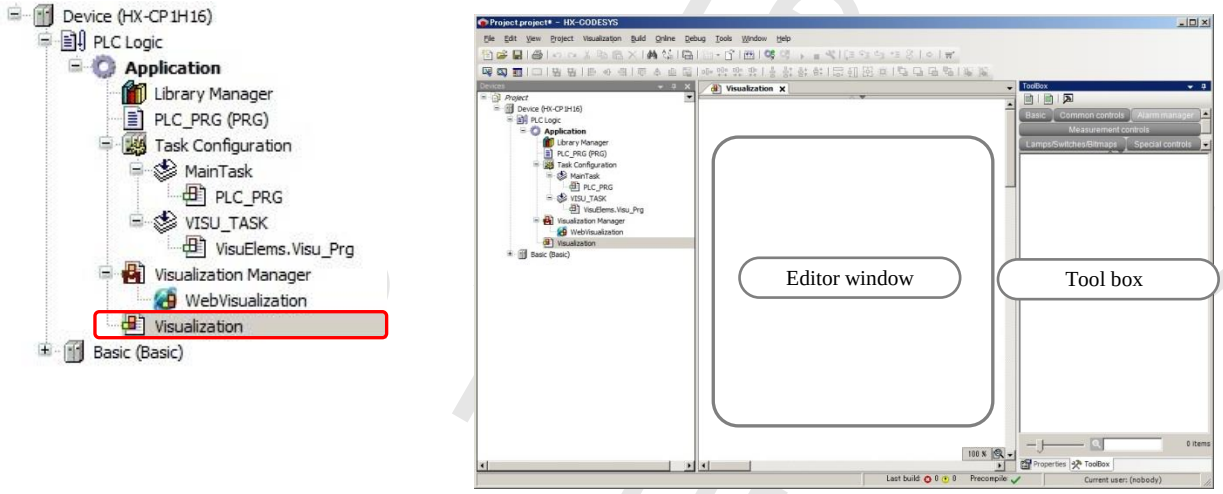


Figure Visualization editor

Usage of elements

There are several parts (this after “elements”) available to create graphical display in Tool box. Select element to use from Tool box and drag it into Editor window. The element is placed on Editor window and Property window appears.

Figure 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.

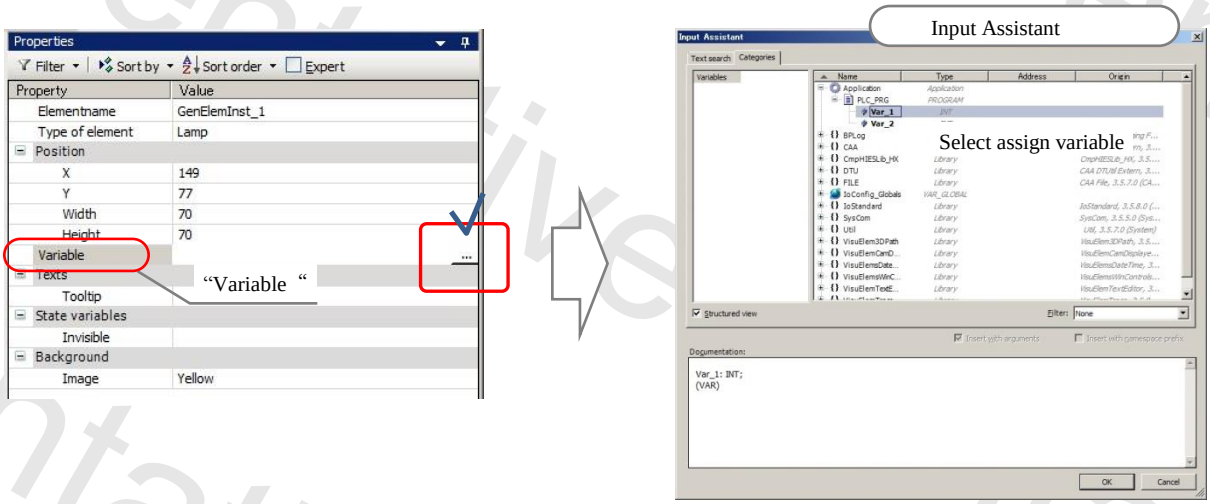


Figure Assignment variable for element

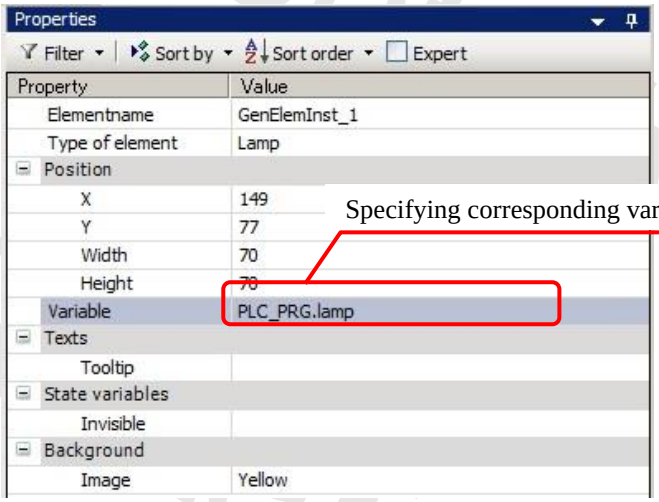
Well use elements

Display BOOL type variable (Read)

Display example

In case of [FALSE]

In case of [TRUE]

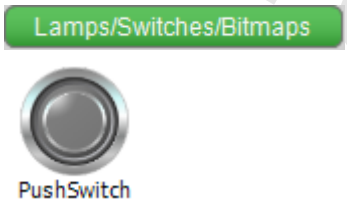


Specifying corresponding variable

・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.)

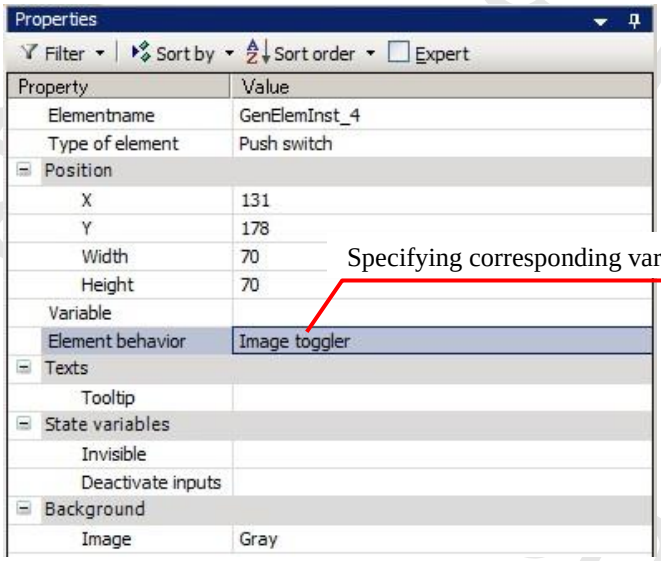
Switching BOOL type variable (Write)



Display example

In case of [FALSE]

In case of [TRUE]



Specifying corresponding variable

・It is possible to specify element behavior at “Element behavior” of Property.
Image toggle: Toggle switch behavior. It turns ON by click once and Turns OFF by one more click.
Image tapper: Push switch behavior. It turns ON during click only.

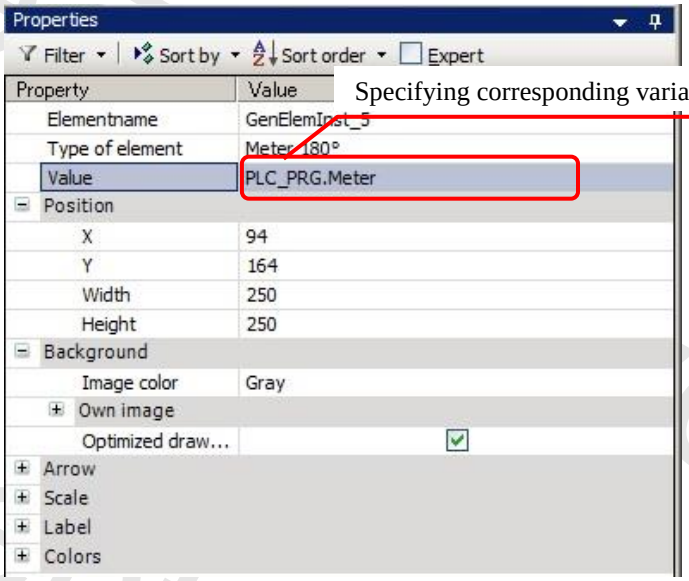
・It is also possible to untenable operation by specified variable value by setting ”State variables”-“Deactivate inputs” of Property.

Refer on-line help for further detail of each element specifications.

Graphical display of Variable value

Meter180

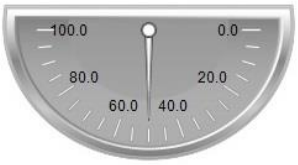
Display example



Specifying corresponding variable

Changing example by Property specifying

Direction



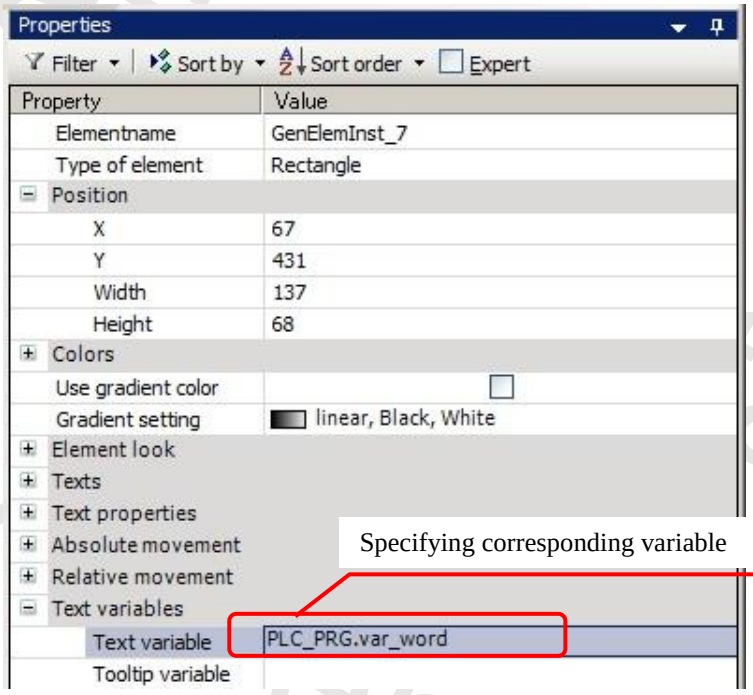
Color, meter indicator, scale of value, display position

Display variable value by text (string)

Rectangle

%d

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



Specifying corresponding variable

*Changing number of digits is possible

%.1f

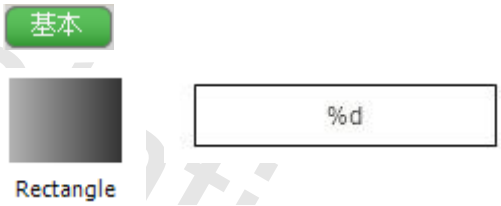
0.3

%.3f

2.552

Refer on-line help for further detail of each element specifications.

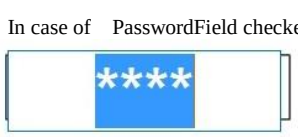
Input variable value by text (string)



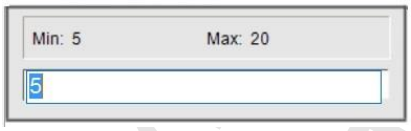
Display example

Input types : Text input

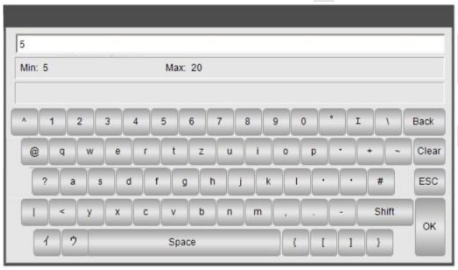
In case of PasswordField checked



Input types : Text input with limits

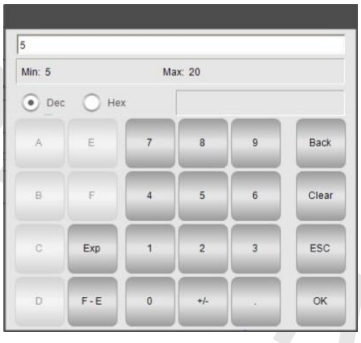


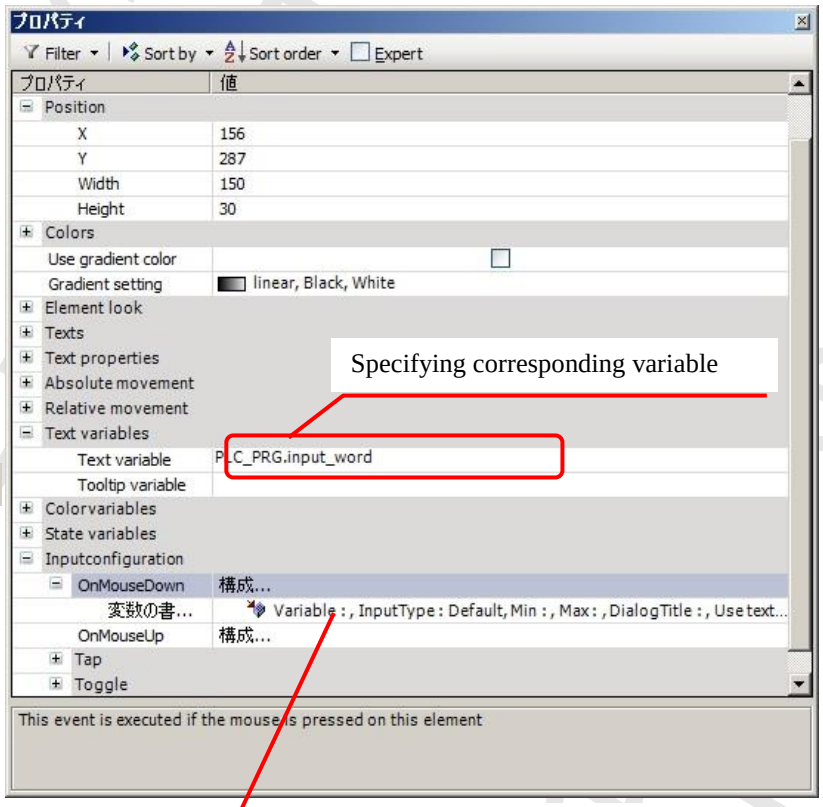
Input types : VisuDialogs Keypad

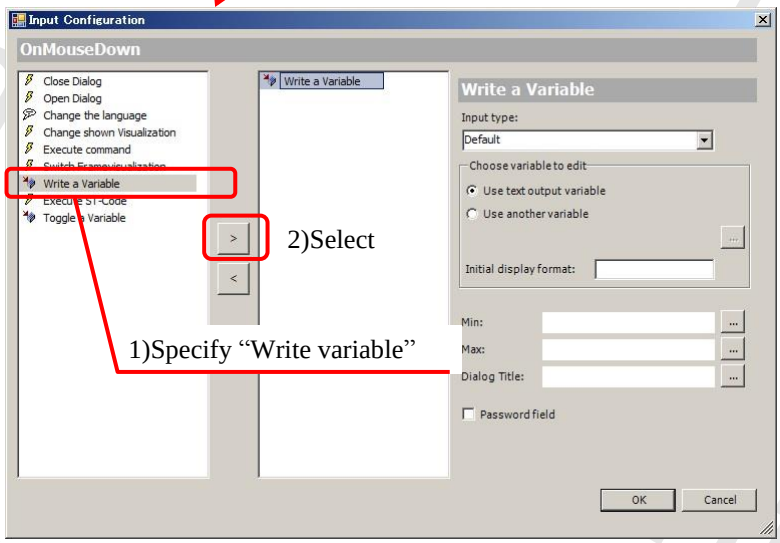


Input types : VisuDialogs Numpad

Input types : VisuDialogs NumpadExtended







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 on-line help for further detail of each element specifications.

Task configuration of Visualization

"VISU_TASK" is added automatically by adding Visualization object.

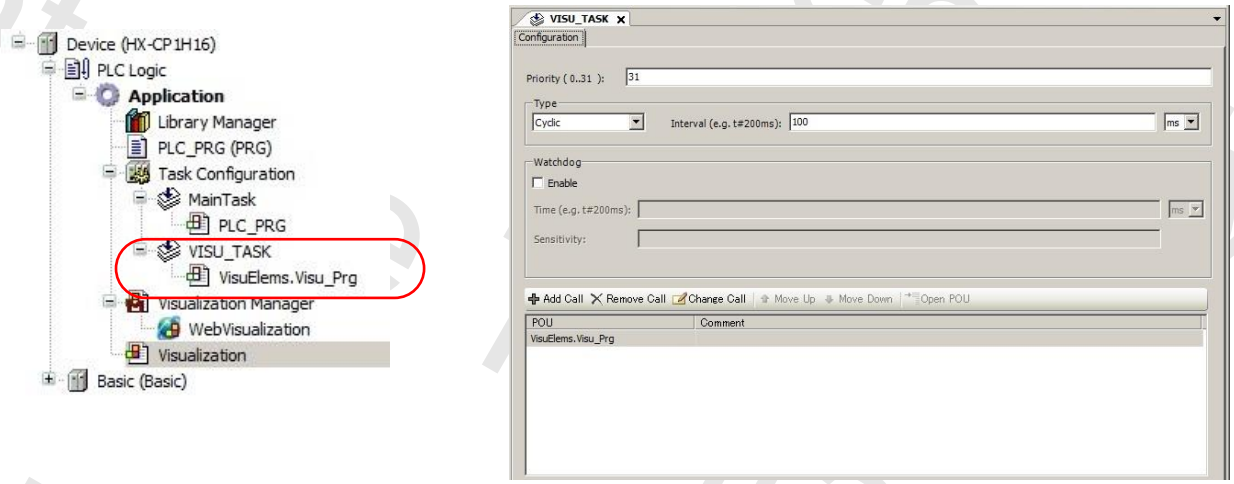


Figure Task configuration of Visualization

Configuration items of Visualization Task

Priority(0,31)	Configure task priority of visualization task. 0(zero) is the highest priority. To avoid impact control program operation, the lowest priority is recommended for this configuration.
Type	Cyclic: Refresh data according to the specified constant cycle. (Default is 100 msec) Event: Refresh by specified condition. It is possible to use specified variable by Input Assistant. Freewheeling: Refresh data by using idle (free) time of each task. Status: Refresh by specified condition. It is possible to use specified variable by Input Assistant.

Visualization manager

This is common configuration items for Visualization and Web Visualization.

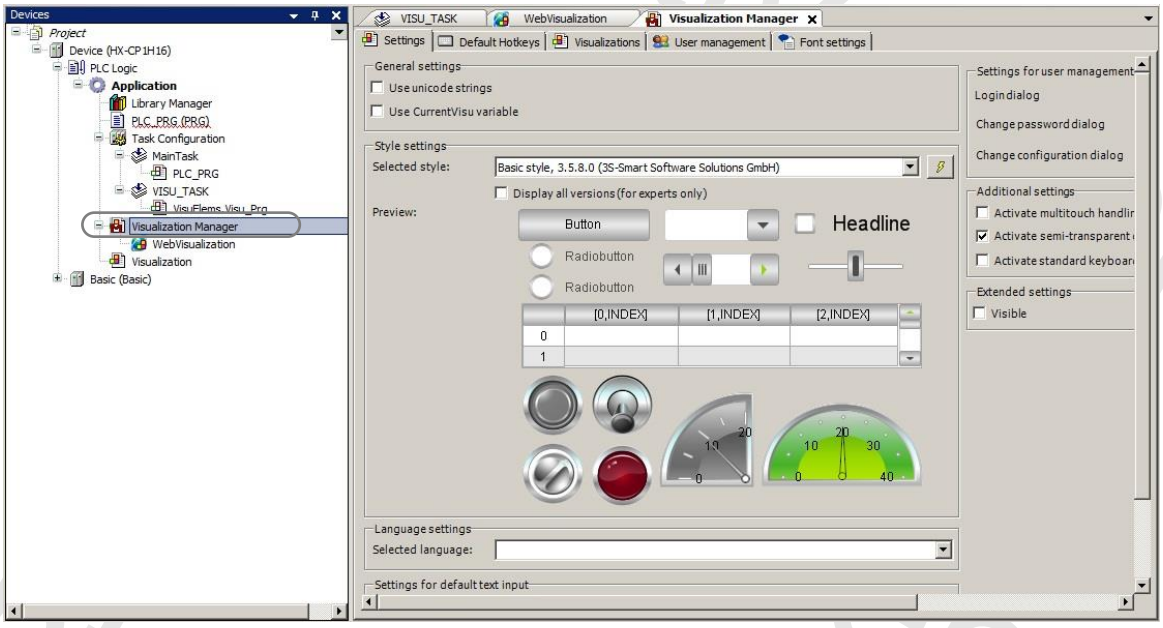
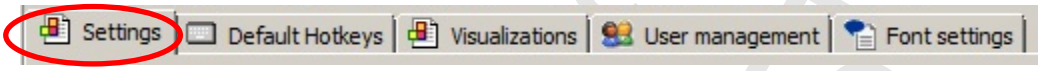
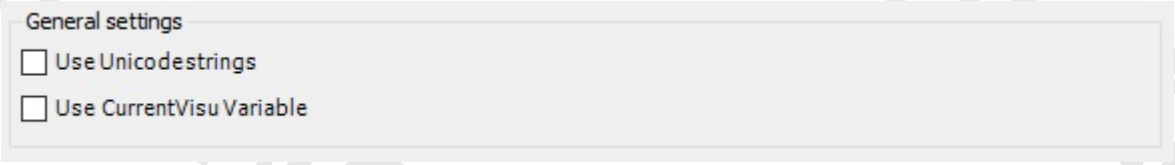


Figure Configuration of Visualization manager

Configuration

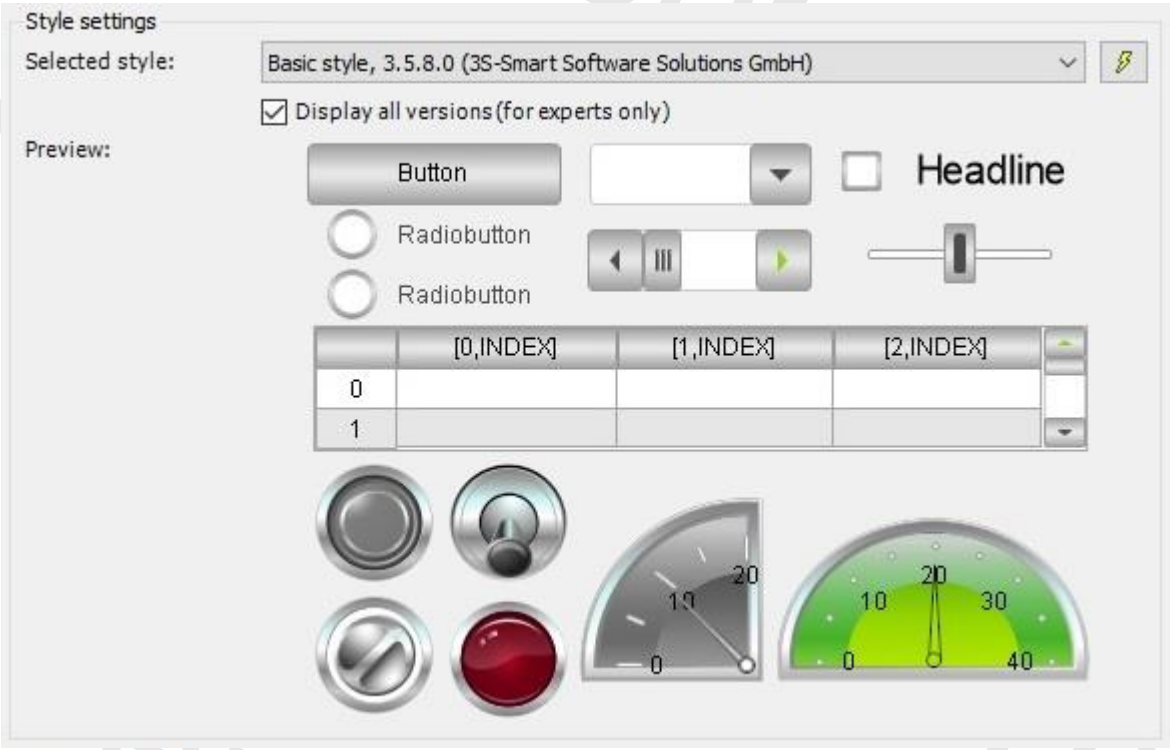


General setting

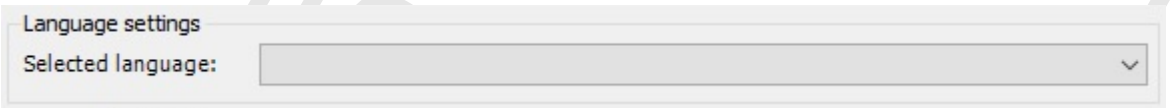


Use Unicodestrings	All string is executed by Unicode format used in Visualization by this option. If Japanese character is displayed for Web visualization, please check this.
User CurrentVisu Variable	Pass name of current display visualization to Global string variable “CurrentVisu”.

Style settings：(Use this default value without change)

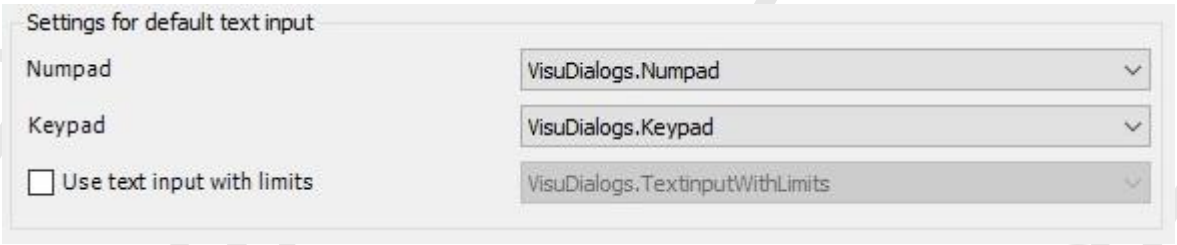


Language settings



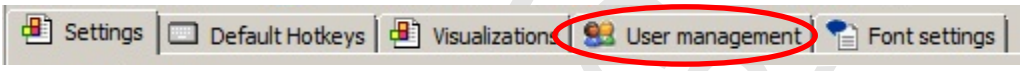
Configured language will be used at the time of start.

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”.

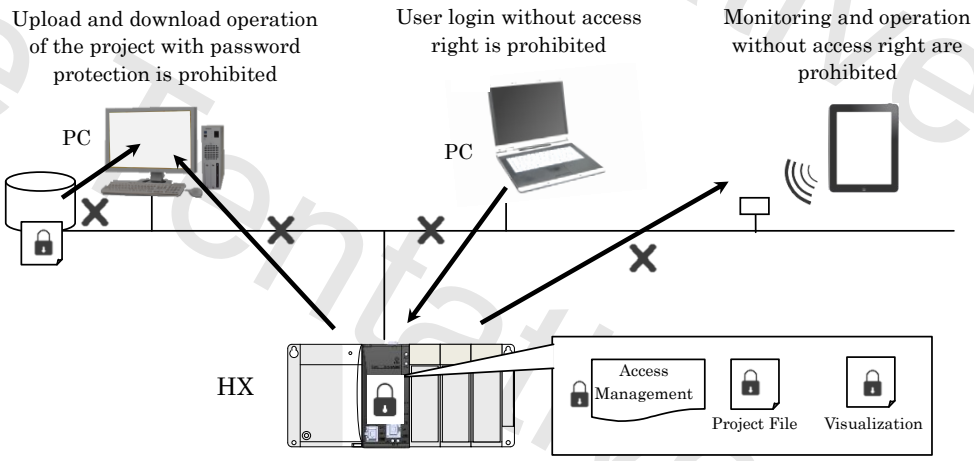
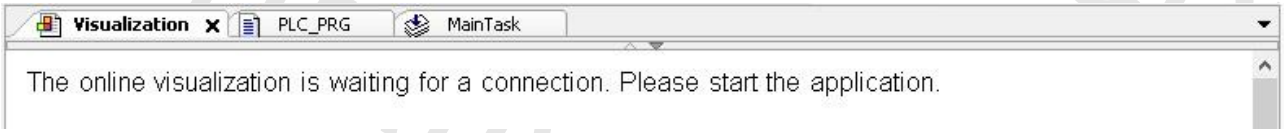


Figure Support function of security protection

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

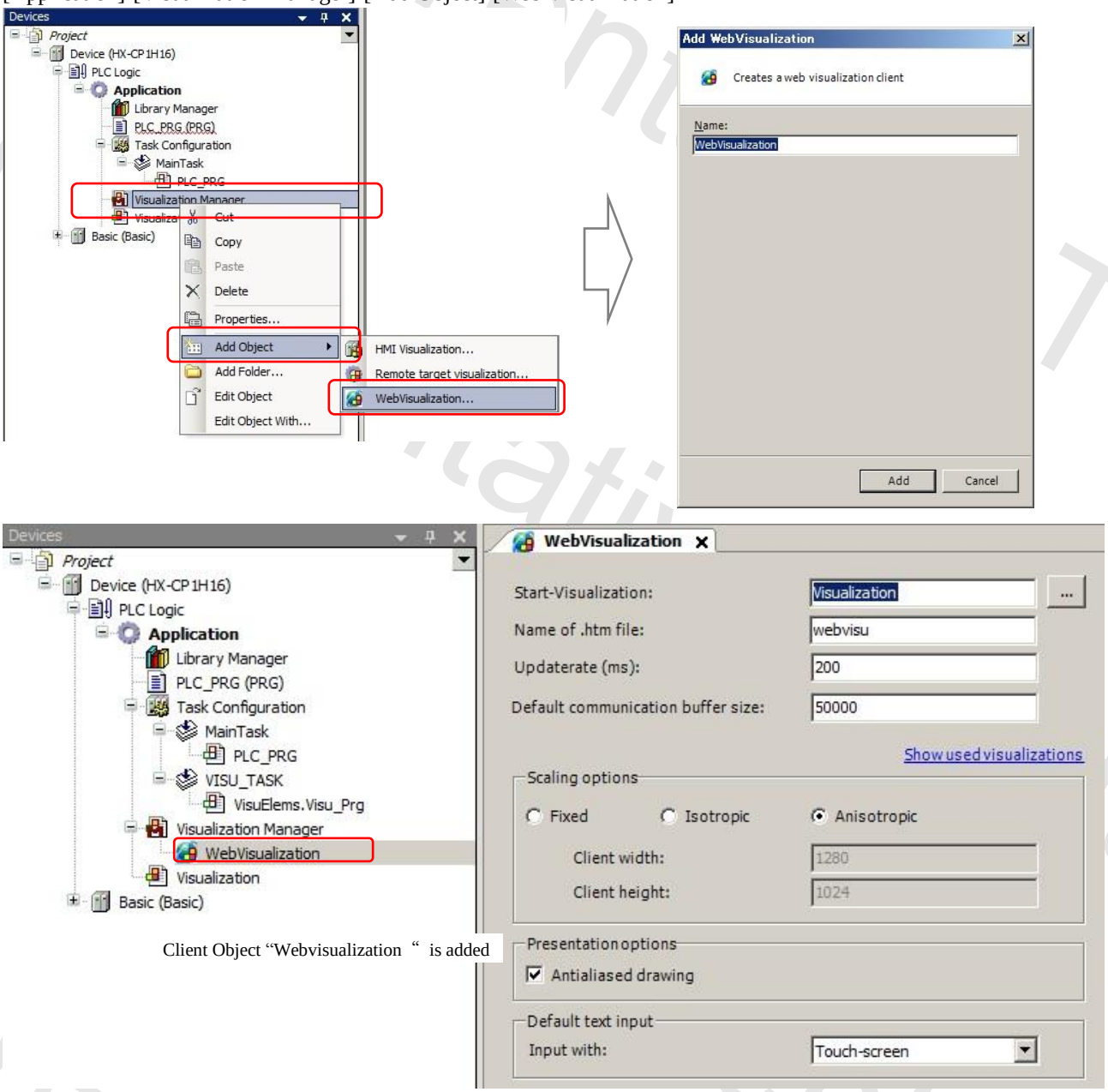
It is possible to access Visualization stored in HX-CPU from general Web browser of PC or tablet. This function is called Web Visualization.
Use Web browser following HTML5 . (Example: Firefox, Chrome, IE9 or later, etc)
Web Visualization can be used during HX-CPU is in RUN status.

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 Web Visualization Object

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

Configuration items of Webvisualization object

Start・Visualization	Input Visualization name when starts Visualization at the time of start automatically.
File name of .htm	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.
Refresh cycle(ms)	Specify refresh cycle in ms units for Web browser. (Default is 200ms)
Default communication buffer size	Specify communication buffer size between browser.

Automatic adjustment

	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.
Width of client	Specify window width size of browser (Pixel)
Height of client	Specify window height size of browser (Pixel)

プレゼンテーション選択

	Check when it needed to display in Visualization Editor of programming system.

Default text input

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

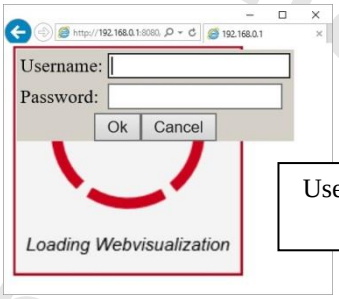
Access method from Web browser

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

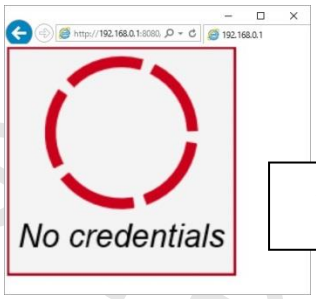
Following display appears after completing access to HX-XPU, and Display of Visualization spears.

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 on-line 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 file deleting
Visualization file registered in HX-CPU, this file is registered dedicated Visu folder inside of HC-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 reset(PLC initialize)

[Device] in the right click menu of device tree.

Element in Tool box

Refer on-line help for further detail of each element specifications.

Caution for Visualization using

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.

List of CAA DTUtil library

Command	Function
GetDateAndTime	Get system clock
SetDateAndTime	Set system clock
GetTimeZoneInfomation	Get time zone information
SetTimeZoneInfomation	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,

Time zone	Configure method	Target
Time zone 1	Function block （SetTimeZoneInfomation）	GetTimeZoneInfomation
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.

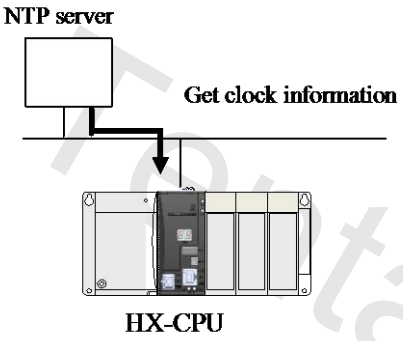
```
PROGRAM POU
VAR
          BLINK_0: BLINK;
          GetDateAndTime_0: DTU.GetDateAndTime;
          SetDateAndTime_0: DTU.SetDateAndTime;
          GET_TIME_ERROR: BOOL;
          SET_TIME_ERROR: BOOL;
END_VAR
```

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

SNTP 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.



NTP Client function

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

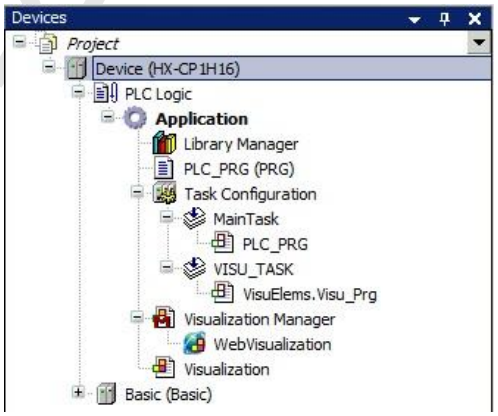
It is possible to get NTP status by using dedicated function block (GetNTPstatus). ExecNormal of GetNTPstatus turn TRUE by getting clock data correctly, therefore if clock data is used in user program, use it after confirming ExecNormal of GetNTPstatus turns TRUE.

Caution

If NTP client function is used, use it after setting time zone. Configuration is done by dedicated function block (SetTimeZoneInformatio). 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...

Configuration Items of NTP client

Item name	Contents	Setting range
NTP server	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 (Changing number is not required in general)
NTP server definition	Select specifying method of NTP server.	IP address (Fixed IP address)
NTP server IP or host name	Specify NTP server.	xxx.xxx.xxx.xxx
Access cycle	Set time interval of clock information.	1-1440 (unit: minute)
Timeout	Set detecting time of timeout.	1-255 (unit: second)
TimeZone	Specify time zone.	UTC-12:00-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:=UsbMountStatus();
//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);
                   lState := 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 possibility 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.

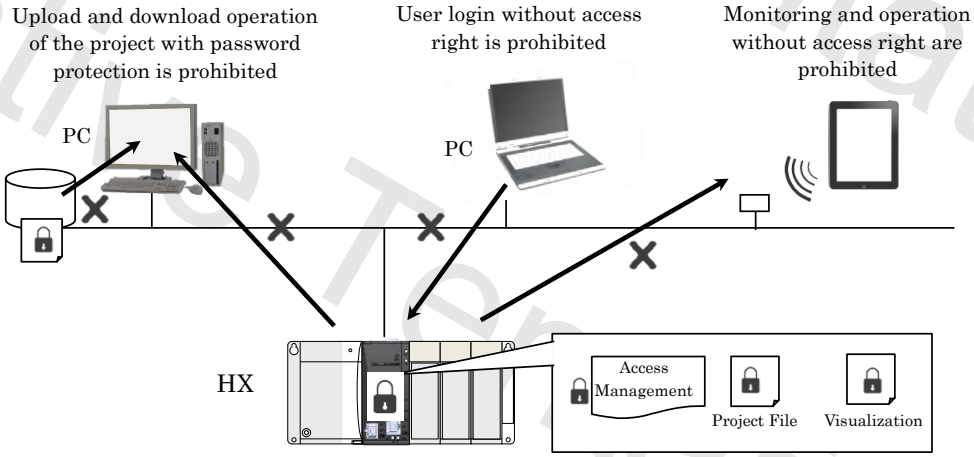


Figure Supporting function for security protection via network access

Limitation of on-line user

It is possible to limit user on-line 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 On-line user]

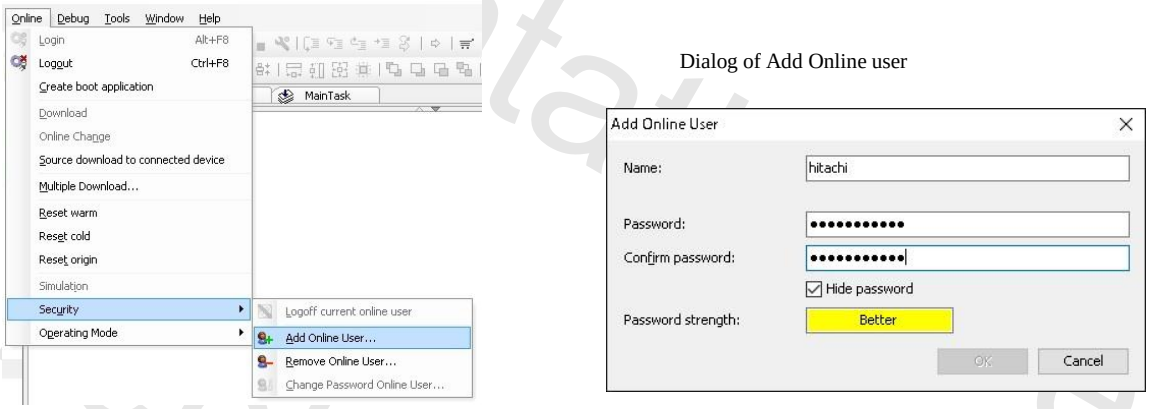


Figure 2.24.2 Add on-line user

Following dialog appears at login after registering on-line user. Input registered name and password.

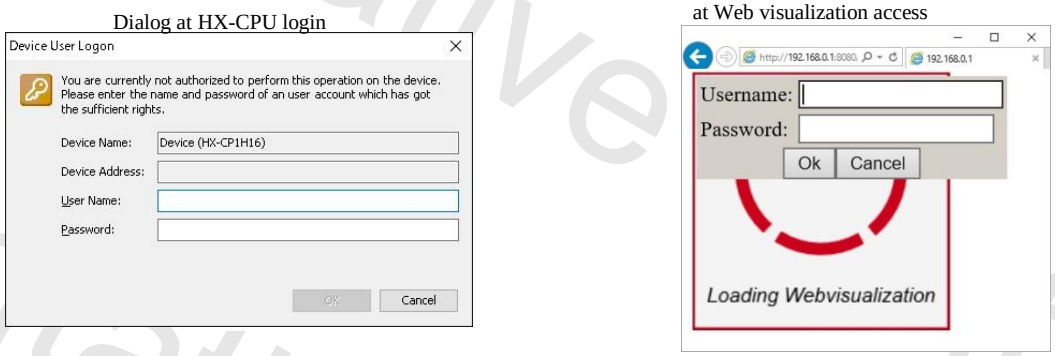


Figure Access of HX-CPU registered on-line user

Dialog at login appears again when user name or password is different from registered. Confirm user name and password with manager registered on-line 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 on-line user registration after logon. If not to use dialog at login, add new user with name "Everyone" and empty password. It is possible login without login dialog from next login.

Caution

Be careful, on-line user is deleted by "Device reset (initialize PLC)" operation.

Password protection of Project

Password protection is possible for project file.

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

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

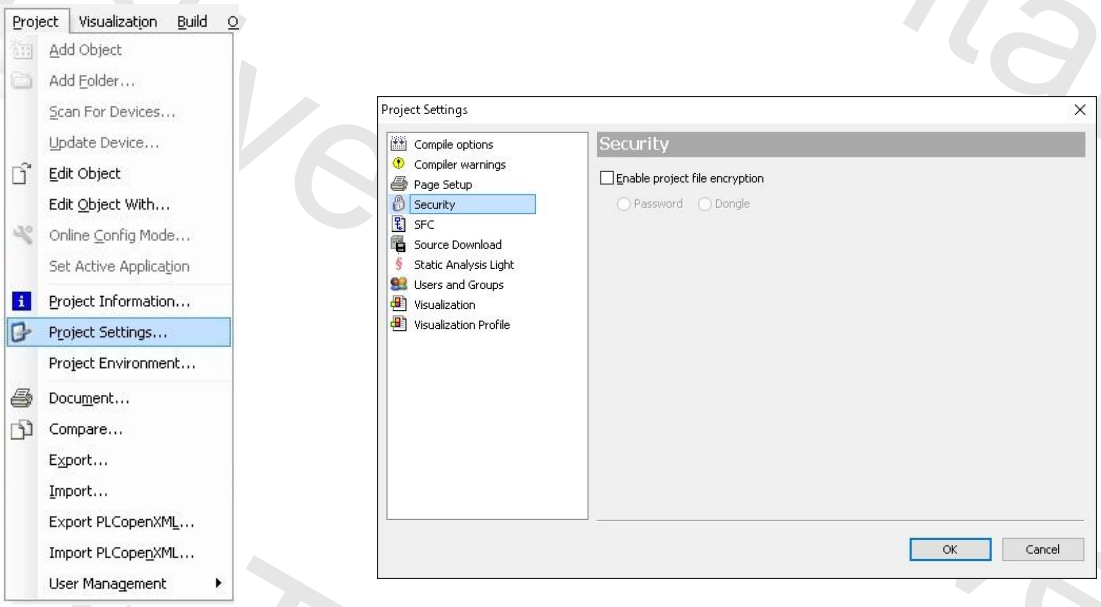


Figure Configuration of Project password

Select “Password” by check on “Enable Coding of Project file”.

Input current password, new password and new password confirming.

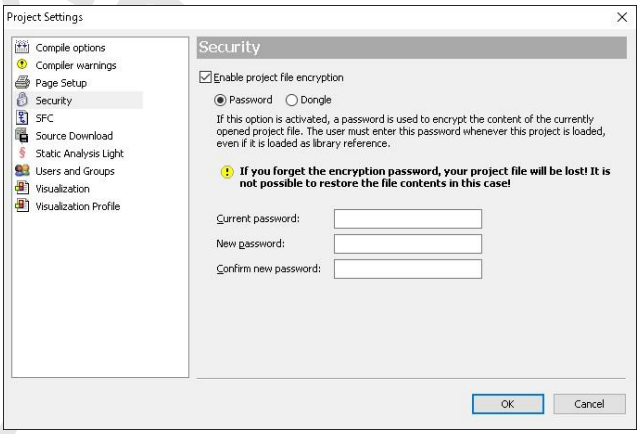


Figure Input Project password

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

以降、保存したファイルを開くときや、HX-CPU に格納されたソースプログラムをアップロードして HX-CODESYS で開く場合の保護となります。

At the time to open project or uploading

In case of password is not correct

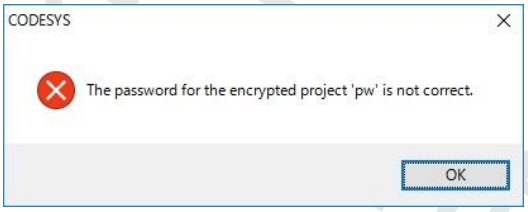


Figure Input Project Password

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.

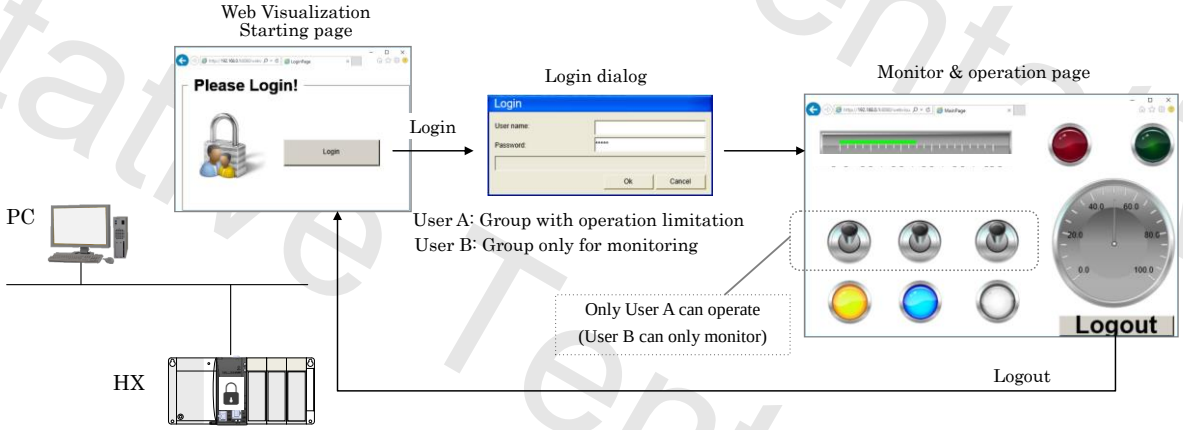


Figure2.24.7 Visualization example with access limitation

User management of Visualization

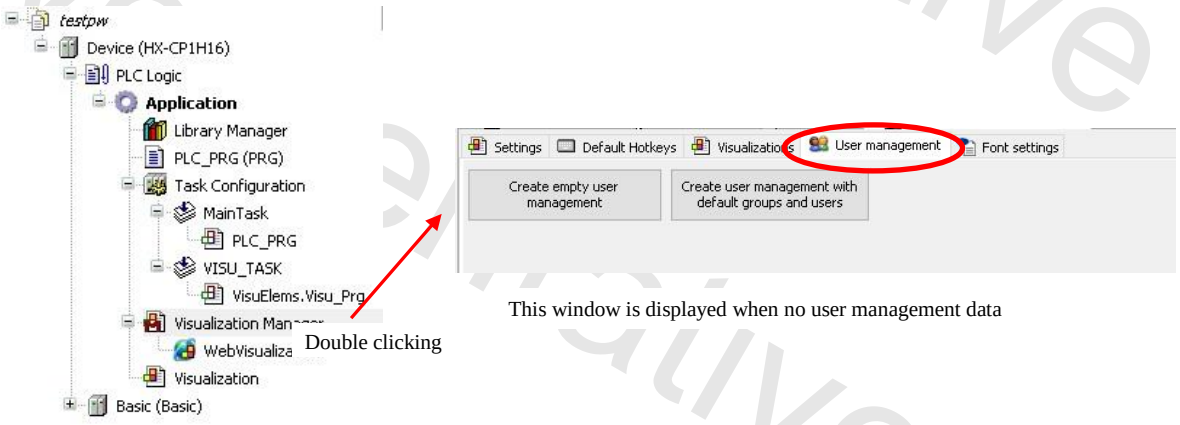
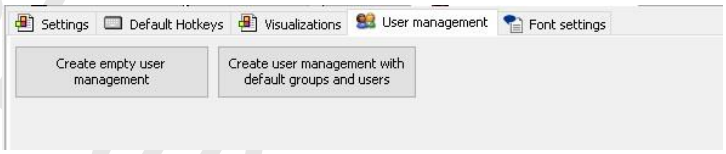
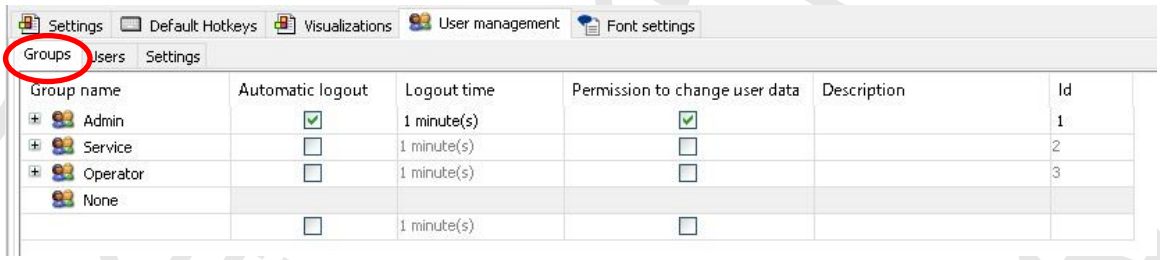


Figure Visualization manager

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.



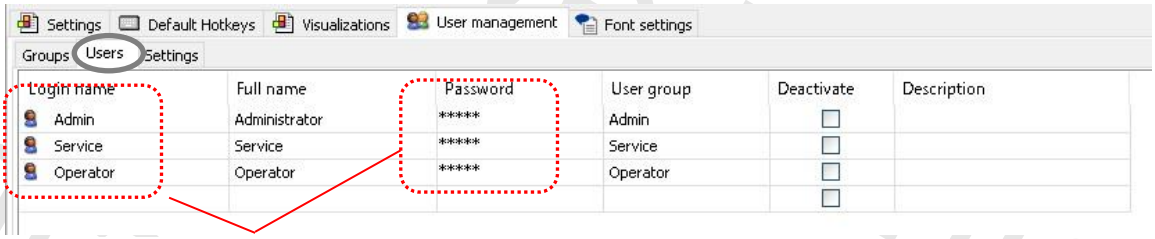
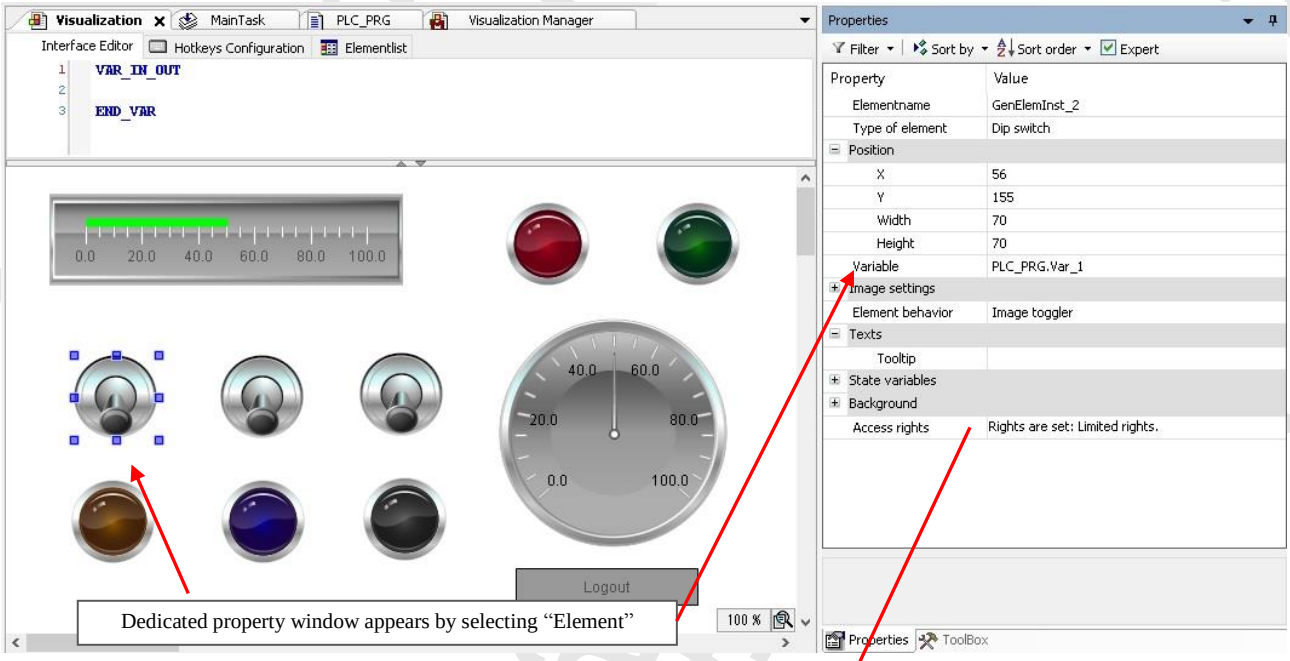
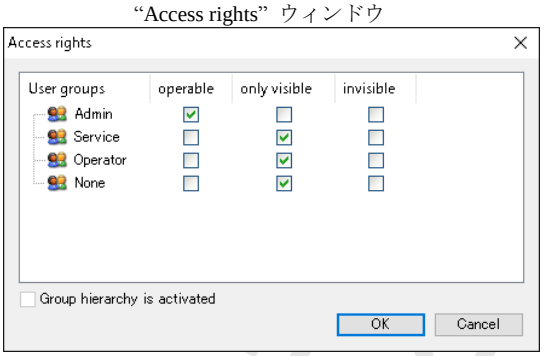


Figure2.24.9 User management Visualization manager

Next is explanation of Element configuration of Visualization.
Configure “Access rights” on the property window of element.



“Access rights” ウィンドウ



Double clicking

Configure other than user of Group “Admin” can’t operate.

Figure 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.

Chapter 5　Debugging function

It is possible to debug program by using HX-CODESYS. In this chapter, following debug functions are explained.

No.	Functions	Refer page
1	Monitor function	5.2
2	Flow control function	5.3
3	Break point function	5.4
4	Single cycle, Step execution function	5.5
5	Set value fourthly, write value function	5.6
6	Trace function	5.7

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

5.1 How to start

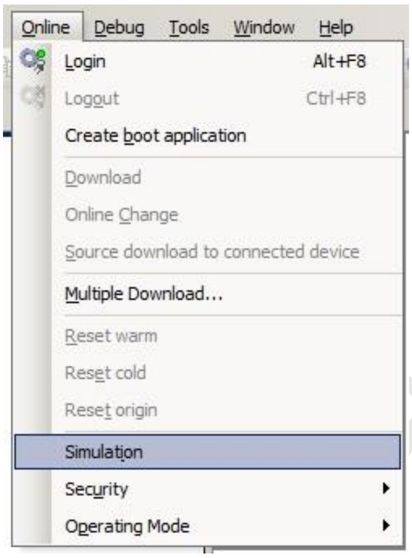
Even it is same operation of debug operation between on-line debug and off-line debug, starting operation is different. Each method how to start are described in below.

How to start off-line debug

Login and start execution.

How to start simulation

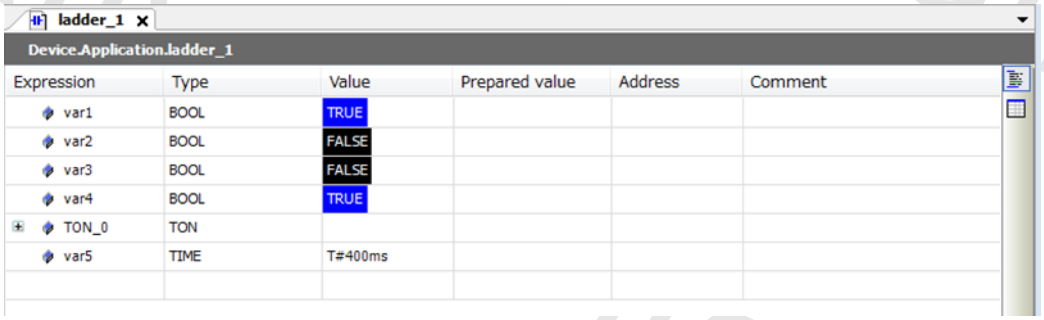
Click “Simulation” of “On-line” menu. Login and start execution.



5.2 Monitor function

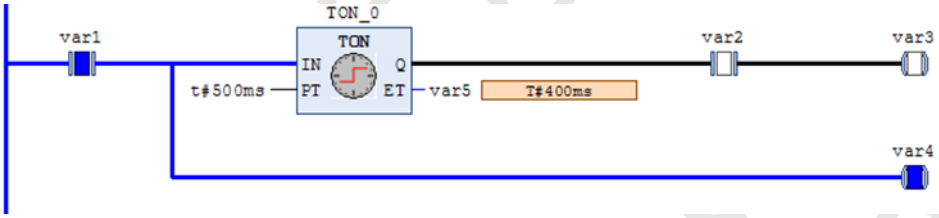
Monitor function is to monitor contact, coil and current value of variable and it is possible to monitor variable declaration part, LD (Ladder diagram logic) display part, FDB (Function Block Diagram) part, ST (Structured Text) display part etc.

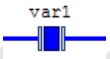
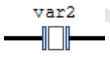
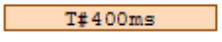
Declaration variable part



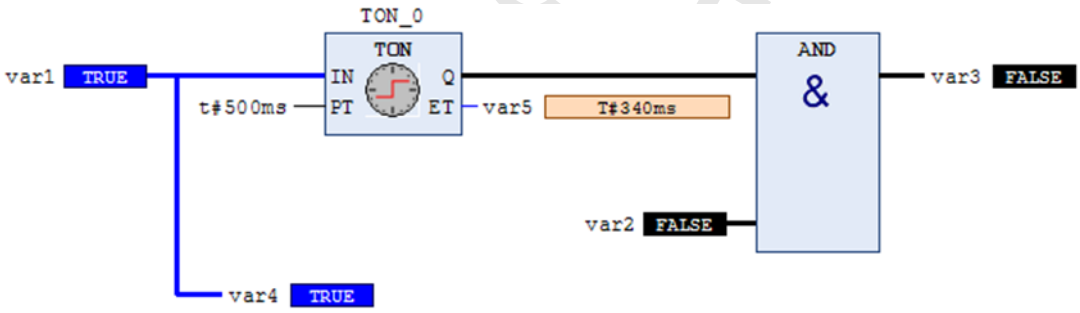
Display	Contents
TRUE	Show variable is ON.
FALSE	Show variable is OFF.
T#400ms	Show value of variable. This shows value of var5 is 400ms.

Ladder diagram



Display	Contents
	Show contact is ON.
	Show contact is OFF.
	Show coil is ON.
	Show coil is OFF.
	Show value of var5 is 400ms.

Function Block Diagram



Display	Contents
TRUE	Show variable is ON.
FALSE	Show variable is OFF.
T#340ms	Show value of variable. This shows value of var5 is 340ms.

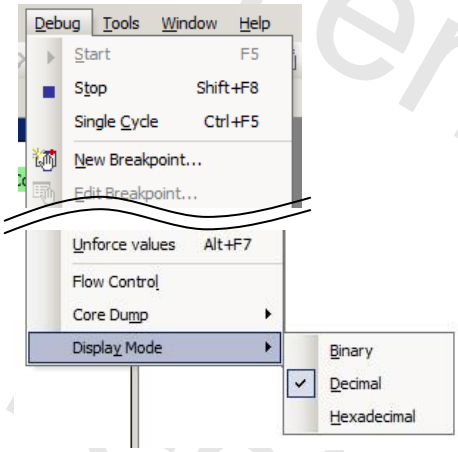
Structured text

```
1  TON_0(IN TRUE :=var1 TRUE,
2        PT T#500ms :=T#500MS,
3        ET T#380ms =>var5 T#380ms,
4        Q FALSE => var6 FALSE);
5  var4 TRUE :=var1 TRUE;
6  var3 FALSE := var6 FALSE AND var2 FALSE; RETURN
```

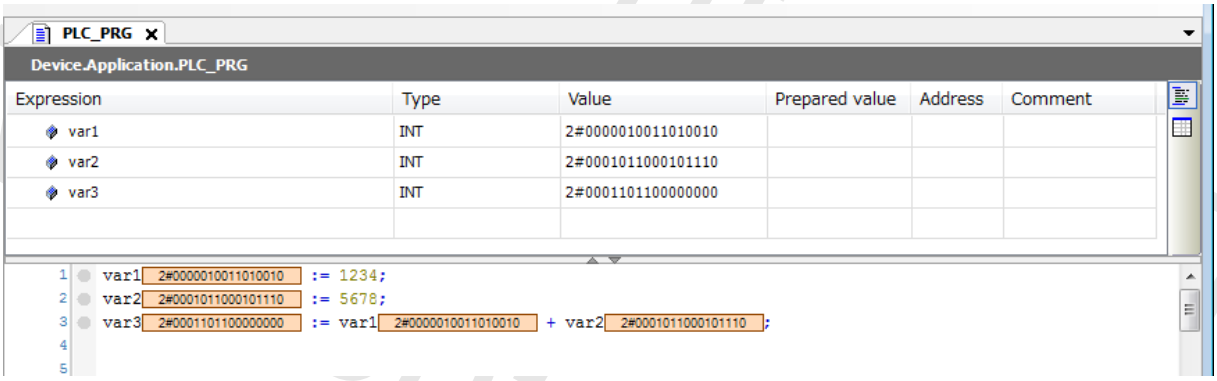
Display	Contents
TRUE	Show variable is ON.
FALSE	Show variable if OFF.
T#500ms	Show value of variable. This shows value of var5 is 500ms.

Change display mode

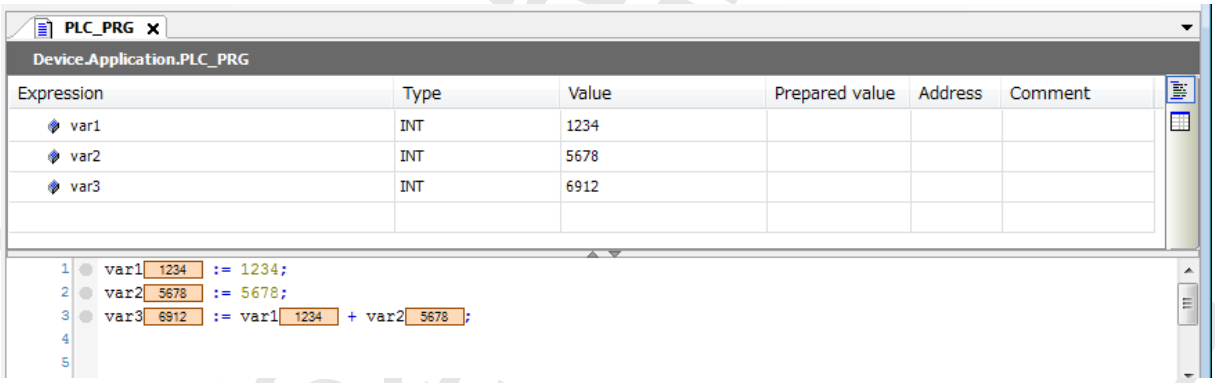
It is possible to change display for variable monitor with binary, decimal or hexadecimal. Select “Debug”-“Display mode”.



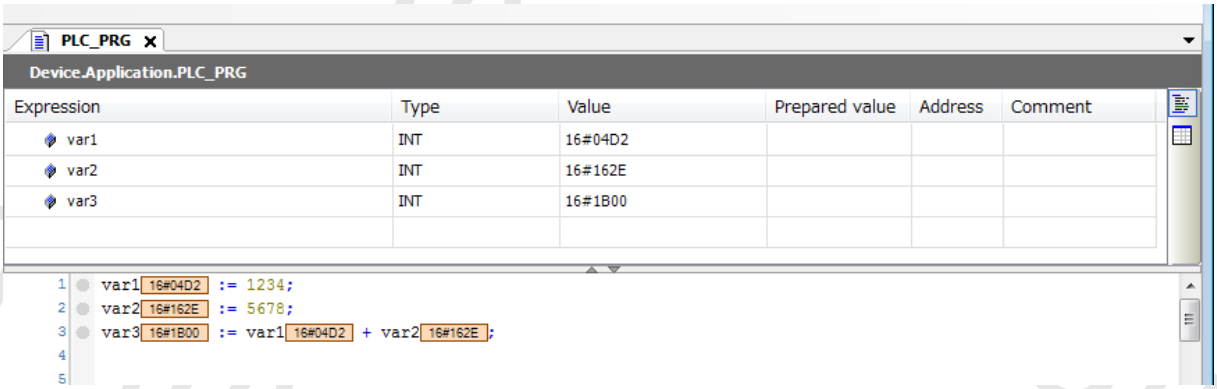
Binary display



Decimal display

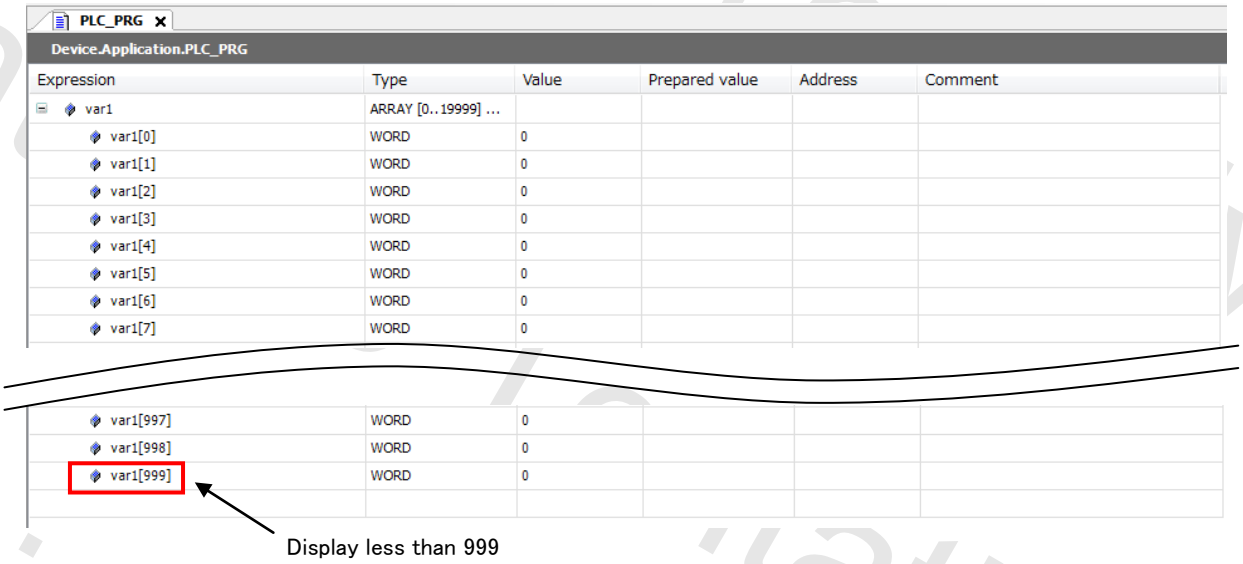


Hexadecimal display

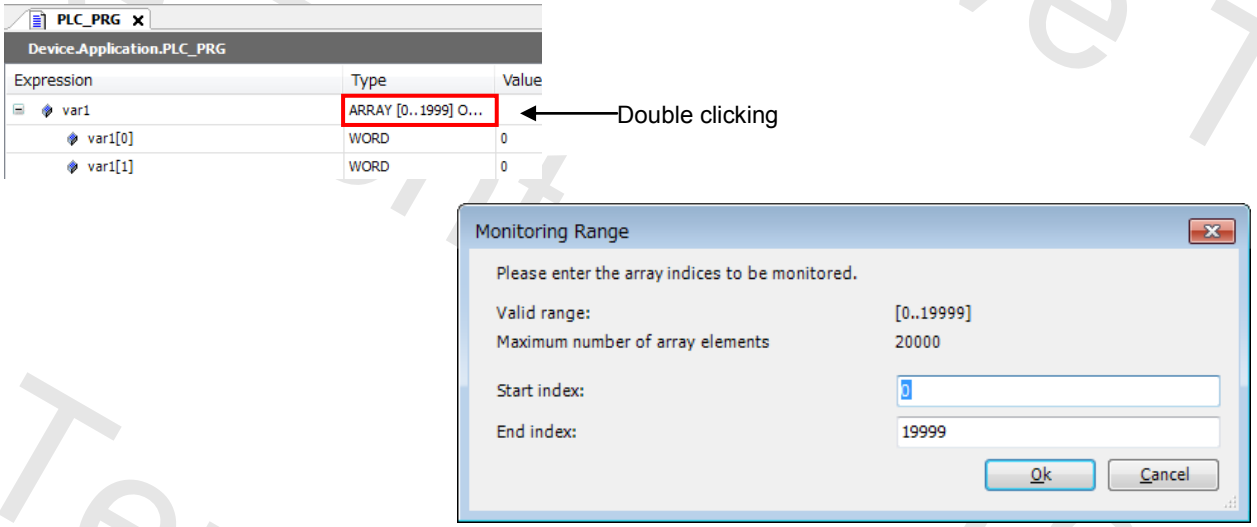


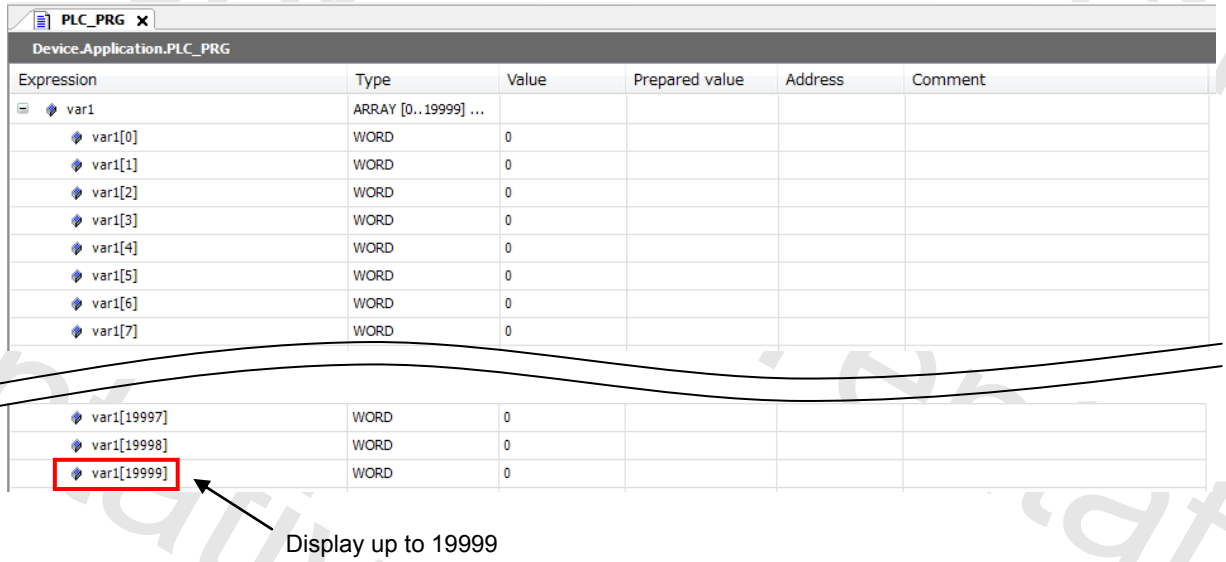
Array variable monitor

It can't be display with list more than 1000 if array declaration exceeds more than 1000 at the default status.



Change range of list display by double clicking "ARRAY[*..***]0.." to monitor array variable more than 1000 after. List display is extended up to 20000 maximum.

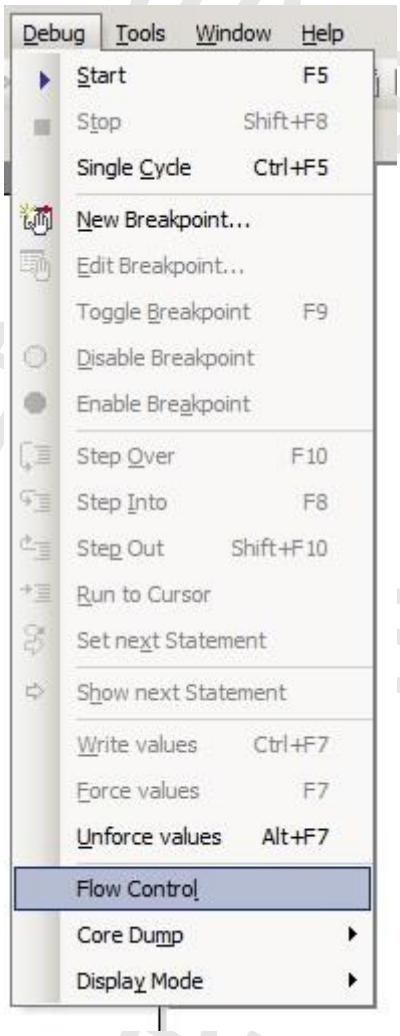




5.3 Flow control function

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

Click “Flow control” of “Debug” after login.



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.

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 function

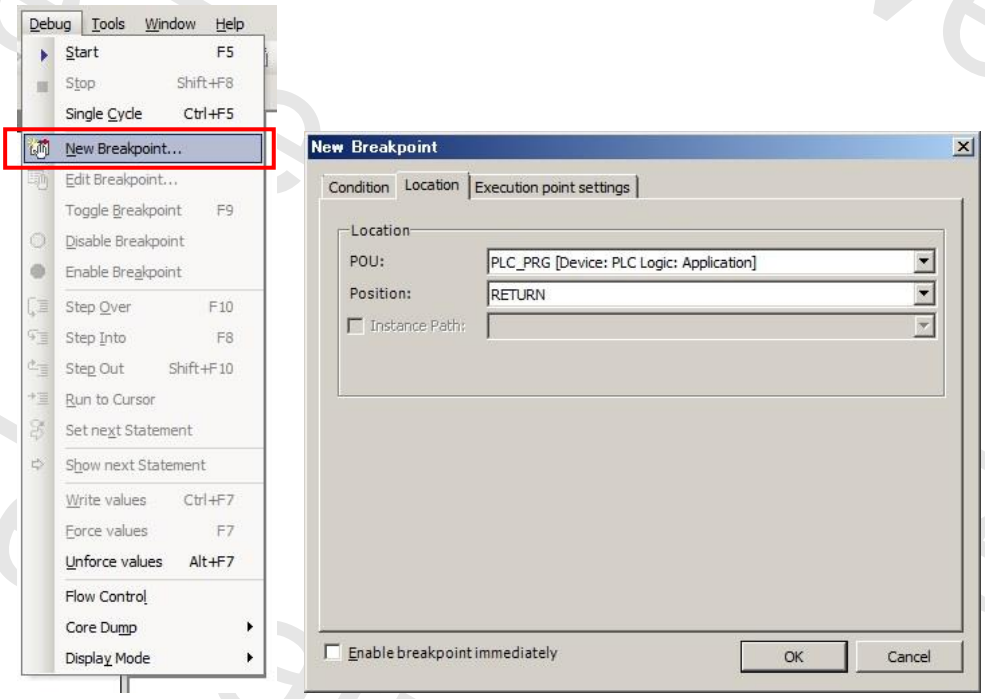
Break point function is possible to stop program at the position specified stopping position of application program for debugging.
The possible break point position is the position can be changed value of variable or the position to call program after branch.

Configuration method of Break point

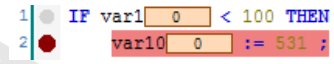
Show example of Break point with structured text.
Grey colored circle appears where break point can be set after login.

```
1  IF var1  0  < 100 THEN
2      var10  0  := 531 ;
3  ELSE
4      var10  0  := 257 ;
5  END_IF RETURN
```

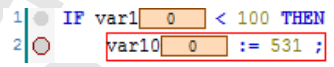
After click “New Break point” of “Debug” menu, property screen of Break point appears and then click “Location” tab. Specify POU to set Break point at “POU”. Specify the line to set Break point at “Position”. Check “Enable Break point immediately” to enable break point immediately after specifying.



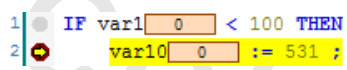
Display red colored circle at valid Break point.



Display grey colored circle at invalid Break point.



Display yellow colored allow at the position program stooped.

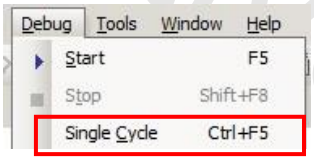


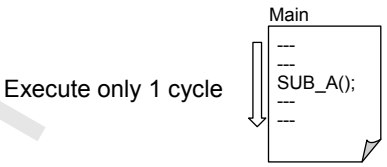
5.5 Single cycle Step function

5.5.1 Single cycle execution function

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

Execute only 1 cycle of program by clicking “Single cycle” of “Debug” menu. If there are several tasks, all tasks are executed only 1 cycle instead of task cycle.





5.5.2 Step execution function

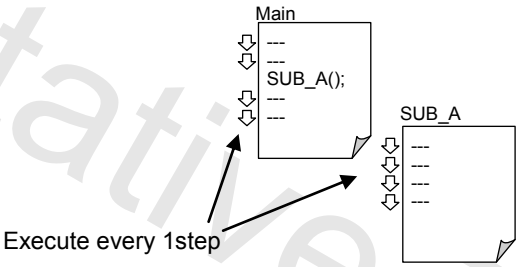
There are four kinds of step execution function in HX-CODESYS.

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

(1) Step-in function

Step-in function execute every one step of function or function block.

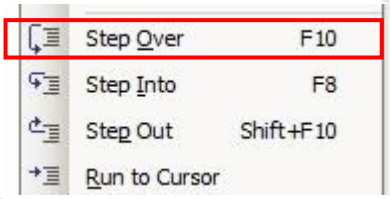
Click “Step-in” of “Debug” menu to execute Step-in.

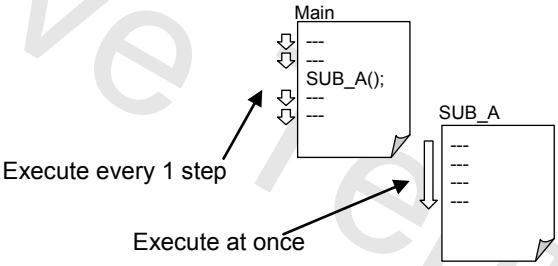


(2) Step-over function

Step-over function execute function or function block at once.

Click “Step-over” of “Debug” menu to execute Step-over.



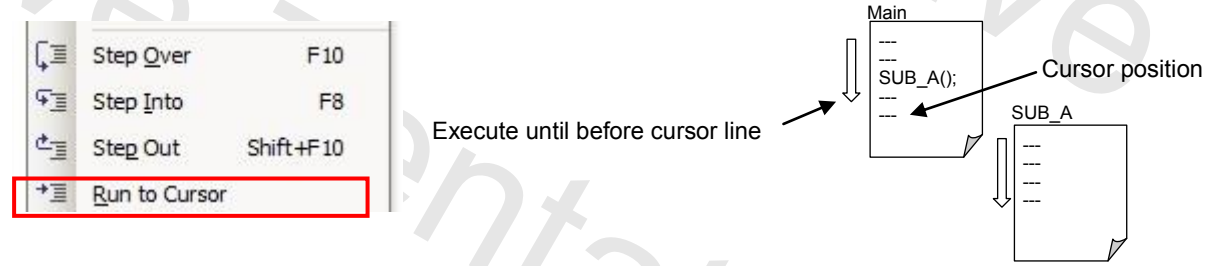


(3) Step-out function

If execute Step-out during the execution of function or function block by Step-in function, execute all program of function or function block and go through original program from function or function block.
Execute Step-in by clicking “Step-out” of “Debug” menu.

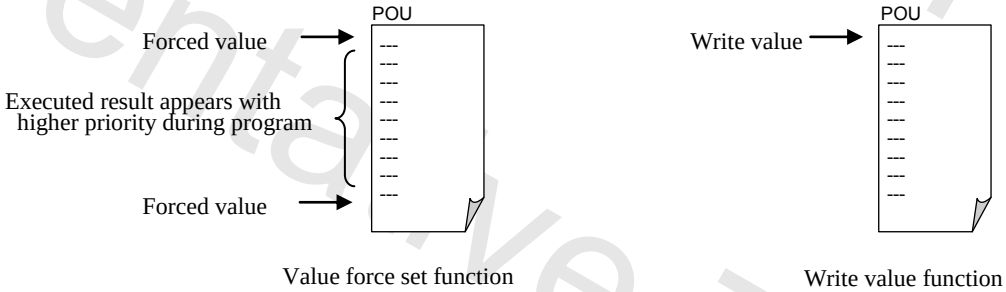
(4) Execute until cursor line

Execute until cursor line execute program until position of cursor on the line.



5.6 Value force set and write value function

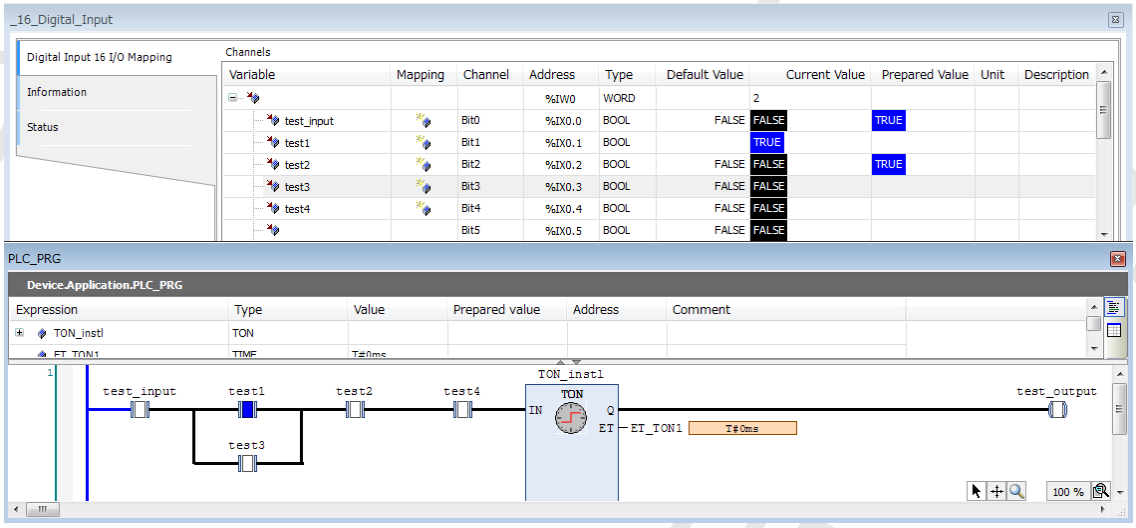
Value force set and write value function changes value of variable by other value. Value force function sets forced value when start and end of program. Write value function set value only once at the start of program.

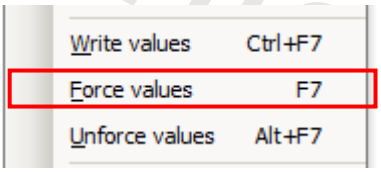


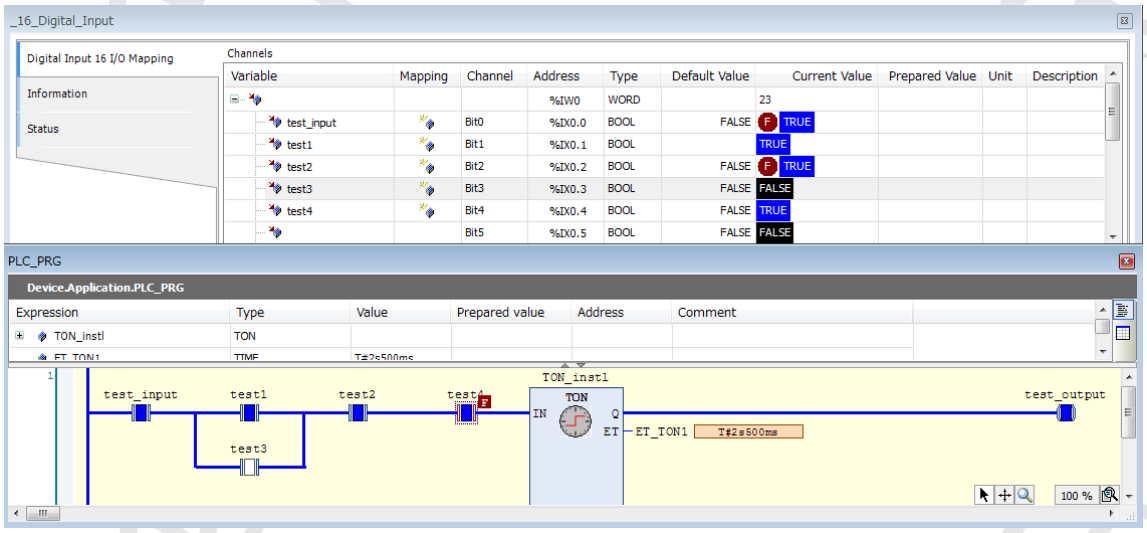
> **⚠ Caution**
>
> There is some risk like malfunction of device, danger of human depends on target system caused by use of Value force set and write value function. Please test enough and confirm stable operation of target system with value of variable to use for Value force set and write value function.

Configuration method of force function

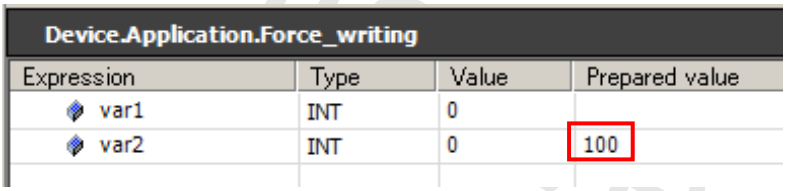
Set writing value of variable by double clicking “Prepared value” of variable declaration part.



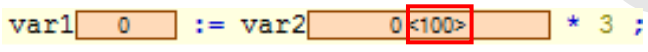




It is possible to set value for variable similar way.



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



Configuration methid of Write value function

Click “Write value” of “Debug” menu 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.

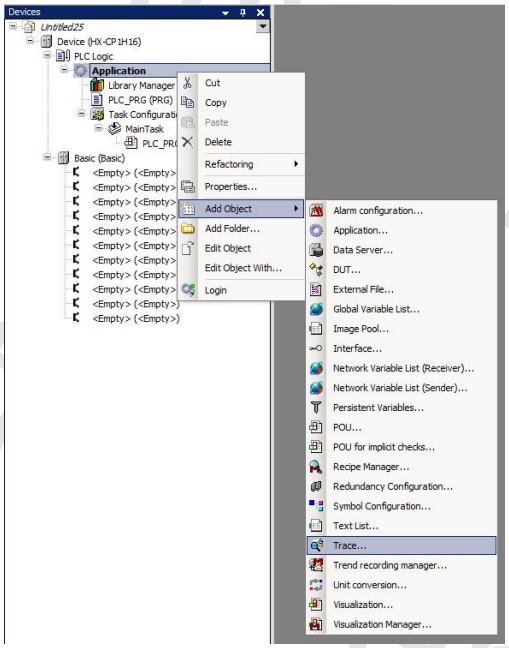
5.7 Trace function

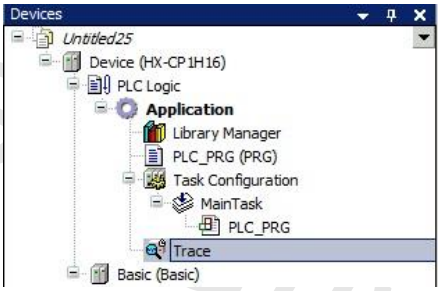
Trace function samples variable without dedicated program.

Configuration method

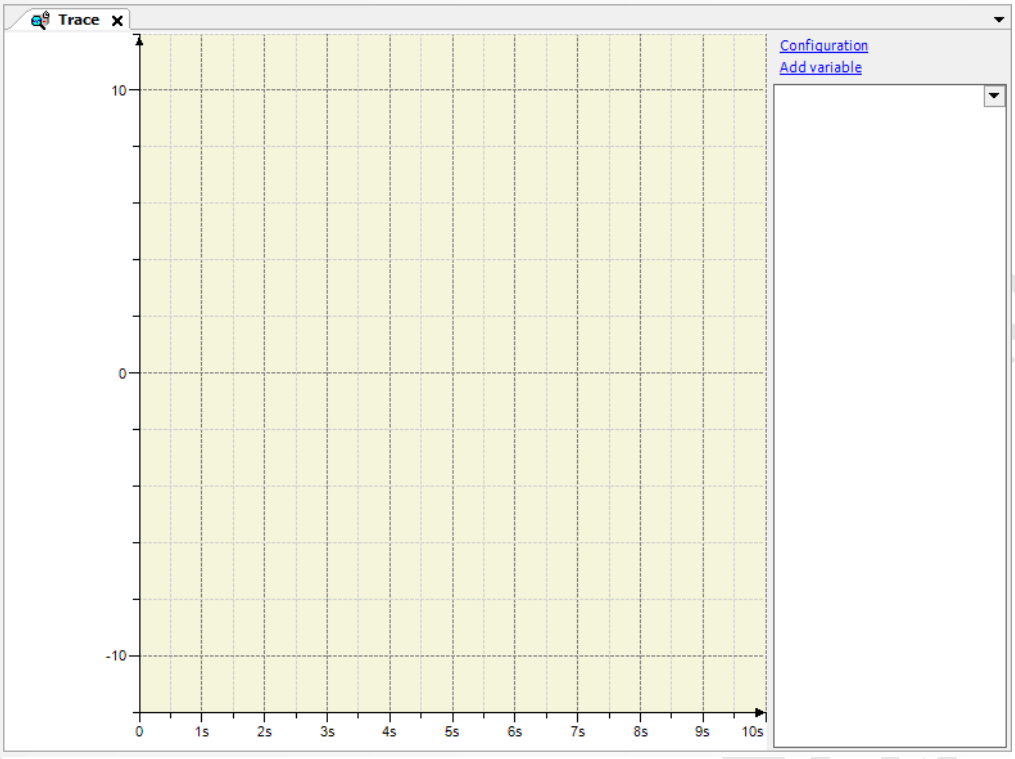
Create trace object

Add trace object by clicking “Add Object” ”Trace” after right clicking “Application”.

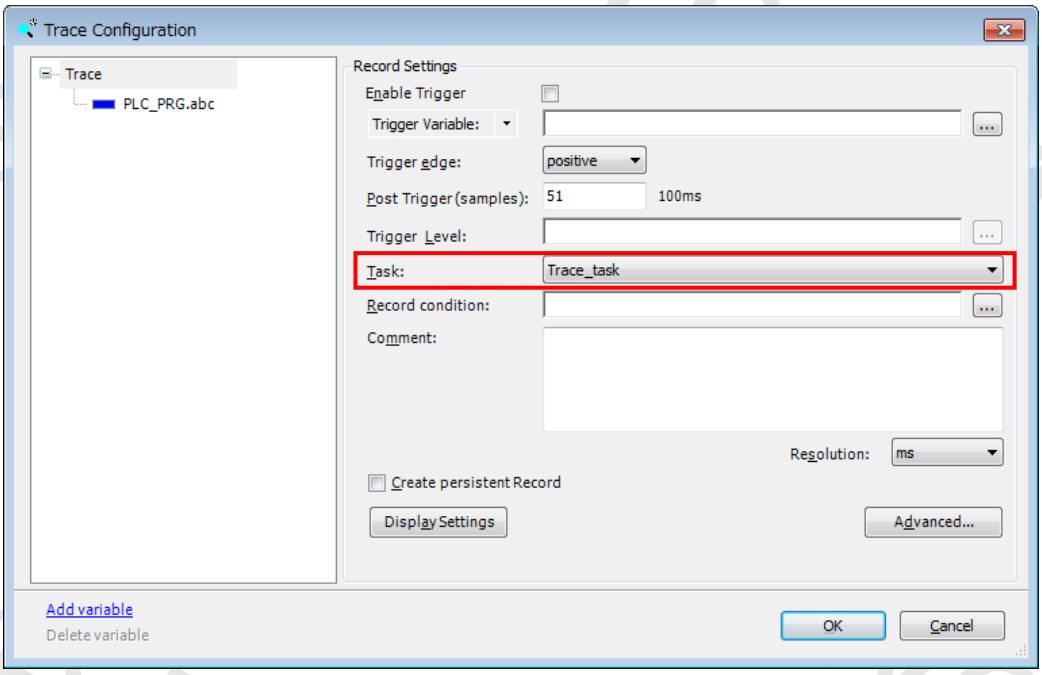




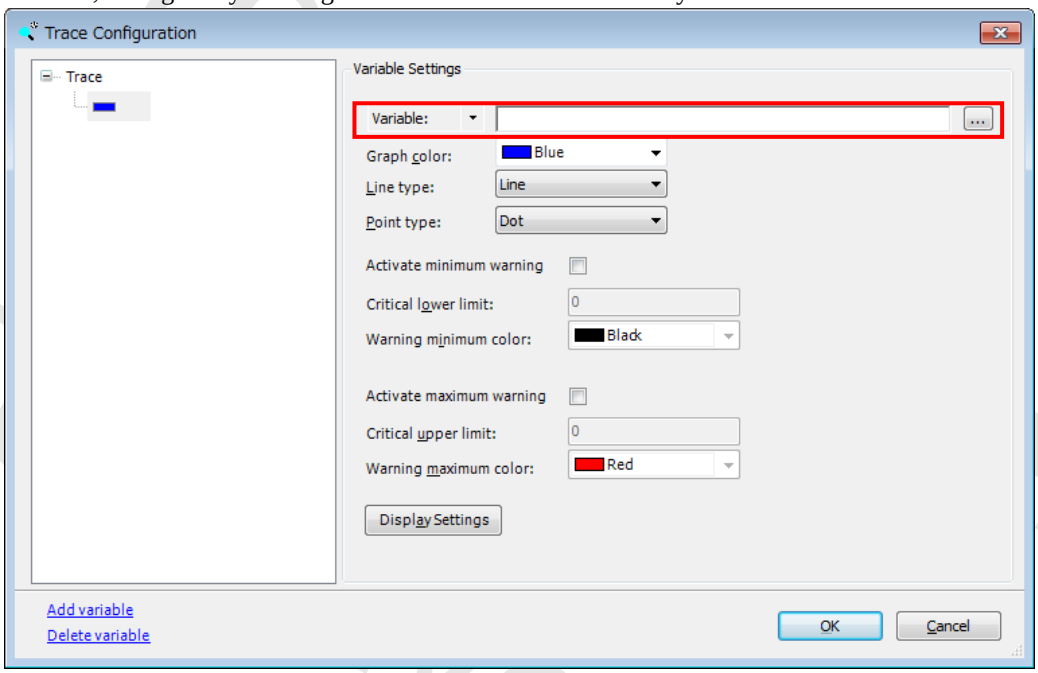
Trace editor appears by double clicking “Trace” object.



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.

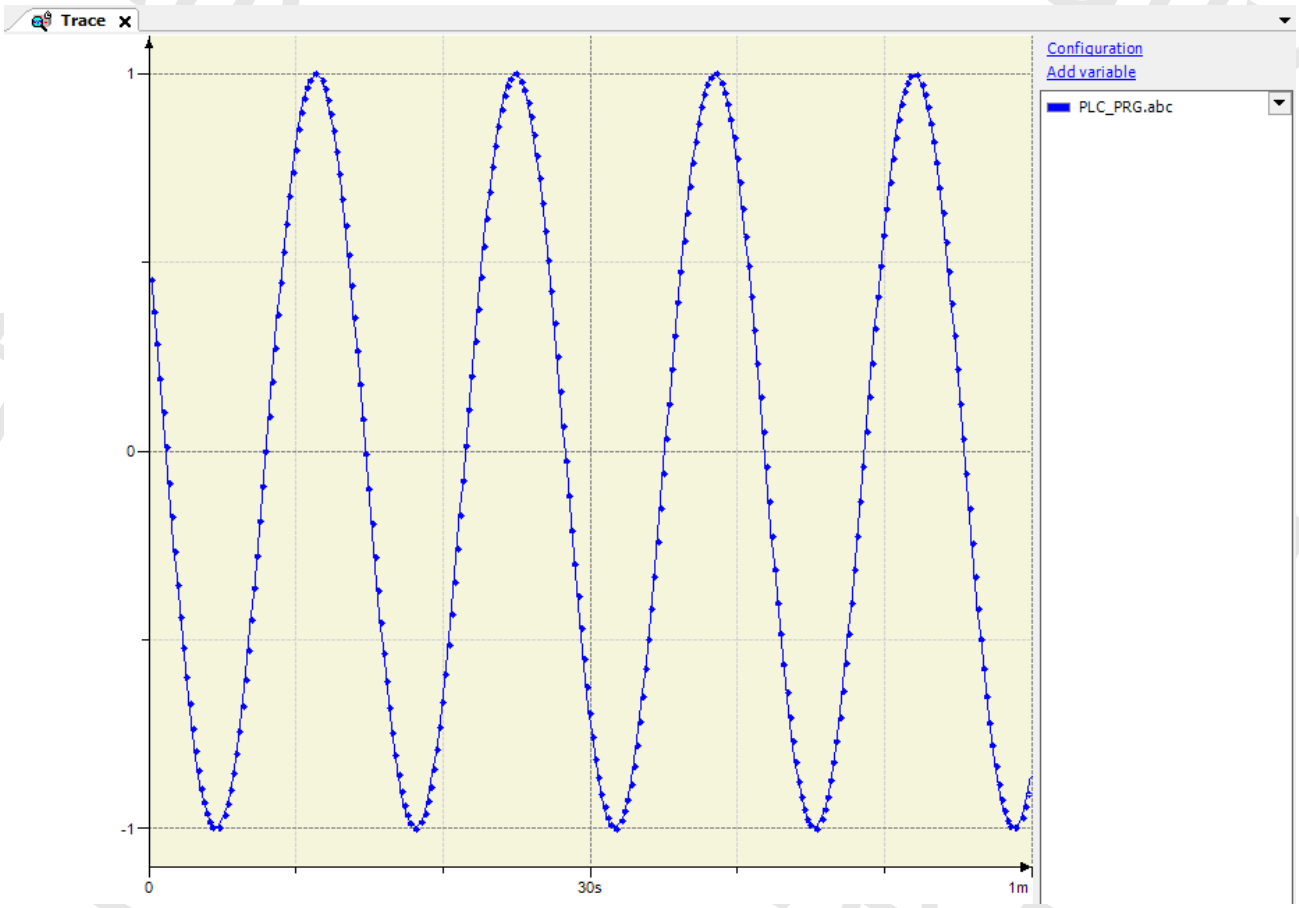


Configure trace variable at “Variable” by clicking “Add variable”. Configure other items if it is needed. In case of adding new variable, configure by clicking “Add variable” with similar way.



Download trace data to HX-CPU by clicking “Download Trace” of “Trace” menu after login.

Display starts immediately according to the configured contents.



Following operation is possible after displaying trace data on “Trace” menu. Please refer on-line help for further detail information.

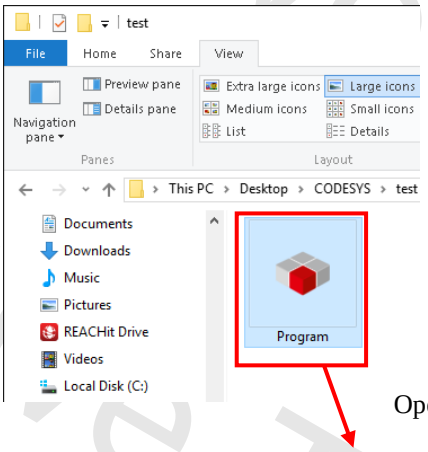
-Start/Stop of Trace data
-Customize display graph
-Access Trace data of HX-CPU
-Save/Read of Trace data

Appendix Known Restrictions

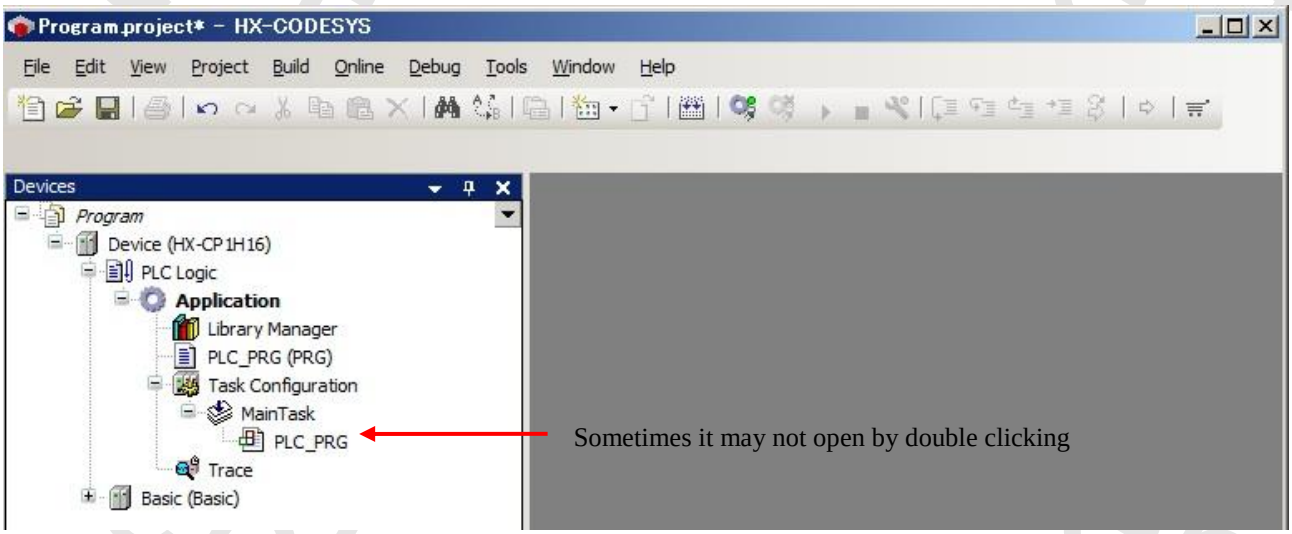
Following restrictions has been recognized with HX-CODESYS V3.5 SP8 Patch4. These are depend on the based software CODESYS V3.5 SP8 Patch4 from 3S-Smart Software Solutions, and will be improved later version.

[Open project]

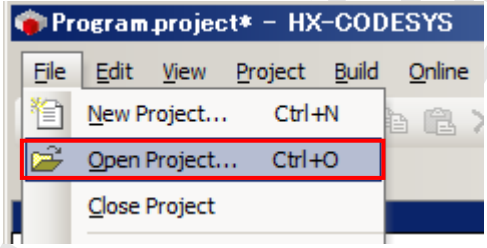
When open project file by specifying file from saved project file, sometimes POU can't be opened.



Open



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



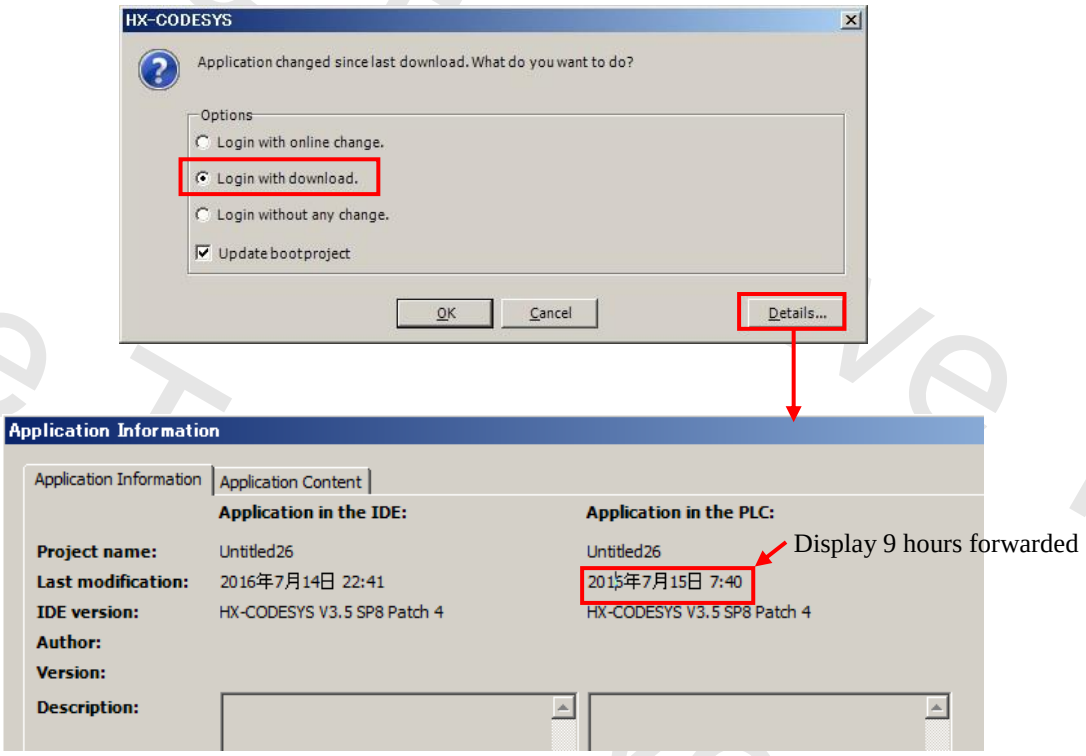
[Open project file]

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

[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 "Download with login". This is no problem on execution.

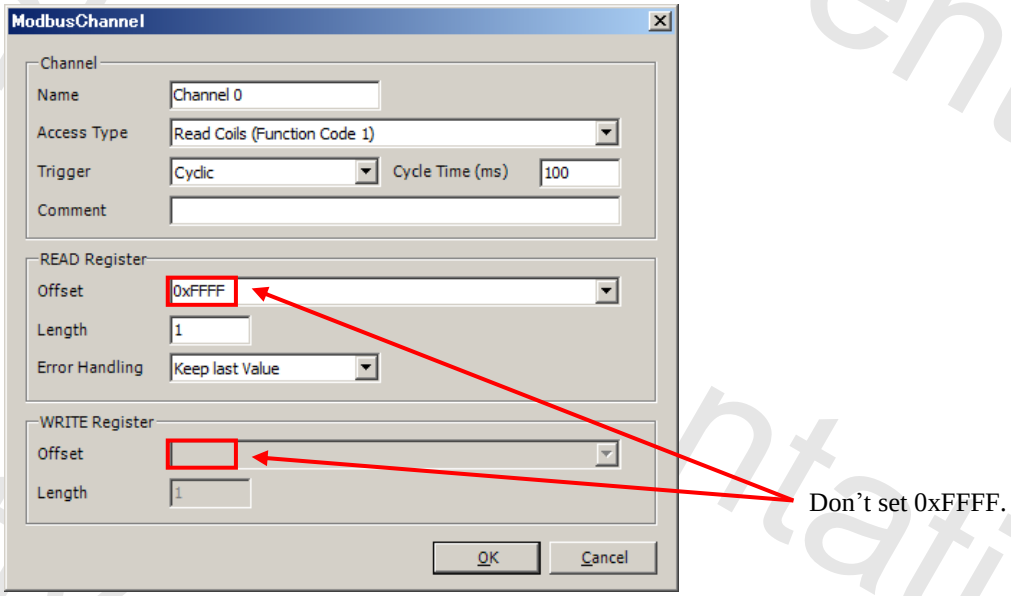
*Latest updating day is displayed correctly, when previous download done by "On-line change with login".



[Modbus-RTU Master]

[Modbus channel offset]

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



[Modbus-RTU Slave]

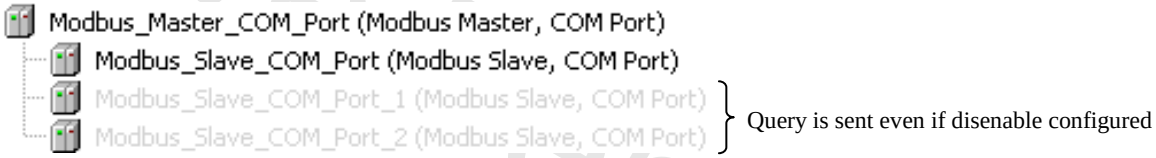
[Disenable of Device]

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

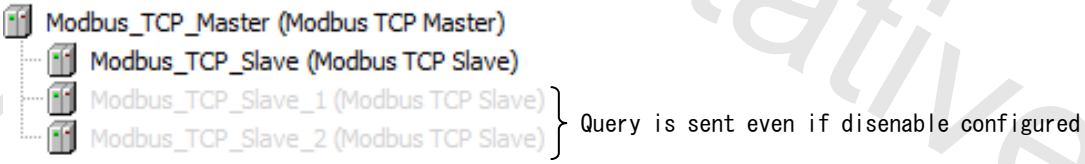
※In case of Modbus-RTU: ModbusSlaveComPort_Diag

In case of Modbus-TCP: ModbusTCPSlave_Diag

In case of Modbus-RTU



In case of Modbus-TCP



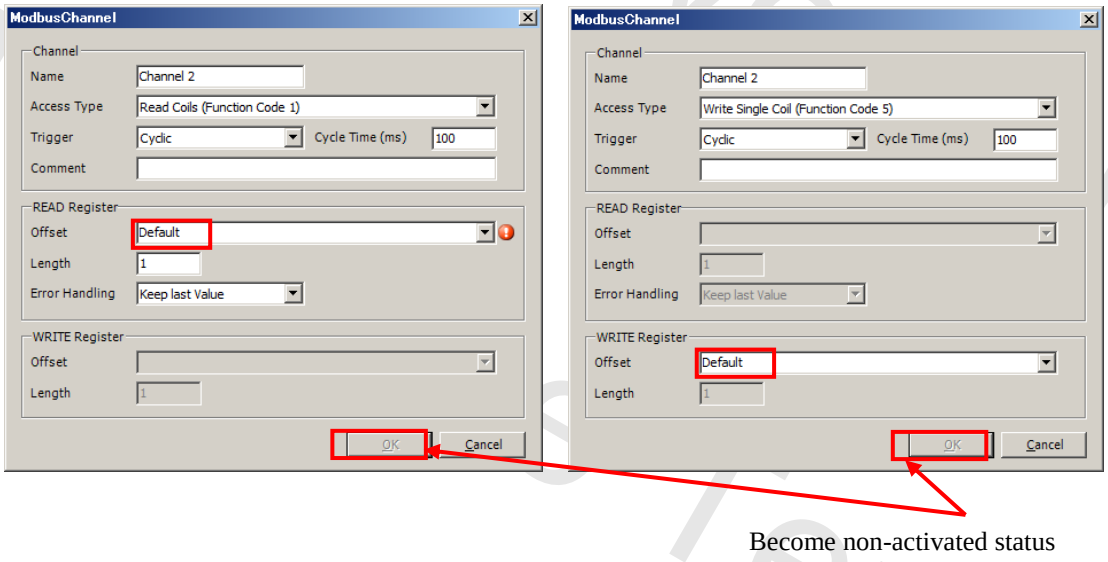
[Modbus-TCP Master]

[Chanel 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.



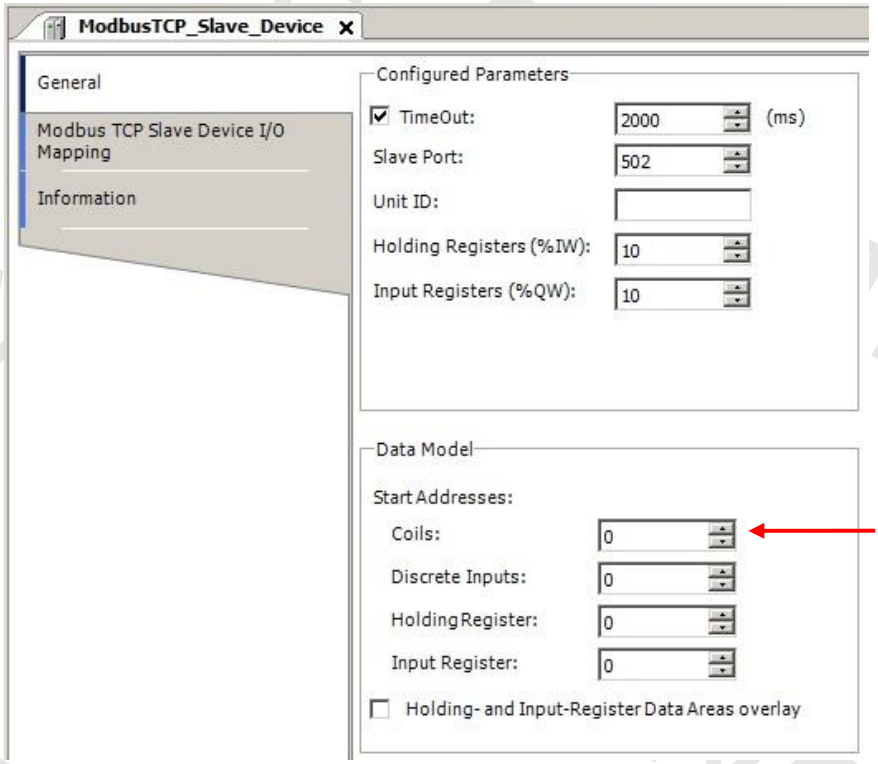
[Modbus-TCP Slave]

[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