HCFa
禾川科技

Q Series Medium-Sized PLC

Software Manual

ATC

Catalogue No.ATC/MQXS2020

June 15, 2021

Preface

Since the twenty-first century, with the rapid development of information technology， traditional industrial control mode finds it harder and harder to meet needs of miniaturization, intellectualization and networking. The market and technology background call for more research on open multi-axis motion controller with new architecture, new technology, high performance ， and low cost.

In early time, only some international companies can program limited types of PLC, which can be used in few areas with high price while poor maintenance. Also， the life cycle of PLC is short due to quick updating, which requires programming devices based on computers. That's why HCFA Technology Co. Ltd. (HCFA) choose CODESYS as platform of our Q series PLC. CODESYS, a powerful national platform for industrial automation software development, provides rich programming languages (IEC61131-3) for project programming of different area and industry, and also supports data interaction of high-level languages. Programming ideas, architectures and complex algorithm of computer area are introduced to industrial controlling area by CODESYS, also equipped with functions like data monitoring, acquisition， and analysis. More types of hardware interface are realized by Q series PLC, like Ethernet, EtherCAT, CANOPEN, RS232, to realize communication with internet and multiple industrial field-bus protocols. High speed IO is also equipped to realize acquisition and output of high-speed pulse signal. Meanwhile, USB 3.0 makes it very convenient for data storage and transfer between PLC and outside peripheral devices.

As a big manufacturing country, China's controlling technology still has a long way to reach developed countries like some European countries, America and Japan, which makes their mature products only choice of our users for a quite long period. However, with more and more national investment and research on embedded controllers, and less restrictions on hardware, we believe that more products like Q series and companies like HCFA will emerge to lead Chinese automation and information technology of industry.

Applicable Reader

Users of HCFA Q series PLC, with basic knowledge of computer and automation, shall realize hardware configuration, software programming and debugging based on this book.

Main Contents

Chapter 1 introduces CODESYS and its programming specification IEC61131-3.

Chapter 2 introduces software installation and interface.

Chapter 3 introduces software architecture of CODESYS and configuration of PLC applications.

Chapter 4 is about introduction and application of HMI project tool.

Chapter 5 introduces creation of new CODESYS project, including hardware adding, downloading communication and program downloading, monitoring and debugging.

Chapter 6 is about realization of multiple communications, including TCP/IP、 Modbus TCP/RTU、OPU UA and EtherCAT.

Terms and Abbreviations

Diagram 1.1-1 Terms and abbreviations

Term/ABB	Description/Full name
Q1	HCFA's medium-size PLC
POU	Program Organization Unit
PLC	Programmable Logic Controller
DUT	Data Unit Type

Catalog

Chapter 1 System Overview 4
1.1 Documentation File 4
1.2 Soft PLC Solution 4
1.3 CODESYS Introduction 4
1.4 IEC61131-3 Programming Specification 4
1.5 Document Update and Publish Status 5
Chapter 2 Software Installation and Interface Introduction 6
2.1 Environment requirements of Installation 6
2.2 Steps of Installation 6
2.3 Version Management 8
2.4 Help System 8
2.5 CODESYS Interface Introduction 9
Chapter 3 CODESYS Software Model 10
3.1 CODESYS Software Architecture 10
3.2 Device Manager 11
3.2.1 Package Manager 11
3.2.2 Device Repository 12
3.3 PLC Application 12
3.3.1 Library Repository 12
3.3.2 Task Configuration 18
3.3.3 PLC Programming 22
3.3.4 Data Unit Type 30
3.3.5 Recipe Manager 37
3.3.6 Upload and Download of Source Code 43
3.4 Trace 45
3.5 Persistent Variable 51
Chapter 4 Visualization Interface Edit 52
4.1 Introduction of Visualization 52
4.2 Create New Visualization 53
4.3 Basic Visualization Operation 54
4.3.1 Add Visualization Element 54
4.3.2 Align Visualization Element 54
4.3.3 Delete Visualization Element 55
4.3.4 Change Visualization Sequence 55
4.3.5 Adjust Size and Position of View Element 56
4.4 Tool Box 57
4.4.1 Basic Control Tools 57
4.4.2 Common Control Tools 64
4.4.3 Alarm Manager 76
4.4.4 Measurement Control Tools 84
4.4.5 Lamps/Switches/Bitmaps 90
4.4.6 Special Controls 92

Chapter 5 Creation of Simple PLC Project 99
5.1 Start CODESYS 99
5.2 Create New CODESYS Project 99
5.3 Establish Communication with Q1 100
5.4 Creation of PLC Project 102
5.4.1 Add Library 103
5.4.2 Task Setting 104
5.4.3 Programming of PLC Program 105
5.5 Add Trace to monitor program variable 108
5.6 Program download and online monitoring 110
5.6.1 PLC Program Compiling and Download 110
5.6.2 Login and Start 110
5.6.3 Online Watching 112
5.6.4 Reset 113
5.7 Simulation 114
5.8 PLC Shell 116
5.9 Implicit check function of program 119
5.10 Add Extension. 124
5.11 Create Motion Project 125
5.11.1 Add SoftMotion Project 125
5.13 Modify EtherCAT master information and communication parameter 127
5.14 Realize of single axis control command 128
Chapter 6 Communication Setting 131
6.1 Configuration of Ethernet TCP/IP 131
6.1.1 TCP/IP Overview 131
6.1.2 TCP/IP Protocol application 131
6.2 Modbus TCP Configuration 132
6.2.1 Modbus protocol overview 132
6.2.2 Modbus TCP overview 133
6.2.3 Modbus TCP Model 133
6.2.4 Modbus TCP Data message structure 133
6.2.5 Modbus TCP Master application 135
6.2.6 Modbus TCP Slave Application 141
6.3 Modbus RTU Configuration 143
6.3.1 Modbus RTU Overview 143
6.3.2 Modbus RTU Master Application 143
6.3.3 Modbus RTU Slave Application 147
6.4 OPC UA Configuration 149
6.4.1 Introduction of OPC UA Protocol 149
6.4.2 OPC UA Component 150
6.5 EtherCAT Configuration 150
6.5.1 EtherCAT Protocol Overview 150
6.5.2EtherCAT Master Application 150

Chapter 1 System Overview

1.1 Documentation File

This file is only for CODESYS V3.5 SP14 and please download the right version and then operate as instructed. Corresponding routine will also be provided for user's reference. The file may update together with the controller as details found in chapter 1.5 and also on HCFA's website. Any suggestion or correction of the file, welcome to contact us via email:

400@hcfa.cn

1.2 Soft PLC Solution

Based on structure of hardware, PLC can be divided into hard PLC and soft PLC. Traditional hard PLC execute commands through hardware or certain ASIC chip. While soft PLC, also called Softlogic, realizes its function through PC or embedded controller. Or we can say that it's executed via software sealed in PC or embedded controller.

Soft PLC technology is to realize PLC function of all hardware through industrial processing computer(IPC) or embedded controller's hardware and software. With development of computer technology, computer standardized communication protocols and LAN technology make networking and data interaction between PLC and outside more convenient, while inconvenience o hard PLC more obvious. Soft PLC is a better choice for future PLC trend and Industry 4.0 and middle size Q series is HCFA's new effort on that.

Soft PLC is an integration of functions of computer and PLC like switch value control, analog quantity control, mathematical operation, numerical processing, network communication, PID adjust. Through one multi-task control kernel, it provides strong instruction set, quick and correct scan cycle, reliable operation, connectable with types of IO system and net with open structure. Compared with hard PLC, it has advantages as below:

① Open structure for types of IO and bus interface.
② Easier maintenance and closer to international standard.
③ Full use of PC source for better software and hardware platform.
④ Lower cost and united standard for better competing environment.

1.3 CODESYS Introduction

CODESYS, full name in Controller Development System, was developed by 3S(Smart Software Solution GmbH), which is headquartered in Bavaria, Germany.

CODESYS is the reliable environment for PLC developing, which supports IEC programming language. The editor and debugger are based on high-level languages like visual C++. It's presently widely used by companies like ABB, BECKOFF, Bachmann, EPEC, Rexroth, etc.

1.4 IEC61131-3 Programming Specification

In March of year 1993, IEC published IEC 61131, which took advanced ideas and technology of information area to industrial control area(like software project, structural and module programming, net communication, etc). This fixed disadvantages of traditional PLC, DCS control system(like in openness,

compatibility, software maintenance and reusability)

IEC631131 is the first international standard of PLC programming and IEC61131-3 is the foundation of united PLC programming language, including two types(textual and graphical programming language), six languages in total. Textual: instruction list(IL) and structured text(ST). Graphical: ladder diagram(LD), function block diagram(FBD), sequence function chart(SFC), continuous function chart(CFC). Multiple languages can be taken to serve the same project, which can optimize the program, reduce dependence of single supplier, increase readability, safety and reduce cost of maintenance.

1.5 Document Update and Publish Status

Diagram 1.5-1 Document Update and Publish Status

Publish time	Manual No.	Update	Publish status
2019/5/28	ATC/MQS01-1.0	1st version	Formally published
2019/8/7	ATC/MQS01-1.1	2nd version	Formally published
2020/5/11	ATC/MQ1S20111	3rd version	Formally published

Chapter 2 Software Installation and Interface Introduction

2.1 Environment requirements of Installation

2.1-1CODESYS V3.5 is the host programming software, which supports programming, debugging and hardware configuration. Due to its complexity and need of dealing with data, it has related requirements on PC hardware and system environment. Least requirements and recommendation as below diagram 2.1-1.

Diagram 0-1 Environment requirements of Installation

Description	Minimum configuration	Recommend configuration
Operating system	Windows 2000 (Windows Vista/Windows 7/8/10)	Windows 7/8/10(32/64bit)
RAM	512M	4GB
Hard disk space	200M	2GB
Processor	Pentium V, Centrino>1.8GHz, Pentium M>1.0GHz	Pentium V, Centrino>1.8GHz, Pentium M>1.5GHz

2.2 Steps of Installation

Close the anti-virus software or put CODESYS into white list before installation, in case some functions can't be realized or deleted during the process.

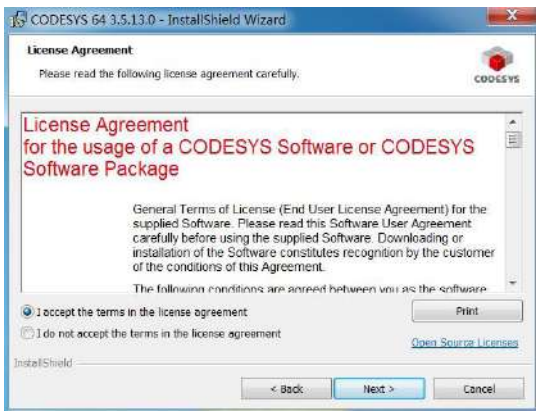
Installation package can be downloaded at official website of HCFA's college, link as below:

http://class.hcfa.cn/mod/folder/view.php?id=709

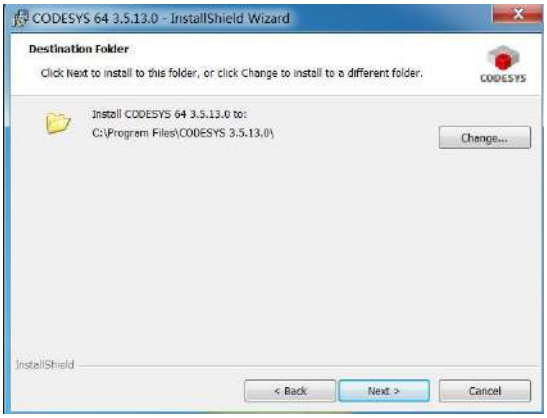
Double click after downloading installation package of CODESYS V3.5 SP14. Before installation, make sure the environment contains Microsoft Visual C++ and Microsoft .NET Framework 4.6(or above version), or the program will install these software automatically and pop up interface of 3S specifications as pic 2.2-2, select Next. Default installation route is C:\Program File\CODESYS 3.5.13.0\" and you can select Change to reset the route. User can choose Complete or Custom Installation in next page, and Complete is recommended for primary users. Click Install for custom installation if needed, shown as pic 2.2-5.

Pic 0-1

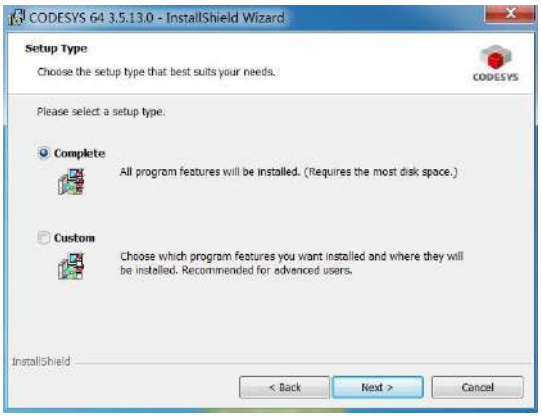
Pic 0-2



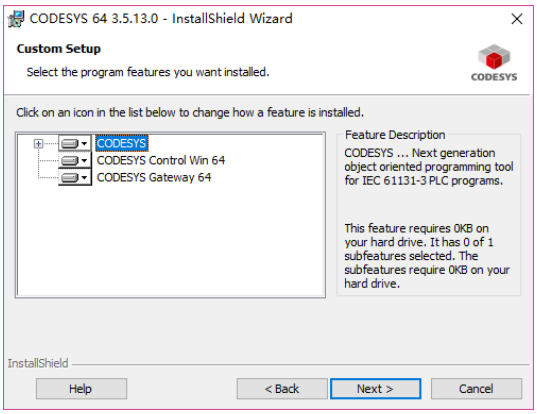
Pic 0-3



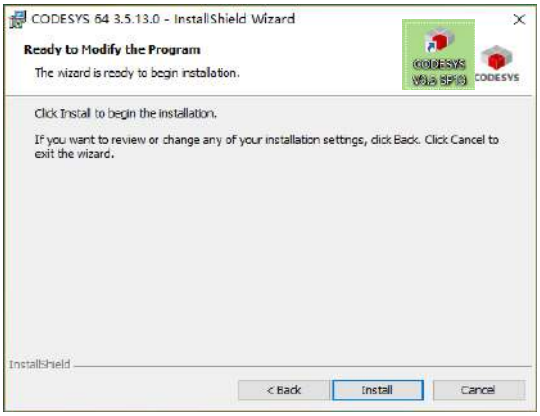
Pic 0-4



Pic 0-5



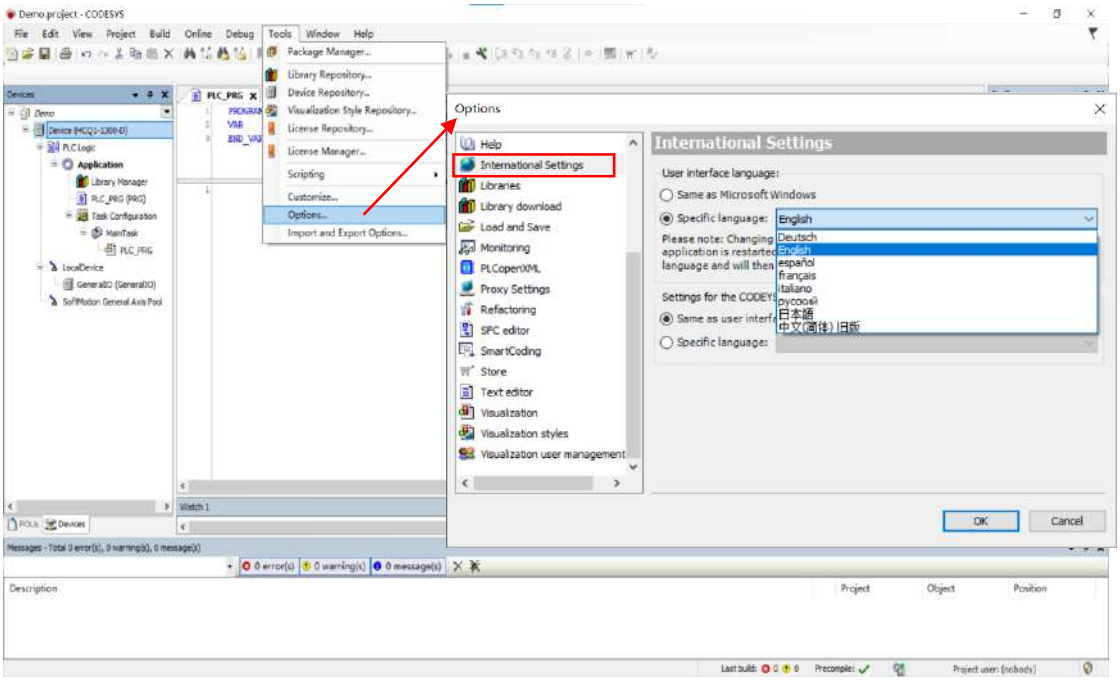
Pic 0-6



Wait for the installation after above steps and then you can find CODESYS on desktop, double click to edit.

After installation, if language of the software interface needs to be changed, menu bar→options→language setting→user language interface, choose needed one among drop-down list, confirm and then restart CODESYS application setting.

Pic 0-7



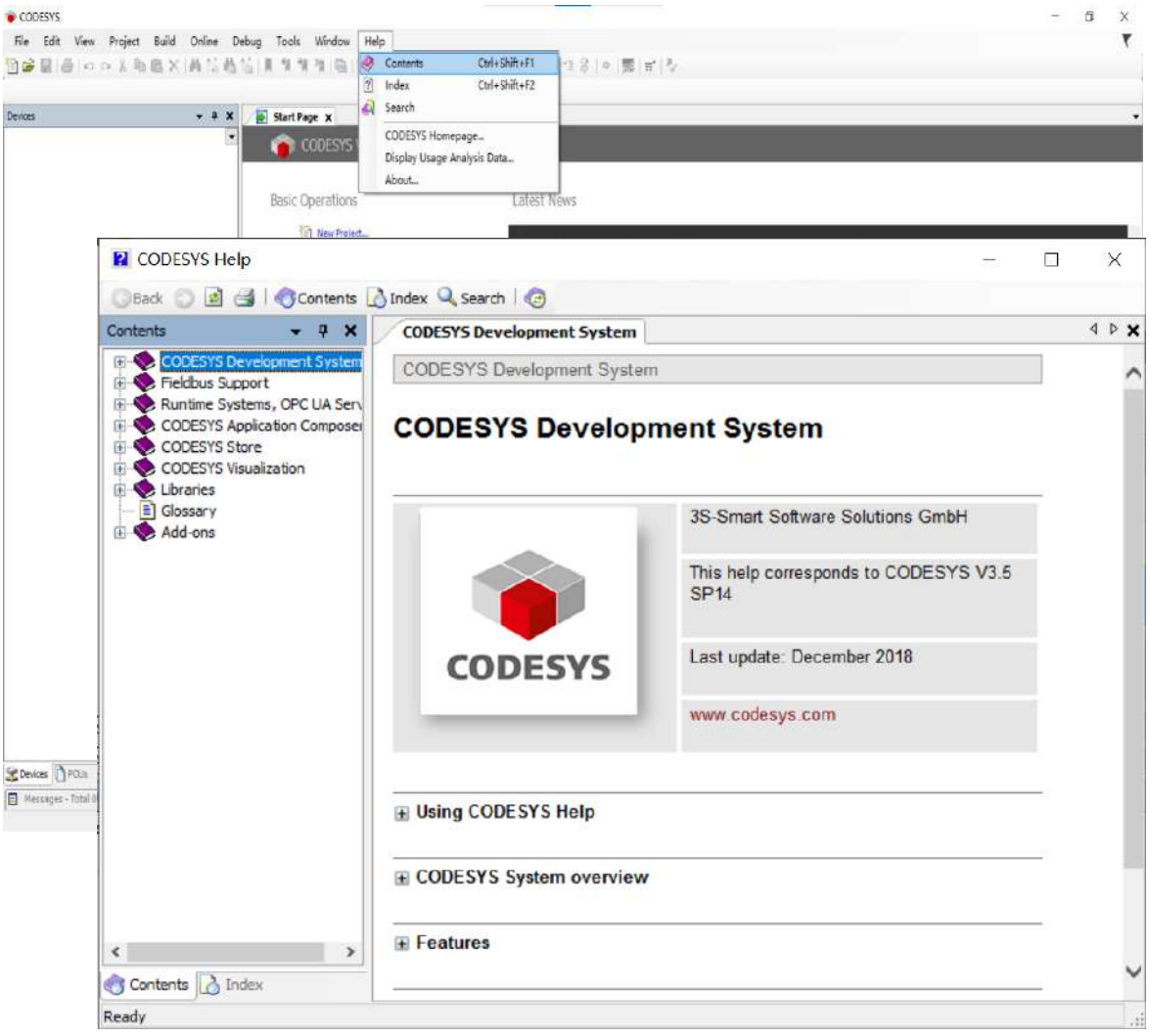
2.3 Version Management

Switch between different versions of software is not supported in CODESYS presently and high version will be compatible with lower ones automatically. Different versions of modules of software can be installed and used as combination. Different versions of program compiler can also be used and new separate function can be added without updating the whole version.

2.4 Help System

Default installation of help system will be proceeded after installation of CODESYS, and user can find Help in menu bar to click Catalogue to get online help. Able to proceed quick search with index or key words.

Pic 0-1

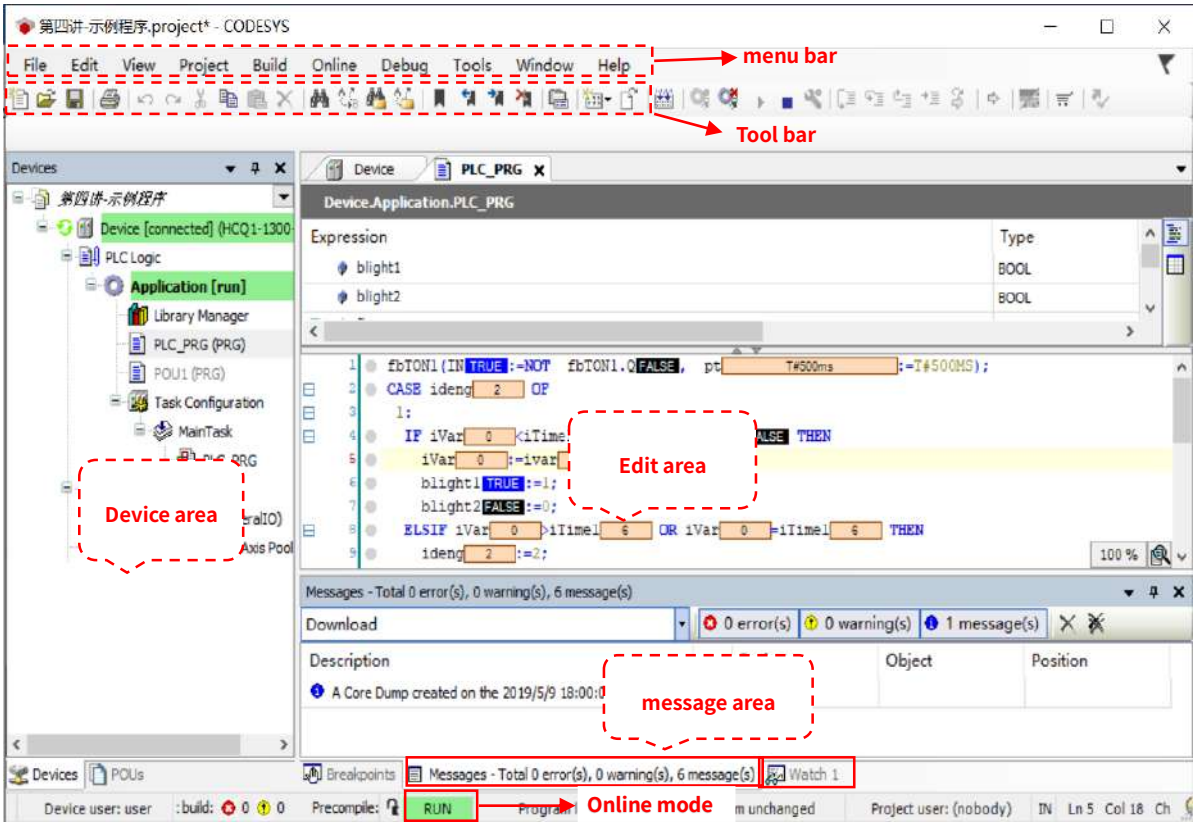


2.5 CODESYS Interface Introduction

Start CODESYS, user interface of CODESYS V3.5 SP14 is consisted of menu bar, tool bar, edit window, device window, monitor window, message window and online mode, etc. Below is detailed introduction.

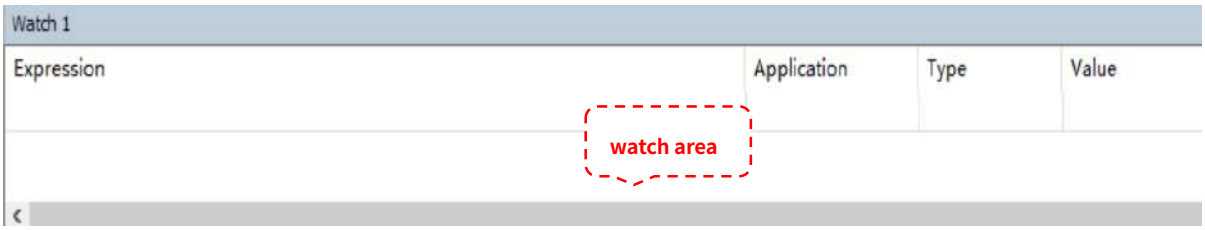
All windows and views in CODESYS interface are not fixed and can be dragged to target place as user like.

Pic 0-1



The TAB under the message window allows you to switch to the monitor window, where you can view any expression in the program.

Pic 0-2



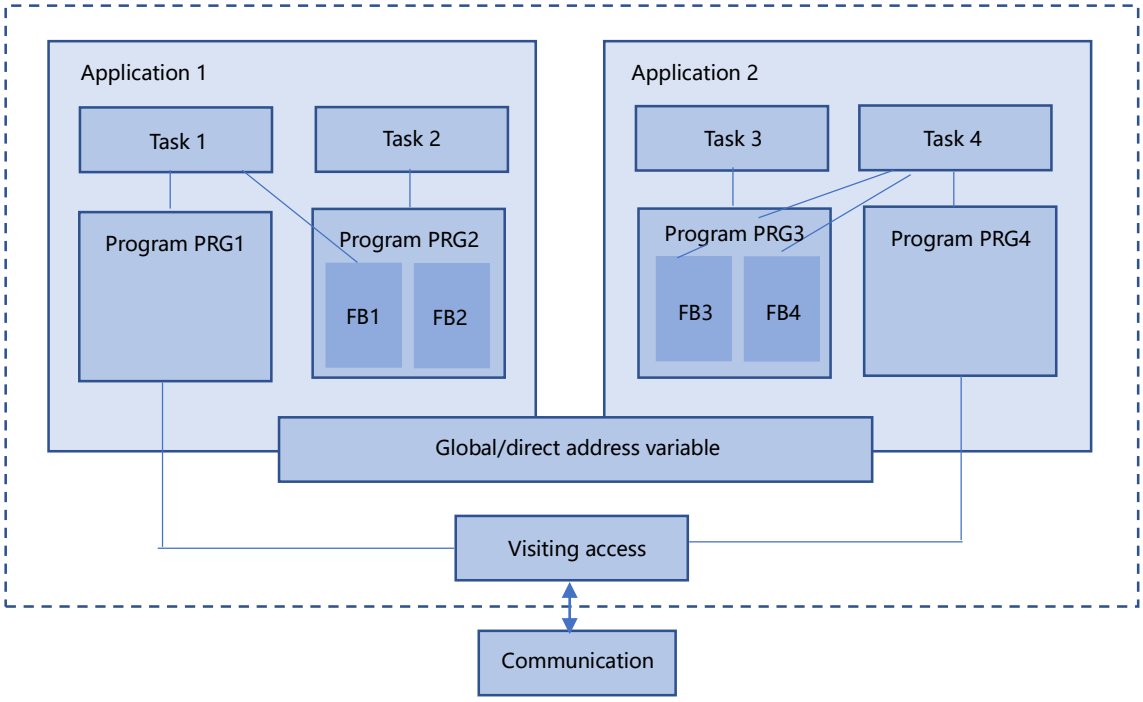
Online mode displays the current running status of the program.

Chapter 3 CODESYS Software Model

3.1 CODESYS Software Architecture

In form of layered architecture, CODESYS software model describes basic function units and interactions including: Device, Application, Task, Global variable, visiting access path and application object. Taking diagram 3.1-1 as example, it describes how a PLC controls multiple stand-alone programs simultaneously and total control is realized.

Diagram 0-1

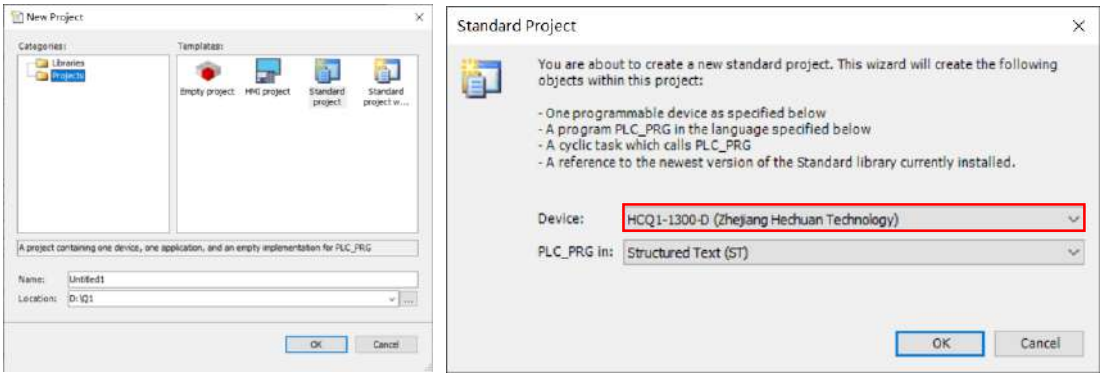


In PLC system, the device provides environment to unite variable applications for data interaction. One or more applications can exist in one device to serve as interface between program and IO. Applications will be distributed in CPU unit of PLC, including global variable, task and program organization unit(POU). Visiting path is for data interaction between different applications, also for communication between variable of each application and other remote devices.

3.2 Device Manager

The system will pop up dialogue of Create New Project when trying to create a new project and select creating an empty or standard project after. Choose the hardware which is actually connected with the Device if standard project is selected.

Diagram 0-1



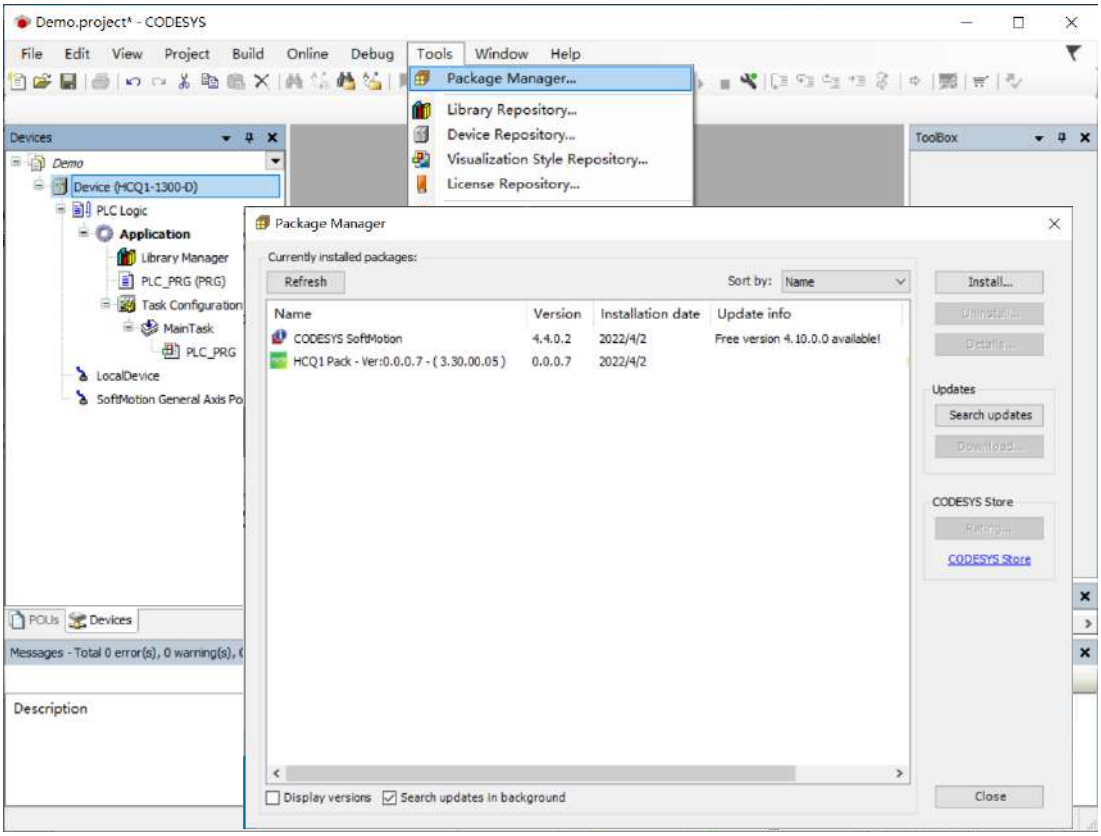
Click OK to finish the creation. Notice: user should install and update devices and software added in CODESYS.

3.2.1 Package Manager

All software packages should be installed as path Tool→Package Manager. And the existing software packages can also be deleted or updated in the manager.

Different configuration files are needed for different hardware devices: code generator, storage manager, PLC function, IO module, library file, device description file, network management driver, ini-files of error code, etc.

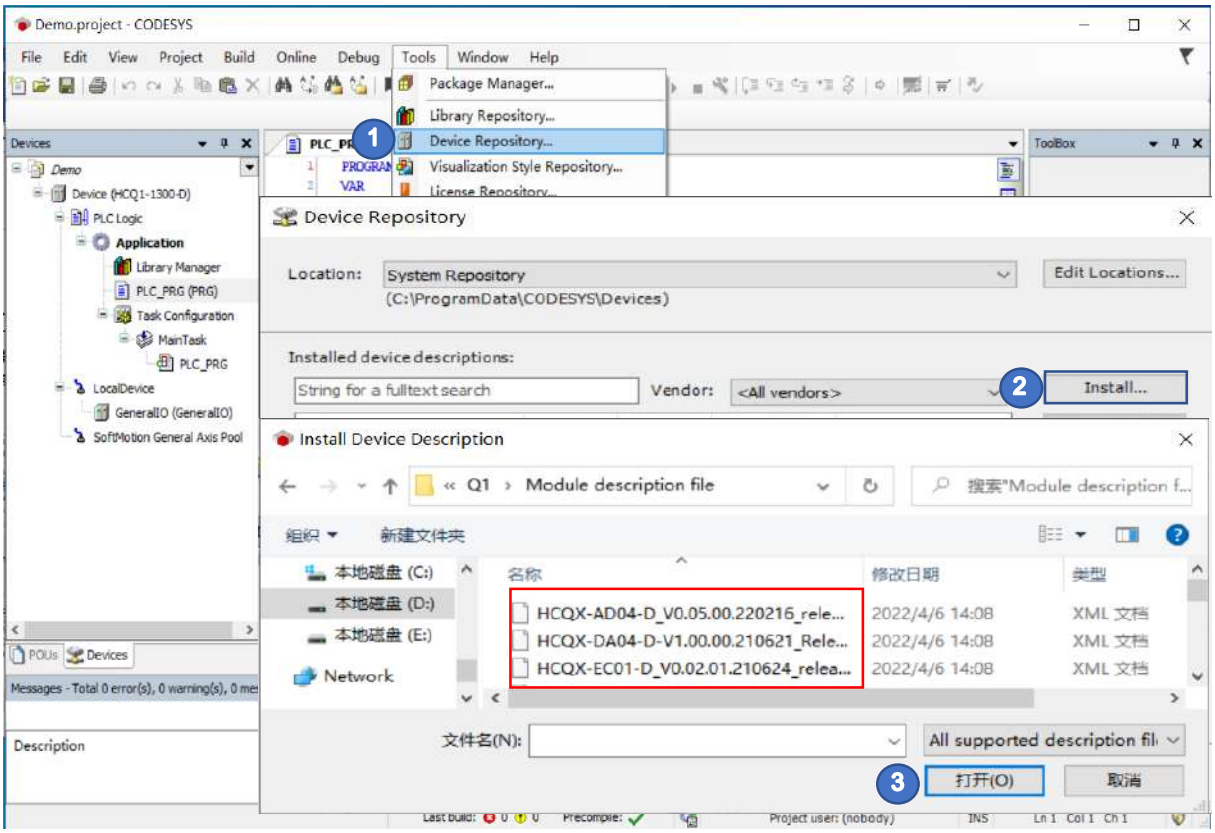
Pic 0-2



3.2.2 Device Repository

Device Library is for installation, uninstalling and checking of device. Sub device can only be found and used after it's added to device. Library. Select Tools→Device Repository and then select Install to import corresponding files. It is applicable for PLC with supplier, SoftMotion controlling device(encoder, driver), site bus, dedicated interface, etc.

Pic 0-3



All imported description files and additional files will be copied in one inner address during installation. Installed device will not be affected if description file is changed and user is suggested to change inner version number of description file of updated device(or to uninstall present device, reinstall and then update description file).

Notice:
Inner device is imported to CODESYS after installation with certain specification and does not support any manual modification or copy. Any add or delete has to be done through dialogue of Device Library.

3.3 PLC Application

CODESYS application objects includes library manager, POU, task configuration, global variable, sample collecting and tracking. Multiple applications can be added in single device.

3.3.1 Library Repository

Library File is for storage of POU which is repeatedly used, which can be existed POU or custom library. Library file is integration of not only function, functional module and program, but also some special structure, enumeration type, etc. Creation and calling of library files with suffix of ".library" and

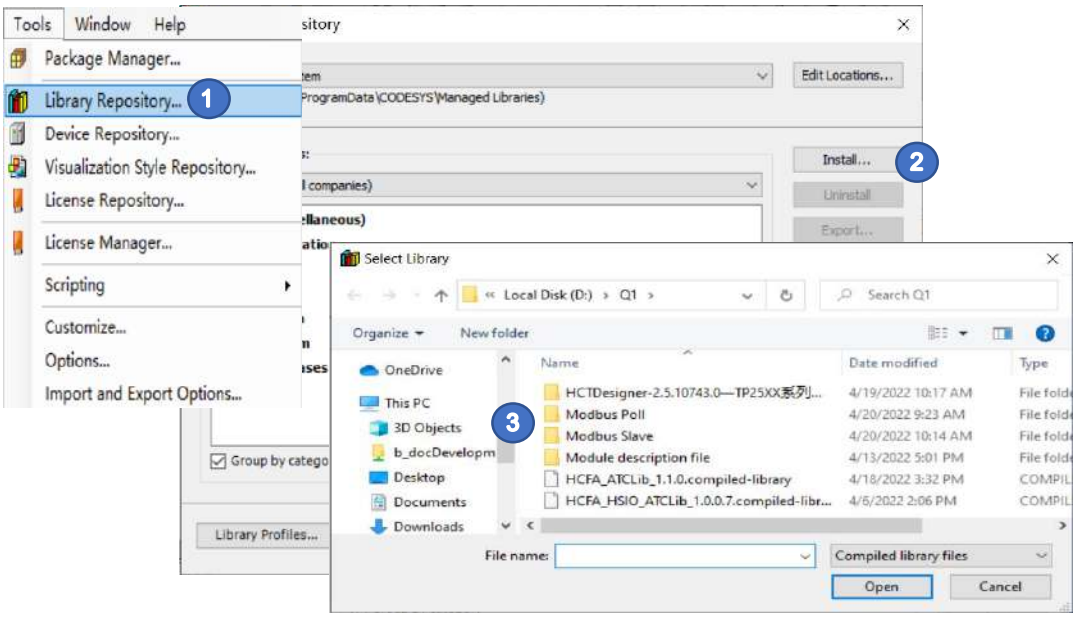
".complied-library". ".library" files can be edited directly as they are standard library files. For compiled ".complied-library" files, they can't be opened and edited directly to check and modify source code, but all POU in library file can be called as normal

Installation of Library File

If external library or custom library is needed, library file has to be installed and called first. CODESYS V3.5 supports "lib" (with standard of CODESYS V2, will be transferred to ".library" file for downward compatibility), "library" and "complied-library".

Select Tools in toll bar and then Library Repository, click and select Install. Click Open for needed library file to install.

Pic 0-1



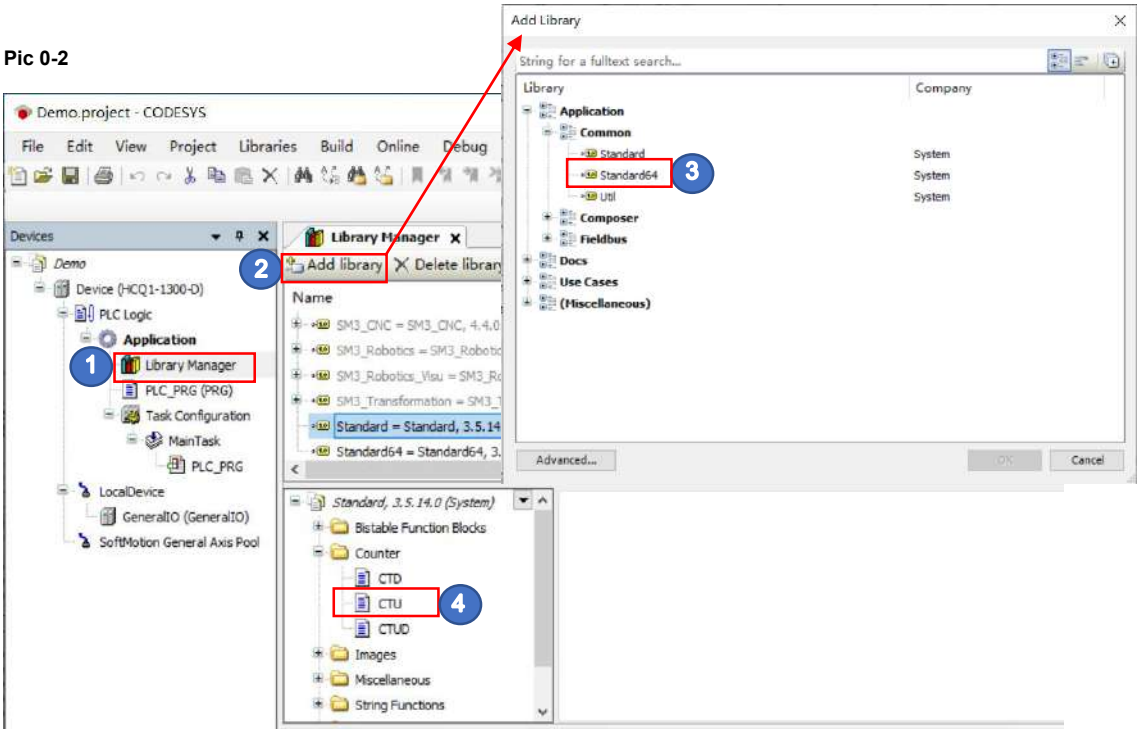
Calling of Library File

Installed library will be added to CODESYS system resource library automatically and user can not add library file to target path manually or edit other files in it. Installation, calling and uninstalling of library file can only be realized through library manager.

Double click Library Manager, click Add Library on left corner of configuration surface. Choose

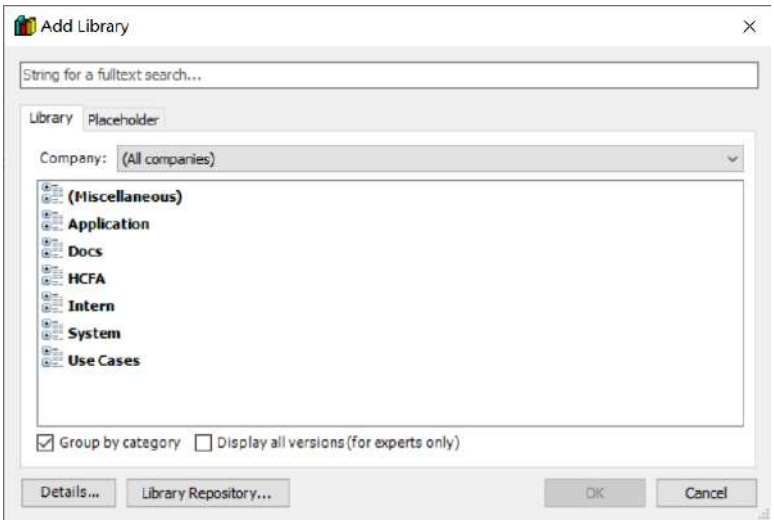
needed library and click “OK” to call. Function, function module and introduction of corresponding library can be checked in library manager.

Pic 0-2



User can also click “ Advance” and select and call library quickly base on supplier name, library file function, version number, placeholder and named function module(which is included in installed library).

Pic 0-3



Creation of Library File

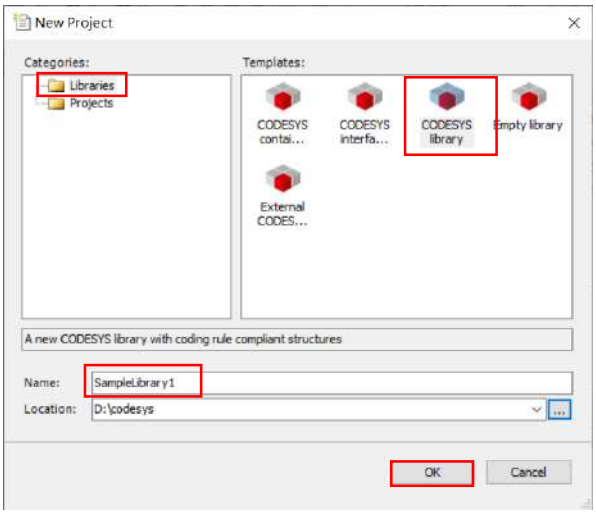
User can seal needed function and function module to create own library file for sharing and application in other projects, steps as below：

- Create empty library file

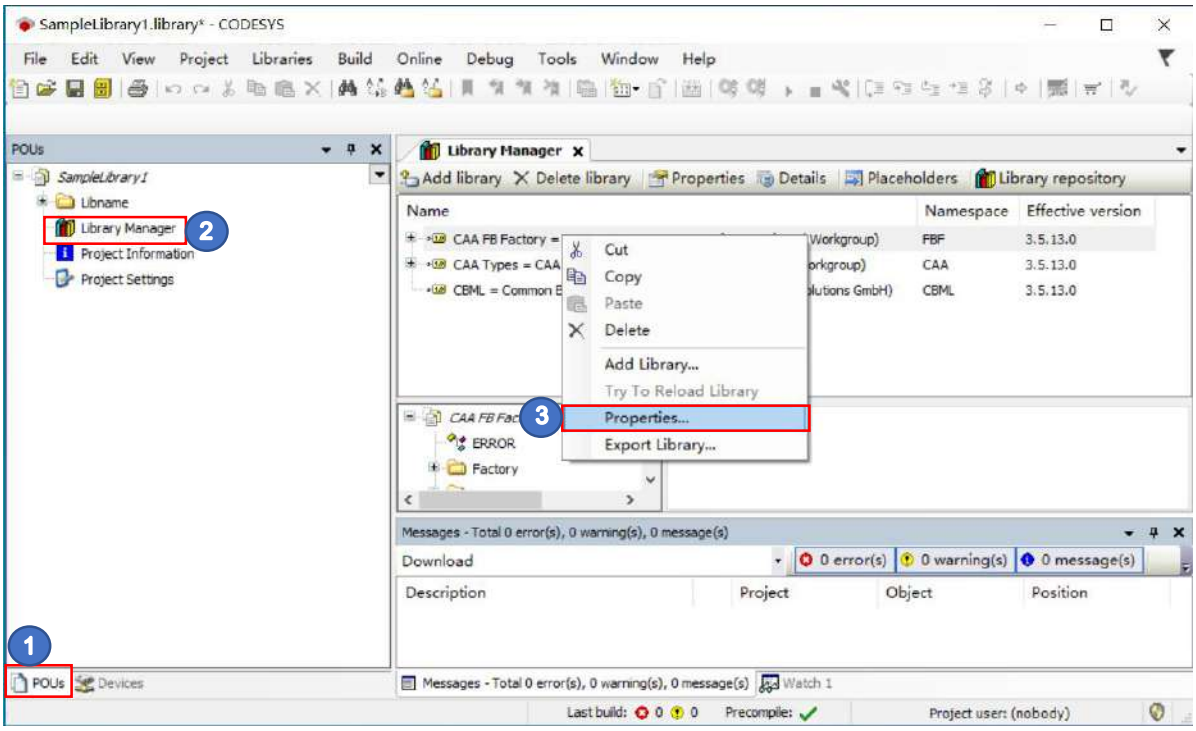
File→New project→Libraries→CODESYS library, enter name of library “SampleLibrary1” in dialogue of Name. Select storing place and click “OK” to generate a new library.

If library is imported when creating a new library project, it can be defined in Properties in each imported library, refer to pic 3.3-6.

Pic 0-4



Pic 0-5

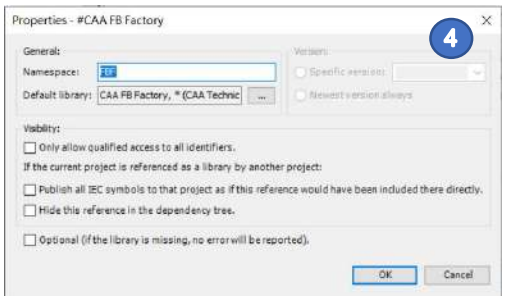


Notice:

① Called library is under Parent library, can be set as show or hide.

② Actions of an imported library including version, name space, visibility, visit properties will depend on settings in dialogue of properties. Then it will act as defined when called by project.

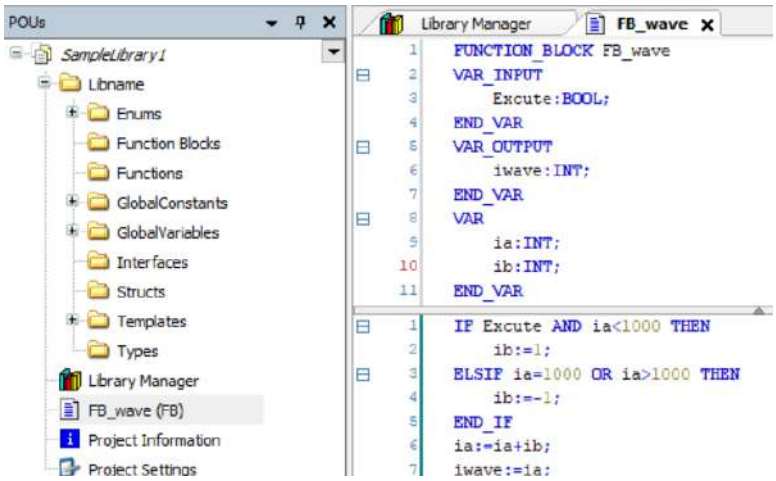
Pic 0-6



- Create POU

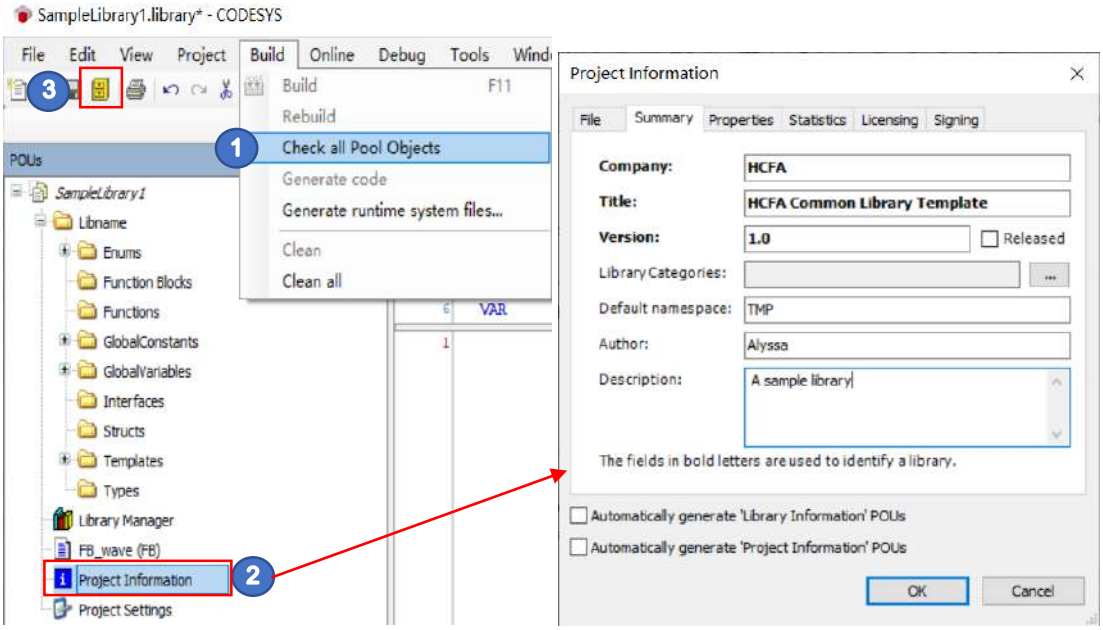
User can create function and function module in library file, or edit enumeration, structure, interface,and global variable. Example: right click “SampleLibrary”→”Add object”→”POU”,select Function module and choose ST as program language.

Pic 0-7



Check and save by Build→Check all Pool Objects, select Project information if no mistake, to edit information like belonging company, title, version number, writer name, introduction, etc. Black and bold part must be filled and finally , save project and load to library to save the library file.

Pic 0-8



After above settings, install the library file through Tools→Library Repository and user can delete it in same route. Pay attention to consistency of version number of CODESYS and uninstall library of old version. Reinstall to finish update of custom library.

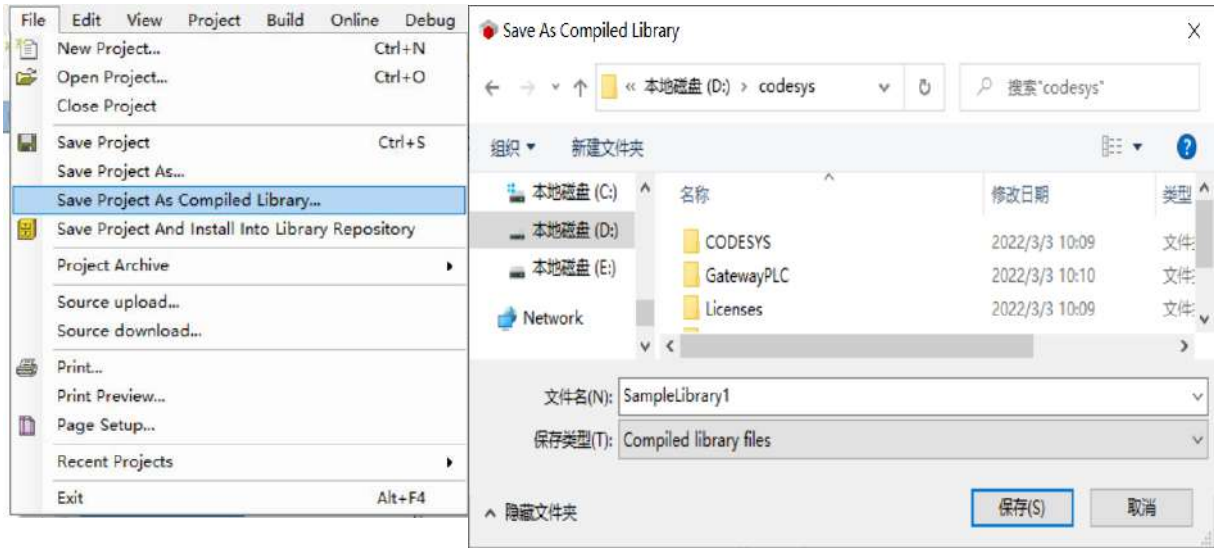
Encryption of Library File

If custom library needs to be shared with closed source code, two ways as below:

- Save library file as compiled version “complied-library”

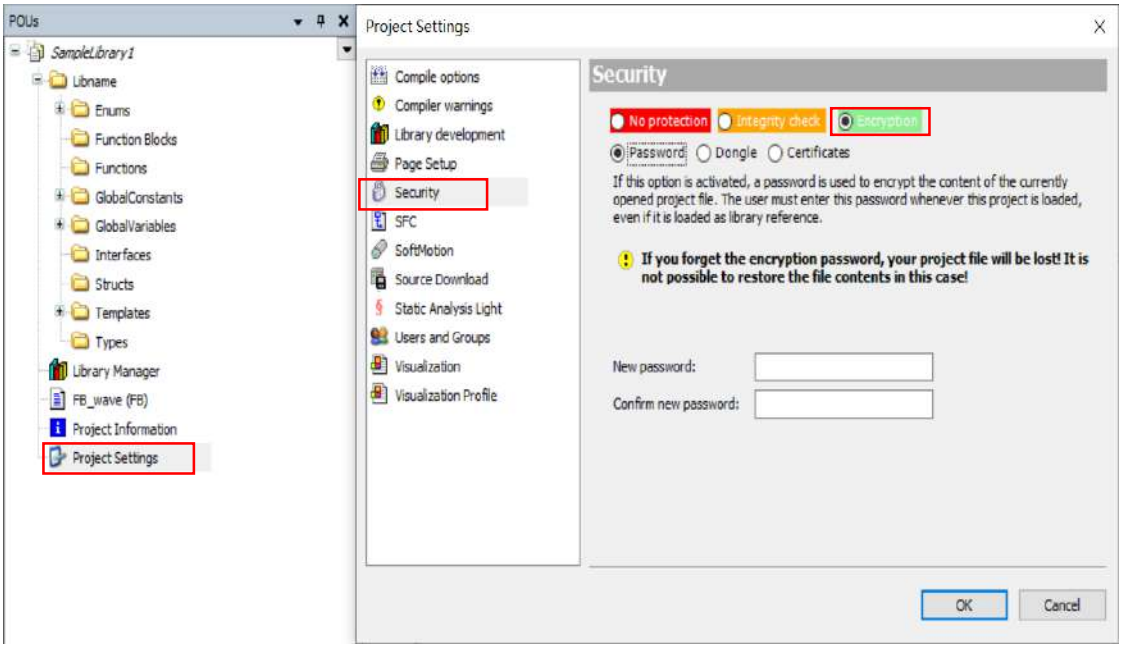
Default format of custom library file is ".library", which can be opened and edited. Select File→Save Project As Compiled Library→Save to save the compiled library file and the the source code will not open to shared users.

Pic 0-9



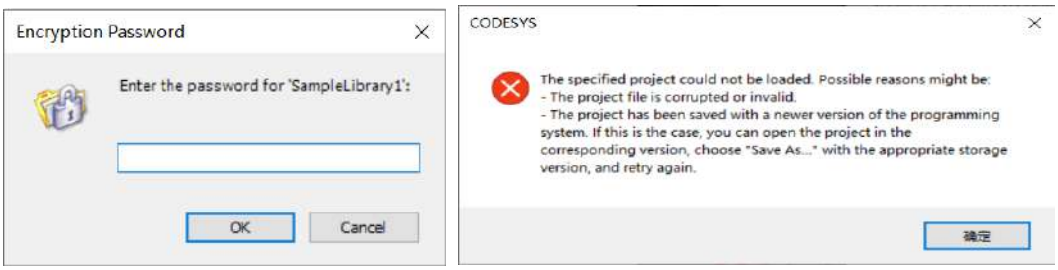
- Item settings→Project settings→Security, then user can choose Password for protection.

Pic 0-10



Save and reinstall the project after password is set and password will be needed each time the library is opened. Source code can't be checked with incorrect password. This method is also applicable to project files in same steps.

Pic 0-11



3.3.2 Task Configuration

Multiple POU can be created within each application and they are executed based on Task. After configuration in Task, POU will execute according to task configuration cycle or certain trigger. Rules as below:

- Maximum cycle task of 100
- Maximum Freewheeling task of 100
- Main program "PLC_PRG" can operate as inertia taxiing task under anytime without manual setting.
- Processing and calling program will execute in top down order according to task configuration.

Before introduction of task window configuration, please understand below PLC execute program process.

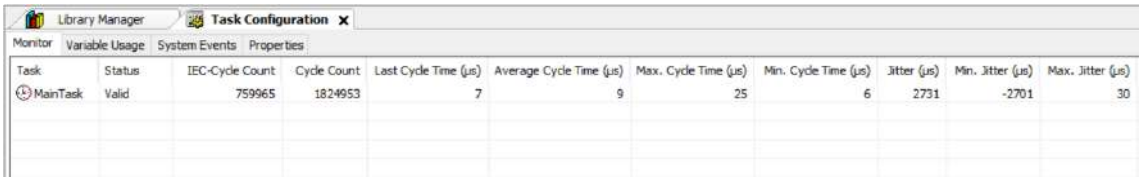
Besides above 3 phases, PLC will also process interior diagnosis, communication, I/O data within one scan cycle. So, the scan cycle length depends on hardware filtering time of input circuit, lag time of output circuit, scan type and user project, etc.

Then it's introduction of Task Configuration options which influence project execute. Find Task Configuration in left tree menu, double click and four options can be found in right configuration interface:

Monitor

After entering online mode, in right monitoring window of Task Configuration, user can monitor present status, cycle times, actual executing time, average/max/mini cycle time and other related parameters.

Pic 0-12



Variable Usage

In Variable Usage interface, user can check use of all variable usage, including variable type, task in which variable is used, number of operations, etc.

Pic 0-13

Library Manager | Task Configuration

Monitor | Variable Usage | System Events | Properties

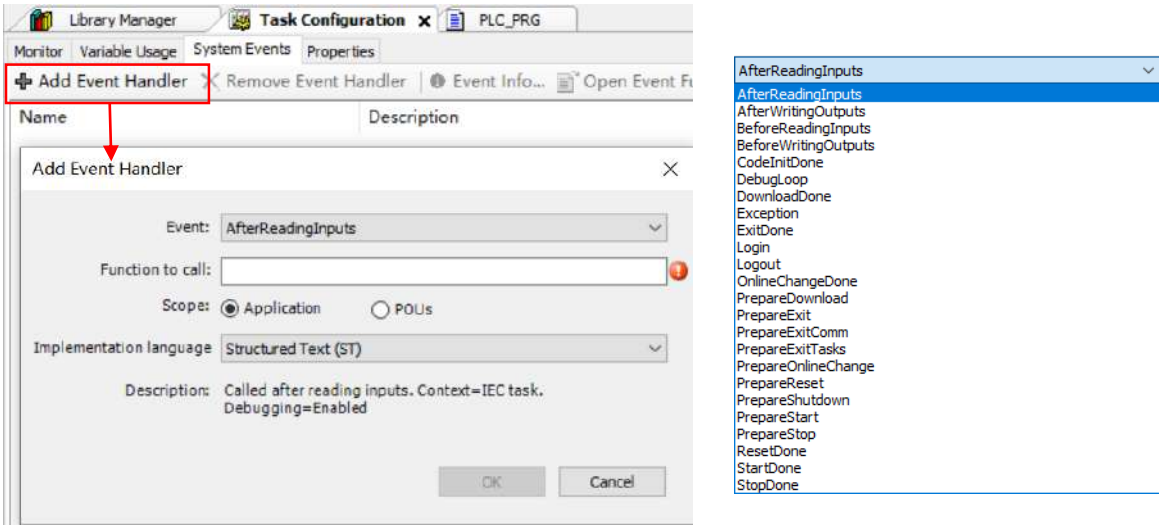
Variables	Type	Count	MainTask
PLC_PRG.fbTON1	TON	1	r
PLC_PRG.ideng	INT	1	rw
PLC_PRG.iVar	INT	1	rw
PLC_PRG.iTime1	INT	1	r
PLC_PRG.blight1	BOOL	1	w
PLC_PRG.blight2	BOOL	1	w
PLC_PRG.iTime2	INT	1	r
PLC_sample.MinInput	INT	1	w
PLC_sample.Input1	INT	1	r
PLC_sample.Input2	INT	1	r
PLC_sample.Input3	INT	1	r
PLC_sample1.FB_PT1Filter_0	FB_PT1Filter	1	r
PLC_sample1.fPeakValue	REAL	1	rw
PLC_sample1.rKp	REAL	1	r
PLC_sample1.tTn	TIME	1	r
PLC_sample1.tPLCCycletime	TIME	1	r
PLC_sample1.rOutput	REAL	1	w
PLC_sample1.FB_Peak_0	FB_Peak	1	r

System Events

Optional system event is decided by actual hardware and will be provided by corresponding library file, which may lead to different system events in different hardware. Processing of system event set in task configuration includes stop, start, login, logout, modification, etc.

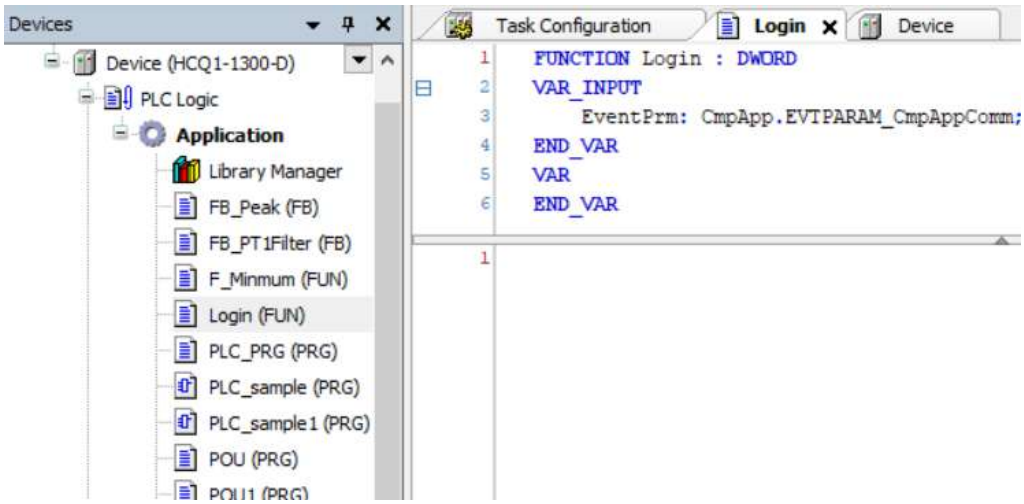
Enter configuration interface through Task configuration→System event→Add Event Handler, select needed event in drop down list.

Pic 0-14



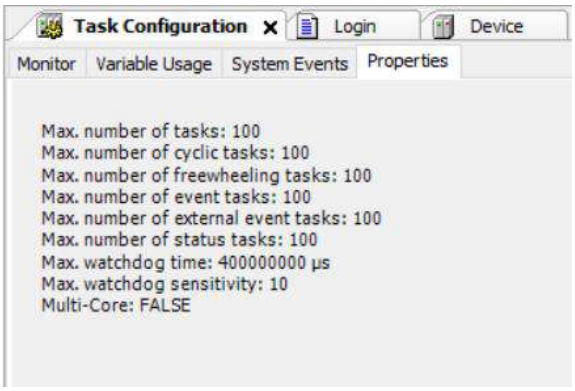
Name the function in “Function to call” after adding and don’t use existed function in POU. Click OK after setting and “Implementation language” is the language to call function.Take Login as example, find system function block under Application and double click to edit with offered language, then Login function will trigger each time the system logs in.

Pic 0-15



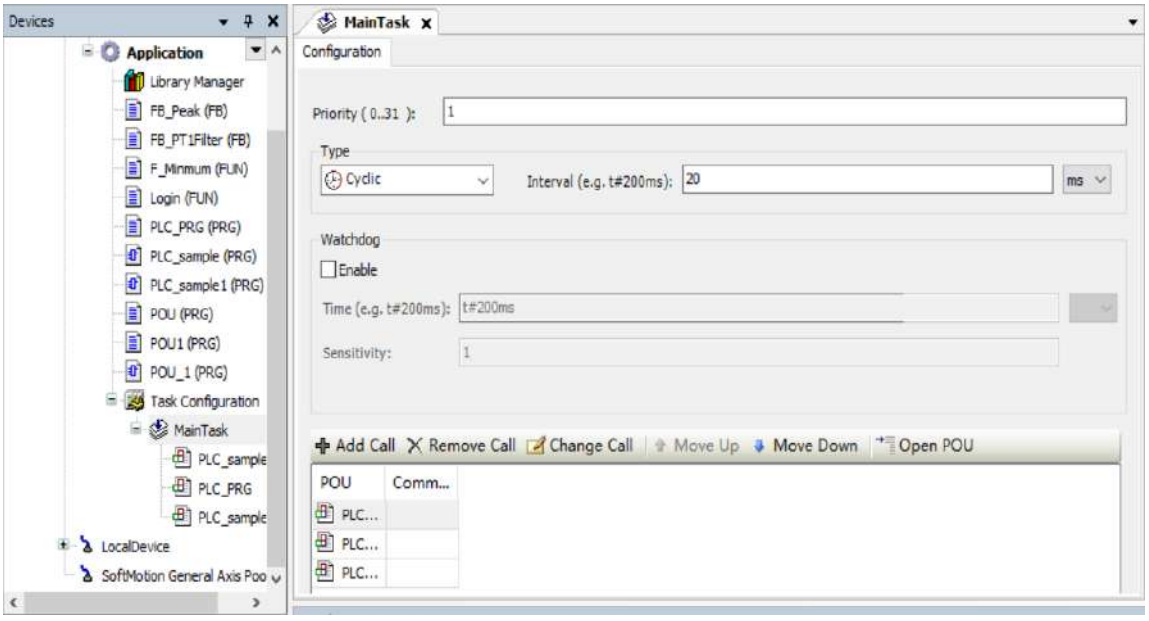
Properties

Basic properties show in tab.



Task configuration also includes execution priority, task activate type, watchdog setting, calling. Below is detailed instruction:

Pic 0-16



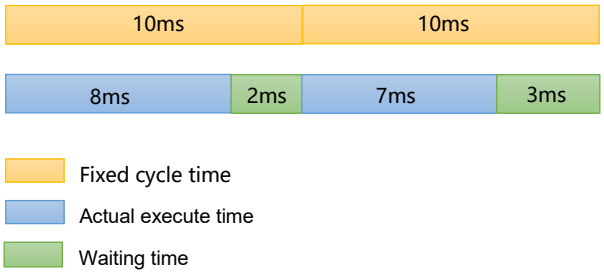
Task Activate Type

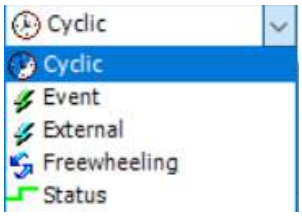
Four activate types are offered in CODESYES as default setting and user can add “External” options after adding description file of Q1 from HCFA. All activate types are optional in application configuration.

1. Cyclic

Cycle execution of projects under this mode will change its time according to program qty and execute orders.

Pic 0-17





If actual execute time is shorter than fixed cycle time, rest will be waiting time, not executed immediately the next cycle. If then lower priority tasks that are not being executed, the wait time is used to execute them.

2. Freewheeling

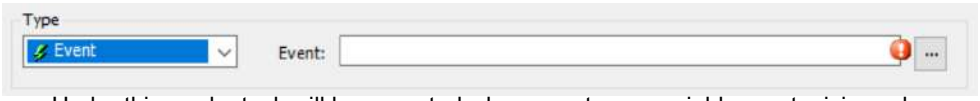
Under this mode, task will be called and executed when project starts and will enter next cycle after each cycle, not affected by scan cycle.

Pic 0-18

No fixed task time in the mode and could be different each time. So it’s not frequently used in actual application.

3. Event

Pic 0-19

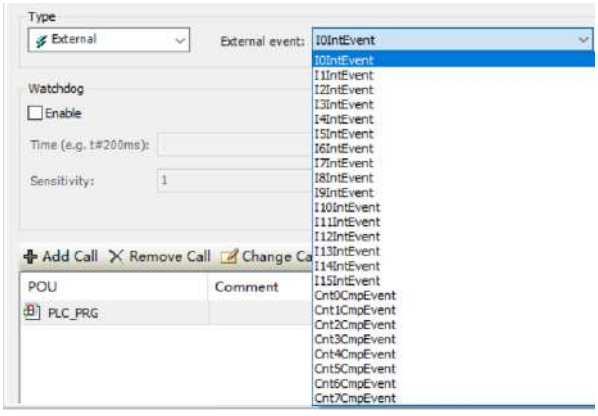


Under this mode, task will be executed when event area variable meets rising edge.

4. External

Only can be used after description file of Q1 is loaded, with two types as below：

Pic 0-20



5. Status

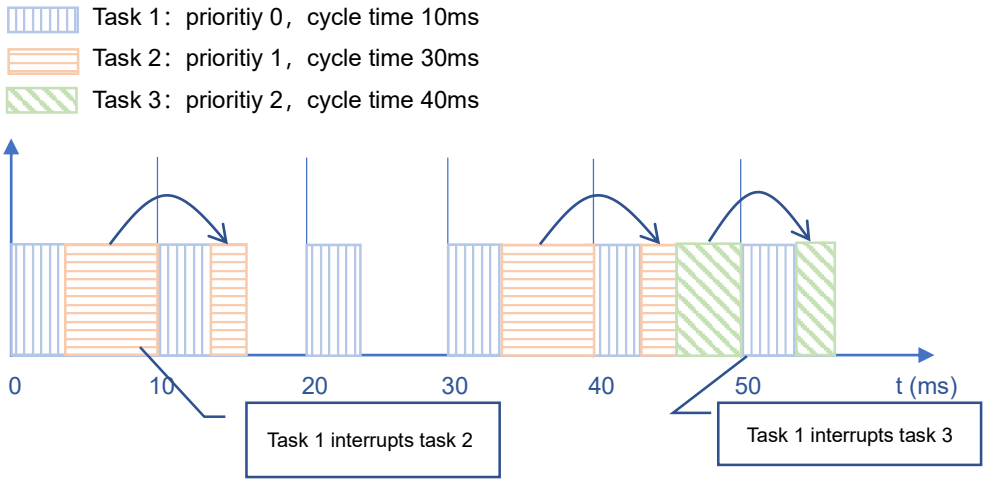
This mode is like event mode, but its trigger is TRUE status of event area variable.

Priority

CODESYS has 32 priorities (0~31, smaller number with higher level) for tasks. Users can not distribute tasks with same priorities in on project. Normally tasks of motion control program should be distributed with higher prioritiy than other tasks in the same interface.

Taking Cycle task as example, if there are 3 different tasks with 3 priorities:

Pic 0-21

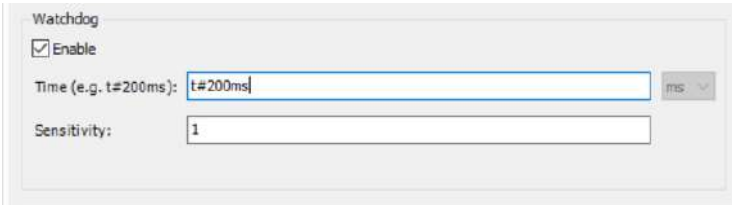


Watchdog

It's hardware type timing device of controller and is closed in default setting, which is used for monitoring abnormality of inner clock during project execution. It will trigger when crash or endless loop occurs, and send signal to reset or stop running program.

Time and sensitivity parameters are needed for watchdog configuration and each task can have separate watch dog. Default time unit is ms; default sensitivity value is 1 and it's used as permitted exceptional value.

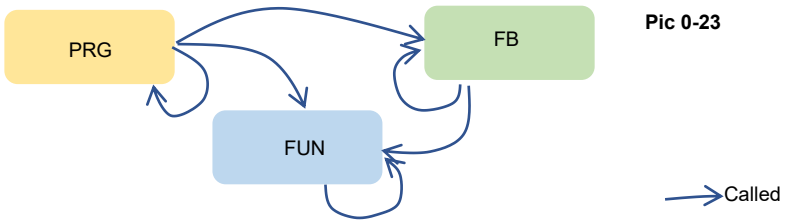
Pic 0-22



So, actual trigger time of watchdog=time*sensitivity. Watchdog will stop present project if its executing time is longer than watchdog setting time. It's normally used in situations which have high requirements on real time and safety, to prevent PLC from crash or endless loop.

3.3.3 PLC Programming

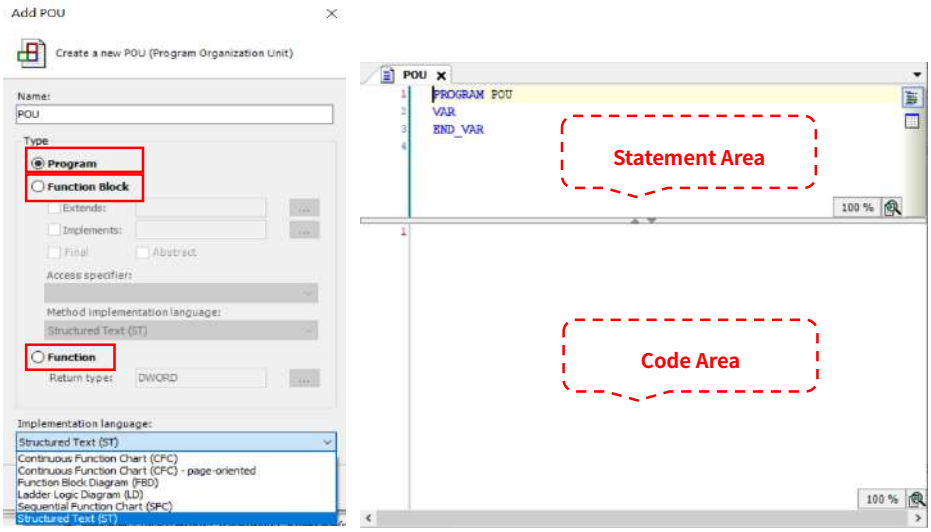
PLC programming interface consists of statement area and code area. POU can be divided into FUN, FB, PRG according to function. For FB and PRG, static variable is used and can be monitored after login and called in next cycle. For FUN, its return value is single and temporary variable is used. Break point is needed, or variable value can't be checked in login status. Calling relation among these three is as below:



Pic 0-23

User can find “Application” in tree menu, click “Add object” → “POU”, Add PRG/FB/FUN, Implementation language, and then choose corresponding programming language.

Pic 0-24



All variable value should be defined in statement area when setting its name, type and initial value: input variable, output variable, input/output variable, local variable and constant. Format of statement is based on IEC61131-3 as below:

- First character of variable can be letter or underline, but not number.
- Name of variable is case insensitive.
- Constant underlines are not permitted in variable name.
- Space and special characters are not permitted in variable name.
- Keywords, FUN and FB names are not permitted in variable name.
- Keywords will appear in blue capitals automatically.
- Single-line comment can express with “//”. Choose “(**)” if comment is needed to be among statement sentence(not within character sting) and Chinese comment is supported.

  Details as below:

  {(*<comment>*)}

  <variable name> {AT<Address>}: <data type>{: =<initial value>}; {//<comment>}

  { } optional part

  Variable with fixed address is in distinguished storage area through I, Q, M. Define variable type with X, B, W, D.

  Define a digital input according above statement:

For users not familiar with variable statement format, Shift+F2 or right click Auto Statement to avoid mistakes. Red dialogue must be filled and other parts according to actual need：

Pic 0-25

Scope：variable scope, local variable or interface variable

Name：refer to IEC61131-3

Type：variable data type

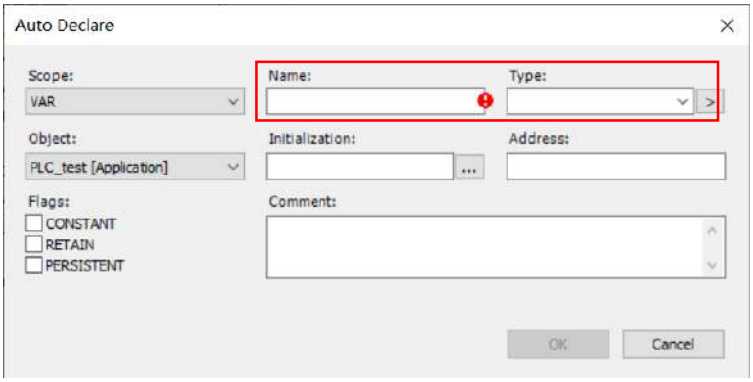
Object：application

Initialization：initial value

Address：external address

Signal：define variable as constant, holding or persistent

Comment：variable comment



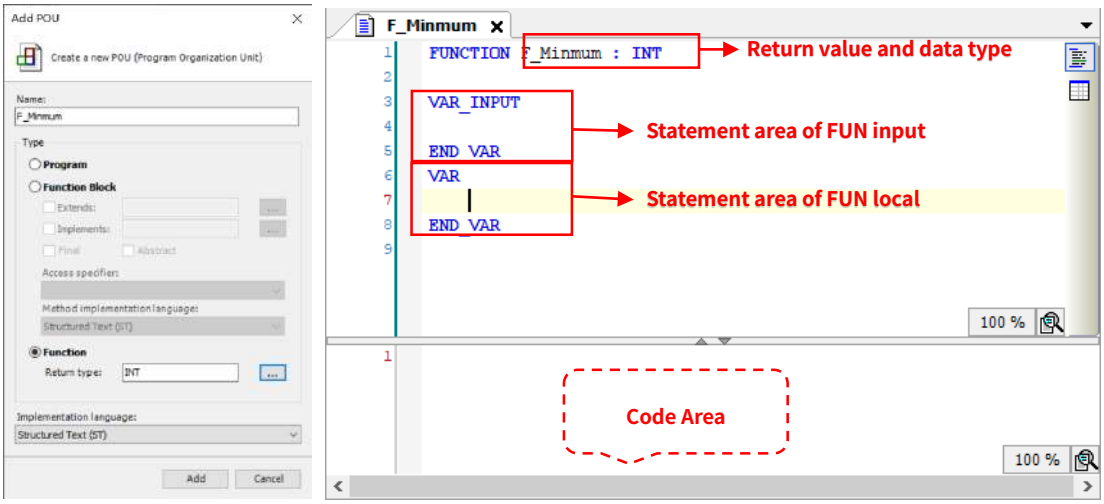
User can select needed programming language according to personal favour in code area. Below will be detailed instruction of three types of POU offered above.

FUN

FUN is basic algorithm unit, which has at least one input variable, no static variable, only one return value. It can be called by FUN, FB and PRG.

Inner logic of FUN can be selected from IEC61131-3 and name of FUN is the return value and FUN output:

Pic 0-26



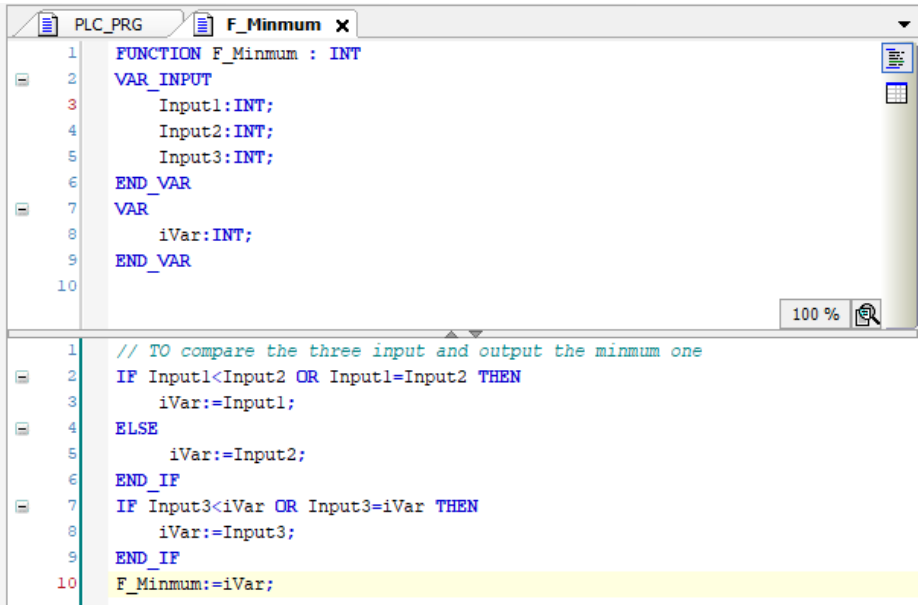
Notice：

- FUN could have multi input variable but only one return value with no type restriction.
- Unlike FB, FUN has no appointed memory allocation. User can appoint in input interface to get the only return value.
- As inner variable can't store values, user need to check values with help of break point.
- Var_INPUT of FUN can be empty, constant, variable or call FUN.

【Example 1】: Create a FUN to output minimum value of three integer variables, according to above.

FUN statement:

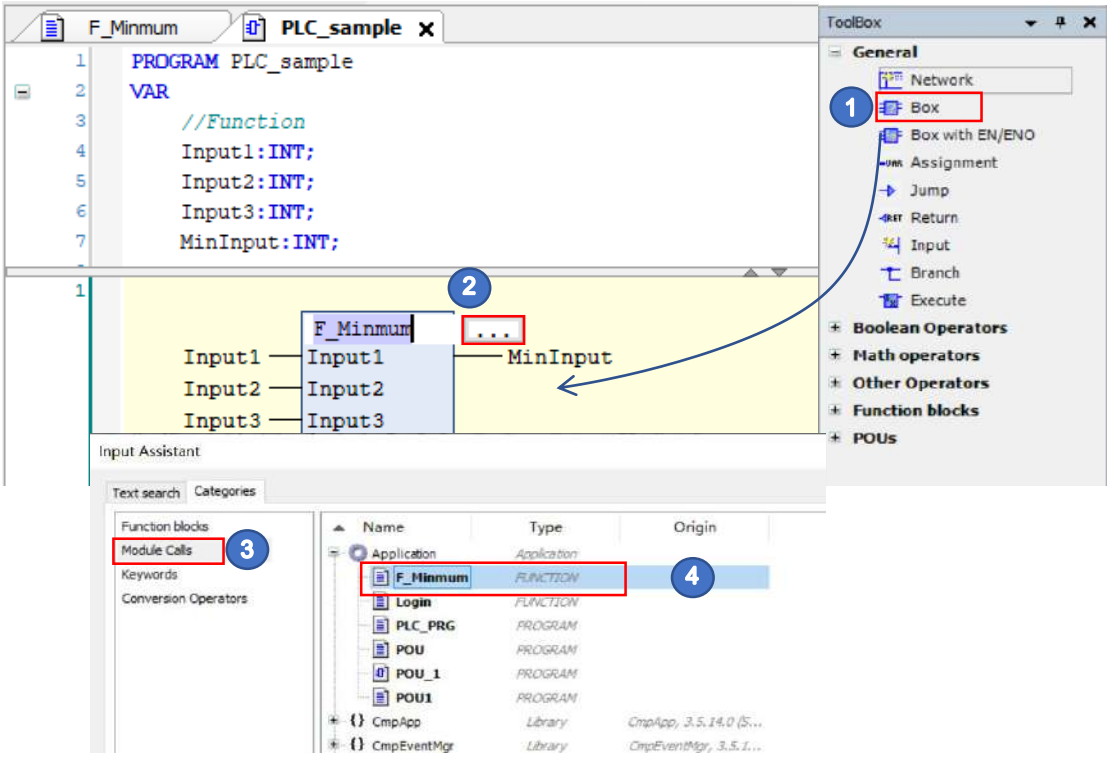
Pic 0-27



Calling FUN in program:

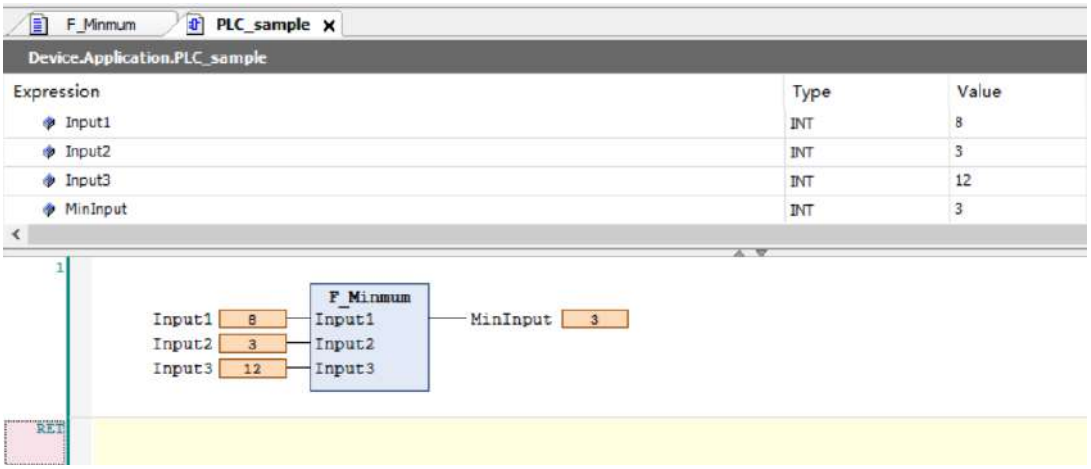
Calling FUN needs not be instantiated and choose FBD as programming language for convenient check of FUN interface, ToolBox→General→Box, drag to edit area and fill according to I/O interface.

Pic 0-28



Online operation result of FUN:

Pic 0-29

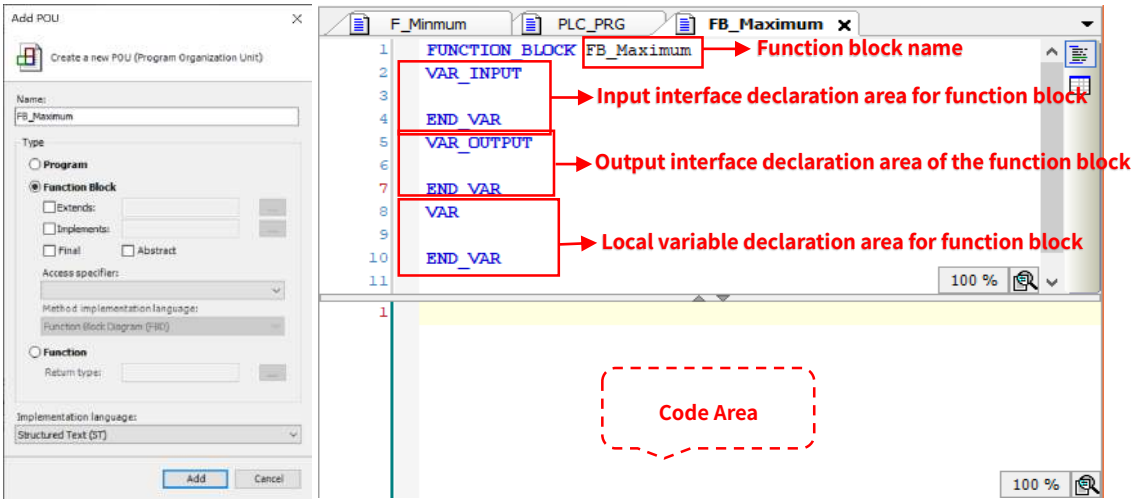


FB

FB is to seal repeatedly used program for convenient calling.

Select any inner logic language of IEC61131-3 for FB and custom block interface as below:

Pic 0-30



Notice:

- Address variable with fixed address(like %IX1.1) should not be partial variable of FB, so it will not reply on hardware and assignment can be used for calling.
- Instantiate FB before calling, which is different from FUN. Different defined instantiations are suggested one FB is called repeatedly in one project.
- FB supports adding Method, Extend, Implement and other programming method

Difference between expression of FUN and FB:

Pic 0-31

	FUN	FB
Memory allocation	Not appointed	All allocated
I/O variable	Only one output variable	No limitation
Calling relation	Can call FUN, not FB	Can call both FUN and FB

【Example 2】: According to above instruction, define a FB to realize PT1 first order low pass filtering algorithm. Make output flat through adjusting rk and tT parameters. Among which, filtering coefficient α is sampling collecting cycle/（filtering time+sample collecting cycle）（filtering coefficient 0<α<1）. User can set low pass filtering gain factor and time constant according to actual need and laws of this algorithm are as below:

- Bigger time constant, smaller filtering coefficient, more stable filter but low sensitivity.
- Smaller time constant, bigger filter coefficient, higher sensitivity but more unstable filter.

FB statement:

Pic 0-32

FB_PT1Filter

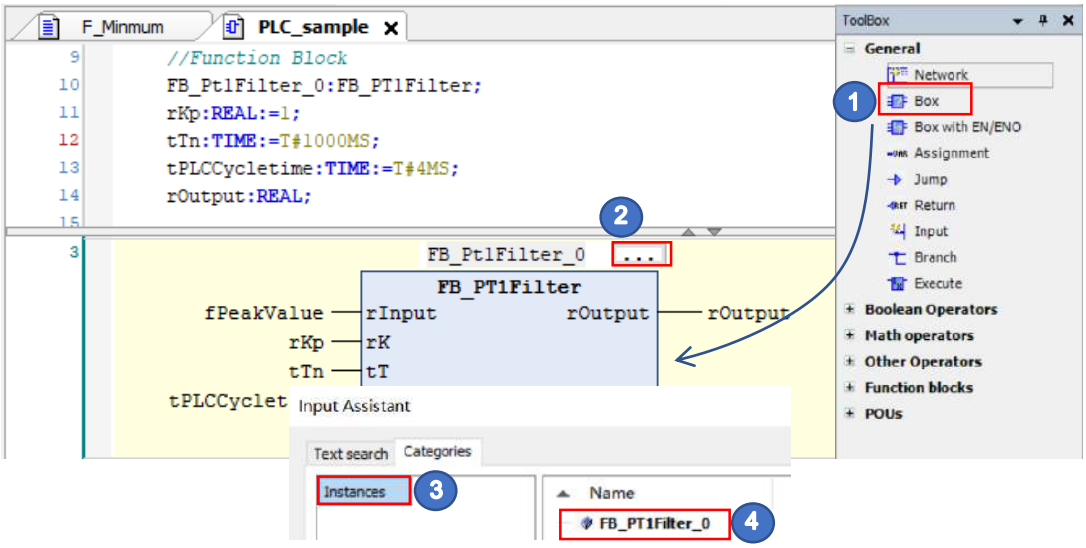
```
FUNCTION_BLOCK FB_PT1Filter
VAR_INPUT
    rInput:REAL:=0.0;  //Sampleing value
    rK:REAL:=1.0;      //Gain
    tT:TIME;          //Time constant
    tPLCCycletime:TIME;  //PLC cycle time
END_VAR
VAR_OUTPUT
    rOutput:REAL:=0.0;      //Output Data
END_VAR
VAR
    _rT:REAL;
    _rPreOutput:REAL;      //The data is output for the previous cycle
END_VAR
```

```
//Simple first-order hysteresis filtering function
IF _rT=0.0 THEN
    rOutput:=rInput;   //Output data initialization
END_IF
_rT:=(TIME_TO_REAL(tT)+TIME_TO_REAL(tPLCCycletime))/TIME_TO_REAL(tPLCCycletime);
IF ABS(rOutput-rInput)<1E-6 THEN
    rOutput:=rInput;
ELSIF _rT>0.0 THEN
    rOutput:=(rK/_rT)*rInput+(1.0-(1.0/_rT))*_rPreOutput;
ELSE
    rOutput:=rInput;
END_IF
_rPreOutput:=rOutput;
```

Calling FB in program:

Choose FBD as programming language and call instantiated FB. ToolBox→General→Box, drag to edit area and fill according to I/O interface.

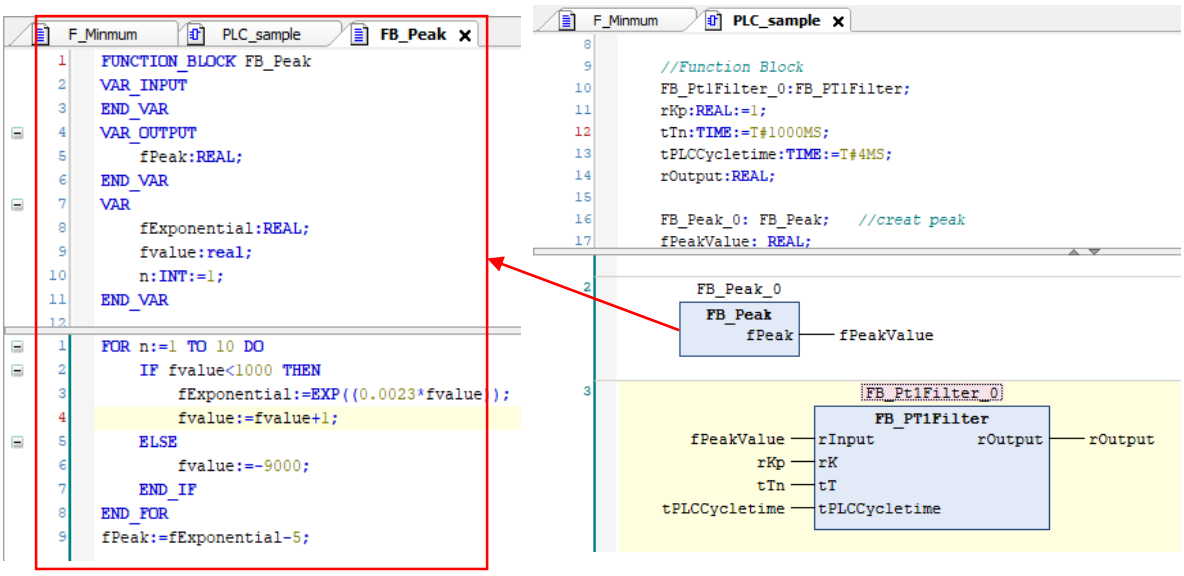
Pic 0-33



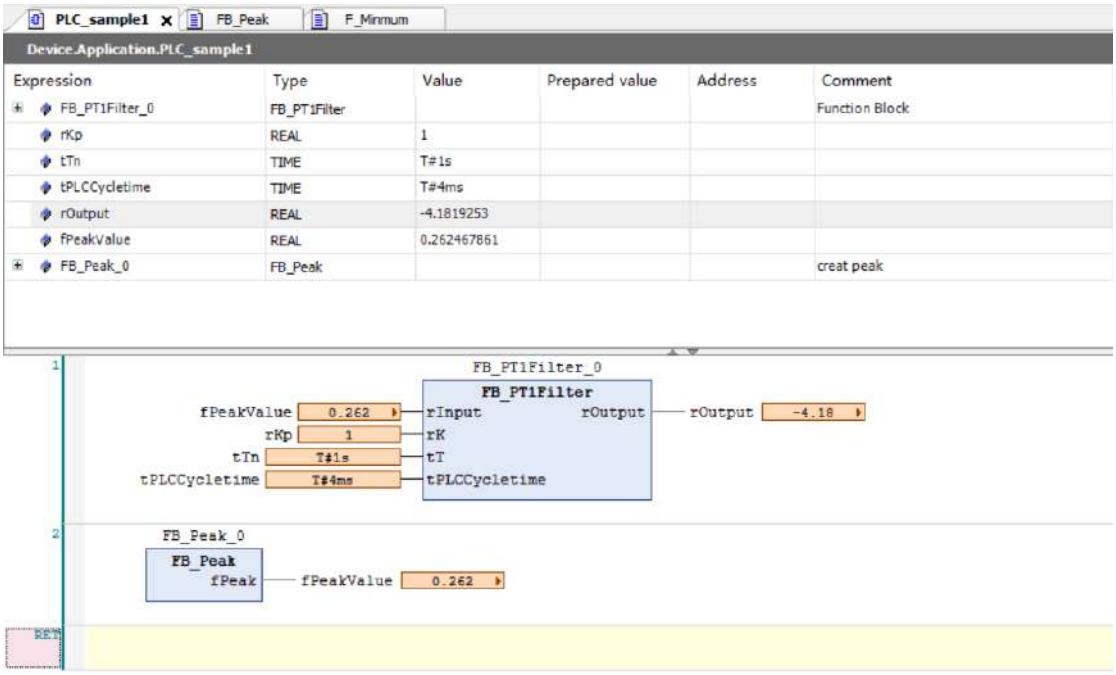
Operation result of FB:

To better visualize the actual filter effect of first-order low-pass filtering, add a unilateral exponential pulse waveform

Pic 0-34

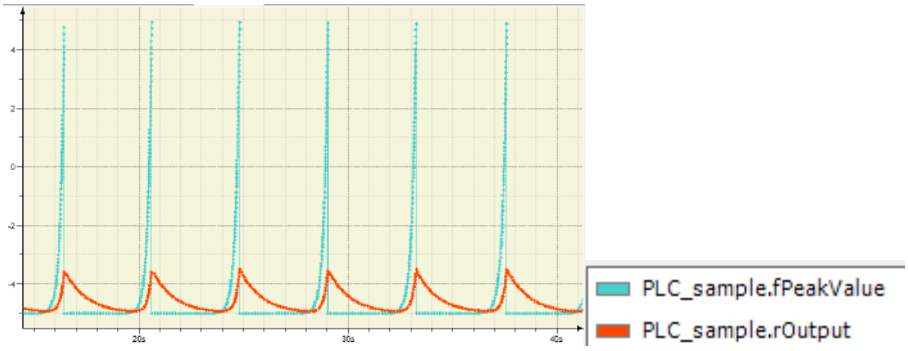


Pic 0-35



rOutput output waveform:

Pic 0-36



Program

As core of a task, partial variable, global variable, external variable(hardware mapping address) can be defined in program. A program may contains configuration of address and allow direct express variable of PLC physical address. Address configuration can only be used in statement of inner variable and can be filled as below format:

bVar AT%IX0.0: INT;

Assign direct express variable as below in program edit area:

%Q0.0: =TRUE;

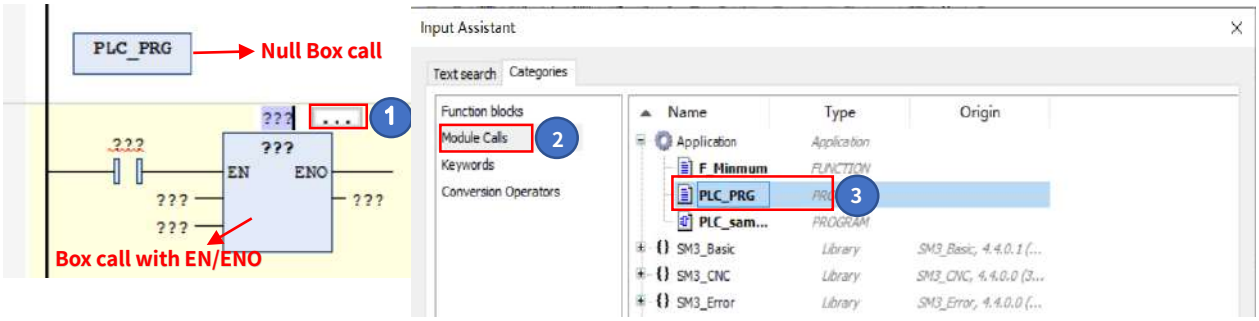
Program organization unit can't call itself but can call other programs. Only call program will be executed.

Calling program will be format as below if ST is selected:

PRGsample ();

Calling program will be format as below if LD is selected:

Pic 0-37



【Example 3】: Write a program to control level. Discharge water when level is 500 higher than alarm and charge water when lower than 100.

Pic 0-38

```
PROGRAM PLC_PRG
VAR
    rLevel: REAL:=350;
    PortAopen: BOOL;
    rLiquid_Speed: REAL:=10;
    PortBopen: BOOL;
END_VAR
```

```
PLC_sample();
IF rLevel>500 THEN
    PortAopen:=TRUE;
    rLevel:=rLevel-0.5*rLiquid_Speed;
ELSIF rLevel<100 THEN
    PortBopen:=TRUE;
    rLevel:=rLevel+0.5*rLiquid_Speed;
END_IF
```

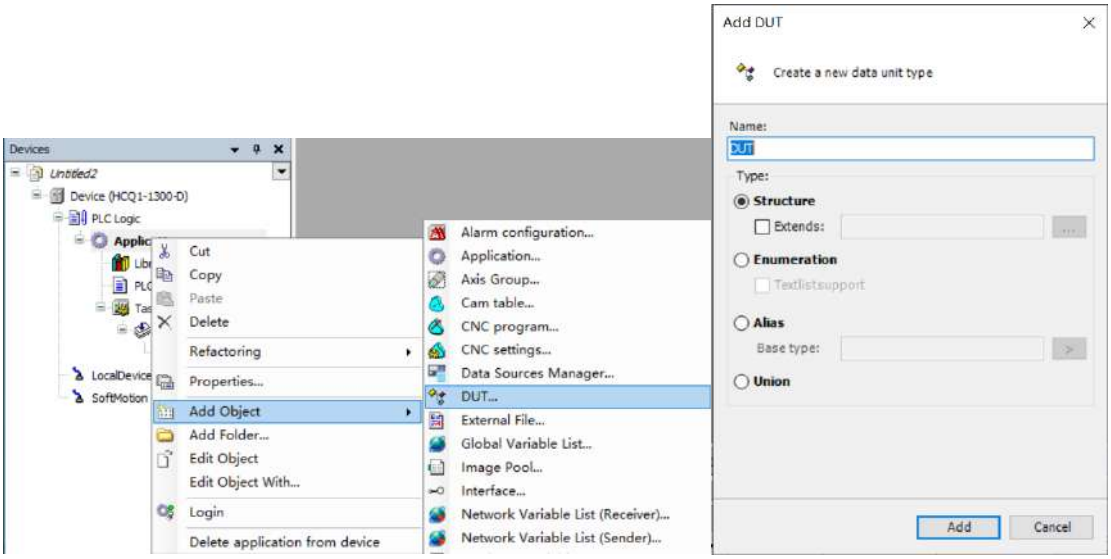
In Structured Text, ();needs to be added after the program object

3.3.4 Data Unit Type

It is also call custom DUT, including Enumeration, structure, alias, union.

Single right click “Application” and select “Add object”→ “DUT”, select and name needed DUT, click Open to create.

Pic 0-39



Like FB, Structure of DUT also supports Extends, which means user can extend another Structure through one existed and defined Structure.

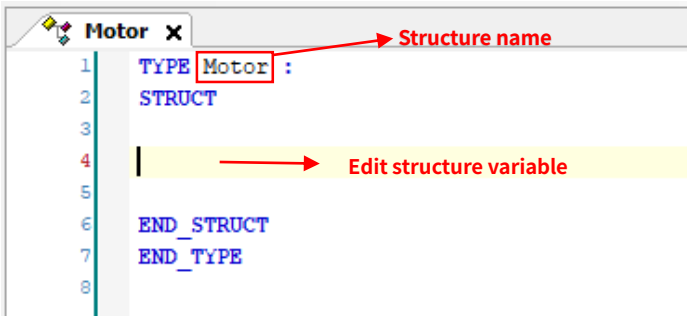
Introduction of three types：

Structure

Consisted of a series data with same or different types, it is a kind of custom defined data type. Integration of different types of data is sometimes imported, like information of a motor including model, factory, rated voltage, rated current and with/without brake. Inner relations between motor and these variables in form of separate variable, which is reason of taking Structure.

Statement as below：

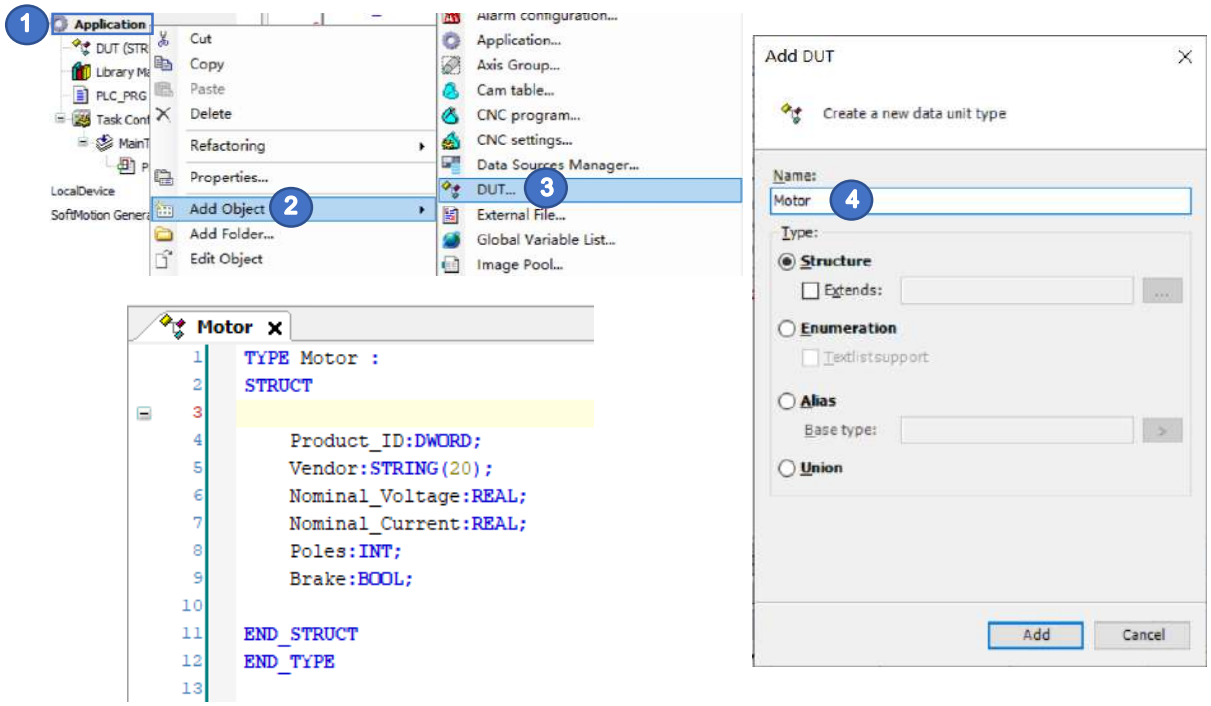
Pic 0-40



[Example1]: State Motor as structure in DUT, including information of model, factory, rated voltage, rated current and with/without brake. Extend structure Motor_hcfa based on structure Motor, add element protection level and rated torque and call the structure in program.

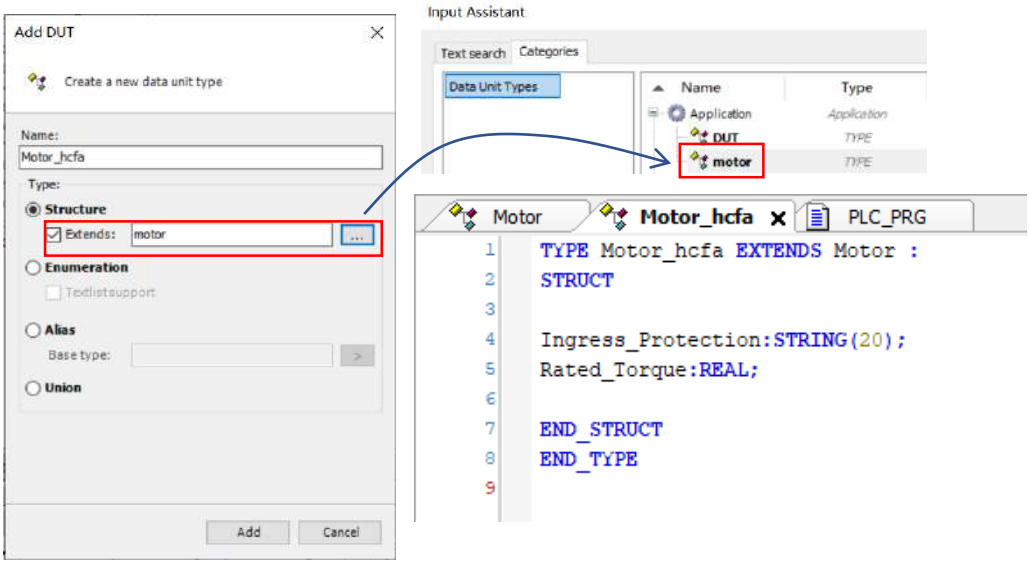
Statement structure Motor:

Pic 0-41



Extended structure Motor_hcfa:

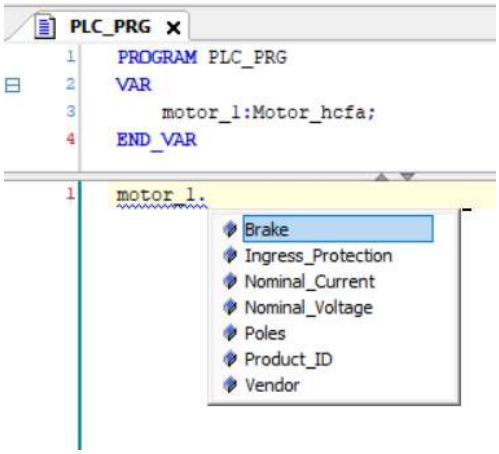
Pic 0-42



Call Motor_hcfa in program:

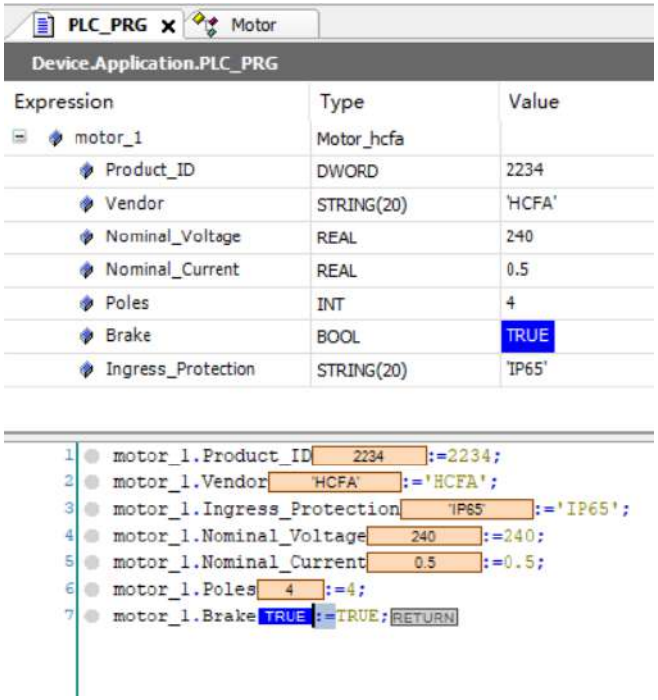
Like FB, instantiate the structure first and then the stated structure can be used directly. Its element of variable can be called through ".".

Pic 0-43



Call and assign element of structure one by one:

Pic 0-44

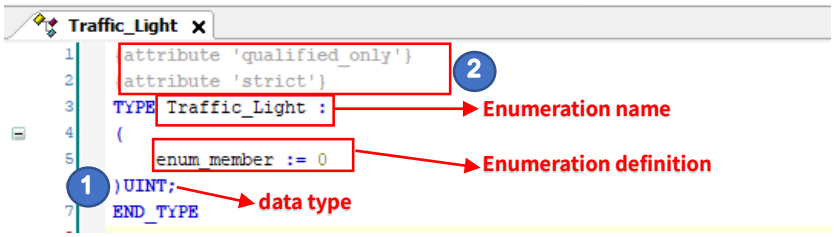


Enumeration

Enumeration is an named integration of integer constant and frequently seen in normal life. A variable with several possible values can be defined as enumeration like traffic lights with values of red, yellow and blue.

Statement of Enum:

Pic 0-45



- Basic data type is INT or can be appointed, shown as above pic ①.
- If not assigned, it will increase form 0 as INT integer constant.
- Integer can be copied to a enumeration.

Notice: Build property will be inserted in default enum statement in position of pic ②. Enum defined through {attribute 'qualified_only'} needs addressing with appointed global variable, while enum defined through {attribute 'strict'} will not attend mathematical operation or assign other data type to enum, including value of constant. User can delete property compile if not needed.

Or build property will occur below error reports.

C0358: 'test' is not a valid value for strict ENUM type 'Weekday'

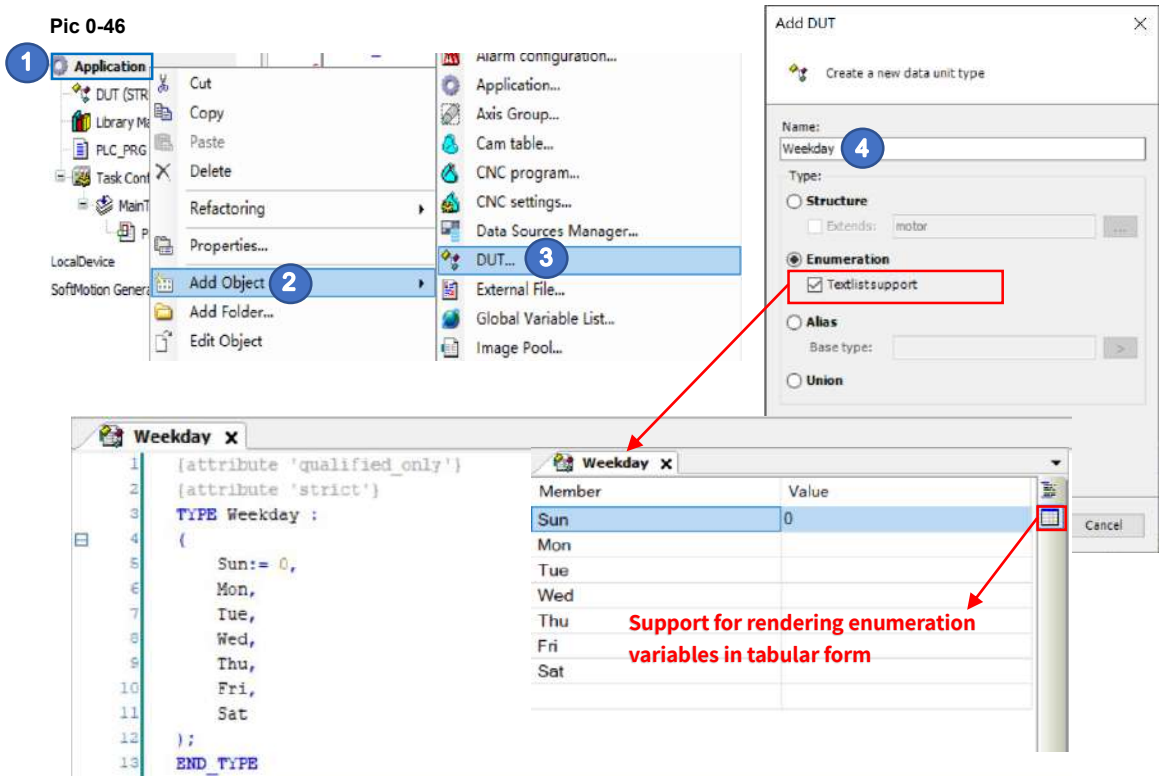
C0359: Arithmetics not allowed on strict ENUM type 'Weekday'

C0358: '(Weekday_1 + UINT#1)' is not a valid value for strict ENUM type 'Weekday'

Example 1: Use enum to display seven days of a week as Sun, Mon, Tue, Wed, Thu, Fri and Sat and change once a task cycle.

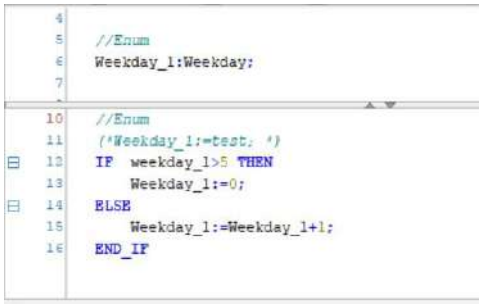
Statement:

Pic 0-46



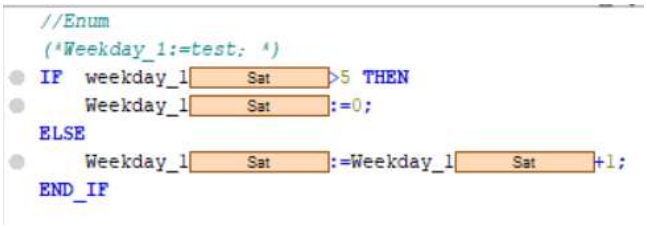
Call enumeration of Weekday in program:

Pic 0-47



Login and check running as below:

Pic 0-48

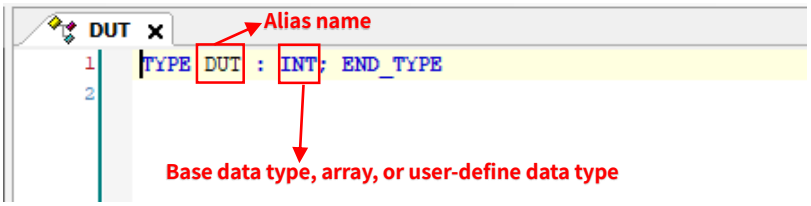


Alias

In brief, provide another name for a basic type data, array or custom data for better management on variable of statement. For example, to define a character string data type with fixed length for storage of IP address, user can take alias for easier distinction and modification of IP address length.

Statement:

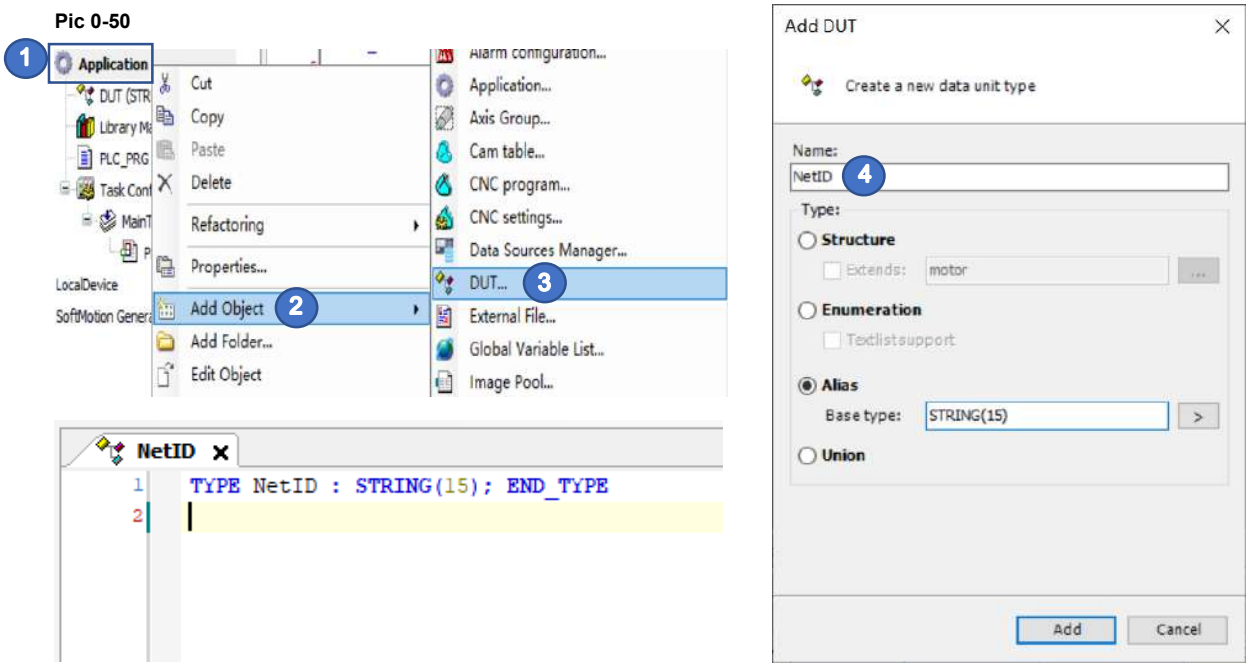
Pic 0-49



【Example 3】Define a string(15) with NetID to store IP address.

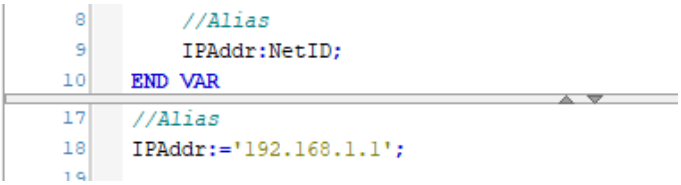
Statement:

Pic 0-50



Call and assign NetID in program, FB, function and other situations, change definition of alias is enough to modify length of NetID:

Pic 0-51



Union

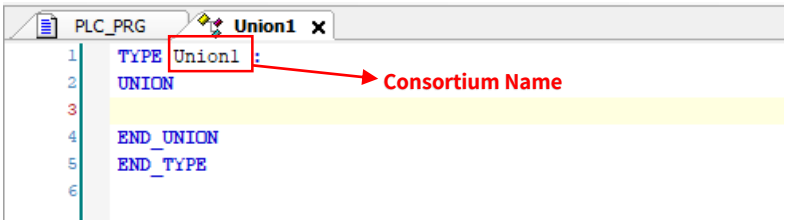
Also called shared, it can store different types of variable in same memory unit, like INT variable, BYTE variable and DWORD type. Shown as below:

Pic 0-52

	16#000	16#001	16#002	16#003
INT				
BYTE				
DWORD				

Statement:

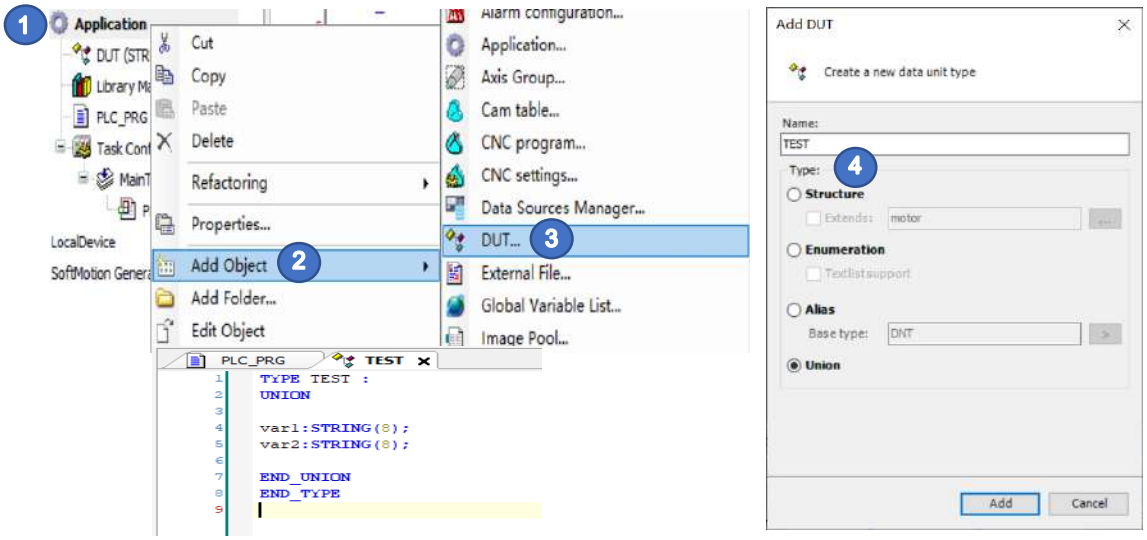
Pic 0-53



【Example 1】Define a union TEST with multi members. Assign one element and check others.

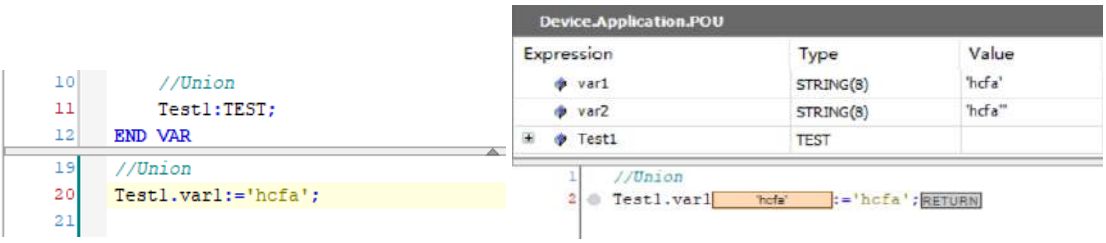
Statement:

Pic 0-54



Call TEST in program and assign element var 1, then var 2 will share input value:

Pic 0-55

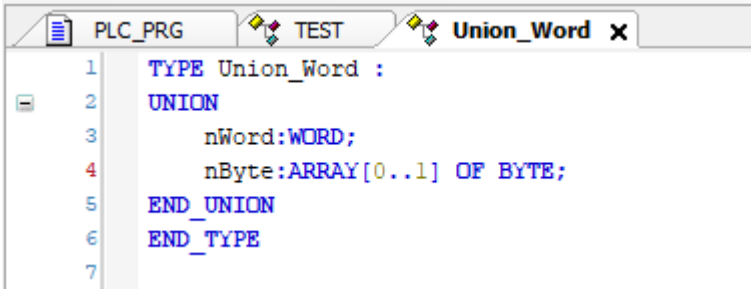


Element data type can also be different in union.

【Example 4】Use union to integrate variable of two bytes into one.

Statement:

Pic 0-56



Call Union_Word in program and compile as below:

Pic 0-57

```
12      Un_Word:Union_Word;
13      nByte_Low:BYTE:=16#12;
14      nByte_High:BYTE:=16#34;
15  END_VAR
21  Un_Word.nByte[0]:=nByte_High;
22  Un_Word.nByte[1]:=nByte_Low;
```

Login and Start. Assigned value of array nByte can be found. Value of nWord is also written due to union:

Pic 0-58

Device.Application.PLC_PRG1

Expression	Type	Value
Un_Word	Union_Word	
nWord	WORD	16#1234
nByte	ARRAY [0..1] OF BYTE	
nByte[0]	BYTE	16#34
nByte[1]	BYTE	16#12
nByte_Low	BYTE	16#12
nByte_High	BYTE	16#34

```
1  Un_Word.nByte[0] 16#34 :=nByte_High 16#34 ;
2  Un_Word.nByte[1] 16#12 :=nByte_Low 16#12 ;RETURN
```

Address mapping of Union_Word is as below:

Pic 0-59

Variable	High 8 bit	Low i bit
nWord	15~8	0~7
nByte[0]	15~8	
nByte[1]		0~7

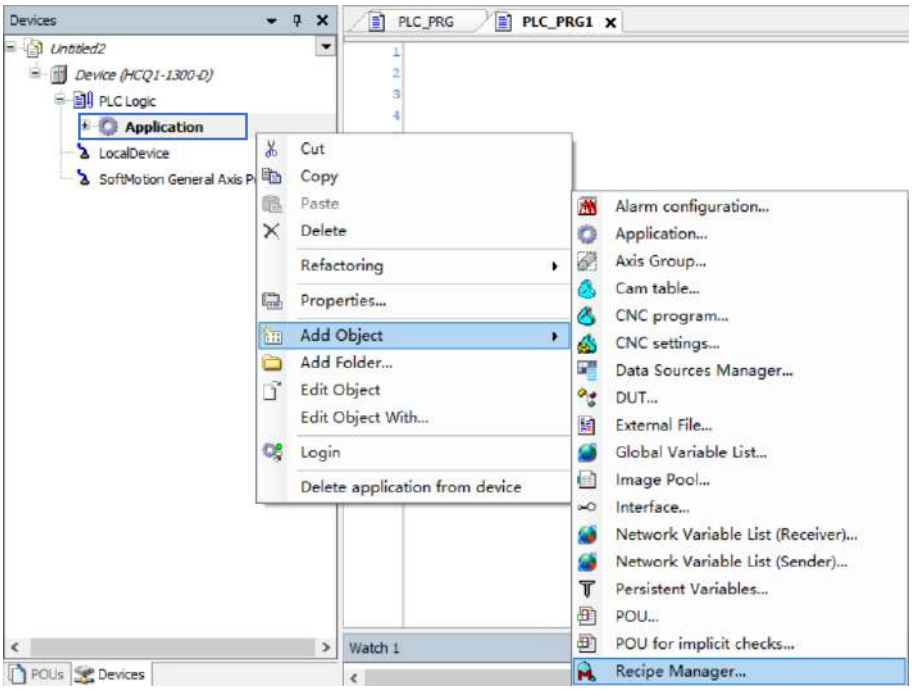
nByte[1] fit low 8 bit of nWord and nByte[0] fit high 8 bit of nWord, which makes two Byte type variable respond to elements of array and integrate two Byte values to one Word type variable. If data type of variable in union statement are not the same, storage of each data should be the same to avoid data error.

3.3.5 Recipe Manager

Recipe is array for providing information of production and control process. For example, materials for bread(including wheat flour, egg, butter, white granulated sugar, etc.) and baking time and other parameters. Recipe can be used for setting and monitoring control parameters. User can read and write through PLC, download from file or generate file.

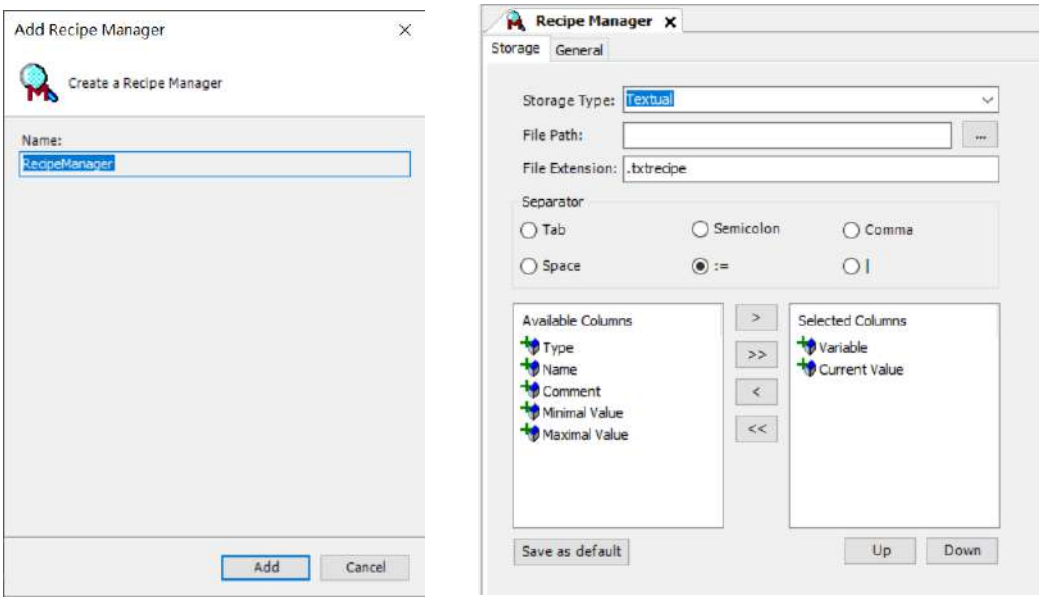
Application, right click to select “Add object” →“Recipe manager” to add manager.

Pic 0-60



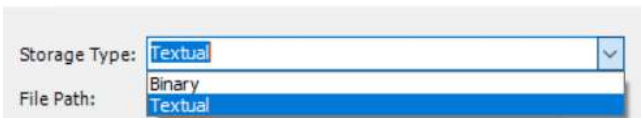
Interface of adding manager in left and interface of configuration after it's added.

Pic 0-61



In configuration interface, user can choose Store type as Textual or Binary,

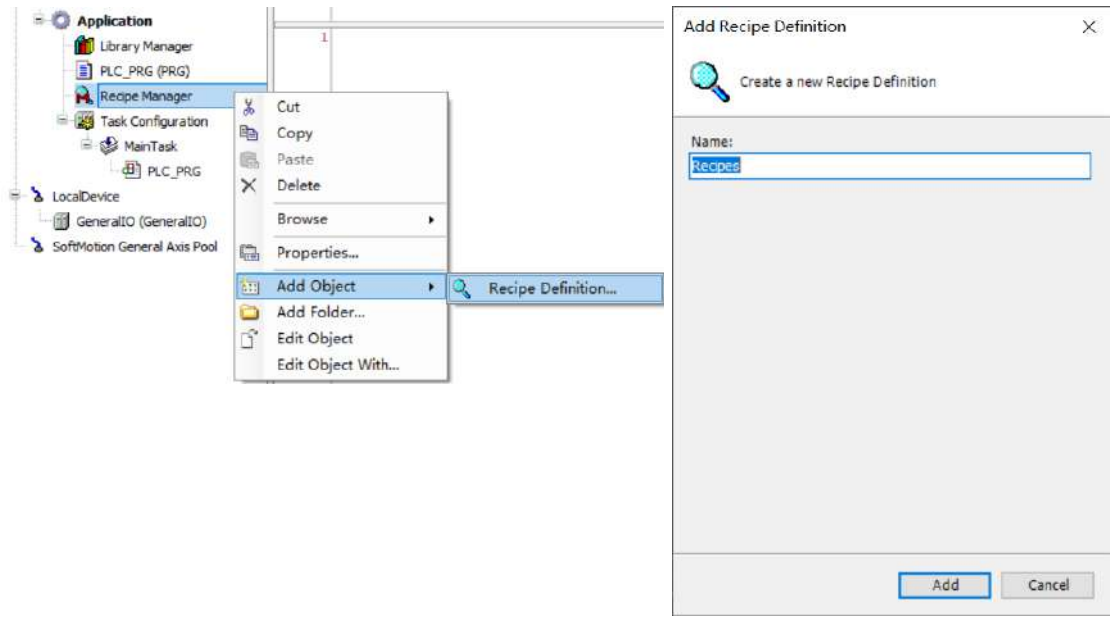
Pic 0-62



File Path can be appointed in Storage options and choose File extension. Text will be divided from recipe name based on selected Separator, and display as <method>.<define method>.<extension file> and Separator will only take effect in storage type is text. All recipe definition columns are displayed as Available columns. Optional columns is on right and will stored.

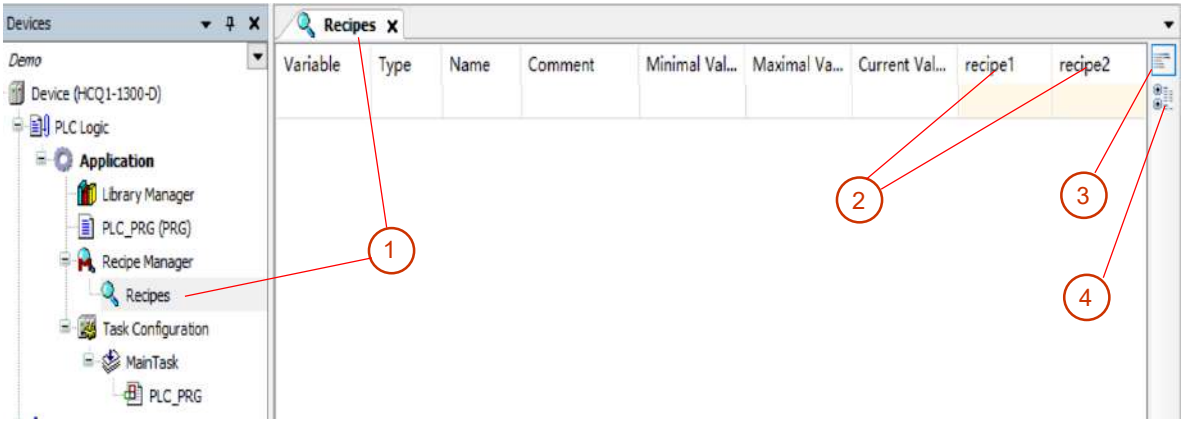
After above settings, right click “Recipe Manager” and select “Add Object” → “Recipe Definition” to add new definition.

Pic 0-63



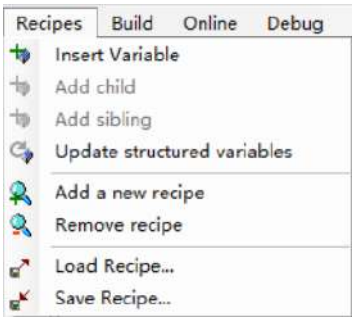
Shown as below, in Recipe ①, user can define different values for different recipe ②, and switch display of definition through ③ and ④. Displayed content is column content in recipe manager, among which, variable can be added through stated variable in right . User c ... ustom Name and Comment in recipe definition, and limit value that could be entered by setting Maximum and Minimum Value.

Pic 0-64



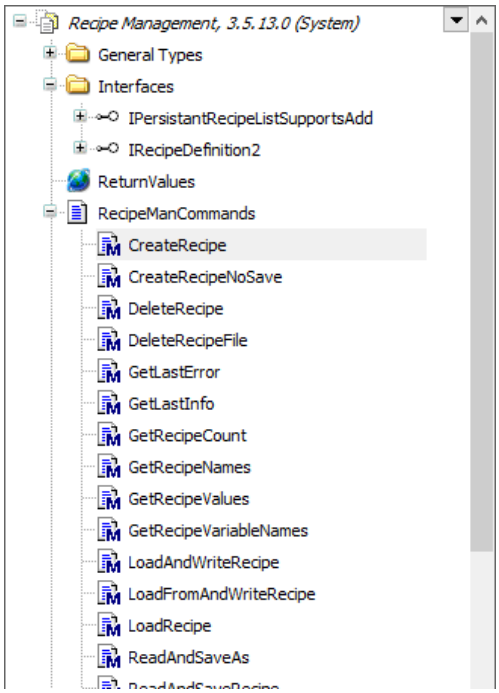
Different Recipe ② can be added through options of Recipes, where user can also delete, load or save recipe files.

Pic 0-65



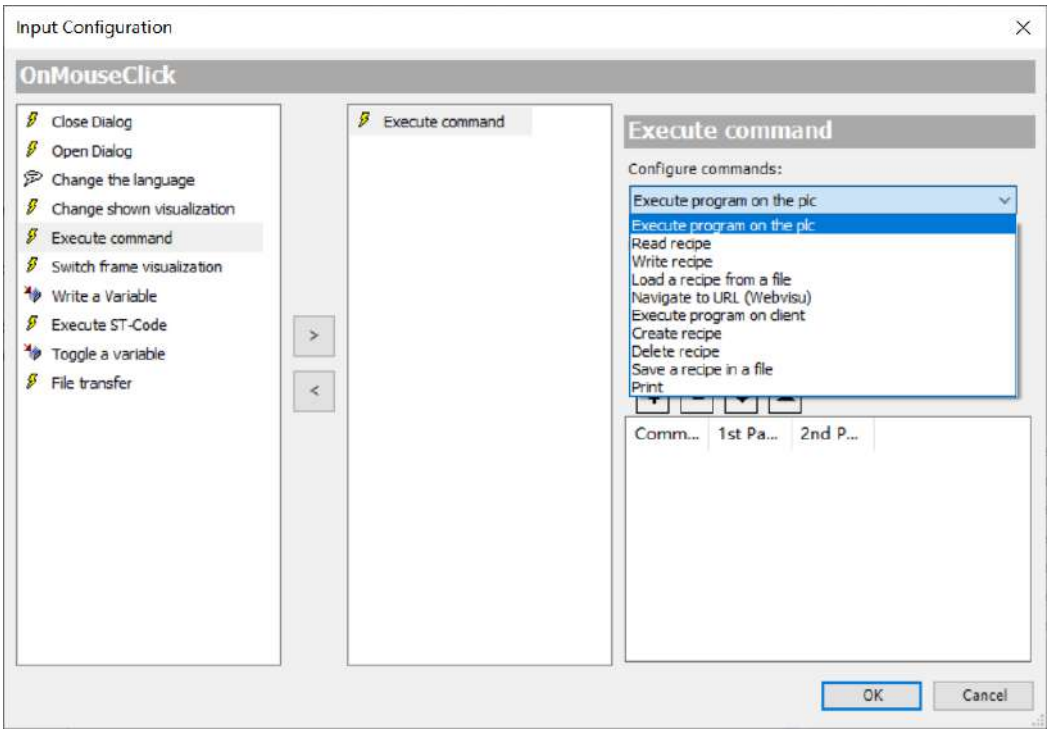
Except for editing recipe file in CODESYS interface, user can also provide “Recipe Management” for operations of create, delete, load, save or error check.

Pic 0-66



Embedded Visualization interface also provides interface for Recipe, which will be explained in details:

Pic 0-67



【Example 1】Create two recipes to produce bread of strawberry and orange

Needed variables:

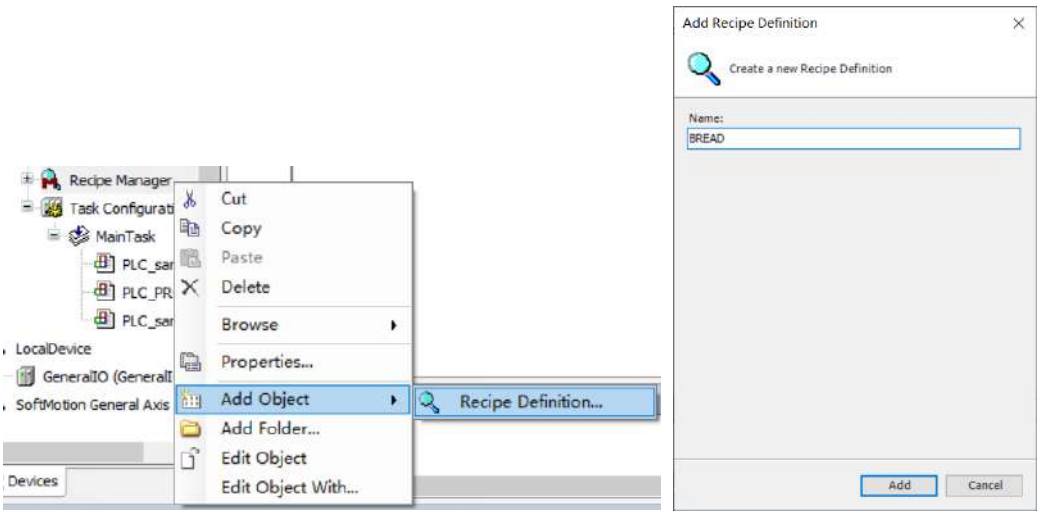
Pic 0-68

```
PROGRAM PLC_PRG1
VAR
    egg:INT;                //Egg
    mlik:REAL;              //Milk
    flour:REAL;             //Flour
    butter:REAL;            //Butter
    sugar:REAL;             //Sugar
    baking_time:TIME;       //Baking time
    strawberry_jam:REAL;    //Strawberry jam
    orange_jam:REAL;        //Orange jam
    duration:TIME;          //duration
    product1:Product;       //The product name

END_VAR
```

Add new recipe definition:

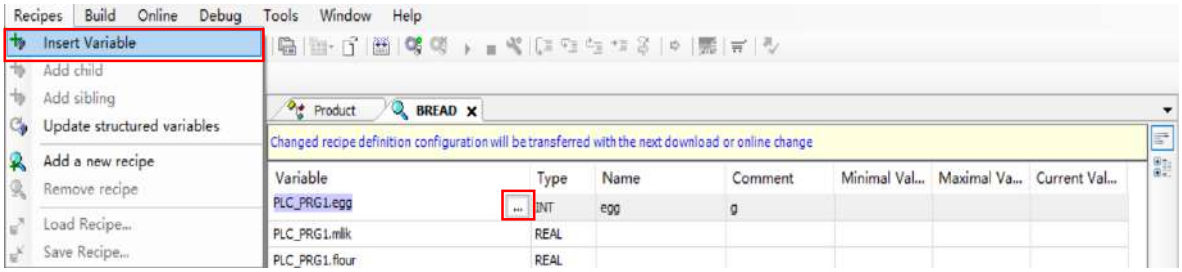
Pic 0-69



Insert variable in recipe:

Add variable through inserted variable or ... in definition:

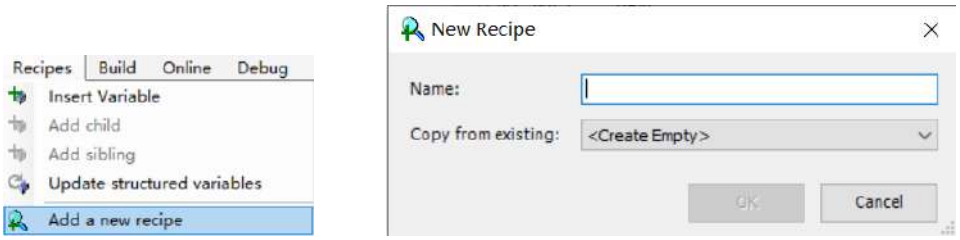
Pic 0-70



Add new recipe:

Recipe of menu bar, Add new recipe, enter Orange_Bread and Strawberry_Bread in dialogue.

Pic 0-71



Enter needed recipe value as below:

Variable and Type are loaded automatically when adding variable from program. Name, Comment, Minimum value, Maximum value are added manually. After login and online monitoring, Present value will display using recipe value.

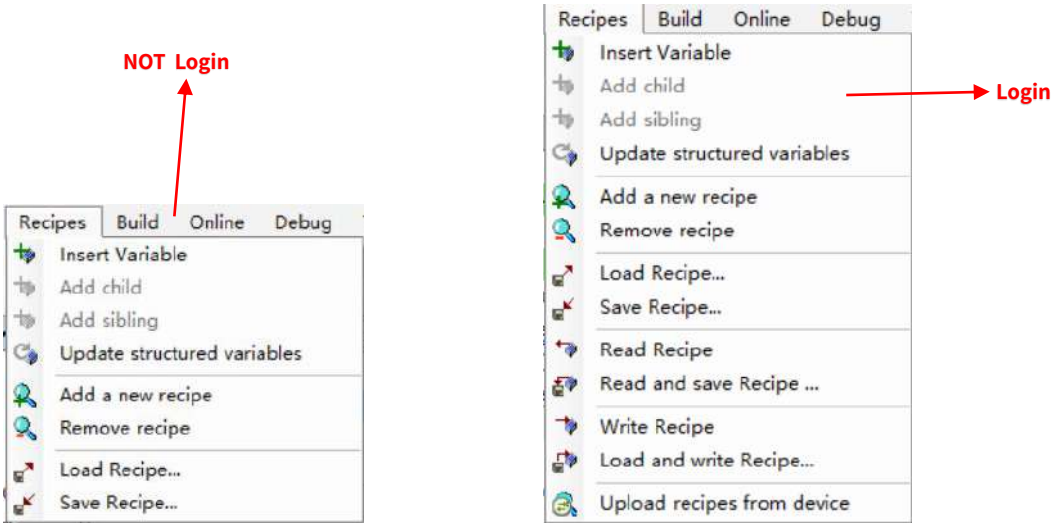
Pic 0-72

Product | BREAD | PLC_PRG1

Variable	Type	Name	Comm...	Minim...	Maxim...	Curren...	Orang...	Starw...
PLC_PRG1.egg	INT	Egg	g			0	50	40
PLC_PRG1.mlik	REAL	Mlik	g			0	200	250
PLC_PRG1.flour	REAL	Flour	g			0	38.8	42.5
PLC_PRG1.butter	REAL	Butter	g			0	20	33.3
PLC_PRG1.sugar	REAL	Sugar	g			0	25.5	20.8
PLC_PRG1.baking_time	TIME	Baking t...	min			T#0ms	t#15m	t#15m
PLC_PRG1.strawberry_jam	REAL	Strawb...	g			0	0	30.3
PLC_PRG1.orange_jam	REAL	Orange ...	g			0	22.2	0
PLC_PRG1.duration	TIME	The pro...	min			T#0ms	t#15m	t#15m

Before login, user can load, delete and save recipe in optional list of Recipe Definition. After login, user can also execute writing and reading of recipe.

Pic 0-73



Current value will be written value after executing recipe write to Orange_Bread.

Pic 0-74

PLC_PRG1 | Product | BREAD | Recipes

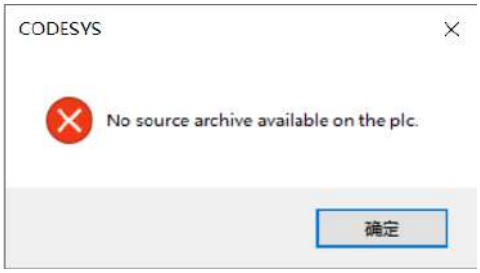
Variable	Type	Name	Comm...	Minim...	Maxim...	Curren...	Orang...	Starw...
PLC_PRG1.egg	INT	Egg	g			50	50	40
PLC_PRG1.mlik	REAL	Mlik	g			200	200	250
PLC_PRG1.flour	REAL	Flour	g			38.8	38.8	42.5
PLC_PRG1.butter	REAL	Butter	g			20	20	33.3
PLC_PRG1.sugar	REAL	Sugar	g			25.5	25.5	20.8
PLC_PRG1.baking_time	TIME	Baking t...	min			T#15ms	t#15m	t#15m
PLC_PRG1.strawberry_jam	REAL	Strawb...	g			0	0	30.3
PLC_PRG1.orange_jam	REAL	Orange ...	g			22.2	22.2	0
PLC_PRG1.duration	TIME	The pro...	min			T#15ms	t#15m	t#15m

Notice: Chinese can be used when editing name and comment of recipe definition, but mistaken code will occur in recipe interface program(included in library file "Recipe Management"). The main reason is that it only supports ASCII code for String type variable here. For example: Recipe Man Command.Get Recipe Variable Names.

3.3.6 Upload and Download of Source Code

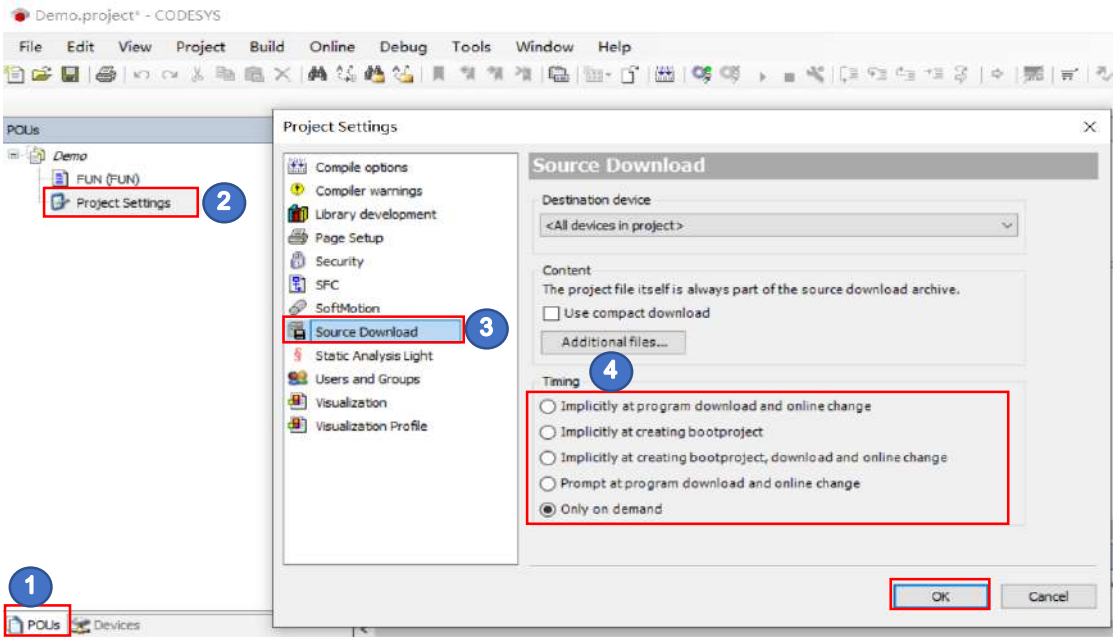
When trying to login and run the program under editing, CODESYS will not download source code to target device in default and uploading program from target device will report errors as below:

Pic 0-75



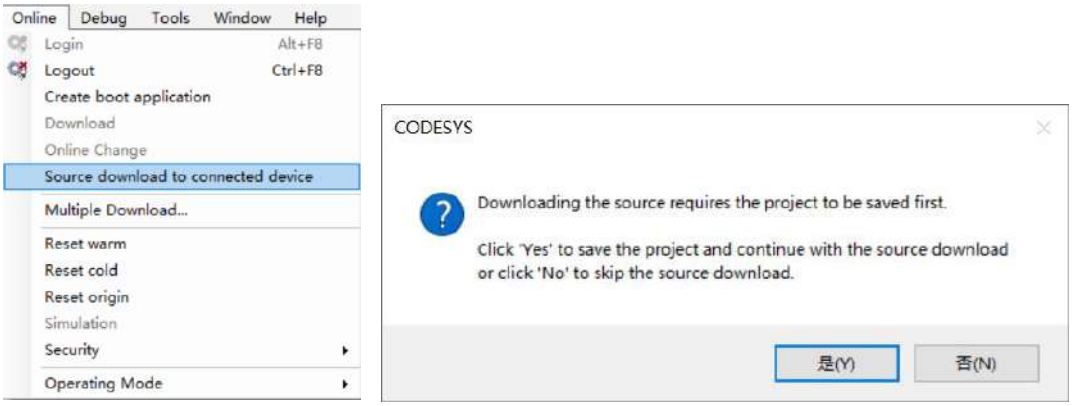
Settings in Project Settings is needed if user still need to upload source code for future upload and edit:

Pic 0-76



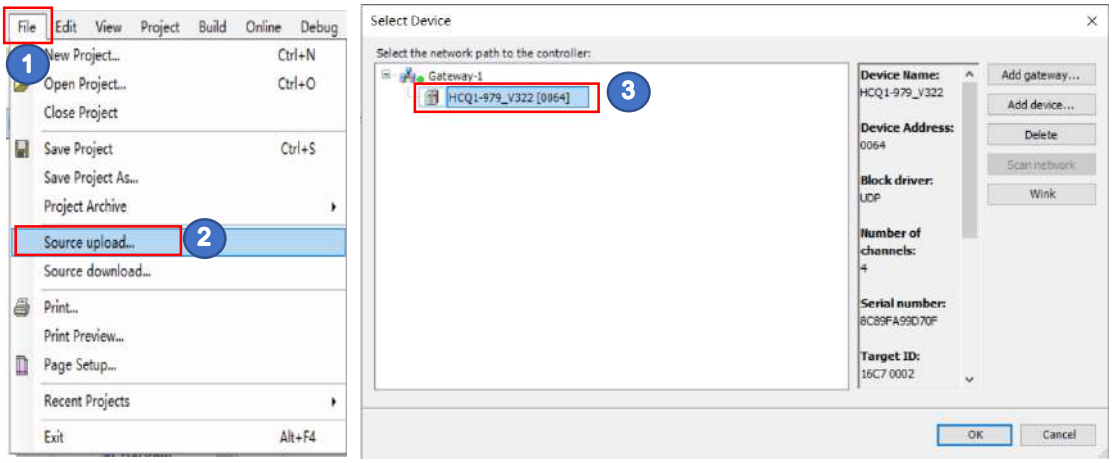
Set in step 4 and click YES to save configuration. For example, after selecting On Requirements Only and login, find in menu Online, select Source download to connected device.

Pic 0-77



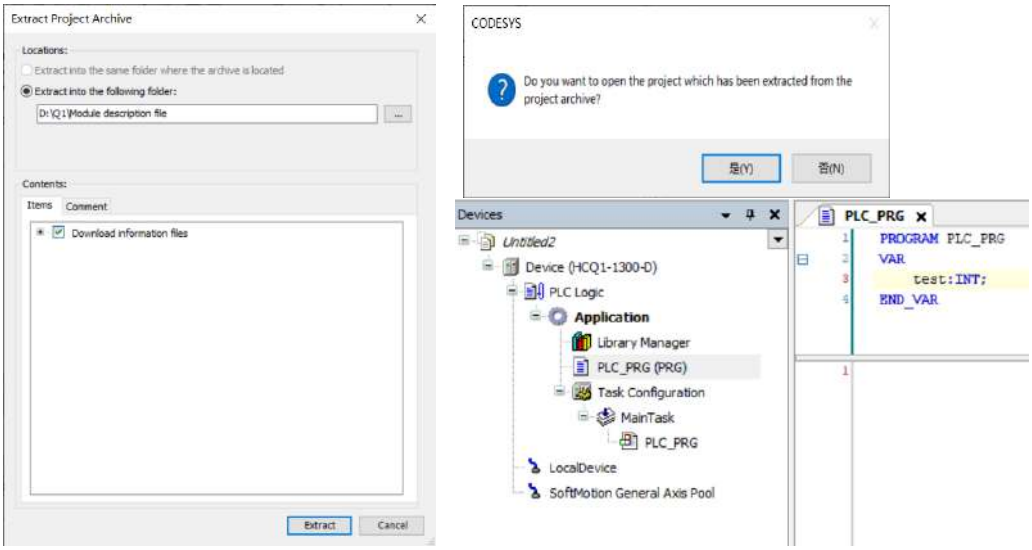
After downloading, future user can upload source code from controller then:

Pic 0-78



Select target path of uploading source code and open, then uploaded source code can be found:

Pic 0-79

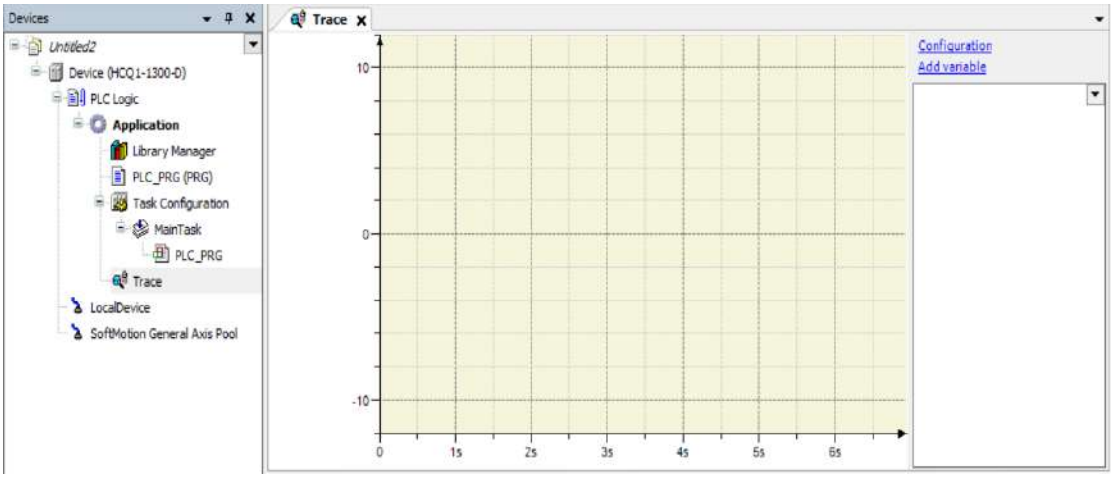


3.4 Trace

Like oscilloscope, as graphic data monitoring software offered by CODESYS, Trace is very useful in program debugging and diagnosis. Data runs too fast for real time analysis, while it can be recorded fully through sampling tracking, for example, current position, speed and acceleration of motor during running. All process of system running can be then observed through data analysis.

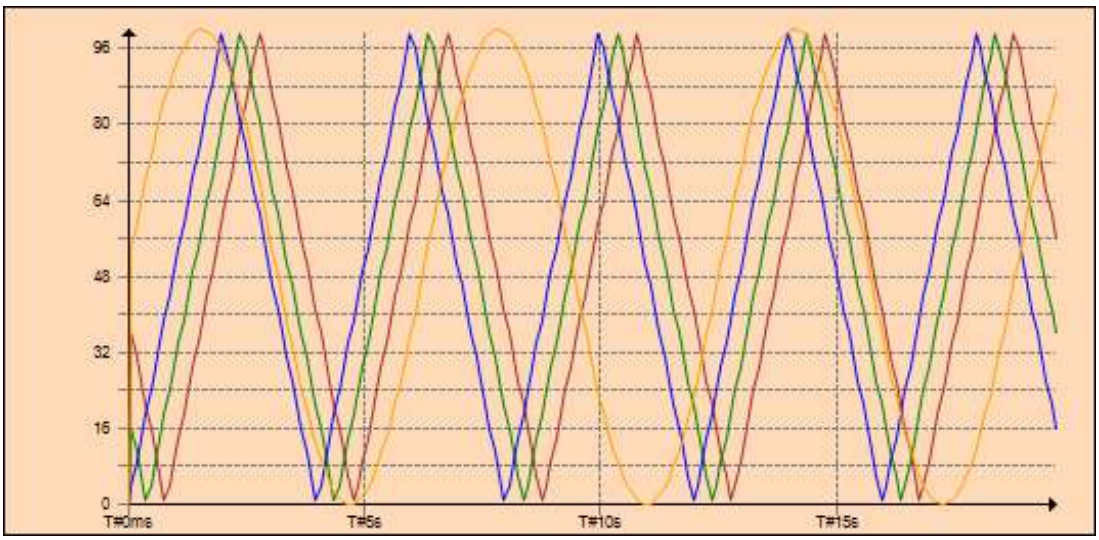
Two windows, Trace Configuration and Trace Variable, are provided to collect oscillogram during PLC running. Multi Trace configuration files can be created in each PLC application and trigger variable, cycle, data save can be set.

Pic 0-1



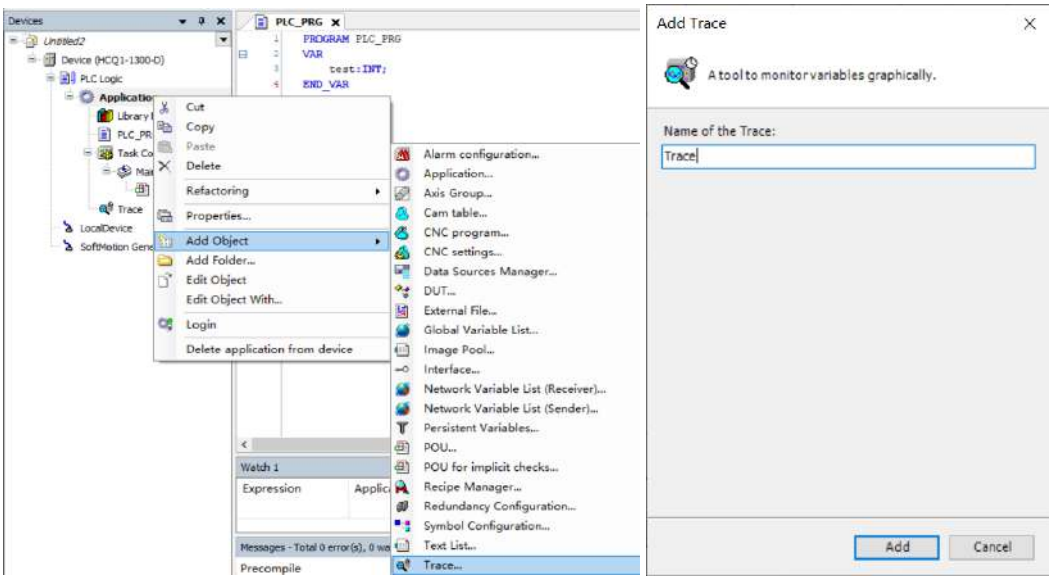
Real time sampling result:

Pic 0-2



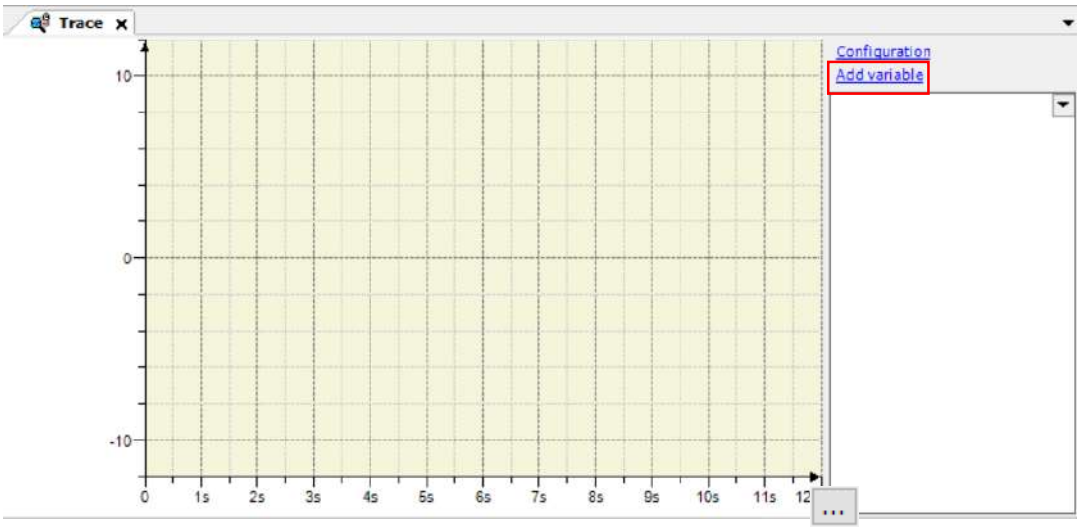
- Create New Trace

  Right click Application, select Add Object, Trace, enter Trace name and click Add to confirm.



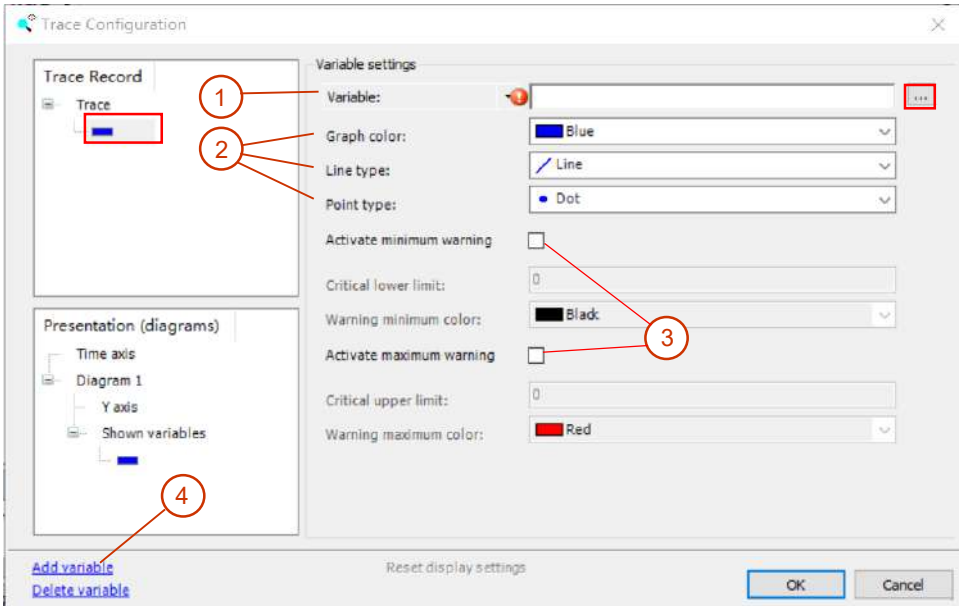
Open new Trace and select Add Variable on right to add tracking variable.

Pic 0-3



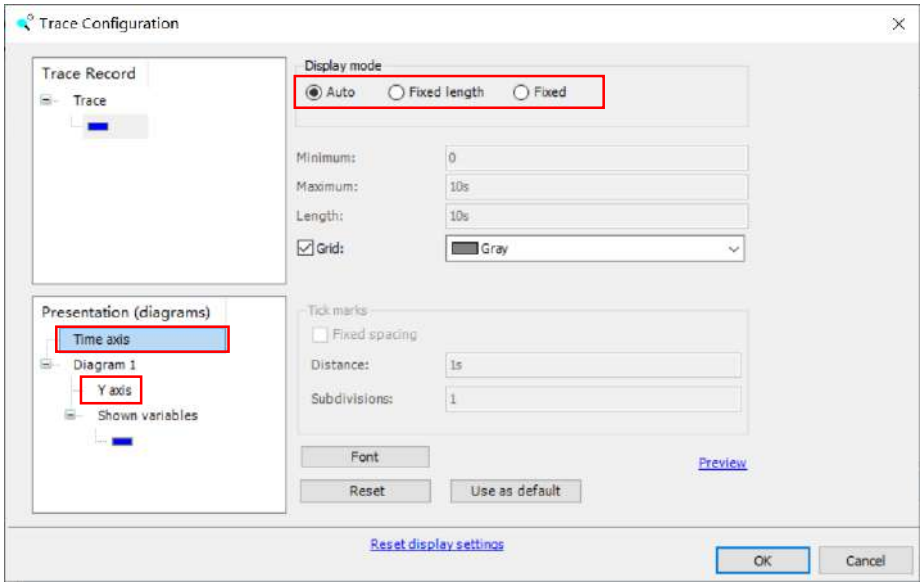
In dialogue of Trace Record, select Add variable in tree menu, through right of variable ①, get in Input Helper and add needed variable for sampling tracking. Set colour, line type and point type through ② and activate maximum value warning through ③, add and delete variable of tracking configuration through ④ at left corner.

Pic 0-4



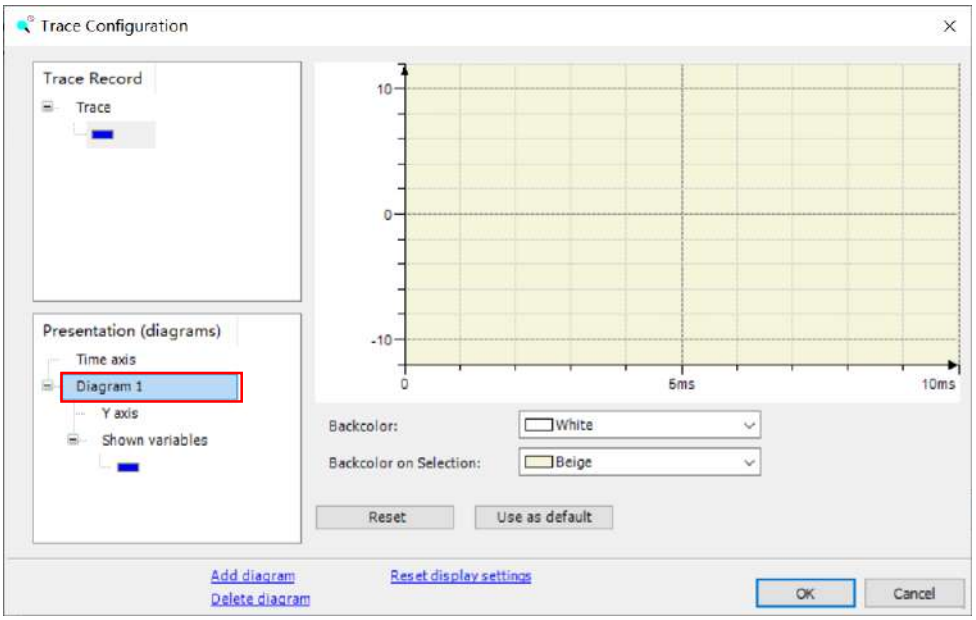
In dialogue of Presentation（Diagrams）, set tracking background and axis, set Time Axis display mode as Auto, Fixed length, or Fixed. Then the time axis will adjust automatically. Similar in Y axis, user can set configuration of lettering, grid and others according to need.

Pic 0-5



Set tracking background in tree menu Diagram 1.

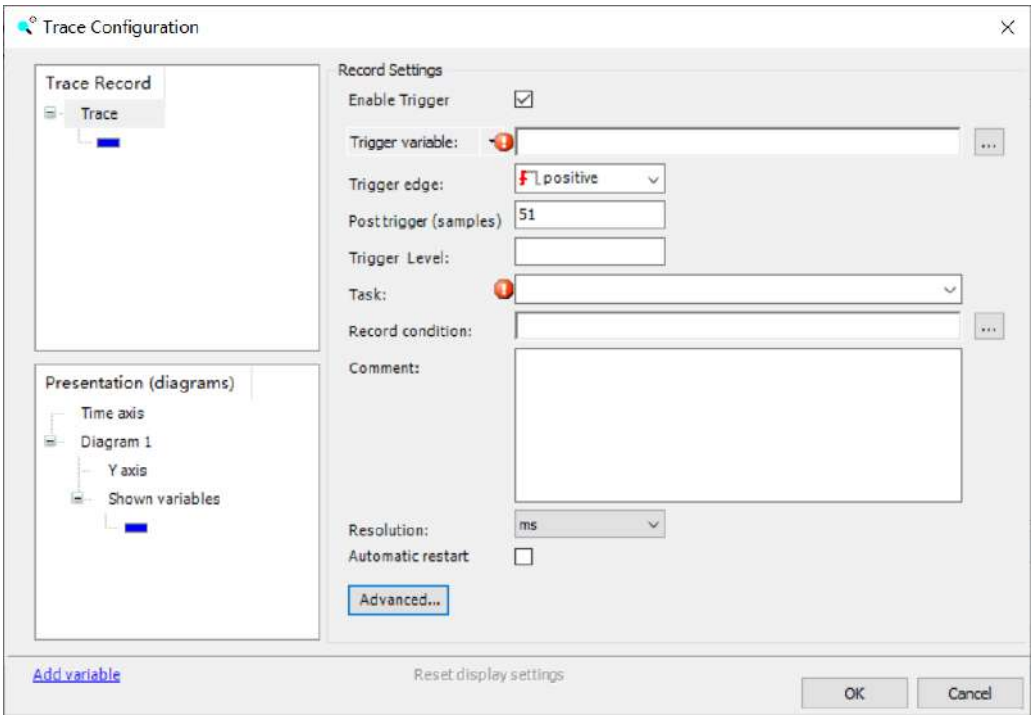
Pic 0-6

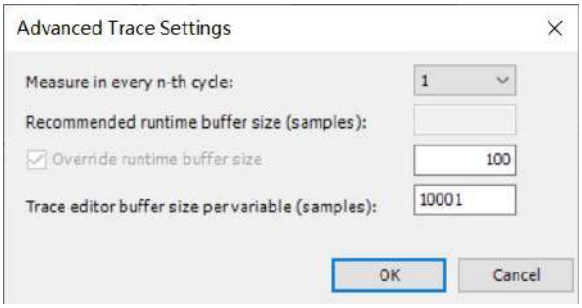


- Triggering Sampling Tracking

Different ways of tracking trigger are offered by CODESYS and select the function in tree menu Trace. If not selected, trigger needs to be activated manually:

Pic 0-7





1. Trigger Variable

   It's a BOOL type variable to express or imitate variable, enum variable, etc. Tracking will take effect according to set time, once Trigger Variable is True or meet certain value.

2. Trigger Edge

   positive：Triggering when rising edge or data signal is bigger than Trigger Level.

   negative：Triggering when falling edge or data signal is bigger than Trigger Level.

   both：Edge trigger, both rising or falling edge.

3. Posttrigger（samples）

   Set number of tracking record from 0 to $(2^{32}-1)$ and default is 50, after trigger event.

4. Trigger Level

   Set number of analog for triggering when it's used as trigger variable.

5. Task

   Select one task to define trigger task for current input signal.

6. Record Condition

   Define a BOOL variable, express, value or property to stop sampling tracking.

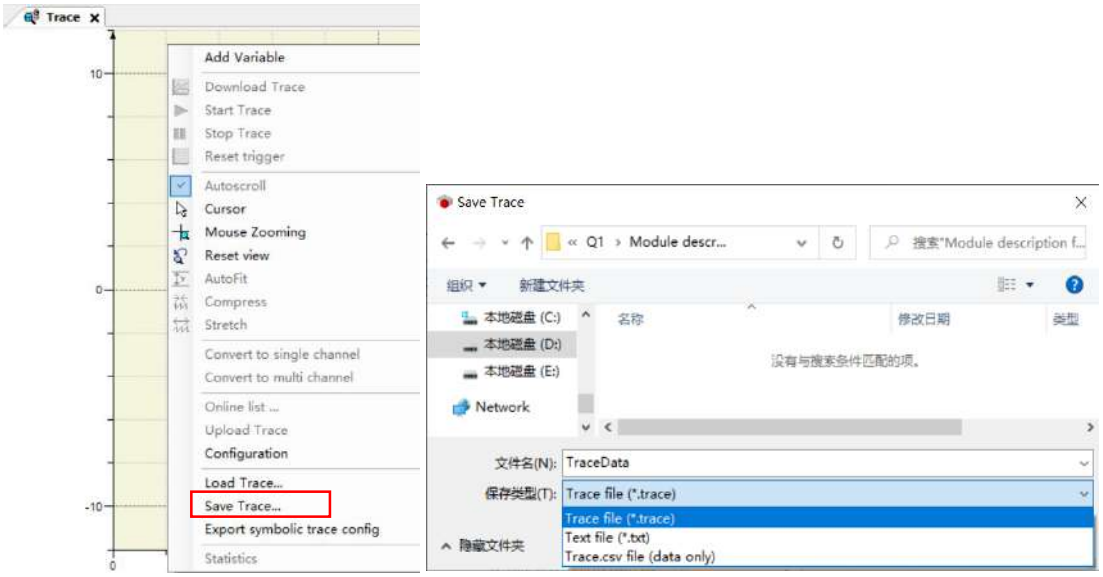
7. Comment

   An text comment of current record.

- Save Data

  Save collected PLC data for future analysis and it can be done by clicking Save Trace and selecting path for saving, naming in sampling tracking interface directly. Format can be ".trace"、".txt"、".trace.csv", among which, .trace" file can be opened by CODESYS, other two via Excel directly.

Pic 0-8



8. Advanced Trace Settings

In dialogue of Advanced Trace Settings, cache memory area can be set in running system.

- Normal Functions of Trace

Pic 0-9

Icon	Instruction
	Download Trace for collecting tracking curve will start immediately if trigger is not set.
	Start or Stop Trace
	Reset trigger for reset after trigger event and restart.
	Cursor for setting X axis value. Two cursors can be added for each tracking to separate X axis and relative position between two cursors.
	Mouse zooming, to activate mouse zoom and draw a rectangle to redefine curve area in tracking window Use wheel to zoom X-Y axis and use <-><+> on keyboard to realize same function. Press <shift> and use wheel to zoom X axis only. Press <ctrl> and use wheel to zoom Y axis only.
	Reset view, for view reset after it's defined in tracking configuration
	AutoFit, for auto adjust appearance of tracking after set in tracking configuration.
	Compress, for observing tracking variable in shorter time for multi orders.
	Stretch, for stretching displayed value of sampling tracking

3.5 Persistent Variable

During running of PLC system, certain variables need to be saved for future calling, in circumstances of shut down or abnormal power off. CODESYS provides RETAIN and PERSISTENT RETAIN, while the second one is more frequently used.

Chapter 4 Visualization Interface Edit

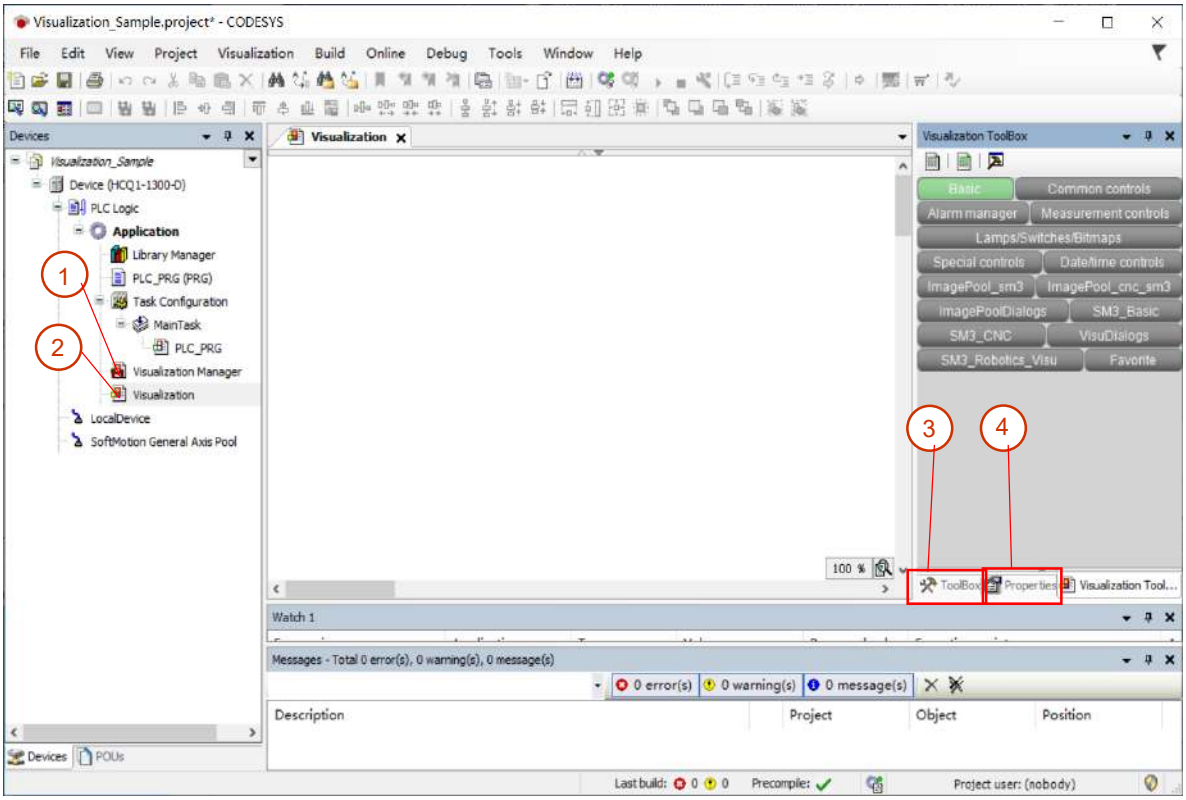
4.1 Introduction of Visualization

Users always find edit of complex programming logic or debugging interface hard to operate, even with complete text comment. For easier debugging and interface, visualization is provided.

Visualization object can be managed in Visualization Interface, including elements based on personal need. Multi objects can be created in one standard Application for interaction.

Through graphic editor, user can display real time internal variable of program and observe running status directly.

Pic 0-1



① Visualization Manager for parameter configuration of visualization.

② Visualization for editing needed interface.

③ ToolBox provides various graphic.

④ Properties for editing relative properties.

Visualization of Q1 also includes different custom terminal;

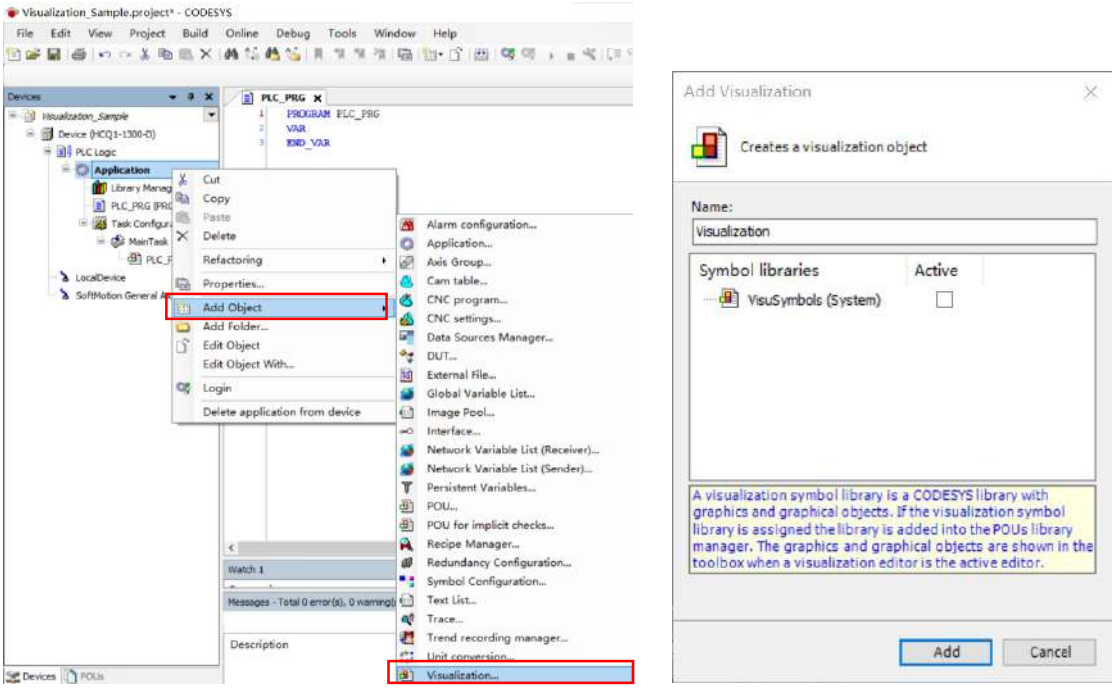
WebVisu：link to application program through web page server

TargetVisu：run visualization in external device.

4.2 Create New Visualization

CODESYS provides visualization for imitating, operating or monitoring machines or devices. Right click Application in tree menu and find Add Object, Visualization, name it and click Add to add new visualization, including an editor and manager.

Pic 0-1



Corresponding visualization library will be added to library manager for configuration automatically after it's added to project.

Pic 0-2

If the library is always set as Placeholder FUN Library, the actual version is not settled and it needs to be added into project. Only in this way can the activated configuration file be certain to use which version.It's different from Placeholder FUN Library of special device as which can work through device description. Default library includes VisuElems, VisuElemMeter, VisuElemWinControls, VisuElemTrace, VisuInputs, which contains more libraries in itself.

Notice:

Visualization library file will be added automatically at first creation in default and will not be used obviously.

4.3 Basic Visualization Operation

User can execute basic operations like adding visualization elements, alignment, delete in the editor.

4.3.1 Add Visualization Element

Two ways to add visualization element: “draw visualization elements directly in the editing area” and “drag visualization element to edit”

- Draw view elements directly in the editing area

Select needed view element in tool box and move it to needed area of view editing area, then the element will be added to view editing based on appointed size and position.

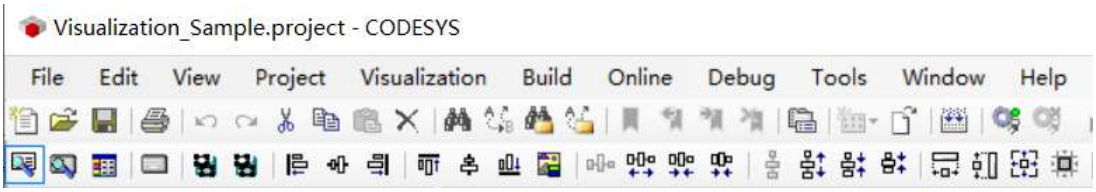
- Drag view element to edit

Select needed element and drag it to view editing, it will then be added in default size.

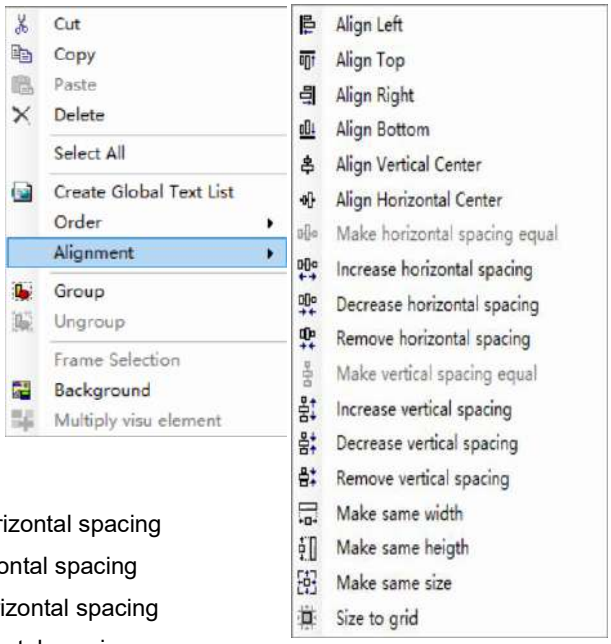
4.3.2 Align Visualization Element

If added elements need to be aligned, select main element(first selected element) and then choose align method, right click Align to confirm. Or other methods can be chosen directly in tool bar as below:

Pic 0-1



Pic 0-2



- Align Left: Align to left for selected visualization elements
- Align Top: Align to top for selected visualization elements
- Align Right: Align to right for selected visualization elements
- Align Bottom: Align to bottom for selected visualization elements
- Align Vertical center: Align to vertical center for selected visualization elements
- Align Horizontal center: Align in horizontal center for selected visualization elements
- Make horizontal spacing equal: in equal horizontal spacing
- Increase horizontal spacing: Increase horizontal spacing
- Decrease horizontal spacing: Decrease horizontal spacing
- Remove horizontal spacing: Remove horizontal spacing
- Make vertical spacing equal: Make vertical spacing equal
- Increase vertica spacing: Increase vertica spacing
- Decrease vertica spacing: Decrease vertica spacing
- Remove vertical spacing: Remove vertical spacing

- Make same width：Make same width
- Make same heigth：Make same heigth
- Make same size：Make same size
- Size to grid：Size to grid

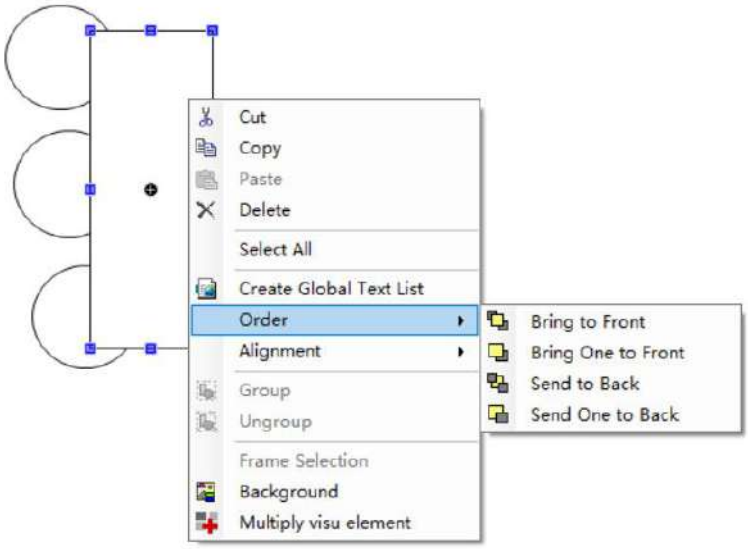
4.3.3 Delete Visualization Element

Select needed element and single right click, select Delete. Or select needed element, press <Delete> on keyboard.

4.3.4 Change Visualization Sequence

After part of elements are added and combination of graphic may be needed, and piled ones will shade on earlier ones during operation. Change the needed graphics like this, right click, find Order in drop down list and select needed action.

Pic 0-3



Bring to Front：Place selected element on top.

Bring One to Front：Move selected element one step upward.

Send to Back：Place selected element at bottom.

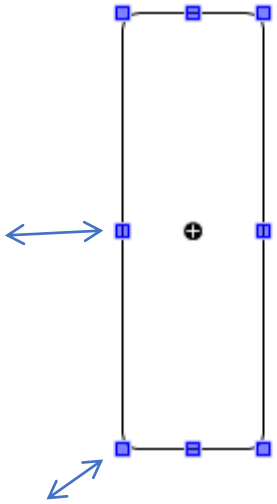
Send One to Back：Move selected element one step downward.

4.3.5 Adjust Size and Position of View Element

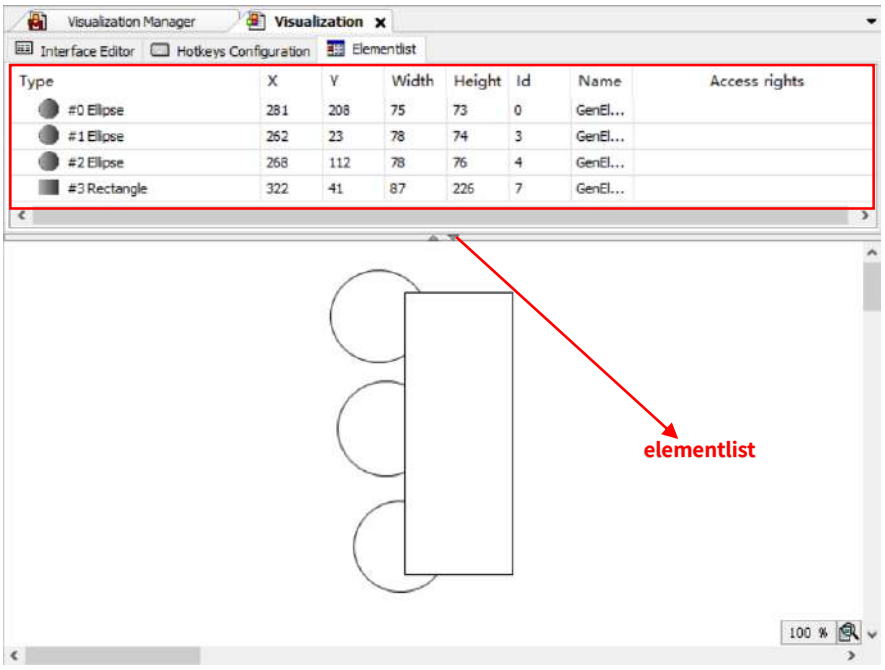
For added element, adjust its size through offered resize box directly after it's selected.

But adjustment has to be restricted to value specification with this method. In Element List called by downward triangle of the interface, user can adjust current X Axis and Y Axis for precise Width and Height, or modify the name of element freely.

Pic 0-4



Pic 0-4



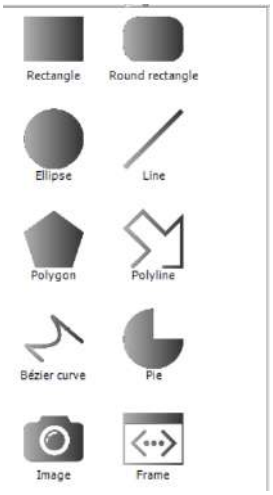
4.4 Tool Box

Together with visualization editor, tool box provides embedded visualization elements.

4.4.1 Basic Control Tools

It has normal graphic elements like element creating text box, text display box, color display box and graphics.

Pic 0-1



Text Box/Text Display Box

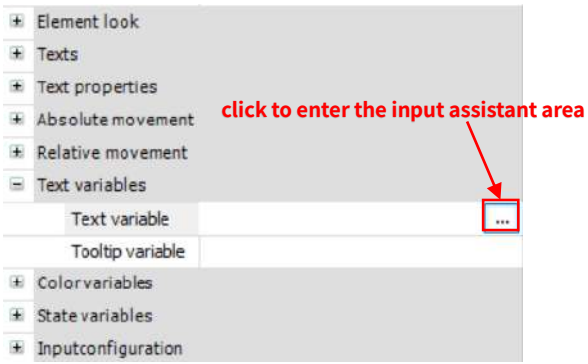
As there's no separate text box in CODESYS, add rectangle/rounded rectangle/oval box.

1. Create Text Box

Two steps: 1. Create mapping relation. 2. Set variable type which needs to e displayed.

Create mapping relation, select added view element(rectangle/rounded rectangle/oval), set Text Variable in properties menu on right, click Input Helper logo to map related mapping and click Confirm to finish.

Pic 0-2



For configuration of display type, enter text in Text Box for fixed constant and separate setting for unfixed constant. For example, enter "%s" for character string, except for "s", it can also be realized through other formats of sprintf in standard C library FUN.

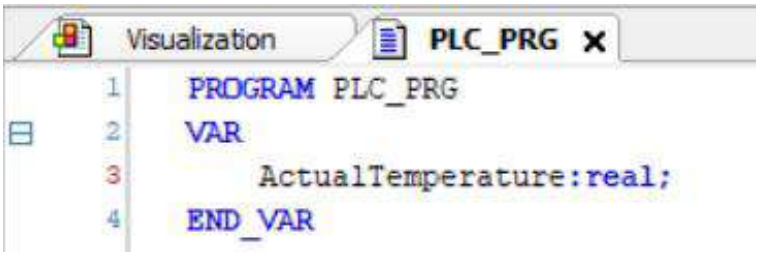
Pic 0-3

Format output order	Description
d、i	decimal
o	Unsigned octal
x	Unsigned hexadecimal
u	Unsigned decimal
c	Single String
s	String
f	Real number

Example 1: Create an Text Box to display actual room temperature.

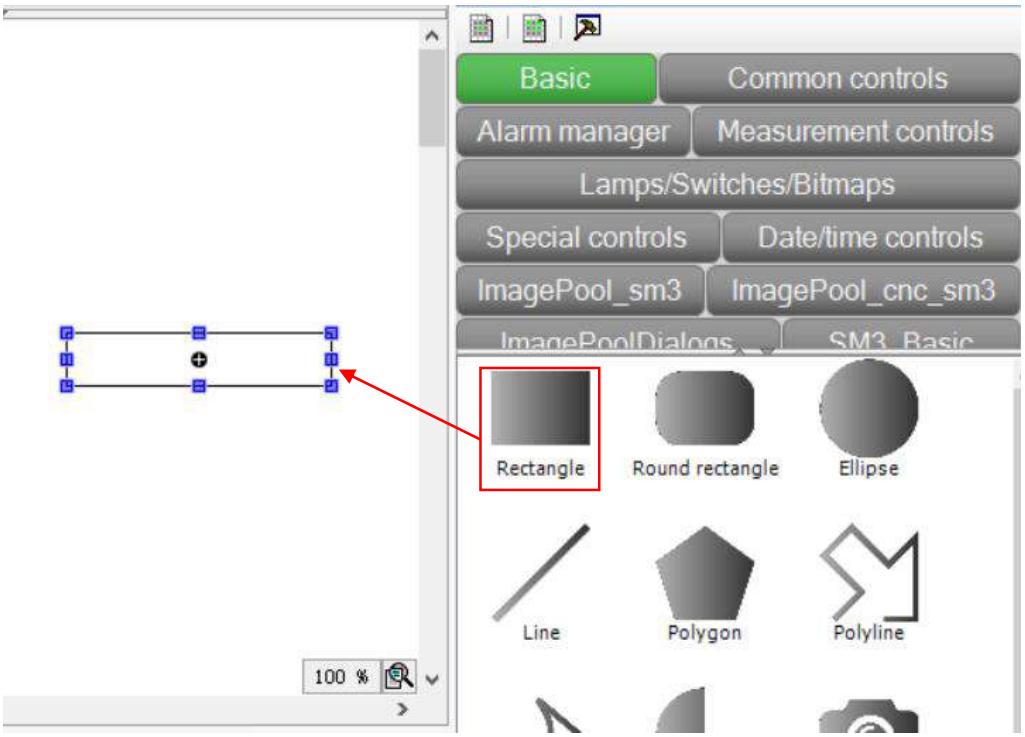
Add needed temperature variable in program.

Pic 0-4



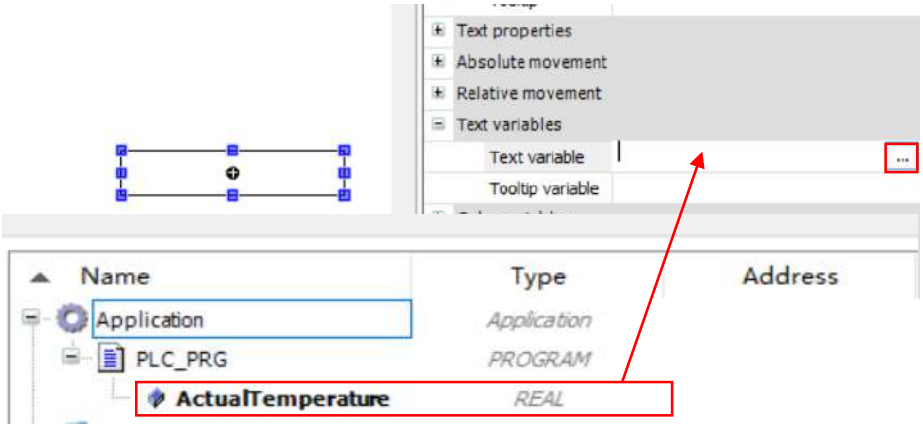
Choose rectangle from Basic in tool box.

Pic 0-5



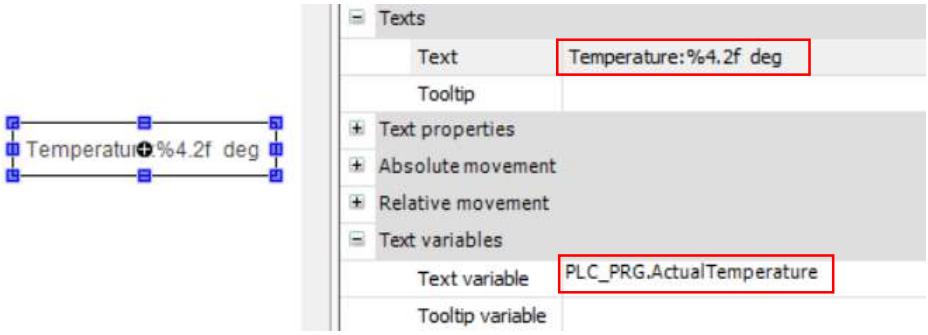
Select rectangle, find Text Variables in Properties menu and complete mapping.

Pic 0-6



For configuration of display type, select Text Box in Properties menu on right, enter "Temperature:%4.2f deg" and "%4.2" is in float type data display, which reserves real number with width of 4 characters and 2 numbers after decimal point, rest will be fixed text.

Pic 0-7



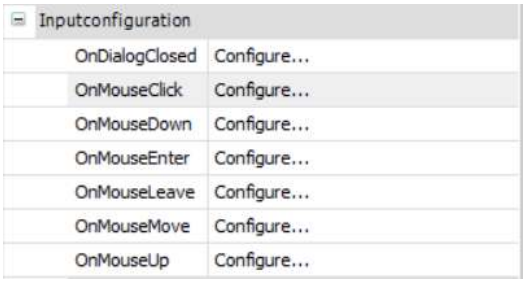
Running result as below:

Pic 0-8

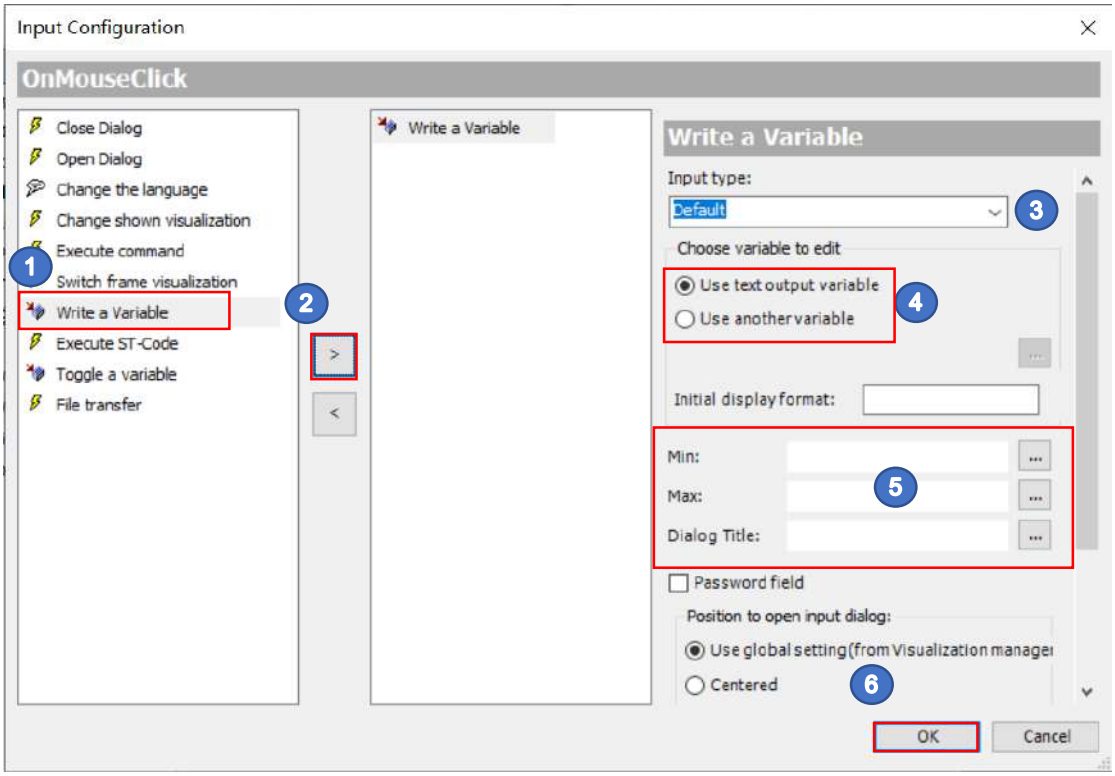
2. InputConfiguration

Text display is needed first for user to check entered variable.So, set Variable Mapping and Display Type first and then set trigger event, click Configuration in Input Configuration of properties.

Pic 0-9



Single click Configuration, pop up dialogue of Input Configuration, operate as below steps.



① Select trigger type, then Write a Variable

② Add to right side via rightward arrow in picture.

③ Enter input type through keyboard or virtual keyboard.

Pic 0-10

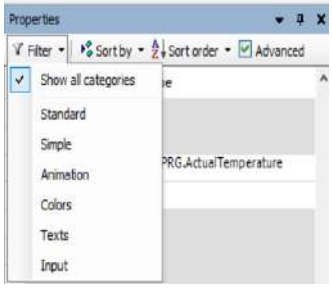
④ Select related input variable, like variable of current output display and other related ones, to decide content of text box.

⑤ For safety, designer can enter limitations of input value here.

⑥ Click OK to finish setting.

Notice:

When rectangle is selected and user can't find "On Mouse Click" in menu of Input Configuration, please select Display All Types in Filter at top of tree menu.



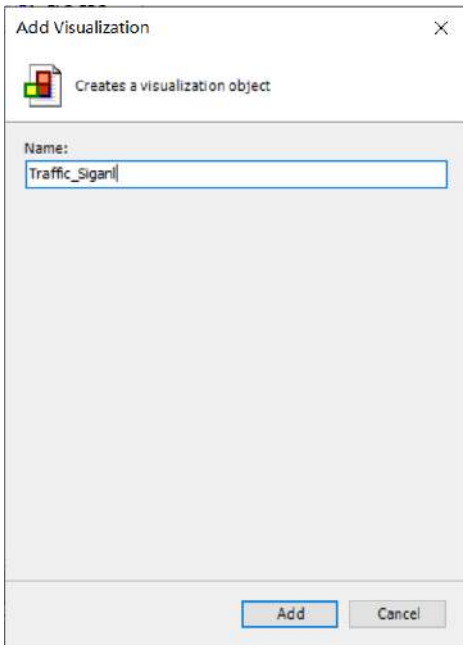
Color Display Box

Basic tools can not only act for text input and display, but also as simple indicator of color display. Color change can be realized through changing filled color of color display box, when corresponding variable program is on status of ON.

【例1】 Create a color display box to imitate traffic lights of cross and change color every 2s.

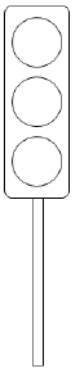
"Traffic_Signal" Create a visualization interface and name it Traffic_Signal.

Pic 0-11



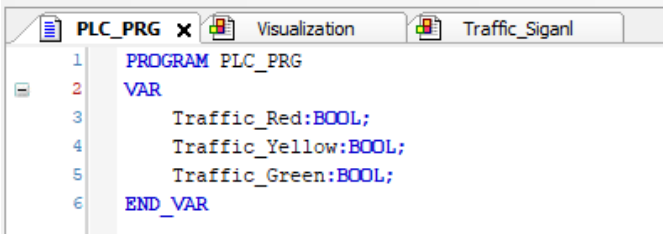
Create graphic of traffic lights with Rectangle, Rounded Rectangle and Oval and place in order as right.

Pic 0-12



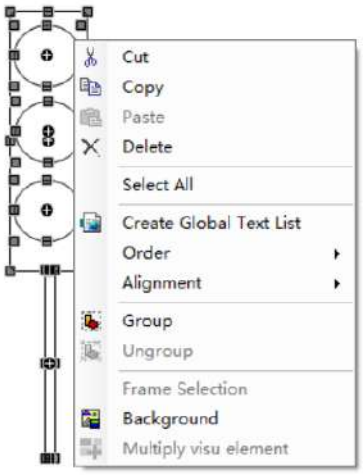
Edit color variable in program.

Pic 0-13



Select all control and right click, select Group for easier movement and size adjustment.

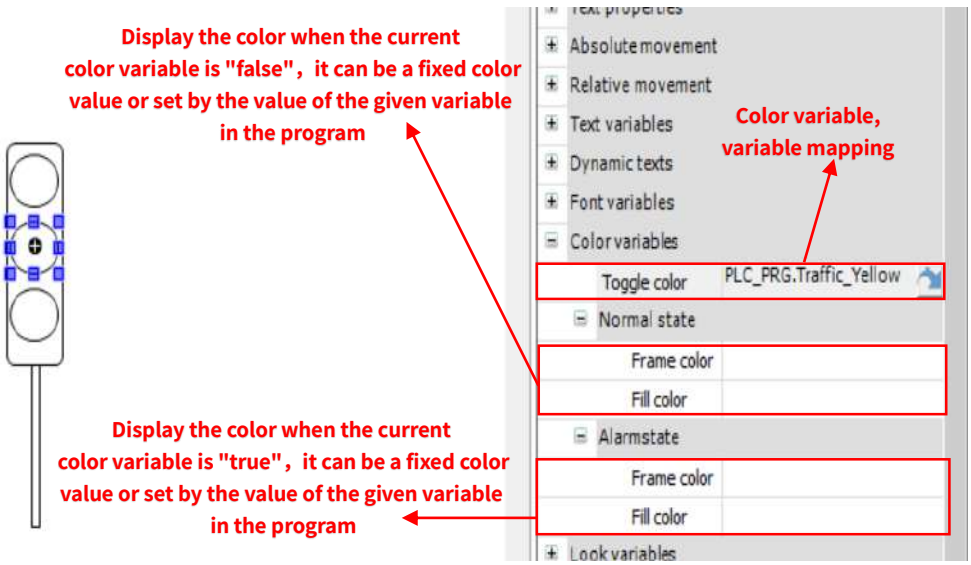
Pic 0-14



Complete Variable Mapping

Set Color Variables in properties menu, single click Input Helper logo to change related mapping. Then box and filling color will be triggered when color variable changes, and it will act as indicator.

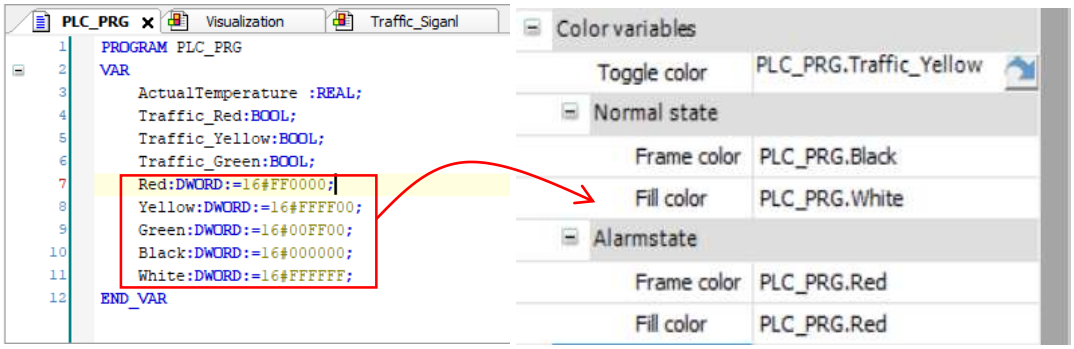
Pic 0-15



Color of indicator in status of ON

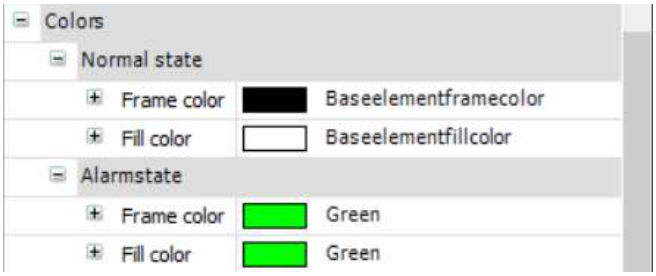
User can complete variable mapping as above through setting variable in program, to change color in different status.Color will change based on appointed variable value(RGB value of corresponding color, hexadecimal in default), which provides user with flexible setting.

Pic 0-16



Also easier setting of color through Color of properties is optional.

Pic 0-17



After above settings, color change online will be same as appointed RGB value.

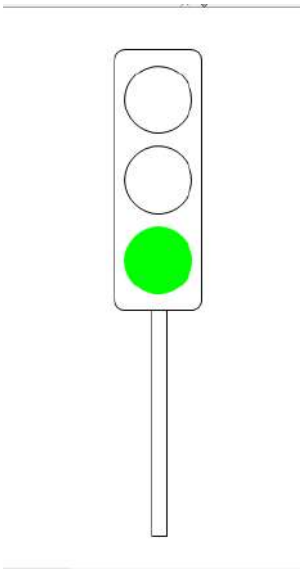
Programming example:

Pic 0-18

```
Ton1(IN:= NOT Ton1.Q , PT:=T#2S ,);
CASE Index OF
1:
Traffic_Red:=TRUE;
Traffic_Yellow:=FALSE;
Traffic_Green:=FALSE;
IF ton1.Q THEN
    Index:=2;
END_IF
2:
Traffic_Yellow:=TRUE;
Traffic_Red:=FALSE;
Traffic_Green:=FALSE;
IF ton1.Q THEN
    Index:=3;
END_IF
3:
Traffic_Red:=FALSE;
Traffic_Yellow:=FALSE;
Traffic_Green:=TRUE;
IF ton1.Q THEN
    Index:=1;
END_IF
END_CASE
```

Running result:(one of the indicators is ON)

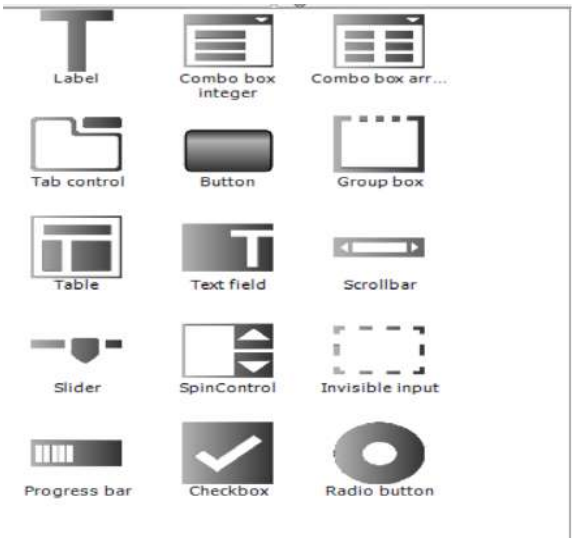
Pic 0-19



4.4.2 Common Control Tools

It is mainly consisted of normal graphic elements, which can be used to make label, box combination, diagram and button, etc. Details as below.

Pic 0-20

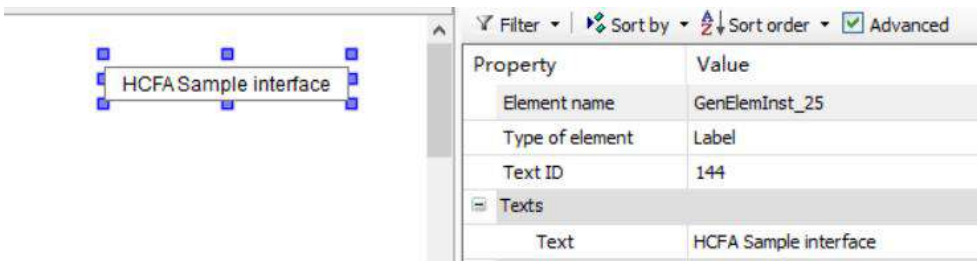


Label

Mainly used in text which can't be edited and labeling objects in window(description for example)

Find label in Tool Box and add it. Set Texts property in interface directly and it will then display in visualization interface, which can't be changed after login.

Pic 0-21



Find State Variables→Invisible in controls of tool box, which is used to add status variable or display, hide control. TRUE represents invisible and FALSE visible.

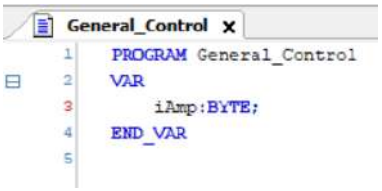
Combo Box Integer

Combo box is divided into combo box integer and combo box. User can select needed integer in drop down list in combo box integer and the data will be written in variable.

- Create Integer Data

Create POU program with name of General_Control and programming as below:

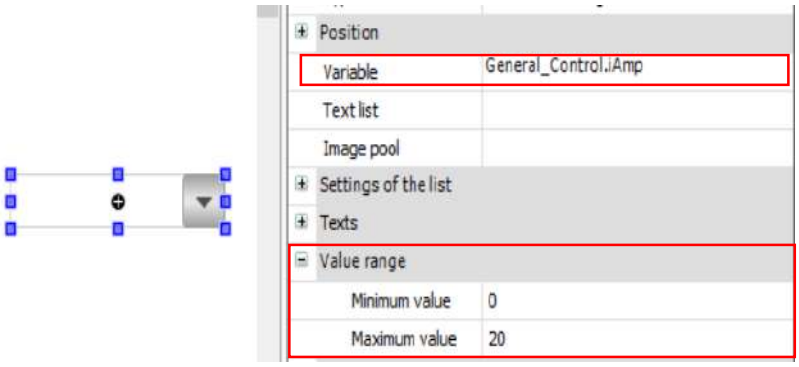
Pic 0-22



- Create Combo Box Integer

Find ToolBox in window and Common control, add combo box integer, complete mapping through Variable in menu bar to finish. Also range of input can be set through Sub-range, select Max Value and Mini Value.

Pic 0-23



Online running result as below:

Pic 0-24

ComboBox Table

Like combo box integer, drop down list is provided, but data object of data array is optional here.

Example 2: Create a two dimensional array and write it to program in corresponding line of visualization.

Create array data and assign it with initial value:

Pic 0-25

```
    iFactor:BYTE;
    arrFactor: ARRAY[0..2, 0..4] OF STRING := ['BMW', 'Audi', 'Mercedes',
 'VM', 'Fiat', 4('150'), '100', 'blau', 'grau', 'silber', 'bronze', 'rot'];
END_VAR
```

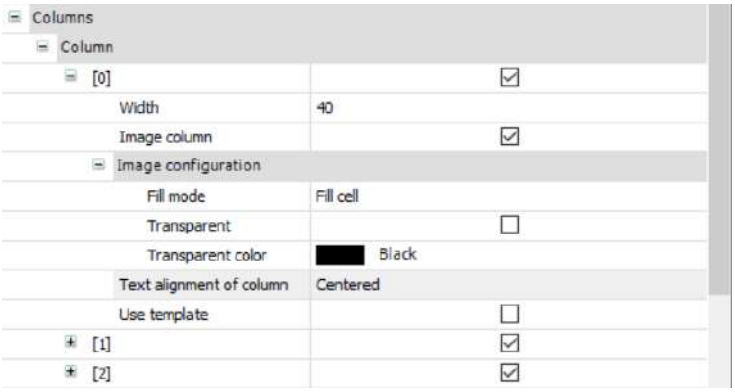
In ToolBox, find Common controls, add Combo box array , set mapping between Variable, Data array and program variable through properties menu.

Pic 0-26

Variable	General_Control.iFactor
Data array	General_Control.arrFactor

Column in properties menu can set numbers of columns based on linked table and display properties of each column.

Pic 0-27



After above steps, for easier switch of combo box information, user can add Combo Box Integer in visualization and map iFactor to the control. Then final display as below:

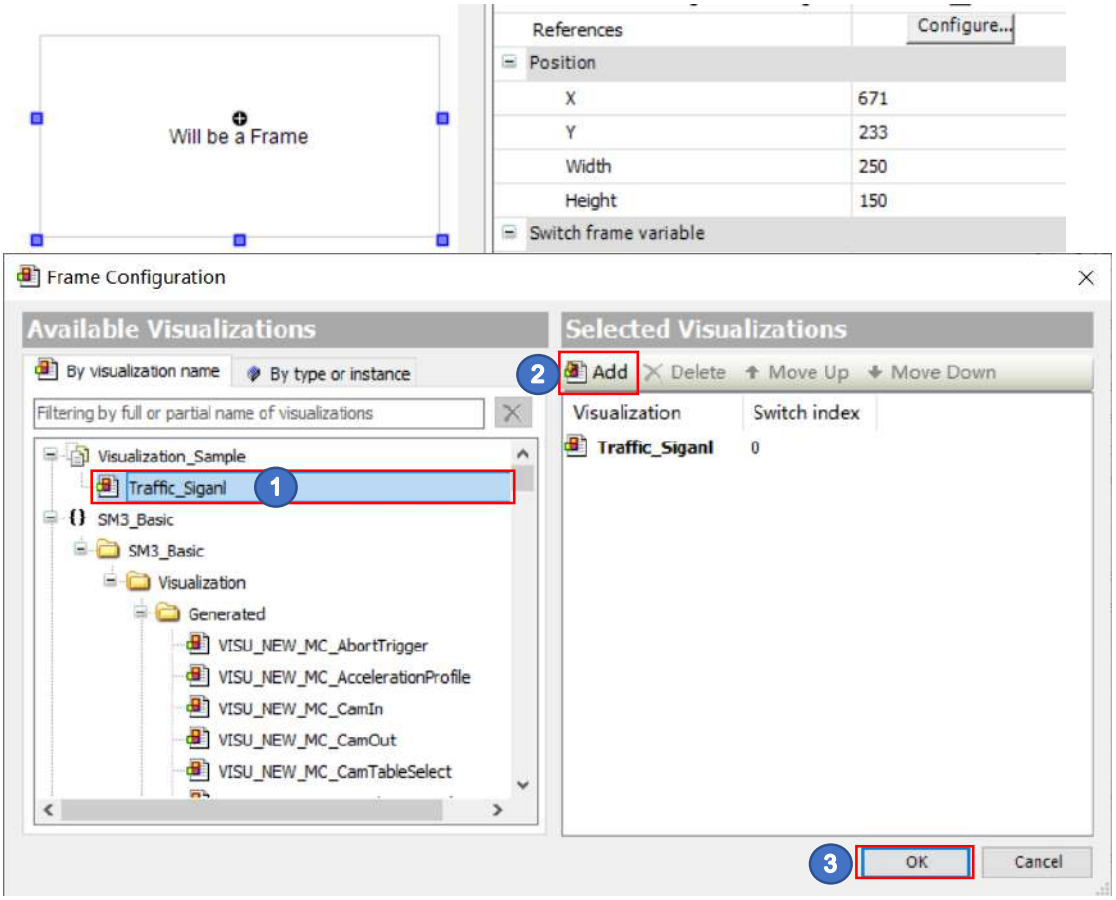
Pic 0-28

Tap Control

It is box which defines part of current visualization and could also contains some other visualization interface, swich in online mode. Through proper configuration,a certain visualization interface will display in box of each related area.

Add Tap Control to edit window of visualization and dialogue of Frame Configuration will pop up, or in References, double click to call Frame Configuration. All available visualization is listed in the

interface and add one to box.



In Tap Control properties menu, select References to modify Heading or Image ID the graphic.

Pic 0-29

Through Scaling Type in properties menu, set the scaling method of Tap Control.

Pic 0-30

Appoint rules for size change of box, through Scaling Type:

Isotropic: keep original image scale in any circumstance.

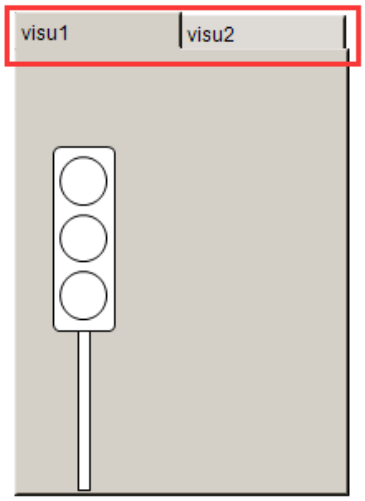
Anisotropic: Each parameter can be modified separately as the box size will change accordingly.

Fixed: Size of original image will not change in any circumstance.

Fixed and scrollable: Image will not zoom and a scroll bar will be added to box automatically to display visible area of image, if it's bigger than the box. Use property of horizontal or vertical scroll bar position to change variable of the bar.

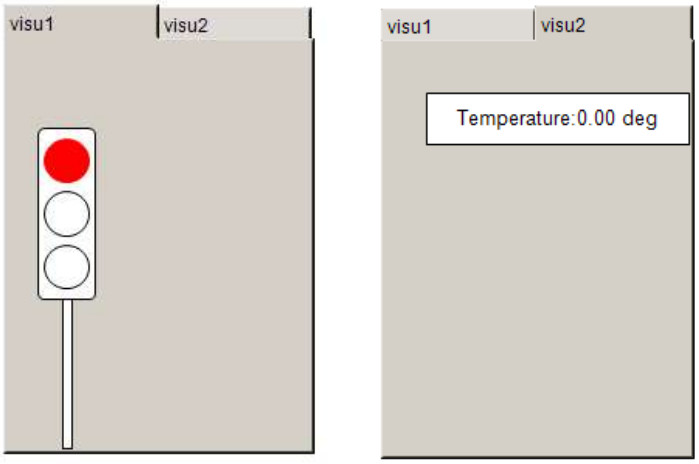
When multi visual objects exist in on Tap Control, user can process switching through notes on top.

Pic 0-31



Running result as below

Pic 0-32



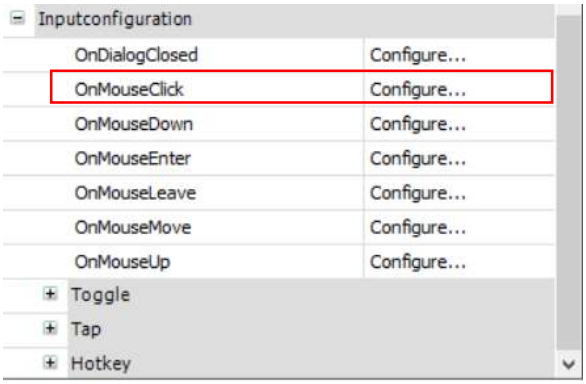
Button

Single click to execute Button and both text and image can be added to button. Set trigger method through property of control.

Example 4: Create a button and single click to execute ST code.

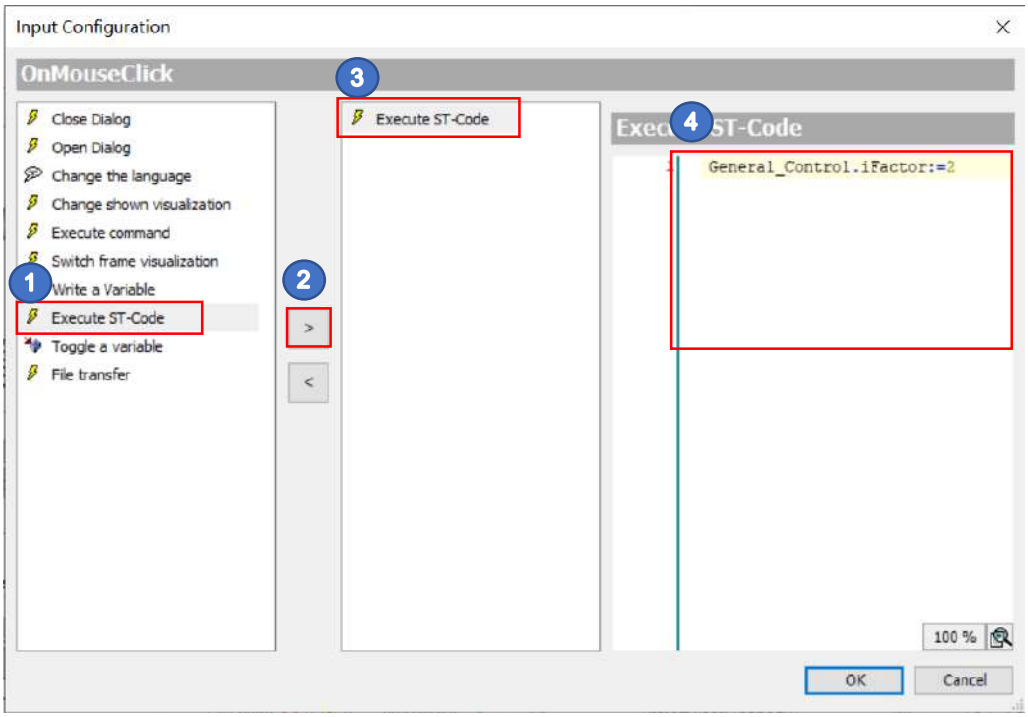
Find Common Control in tool box, find and add to view edit window. Set triggering action as OnMouseClick in Inputconfiguration, which means trigger is single click on the control.

Pic 0-33



Single click Input configuration, select and add Execute ST-Code to right by > . Variable in ST-code is global variable and please add naming space when calling local variable, or it can't be builded.

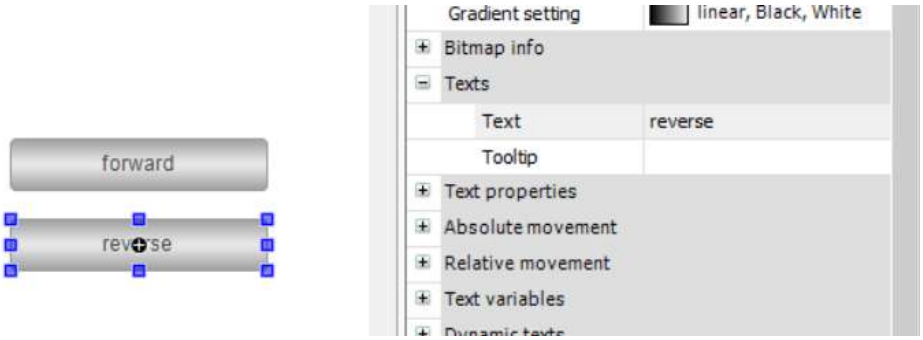
Pic 0-34



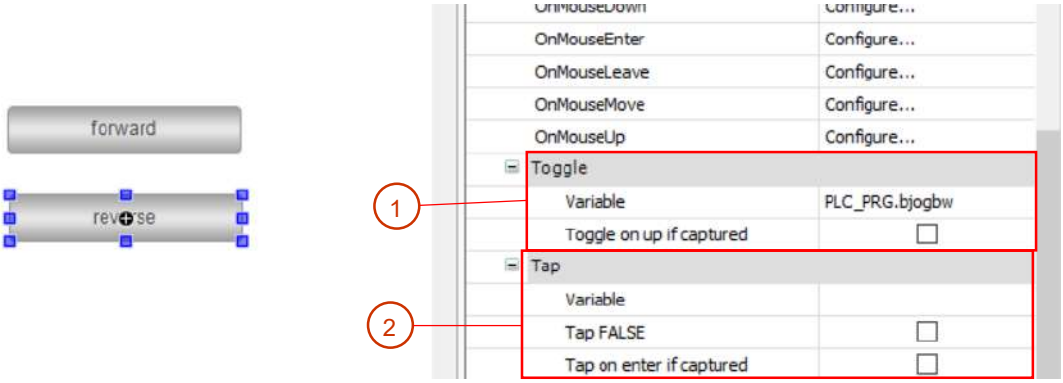
Example 5: Create two buttons in visual interface as forward and reverse of motor. Press and release forward button for forward rotation or to stop. Same for reverse button.

Add two buttons to visual edit area from tool bar and enter text Forward Rotation and Reverse Rotation as label.

Pic 0-35



For the press and release operation, find Input Configuration in properties and select entering method of Toggle, complete mapping in Variable with input helper.



Two ways of trigger are Tap and Toggle.

① Tap，Press the button to trigger and variable is ON, release to stop and variable is FALSE.

② Toggle，Press the button and variable keeps as ON, and manual trigger is needed to make it FALSE. It's the same in Toggle→Variable mapping. Tick FALSE if it's needed in FALSE when pressing the button.

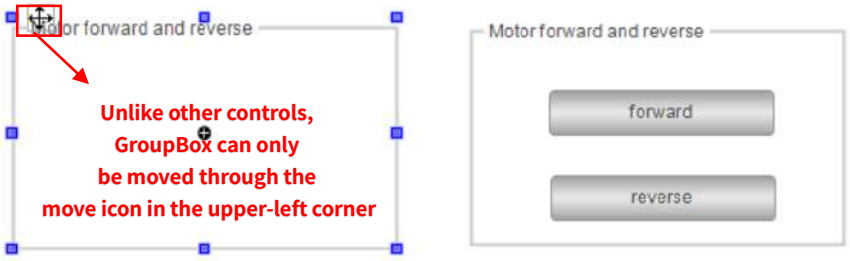
Group Box

It's used to provide group for other tools and to subdivide visual function. Boxes will be provided to display title, while it can't use scroll bar.

Find Common Control in tool box, select and add

to visual edit window, title can be changed in Text of properties menu. Then drag tools into the group box to realize grouping of visual element.

Pic 0-36



Text Field

Text can display in the field by entering in properties menu or through Text Variable. Refer to configuration instruction of rectangle as they are alike. Logo of Text Field is

.

Scroll Bar

Find Common Control in tool box, select and add

to visual edit window. It's horizontal display

in default. If its height is bigger than width after adjustment by mouse, it will then display in vertical.

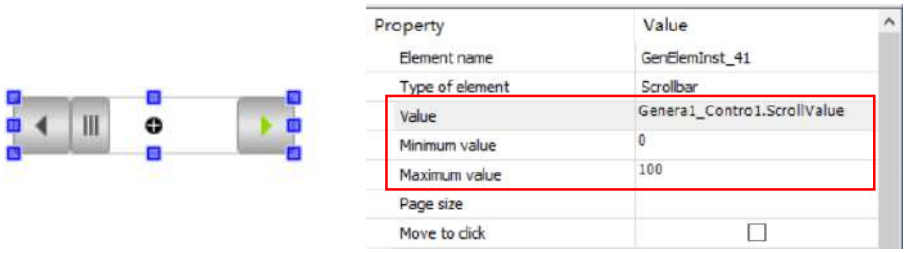
Add below variables to program:

Pic 0-37

```
PROGRAM General_Control
VAR
    ScrollValue:LREAL;
END_VAR
```

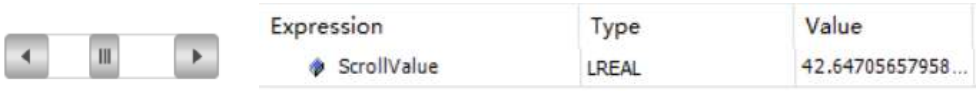
Complete mapping through Value of properties and set Maximum Value, Minimum Value.

Pic 0-38



Online display as below. Mapped variable value will change with the scroll bar position:

Pic 0-39



Slider

Alike scroll bar, drag the slider to change mapped variable value. Find Common Control in ToolBox, select and add to visual edit window.

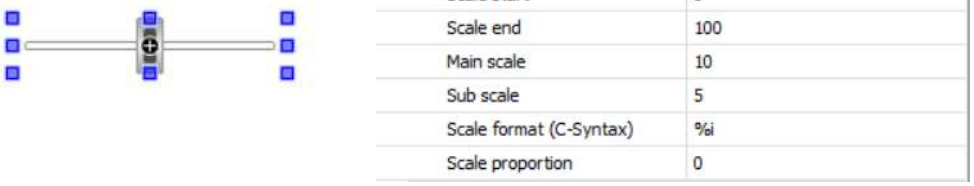
Add below variables to program:

Pic 0-40

```
General_Control
PROGRAM General_Control
VAR
    SliderValue:LREAL;
END_VAR
```

Select Slider control and complete mapping through Variable, set its Scale start and Scale end, or Main/Sub scale. Details as below:

Pic 0-41



Online display as below. Mapped variable value changes with slider position.:

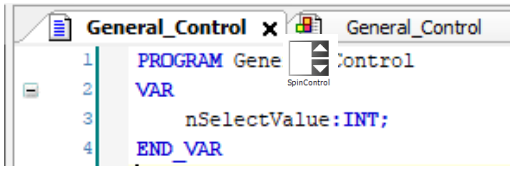
Pic 0-42

Spin Control

As a control of display and input spin, it provides upward and downward arrow to adjust mapping value spin, and direct entering is also optional. Set maximum value and input value replace it when input one is bigger. Same for minimum value.

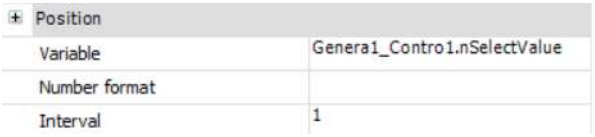
Add below variables to program:

Pic 0-43



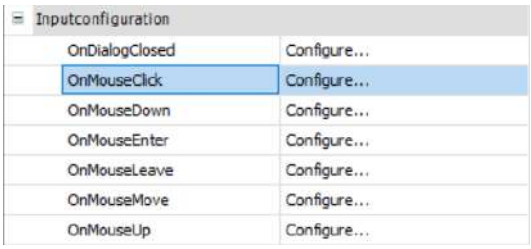
Find Common Control in ToolBox, select and add spin control, complete mapping through Variable. Space between spin can be adjusted through the arrows.

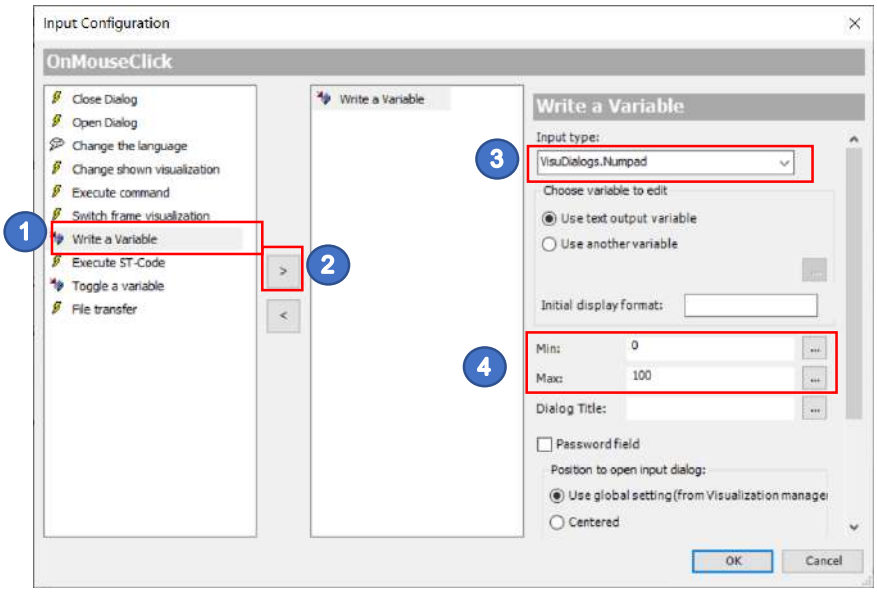
Pic 0-44



Select On Mouse Click in Inputconfiguration, single click Configuration to enter value and set limitation.

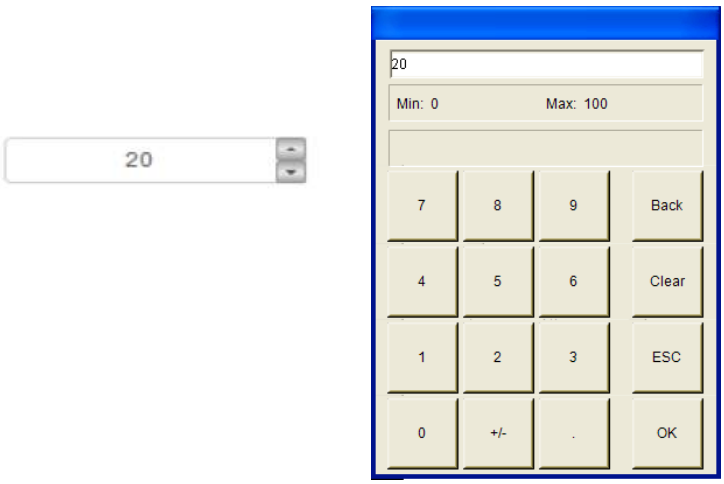
Pic 0-45





Online running as below:

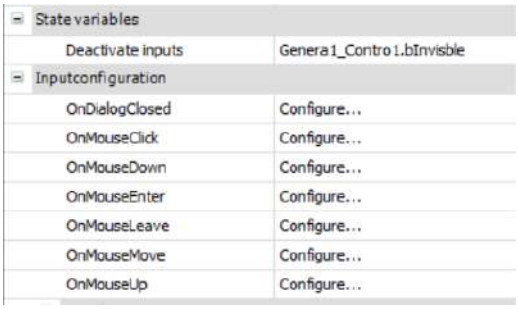
Pic 0-46



Invisible Input

The element is dotted rectangle box after added and will hide on online mode. Actions of it can be set in Input Configuration.

First add from State Variables



Progress Bar

With it, current process variable can be mapped to control and maximum and minimum value can be set. Current process will be displayed according to current mapping variable.

Add self accumulation program via the program to imitate process of bar increasing, programming as below:

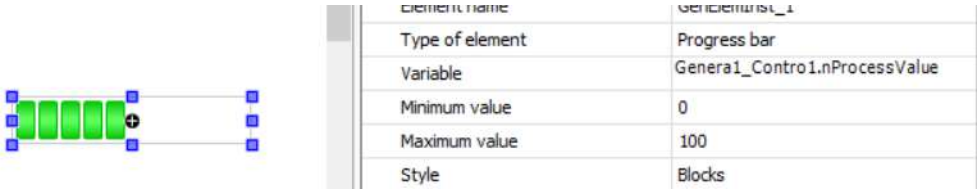
Pic 0-47

```
    nProcessValue:INT;
    ton1:TON;
END_VAR


ton1(IN:=NOT ton1.Q ,PT:=T#1S ,);
IF nProcessValue<100  AND ton1.Q THEN
    nProcessValue:=nProcessValue+1;
ELSIF nProcessValue=100  AND ton1.Q THEN
    nProcessValue:=0;
END_IF
```

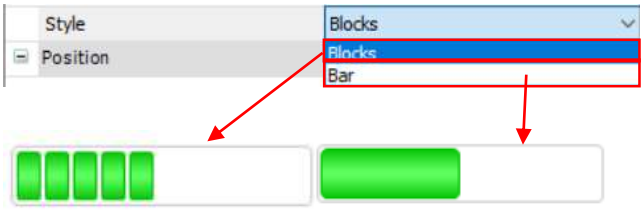
Find General Control in tool box, add process bar control to visual edit window, then complete mapping via Variable. Also, Maximum Value and Minimum can be set.

Pic 0-48



Two display modes of process bar can also be set in Type of properties of menu.

Pic 0-49



The bar will increase slowly with variable change in online mode:

Pic 0-50

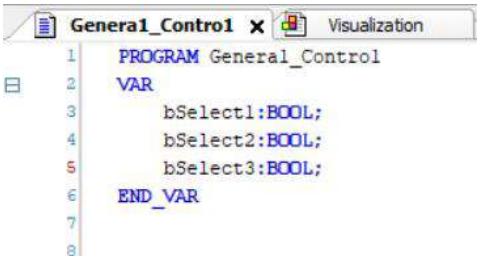
Checkbox

It is often used to provide options like have/true or false

Basic usage of checkbox:

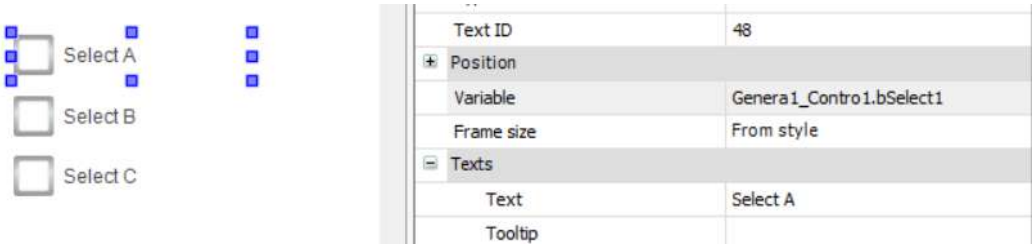
Find General Control in tool box, add checkbox . Create needed variable in program to provide checkbox options.

Pic 0-51



Select checkbox in edit area of visual view and create map between Variable and program, then checkbox options will be written to BOOL variable. Meanwhile, enter description of the option in text box. Map variables of three checkbox one by one.

Pic 0-52



Online running as below:

Pic 0-53

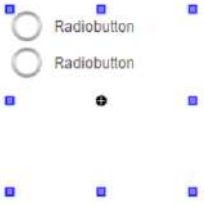
Radio Button

Unlike checkbox, it is integration of two or more mutually exclusive options. When one option is selected, other options of the integration can't be chosen.

Basic usage:

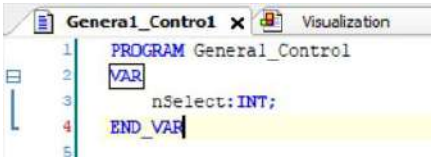
Find General Control in tool box, add radio button.

Pic 0-54



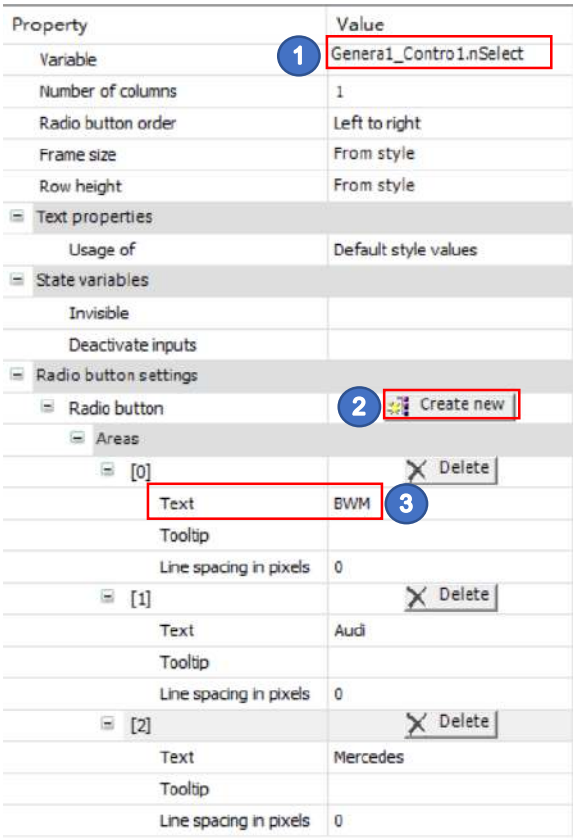
Add example variable in program window:

Pic 0-55



Complete mapping through Variable and confirm numbers of options. Single click Create New under Radio Button to needed amount. Then enter Text as label:

Pic 0-56



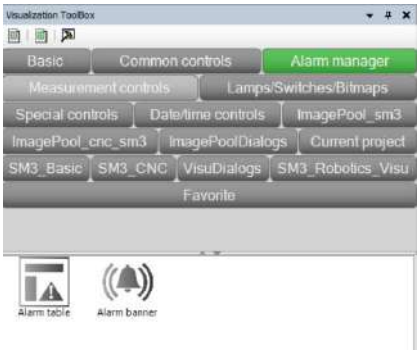
Check online running, only one option can be selected among three at the same time.

Pic 0-57

4.4.3 Alarm Manager

It mainly contains alarm diagram and label.

Pic 0-58



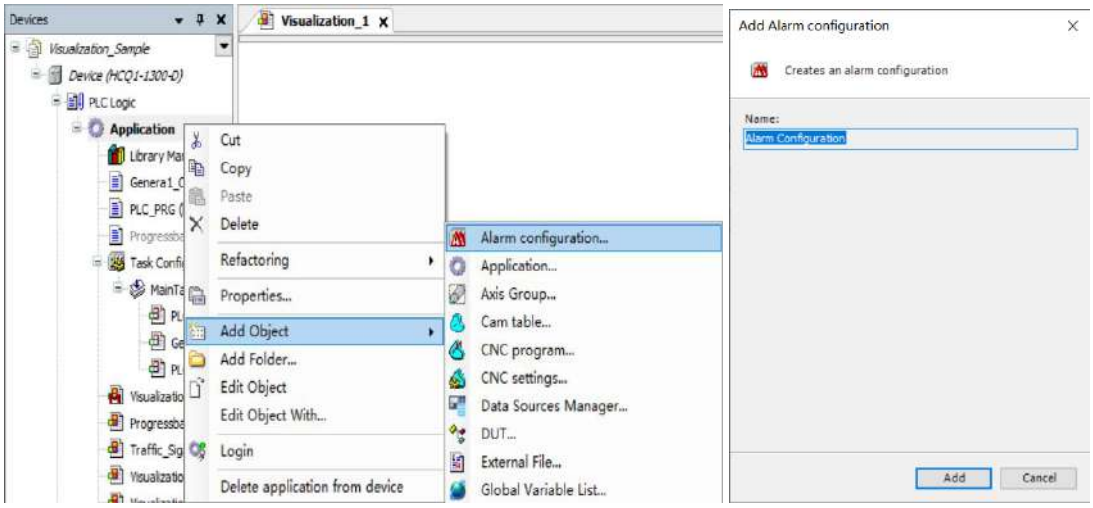
Alarm table

User can define visual alarm with pre-set configuration. Find Alarm Managerin ToolBox, drag alarm table to visual edit window. Configuration: set alarm configuration in Application; Set configuration of alarm table or banner.

- Add configuration in Application

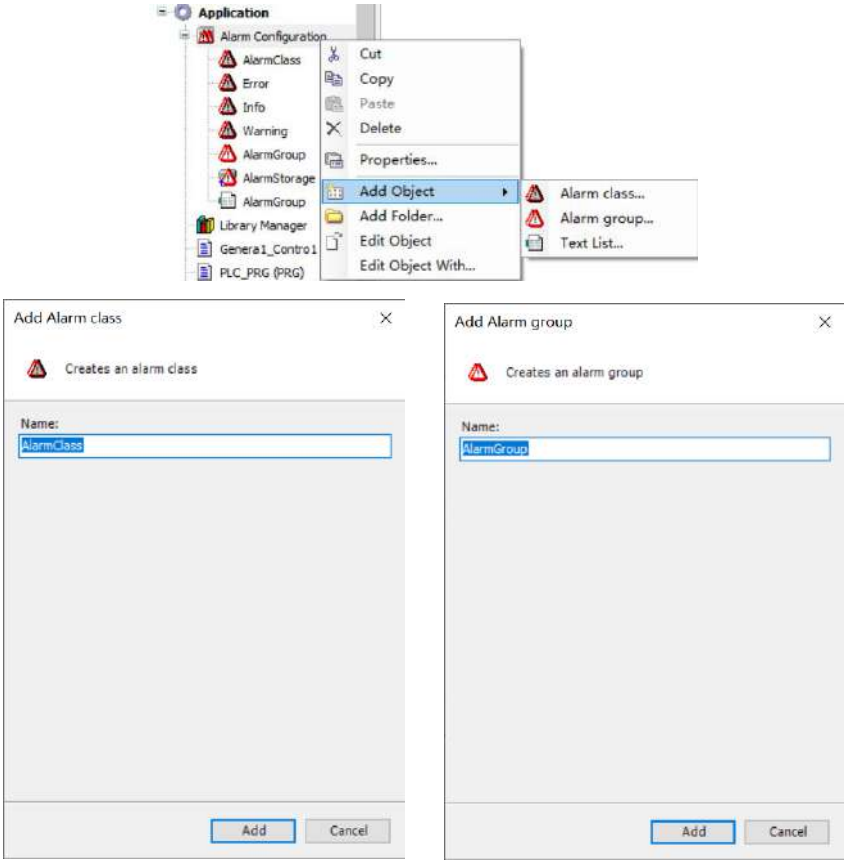
Single click Application, select Add Object, Alarm Configuration, single click and enter configuration name in dialogue, single click Add.

Pic 0-59



Alarm content and trigger should be set in Alarm Configuration. Single right click, find Add Object, add Alarm Class, Alarm Group, name before open via single click.

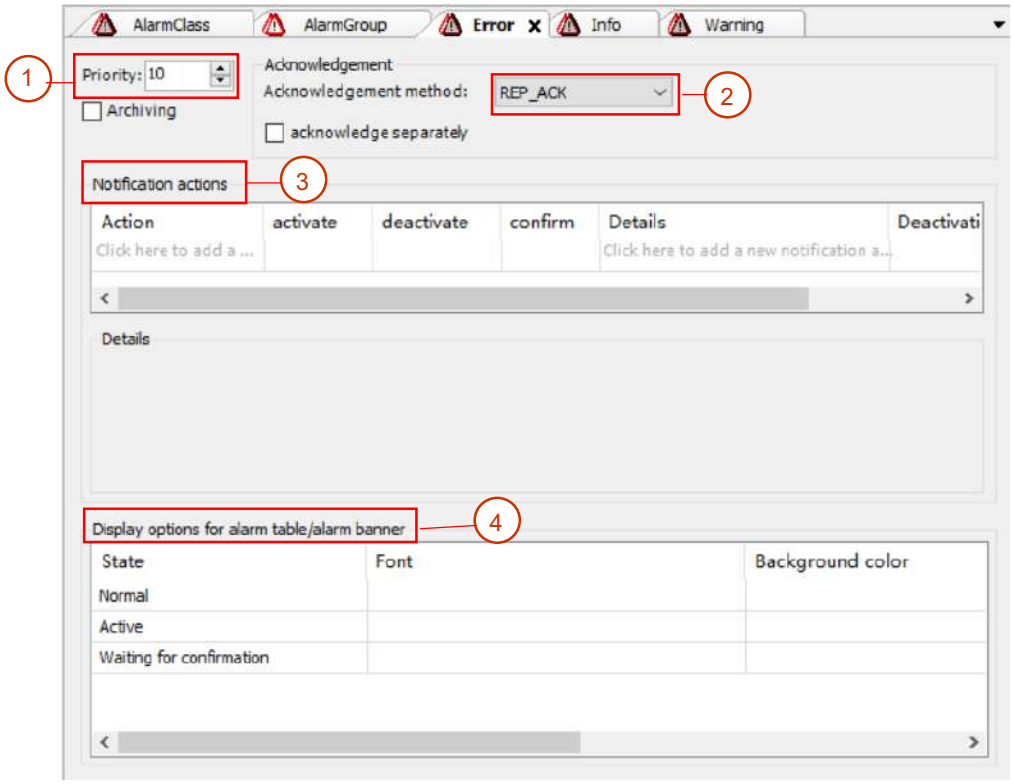
Pic 0-60



- Alarm Class

Alarm class is divided into 3 types in default: Error, Info, Warning. Main differences between them are priority and acknowledgement method. Configuration interface is as below.

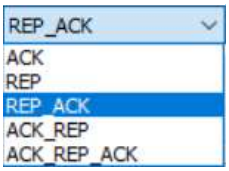
Pic 0-61



① Define priority of all needed alarm from 0~255. Smaller number means higher priority, which means 0 the highest. Immediate acknowledgement is needed for middle and high level, few requirements on low level alarm. For alarm which has lost trigger condition(for example, high temperature of motor has been back to normal), the alarm will not be canceled until acknowledge from user.

② User/system has to acknowledge alarm, which doesn't mean it's settled or back to normal. Sometimes it can get back to normal without external interfere. Some acknowledge methods as below:

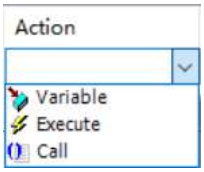
Pic 0-62



REP_ACK: Alarm doesn't activate after repair and acknowledge
ACK: Acknowledge
REP: Remove the problem and not alarm
ACK_REP: Not alarm after acknowledge and repair
ACK_REP_ACK: Not alarm after receive, repair and acknowledge.

③ It contains Action, Activate, Acknowledge, Detail, Stop, among which action provides options of variable, execute, call; Call or not with tick; Acknowledge before calling an action or not; Detail for mapping variable of notification actions; Stop to notify whether the action should be activated. Instruction of three actions as below:

Pic 0-63

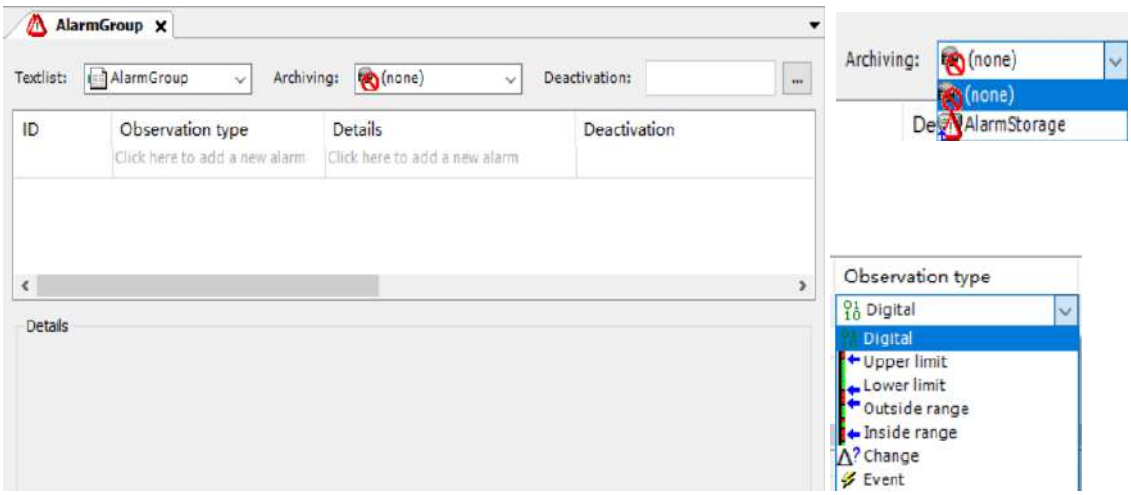


Variable: Select Variable and set variable or express for the alarm in dialogue
Execute: Enter name of Execute File on current alarm and call any parameter in Detail directly
Call: Enter name of needed FB to call.

- Alarm Group

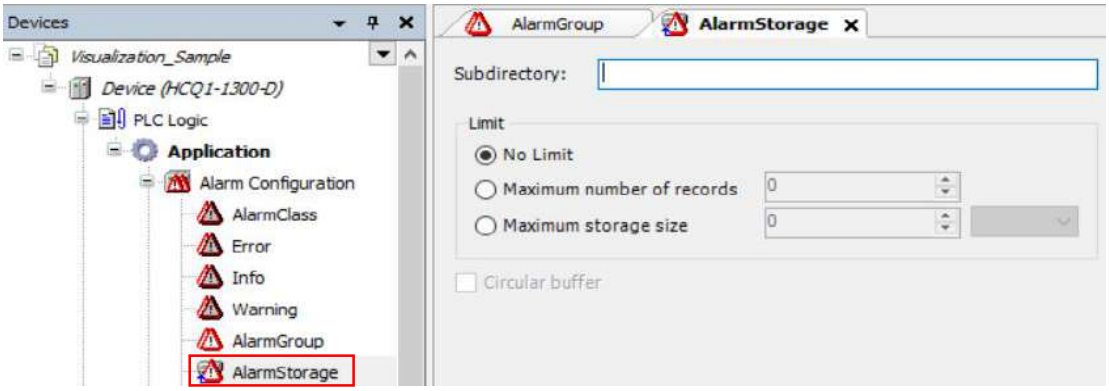
Set Observation Type, which means trigger type in Alarm Group, and also for file of alarm:

Pic 0-64



Find Alarm Storage in Alarm Configuration, set path and limit to activate Alarm Storage.

Pic 0-65



Trigger types as below:

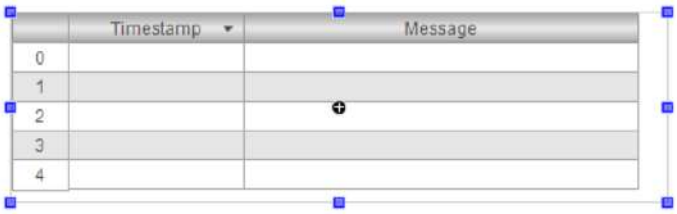
Pic 0-66

Type	Instruction
Digital	Enter monitored express on left and express on right for check, select wanted operation mark (=or<>)
Upper limit	Like Digital, but selectively use "lagging%" for comparative marks >or>=
Lower limit	Like Digital, but selectively use "lagging%" for comparative marks <or<=
Inside range	Enter monitored express. Area: alarm when express reaches value of internal range. Enter express on left for lower limit and upper limit on right. Monitored express displays in un-editable area.
Outside range	Enter monitored express. Area: alarm when express reaches value out of range. Enter express on left for lower limit and upper limit on right. Monitored express displays in un-editable area.
Change	Express: alarm when value of entered express changes.
Event	Trigger via application and use FUN of AlarmManager.library

- Set alarm in visual interface

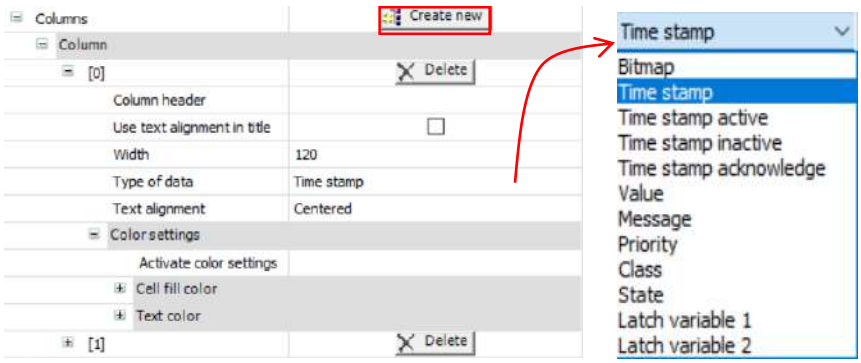
  In Alarm Management of tool box, drag Alarm Management to visual edit window.

Pic 0-67



Except for default two columns, user can add more. Find Column in properties menu to add and set title, width, text display mode, etc. Content of column is optional in drop down Type of data.

Pic 0-68



Time stamp ： alarm date andtime of last status change

Time stamp active：alarm date and time of last action

Time stamp inactive：last date and time of ineffective alarm

Time stamp acknowledge：date and time of last acknowledge

Value：value of current express

Message：value of monitor

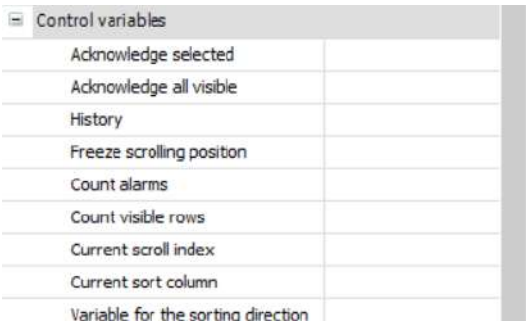
Priority：priority of alarm

Class：alarm class

State：alarm status

The alarms need to be confirmed by operator, and variables related to the confirmation action can be set in "Control variables"

Pic 0-69



Confirm selected variables：If variable is TRUE, the alarm selected in alarm table will be acknowledged

Acknowledge all visible：If this variable is TRUE, all alarms in table will be acknowledged

History：If the variable is TRUE, the alarm table will be converted to history mode. This means that all alarms will be sorted in descending order by date. Any new events will be added to current table.

Freeze scrolling position：If this variable is TRUE, current position of the scroll bar will be locked even if a new alarm is activated in history mode. Otherwise, the scroll bar will jump to the first row of the alarm table.

Count alarms：current number of alarms as displayed will be stored as this variable

Count visible rows：set visible lines on the variable.

Current scroll index： jump to first quote to create line via the variable.

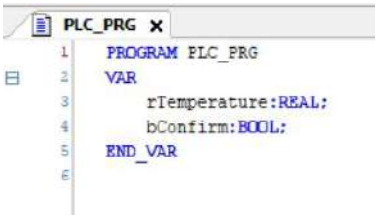
Current sort column：Sort warnings by this variable.

Variable for the sorting direction：define sorting direction via the variable, TRUE for ascending and FALSE for descending.

Example 1: Set an alarm temperature. Alarm when the temperature is higher than 50 ° or lower than 10 °.

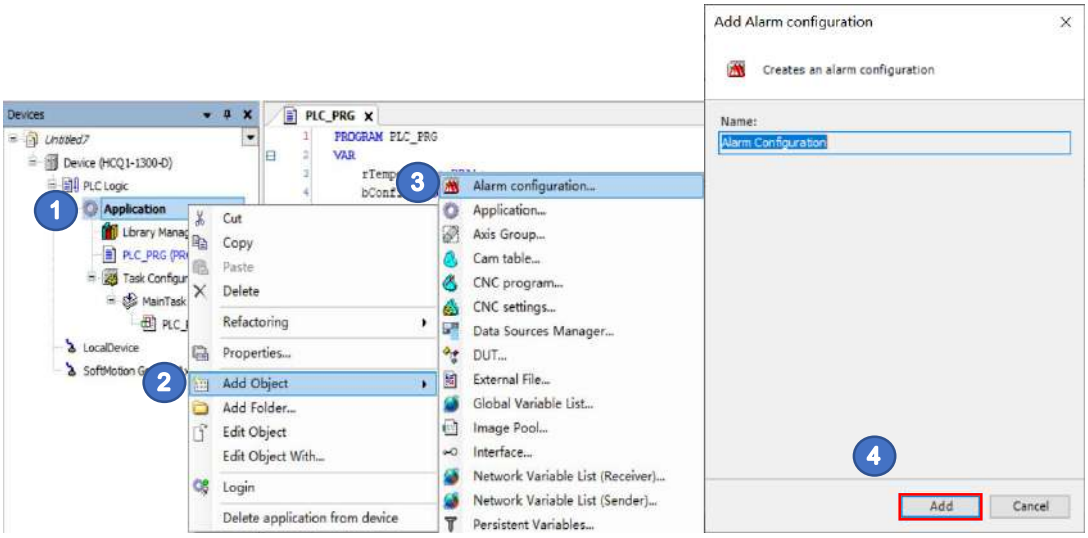
Add temperature variable rTemperature in program, the data type is real, and add bConfrim as confirmation signal.

Pic 0-70



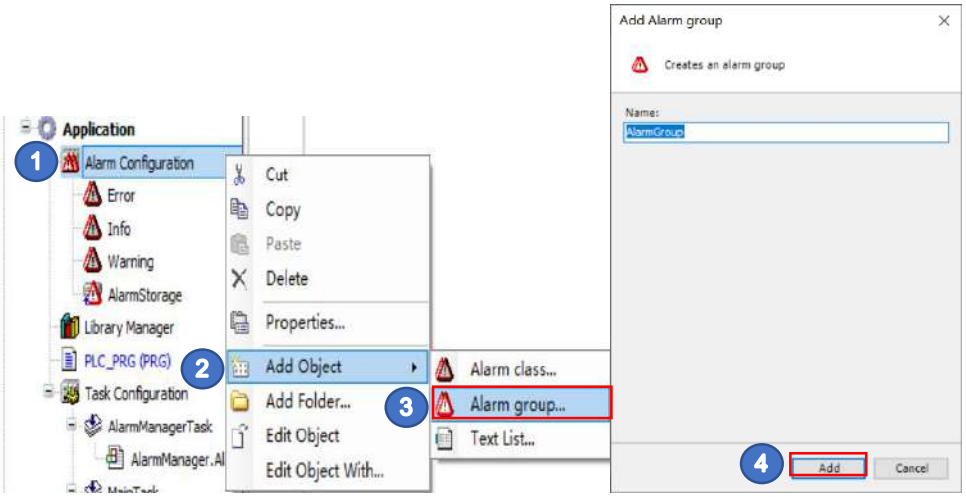
Right click "application" to add alarm configuration, name alarm configuration, click "open" to create a new one

Pic 0-71



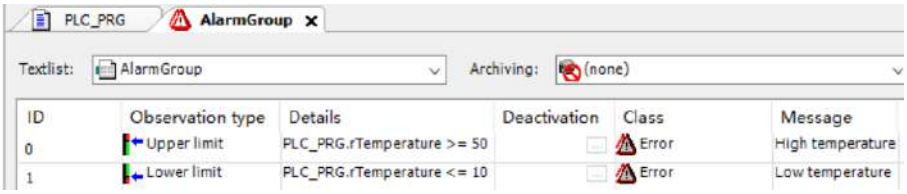
Add a new alarm group under the new alarm configuration, and right-click "Alarm Configuration"

Pic 0-72



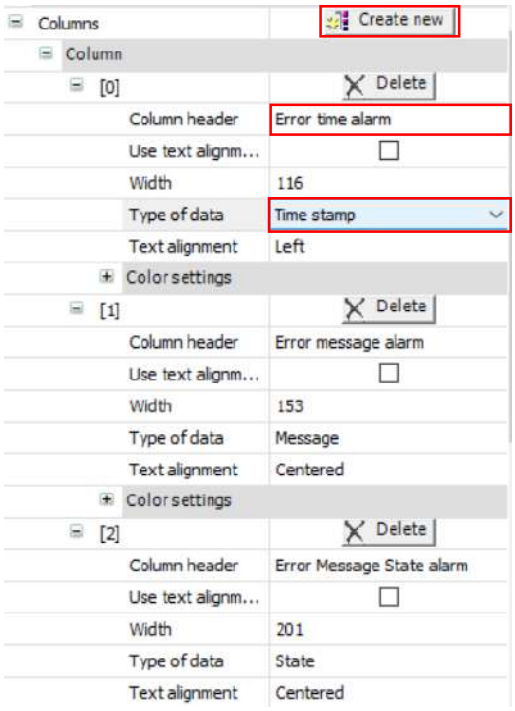
Content of alarm group is set as the trigger mode of upper and lower limit, and corresponding text alarm information is added in "Message".

Pic 0-73



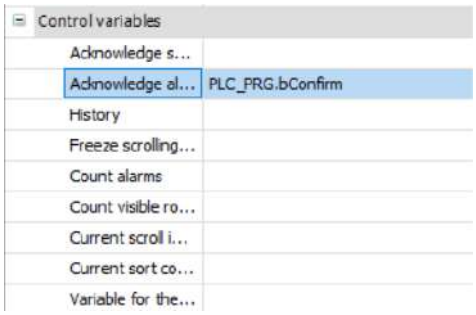
Add an alarm table in the visual interface and define column attributes. By default, the alarm table has only two columns and can be added. A new column can be added through "create new" button. View the column information by customizing column title. The content source is displayed by selecting the column in the drop-down menu of data type.

Pic 0-74



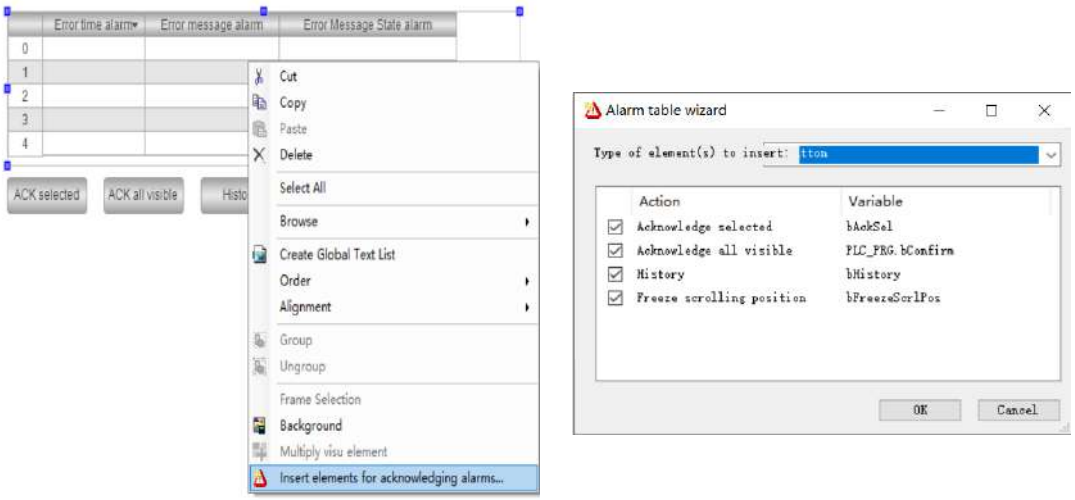
Find "control variables" and "bconfirm" in the mapping program as the confirmation signal in "bconfirm all visible variables"

Pic 0-75



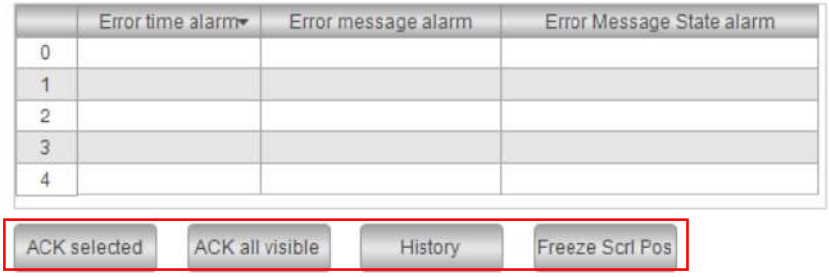
Add control variable Button and realize part of functions of four attribute buttons provided in default. Right click alarm table in the visual edit area to find "insert element through confirmed warning", and check the additional button to be added in the pop-up dialog box. The checked additional button will automatically map the variables added by the user in the "control variables". If the user does not add variables defined in the application in the control variables (such as plc_prg. Bconfirm), system will automatically create and input a local visual variable, such as backael, to confirm selection for example.

Pic 0-76



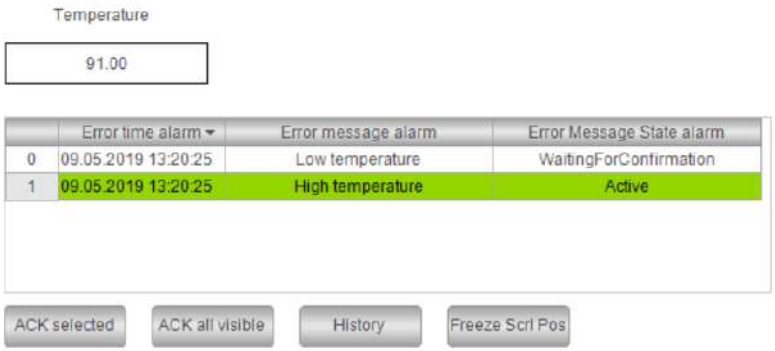
After clicking OK, corresponding additional keys will be added to the bottom of the table by default.

Pic 0-77



Check online running result:

Pic 0-78



Alarm Bar

Like alarm diagram but more simplified, it's used for visual configuration of a single alarm of alarm groups and classes. It belongs to the "alarm configuration" of special alarm categories.

Refer to that of alarm table configuration process.

4.4.4 Measurement Control Tools

It mainly contains common graphics like Display image bar, Dashboard display and Histogram, etc.

Pic 0-79

Bar Display

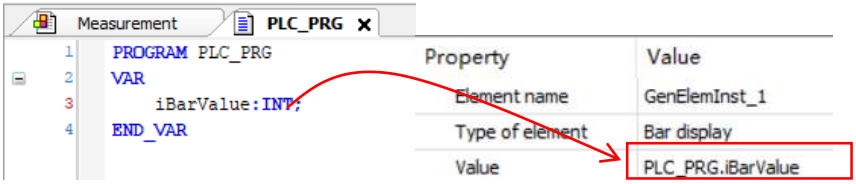
Also called bar graph, single click tool box, find Measure Control, select and add to visual interface. The control can generally be used to display the active values in a fixed interval, such as liquid level display, air pressure display, temperature display, etc.

Pic 0-79

Bar Display is used to display value of the mapped fixed variable and indicate changes of other values via histogram. Basic usage as below.

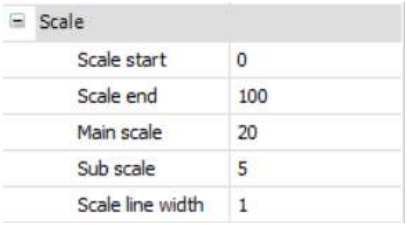
Drag needed Bar Display, in Measure Control, to visual edit area. In the "value" column of the attribute menu, map the variables in the program.

Pic 0-80



After completing variable mapping, set relevant parameters in the Scale, including "scale start", "scale end", "main scale" and "sub scale", etc.

Pic 0-81



After above, actual running as below:

Pic 0-82

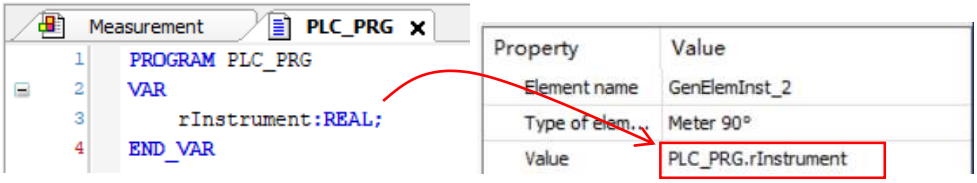
Dashboard (90°, 180°, 360°)

It's used to display value change of variable with fixed upper and lower limit, common to see in life, like display of oil and current speed. Find controls of "M 90 °," m 180 °, "m 360 °" in Measure Control, select any of them and add to visual interface.

Pic 0-83

Here is introduction of using dashboard control in visual edit area. Find instrument and add in the "measurement control" of the toolbox and drag it to the visual editing area. Complete mapping, select control in the visual editing area, map variables in program in "value" column of property menu.

Pic 0-84



Complete variable mapping, set scale format and find scale format under "label" in attribute menu. The scale format (c-syntax) refers to the format of defining scale label according to c-syntax. The default value is "%. 1F", where 1 before the decimal point means to output on the output device, and 1 after the decimal point means to keep one decimal place after rounding. "F" is an abbreviation for float, which means floating-point data.

Pic 0-85

Label	
Label	Inside
Unit	
Font	Times New Roman; 9
Scale format (C-Syntax)	%.1f
Max. text width of labels	34
Text height of labels	15
Font color	Fontcolor meter

Then set relevant parameters of scale, including "Scale start", "Scale end", "Main scale" and "Sub scale", etc.

Pic 0-86

Scale	
Sub scale position	Outside
Scale type	Lines
Scale start	0
Scale end	100
Main scale	20
Sub scale	5
Scale line width	1
Scale color	Scalecolor meter
Scale in 3D	☑

After above, actual running as below:

Pic 0-87

Device.Application.PLC_PRG

Expression	Type	Value
rInstrument	REAL	45

Histogram

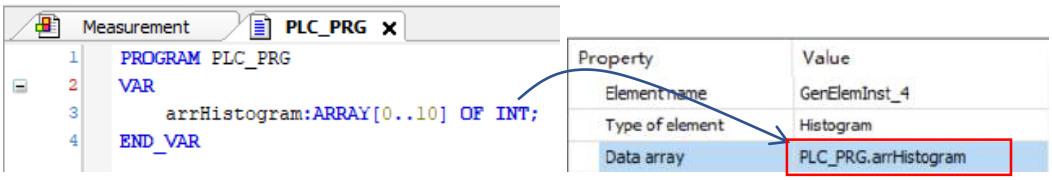
Also called mass distribution curve, it's one of the main tools in visualization interface. The distribution state of product quality characteristics can be seen intuitively with histogram, for better overall judgement.

Pic 0-89

Basic use as below:

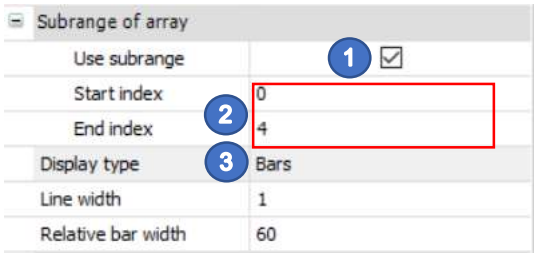
Find needed Histogram in Measure Control, drag it to visual edit area. Complete mapping first, map variables in the program in the "Data array" column of the property menu.

Pic 0-88



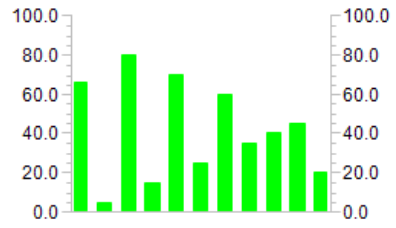
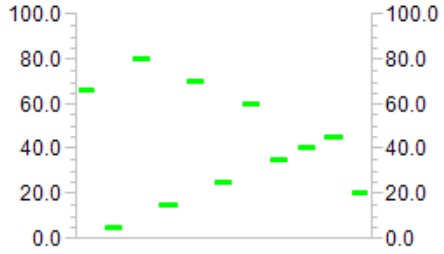
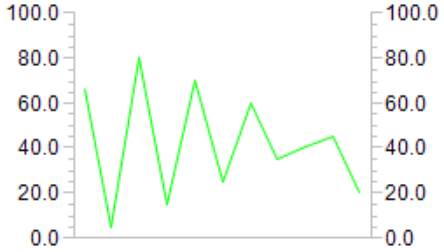
Then, select checkbox of Enable Sub-range in Usage Sub-range, modify Start Quote and Stop Quote to change numbers of variables here.

Pic 0-89



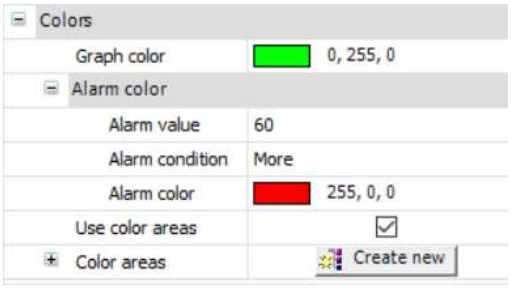
At last, select display type of graphic as Bar, Line or Curve:

Pic 0-90

Display type	Example
Bar	
Line	
Curve	

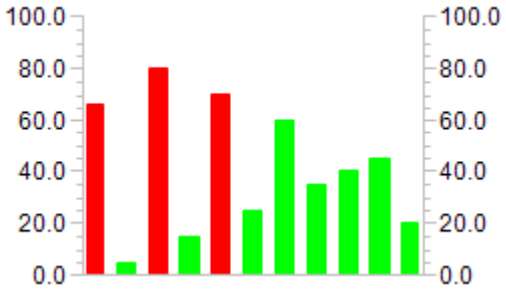
When value exceeds / is less than a set one, alarm color can be used. For example, when the value exceeds or is equal to 60, the alarm color can be enabled to display the histogram. Under "color" in attribute menu, the display color of the histogram in normal state can also be set.

Pic 0-91



Display as below:

Pic 0-92

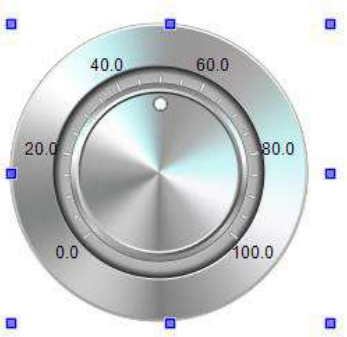


arrHistogram	ARRAY [0..10] OF INT	
arrHistogram[0]	INT	66
arrHistogram[1]	INT	5
arrHistogram[2]	INT	80
arrHistogram[3]	INT	15
arrHistogram[4]	INT	70
arrHistogram[5]	INT	25
arrHistogram[6]	INT	60
arrHistogram[7]	INT	35
arrHistogram[8]	INT	40
arrHistogram[9]	INT	45
arrHistogram[10]	INT	20

Potentiometer

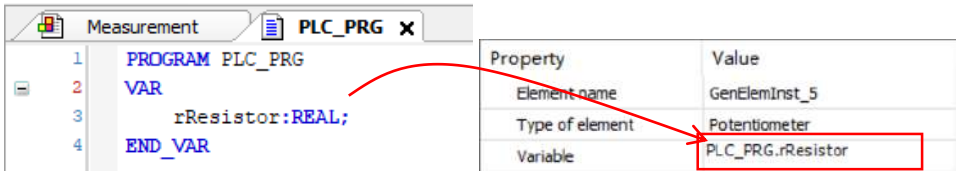
In life, when brush moves along the resistance body, potentiometer obtains the resistance value or voltage value in a certain relationship with the displacement at output end. In visual interface, it will be converted into the display of corresponding value based on current sliding position.

表 0-93



Then is introduction of how to use potentiometer control in visual edit area, example in resistance regulation. Find Measure Control in tool box, find and drag needed potentiometer to visual edit area. Complete mapping in Variable in property menu after the control is selected.

Pic 0-94



Set scale format after mapping, find Label and format under it. Scale format (c-syntax) refers to the

format of the scale label defined on c-syntax, in default value of “%.1f”, and 1 before point means output on device and 1 after point means to keep one decimal place after rounding. “f” is abbreviation of “float”, which means float type data.

Pic 0-95

Label	
Label	Inside
Unit	
Font	Times New Roman; 9
Scale format (C-Syntax)	%.1f
Max. text width of labels	34
Text height of labels	15
Font color	Fontcolor potentiometer

Then set related parameters in scale, including Scale Start, Scale End, Main Scale, Sub Scale, etc.

Pic 0-96

Scale	
Sub scale position	Outside
Scale type	Lines
Scale start	0
Scale end	100
Main scale	20
Sub scale	5
Scale line width	1
Scale color	Scalecolorpotentiometer
Scale in 3D	☑

User also needs to define angle value of the scale on potentiometer through "start arrow" and "end arrow" in menu, and complete definition directly by appointed angle value or variable. Valid variable here needs integer data to define the angle value. The zero position of the scale is from clockwise to the starting position of the scale, the zero position of the scale is "3 o'clock", and the angle range is from 0 ° to 360 °.

Pic 0-99

Arrow	
Arrow type	Circle
Color	255, 255, 255
Arrow start	220
Variable	PLC_PRG.iStart
Arrow end	-40
Variable	PLC_PRG.iEnd

```
    iStart:INT:=220;
    iEnd:INT:=-40;
END VAR
```

After above, actual running as below:

Pic 0-97

rResistor	REAL	33
iStart	INT	220
iEnd	INT	-40

4.4.5 Lamps/Switches/Bitmaps

Lamp, switch, bitmap tools provided in ToolBox, allows user to edit interface with proper lights and switches according to actual need.

Pic 0-98

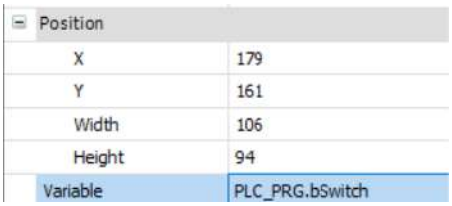
Image Switcher

Like BOOL variable, it can be open or closed with entered certain image. Click on image element, related variable will turn to True.

Steps: find Lamps/switches/bitmaps in ToolBox, select needed switch and drag to visual edit area.

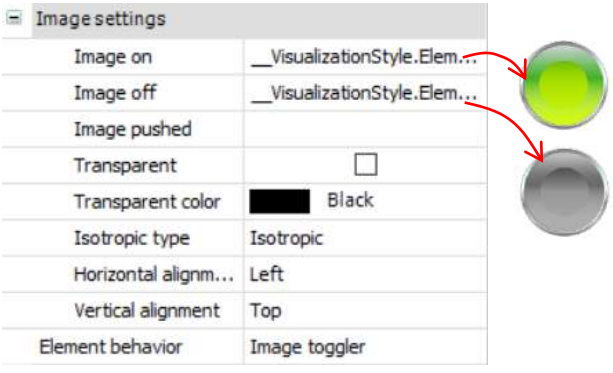
Complete mapping after adding “bSwitch” variable to main program.

Pic 0-99



In menu, add graphics corresponding to "Image on" and "Image off" in the attribute menu "Image settings", and map the variables "True" and "False" respectively.

Pic 0-100



Check online running, single click image switch can switch status of related variables:

Pic 0-101

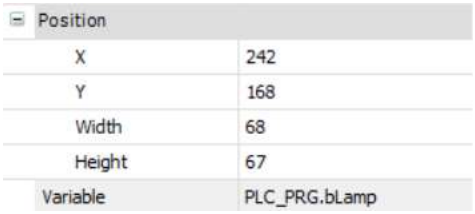
Light

Light will be on when linked to related variable and color of it can be selected in Background in background, menu.

Steps: find Lamps/switches/bitmaps in ToolBox, select needed light and drag to visual edit area.

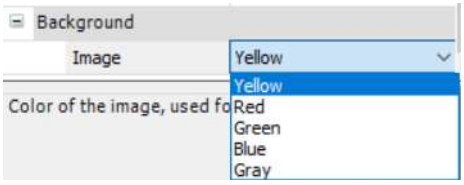
Complete mapping after adding “bLamp” variable to main program.

Pic 0-102



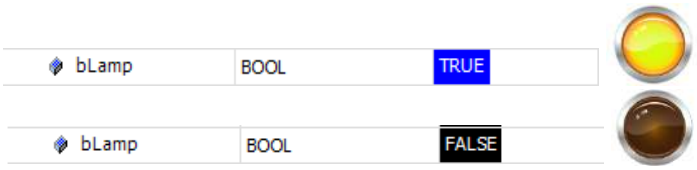
In attribute menu "background", find " image" drop-down to select the color of light.

Pic 0-103



Change status of “bLamp” and check online running:

Pic 0-104



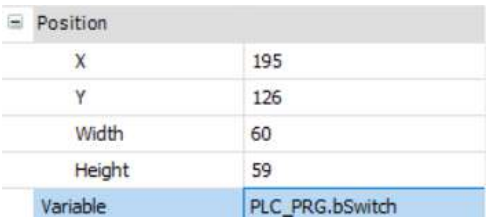
Dip / Power / Push / Push LED / Rocker / Rotary switch

There are many kinds of position contact switches with similar configurations. Therefore, choose one of them to introduce. Take the key switch LED as example:

Find find Lamps/switches/bitmaps in ToolBox, select and add needed Key Switch LED and drag to visual edit area.

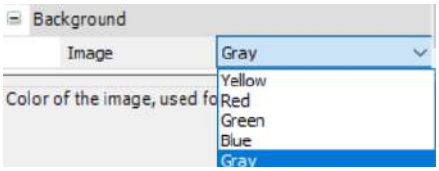
Complete mapping after adding “bSwitch” in main program.

Pic 0-105



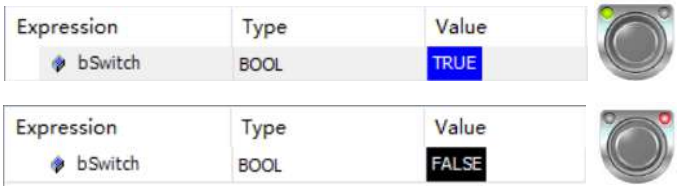
Set needed color in Background Image, Background according to need after mapping.

Pic 0-106



Single click Key switch LED to check online running:

Pic 0-107



4.4.6 Special Controls

It contains trend chart, Active X and others as below, instructions:

Pic 0-108

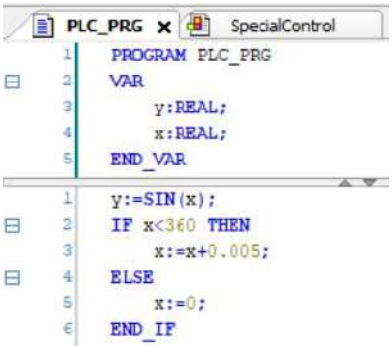
Trace

Used to insert tracking diagram and appointed in properties of tracking control.

Example 1: Create a new Trace to monitor filter of y:=sin(x) in visual view.

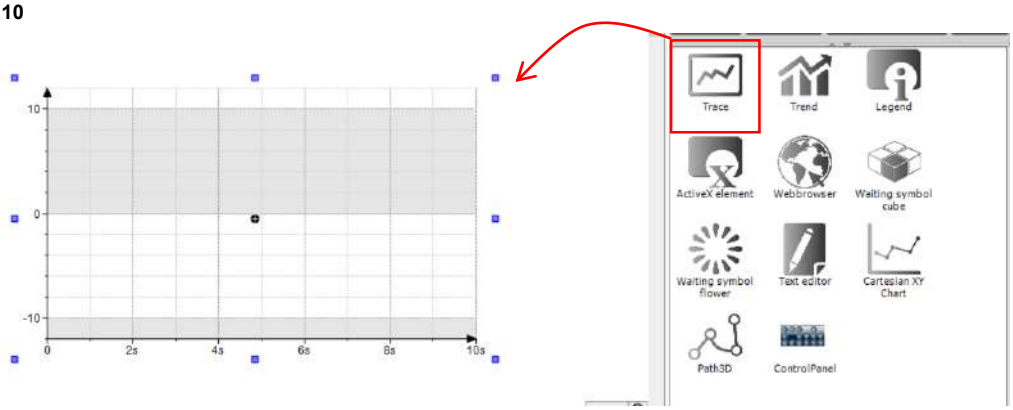
Programming example program in main program first.

Pic 0-109



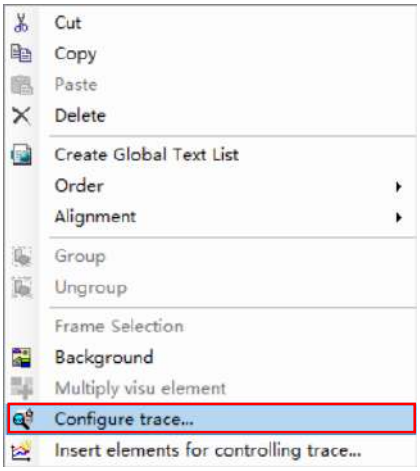
Find and add Trace to visual edit area from Special Control, tool box.

Pic 0-110



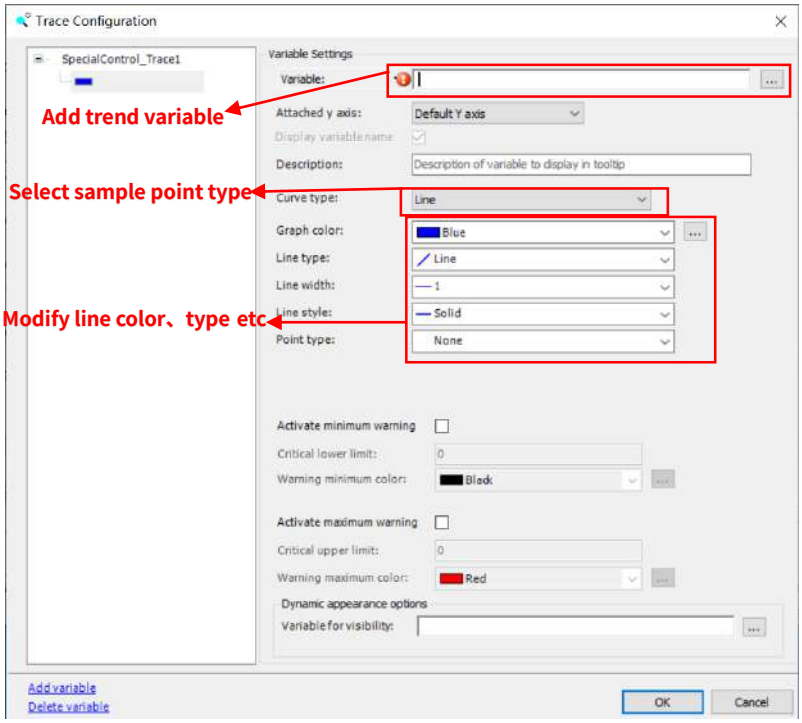
Single click Configuration Tracking to open it in Special Control, visual edit area.

Pic 0-111



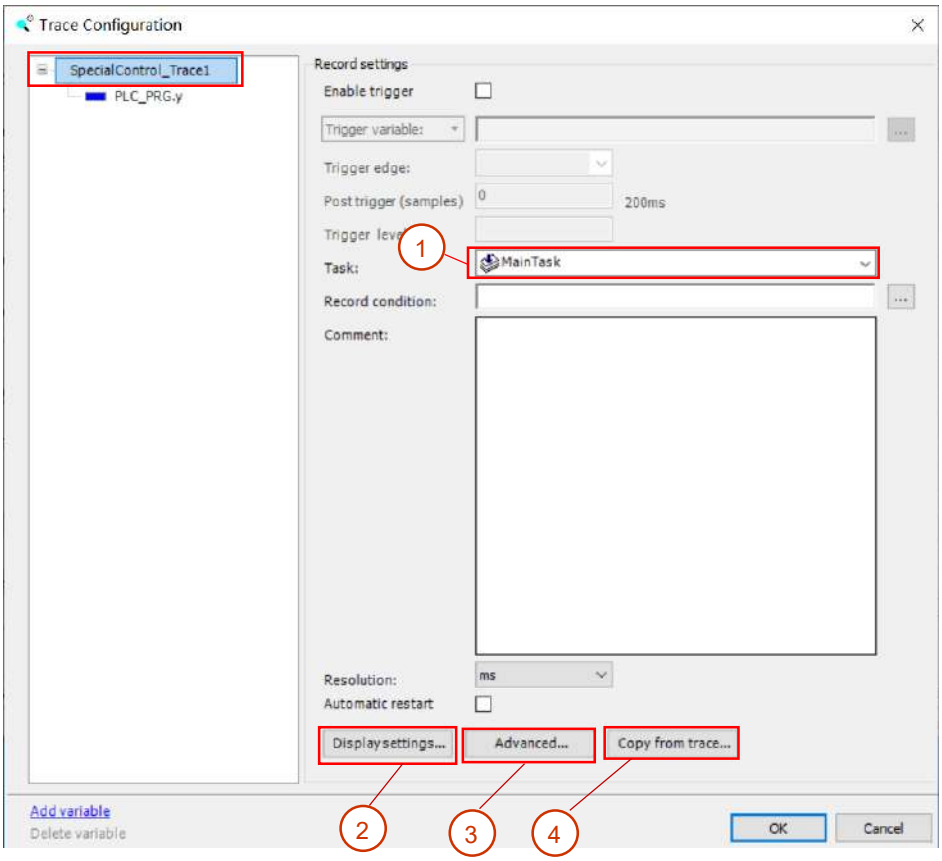
Add variable and set configuration in dialogue, and Trace point type, color of variable line type can also be set.

Pic 0-112



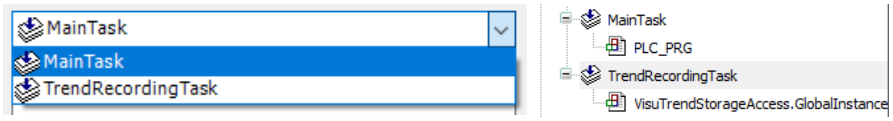
Single click first column in tree menu, including Trace task configuration, sampling trigger, etc.

Pic 0-113



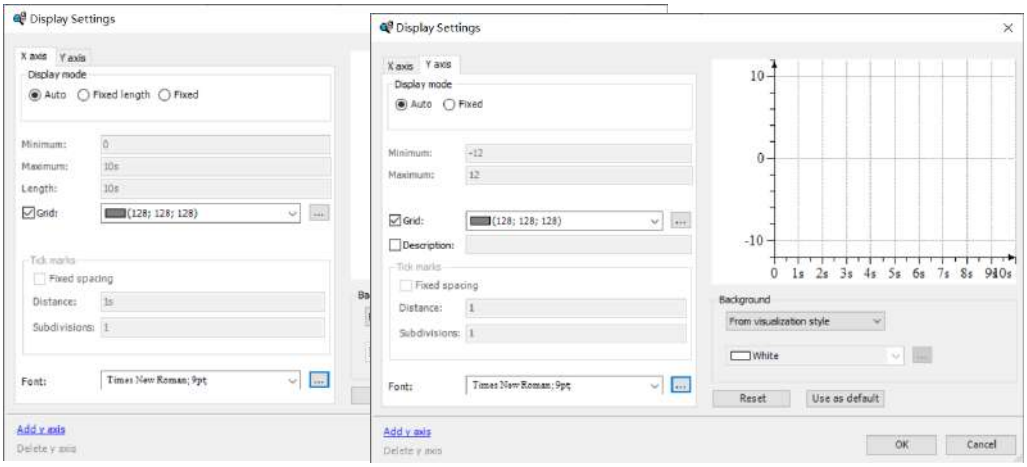
① "MainTask" 和 "TrendRecordingTask" are provided as two sampling cycles and MainTask as main, while "TrendRecordingTask" will be generated automatically after Trace is added.

Pic 0-114



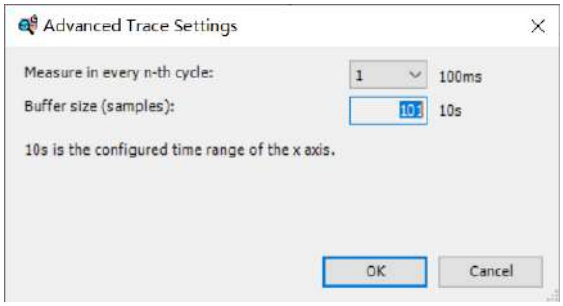
② X axis and Y axis can be tracked by setting in display, through default auto method or fixed length.

Pic 0-115



③ In advance options, user can choose to sampling after multi cycles of X axis.

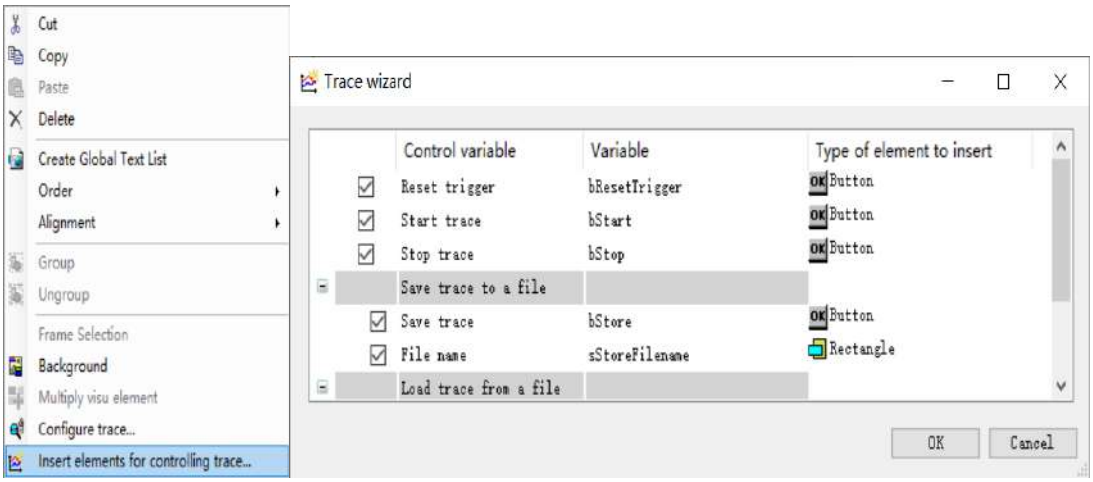
Pic 0-116



④ Variable configuration can be copied through Copy From Tracking.

Select tracking control, single right click, select Insert Element Control Tracking, select needed button and display box, all ticks in default, then system will generate control variable and button, display box.

Pic 0-117

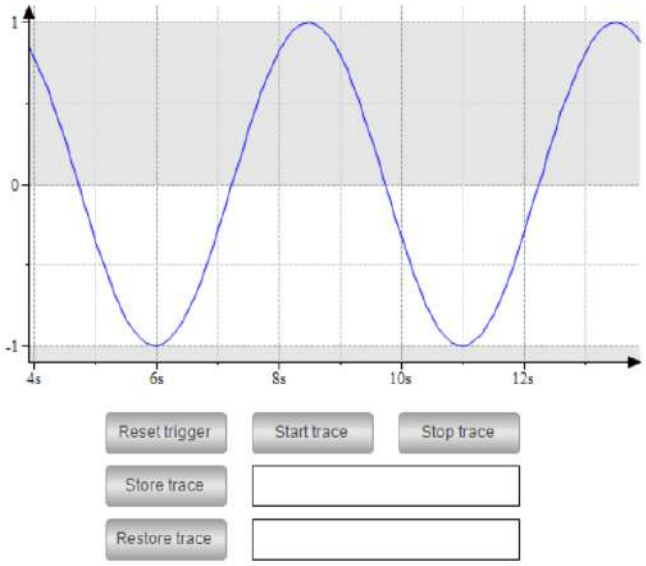


Interface as below after ticking all plug-in units:

Pic 0-118

Check online running:

Pic 0-119



ActiveX Element

Used for Active X control in Windows32 interface. Find ActiveX Element in Special Control of tool box, drag it to visual edit area.

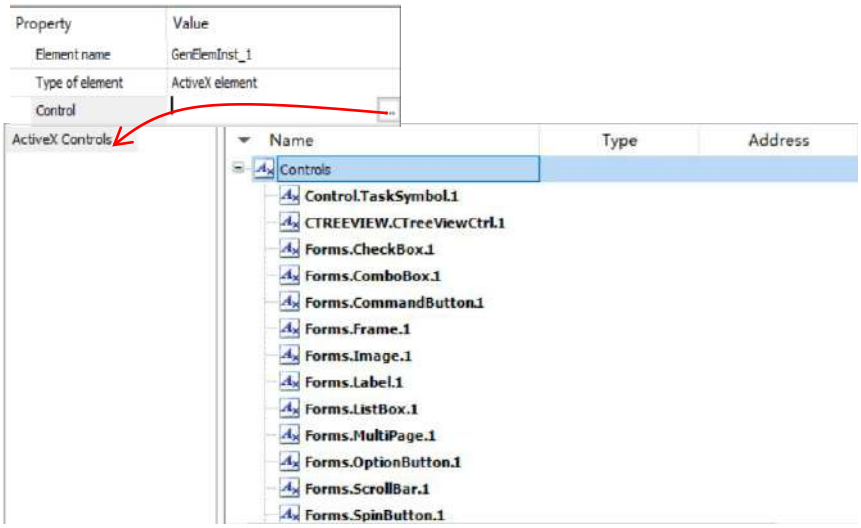
Three ways of calling as below:

- Initial call: Effect in only the first task cycle.
- Cycle call: Effect in each task cycle.
- Condition call: Effect in updating of visual event. Unlike previous two, it's effect only in rising edge.

Basic usage of ActiveX element.

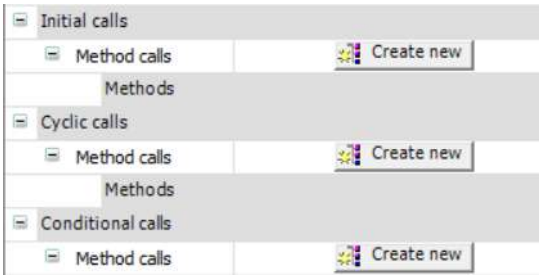
In Special Control of tool box, find and drag ActiveX Element to visual edit area. Find Control option in properties menu, select plug-in unit type via Input Helper logo.

Pic 0-120



Set trigger after needed unit is called in three calls. Complete mapping via Create New in properties menu.

Pic 0-121

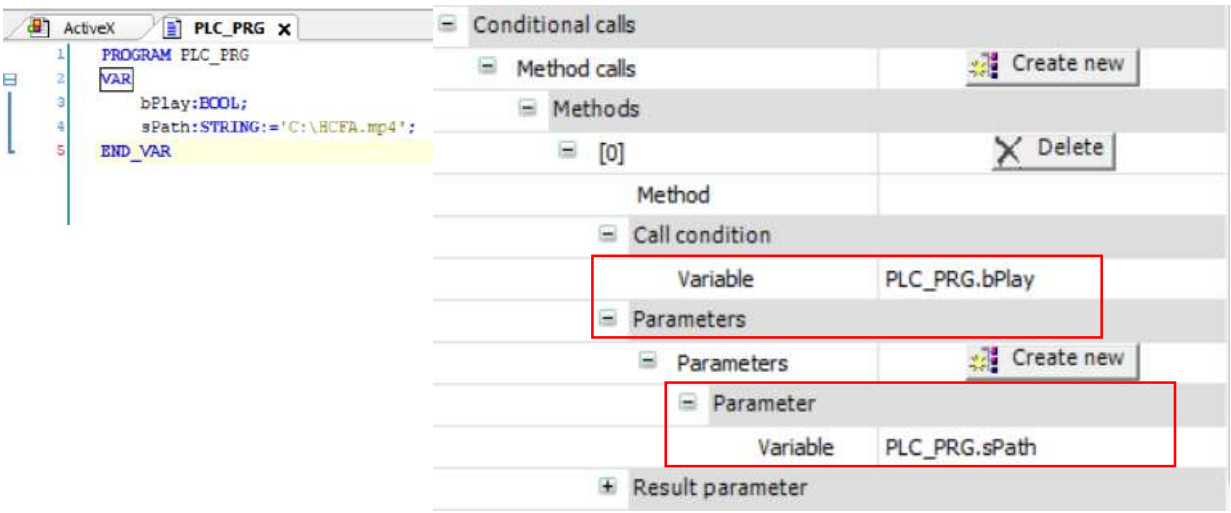


Example 1: Play video after rising edge signal is received by ActiveX element.

In Special Control o tool box, select ActiveX Element, find Control in properties menu, add "WMplayer.OCX.7" via Input Helper.

After adding, add "bPlay" in Variable, create "sPath" as system parameter in Parameter, then path of video can be quoted the added variable.

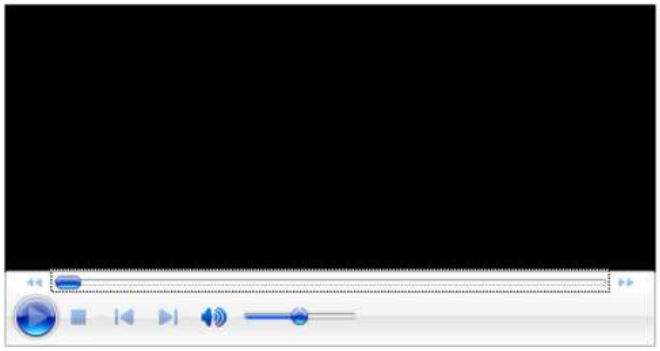
Pic 0-122



Add button and map to "bPlay" for video playing after triggered. Final interface as below:

Pic 0-123

Check online running:



PLAY

Chapter 5 Creation of Simple PLC Project

5.1 Start CODESYS

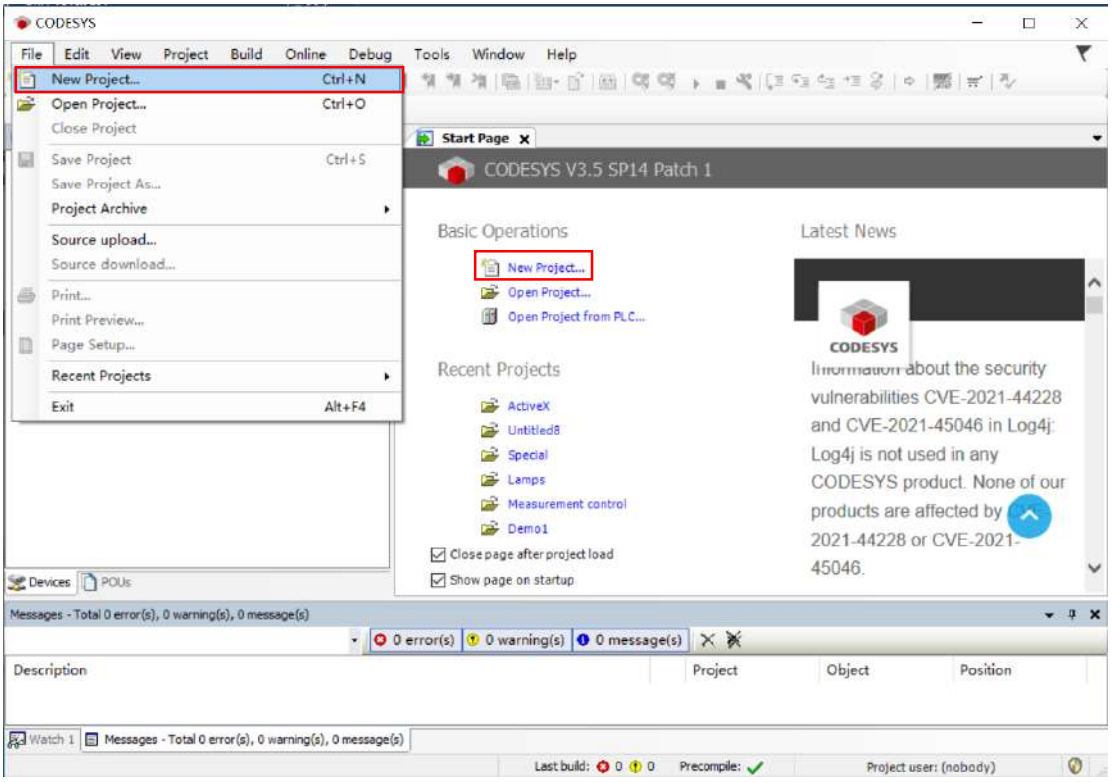
To create project in controller for certain plan, user need to connect to Q1 first, then edit corresponding IO group, logic program based on requirements, at last online debugging. Instruction of project creation is in this chapter.

Install CODESYS on PC and double click its shortcut on desktop.

5.2 Create New CODESYS Project

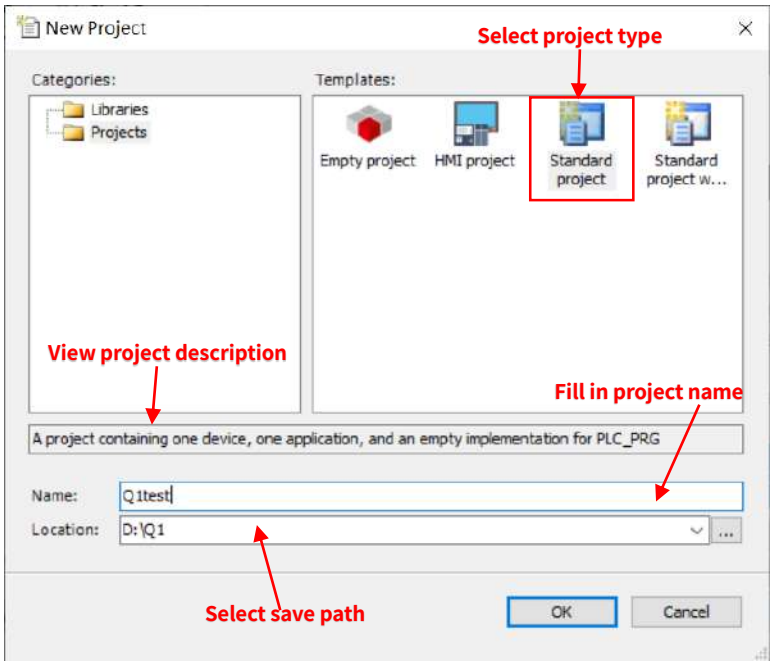
In initial page of CODESYS, find New Project, or find in File, or to open it directly from Previous Project List if edition was taken before.

Pic 0-1



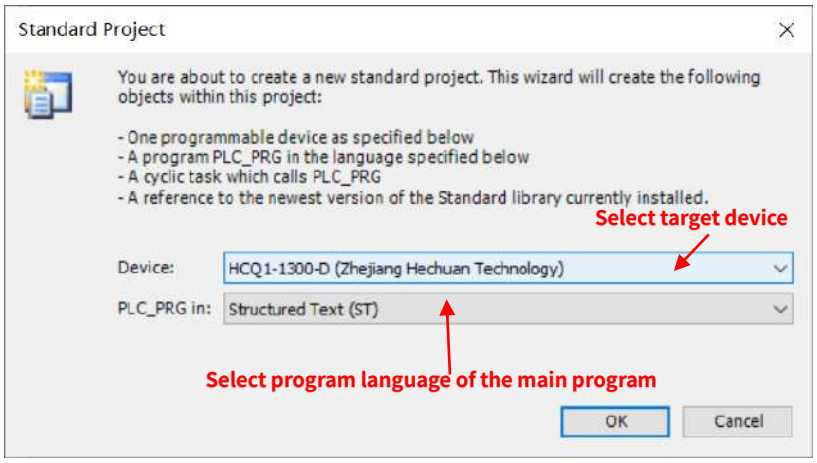
Select needed Model, enter name and storage path for it, and introduction of each model will show after it's selected. Click OK.

Pic 0-2



According to default guidance of CODESYS, select target device and main program PLC_ For PRG programming language. Q1 device is not installed by default, so Q1 device needs to be installed first, otherwise the correct target device cannot be selected. Refer to 3.2 for the installation steps of new description file.

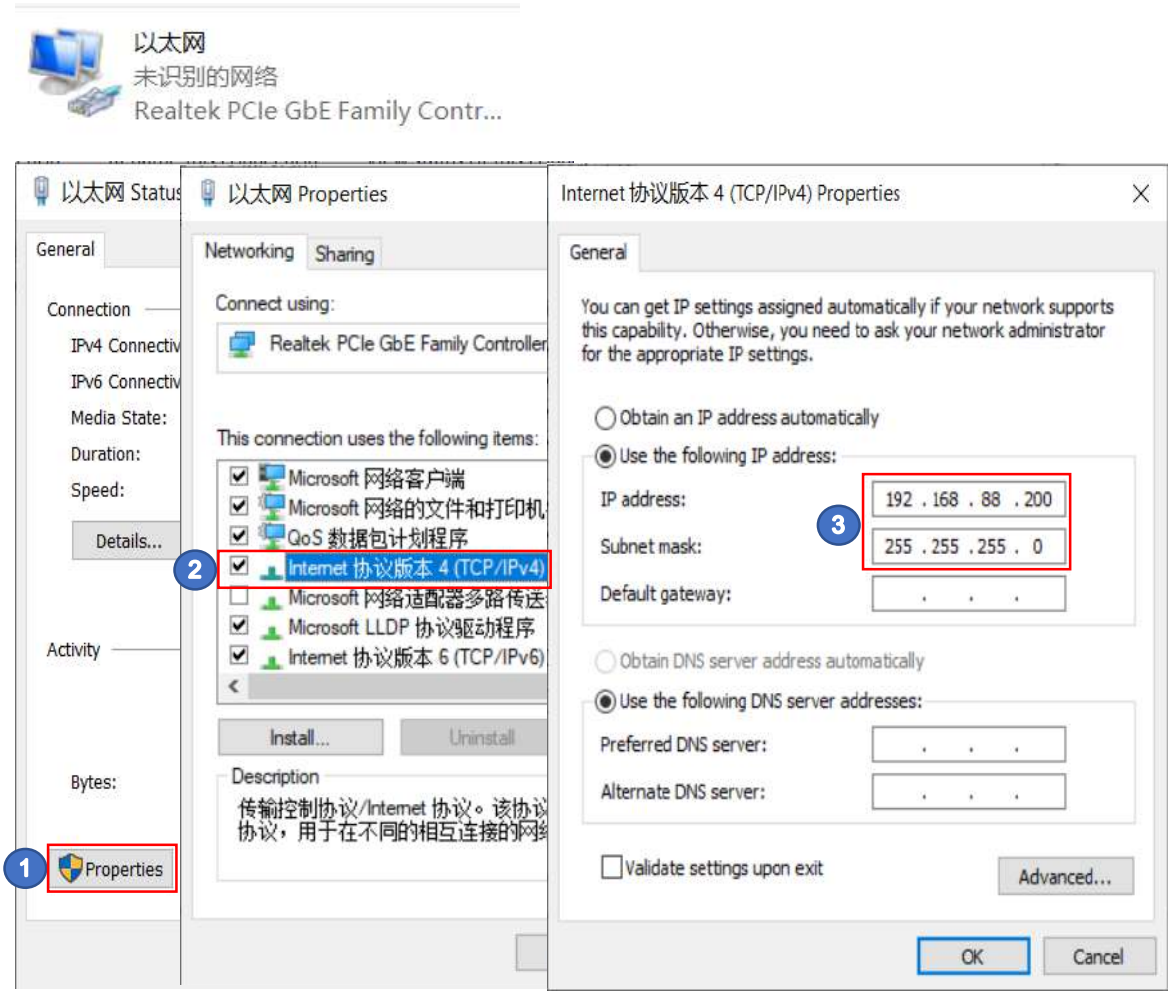
Pic 0-3



5.3 Establish Communication with Q1

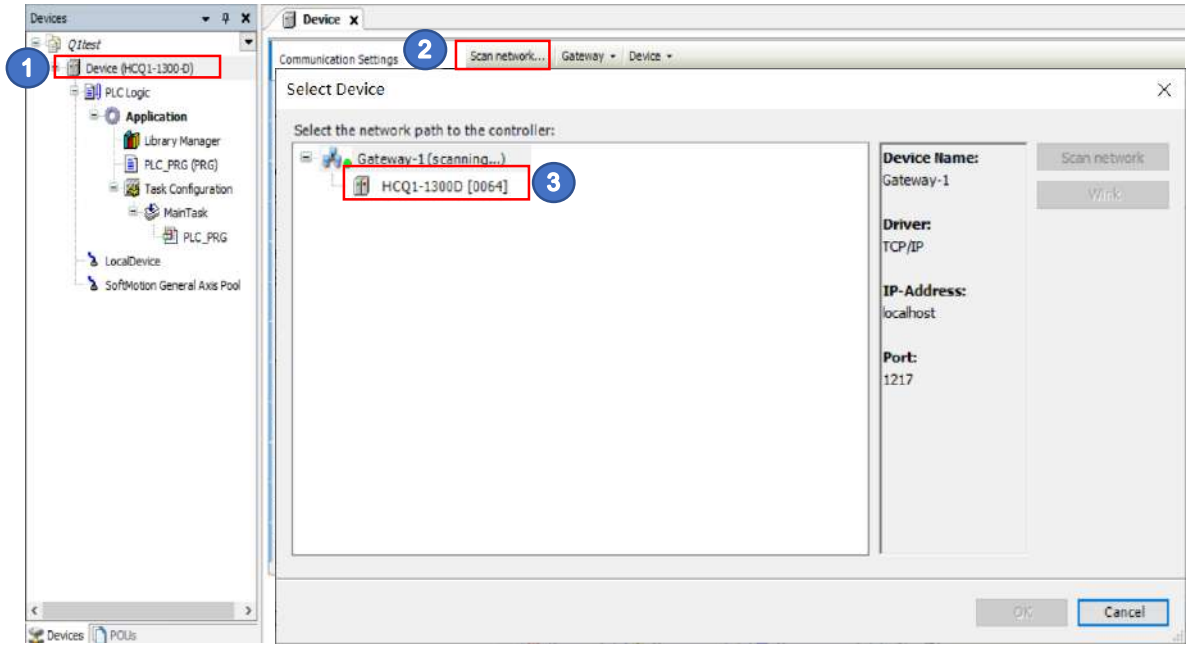
In Q1, default IP address of port1: 192.168.188.100 subnet mask: 255.255.255.0 Modify IP address of PC in its net adapter to same network segment(default IP address of port: 192.168.88.xxx subnet mask: 255.255.255.0).

Pic 0-1



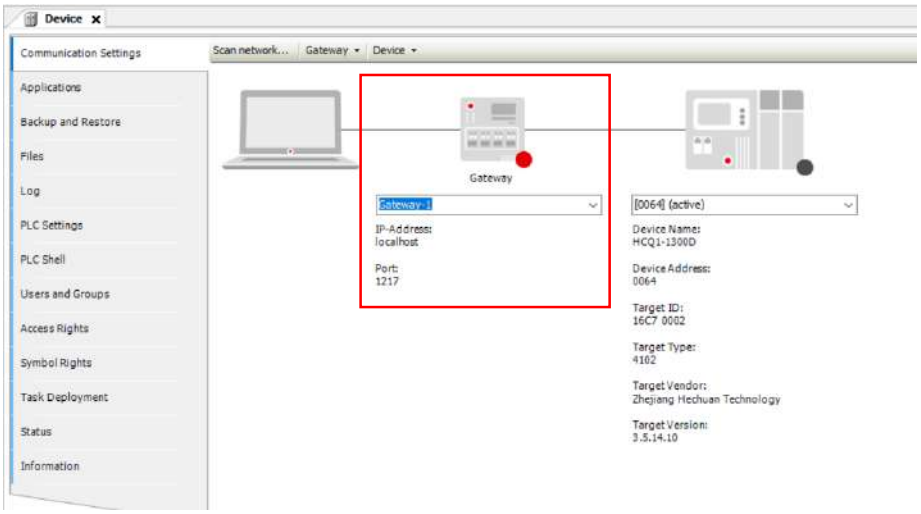
After above, double click Device in tree menu of create new in CODESYS, to enter communication device. Open the gateway correctly, click Scan network, select and add Q1.

Pic 0-2



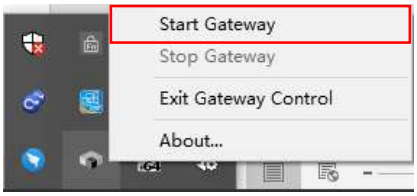
If gateway is not opened, it will display in red in Communication Device and need open manually.

Pic 0-3



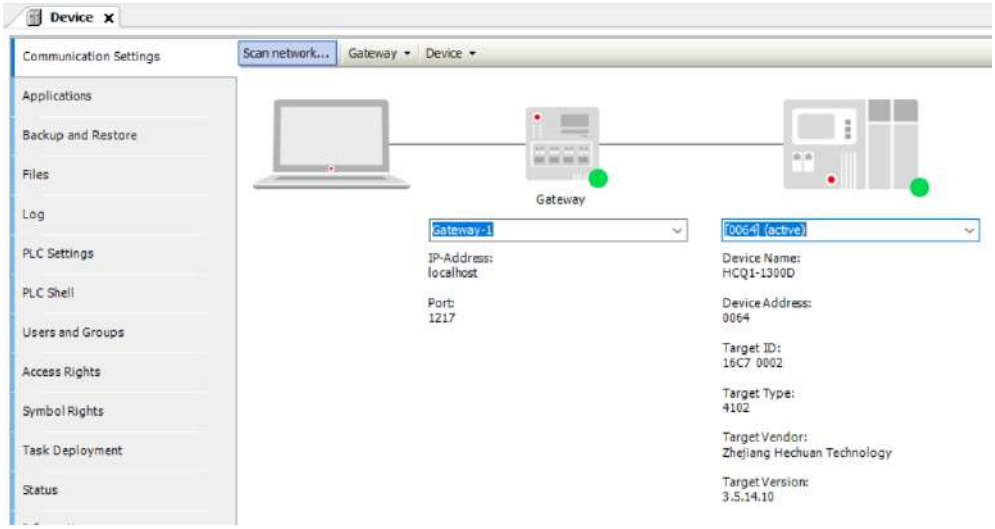
Find CODESYS logo at low right corner of PC, click Start Gateway and execute scan and add.

Pic 0-4



Device added correctly displayed as below:

Pic 0-5



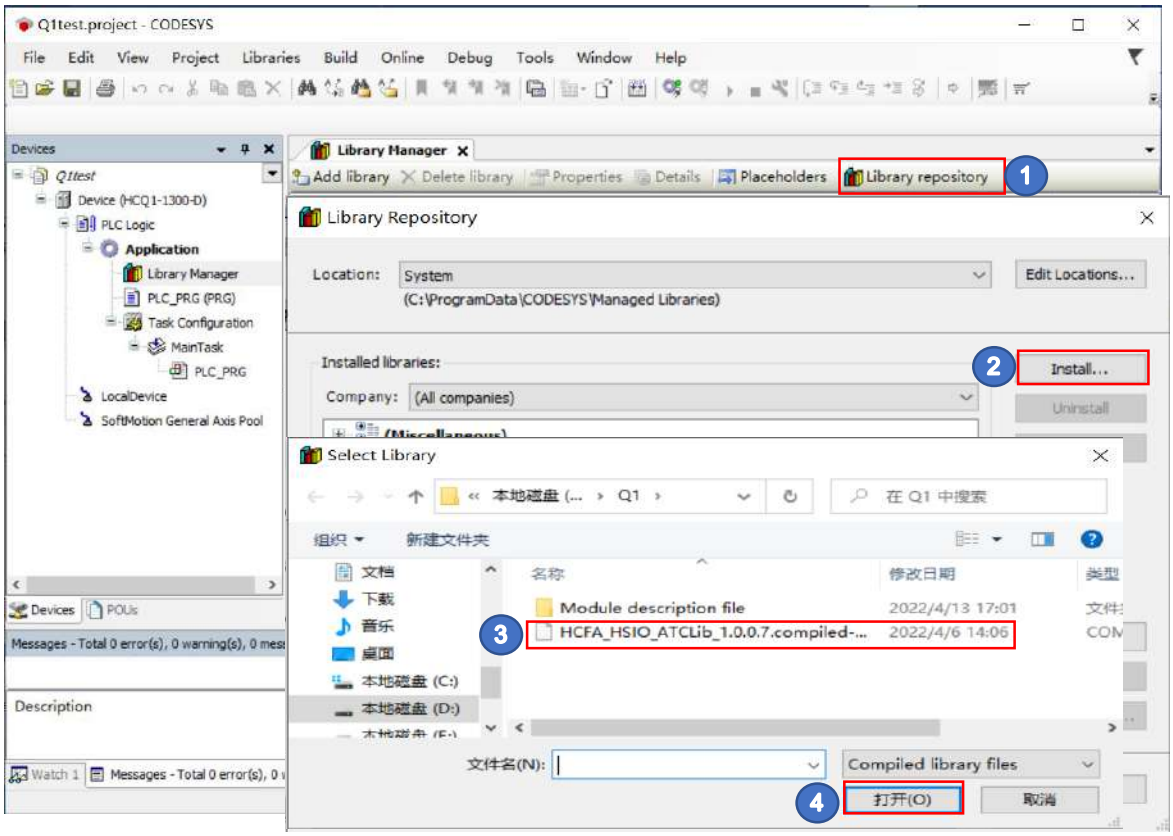
5.4 Creation of PLC Project

A standard PLC project, including Libraries, Tasks and Programs, can add Trace to monitor the variables in the program in real time. In order to facilitate debugging, users can also choose to establish a V. The creation of PLC program file is not only the establishment of operation structure and operation sequence, but also the establishment of programming mode. For a new PLC project, the system will assign a continuous task by default, which contains a default program "PLC_PRG".

5.4.1 Add Library

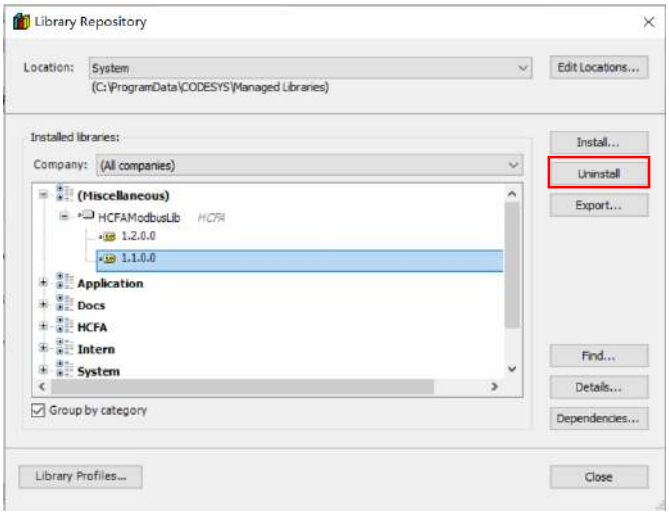
User need to install library file if extern library is needed. Double Library Manager in tree menu to enter it, find Library repository→Install→Library needed for installation→Open to finish installation.

Pic 0-1



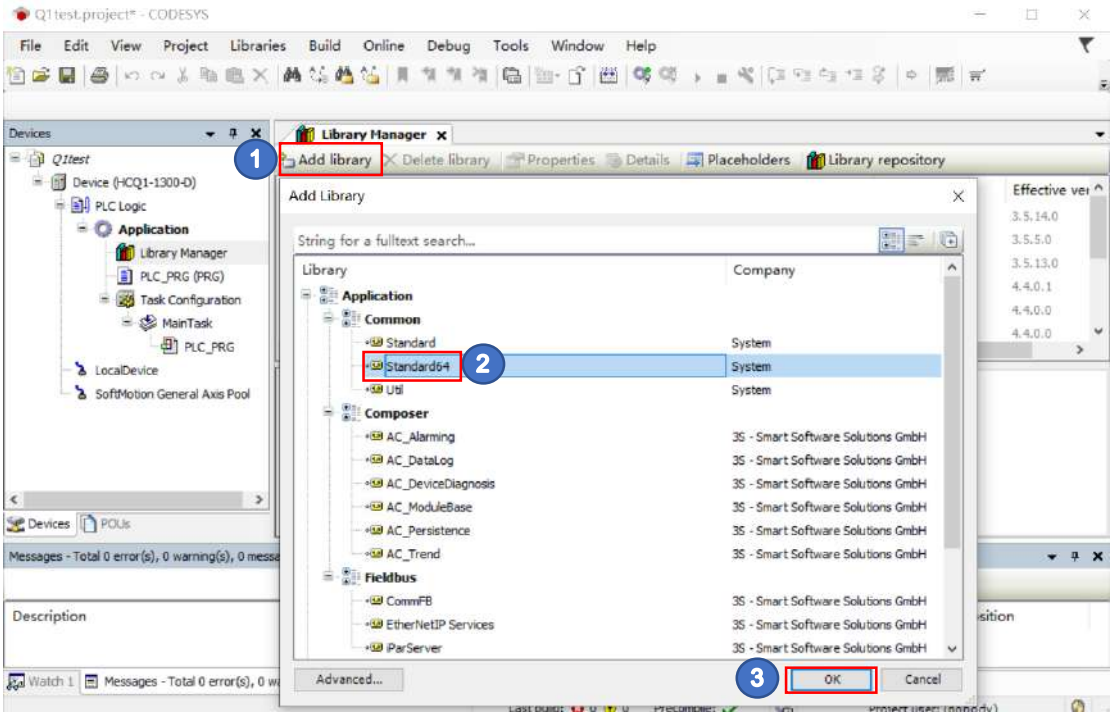
CODESYS can automatically identify the library version and support the switching between multiple versions of library files. If need to delete, select the library file to delete and click the delete button in the same interface.

Pic 0-2



Installed library can be called by adding library button and all FB and FUN can be used in program. All file will occupy memory of PLC and user can add library according to personal need.

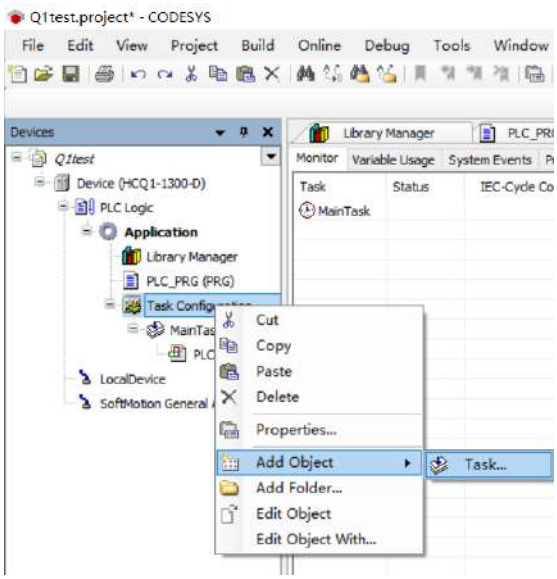
Pic 0-3

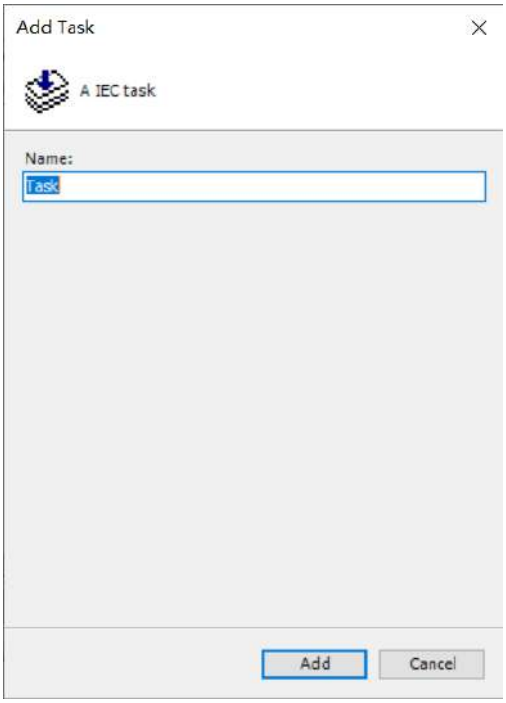


5.4.2 Task Setting

Manage all tasks in task configuration of tree menu and a new standard PLC project will automatically generate a cycle executed task, which will be linked to PLC_PRG automatically, with default task cycle of 4ms and priority of 1. PLC program will attend compile and execute only after being called. Right click Task Configuration→Add Object→Task, name the task and finish creation. At most 100 tasks of different types can be created and executed based on priority set by user. Smaller number indicates higher priority and tasks will be executed top downwards based on task configuration if with same priority.

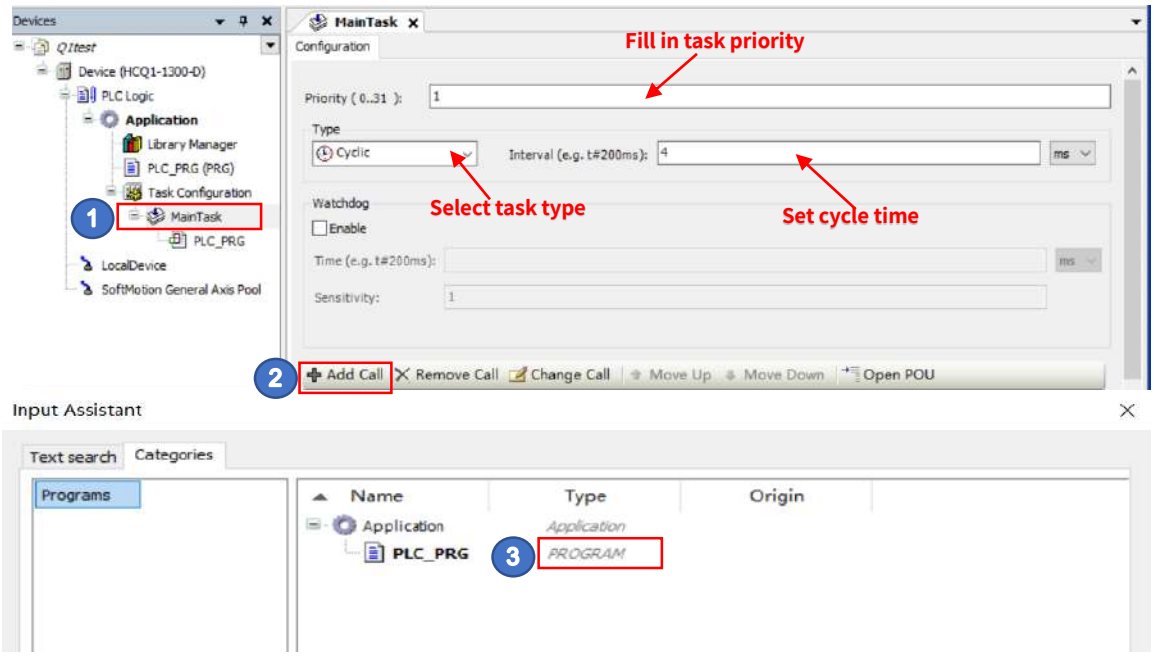
Pic 0-4





Manual configuration and call is needed for newly created PLC program. Double click MainTask→Add call→select needed PLC program and finish calling.

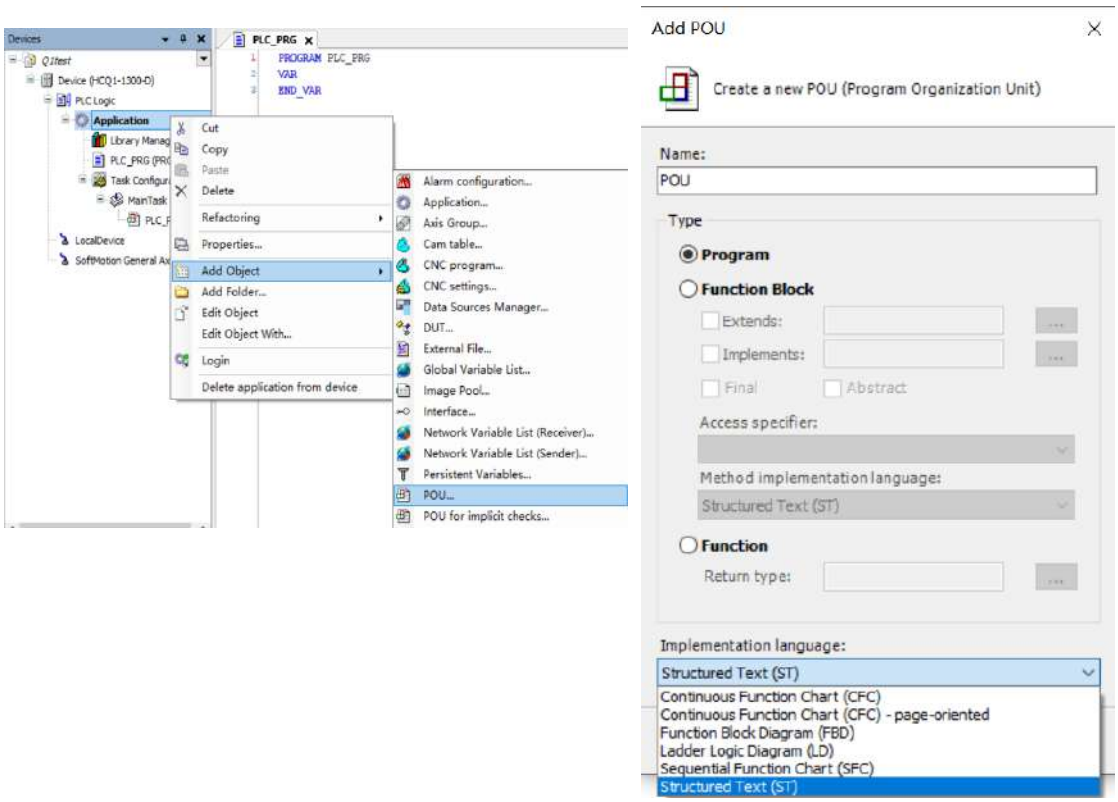
Pic 0-5



5.4.3 Programming of PLC Program

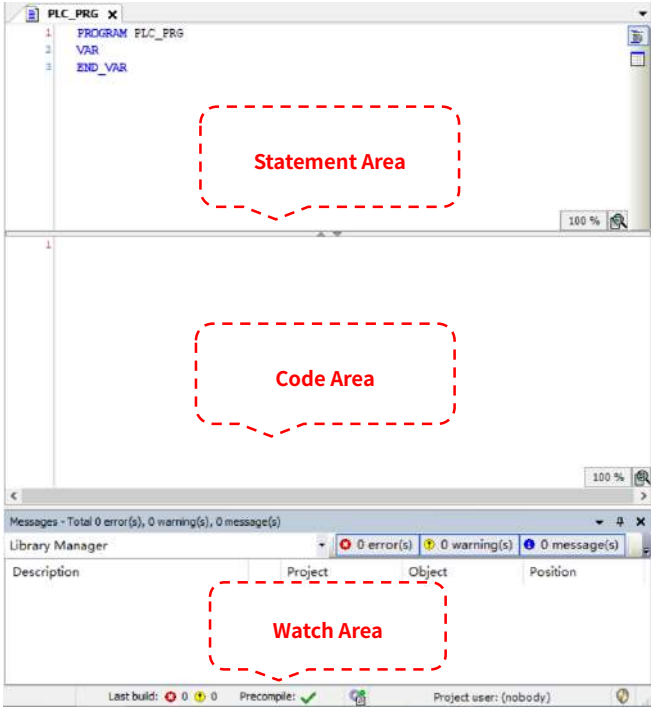
Create POU: single right click Application→Add Object→POU, enter name of it in dialogue and select program block type. Six Implementation languages are supported in CODESYS and ST is shown as example here as below:

Pic 0-6



Double click PLC_PRG to enter programming interface.

Pic 0-7



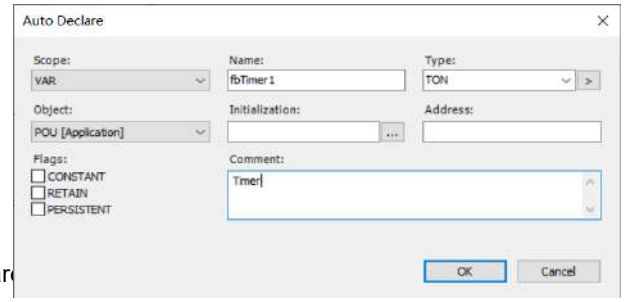
Example programming of square wave with adjustable duty cycle is as below:

State needed variables according to IEC61131-3 in program first.

Express of variable: name: type: =initial value; optional to appoint initial value or not, as variable has default initial value. Operate single line comment via "//"and Chinese comment is supported, but English comment is recommended due to false position display of cursor with Chinese comment.

Shift+F2 in statement window of variable to pop up dialogue, and it's necessary to fill name and type.

Pic 0-8



Scope: range of variable

Name: name of variable

Type: variable data type

Object: corresponding application

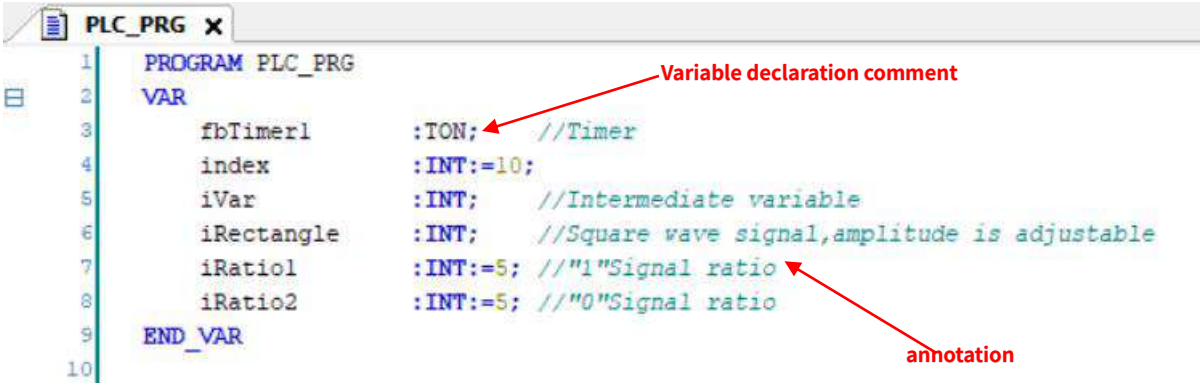
Initialization: initial value

Address: mapping between variable and external hardwar

Flags: variable type can be constant, lossless or persistent type

Comment: comment in format of "comment content"

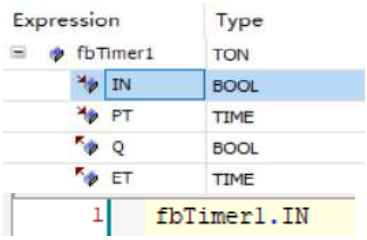
Pic 0-9



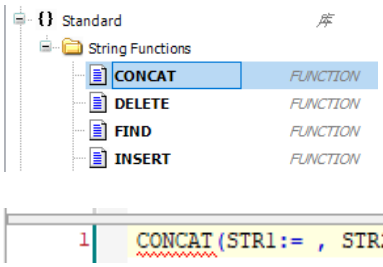
Programming example program in main program window, statement of variable can be called via F2 in main program window. Pay attention to Calling type.

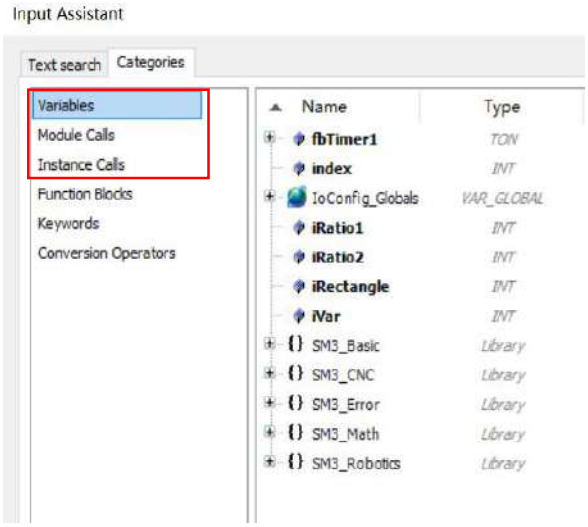
Variable：statement of single variable is needed

Pic 0-10

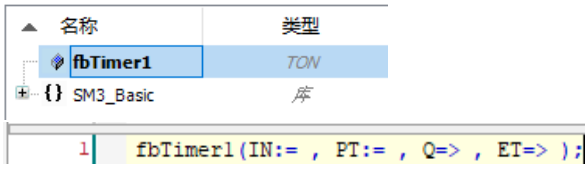


Module Calls：FNC, direct and needs not statement



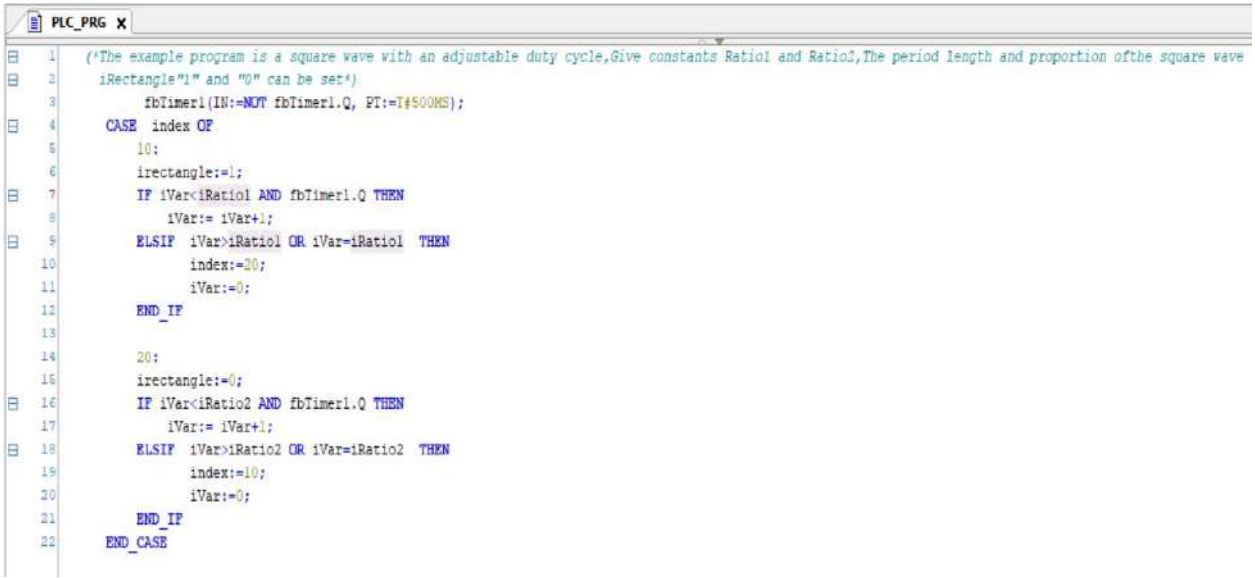


Instance Calls：FB, needs statement



Programming below per above rules：

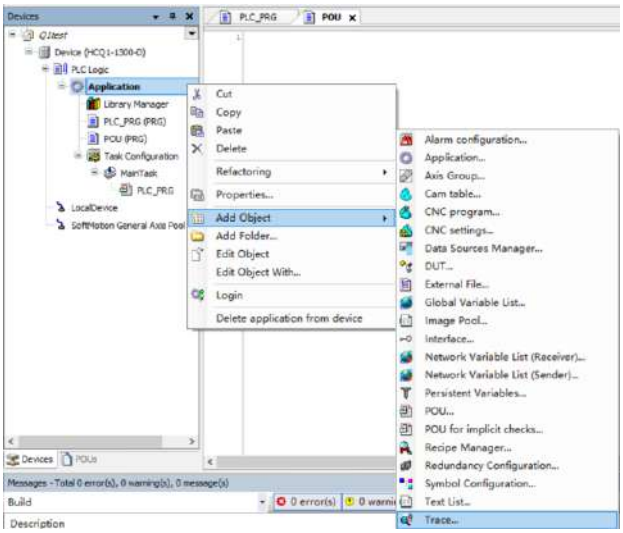
Pic 0-11

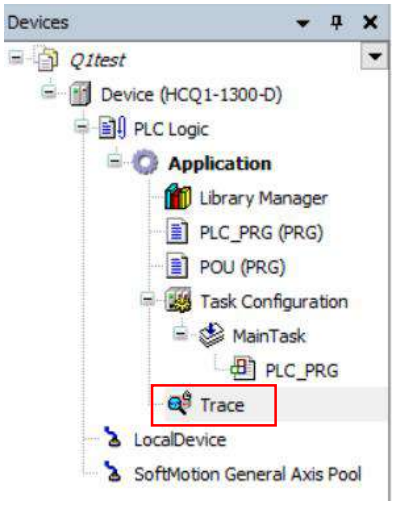


5.5 Add Trace to monitor program variable

Right click Application→Add Object→Trace, then Trace will appear in left side tree menu.

Pic 0-1



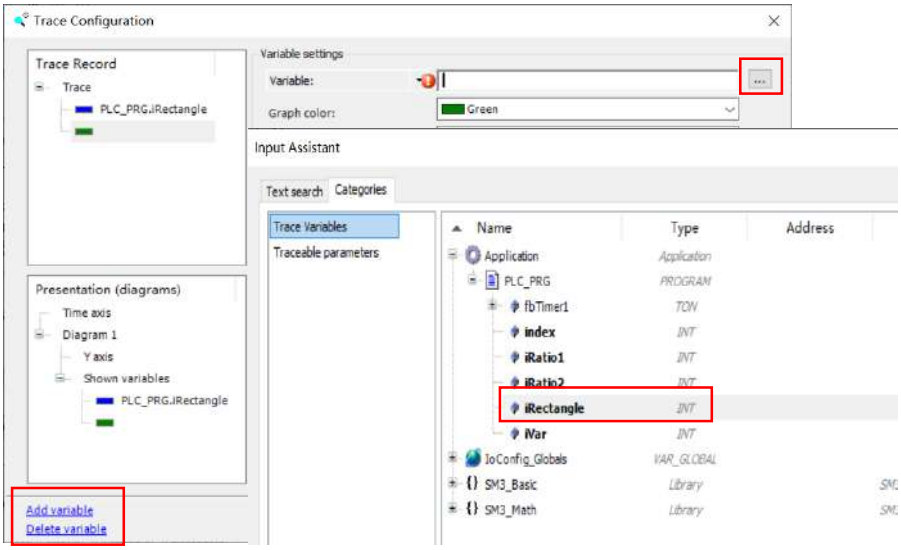


Double click to enter interface of configuration.

Pic 0-2

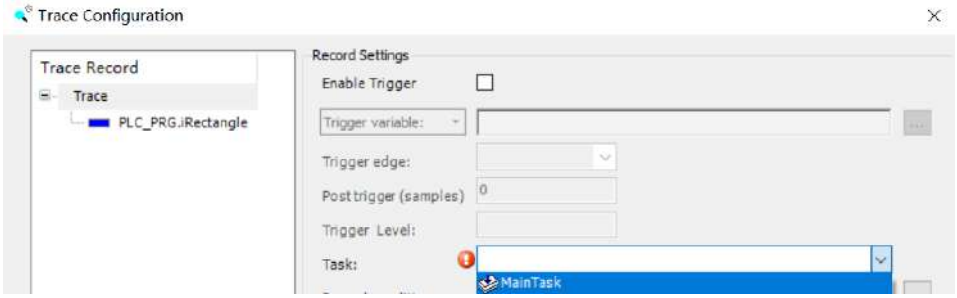
Single Configuration button in upper right corner of Trace interface, add/delete monitor variable and set sampling cycle. Open trace configuration interface and add/delet via add/delete variable button in lower left corner.

Pic 0-3



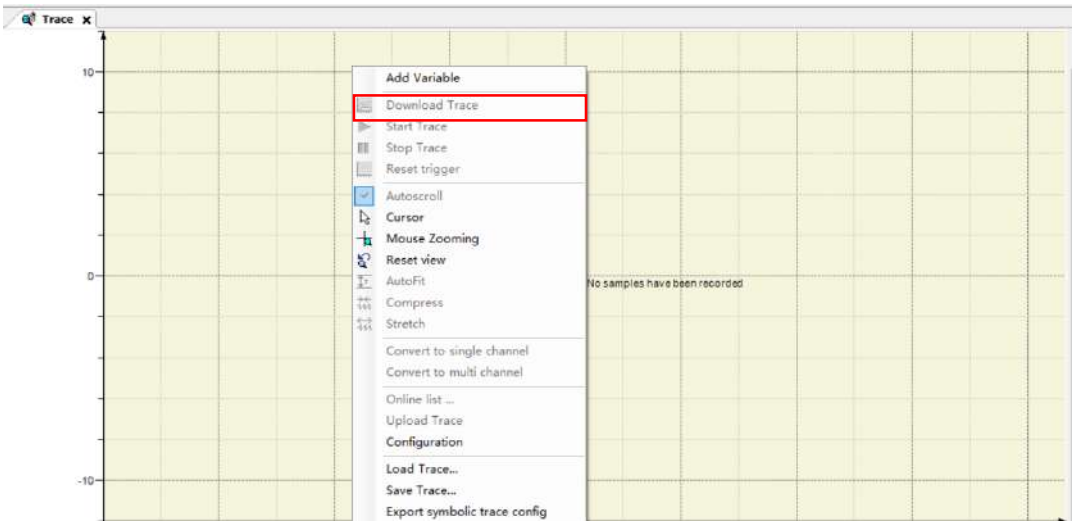
Single click task that follows Trace in left tree menu, select MainTask as example in drop down list.

Pic 0-4



After basic configuration, Login and Start first, then operations like Download Trace, observe variable running track can be executed. Or the options will be in grey and can't be selected.

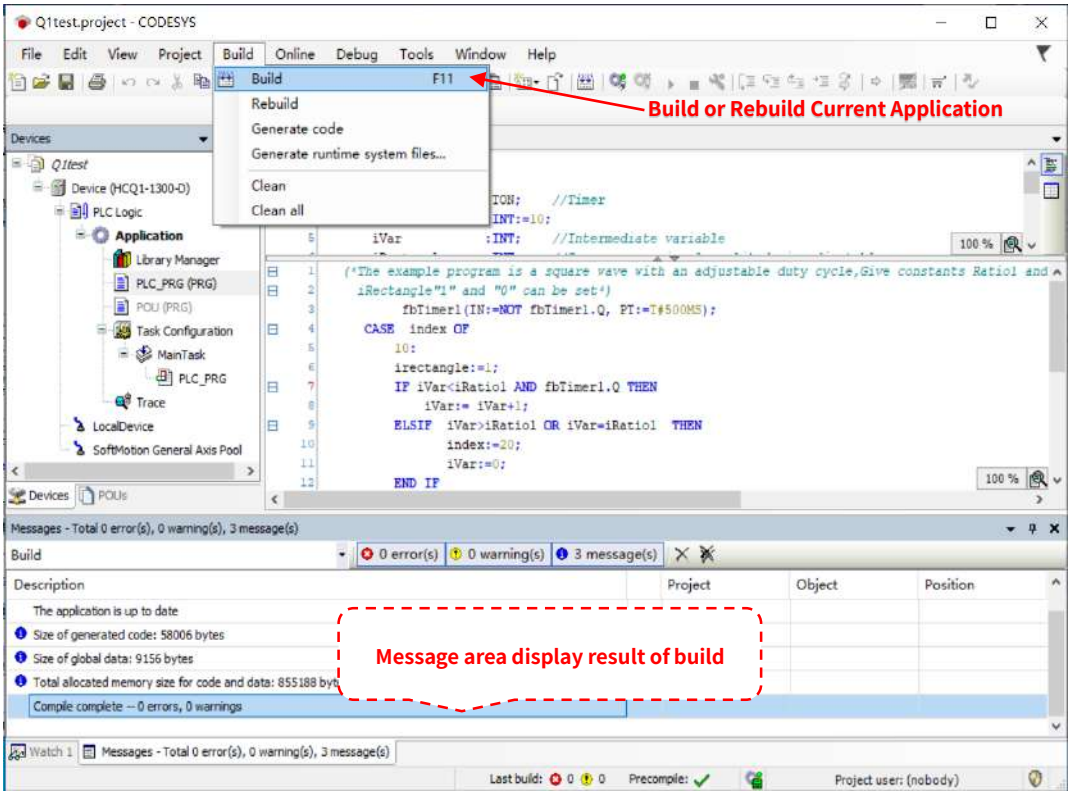
Pic 0-5



5.6 Program download and online monitoring

5.6.1 PLC Program Compiling and Download

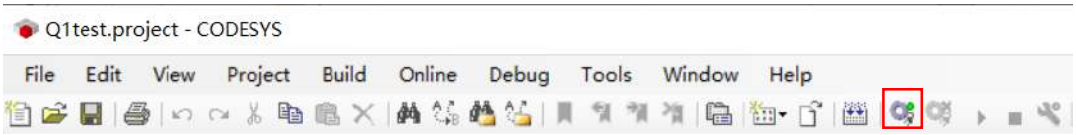
Building is needed before Login. Building orders will check language grammar of program. Created POU should be added to Task before build. Select Build in menu bar, Build, then result will appear in news window, needed information can display separately via selecting error, warning and news.



5.6.2 Login and Start

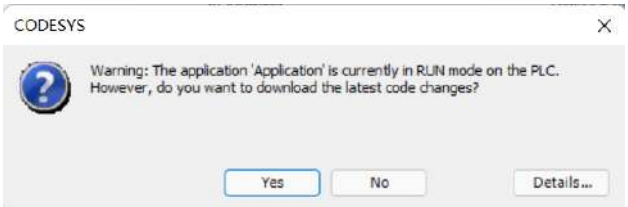
After building correctly, execute login command: find login logo , single click to enter program and online monitoring.

Pic 0-1



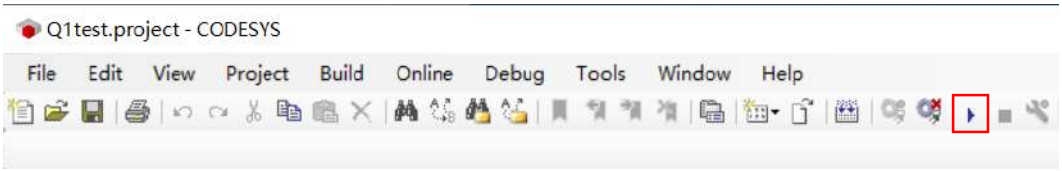
Remind in dialogue: downloaded application will cover original application, to execute or not, click YES

Pic 0-2



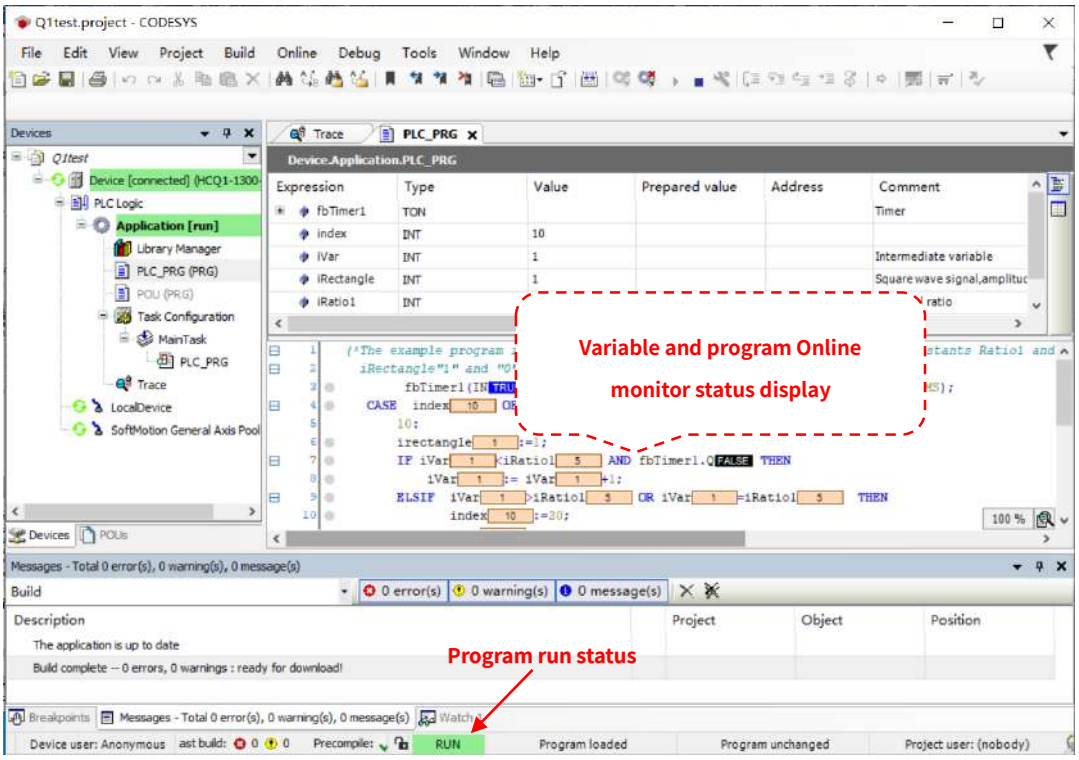
After login, click Start logo to run.

Pic 0-3



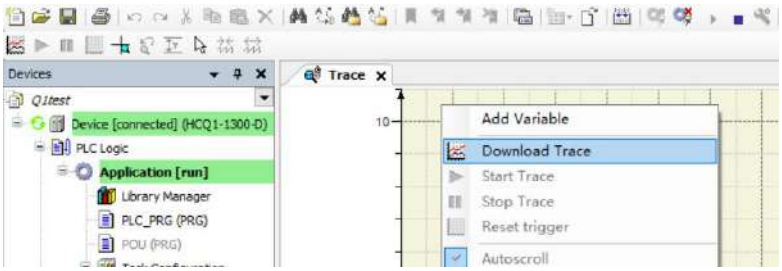
Running interface as below after correct login:

Pic 0-4



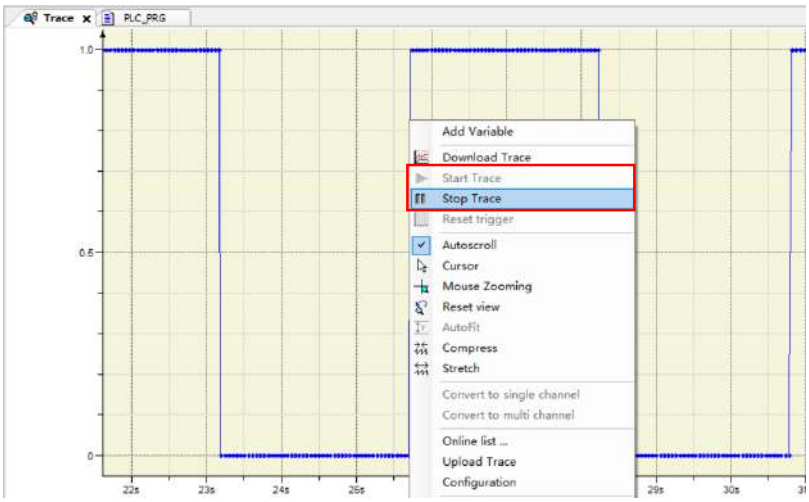
Download Trace for sampling running trace of variable.

Pic 0-5



Right click in any position of Trace interface to pause/start tracking, move the mouse to swift observe history, wheel can be used to zoom in or out tracking.

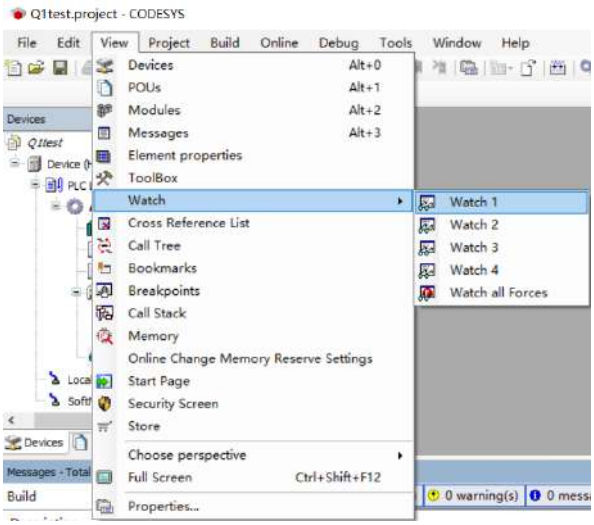
Pic 0-6



5.6.3 Online Watching

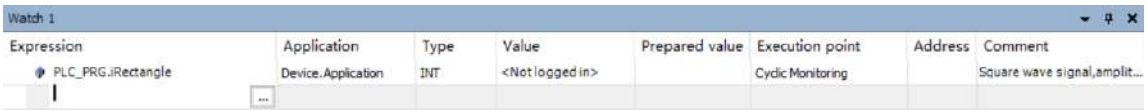
After login of program, actual value of variable will display in main program and statement window. If multi POU need to be monitored , integrate the needed variables. In menu bar, find View→Watchr→Watch 1, then list of Watch 1 will be created and 4 Watches can be created.

Pic 0-7



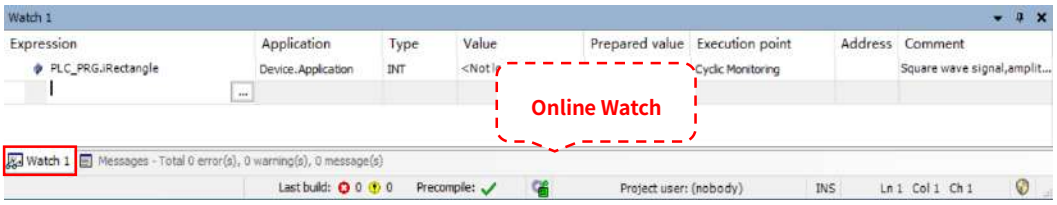
Open Watch 1, single empty space under the express, variables or global variables can be Called via point quote, or click ... in right to open input helper and add variable for online watching.

Pic 0-8



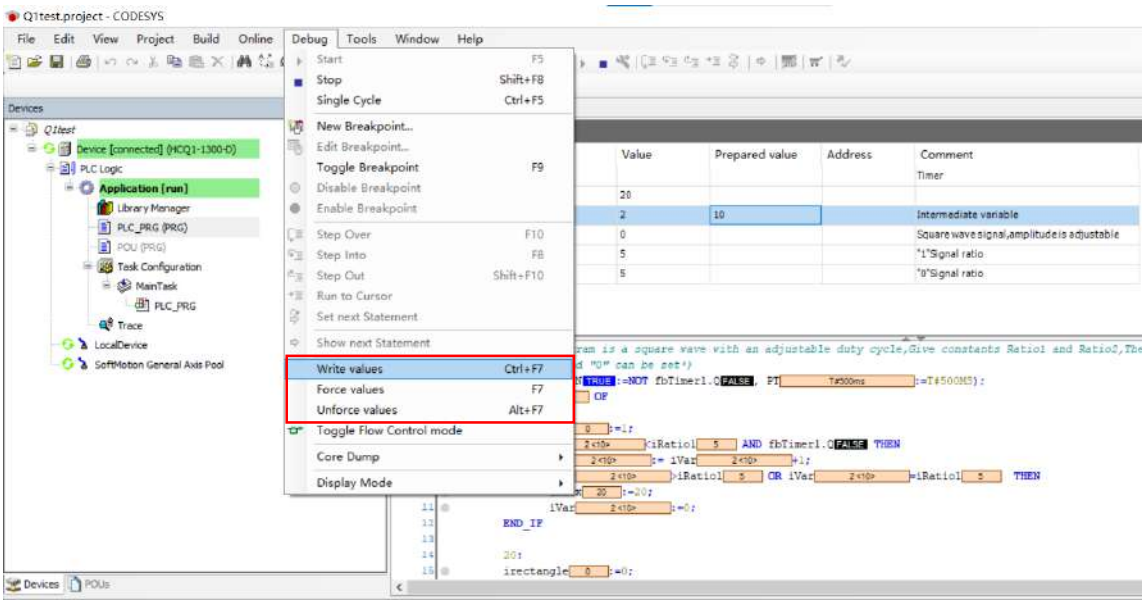
After configuration, during normal operation of the program, user can view the real-time value of variables and assign values to the current variables through "monitoring 1" tab under the message window. Even during operation, user can manually add or delete monitoring variables, which is very convenient.

Pic 0-9



In login condition, needed value can be written to prepare values in statement window, main program window or monitoring window. After that, in menu bar, Debug→Write values to assign. Two ways to assign in CODESYS: Write values(Ctrl+F7) and Force values(F7). Value written through Force values will have logo F on left and can be released via Release Value

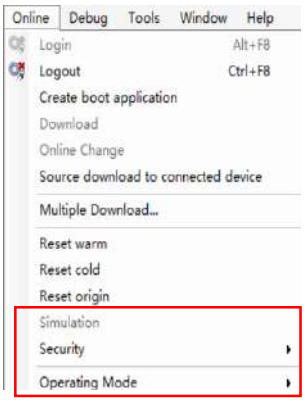
Pic 0-10

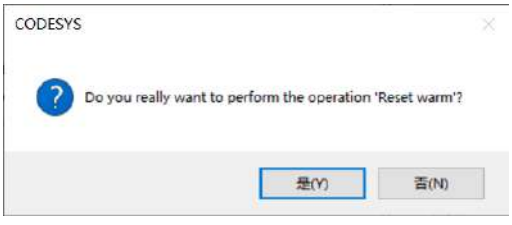


5.6.4 Reset

If reset is needed during debugging, here are three types of them: Reset warm, Reset cold and Reset origin. In menu of Online, single click to pop out dialogue to choose a type. Influence of different type on variables as below:

Pic 0-11





Notice: X: reserved value O: initial value

Pic 0-12

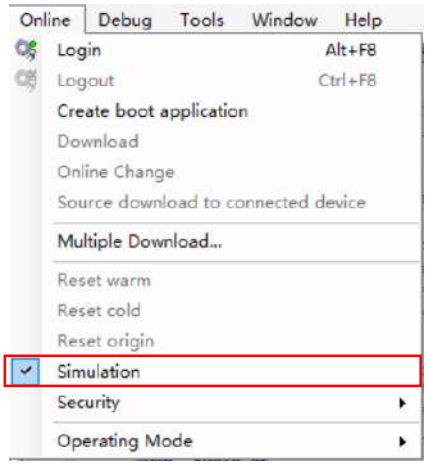
Online Command	VAR	VAR RETAIN	VAR RETAIN PERSISTENT
Reset warm	O	X	X
Reset cold	O	O	X
Reset origin	O	O	O
Login with Download	O	O	X
Login with Online change	X	X	X

5.7 Simulation

CODESYS allows user to use simulation function without actual hardware, and simulation of PLC program on PC can assure complete test before actual debugging and find logic errors as early, reduce developing time. During simulation, no PLC hardware is needed and user can compile, login, run edited program on PC directly.

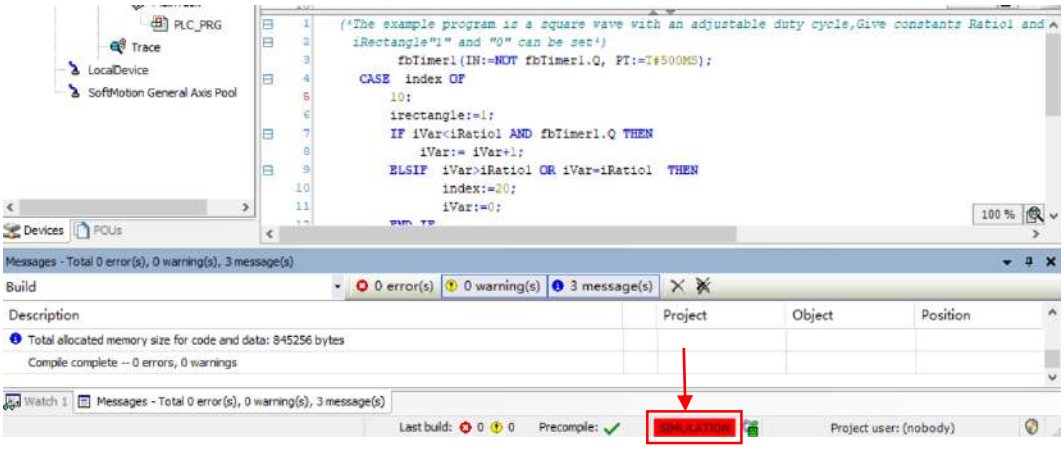
Steps: in menu bar, find Online → Simulation to enter simulation mode, tick and the logo will appear on left of simulation.

Pic 0-1



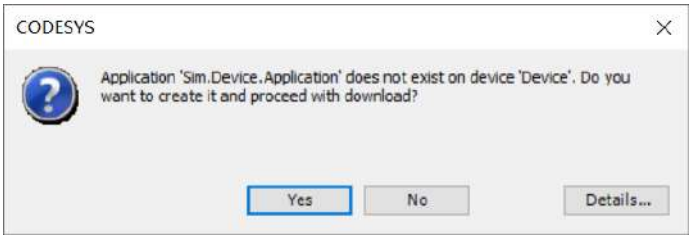
Entering the mode and red logo of simulation will appear at bottom of the interface.

Pic 0-2



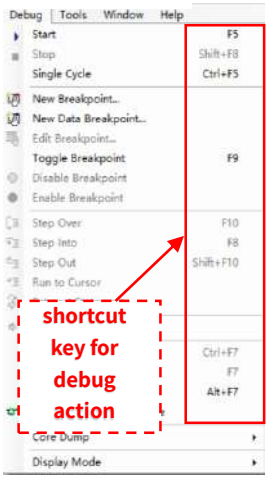
Compile directly and download with no mistake. Single click Yes to continue when the dialogue about program reminder pops up at beginning.

Pic 0-3



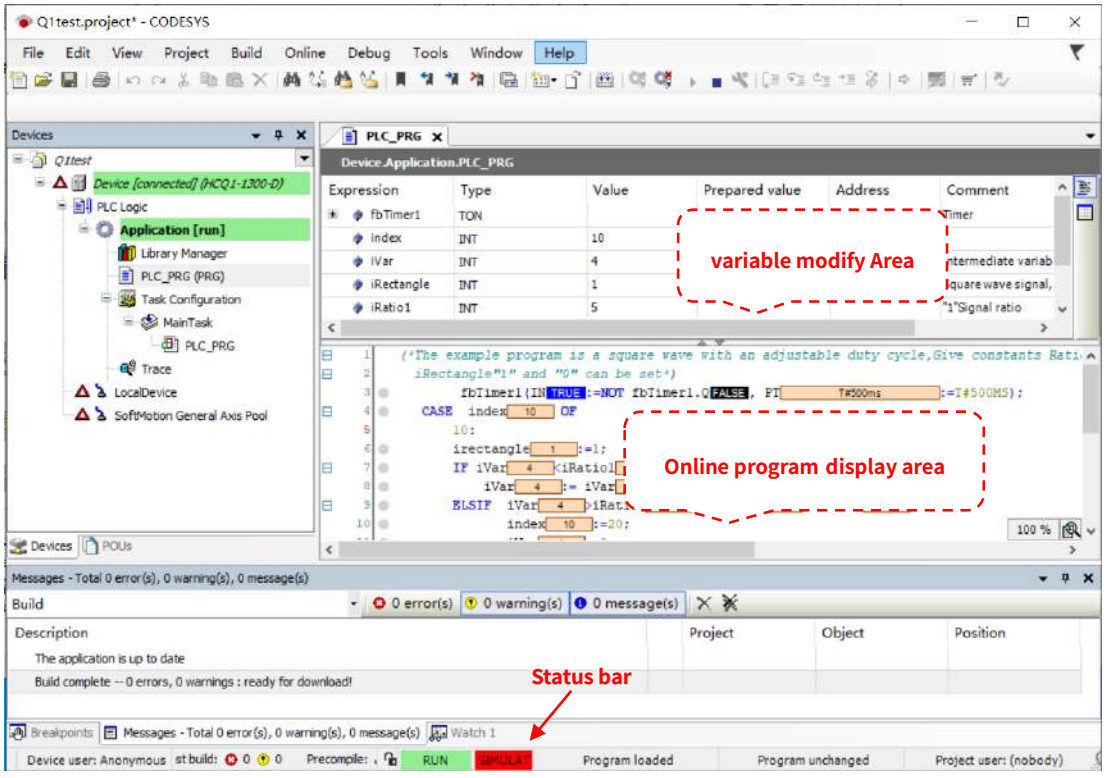
There's no difference between operations in simulation and actual PLC, and user can execute modification of variabl. Common functions like <Ctrl+F7>, <F7>, <Alt+F7> can all be realized with short keys. Refer to more options under Debugging in menu bar.

Pic 0-4



Running as below:

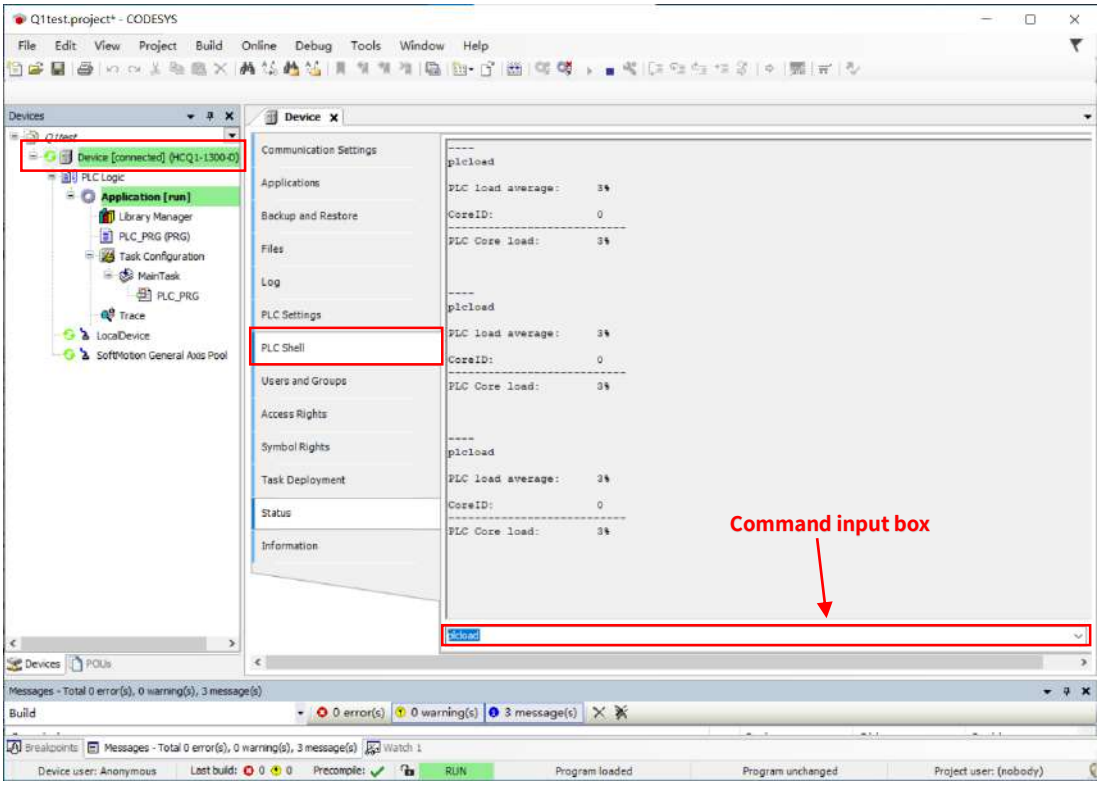
Pic 0-5



5.8 PLC Shell

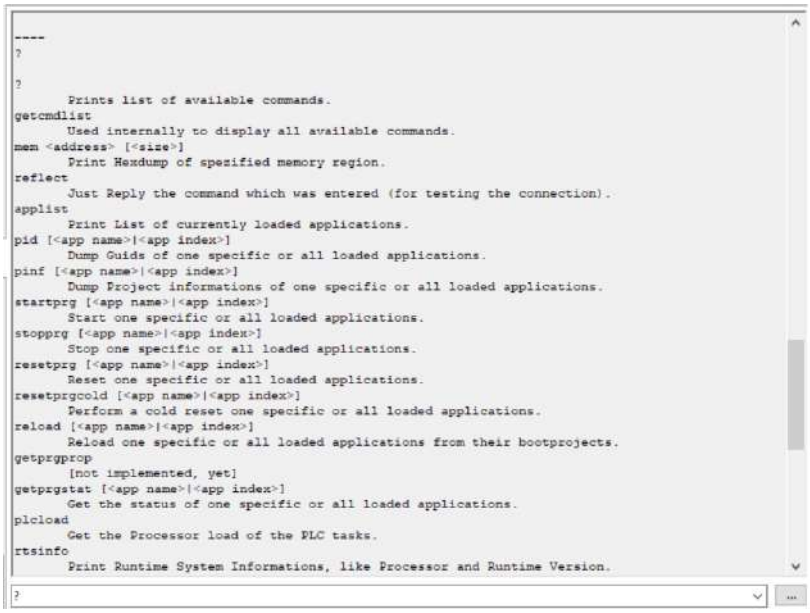
As a control monitoring(terminal) based on text, it gets orders from controller with certain information, input as a input line and sent to controller as a character string, return corresponding string in scripting window. The function is used for debugging and its interface as below:

Pic 0-1



Detailed commands of PLC scripting are in below chart. Double click Device, find PLC Shell and command input box under it. Corresponding command in the box will be returned with feedback from Q1. Enter “?” and all commands supported by Q1 will be returned.

Pic 0-2



```
----
?

?
       Prints list of available commands.
getcmdlist
       Used internally to display all available commands.
mem <address> [<size>]
       Print Hexdump of specified memory region.
reflect
       Just Reply the command which was entered (for testing the connection).
applist
       Print List of currently loaded applications.
pid [<app name>|<app index>]
       Dump Guids of one specific or all loaded applications.
pinf [<app name>|<app index>]
       Dump Project informations of one specific or all loaded applications.
startprg [<app name>|<app index>]
       Start one specific or all loaded applications.
stopprg [<app name>|<app index>]
       Stop one specific or all loaded applications.
resetprg [<app name>|<app index>]
       Reset one specific or all loaded applications.
resetprgcold [<app name>|<app index>]
       Perform a cold reset one specific or all loaded applications.
reload [<app name>|<app index>]
       Reload one specific or all loaded applications from their bootprojects.
getprgprop
       [not implemented, yet]
getprgstat [<app name>|<app index>]
       Get the status of one specific or all loaded applications.
plcload
       Get the Processor load of the PLC tasks.
rtsinfo
       Print Runtime System Informations, like Processor and Runtime Version.
```

Commands as below:

Pic 0-3

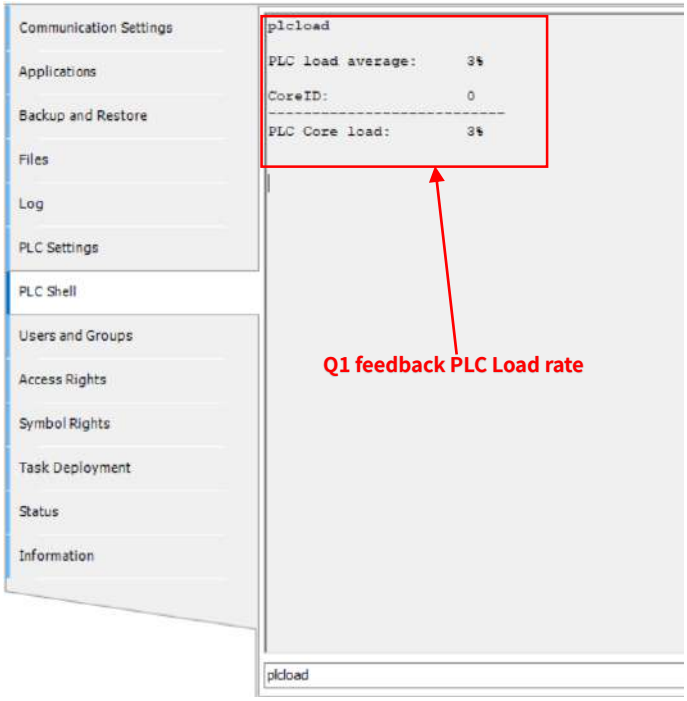
Command	Description
?	Display all valid scripting information
getcmdlist	Display all available internal commands
mem <address> [<size>]	Display memory range of hexadecimal
reflect	Return the last input instruction (mainly used for connection test)
applist	Display list of currently loaded apps
pid [<app name>|<app index>]	A specific jump or all loaded apps
pinf [<app name>|<app index>]	A specific jump project information or all loaded apps
startprg [<app name>|<app index>]	Launch a specific or all apps
stopprg [<app name>|<app index>]	Stop a specific or all apps
resetprg [<app name>|<app index>]	Reset a specific or all apps
resetprgcold [<app name>|<app index>]	Perform a cold reset for a specific or loaded application
reload [<app name>|<app index>]	Reload a specific application or application loaded from the startup project
getprgprop	Target has not been realized yet
getprgstat [<app name>|<app index>]	Get the status of a specific or all applications
plcload	Get the load rate of the current PLC task
rtsinfo	Display information of real-time nuclear operation system
channelinfo	Return communication channel information
rtc-get	Get UTC from time string
rtc-set	Set UTC by time string (obey format in "`rtc-set YYYY-MM-DDThh:mm:ss[,sss]`")
saveretains [<applicationname>]	Save retain power down hold data to a file (specific application)
restoreretains [<applicationname>]	Re-store retain power down hold data in a file (specific application)

```
Notice:
    Login of PLC is needed before user can use scripting function.
```

Example1: Check operating load rate with PLC Shell

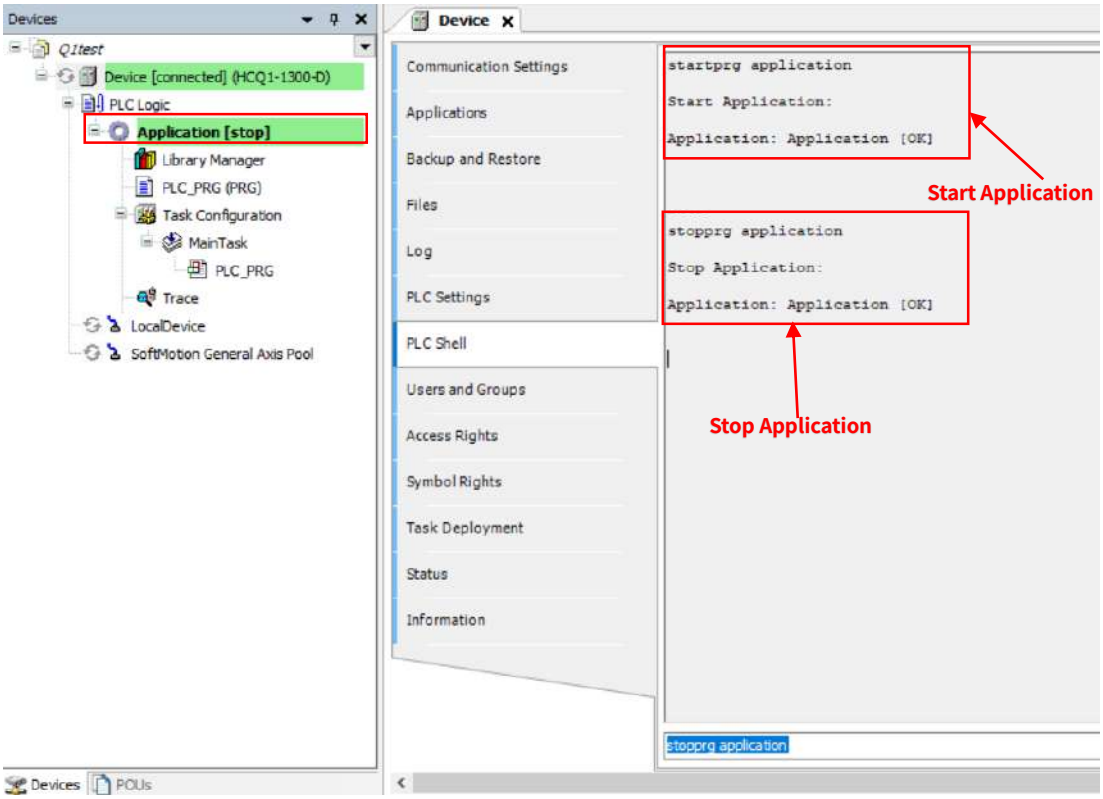
After communication with Q1 is created, in left tree menu, find PLC Shell, enter “plcload” in interface provided at right side.

Pic 0-4



【Example 3】Activate/stop applied program with PLC Shell

Enter "?" and find needed command startprg/stopprg and enter "command+program name": startprg application(case insensitive)



5.9 Implicit check function of program

Below situations may happen during programming:

- Divisor is zero during division operation.
- The pointer points to a null address during assignment
- The superscript and subscript of the array are out of bounds during the call to the array

In this case, FUN to check implicit is provided by CODESYS and user can add these fun to check edited program. Below FUN is provided in “POU to check implicit”:

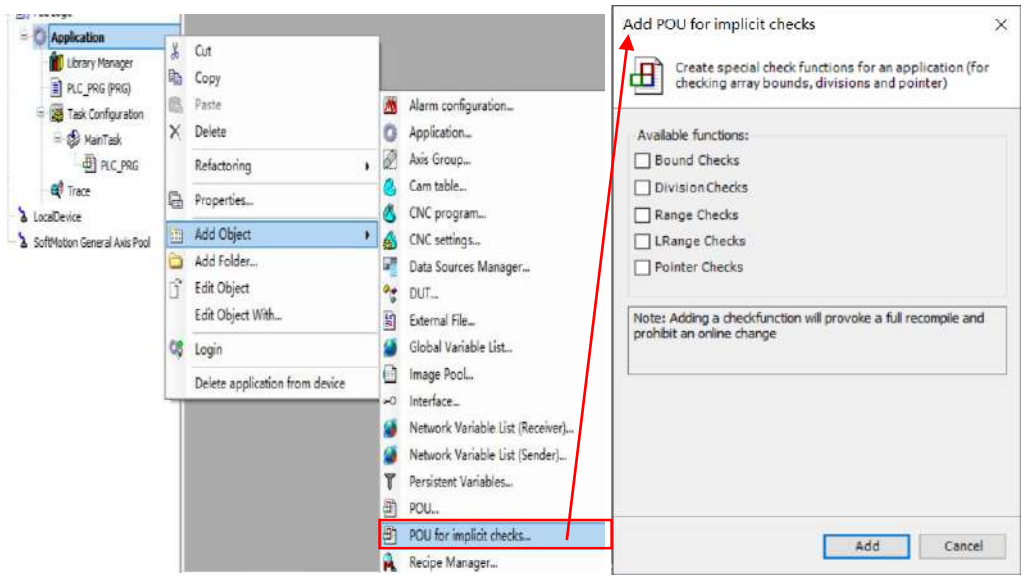
- CheckBounds
- CheckDivDInt/ CheckDivLInt/ CheckDivReal/ CheckDivLreal
- CheckRangeSigned/ CheckRangeUnsigned
- CheckLRangeSigned/ CheckLRangeUnsigned
- CheckPointer

Notice:

Above FUN will check automatically after added during program running, so user can add the FUN during debugging and delete them at actual site to save memory of CPU. And add the FUN if above error still happen during actual use.

After adding a check in the POU, it will be opened according to the selected programming language. The default programming environment of the system is st structured text.User can right-click "Application" and select "Add Object" in the pop-up shortcut menu to find "POU for implicit checks..." then the system will pop up the "Add POU for i implicit checks " dialog box. These common functions are introduced as below

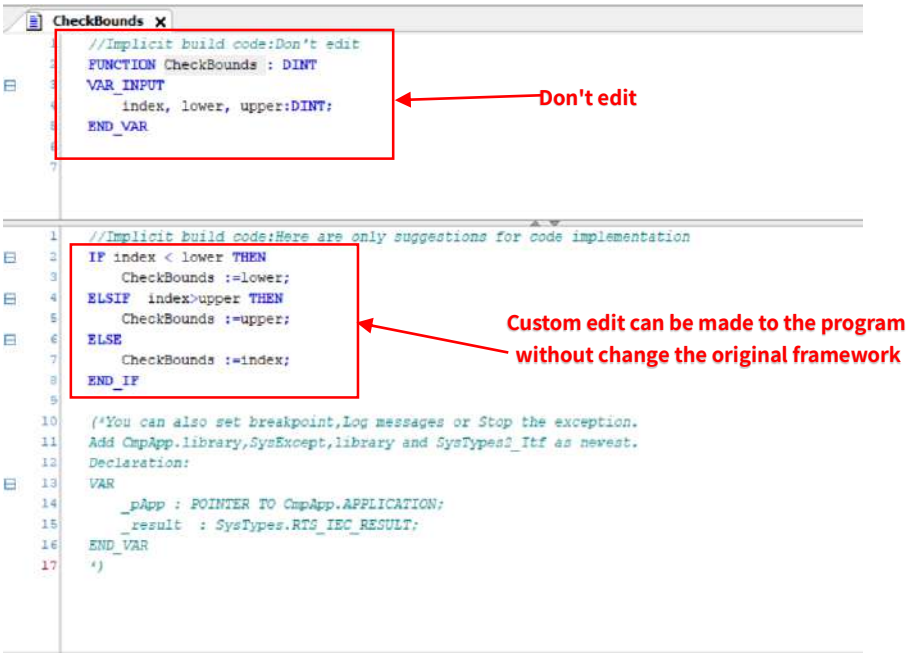
Pic 0-1



CheckBounds

This function can check whether the boundary of the array exceeds. It is usually used in applications with variable array types.It can effectively avoid the array from boundary overflow. After adding the array boundary check function, source code of the function in the program editing area as follow:

Pic 0-2



Notice:

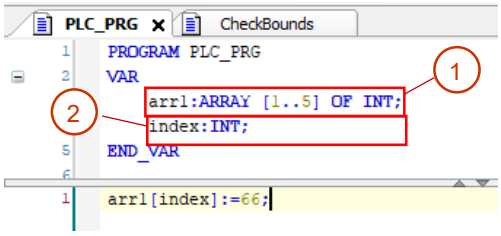
All variable declarations of the check function can't be edited. If user deletes and adds variables, the program editing will be wrong, but the program area of the check function is editable. User can self-define the program without changing original program architecture.

Below is an example to check the CheckBounds

【Example 1】Add CheckBounds and check variable array.

Add variable array first with below statement:

Pic 0-3

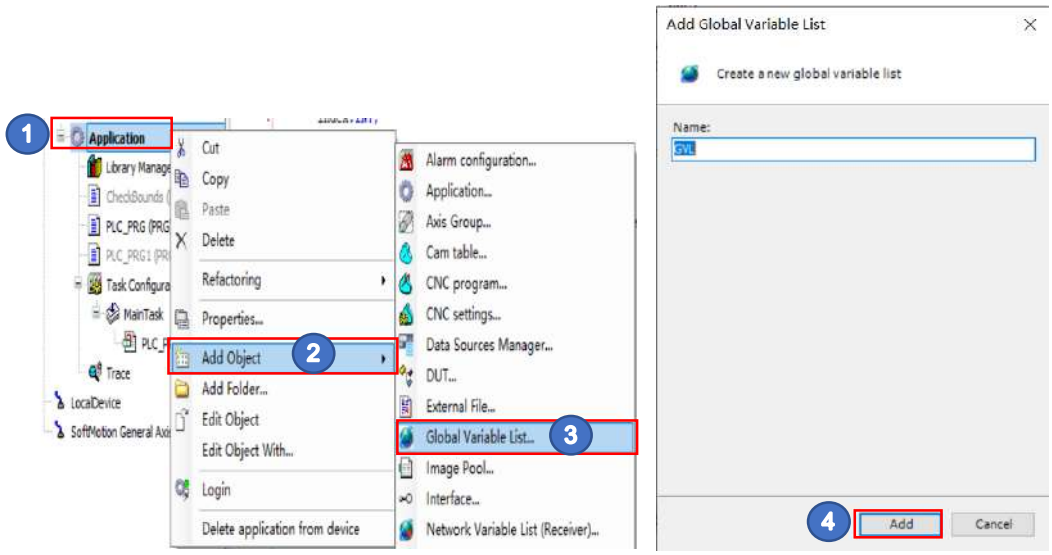


In the program, the array Arr1 has only five elements of 1 ~ 5, and the index is an INT integer variable. If no initial value is given, the default value is 0. After the program runs, if no other value of index is given, the execution result is Arr1 [0]: = 66; while the element Arr1 [0] is not defined in original array, "0" has exceeded the definition range of the array, and the running program may crash.

For above example, user can use the checkbounds function. After adding the array boundary check function, the program will automatically limit the index to "1" - "5". For those exceeding, the upper and lower limits will be selected for writing. For example, in this example, Arr1 [index] will eventually be limited to Arr1 [1]: = 66;, Not Arr1 [0]: = 66;

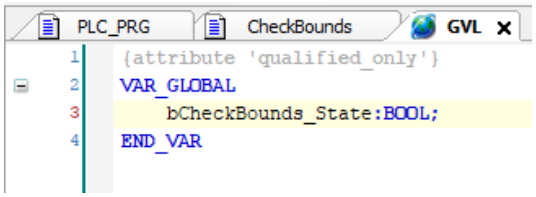
In order to check whether the array is out of bounds after online, add "bcheckbounds" to the global variable_ The "state" variable is used to display the out of bounds state of the current array. First, right-click "Application" and select "Add Object" to find the "Global Variable List" under it. After the name is given in the pop-up dialog box "Add Global Variable List", click open to create:

Pic 0-4



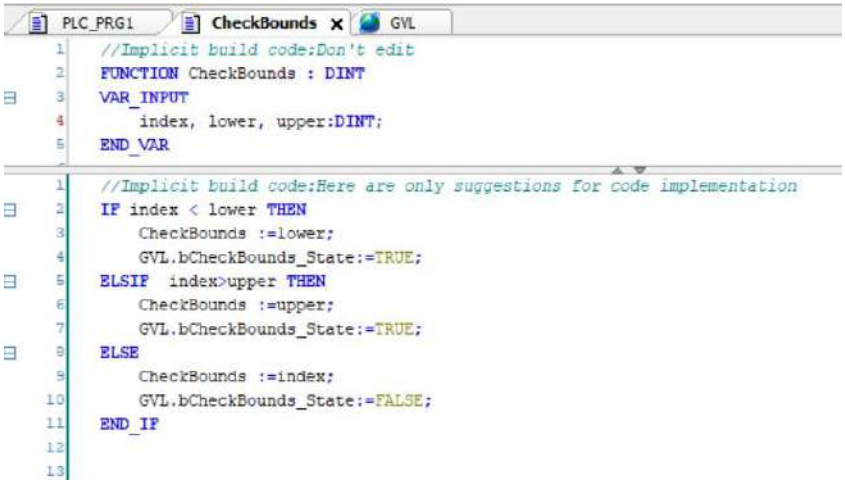
After global variable list is added, add “bCheckBounds_State” to display current status of over bounds:

Pic 0-5



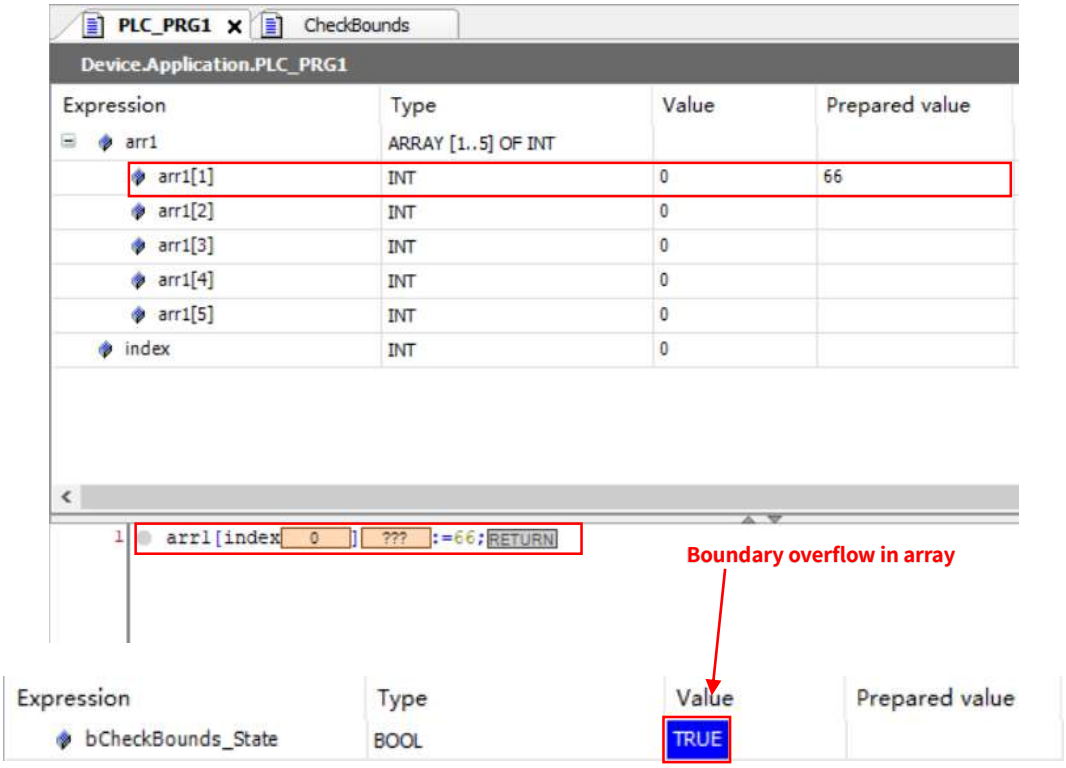
Add status variable in CheckBounds as below:

Pic 0-6



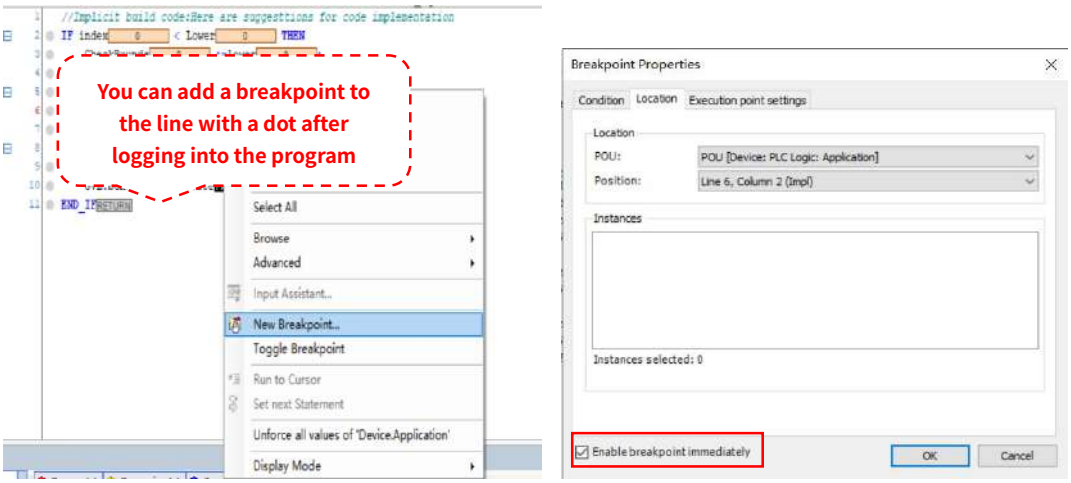
Check online running:

Pic 0-7



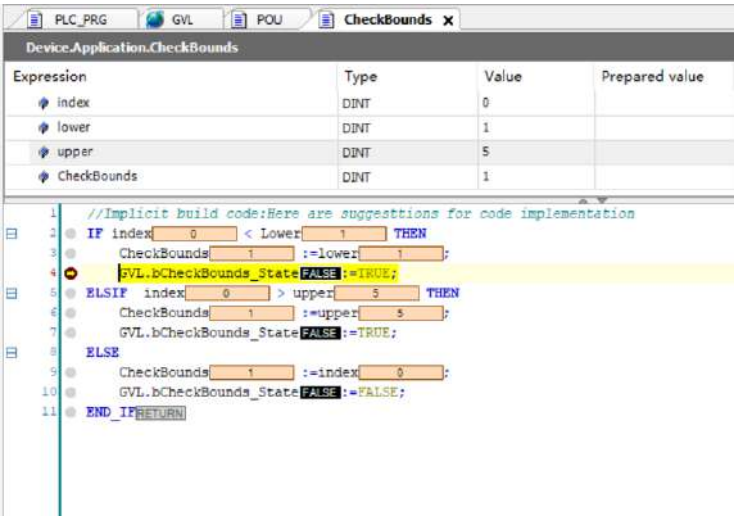
Since the check function belongs to FUN, it is impossible to directly view the value of the variable in online state. User can also add breakpoints to view the online value of the variable, select the line to add power failure, right-click and select "new breakpoint" in the pop-up shortcut menu, or directly add and activate the breakpoint through "F9" on the keyboard:

Pic 0-8



After above adding, online value of variable can be checked:

Pic 0-9



5.10 Add Extension.

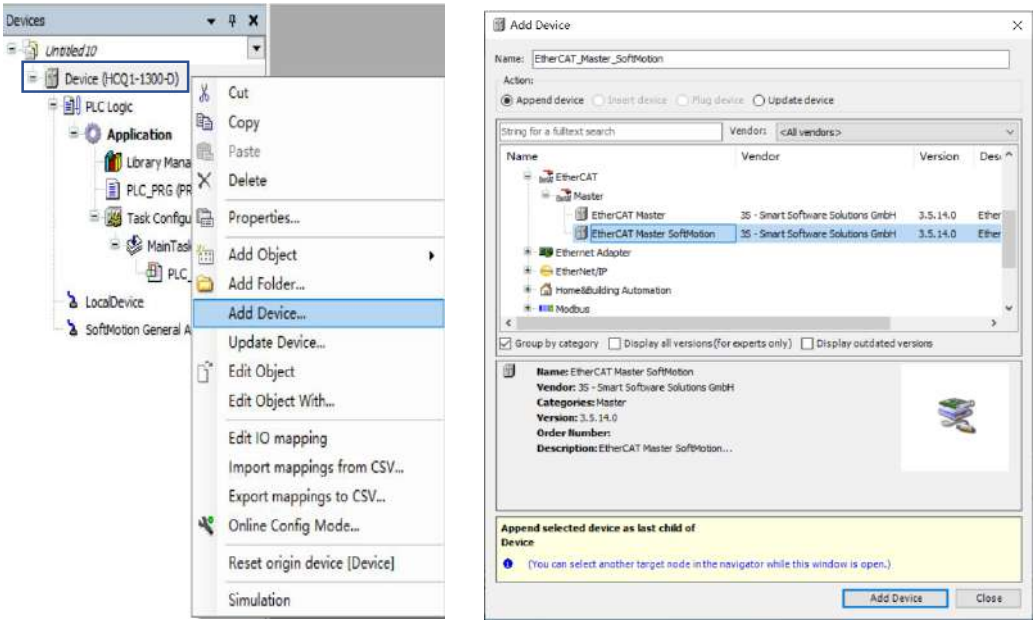
Q1 V330 supports extension of local IO at right side with maximum 16W and below chart shows power of each module(Data is in float calculation of 10%~15%, leave some space as bus of sheet-metal will occupy some power after long time use):

Pic5.10-1

序号	模块型号	参考功率
1	HCQX-EC-D	1.344W
2	HCQX-ID16-D	0.78W
3	HCQX-0D16-D	1.32W
4	HCQX-MD-D	1.032W
5	HCQX-AD04-D	1.044W
6	HCQX-DA04-D	1.056W

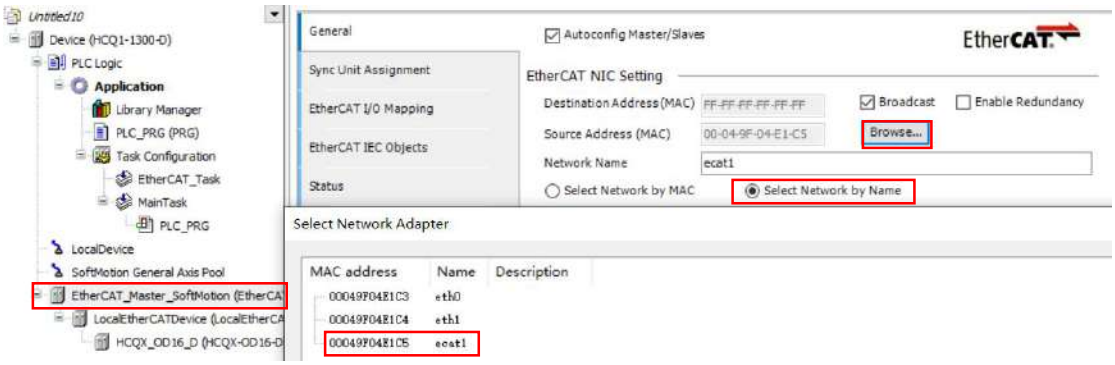
In tree menu on right, Device→Add device, select EtherCAT Master SoftMotion, click to add device.

Pic5.10-2



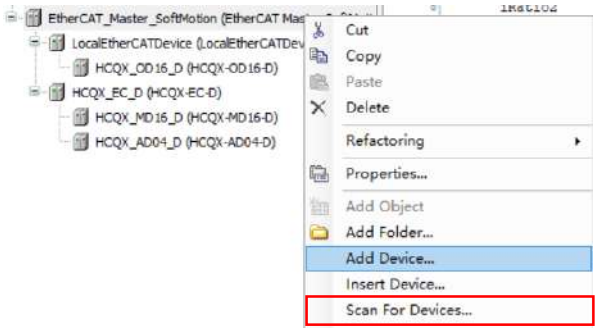
After above adding, select correct netcard and download to Q1(login Q1). It would be better to Select Network by Name to enable the program in more devices(different Mac address in different devices).

Pic5.10-3



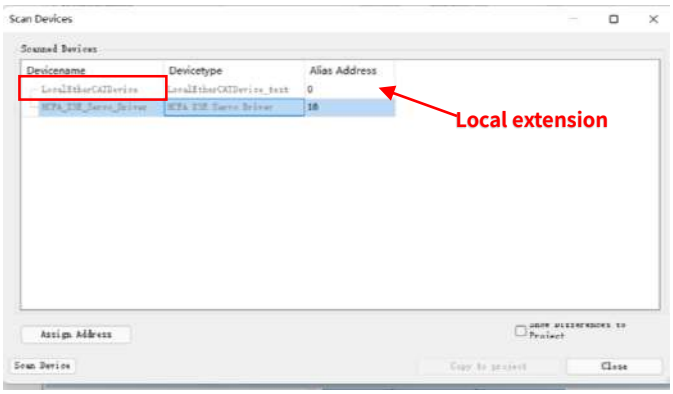
After login, click EtherCAT_Master_SoftMotion to select Scan For Devices. Adding device through scan is suggested, if not workable, add all devices to device tree.

Pic5.10-4



If unknown device appears during scan, user needs to check the description file. Modules correctly scanned as below:

Pic5.10-5



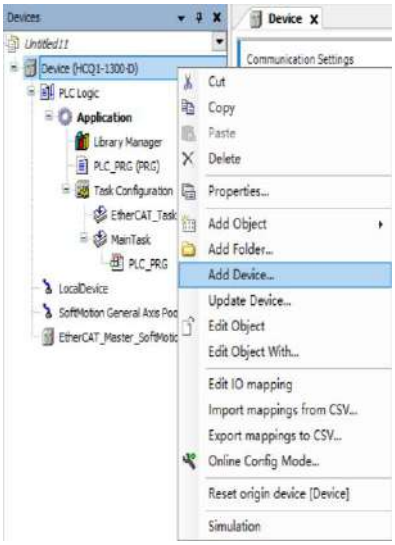
Select and copy all devices to project and then module is added. Q1 of Version 3.30 supports local extension, but LocalEtherCATDevice should be added for later module adding.

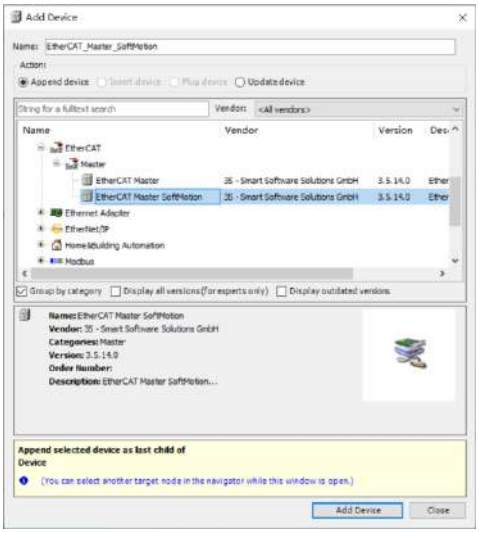
5.11 Create Motion Project

5.11.1 Add SoftMotion Project

In tree menu, right click Device→Add device, as EtherCAT is used between X3T driver and controller, in dialogue select EtherCAT Master SoftMotion and add device.

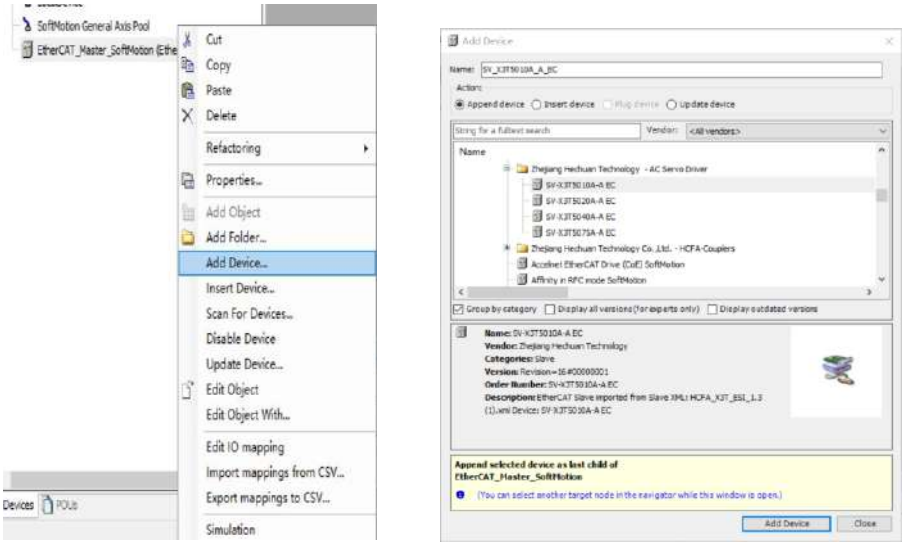
Pic 5.11-1





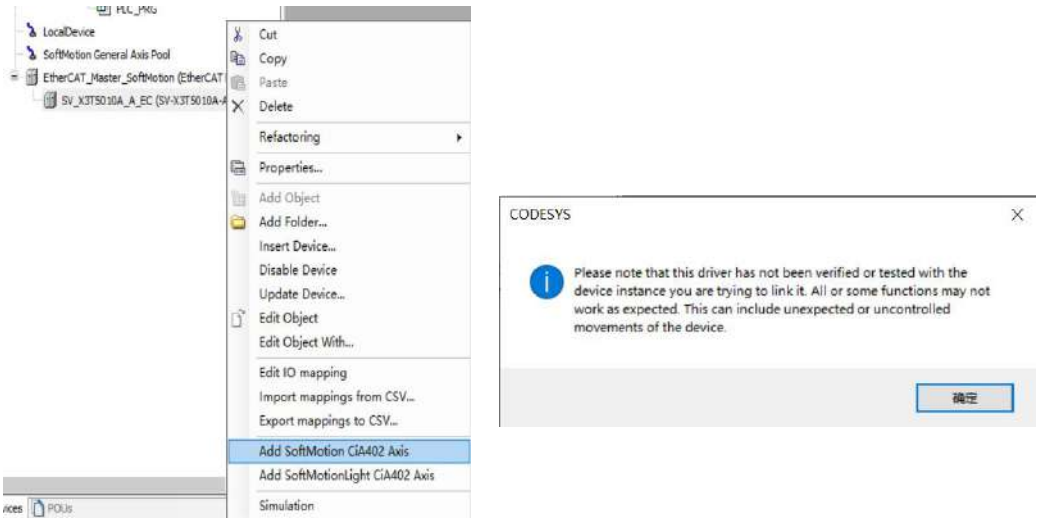
Right click in EtherCAT Master SoftMotion and right click to add device, add driver for test is needed here, find HCFA driver SV-X3T5010A-A-EC in slave and add device.

Pic 5.11-2



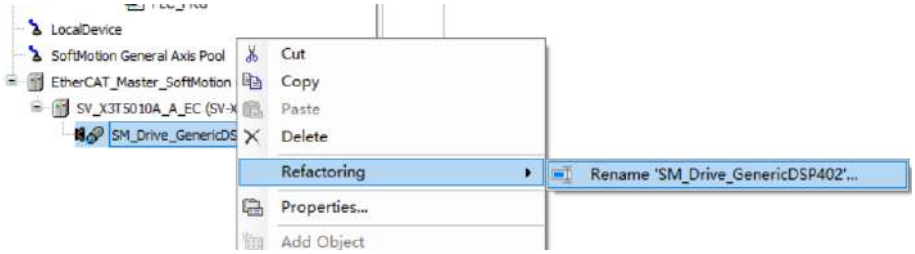
Right click added slave SV-X3T5010A-A-EC, Select Add SoftMotion CiA402 Axis and click to confirm in following dialogue.

Pic 5.11-3



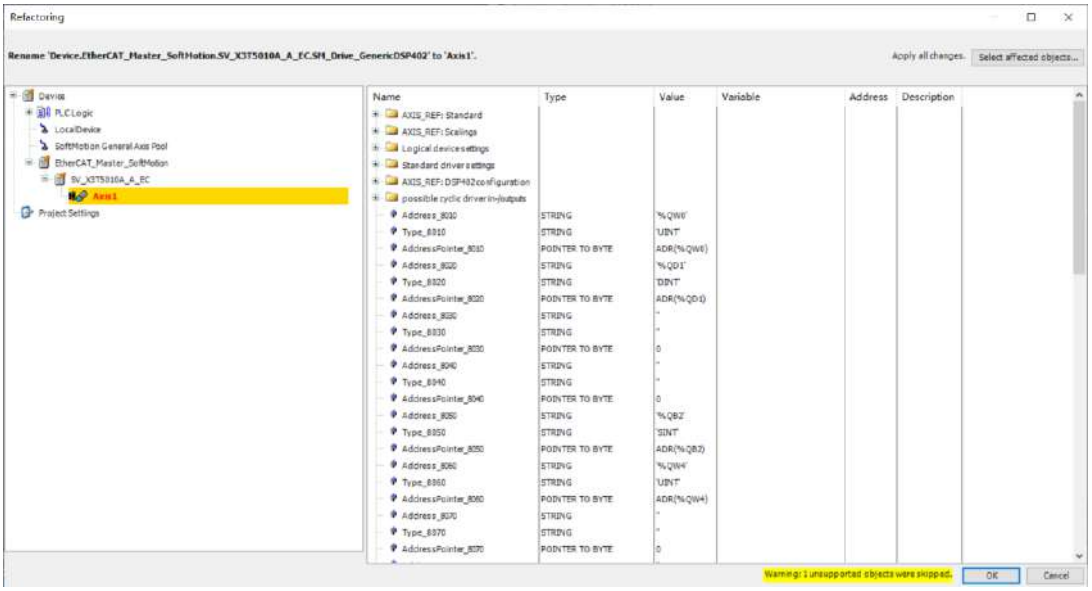
Right click to add a good SM_ Drive_ Genericdsp402, select Refactoring → rename 'SM'_ Drive_ Genericdsp402 'is modified to axis1 so that relevant axis variables can be called later in the PLC program.

Pic 5.11-4



Rename and select OK in following dialogue.

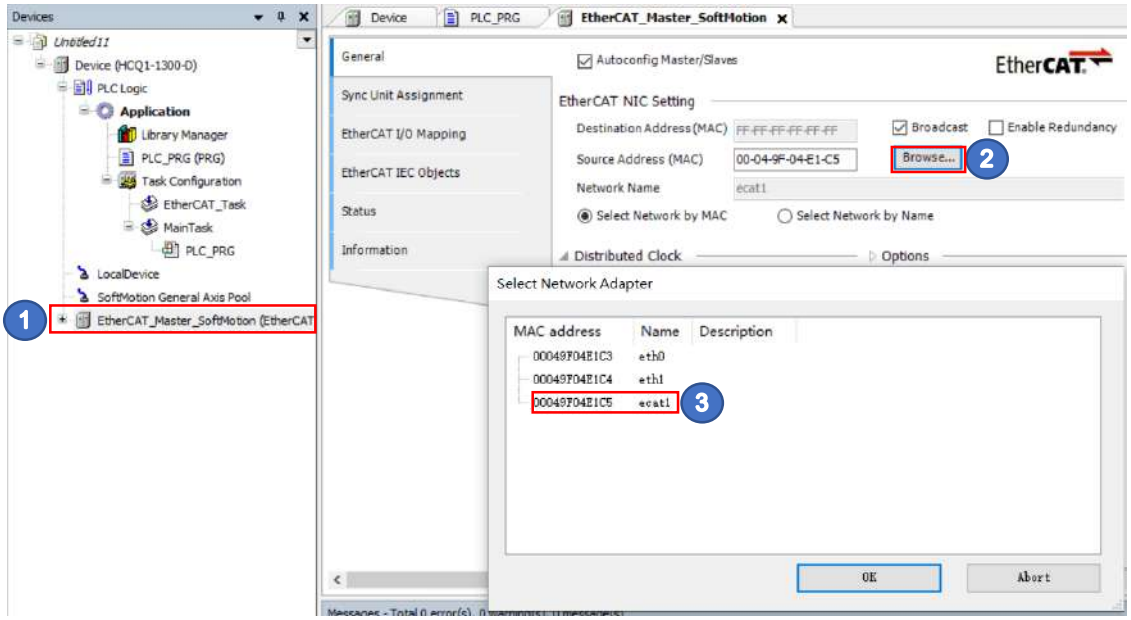
Pic 5.11-1



5.13 Modify EtherCAT master information and communication parameter

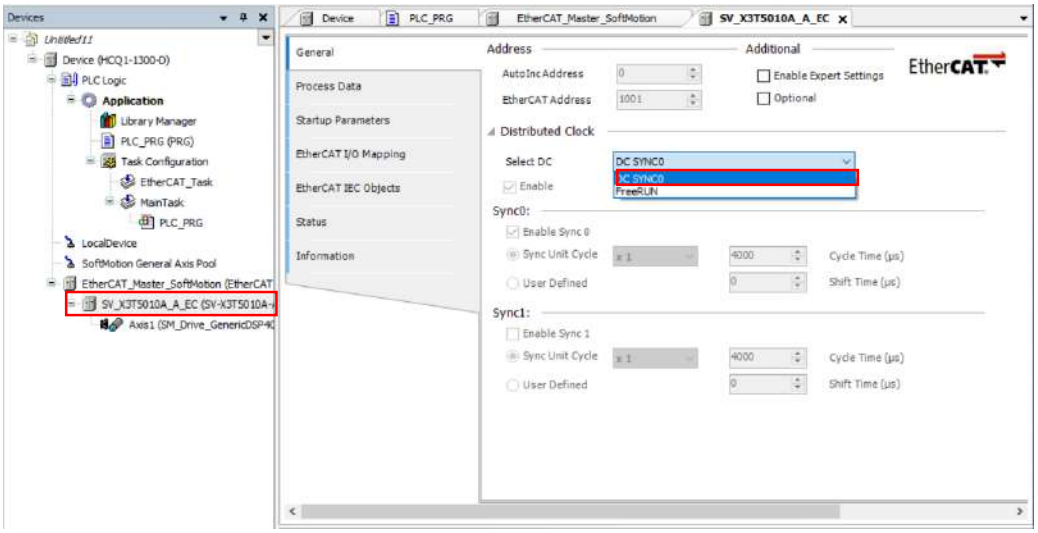
EtherCAT master added by double click needs matched network adapter for communication first. Browse to the right of the Source Address in the General page, select the Network Adapter whose name column is ecat1, and click to OK.

Pic 5.13-1



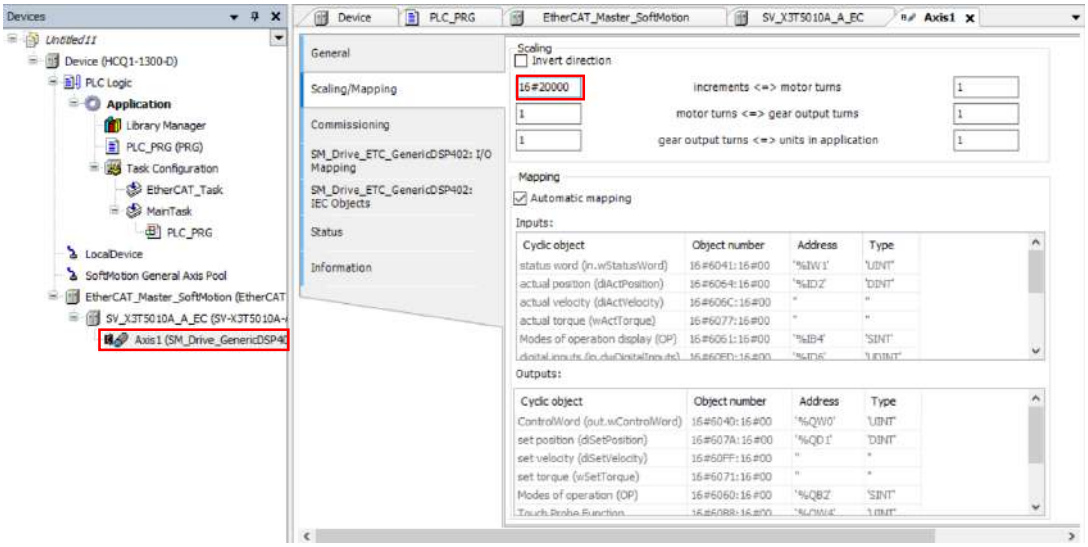
Double click EtherCAT slave SV_X3T5010A_A_EC, expand distributed clocks in the overview on the right→ select DC → DC SYNC0, select DC synchronous mode.

Pic 5.13-2



Modify parameters about encoder of Axis1, X3T is 17bit encoder, change increment to 16#20000.

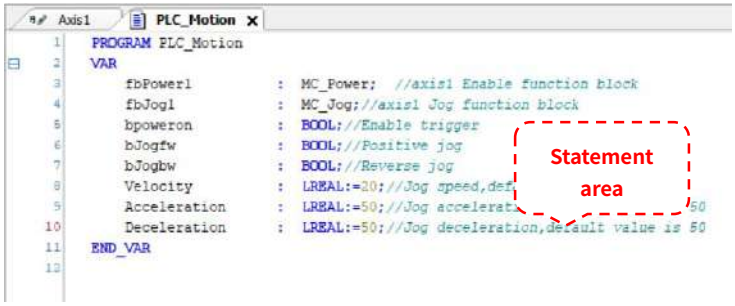
Pic 5.13-3



5.14 Realize of single axis control command

Create new POU and name it PLC_Motion to write single axis motion control command. Detailed steps refer to 4.3. Write command as below:

Pic 5.14-1

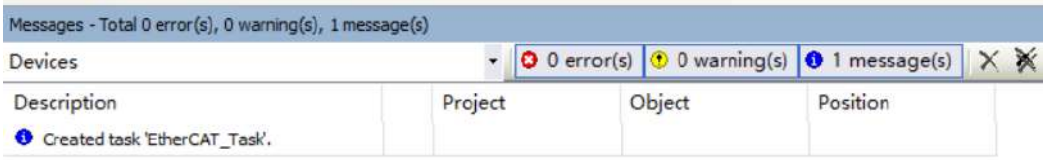


```
1   fbPower1 (
2       Axis:=axis1,
3       Enable:=TRUE,
4       bRegulatorOn:=bpoweron,
5       bDriveStart:=TRUE,  );
6   fbJog1 (
7       Axis:=axis1,
8       JogForward:=bJogfw,
9       JogBackward:=bJogbw,
10      Velocity:=Velocity,
11      Acceleration:=Acceleration ,
12      Deceleration:=Deceleration  ,
13      Jerk:= ,  );
```

Main program area

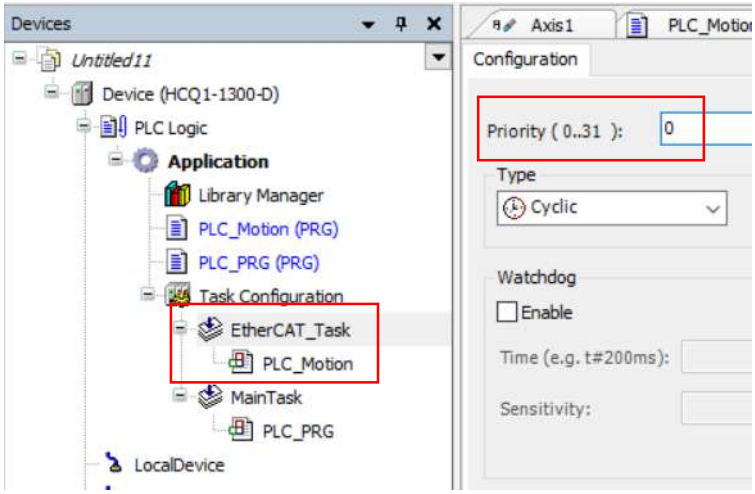
After adding Motion project in CODESYS, a EtherCAT Task will be generated automatically under task configuration.

Pic 5.14-2



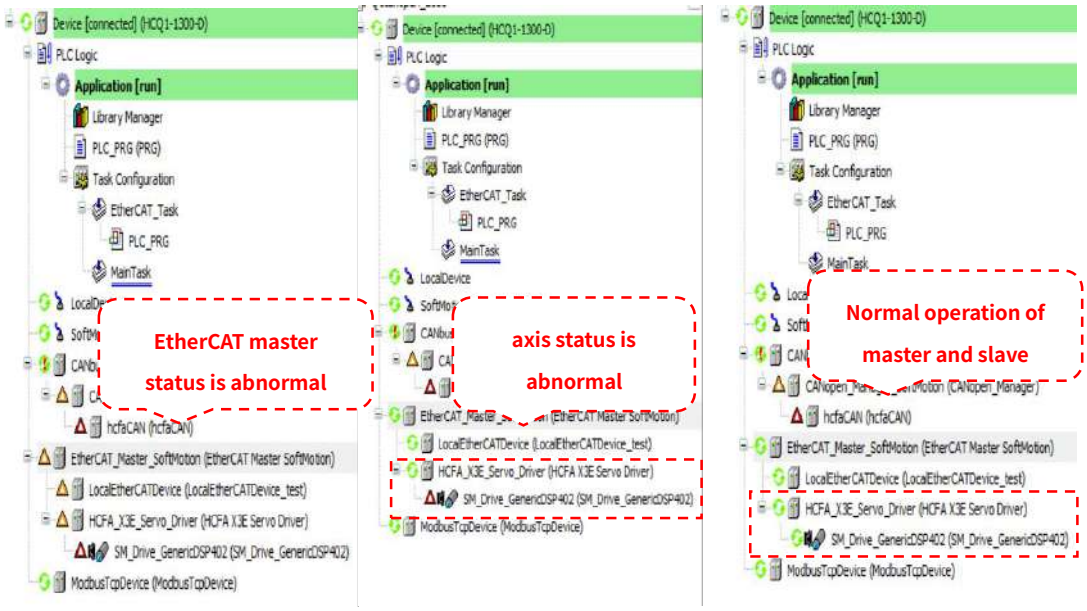
Related parameters of EtherCAT Task need manual configuration, or loss error will be reported during the operation of subsequent shafts. Double click Application→Task Configuration→EtherCAT Task, call PLC_Motion in the task, steps refer to 4.2, and set priority as “0”.

Pic 5.14-3



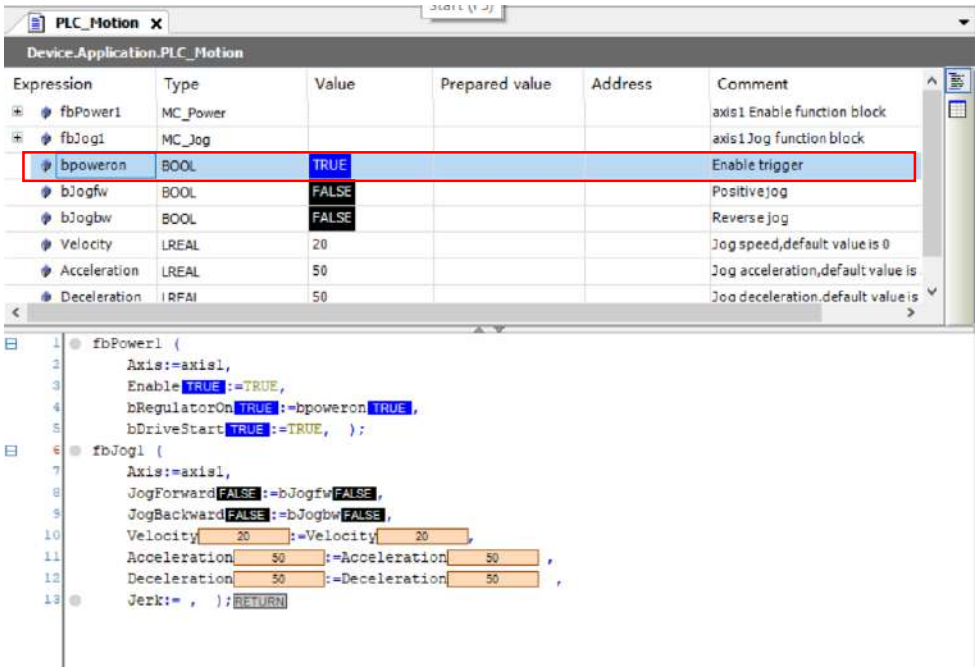
After configuration, login and run the program, wait till driver and axis status get back normal, double click PLC_Motion.

Pic 5.14-4



Enable the axis first, set bpoweron as TRUE.

Pic 5.14-5



After above operation, set bjogfw as TRUE to jog the axis in the forward direction and bjogbw to jog the axis in the reverse direction. For editing and using other axis function blocks, refer to CODESYS for online help：https://help.CODESYS.com/

Chapter 6 Communication Setting

6.1 Configuration of Ethernet TCP/IP

6.1.1 TCP/IP Overview

TCP/IP protocol, also called TCP/IP protocol stack or internet protocol series, contains a series of network protocols as foundation. These protocols originated from internet project DARPA of the US Department of Defense. It represents literally: TCP transmission control protocol and IP internet protocol.

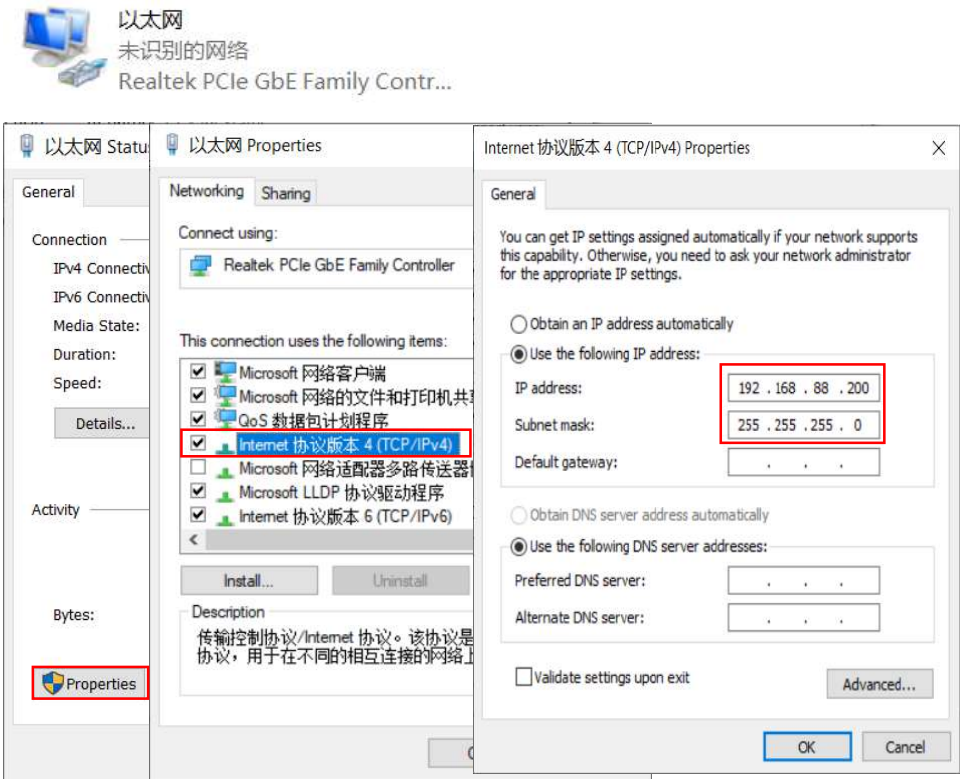
Ethernet port serves Ethernet TCP/IP protocol, and it's used for communication between Q1 and PC. Install CODESYS on PC and user can then write in Q1 program, login or change variable after communication is done.

6.1.2 TCP/IP Protocol application

Port1 port and port2 port of Q1 controller can communicate with the computer through TCP / IP protocol. The default IP address of port1 is 192.168.188.100, and the subnet mask is 255.255.255.0; The default IP address of port2 is 192.168.88 100, , and the subnet mask is 255.255.255.0

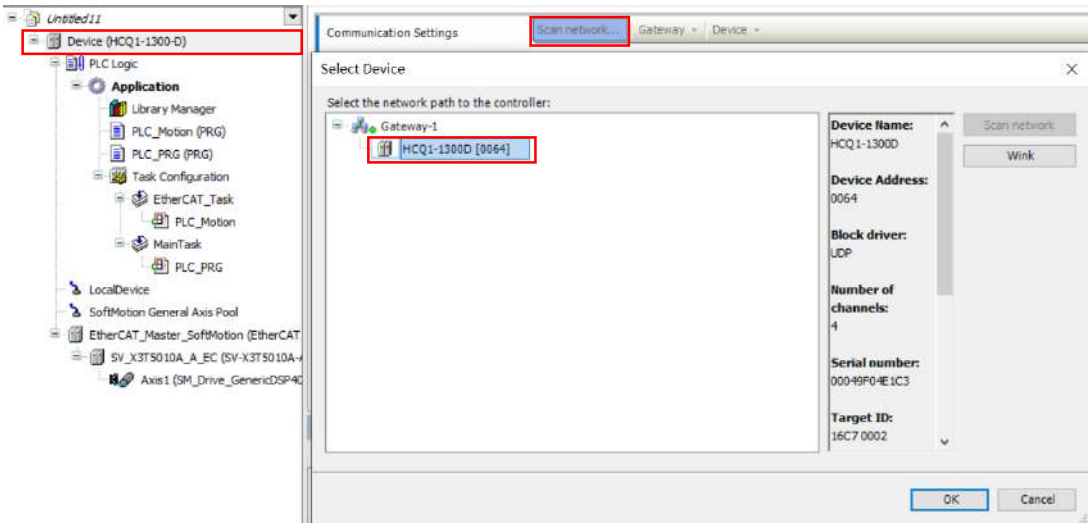
Modify PC IP address first.

Pic6.1-1



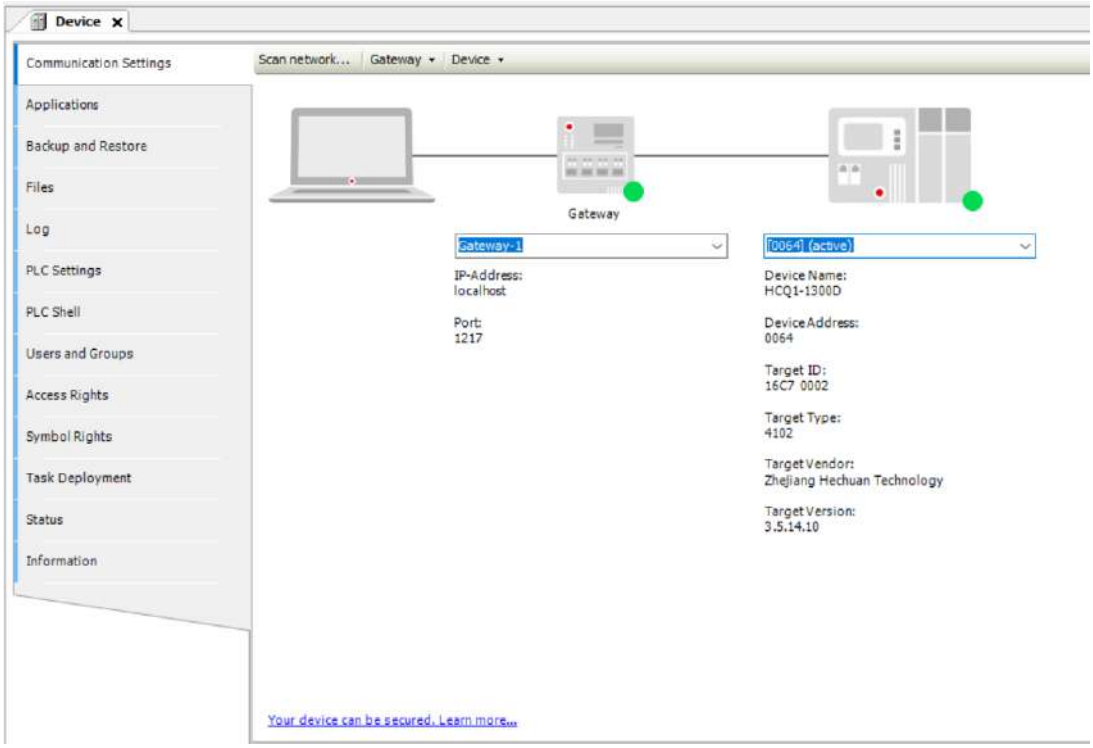
Double click the tree menu "Device" on lower left the new project of Codesys software to enter Communication Settings. After ensuring that the gateway is opened correctly, click "Scan network". After scanning to Q1, select the device and click OK to add

Pic6.1-2



Software displays as below after communication is complete:

Pic6.1-3



Program Logining and Watching are also based on Ethernet TCP/IP communication, refer to 5.6.

6.2 Modbus TCP Configuration

6.2.1 Modbus protocol overview

Modbus protocol was developed by Modicon (now a brand of Schneider Electric) in 1979 and is a general language applied to electronic controllers.

Controllers can communicate with each other, controllers and other devices via network through Modbus protocol. The Modbus network consists of one MODBUS master and multiple slaves.

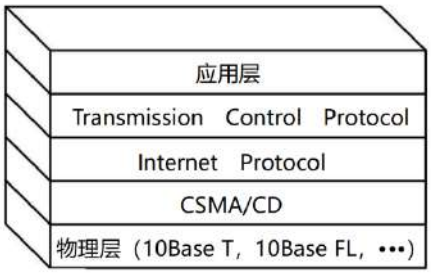
6.2.2 Modbus TCP overview

Modbus TCP is a Modbus communication protocol based on Ethernet TCP / IP, and its communication interface is standard Ethernet RJ45 port. Port1 and port2 of Q1 support standard Modbus TCP master / slave protocol to realize communication with touch screen, frequency converter and other devices. Each slave station can add up to 50 read-write channels.

6.2.3 Modbus TCP Model

- Physical layer: Provide port of device
- data link layer: Transmitting data frames in the same network
- Network layer: Implement IP packet with 32-bit IP address
- Transmission layer: Realize reliable connection, transmission, Port service and transmission scheduling
- Application layer: Modbus protocol message

Pic6.2-1



6.2.4 Modbus TCP Data message structure

Modbus is a request / response protocol and provides the services specified by the function code. The message structure of Modbus TCP is introduced below.

Take below request and response as example:

Request: 97 76 00 00 00 06 04 04 00 00 00 01

Response: 97 76 00 00 00 05 04 04 02 12 34

Request message:

97 76 00 00 00 06 04 04 00 00 00 01				
	Example	Length	Instruction	Remark
Map message head	0x97	1	transaction ID Hi	Client initiated, server replicated, for transaction pairing
	0x76	1	transaction ID Lo	
	0x0000	2	protocol identifier	Client initiated, server replicated Modbus protocol= 0.
	0x0006	2	length	From next to last in this byte
	0x04	1	unit identifier	Client initiated, server replicated Identification of remote terminal on serial link or other bus
Function code	0x04	1	function code, read register	refer to standard modbus protocol
Data	0x0000	2	Starting address	
	0x 0001	2	number of registers	
Check				

Response message:

97 76 00 00 00 05 04 04 02 12 34				
	Example	Length	Instruction	Remark
Map message head	0x97	1	transaction ID Hi	Client initiated, server replicated, for transaction pairing
	0x76	1	transaction ID Lo	
	0x0000	2	protocol identifier	Client initiated, server replicated Modbus protocol= 0.
	0x0005	2	length	From next to last in this byte
	0x04	1	unit identifier	Client initiated, server replicated Identification of remote terminal on serial link or other bus
Function code	0x04	1	function code, read register	refer to standard modbus protocol
Data	0x02	1	Number of byte	
	0x ----		Data	Data in this message is 12 34
Check				

Definition of part function code as below:

Function code	Description	Visit type	Q1 add.	Data type	Operation number
0x01	Coil	read	%QX0.0 – %QX8191.7	bit	single/multi
0x02	Discrete input	read	%IX0.0 – %IX8191.7	bit	single/mult
0x03	Holding register	read	%MW0 - %MW65535	byte	single/mult
0x04	Inputs register	read	%MW0 - %MW65535	byte	single/mult
0x05	Single Coil	write	%QX0.0 – %QX8191.7	bit	single
0x06	Single register	write	%MW0 - %MW65535	bit	single
0x0F	multiple Coils	write	%QX0.0 – %QX8191.7	bit	multi
0x10	multiple Holding registers	write	%MW0 - %MW65535	byte	multi
0x17	multiple Holding registers	read/ write	%MW0 - %MW65535	byte	multi

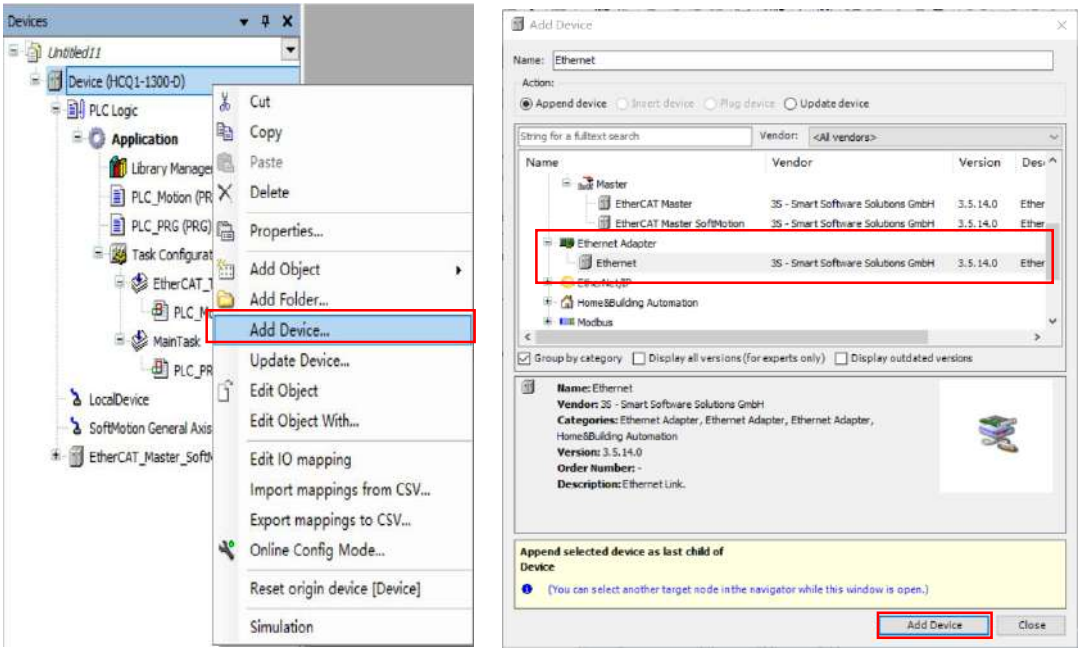
6.2.5 Modbus TCP Master application

Q1, as Modbus TCP master, it can use functions of CODESYS. Below is instruction of how to establish Modbus TCP master application:

Create new project and complete communication between it and Q1. Details refer to 5.1 and 5.2.

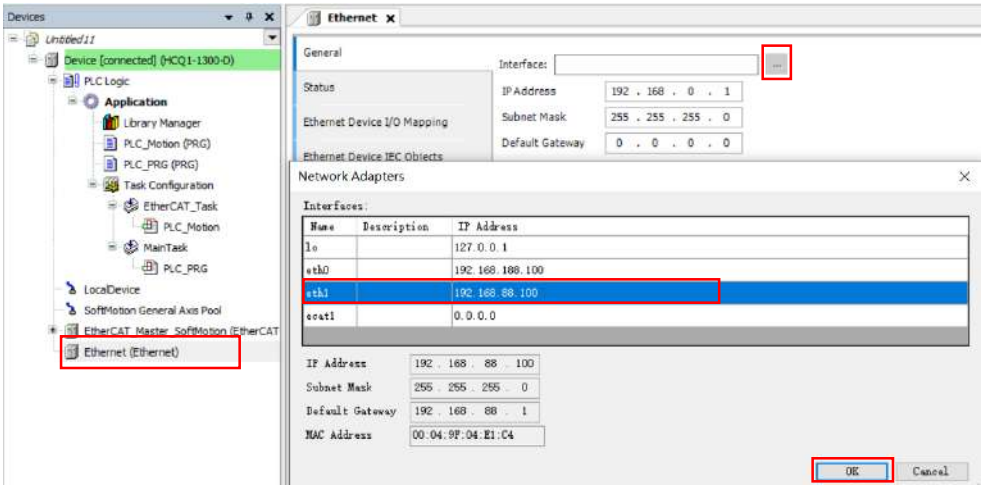
Right click the tree menu Device → Add Device, select Ethernet under "Ethernet Adapter" in the pop-up dialog box, and select " Add Device ".

Pic6.2-2



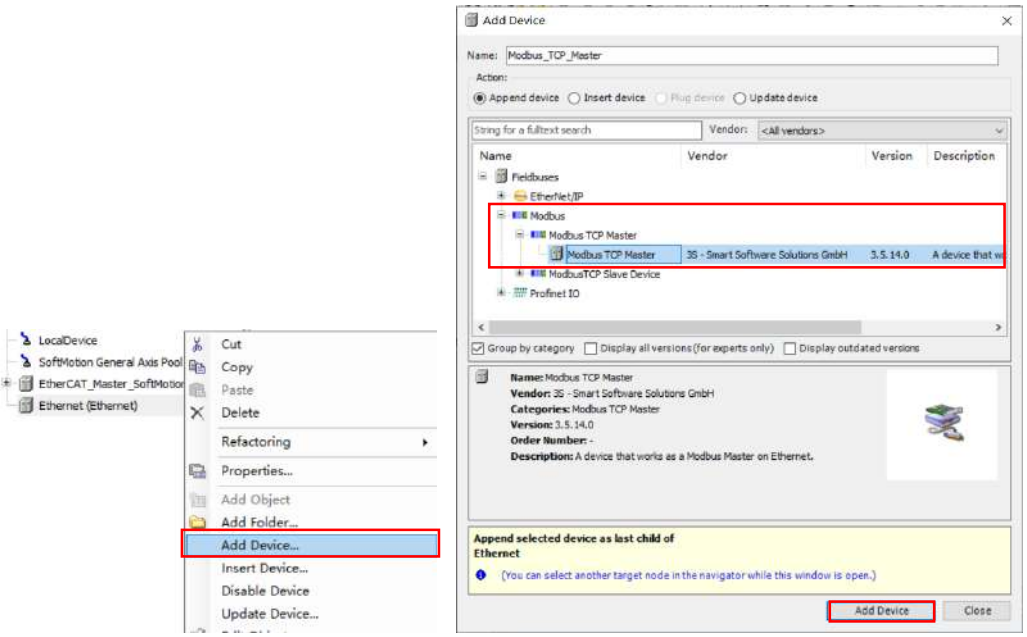
Click the Ethernet after adding, click the button on the right side of the interface in the General setting, select lan1 (port1 port of Q1) in the pop-up window, and click OK to confirm.

Pic6.2-3



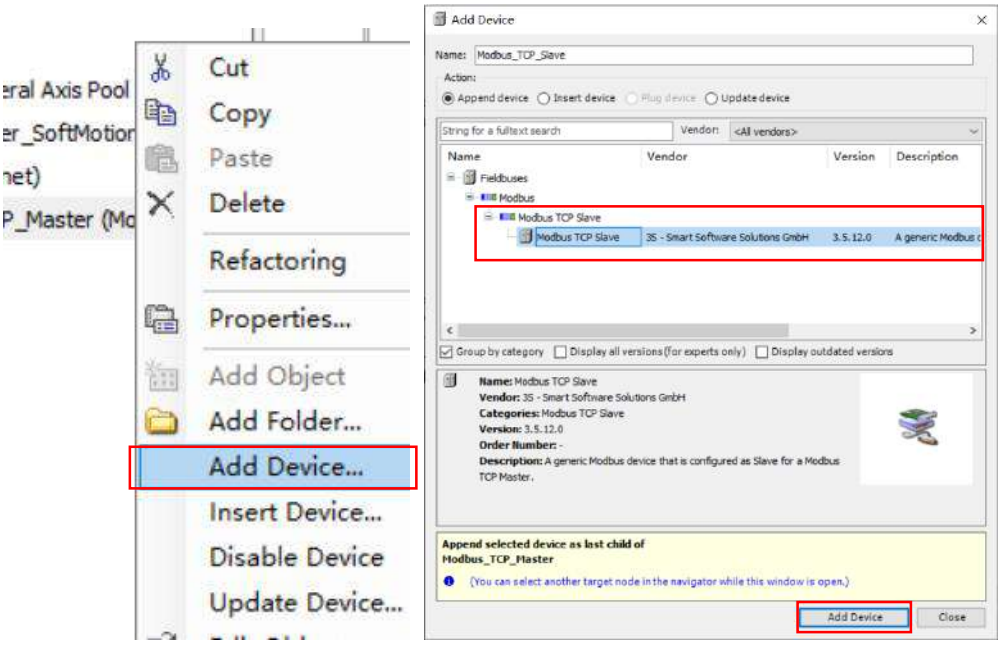
In tree menu, right click Ethernet→Add Device, then in dialogue, select Modbus→ModebusTCP Master→Modbus TCP Master, click " Add Device " to confirm.

Pic6.2-4



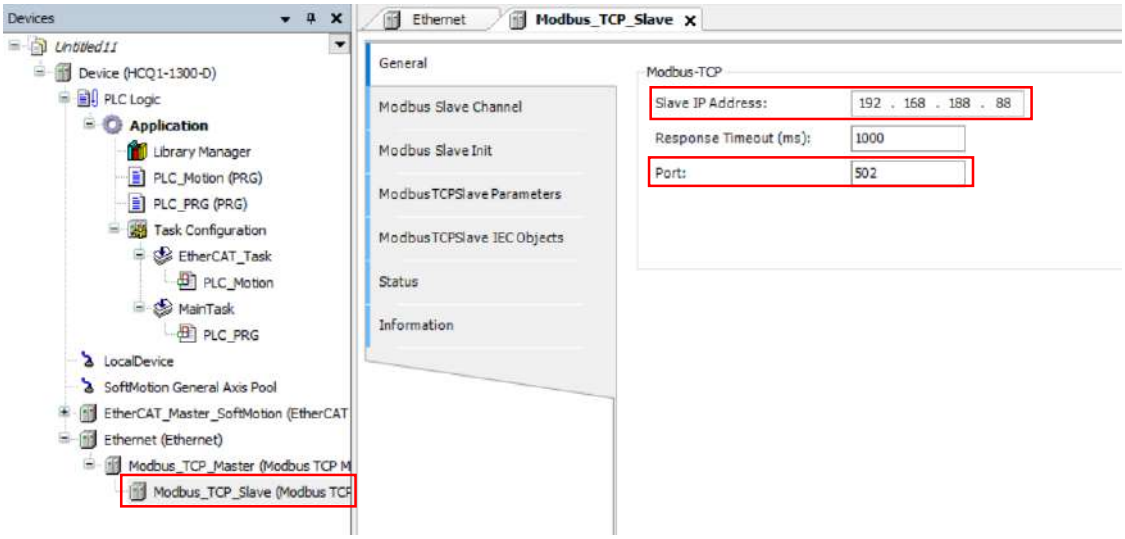
Then configuration of slave information. Right click Modbus TCP Master → Add Device, select Modbus TCP Slave in the pop-up window, and click " Add Device " to confirm.

Pic6.2-5



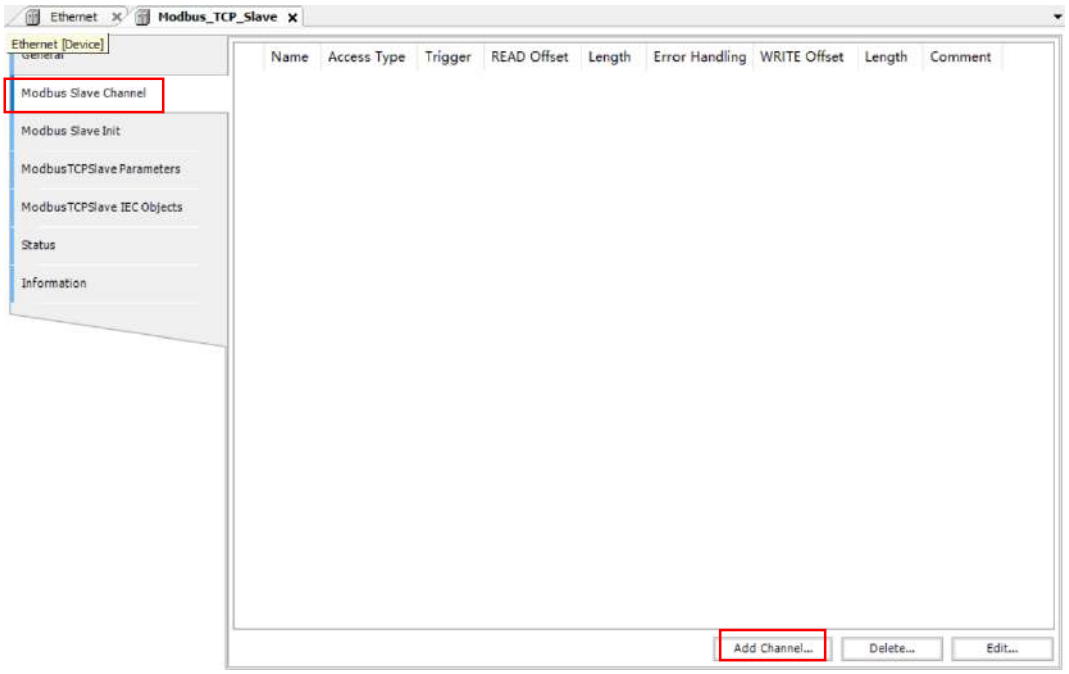
Click Modbus TCP Slave after adding, and on the General tab, modify the slave IP address to 192.168.188.88 (the slave device IP address ensures the same network segment), and set the port to 502

Pic6.2-6



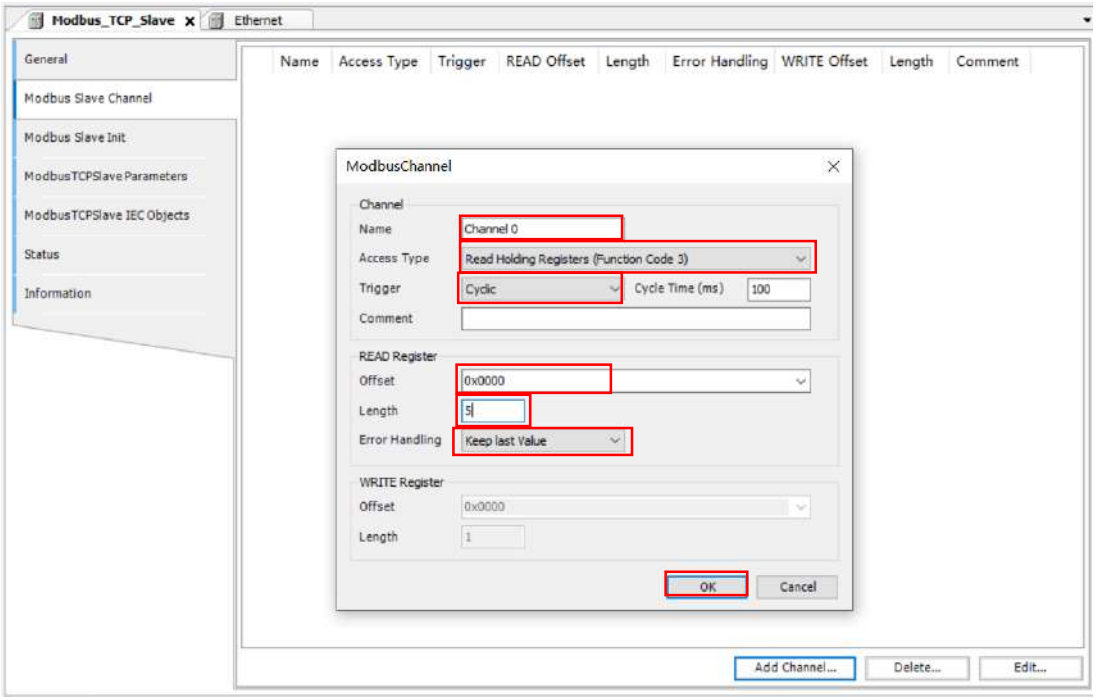
Click on Modbus slave channel→Add Channel

Pic6.2-7



Set channel Name, Access type, Offset and Length. In this test, set the channel name as channel0, the access type as Read Holding Registers, the length as 5, and the offset as 0. Click OK to establish the channel.

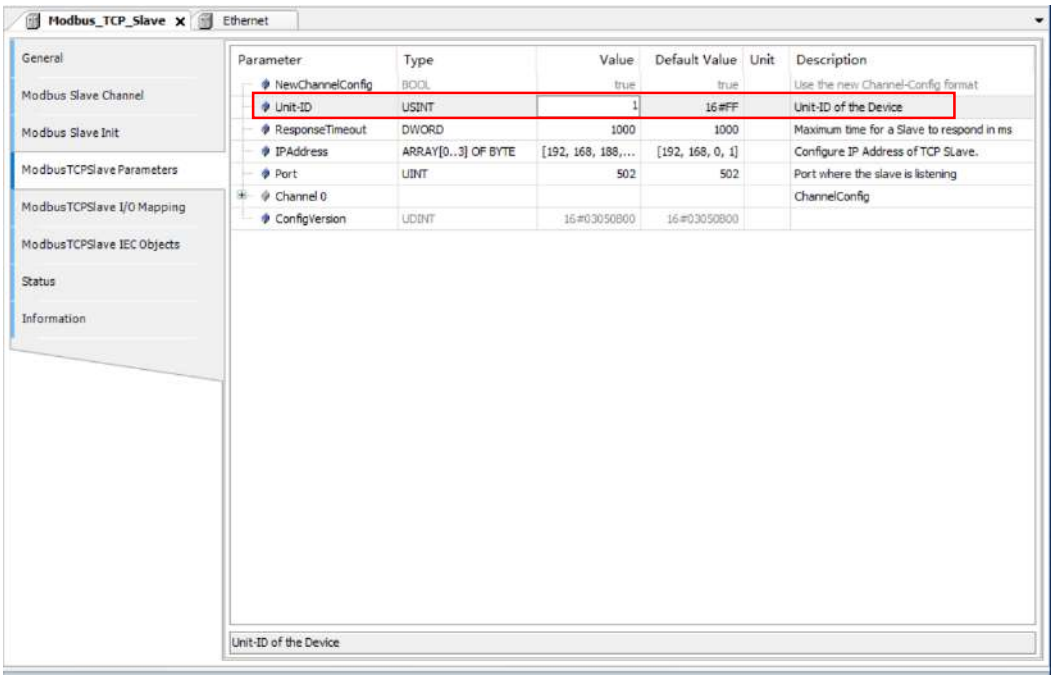
Pic6.2-8



Item	Type	Instruction
Name		Name of channel
Access type	Read Coils/Read Discrete Inputs/Read Holding Registers/ Read Inputs Registers/Write Single Coil/ Write Single Register/ Write Multiple Coils/ Write Multiple Registers/ Read or Write Multiple Registers/	9 options of type
Trigger	Cyclic / Rising edge / Application	Trigger type
Cycle		Trigger cycle
Offset		Starting address of the master and slave is offset, which can be set
Length		Indicate the length of data read
Error processing	Keep the last value / set as 0	Value of the register when an error occurs

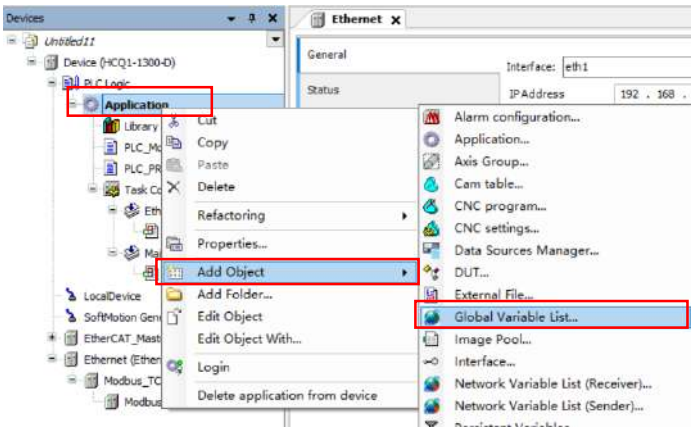
Click on ModbusTCPSlave configuration → modify the unit ID (slave site address) to 1.

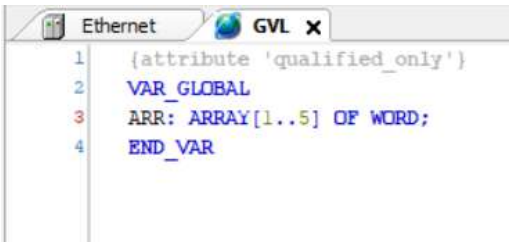
Pic6.2-9



Right click the tree menu Application to Add Object, Global Variable List and create a new array variable as follows:

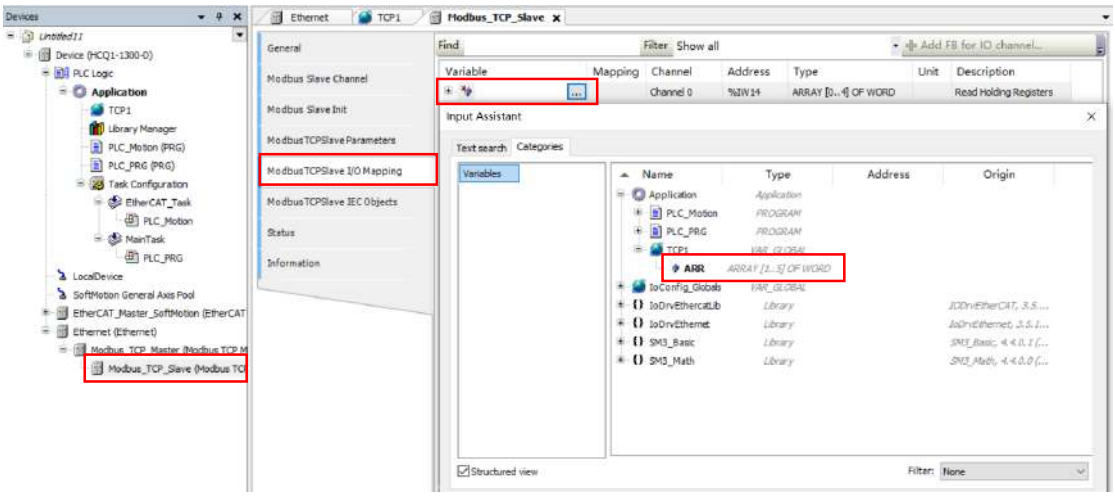
Pic6.2-10





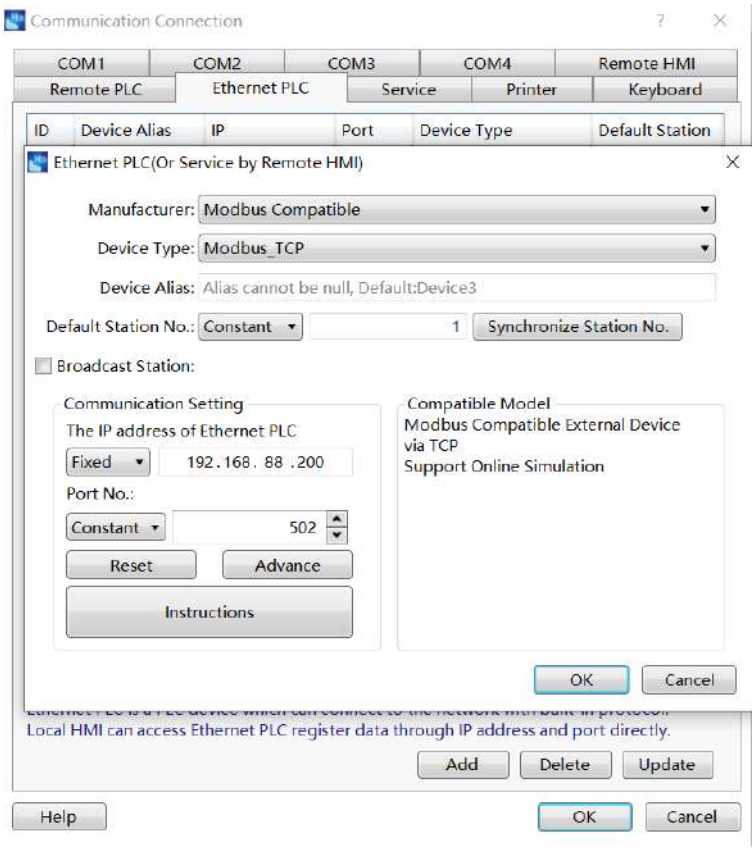
Click ModbusTCPSlave, open the ModbusTCPSlave I / O Mapping, perform variable mapping, and click OK to confirm, as shown in the figure:

Pic6.2-11



Set communication connection on touch screen software (HCTDesigner) and write the content to download to TP2000. IP address is the address of the LAN port connected between the touch screen and Q1.

Pic6.2-12

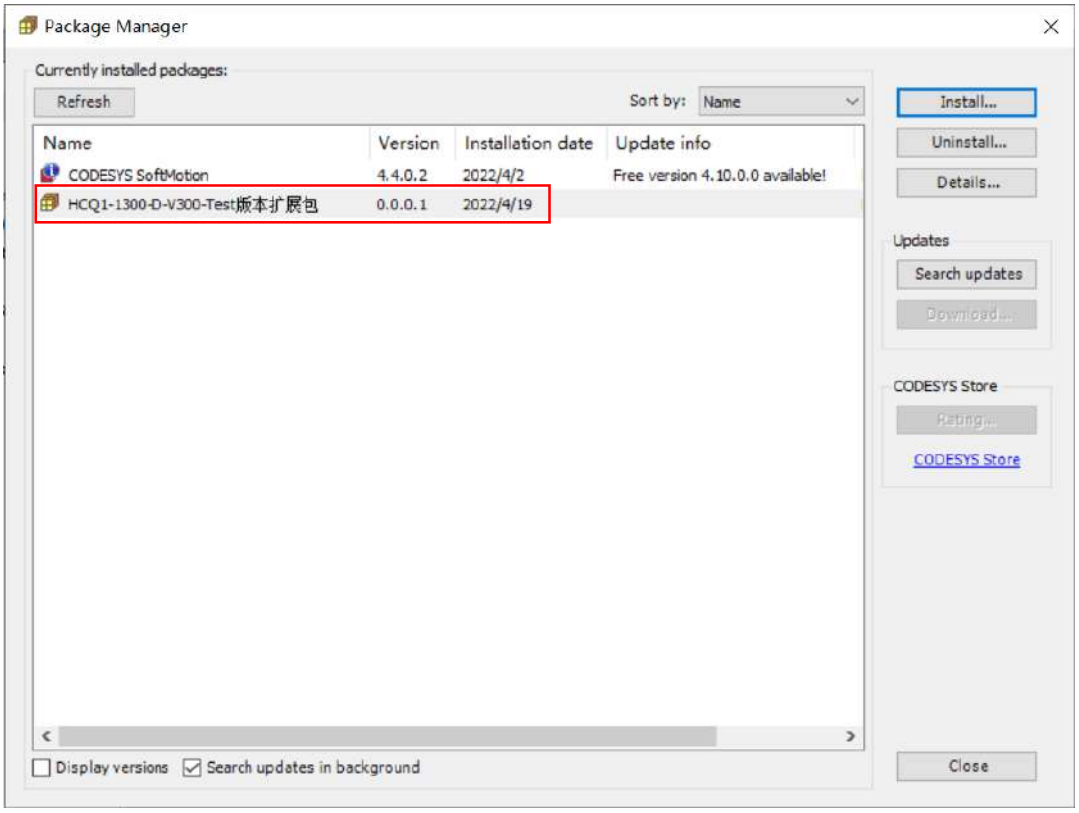


Download program to Q1 to finish communication.

6.2.6 Modbus TCP Slave Application

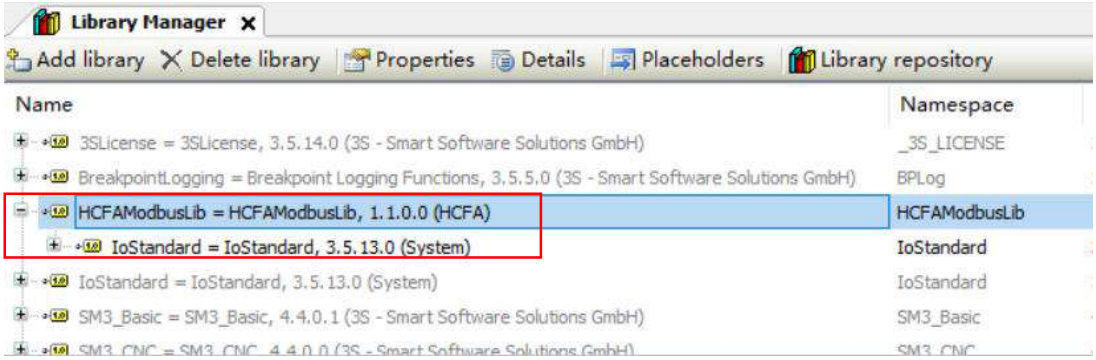
To use Modbus communication function, first install the latest version of Q1 software package, details refer to 3.2.

Pic6.2-13



Shown as below after installation,

Pic6.2-14



Modbus TCP FB and instruction as below:

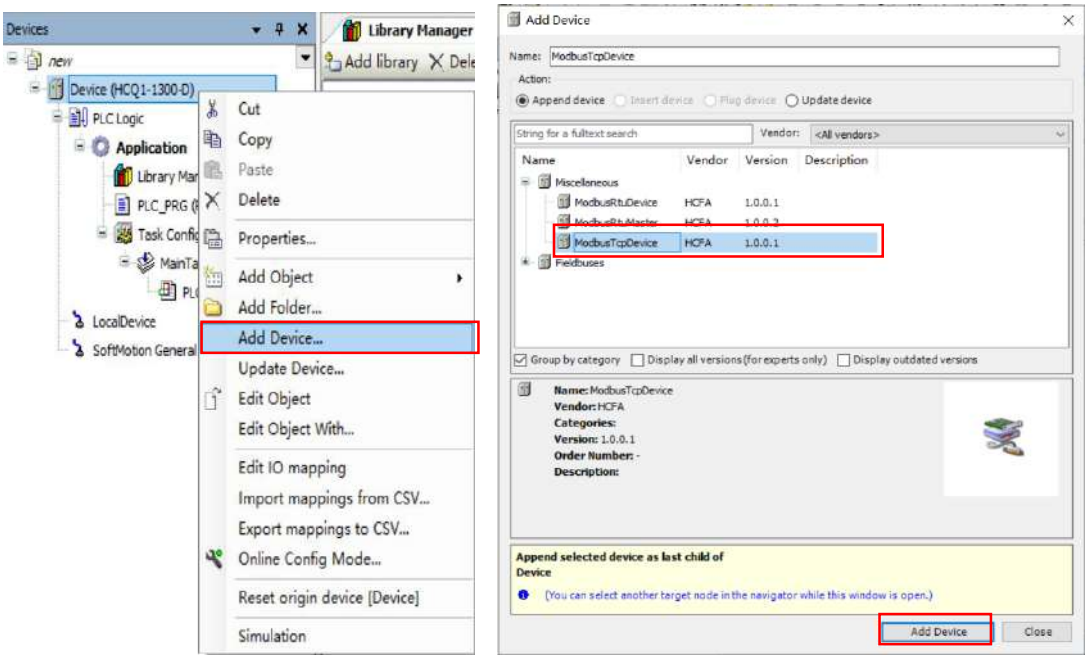
Pic6.2-15

Name	Type	Input/Output	Instruction
bRestart	BOOL	in	Restart slave device and reset error code

bDone	BOOL	out	Slave device is turned on successfully, and execution is completed and set to true
bError	BOOL	out	Error code
wErrID	MODBUS_Error	out	Error code

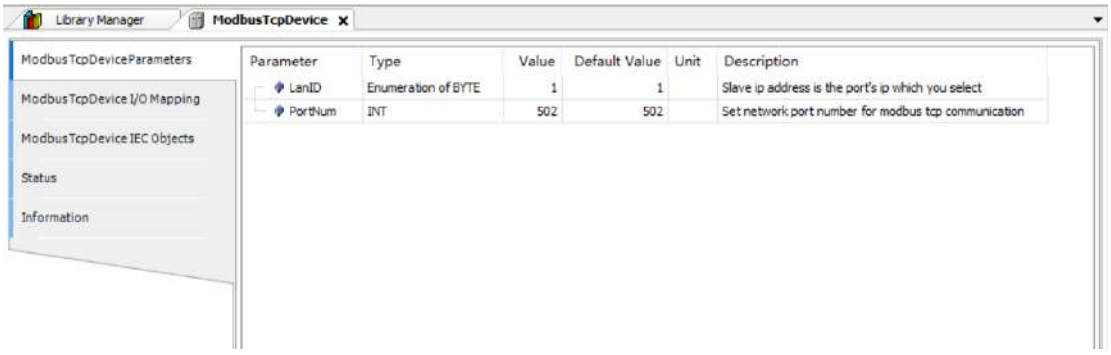
Right click the tree menu Device → Add Device, select ModbusTcpDevice in miscellaneous, click to Add device

Pic6.2-16



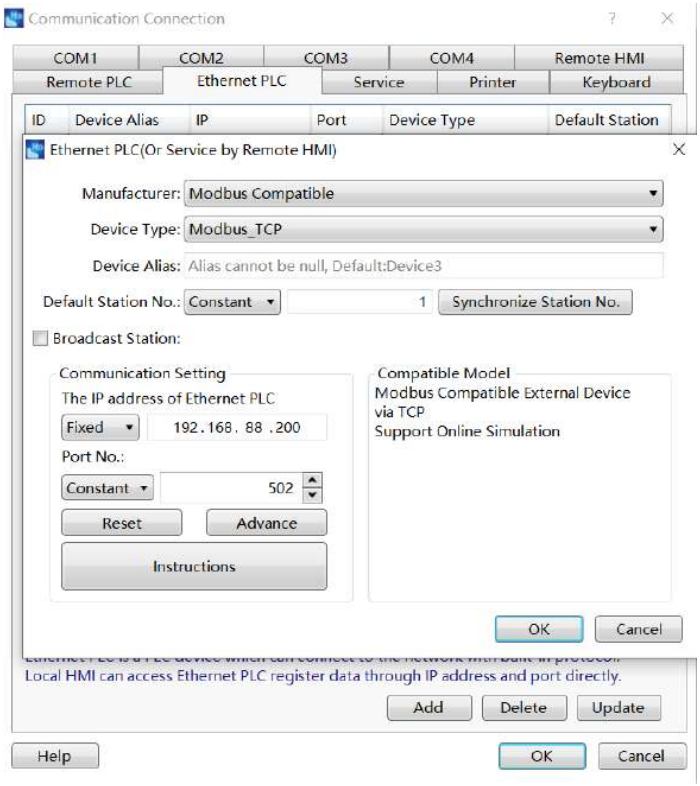
Click on ModbusTcpDevice after adding and select ModbusTcpDevice configuration. LanID specifies that IP address of the communication network port (1: lan1, 2: lan2) is IP address of selected network port. If it is set to 1, the corresponding IP address is IP address of lan1 network port: 192.168.188.100. Portnum is default value of 502 and default port number.

Pic6.2-17



Set communication connection on touch screen software (HCTDesigner) and write the content to download to TP2000. The IP address is address of the LAN port connected between the touch screen and Q1.

Pic6.2-18



Download program to Q1 to finish communication.

6.3 Modbus RTU Configuration

6.3.1 Modbus RTU Overview

COM1 port and COM2 port of Q1 are RS485 ports ,COM3 is RS232 port and support Modbus RTU master / slave protocol.

6.3.2 Modbus RTU Master Application

Modbus RTU master FB and instruction as below:

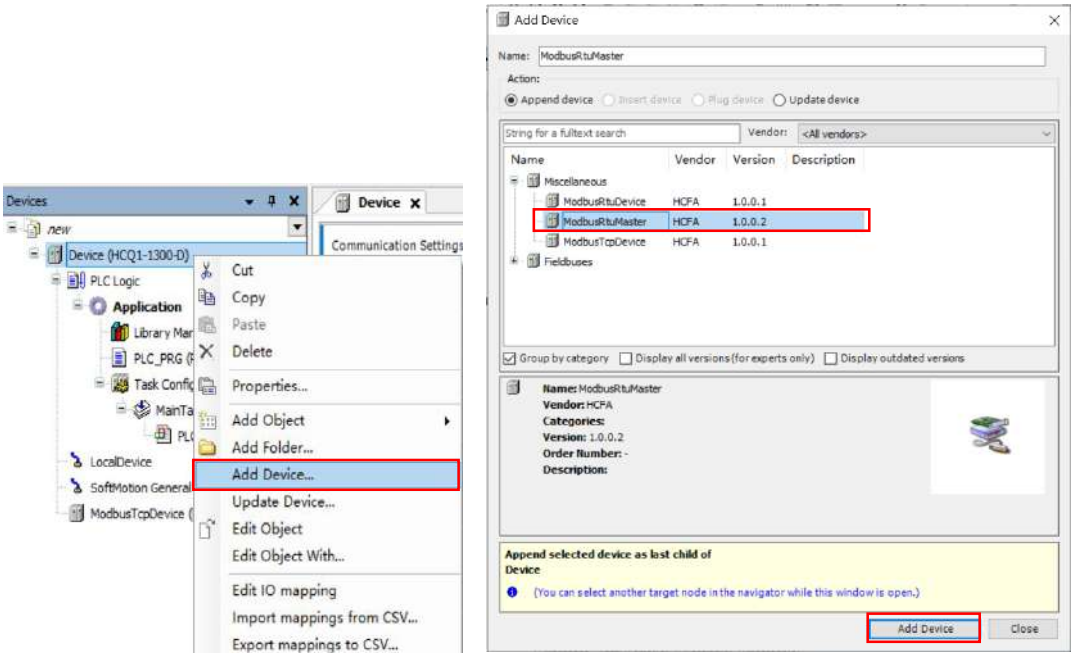
Pic6.3-1

Name	Type	Input/Output	Instruction
bStop	BOOL	in	When set to 1, suspend sending new request, and when set to 0, it continues
bReset	BOOL	in	Reset master on rising edge, clear unsent message
bAllSlavesNum	BOOL	out	All slaves communication normal
wOfflineSlavesNum	WORD	out	Number of offline slaves

bDone	BOOL	out	Master initialization succeeded
bError	BOOL	out	Master false status
wErrID	MODBUS_Error	out	Master false code

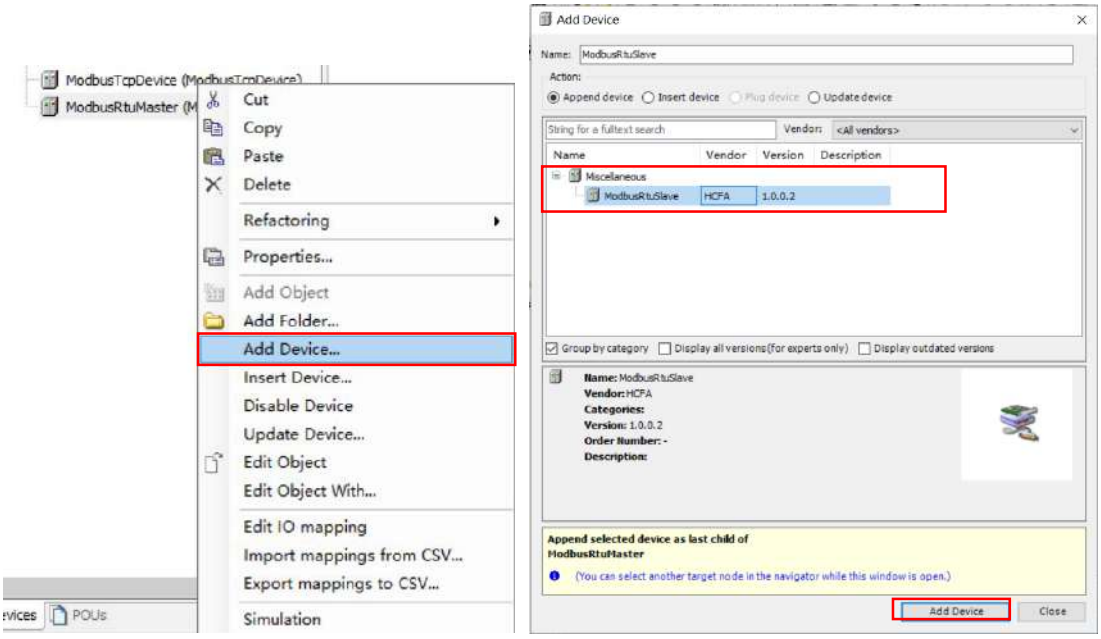
Right click tree menu Device→Add Device, select ModbusRtuMaster in miscellaneous, and click to Add Device.

Pic6.3-2



Right click ModbusRtuMaster and select "Add Device", select ModbusRtuSlave in the pop-up window and click to Add Device.

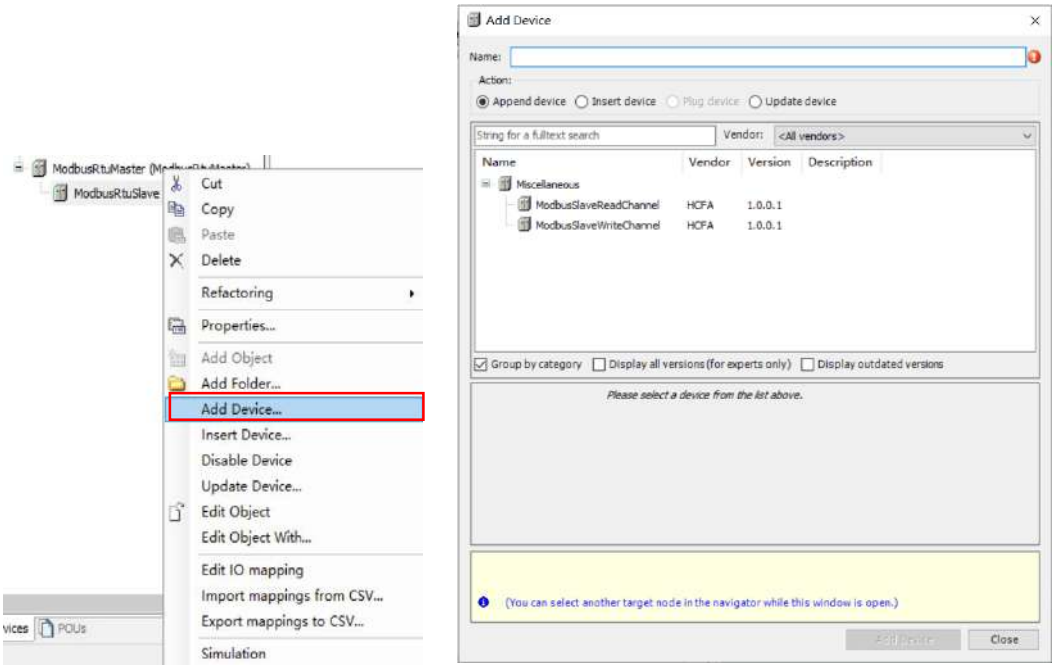
Pic6.3-3



Right click ModbusRtuMaster and select Add Device. In the pop-up window, select ModbusSlaveReadChannel and ModbusSlaveWriteChannel according to requirements, and click to add

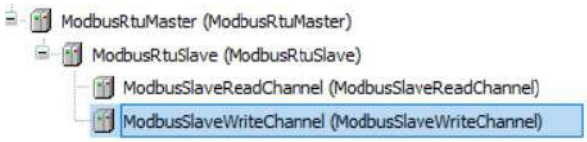
device.

Pic6.3-4



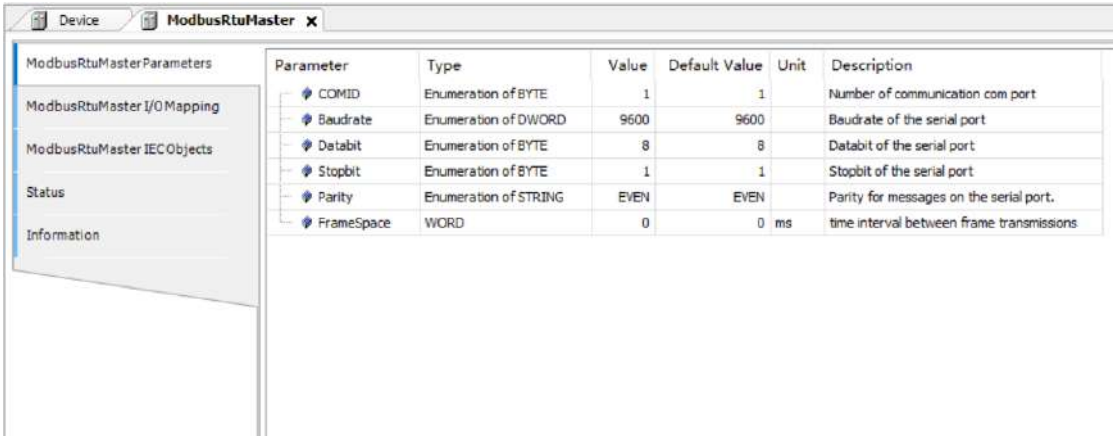
A set of read and write is added to the routine, shown as below:

Pic6.3-5



Click Added ModbusRtuMaster, select ModbusRtuMaster configuration.

Pic6.3-6

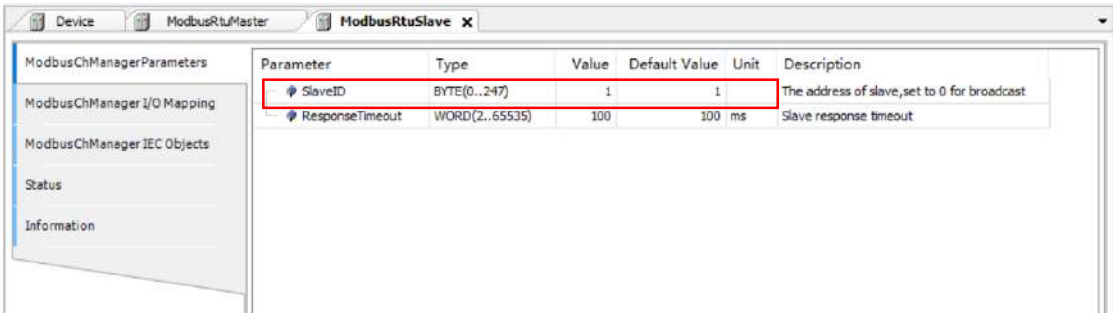


Name	Instruction	Default value	Comment
COMID	COM port	1	COM port of master (Q1)
Baudrate	Baud rate	9600	Five rate options：4800、96、19200、57600 and 115200
Databit	Data bit	8	Two options：7、8

Stopbit	Stop bit	1	Two options：1、2
Parity	Check bit	EVEN	Three options：EVEN（even check）、NONE（no check）、ODD（odd check）

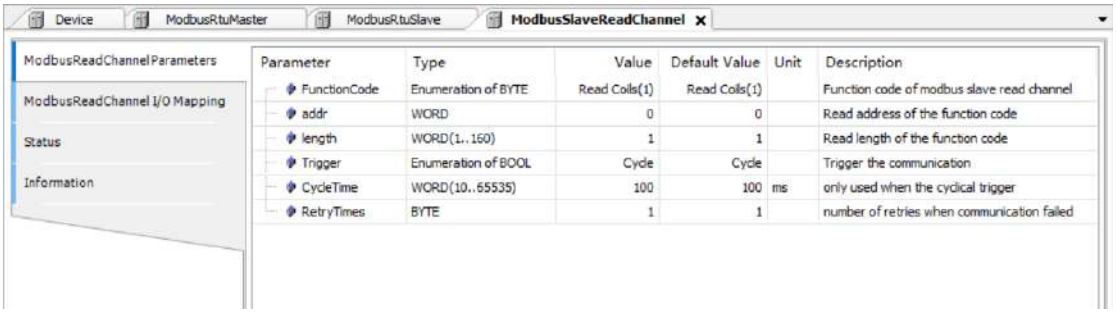
Open added ModbusRtuSlave, select ModbusRtuSlave configuration. Slave ID is COM port of corresponding slave.

Pic6.3-7



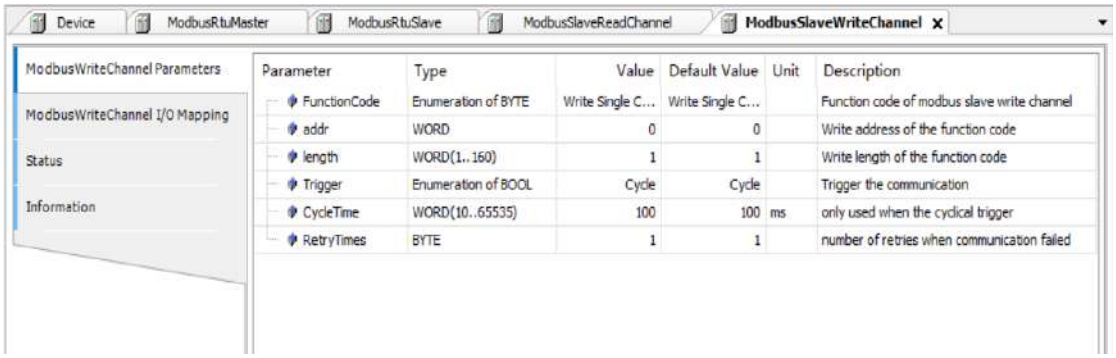
Open added ModbusSlaveReadChannel, select ModbusSlaveReadChannel configuration. Function code, select read data type from slave and 4 types: Read Coils, Read Discrete Inputs, Read Holding Register, Read Inputs Register.

Pic6.3-8



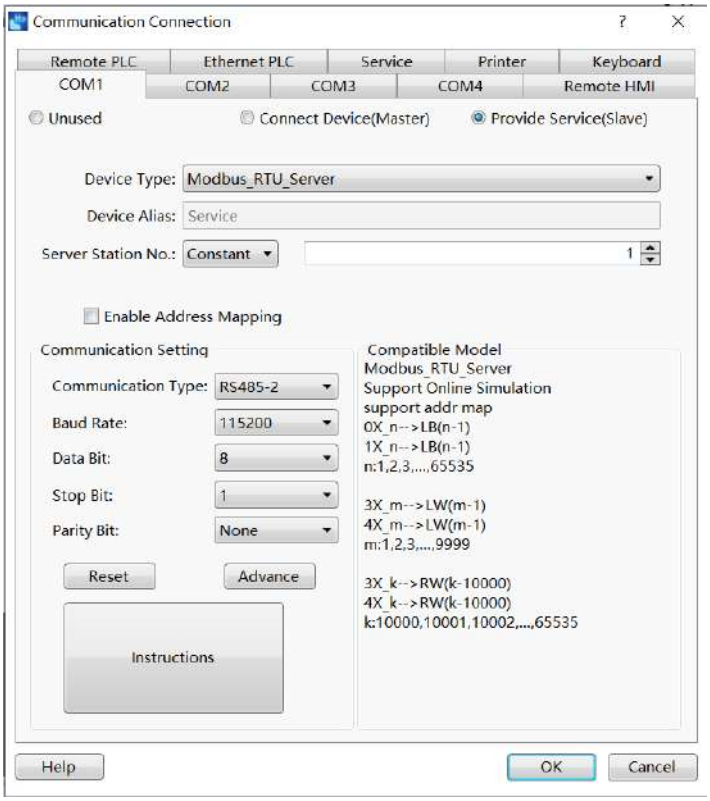
Click added ModbusSlaveWriteChannel, select ModbusSlaveWriteChannel configuration. Function code, select type of data written to slave and 4 types: Write Single Coils, Write Multiple Coils, Write Single Register, Write Multiple Register.

Pic6.3-9



On software of touch screen, set communication connection and compile, download to TP2000. Connect COM1 port of touch screen to COM1 port of Q1.

Pic6.3-10



6.3.3 Modbus RTU Slave Application

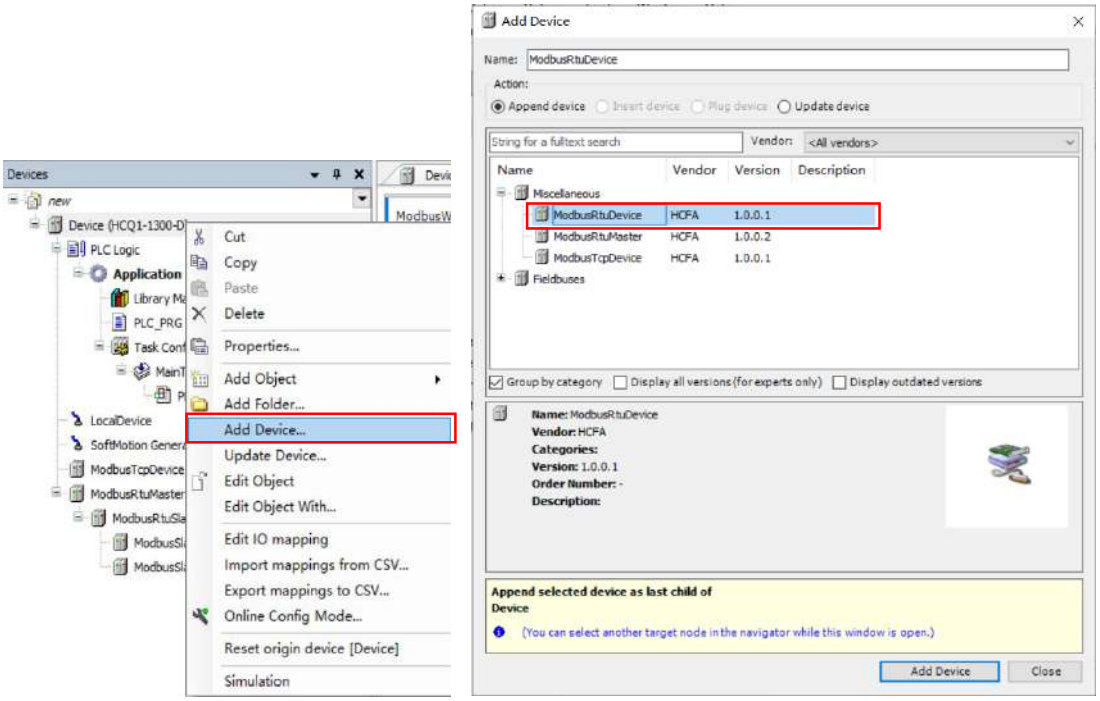
Modbus RTU slave FB and instruction as below:

Pic6.3-11

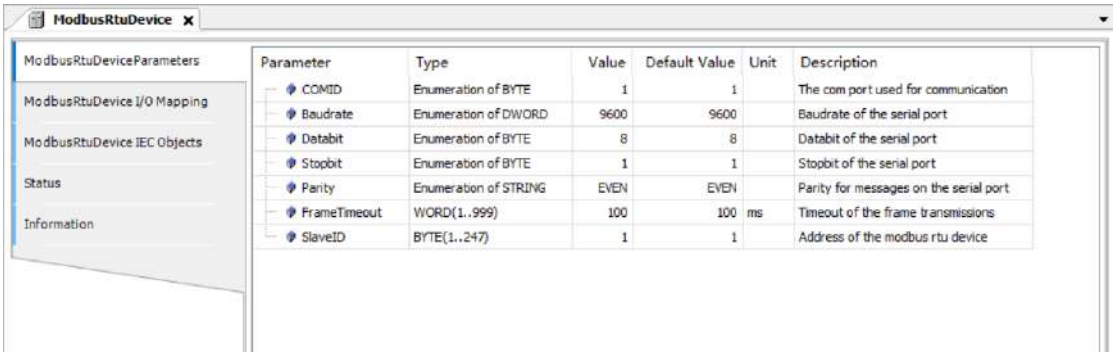
Name	Type	Input/Output	Instruction
bRestart	BOOL	in	Reset slave and error code
bDone	BOOL	out	Salve device opened, set TRUE after execution
bError	BOOL	out	False status
wErrID	MODBUS_Error	out	Error code

Right click the tree menu Device→Add device, select ModbusRtuDevice in miscellaneous, and click to Add device.

Pic6.3-12



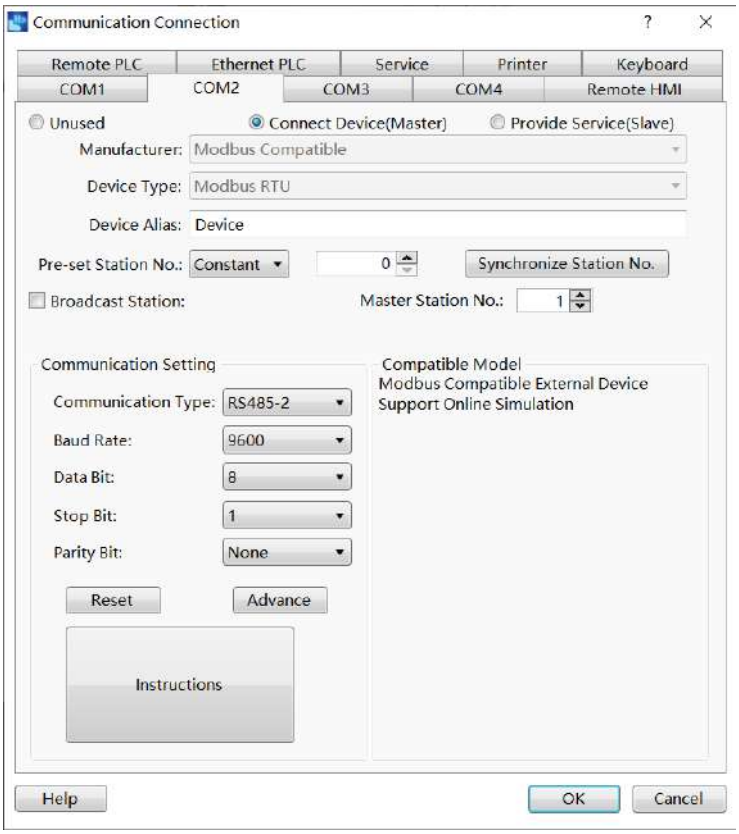
Open added ModbusRtuDevice, select ModbusRtuDevice configuration.



Pic6.3-13

Name	Instruction	Fault value	Comment
COMID	COM port	1	COM port of master
Baudrate	Baud rate	9600	Five rate options：4800、96、19200、57600 and 115200
Databit	Data bit	8	Two options：7、8
Stopbit	Stop bit	1	Two options：1、2
Parity	Check bit	EVEN	Three options：EVEN（even check）、NONE（no check）、ODD（odd check）
SlaveID	Slave COM port	1	COM port of slave (Q1)

On software of touch screen, set communication connection and compile, download to TP2000. Connect COM2 port of touch screen to COM1 port of Q.



6.4 OPC UA Configuration

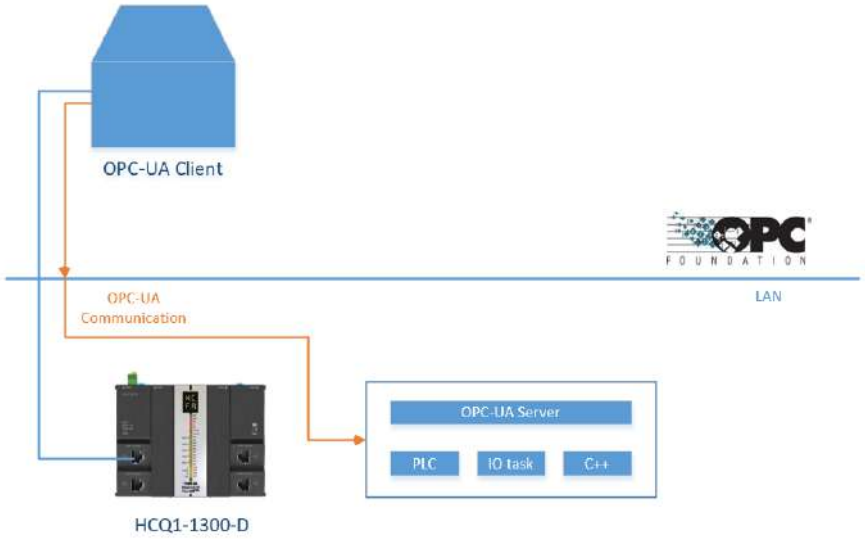
6.4.1 Introduction of OPC UA Protocol

OPC UA（Unified Architecture， unified structure) is OPC standard of next generation, which provides a complete, safe and reliable cross-platform structure, to catch real time and history data and time.

As an international standard communication protocol, previous exchange of machine data is independent of the manufacturer and platform. All required information can be used by each authorized application and authorized personnel at any time and anywhere. OPC-UA directly integrates general security standards in the protocol. Another major advantage of OPC-UA is that it is independent from traditional COM/DCOM system.

6.4.2 OPC UA Component

Software component	Brief description
OPC UA Server	OPC UA interface is provided to connect OPC client with Q1



6.5 EtherCAT Configuration

6.5.1 EtherCAT Protocol Overview

EtherCAT is the main bus communication protocol of HCFA Q series controller, which was real-time industrial Ethernet technology brought by Beckhoff Automation in 2003. It is also a real-time industrial fieldbus communication protocol and international standard based on Ethernet. It has the characteristics of openness, high compatibility and fast transmission speed. It supports a variety of device connection topology. The slave node takes a special control chip, and the master takes a standard Ethernet controller.

EtherCAT is supported on Port3 of Q1, and multi series of HCFA driver and controller can be connected. Meantime, it also supports related EtherCAT communication device from third party.

Q1 can have maximum 128 extension slaves, 65535 I/O points. If extended I/O points are still needed, take use of EC couple.

6.5.2EtherCAT Master Application

Based on standard of PLCopen platform and IEC61131-3 programming standard, Q1 owns powerful process and motion control ability, which can satisfy various needs of our customers. Application details of Q1 as EtherCAT refer to 5.12.

用我们的工作　　创造美好的生活

浙江禾川科技股份有限公司
电话：0570-7117888
地址：浙江省龙游县工业园阜财路9号

杭州研发中心
技术支持热线:400126969
地址：杭州市余杭区五常街道文一西路1001号D幢4楼