



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	
3.1 CODESYS Software Architecture	
3.2 Device Manager	11
3.2.1 Package Manager	11
3.2.2 Device Repository	
3.3 PLC Application	
3.3.1 Library Repository	
3.3.2 Task Configuration	
3.3.3 PLC Programming	22
3.3.4 Data Unit Type	
3.3.5 Recipe Manager	
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	

KCFa

Chapter 5 Creation of Simple PLC Project	
5.1 Start CODESYS	
5.2 Create New CODESYS Project	
5.3 Establish Communication with Q1	
5.4 Creation of PLC Project	
5.4.1 Add Library	
5.4.2 Task Setting	
5.4.3 Programming of PLC Program	
5.5 Add Trace to monitor program variable	
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	
5.13 Modify EtherCAT master information and communication parameter	
5.14 Realize of single axis control command	128
Chapter 6 Communication Setting	
6.1 Configuration of Ethernet TCP/IP	
6.1.1 TCP/IP Overview	
6.1.2 TCP/IP Protocol application	
6.2 Modbus TCP Configuration	
6.2.1 Modbus protocol overview	
6.2.2 Modbus TCP overview	
6.2.3 Modbus TCP Model	
6.2.4 Modbus TCP Data message structure	
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	
6.5 EtherCAT Configuration	150
6.5.1 EtherCAT Protocol Overview	
6.5.2EtherCAT Master Application	

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 Upd	late 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.

•	-	
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

Diagram 0-1 Environment requirements of Installation

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.







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.



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.



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

第四讲-示例程序.project* - CODESYS	3	- 0	Х
File Edit View Project Build	Online Debug Tools Window Help		₹
🗎 🚔 🔚 🕼 🗠 여 🐰 🛅 🛍 🕽	< 🏘 🎼 🌿 📕 🐄 🦄 🐴 💼 🋅 - 🔓 🧐 🥰 🕨 🔹 🔧 🖓 🛱 - 😤 🗊 🖆 👘 🖇 4	🎫 🐨	
	Tool bar		
Devices 🗸 🗸 🗙	Device PLC_PRG X		•
= 前第四讲-示例程序	Device.Application.PLC_PRG		
E-S Device [connected] (HCQ1-130	P. Expression	Туре	^ 🛒
PLC Logic	blight1	BOOL	
Application [run]	blight2	BOOL	~
Library Manager	<	>	
POU1 (PRG)	1 fbTON1(IN TRUE :=NOT fbTON1.QFALSE, pt T#500ms :=T#500MS);		^
Task Configuration	E 2 CASE ideng 2 OF		
🖹 🍪 MainTask	a i: 4 IF iVar 0 kiTime!		
RG and RG	5 Var :=ivar Edit area		
÷	6 blightl TRUE :=1;		
Device area	7 blight2FALSE :=0; B B ELSIF iVar 0 piTimel 6 OR iVar 0 =iTimel 6 THEN		
Axis Po		100 %	₫ ~
~	Messages - Total 0 error(s), 0 warning(s), 6 message(s)	•	ч х
	Download • O error(s) • O warning(s) • 1 mess	age(s) 🗙 💥	
	Description Object	Position	
	A Core Dump created on the 2019/5/9 18:00:0 message area		
	i j		
Devices POUs Device user: user : build: 0 0 0	Breakpoints Messages - Total 0 error(s), 0 warning(s), 6 message(s) Garden 1 Precomple: RUN Program Online mode unchanged Project user: (nobod	iy) IN Ln 5 Col 18	-

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

Watch 1				
Expression		Application	Туре	Value
	,			
	watch area			
<				

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.





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

New Pro				×	Standar	d Project		×
A project co Name: Location:	raries	Templates:	project	PRG		objects withi - One program - A program I - A cyclic tasl	t to create a new standard project. This wizard will create the follow this project: mmable device as specified below PLC_PRG in the language specified below which calls PLC_PRG to the newest version of the Standard library currently installed. HCQ1-1300-D (Zhejiang Hechuan Technology) Structured Text (ST)	ving
			OK	Cancel			ОК Са	ancel

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.





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

Тоо	ls Window Help	sitor	ry X	
Ø	Package Manager	:em	✓ Edit Locations	
1 1 1 1	Library Repository Device Repository Visualization Style Repository	Progr	ramData\CODESYS\Managed Libraries) Install 2	
	License Repository	:llan	rpanies) Vininstall Export	
	Scripting +	atio		×
	Customize Options Import and Export Options	n Ises	HCTDesigner-2.5.10743.0—TP25XX系列 4/19/2022 10:17 AM	Type File folde
	Group by c	atego	3D Objects Modbus Poll 4/20/2022 9:23 AM b_docDevelopm Modbus Slave 4/20/2022 10:14 AM Desktop Modbus Carcino file 4/13/2022 5:01 PM Desktop HCFA_ATCLib_1.1.0.compiled-library 4/18/2022 3:32 PM HCFA_HSIO ATCLib 1.0.0.7.compiled-libr 4/6/2022 2:06 PM	File folde File folde COMPIL COMPIL
	Library Profile	es	Downloads V File name: Open Open	Cancel

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.

		Add Library	×
Pic 0-2		String for a fulltext search	
Demo.project - CODESYS File Edit View Project Librar File Bit Strategy Stra		Library Application 	Company System System System
Device (HCQ1-1300-D) Device (HCQ1-130	Name SM3_CNC = SM3_CNC, 4.4.0 SM3_Robotics = SM3_Robotic SM3_Robotics_Visu = SM3_Robotics_Visu = SM3_Ro	Use Cases Set (Hiscellaneous)	OK Cancel
 SoftMotion General Axis Pool SoftMotion General Axis Pool 	Standard, 3.5.14.0 (System)		

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).

🛍 Add Library		×
String for a fulltext search		
Library Placeholder		
Company: (All companies)		~
 (Miscellaneous) Application Docs HCFA Intern System 		
Group by category Display all versions (for experts only)		
Details Library Repository	OK	Cancel

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

Pic 0-4

File \rightarrow New project \rightarrow Libraries \rightarrow 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.

Categories		Templates:			
🚞 Lib		CODES:	CODESYS interfa	CODESYS library	empty library
A new COD Name:	ESYS library with codi SampleLibrary1	ng rule compliant struct	res		
Location:	D:\codesys				~ <mark></mark>
ndow	Help	• % ((= ?=	C	OK	Cancel

Pic 0-5

SampleLibrary1.library* - CODESYS					- 0	×
File Edit View Project Libraries Build 🎦 🖙 🖬 🗒 🎒 어 어 & 🖿 🖷 🗶 🏘 입	and the second second second	Window Help	• ■ 4 ° 〔≡ ¢	5 t= +1 8	¢ 罰 ≓ ∛	₹
POUs 👻 🕂 🗙	Library Manager 🗙					•
SampleLibrary1 SampleLibrary1 Dibname Dibname	Add library X Delete lii Name ↔ ↔ CAA FB Factory = ↔ ↔ CAA Types = CAA ↔ CBML = Common E	brary Properties i C Cut Copy Paste Delete Add Library	Workgroup) orkgroup) olutions GmbH)	eholders milling Namespace FBF CAA CBML	Effective version 3.5.13.0 3.5.13.0 3.5.13.0 3.5.13.0	
	CAA FB Fac 3	Try To Reload Library Properties Export Library				
	Messages - Total 0 error(s), 0 wa				+	φ×
	Download	•			0 message(s)	-
1	Description	Project		bject	Position	
POUs 😤 Devices	Messages - Total 0 error(s), 0					
	Last	build: 🔕 0 🕐 0 Precompile	• 🗸	Project user:	(nobody)	0

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.





Create POU

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



Check and save by Build \rightarrow 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 \blacksquare , save project and load to library to save the library file. **Pic 0-8**

SampleLibrary1.library* - CODESY	YS	
File Edit View Project Bui	ild Online Debug Tools W	
1 3 B 🔠 🖉 🗠 🖓 🕮	Build F11	Project Information X
	Rebuild	File Summary Properties Statistics Licensing Signing
POUs 1	Check all Pool Objects	Company: HCFA
SampleLibrary1	Generate code	Company: HCFA
John SampleLibrary 1	Generate runtime system files	Title: HCFA Common Library Template
Enums	Clean	Version: 1.0 Released
- Eunction Blocks	Clean all	Library Categories:
	6 VAR	Default namespace: TMP
🗈 🚞 GlobalConstants	1	Author: Alyssa
🕀 🚞 GlobalVariables		Description: A sample library
- 🗀 Interfaces		
🗀 Structs		~ · · · · · · · · · · · · · · · · · · ·
🗉 🚞 Templates		The fields in bold letters are used to identify a library.
Types		
🛗 Library Manager		Automatically generate 'Library Information' POUs
FB_wave (FB)		Automatically generate 'Project Information' POUs
Project Information 2		OK Cancel

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 \rightarrow Save Project As Compiled Library \rightarrow Save to save the compiled library file and the source code will not open to shared users.

Pic 0-9

File	Edit View Project Build Online Debug	Save As Compiled Library	Х
睝	New Project Ctrl+N		~
2	Open Project Ctrl+O Close Project	← → ▼ ↑ ≪ 本地磁曲 (D:) > codesys ▼ ひ	搜索"codesys"
	Save Project Ctrl+S	组织 ▼ 新建文件夹	8== ▼ (?)
	Save Project As Save Project As Compiled Library	🏪 本地磁盘 (C:) 🔺 名称 🍐	验日期 类型 ^
	Save Project And Install Into Library Repository	本地磁盘 (D:) CODESYS	022/3/3 10:09 文件:
	Project Archive	→ 本地磁盘 (E:) GatewayPLC 20	022/3/3 10:10 文件:
	Source upload Source download	🥏 Network	022/3/3 10:09 文件:
6	Print	- · · · · ·	>
	Print Preview	文件名(N): SampleLibrary1	~
	Page Setup	保存类型(T): Compiled library files	~
	Recent Projects		
	Exit Alt+F4		保存(S) 取消
_	Exit Alt+F4	▲ 隐藏文件夹	保存(S)

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

Pic 0-10

POUs 🗸 🕂 🗙	Project Settings	Х
SampleLibrary1		~
 Libname Function Blocks Function Blocks Functions Global/Constants Global/Variables Interfaces Structs Templates Templates Types Ibrary Manager FB_wave (FB) Project Information 	 Compile options Compile warnings Library development Page Setup Security Security Security Security Security Secure Download Static Analysis Light Users and Groups Visualization Visualization Profile Security 	ed,
	OK Can	cel

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.

Encryption	Password	×	CODESYS	
ß	Enter the password for 'SampleLibrary	(1):	The specified project could not be loaded. Possible reasons might be: - The project file is corrupted or invalid. - The project has been saved with a newer version of the programming system. If this is the case, you can open the project in the corresponding version, choose "Save As" with the appropriate storage version, and retry again.	
	OK Cancel		确定	

3.3.2 Task Configuration

Pic 0-11

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

Ubrary Manager 🛛 📓 Task Configuration 🗙											
Monitor Varial	Monitor Variable Usage System Events Properties										
Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle Time (µs)	Average Cycle Time (µs)	Max. Cycle Time (µs)	Min. Cycle Time (µs)	Jitter (µs)	Min. Jitter (µs)	Max. Jitter (µs)	
🕑 MainTask	Valid	759965	1824953	7	9	25	6	2731	-2701	30	

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.

Monitor Variable Usage System Even	nts Properties		
Variables	Туре	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_sample 1.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

/ 🎁 🛛 Li	brary Manager	Task Configuration 🗙 📄 PLC_PRG	
Monitor	Variable Usage Sys	stem Events Properties	
🕂 Add E	Event Handler 🗙	Remove Event Handler 🛛 Event Info 🖹 Open Event	FL AfterReadingInputs ~
Name		Description	AfterWritingOutputs BeforeReadingInputs BeforeWritingOutputs
Add Ev	vent Handler	×	DebugLoop DownloadDone
	Event:	AfterReadingInputs ~	Exception ExitDone Login
	Function to call:		Description Logout OnlineChangeDone PrepareDownload
	Scope:	Application O POUs	PrepareExit PrepareExit
Implem	nentation language	Structured Text (ST) \vee	PrepareExitTasks PrepareOnlineChange
	Description:	Called after reading inputs. Context=IEC task. Debugging=Enabled	PrepareReset PrepareShutdown PrepareStart PrepareStop ResetDone
		OK Cancel	StartDone StopDone

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.



Properties

Basic properties show in tab.



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

Devices	→ ∓ X	MainTask X
	🗏 💮 Application 🛛 💌 ^	Configuration
	👘 Library Manager	
	FB_Peak (FB)	Priority (031): 1
	FB_PT1Filter (FB)	Туре
	F_Minmum (FUN)	Cyclic Interval (e.g. t#200ms): 20 ms
	Login (FUN)	
	PLC_PRG (PRG)	Watchdog
	PLC_sample (PRG)	
	PLC_sample 1 (PRG)	
	POU (PRG)	Time (e.g. t#200ms): t#200ms
	POU1 (PRG)	Sensitivity: 1
	POU_1 (PRG)	
	🖹 🎆 Task Configuration	
	🖃 🍲 MainTask	🗣 Add Call 🗙 Remove Call 📝 Change Call 🕼 Move Up 🔮 Move Down 🎽 Open POU
	PLC_sample	
	PLC_PRG	POU Comm
	PLC_sample	一 PLC
Œ	- 🏅 LocalDevice	曲 PLC
	🗝 🏅 SoftMotion General Axis Poo 🗸	曲 PLC
<	>	

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	0-1	8
-----	-----	---

8ms				4ms		10ms							
		~										1. 66	

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		
Type	Event:	9

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:

Туре			
🞸 External	~	External event:	I0IntEvent ~
			I0IntEvent
			I 1IntEvent
Watchdog			I2IntEvent
Enable			I3IntEvent I4IntEvent
			ISIntEvent
Time (e.g. t#200ms):			I6IntEvent
			I7IntEvent
Sensitivity:	1		18IntEvent
			I9IntEvent
			I 10IntEvent
			I11IntEvent I12IntEvent
		4.0	
🕂 Add Call 🗙 Ren	nove Cal	Change Ca	I 14IntEvent
POU		Comment	I15IntEvent
		comment	Cnt0CmpEvent
PLC_PRG			Cnt1CmpEvent
			Cnt2CmpEvent
			Cnt3CmpEvent Cnt4CmpEvent
			Cnt5CmpEvent
			Cnt6CmpEvent
			Cnt7CmpEvent

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			
Watchdog Enable			
Time (e.g. t#200ms):	t#200ms	ms	\sim
Sensitivity:	1		

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:



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

Add POU X	
Create a new POU (Program Organization Unit)	POU X
Name: POU Type Program Function Block Extends: Implements: Final Abstract Access specifier: Method implementation language: Structured Text (ST)	
Function Return type:	Code Area
Implementation language:	
Structured Text (ST) Continuous Function Chart (CFC) Continuous Function Chart (CFC) - page-oriented Function Block Diagram (FBD) Ladder Logic Diagram (LD) Sequential Function Chart (SFC)	100 % @

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:

) PL	C_test X	•
3		^ <u>B</u>
4	VAR RETAIN PERSISTENT	
5	//digtal input	
6	binput AT %IX0.0:BOOL;	
7	END VAR	100 % 📵 🗸
8	-	100 % 🔍 🗸

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:

D: 0.05

	PIC 0-25		
Scope: variable scope, local variable or	Auto Declare		×
interface variable	Scope:	Name:	Туре:
Name: refer to IEC61131-3	VAR ~	9	~ >
Type: variable data type	Object:	Initialization:	Address:
Object: application	PLC_test [Application] ~	Comment:	
Initialization: initial value	CONSTANT RETAIN		^
Address: external address			~
Signal: define variable as constant, holding			
or persistent			OK Cancel
Comment: variable comment	1		

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:



Calling FUN in program:

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



Online operation result of FUN:

Pic 0-29

F_Minmum PLC_sample X			
Device.Application.PLC_sample			
Expression		Туре	Value
🛷 Inputi		INT	8
Input2		INT	3
Input3		INT	12
Ø MinInput		INT	3
<			
1 Input1 8 Input2 3 Input2 Input2 Input3 12 Input3	MinInput 3		
RET			

FΒ

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

Add POU X	/ 🖹 F.	Minmum	PLC_PRG FB_Maximum X
Create a new POU (Program Organization Unit)	1	FUNCTION_BLO	CK FB_Maximum -> Function block name
	2	VAR INPUT	
Name:	3	-	→ Input interface declaration area for function block
FB_Maximum	4	END VAR	
Туре	5	VAR OUTPUT	
○ Program	6	-	Output interface declaration area of the function block
Function Block	7	END VAR	
Extends:	8	VAR	
Implements:	9		→ Local variable declaration area for function block
Final Abstract	10	END VAR	Local variable declaration area for function block
Access specifier:	11		100 % 🔍 🗸
Method implementation language:			
Function Block Diagram (FBD)	1		
OFunction			
Return type:			
in the second se			
Implementation language:			Code Area
Structured Text (ST)			
			` </td
Add Cancel			100 % 🖳

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:

	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 $0 < \alpha < 1$). 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:



Calling FB in program:

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



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

B PLC_sample1 X FB_Peak F_Minmum Device.Application.PLC_sample1						
# Ø FB_PT1Filter_0	FB_PT1Filter				Function Block	
🖗 гКр	REAL	1				
🗼 tTn	TIME	T#1s				
tPLCCycletime	TIME	T#4ms				
rOutput	REAL	-4.1819253				
fPeakValue	REAL	0.262467861				
🗄 <pre> # <pre> # </pre> # <pre> # </pre> # </pre> # <pre> # </pre> # #	FB_Peak				creat peak	



rOutput output waveform:



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.



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
	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:



Extended structure Motor_hcfa:

Add DUT ×	Input Assistant
Add DUT X	Text search Categories
◆ Create a new data unit type	Data Unit Types 🔺 Name Type
	- O Application Application
Name:	DUT TYPE
Motor_hcfa	motor TYPE
Type:	
Structure	Motor Motor_hcfa 🗙 📄 PLC_PRG
Extends: motor	1 TYPE Motor hcfa EXTENDS Motor :
O Enumeration	2 STRUCT
Textlistsupport	
0.1	3
○ Alias	4 Ingress_Protection:STRING(20);
Base type: >	5 Rated Torque:REAL;
◯ Union	6
	7 END_STRUCT
	8 END TYPE
	9 -
Add Cancel	

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 ".".



Call and assign element of structure one by one:

PLC_PRG 🗙 🌱 Motor		
Device.Application.PLC_PRG		
Expression	Туре	Value
🗏 < motor_1	Motor_hcfa	
Product_ID	DWORD	2234
Vendor	STRING(20)	'HCFA'
Ø Nominal_Voltage	REAL	240
Ø Nominal_Current	REAL	0.5
Poles	INT	4
Ø Brake	BOOL	TRUE
Ingress_Protection	STRING(20)	'IP65'
1 motor_1.Product_ 2 motor 1.Vendor		2234; :FA';
3 motor_1.Ingress	Protection	IP65' :='IP65';
4 🔵 motor_1.Nominal		:=240;
5 motor_1.Nominal_		:=0.5;
	4 :=4;	
6 • motor_1.Poles 7 • motor_1.Brake		

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:



- 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.

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

- OC0359: Arithmetics not allowed on strict ENUM type 'Weekday'
- O 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.



Call enumeration of Weekday in program:

Pic 0-47

	4	//Enum
		Weekday 1:Weekday;
	7	
	10	//Enum
	11	(*Weekday 1:=test; *)
	12	IF weekday_1>5 THEN
	13	Weekday_1:=0;
	14	ELSE
	15	Weekday_1:=Weekday_1+1;
	16	END IF

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:



[Example 3] Define a string(15) with NetID to store IP address.

Statement:

Pic 0-50				Add DUT		\times
1 Application	Cut	<u>0</u>	Alarm configuration Application	🔶 Create a	new data unit type	
🖓 DUT (STR 🕫	Сору	Ø	Axis Group	- Create a		
PLC_PRG	Paste	8	Cam table	Name:		_
Task Cont X	Delete	8	CNC program	NetID 4		
🖹 🕸 MainT	Refactoring +		CNC settings	Type:		
P	Properties	9	Data Sources Manager	O Structure	: motor	1
LocalDevice		\$	DUT 3			11
SoftMotion Genera 🛄	Add Object 2		External File	O Enumerati		
	Add Folder	۵	Global Variable List	Textlists	support	
Dĩ	Edit Object		Image Pool	Alias		
				Base type:	STRING(15) >	
🔶 🔶 NetI	D X			O Union		
1	TYPE NetID : STRING	;(1	5); END_TYPE			_
2						
					Add Cancel	

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 8 //Alias
9 IPAddr:NetID;
10 END VAR 17 //Alias
18 IPAddr:='192.168.1.1';
19

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:

	16#000	16#001	16#002	16#003	
INT					
BYTE					
DWORD					
Statement					
-----------	----------	------------------------	--------	-----------	--
F	Pic 0-53				
[PLC	_PRG Unior	1 X		
	1 2	TYPE Unionl : UNION	Consor	tium Name	
	3				
	4	END_UNION END TYPE			
	6	END_TIPE			

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

Pic 0-54			
Application		Alarm configuration	Add DUT ×
- A DUT (STR	X Cut	Application	
🚹 Library Ma	🗈 Сору	Axis Group	পিঃ Create a new data unit type
PLC_PRG	Paste	🔕 Cam table	Name:
🖃 🎆 Task Conf	X Delete	CNC program	TEST
🛓 🍪 MainT	Refactoring +	SCNC settings	Type: 4
P	Properties	Data Sources Manager	Extends: motor
LocalDevice		Ŷ\$ DUT 3	○ Enumeration
SoftMotion Genera	Add Object 2 🔹 🖡	📓 External File	Textlistsupport
	Add Folder	🧭 Global Variable List	Alias
	ີ Edit Object	image Pool	Base type: DNT >
	PLC_PRG		Union Add Cancel

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

Pic 0-55

Statement:

		Device.Application.POU					
		Expression		Туре	Value		
10	//Union	< var1		STRING(8)	'hcfa'		
11	Test1:TEST;	var2		STRING(8)	'hcfa'''		
12	END VAR	🗄 🧳 Test1		TEST			
19	//Union	1	//Union				
20	Testl.varl:='hcfa';		[estl.varl 'hofa'	:='hcfa';RETU	RN		
21							

Element data type can also be different in union.

[Example 4] Use union to integrate variable of two bytes into one.

Statement:



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

xpression	Туре	Value
Un_Word	Union_Word	
nWord	WORD	16#1234
🖃 🧳 nByte	ARRAY [01] OF B	YTE
ø nByte[0]	BYTE	16#34
ø nByte[1]	BYTE	16#12
ø nByte_Low	BYTE	16#12
nByte_High	BYTE	16#34

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.

	•	чх	PLC_PRG DLC_F	PRG1	×
Untitled2 Untitled2		•			
	X	Cut			
SoftMotion General Axis P		Сору			
	Ē.	Paste		8	Alarm configuration
	\times	Delet	e	0	Application
		Refac	toring +	8	Axis Group
	ca.	Prop	erties	8	Cam table
				8	CNC program
			Dbject •		CNC settings
	n°				Data Sources Manager
			Dbject Dbject With	**	DUT
			-		External File
	ОŞ	Logir			Global Variable List
		Delet	e application from device		Image Pool
	_				Interface
				2	Network Variable List (Receiver)
				2	Network Variable List (Sender)
				T	Persistent Variables
				Ð	POU
POUs 😪 Devices		>	Watch 1	4) A	POU for implicit checks Recipe Manager

Pic 0-60

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

Pic 0-61

Add Recipe Manager	×	🔒 Recipe Manager 🗙	:	
~		Storage General		
Create a Recipe Manager		Storage Type: Textual		~
-			l	•
Name:		File Path:		
RecipeManager		File Extension: .txtrecip	be	
		Separator		
		🔿 Tab	○ Semicolon	⊖ Comma
		⊖ Space	• :=	01
		Available Columns	>	Selected Columns
		₩ Type ₩ Name	>>	₩ Variable ₩ Current Value
		Comment	<	W Current Value
		Minimal Value		
		Maximal Value	<<	
Add	Cancel	Save as default		Up Down

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

Storage Type:	Textual	~
File Path:	Binary Textual	

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

× ■ ■ ×	Cut Copy Paste Delete Browse Properties	•			Add R		×
****	Add Object Add Folder	•	Q	Recipe Definition			
ſ	Edit Object Edit Object With						
						Add C	ancel

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**

Devices 👻	џ х	🔍 Recip	es X							•
Demo	•	Variable	Туре	Name	Comment	Minimal Val	Maximal Va	Current Val	recipe1	recipe2
Device (HCQ1-1300-D)									/ /	
🗏 🗐 PLC Logic										
🖹 🔘 Application										
📶 Library Manager							(2		(3) /
PLC_PRG (PRG)			5					9		\smile /
🖹 🔒 Recipe Manager			(1)							
Recipes										(4)
🖹 🎉 Task Configuration										\smile
🖮 🍪 MainTask										
PLC_PRG										

Different Recipe ② can be added through options of Recipes, where user Pic 0-65 can also delete, load or save recipe files.



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	
🖃 👘 Recipe Management, 3.5.13.0 (System)	• •
🗉 🗀 General Types	
🖹 🗀 Interfaces	
IPersistantRecipeListSupportsAdd	
🧭 ReturnValues	
RecipeManCommands	
CreateRecipeNoSave	
DeleteRecipeFile	
GetLastError	
🕞 🖬 GetLastInfo	
GetRecipeCount	
GetRecipeValues	
GetRecipeVariableNames	
LoadAndWriteRecipe	
ReadAndSaveAs	
DeadAndSaveDecine	

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



[Example 1] Create two recipes to produce bread of strawberry and orange

Needed variables:

PI	C_PRG1 X	C BREAD	Proc	duct	
1	PROGRAM	PLC_PRG1			
3 2	VAR				
3	egg	: INT;		//Egg	
4	mli	k:REAL;		//Milk	
5	flo	ar:REAL;		//Flou	r
6	but	ter:REAL;		//Butt	er
7	sug	ar:REAL;		//Suga	r
8	bak	ing_time:TI	ME;	//Baki	ng time
9	str	awberry_jam	:REAL;	//Stra	wberry jam
10	ora	nge_jam:REA	L;	//Oran	ge jam
11	dur	ation:TIME;		//dura	tion
12	pro	ductl:Produ	ct;	//The	product name
13					
14	END VAR				
15					

Add new recipe definition:



							Add Recipe Definition	×
							Q Create a new Recipe Definition	
							Name: BREAD	
🗄 🙀 Recipe Manager								
🗏 🎆 Task Configurati	V	Cut		1				
🖹 🍪 MainTask		Сору						
PLC_sar	Ē	Paste						
PLC_PR		Delete						
PLC_sar		Browse	•					
LocalDevice	Ē.	Properties						
🐨 🔟 GeneralIO (GeneralI	4:=		_			1		
SoftMotion General Axis	**	Add Object	•	Q	Recipe Definition			
		Add Folder		-				
	D°	Edit Object						
Devices		Edit Object With					Add Cancel	

Insert variable in recipe:

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

Pic 0-70

Rec	ipes Build Online Debug	Tools Window Help							
₩	Insert Variable	🛍 🛅 - 🖸 🕮 💝 💖 🔸 🔳 %	(j= 4j= 4	1*1 8 ¢ 📰	≓ ∛				
-10	Add child								
49	Add sibling	Product Q BREAD X							•
C)	Update structured variables	Changed recipe definition configuration will be	ransferred	with the next download (or online change				
0	Add a new recipe	changes respe dermition configuration win ber	anarenea	when the next download t	of offinite change				
¹	Remove recipe	Variable	Туре	Name	Comment	Minimal Val	Maximal Va	Current Val	
-0	· · · · · · · · · · · · · · · · · · ·	PLC_PRG1.egg	INT	egg	g				
B _2	Load Recipe	PLC_PRG1.mlik	REAL						
E	Save Recipe	PLC_PRG1.flour	REAL						

Add new recipe:

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

Pic	0-71
FIC	0-71

		Rev Recipe		×
* *	ipes Build Online Debug Insert Variable Add child Add sibling	Name: Copy from existing	<pre>Create Empty></pre>	~
C.	Update structured variables		OK	Cancel
Q,	Add a new recipe			

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 🔍 🔍 E	Product READ X PLC_PRG1										
Variable	Туре	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**

Debug

► Login

PLC_PRG1 V BREAD X 🧟 Recipes										
Variable	Туре	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:



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

)nl	ine Debug	Tools	Window	Help				
ļģ.	Login			Alt+F8				
Ş	Logout		C	Ctrl+F8				
	Create boot a	applicatio	n					
	Download				C	ODESYS	5	
	Online Chang	je						
	Source down	load to co	onnected d	evice				
	Multiple Dow	nload				?	Downloading the source requires the project to be saved first.	
	Reset warm					-	Click 'Yes' to save the project and continue with the source download	
	Reset cold						or click 'No' to skip the source download.	
	Reset origin							
	Simulation							
	Security				•		是(Y) 否(N)	
	Operating M	ode			•		XE(1) H(1)	

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 :



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.









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.



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.

Image: Construction (diagrams) Time axis Diagram 1 Y axis Critical upper limit: 0 Critical upper limit: 0 Critical upper limit: 0 Red	Trace Configuration			
Trace 1 Qraph color: Line type: Point type: Point type: Activate minimum warning Critical lower limit: Warning minimum color: Activate maximum warning Critical upper limit: O Yaxis Shown variables 4	Trace Record	Variable settings		
2 Line type: Point type: Point type: Activate minimum warning Critical lower limit: Warning minimum color: Bladx Jagram 1 Yaxis Shown variables 4	(1)	Variable:	••	
Presentation (diagrams) Time axis Diagram 1 Y axis Shown variables 4 Line type: Point type: • Dot Activate minimum warning 0 Critical lower limit: 0 Critical upper limit: 0 4		Graph color:	Blue	~
Presentation (diagrams) Time axis Diagram 1 Y axis Shown variables		Line type:	Line	~
Presentation (diagrams) Time axis Diagram 1 Y axis Shown variables 4 Critical upper limit: 0 3		Point type:	• Dot	~
Presentation (diagrams) Time axis Diagram 1 Yaxis Shown variables 4		Activate minimum warning		
Presentation (diagrams) Time axis Diagram 1 Yaxis Shown variables Uarning maximum color:		Critical lower limit:	0	
Time axis Diagram 1 Y axis Shown variables Uarning maximum color: Red	Presentation (diagrams)	Warning minimum color:	- Y \	~
Y axis Shown variables 4	Time axis	Activate maximum warning		
4		Critical upper limit:	0	
4	Shown variables	Warning maximum color:	Red	¥
(4)				
	4			
	Add variable	Reset display setti	ngs	OK Cancel

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

Trace Record	Display mode		
Trace	Auto O F	ixed length 🔘 Fixed	
	Minimum:	0	
	Maximum:	10s	
	Length:	10s	
	Grid:	Gray	~
Presentation (diagrams) Time axis	Tick marks Fixed spacing		
⊟- Diagram 1	Distance:	1s	
Y axis Shown variables	Subdivisions:	1	
_			
	Font		Preview

Set tracking background in tree menu Diagram 1. Pic 0-6

< ဳ Trace Configuration						×
Trace Record	10-					
	0					
Presentation (diagrams) Time axis	-10-0		5ms			10ms
Y axis Shown variables	Backcolor: Backcolor on Selection:	White Beige		~		
Add diagram Delete diagra	Reset display sett	se as default		ОК	Canc	el

• 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:

Trace Configuration		×
Trace Record	Record Settings Enable Trigger	
	Trigger variable: Trigger edge: Posttrigger (samples)	
	Trigger Level:	~
Presentation (diagrams)	Record condition: Comment:	
Time axis Diagram 1 Yaxis Shown variables		
	Resolution: ms V Automatic restart Advanced	
Add variable	Reset display settings OK	Cancel

Advanced Trace Settings		
Measure in every n-th cycle:	1	~
Recommended runtime buffer size (samples):		
🗹 Override runtime buffer size		100
Trace editor buffer size per variable (samples):	10001	

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
 - F 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.

Ft 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.

- 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", ".trace", ".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

Icon	Instruction					
36	Download Trace for collecting tracking curve will start immediately if trigger is not set.					
	Start or Stop Trace					
1	Reset trigger for reset after trigger event and restart.					
Ŀ3	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.</ctrl></shift>					
ଦ	Reset view, for view reset after it's defined in tracking configuration					
<u>‡</u> y	AutoFit, for auto adjust appearance of tracking after set in tracking configuration.					
ti	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

Visualization_Sample.project - CODESYS

File	Edit	View	Proj	ject	Visu	ualizati	on	Build	d (Onlin	е	Deb	ug	Tools	M	/indov	v H	Help
1		3 10	പ 🐰		Ē	×IÅ	6 🕼	ø <u>ð</u>	۱ <u>۵</u>	Щ.	훼	M >) D	à 🏜	· Dî		OŞ	0ğ
R (2)		- H	8	₽ •(릐	TO	\$ <u>0</u>		000	00 0	D[]o	00 ₽	2	음: 음;	₿‡		0 5	日本 日本

- Align Left: Align to left for selected visualization elements
- Top: Align to top for selected Align 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 horizo
- 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

For questions or comments, feel free to email 400@hcfa.cn

2		\$‡ {	밝 방 🛱 🛄 🛱 🌞		
	Pic 0-2				
∦ ⊫ ×	Cut Copy Paste Delete Select All Create Global Text List Order	~ ~ ~ ~ ~	Align Left Align Top Align Right Align Bottom Align Vertical Center Align Horizontal Center		
	Alignment •	00e	Make horizontal spacing equal		
	Group Ungroup Frame Selection Background	움, 당, 당, 이네 맨	Increase horizontal spacing Decrease horizontal spacing Remove horizontal spacing Make vertical spacing equal Increase vertical spacing		
izor	Multiply visu element	하하하다	Decrease vertical spacing Remove vertical spacing Make same width		

Make same size

Size to grid

i.

- 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.



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



Visualization Manage	r 🛛 📳 Visuali	ization	×				
🗉 Interface Editor 🔲 Ho	tkeys Configuration	🔠 Ele	ementlist				
Туре	х	Y	Width	Height	Id	Name	Access rights
#0 Ellipse	281	208	75	73	0	GenEl	
#1 Ellipse	262	23	78	74	3	GenEl	
#2 Ellipse	268	112	78	76	4	GenEl	
#3 Rectangle	322	41	87	226	7	GenEl	
						el	ementlist
		(L				100 %

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.



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	
Ŧ	Element look	
±.	Texts	
	Text properties	
±.	Absolute movement	click to enter the input assistant area
Ŧ	Relative movement	$\mathbf{\lambda}$
	Text variables	<u> </u>
	Text variable	
	Tooltip variable	
Ŧ	Color variables	
Ŧ	State variables	
÷	Inputconfiguration	

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.

Format output order	Description				
d, i	decimal				
0	Unsigned octal				
х	Unsigned hexadecimal				
u	Unsigned decimal				
С	Single String				
S	String				
f	Real number				

Pic 0-3

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

Add needed temperature variable in program.



Choose rectangle from Basic in tool box.



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.

		Texts		
		Text	Temperature:%4.2f deg	
		Tooltip		
	Ŧ	Text properties		
Temperatur®:%4.2f deg u	Ŧ	Absolute movement		
	Ŧ	Relative movement		
		Text variables		
		Text variable	PLC_PRG.ActualTemperature	
		Tooltip variable		
Running result as below:				
Pic 0-	8			

Temperature:0.00 deg

2. InputConfiguration

Pic 0-7

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.

F	Pic 0-9									
	Inputconfiguration									
			OnDialogClosed	Configure						
			OnMouseClick	Configure						
			OnMouseDown	Configure						
			OnMouseEnter	Configure						
			OnMouseLeave	Configure						
			OnMouseMove	Configure						
			OnMouseUp	Configure						

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

Input Configuration		X
OnMouseClick		
 Close Dialog Open Dialog Change the language Change shown visualization Execute command Switch frame visualization Write a Variable Execute ST-Code Toggle a variable File transfer 	Virite a Variable	Write a Variable Input type: Perault Choose variable to edit Use text output variable Use another variable Use an
		OK Cancel

- ① Select trigger type, then Write a Variable
- ② Add to right side via rightward arrow in picture.

 $(\centrify) \label{eq:constraint} (\centrify) \label{eq:constraint} Enter input type through keyboard or virtual keyboard.$

7	8	9	Back	° ! ″ § \$ % & / () = ? ′ \ Bac	:k
4	5	6	Clear		ear
1	2	3	ESC	€ A S D F G H J K L Ö Ä ' ES	sc
0	+/-		ок	2 3 Space { [] }	ж

- 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.



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.

Add Visualization	×
Creates a visualization object	
Name:	
Traffic_Siganl	
Add	Cancel

Create graphic of traffic lights with Rectangle, Rounded Rectangle and Oval **Pic 0-12** and place in order as right.

Edit color variable in program. Pic 0-13 PLC_PRG 🗙 💾 Visualization 1 Traffic_Siganl PROGRAM PLC PRG 1 2 VAR -Traffic_Red:BOOL; 3 Traffic_Yellow:BOOL; 4 5 Traffic_Green:BOOL; 6 END VAR

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

	PLC	PRG X 📳 Visualization 📳 Traffic_Siganl	 Color variables
	2	PROGRAM PLC_PRG VAR	Toggle color PLC_PRG.Traffic_Yellow 📉
	3 4	ActualTemperature :REAL; Traffic_Red:BOOL;	Normal state
	5 6	Traffic_Yellow:BOOL; Traffic_Green:BOOL;	Frame color PLC_PRG.Black
	7 8	Red:DWORD:=16#FFF0000; Yellow:DWORD:=16#FFFF00;	Fill color PLC_PRG.White
1	9 0	<pre>Green:DWORD:=16#00FF00; Black:DWORD:=16#000000;</pre>	Alarmstate
1		White:DWORD:=16#FFFFFF; END VAR	Frame color PLC_PRG.Red
		-	Fill color PLC_PRG.Red

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



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

Programming example:



Running result:(one of the indicators is ON)



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.



Label

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

N. 0. 0	^	🏹 Filter 🔹 🎼 Sort b	y ▼ 👌 Sort order ▼ 🗹 Advanced
		Property	Value
HCFA Sample interface		Element name	GenElemInst_25
		Type of element	Label
		Text ID	144
		= Texts	
		Text	HCFA Sample interface

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

General_Control 🗙						
1	PROGRAM General_Control					
2	VAR					
3	iAmp:BYTE;					
4	END_VAR					
5						

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 Subrange, 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:



```
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							
Columns	Columns						
Column							
□ [0]							
Width	40						
Image column	\checkmark						
Image configuration							
Fill mode	Fill cell						
Transparent							
Transparent color	Black						
Text alignment of column	Centered						
Use template							
⊞ [1]							
€ [2]	\square						

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

	2	•	
BMW	150	blau	•
BMW	150	blau	
Audi	150	grau	
Mercedes	150	silber	
VM	150	bronze	
Fiat	100	rot	

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.

	Re	eferences		Configure	
	😑 Po	sition			
		Х		671	
■ ₩ill be a Frame		Y		233	
		Width		250	
		Height		150	
• • •	E Sw	itch frame variable			
Frame Configuration					×
Available Visualizations		Selected Visua	lizations		
By visualization name 🔌 By type or instance	2	🗿 Add 🗙 Delete	♠ Move Up	✤ Move Down	
Filtering by full or partial name of visualizations	\times	Visualization	Switch inde	x	
Visualization_Sample	^	🕘 Traffic_Siganl	0		
Traffic_Siganl					
	_				
🖻 🚞 SM3_Basic					
🗄 💼 Visualization					
🖹 🗀 Generated					
VISU_NEW_MC_AbortTrigger					
VISU_NEW_MC_AccelerationProfile					
VISU_NEW_MC_CamIn					
VISU_NEW_MC_CamOut					
VISU_NEW_MC_CamTableSelect					
C	>				
		L			
			3	OK Canc	ei

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

Pic 0-29	
References	Configure
Traffic_Siganl	0
Heading	
Image ID	

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

Pic 0-30

Scaling ty	pe	Isotropic 🗸 🗸
Deactivat	e the background drawing	Isotropic Anisotropic
Reference	es	Fixed
🖃 Traffi	c Siganl	Fixed and scrollable

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

visu1	visu2	L

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.

Inputconfiguration			
OnDialogClosed	Configure		
OnMouseClick	Configure		
OnMouseDown	Configure		
OnMouseEnter	Configure		
OnMouseLeave	Configure		
OnMouseMove	Configure		
OnMouseUp	Configure		
± Toggle			
🗄 Тар			
 Hotkey 			

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**

Input Configuration			×
OnMouseClick	3		
 Close Dialog Open Dialog Change the language Change shown visualization Execute command Switch frame visualization Write a Variable Execute ST-Code Toggle a variable File transfer 	Execute ST-Code	Exec 4 ST-Code	tor:=2
-		ОК	Cancel

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.



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:



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

	_	Expression	Туре	Value	
•	III	- ×	ScrollValue	LREAL	42.64705657958

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:





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

_	Expression	Туре	Value
	SliderValue	LREAL	18.38235282897

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:



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

Pic 0-44

Ŧ	Position	
	Variable	Genera1_Contro1.nSelectValue
	Number format	
	Interval	1

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



Online running as below:

Pic 0-46

	20			
	Min: 0		Max: 100	
20				
	7	8	9	Back
	4	5	6	Clear
	1	2	3	ESC
	0	+/-		ок

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

 State variables 	
Deactivate inputs	Genera1_Contro1.bInvisble
Inputconfiguration	
OnDialogClosed	Configure
OnMouseClick	Configure
OnMouseDown	Configure
OnMouseEnter	Configure
OnMouseLeave	Configure
OnMouseMove	Configure
OnMouseUp	Configure

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:

nProcessValue: INT; tonl:TON; END VAR tonl(IN:=NOT tonl.Q , PT:=T#1S ,); IF nProcessValue<100 AND tonl.Q THEN nProcessValue:=nProcessValue+1; ELSIF nProcessValue=100 AND ton1.Q THEN nProcessValue:=0; END IF

Find General Control in tool box, add mapping via Variable. Also, Maximum Value and Minimum can be set.

Pic 0-48

Clement name	Gencleminist_1
Type of element	Progress bar
Variable	Genera1_Contro1.nProcessValue
Minimum value	0
Maximum value	100
Style	Blocks

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

Expression	Туре	Value	
nProcessValue	INT	90	

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

```
General_Control X Visualization

      1
      PROGRAM General_Control

      2
      VAR

      3
      bSelectl:BOOL;

      4
      bSelect2:BOOL;

      5
      bSelect3:BOOL;

      6
      END_VAR

      7
      7
```

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

e e	Text ID	48
Select A		
Select B	 Variable	Genera1_Contro1.bSelect1
Select B	Frame size	From style
Select C	Texts	
	Text	Select A
	Tooltip	

Online running as below:

Pic	0-53
-----	------

Select A	Expression	Туре	Value	
	bSelect1	BOOL	TRUE	
Select B	bSelect2	BOOL	FALSE	
Select C	bSelect3	BOOL	FALSE	

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, ad	ld Radio button	radio bu	itton.
	Pic 0-5	4	
	\simeq	Radiobutton	•
	•	0	•
	•	•	•

Add example variable in program window:

1	Genera1_Contro1 🗙 🕘 Visualizatio
	1 PROGRAM General_Control
3	2 VAR
Ľ	3 nSelect:INT;
L	4 END VAR
	5

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

Property	-	Value
Variable	1	Genera1_Contro1.nSelect
Number of colu	mns	1
Radio button or	rder	Left to right
Frame size		From style
Row height		From style
Text properties	3	
Usage of		Default style values
State variables		
Invisible		
Deactivate	inputs	
Radio button set	ettings	
Radio butto	on	2 Create new
🖃 Areas		
= [0]		🗙 Delete
	Text	BWM 3
	Tooltip	
	Line spacing in pixels	0
		🗙 Delete
	Text	Audi
	Tooltip	
	Line spacing in pixels	0
= [2]		X Delete
	Text	Mercedes
	Tooltip	
	Line spacing in pixels	0

Check online running, only one option can be selected among three at the same time. **Pic 0-57**

BWM	Expression	Туре	Value
 Audi 	nSelect	INT	1
Mercedes	•		

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 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

Devices • 4	X I Visualization_1 X		Add Alarm configuration X
 ☐ Visualization_Sample ☐ Device (HCQ1-1300-D) ☐ ① PLC Logic 			Creates an alarm configuration
Constant Application Constant Applica	Cut Copy Paste Delete		Name:
Progressba		🕅 Alarm configuration	
😑 🌃 Task Confi	9	Application	-
🖹 🚸 MainTa 🕞	Properties	Axis Group	
	Add Object	🔁 Cam table	
- 셴 Ge - 셴 PL · · · · · · · · · · · · · · · · · · ·	Add Folder Edit Object Edit Object With	Image: CNC program Image: CNC settings Image: Data Sources Manager Image: DUT	
📲 Traffic_Sig 😋	Login	External File	
Visualizatio	Delete application from device	Global Variable List	Add Cancel

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							
= 🜍 Application							
🗏 📶 Alarm Configuration							
AlarmClass	*	Cut	1				
- 🖄 Error		Сору					
	麅	Paste					
	×	Delete					
- 🖄 AlarmGroup	G	Properties					
- 🖄 AlarmStorage	¥	Add Object	•	-	Alarm class		
AlarmGroup		Add Object Add Folder	•				
Library Manager		Edit Object			Alarm group	9)	
	D°		-		Text List		
PLC_PRG (PRG)		Edit Object V	/itn				
Add Alarm class		×	Add Alarn	n are	quo		×
				Ŭ			
Creates an alarm class			🔼 Cre	ates	an alarm group		
Name:							
AlarmClass			Name: AlarmGroup				
			Marmorou	-			
Add	Can	cel				Add	Cancel

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.

Priority: 10	Acknowledgeme Acknowledgem acknowledg	ent method:	REP_ACK	<u> </u>	
Notification actions	3				
Action Click here to add a		deactivate	confirm	Details Click here to add a new not	Deactiva
< Details					3
Details Display options for al		iller	4		
Details		nner(4	Backs	round color
Details Display options for al State		iller	4)	Backç	

- 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:

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:

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

/ 💧 Alar	mGroup X			•	
Textlist:	📑 AlarmGroup 🗸 Archivi	ng: 👻(none) 🗸	Deactivation:		Archiving: 🗞 (none) 🗸
C Details	Observation type Click here to add a new alarm	Details Click here to add a new alarm	Deactivation	>	De AlarmStorage

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

Pic 0-65

Devices 👻	φ×	/ AlarmGroup 🛛 🕅 Alarm	Storage X	
Visualization_Sample Visualization_Sample Visualization_Sample Visualization_Sample Visualization Visuali	•	Subdirectory:	0	
Marning Marning AlarmGroup		Circular buffer		

Trigger types as below:

Туре	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<=< td=""></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 uneditable 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.





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

Columns	💥 Create new	Time stamp V
Column		
[0]	× Delete	Bitmap
Column header		Time stamp
Use text alignment in title		Time stamp active
Width	120	Time stamp inactive Time stamp acknowledge
Type of data	Time stamp	Value
Text alignment	Centered	Message
 Color settings 		Priority
Activate color settings		Class
		State
		Latch variable 1
■ [1] ■	× Delete	Latch variable 2

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

 Control variables 	
Acknowledge selected	
Acknowledge all visible	
History	
Freeze scrolling position	
Count alarms	
Count visible rows	
Current scroll index	
Current sort column	
Variable for the sorting direction	

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

			Add Alarm configuration	×
			Creates an alarm configuration	
Devices	▼ ₽ X		Name:	
□ □ Untitled7	▼ 1 PROGRAM PLC_PRG □ 2 VAR		Alarm Configuration	
Device (HCQ1-1300-D)	3 rTempe			
PLC Logic	- DOULLA	Alarm configuration		
Application	Cut	Application		
Library Manag	Сору	Axis Group		
PLC_PRG (PR	A. Paste	🙆 Cam table		
🖻 🧱 Task Configur	Delete	CNC program		
🖻 🍪 MainTask 🥎		🔬 CNC settings		
DIC_	Refactoring +	Data Sources Manager		
LocalDevice	Properties	** DUT		
SoftMotion G 2	Add Object	External File		
	Add Folder	Global Variable List		
Ľ.	Edit Object	image Pool	_	
		⊷ Interface	4	
q	§ Login	 Network Variable List (Receiver) Network Variable List (Sender) 	Add Cancel	_
	Delete application from device	T Persistent Variables	Add Calcel	

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

Pic 0-72				
			Add Alarm group	×
- O Application			Name: MarmGroup	
1 M Alarm Configuration	% Cut Image: Copy Paste X Delete Image: Properties Properties			
PLC_PRG (PRG)	Add Object Add Folder Children	Alarm class Alarm group Text List	Add	Cancel

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

PLC	_PRG 🛛 🖄 AlarmGro	up X			
Textlist:	AlarmGroup	~ Arc	hiving: 🔞(nor	ne)	
ID	Observation type	Details	Deactivation	Class	Message
0	+ Upper limit	PLC_PRG.rTemperature >= 50		A Error	High temperature
1	Lower limit	PLC PRG.rTemperature <= 10		A Error	Low temperature

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	
Control variables	
Acknowledge s	
Acknowledge al	PLC_PRG.bConfirm
History	
Freeze scrolling	
Count alarms	
Count visible ro	
Current scroll i	
Current sort co	
Variable for the	

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

Error time a	larm • Error message a	larm	Error Message State alarm						
0									
1		*	Cut						
2			Сору		A Ala	rm table wizard	_		×
3		ß	Paste					_	
4		X	Delete		Туре	of element(s) to insert: ttor	1		\sim
ACK selected	ACK all visible His	0	Select All			Action	Variable		
		11	Browse		•	Acknowledge selected	bAckSel		
			Create Global Text List			Acknowledge all visible	PLC_PRG. bConfirm		
			Order		, 🛛	History	bHistory		
			Alignment		•	Freeze scrolling position	bFreezeScrlPos		
		1	Group						
		ж,	Ungroup						
			Frame Selection				OK	Cance	.1
			Background				20	Cance	
		\mathbb{R}_{p}^{n}	Multiply visu element						
		$\underline{\mathbf{a}}$	Insert elements for acknowledging	alarms					

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

	Error time	alarm v	Error me	essage alarm		Error Message S	tate alarm
0							
1							
2							
3							
4							
ACKs	selected	ACK all v	isible	History	F	reeze Scrl Pos	

Check online running result:

Pic 0-78

	Error time alarm 🔻	Error message alarm	Error Message State alarm
0	09.05.2019 13:20:25	Low temperature	WaitingForConfirmation
1	09.05.2019 13:20:25	High temperature	Active

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.



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.



/#) Me	easurement		
	1	PROGRAM PLC_PRG	Property	Value
	2 3	VAR iBarValue:INTP;	Element name	GenElemInst_1
	4	END_VAR	Type of element	Bar display
			Value	PLC PRG.iBarValue

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

Pic 0-81

🗏 Scale						
Scale start	0					
Scale end	100					
Main scale	20					
Sub scale	5					
Scale line width	1					

After above, actual running as below:

Pic 0-82

Device.Application				_	_	_		
Expression	Туре	Value						
< iBarValue	INT	56	0.0	20.0	40.0	60.0	80.0	100.0

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.



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.



/ P	Me:	asurement		
	1	PROGRAM PLC PRG	Property	Value
	2	VAR	Element name	GenElemInst_2
	3	rInstrument:REAL;	Type of elem	Meter 90°
	4	END_VAR	Value	PLC_PRG.rInstrument

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	\checkmark						

After above, actual running as below:

Pic 0-87

Device.Application	11111111		
Expression	Туре	Value	0.0 20.0 40 0 60.0 50.0 50.0 50.0 50.0 50.0 50.0 50.0
rInstrument	REAL	45	100.0

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.

0-89



Basic use as below:

Find needed Histogram \coprod 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.



/#	Mea	asurement PLC_PRG X		
	1	PROGRAM PLC_PRG	Property	Value
	2	VAR	Element name	GenElemInst_4
	3	arrHistogram:ARRAY[010] OF INT;	Type of element	Histogram
	4	END_VAR	Data array	PLC_PRG.arrHistogram

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



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

Pic 0-90

Display type	Example	
Bar	100.0 80.0 60.0 40.0 20.0 0.0	100.0 80.0 60.0 40.0 20.0 0.0
Line	100.0 80.0 60.0 40.0 20.0 0.0	100.0 80.0 60.0 40.0 - 20.0 0.0
Curve	100.0 80.0 60.0 40.0 20.0 0.0	100.0 80.0 60.0 40.0 20.0 0.0

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 0-92



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.



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

/	Me	asurement PLC_PRG X		
	1	PROGRAM PLC_PRG	Property	Value
	2	VAR	Element name	GenElemInst_5
	3	rResistor:REAL;	Type of element	Potentiometer
	4	END_VAR	Variable	PLC_PRG.rResistor

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.

🖃 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.

⊟ Scale	🖃 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 °.

Arrow		
Arrow type	Circle	
Color	255, 255, 255	iStart:INT:=220;
Arrow start	220	iEnd: INT:=-40;
Variable	PLC_PRG.iStart	END VAR
Arrow end	-40	
Variable	PLC_PRG.iEnd	

After above, actual running as below:

Pic 0-96



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.



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	
Position	
X	179
Y	161
Width	106
Height	94
Variable	PLC_PRG.bSwitch

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		
Image settings		
Image on	VisualizationStyle.Elem	
Image off	VisualizationStyle.Elem	
Image pushed		
Transparent		
Transparent color	Black	
Isotropic type	Isotropic	
Horizontal alignm	Left	
Vertical alignment	Тор	
Element behavior	Image toggler	

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

Expression	Туре	Value	
bSwitch	BOOL	TRUE	

Ex	pression	Туре	Value	
	bSwitch	BOOL	FALSE	

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				
Position				
X	242			
Y	168			
Width	68			
Height	67			
Variable	PLC_PRG.bLamp			

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

Pic 0-103

Background	
Image	Yellow 🗸
Color of the image, used	Yellow fo Red Green Blue Gray

Change status of "bLamp" and check online running:

Pic 0-104

< bLamp	BOOL	TRUE	
			(m
bLamp	BOOL	FALSE	

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

Position	
X	195
Y	126
Width	60
Height	59
Variable	PLC_PRG.bSwitch

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

Pic 0-106

Background				
Image	Gray 🗸 🗸			
Color of the image, used fo	Yellow Red Green Blue Grav			

Single click Key switch LED to check online running:

Pic 0-107

Expression	Туре	Value	
bSwitch	BOOL	TRUE	
Evenenting	Turne	Value	0_0
Expression	Туре	value	

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 race to visual edit area from Special Control, tool box.



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

Pic 0-111

1	Insert elements for controlling trace
€ Ĵ	Configure trace
5¢.	Multiply visu element
	Background
	Frame Selection
١Ę	Ungroup
6	Group
	Alignment •
	Order •
	Create Global Text List
_	
X	Delete
Ē	Paste
	Сору
Ж	Cut

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

SpecialControl_Trace1	Variable Settings		
L. 💻	Variable:	D	
Add trend variable	Attached y axis:	Default Y axis 🗸 🗸	
	Display variable name		
	Description:	Description of variable to display in tooltip	
elect sample point typ	Curve type:	Line 🗸	
	Graph color:	Blue	
	Line type:	/ Line /	
	Line width:		
differ the end of the second	Line style:	— Solid V	
odify line color、type e	Point type:	None	
	Activate minimum warni	ing 🗍	
	Critical lower limit:	0	
	Warning minimum color	T: Black 🗸	
	Activate maximum warn	ing	
	Critical upper limit:	0	
	Warning maximum colo	r: Red 🗸	
	Dynamic appearance op		
	Variable for visibility:		

SpecialControl_Trace1	Record settings	
PLC_PRG.y	Enable trigger	
	Trigger variable: *	
	Trigger edge:	
	Posttrigger (samples) 0 200ms	
	Trigger leve	
	Task:	~
	Record condition:	
	Comment:	
	Resolution: ms V	
	Automatic restart	
	Displaysettings Advanced Copy from trace	

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.



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

Pic 0-115

🗬 Display Settings	🗬 Display Settings	×
X axis Y axis Display mode	X axis Y axis Display mode (a) Auto () Fixed	10
Minimum: 0 Maximum: 10s Length: 10s ✓ Grid: Ital: (128; 128; 128) Tick marks	Minimum: -12 Maximum: 12 Image: Constraint of the state of	0 -10 -0 1s 2s 3s 4s 5s 6s 7s 8s 980s Badground
Subdivisions: 1 Font: Times New Roman; 9pt v	Subdivisions: 1 Font: Times New Roman: 9pt	From visualization style V White V Reset Use as default
<u>Add y axis</u> Delete y axis	<mark>Add y axis</mark> Delete y axis	OK Cancel

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

VisuTrendStorageAccess.GlobalInstance

Pic 0-116

🚭 Advanced Trace Settings			×
Measure in every n-th cycle: Buffer size (samples):	1 ~	100ms 10s	
10s is the configured time range of the x axis.			
	ОК	Cancel	

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

¥	Cut							
Ð	Сору							
Ċ.	Paste	Ľ	🗳 Tra	ice wiza	ard		- 🗆	X
X	Delete					1		_
	Create Global Text List				Control variable	Variable	Type of element to insert	^
	Order			\checkmark	Reset trigger	bResetTrigger	OK Button	
	Alignment			\checkmark	Start trace	bStart	ok Button	
5	Group			\checkmark	Stop trace	bStop	or Button	
1.	Ungroup		Β		Save trace to a file			
1111	Frame Selection	-		\checkmark	Save trace	bStore	OK Button	
	Background			\checkmark	File name	sStoreFilename	🔁 Rectangle	
ų.	Multiply visu element		в		Load trace from a file			~
€ ∮	Configure trace						0K Cancel	
ĕ	Insert elements for controlling trace						OK Califer	·

Interface as below after ticking all plug-in units:





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.





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

menu.

Pic 0-121	
Initial calls	
Method calls	🗱 Create new
Methods	
Cyclic calls	
Method calls	🛒 Create new
Methods	
Conditional calls	
Method calls	😴 Create new

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

ActiveX PLC_PRG X	 Conditional calls 	
1 PROGRAM PLC_PRG 2 VAR	Method calls	Create new
<pre>3 bPlay:BOOL; 4 sPath:STRING:='C:\HCFA.mp4';</pre>	 Methods 	
L 5 END_VAR	= [0]	X Delete
	Method	
I	Call condition	
	Variable	PLC_PRG.bPlay
	Parameters	
	Parameters	💥 Create new
	🖃 Parameter	
	Variable	PLC_PRG.sPath
	Result parameter	

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

Pic 0-123

	WM	IPlayer.OCX.	7	
PLAY				

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

	ODESYS														-		×
File		View	Project	Build	Online		Tools	Window									
1	New Proje					Ctrl+N	જી જે	省目島	御• 🖻	≊ 0; (0ğ	🕨 🖬 火 🖓	F≣ ¢⊒ +≣ \$		8 = ?	2	
2	Open Pro	-			C	Ctrl+O											
	Close Pro	-					🚯 Sta	rt Page 🗙	٢								
H	Save Proje				(Ctrl+S	- 1		ESYS V3.	5 SP14 P	atc	h 1					
	Save Proje																
	Project Ar	chive				•							1 - 1 - 1 N				
	Source up	oload					Bas	ic Opera	ations				Latest Nev	/S			
	Source do	ownloa	ad					1 Nev	w Project								
5	Print							💕 Оре	en Project								
	Print Previ	iew						🕤 Оре	en Project fr	om PLC			- 🍙				
	Page Setu	ıp											CODES	rs			
	Recent Pro	oiects				•	Rec	ent Proj	jects				Inorman	abou	t the se	curity	
	Exit	-,				Alt+F4		🚔 Acti	iweY				vulnerabi	lities CV	/E-2021	-44228	}
_	Exit					AIT+F4		Unt					and CVE	-2021-4	5046 in	Loa4i:	
								🚔 Spe					Log4j is r			5,	
								🚔 Lan					CODESY			o of ou	
								_	asurement o	ontrol							<u> </u>
								🚔 Der					products				
								_	fter project l	oad			2021-442	28 or C	VE-202	1-	
e De	vices 👔 PC)Us						ow page or					45046.				`
_	ges - Total 0		0 warning	1(c) 0 moor			_									-	ą :
16220	yes - rotal U	enor(s)	n o warring	(o), o mess	age(s)	- 0 0	error(s)	 0 warn 	ning(s)	0 message	e(s)	××				•	4
)ecr	ription										- (-)	Project	Object		Positio		_
/est	inpuon											rioject	object		FUSICIO		
d Wa	itch 1 🔳 M	essage	s - Total 0 e	error(s), 0 v	varning(s),	0 message(s)											
									Last build:	0 • 0 0	P	recompile: 🧹	Pro	ject user: ((nobody)		\bigcirc

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

New Pro	oject		Sele	ct project t	ype ×
Categories	:	Templates:		Ļ	
	raries ojects	Empty project	HMI project	Standard project	Standard project w
	ontaining one device, o	one application, and an	empty implemen		roject name
Name:	Q1test				
Location:	D:\Q1				~
	Selec	t save path	Г	OK	Cancel

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		
Standard	Project	×
	objects withir - One progran - A program P - A cyclic task	t to create a new standard project. This wizard will create the following n this project: mmable device as specified below LC_PRG in the language specified below which calls PLC_PRG to the newest version of the Standard library currently installed. Select target device
	Device:	HCQ1-1300-D (Zhejiang Hechuan Technology) 🗡 🗸 🗸
	PLC_PRG in:	Structured Text (ST)
	S	elect program language of the main program OK Cancel

5.3 Establish Communication with Q1

Pic 0-2

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	
以太网 未识别的网络 Realtek PCIe GbE Family C	Contr
📱 以太网 Status 🏮 以太网 Properties	Internet 协议版本 4 (TCP/IPv4) Properties X
General Networking Sharing	General
Connection IPv4 Connectiv IPv6 Connectiv Media State: Duration: Speed: Details Activity Connect using: Connect using: Connect using: Connect using: Connect using: Connect using: Microsoft 网络客, Connection uses the for Microsoft 网络客, Microsoft 网络客, Microsoft 网络客, Microsoft 网络客, Microsoft 网络客, Microsoft 网络客, Microsoft PM络这, Microsoft LLDP to Internet 协议版本	ollowing items: 户端 文件和打印机, 程序 Subnet mask: 192、168、88、200 255、255、0 Default gateway:
Bytes: Description 传输控制协议/Internet 协议,用于在不同的相	Alternate DNS server: 相互连接的网络
1 Properties	Validate settings upon exit Advanced OK Cancel

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

Devices 👻 🕈 🗙	Device X	
QItest	Communication Settings 2 Scan network Gateway - Device -	
PLC Logic	Select Device	X
Application Library Manager LC_PRG (PRG) SoftMotion General Axis Pool	Select the network path to the controller: Gateway-1 (scanning) HCQ1-1300D [0064] HCQ1-1300D [0064]	Scan network Wink
<	0X	Cancel
Sevices POUs		

For questions or comments, feel free to email <u>400@hcfa.cn</u>

If gateway is not opened, it will display in red in Communication Device and need open manually. **Pic 0-3**

Files	Gateway Gateway IP-Address: localhost Port:	Devic	(active)
Files	Gateway Gateway IP-Address: localhost	Devic	ł] (active)
Files	Gateway Sateway IP-Address: localhost	Devic	ł] (active)
Users and Groups	Gateway-1 IP-Address: localhost	Devic	
PLC Shell Users and Groups	IP-Address: localhost	Devic	
PLC Shell Users and Groups Access Rights		HCQ1	
		Devic	-1300D e Address:
Access Rights	1217	0064	
		Targe 16C7	
Symbol Rights		Targe 4102	t Type:
Task Deployment			t Vendor: ang Hechuan Technology
Status		-	t Version:

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



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 \rightarrow Install \rightarrow Library needed for installation \rightarrow Open to finish installation. **Pic 0-1**

Q1test.project - CODESYS		-	
File Edit View Project Librari	es Build Online Debug Tools Window Help		₹
12 📽 🖬 🍯 🗠 🗠 🕹 🖷 🗙	♣ \$\$ ♣ \$\$ ■ \$1 \$1 \$1 \$1 \$1 \$1 \$6 \$100 \$1 \$100 \$1 \$100 \$1 \$100 \$1 \$100 \$1 \$100	₹	Ŧ
Devices 🗸 🕈 🗙	/ 👔 Library Manager 🗙		-
Citest	😫 Add library 🗙 Delete library 🖙 Properties 🟐 Details 🖃 Placeholders 🎁 Library repository	(1)	
Device (HCQ1-1300-D)	in Library Repository		×
Application			
Library Manager	Location: System 🗸	Edit Lo	ocations
PLC_PRG (PRG)	(C:\ProgramData\CODESYS\Managed Libraries)		
🖹 🎆 Task Configuration			
PLC_PRG	Installed libraries:	In	stall
LocalDevice	Company: (All companies)		
SoftMotion General Axis Pool		Un	ninstall
	(Miscellaneous)		×
	← → ↑ ↑ ≪ 本地磁盘 (> Q1 > v 0 P 在 Q1 中搜索		
	组织 ▼ 新建文件夹 目== ▼		0
< >	文档 ^ 名称 修改日期		类型
Cevices POUs	➡ 下载 Module description file 2022/4/13 17:0	01	文件:
Messages - Total 0 error(s), 0 warning(s), 0 mess	♪ 音乐 3 □ HCFA_HSIO_ATCLib_1.0.0.7.compiled 2022/4/6 14:06	i.	CON
incode a crocory o warming (b); o mas		-	
	🏪 本地磁盘 (C:)		
Description	本地磁盘 (D:)		
			·
Watch 1 🔳 Messages - Total 0 error(s), 0 v	文件名(N): Compiled library file	łs	~
	4 打开(O)	取消	

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

ocation:	System (C:\ProgramData\CODESYS\Managed Libraries)	~	Edit Locations.
Installed lib	raries:		Install
Company:	(All companies)	~	Uninstall
	1iscellaneous) ■ HCFAModbusLib <i>HCF</i> A	^	Export
	•• 1.2.0.0		
	- • · · · · · · · · · · · · · · · · · ·		
	DCS		
H			
B 🗄 S		~	Find
<		>	Details
Group l	by category		Dependencies.

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

File Edit View Project Lib	raries Build Online Debug Tools Wind	dow Help	T
E 🗳 🛃 🎒 ∽ റ 🍓 🖬 🛍	× ぬいぬとい 11 11 11 11 11 11 11 11 11 11 11 11 1	ר מון	
)evices 👻 म्	X Library Manager X		
Q1test	🚹 🔁 Add library 🔀 Delete library 🛛 🕾 Properti	es 📷 Details 🛛 🔄 Placeholders 🛛 🎁 Library repository	
□ 🛐 Device (HCQ1-1300-D)	Add Library	x	Effective ver
PLC Logic	Addebiary	~	3.5.14.0
- O Application	String for a fulltext search		3.5.5.0
Library Manager	Library	Company	3.5.13.0
PLC_PRG (PRG)	Rectance in the second	company	4.4.0.1
⊟-∭ Task Configuration ⊟-⊗ MainTask			4.4.0.0
PLC_PRG	• Standard	System	4.4.0.0
	Standard64	System	>
SoftMotion General Axis Pool		System	
	Composer		
	• Se AC_Alarming	3S - Smart Software Solutions GmbH	
	→ C AC_DataLog	3S - Smart Software Solutions GmbH	
	AC_DeviceDiagnosis	3S - Smart Software Solutions GmbH	
	> AC_ModuleBase	3S - Smart Software Solutions GmbH	
POUs	AC_Persistence	3S - Smart Software Solutions GmbH	
lessages - Total 0 error(s), 0 warning(s), 0 i	nessa AC_Trend	3S - Smart Software Solutions GmbH	→ 쿠 :
	⊨		
Description		3S - Smart Software Solutions GmbH	
Description	• • EtherNetIP Services	3S - Smart Software Solutions GmbH	n
	•100 ParServer	3S - Smart Software Solutions GmbH 🗸	
Watch 1 📄 Messages - Total 0 error(s)	0 wa Advanced	3 OK Cancel	

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.



Add Task	×
A IEC task	
Name:	
Task	
	Add Cancel

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

Devices • 4 ×	Fill in teals priority		•
Device (HCQ1-1300-D) Device (HCQ1-1300-D) Device (HCQ1-1300-D) Device (HCQ1-1300-D) Device Application Device Application Device An antask Device A LocalDevice SoftMotion General Axis Pool	Priority (031): 1 Type Cyclic Interval (e.g. t#200ms): 4 Watchdog Select task type Set cycle time Time (e.g. t#200ms): Sensitivity: 1	ms V	< .
Input Assistant	-		\times
Text search Categories	Name Type Origin Application Application PLC_PRG PROGRAM		

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:

Devices • 7 X	Add POU ×
□ ① PROGRAM PLC_PRG □ □	Create a new POU (Program Organization Unit)
Ibray Mana	Name: POU Type Program Punction Block Extends: Implements: Final Abstract Access specifier: Method implementation language: Structured Text (ST)
	Function Return type:
	Implementation language:
	Structured Text (ST) Continuous Function Chart (CFC) - page-oriented Function Block Diagram (FBD) Ladder Logic Diagram (LD) Sequential Function Chart (SFC)





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	Auto Declare				×
Scope: range of variable		Scope: VAR	~	Name: fbTimer 1	Type: TON	~ >
Name: name of variable		Object:		Initialization:	Address:	
Type: variable data type		POU [Application]	\sim	Comment:		
Object: corresponding application	ı	CONSTANT RETAIN PERSISTENT		Timer		^
Initialization: initial value		D PERSISTENT				~
Address: mapping between varia				ОК	Cancel	
Flags: variable type can be constant, lossless or persistent type						
Comment: comment in format of "comment content"						

Pic 0-9

1	PROGRAM PLC_PRG	Variable declaration comment
2	VAR	
з	fbTimerl	:TON; //Timer
4	index	:INT:=10;
5	iVar	:INT; //Intermediate variable
6	iRectangle	:INT; //Square wave signal, amplitude is adjustable
7	iRatiol	:INT:=5; //"1"Signal ratio
8	iRatio2	:INT:=5; //"O"Signal ratio
9	END VAR	
10	Carlo Carlo Carlo Carlo	annotation

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

Expression	Туре
🗏 < fbTimer1	TON
🍫 IN	BOOL
牧 рт	TIME
™ ≱ Q	BOOL
🍫 ET	TIME
1 fbT	imerl.IN

Module Calls: FNC, direct and needs not statement





Programming below per above rules:

库

fbTimerl(IN:= , PT:= , Q=> , ET=>);

Pic 0-11

E {} SM3_Basic

1

PLC_PRG X ('The example program is a square wave with an adjustable duty cycle, Give constants Ratio1 and Ratio2, The period length and proportion ofthe square wave Β iRectangle"1" and "0" can be set*) fbTimer1(IN:=NOT fbTimer1.Q, PT:=T#500MS); CASE index OF 4 5 10: irectangle:=1; 6 7 IF iVar<iRatiol AND fbTimerl.Q THEN iVar:= iVar+1; ELSIF iVar>iRatiol OR iVar=iRatiol THEN 9 10 index:=20; 11 iVar:=0; 12 END IF 14 20: 15 irectangle:=0; 16 IF iVar<iRatio2 AND fbTimerl.Q THEN Β iVar:= iVar+1; 17 18 ELSIF iVar>iRatio2 OR iVar=iRatio2 THEN 19 index:=10; 20 iVar:=0; 21 END IF END CASE 22
5.5 Add Trace to monitor program variable

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



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.

Trace Configuration				×
Trace Record	Variable settings			
	Variable:	-0		
PLC_PRG.iRectangle	Graph color:	Green	~	
	Input Assistant		· · ·	
	Text search Categories			
	Trace Variables	▲ Name	Ture	Address
			Туре	Address
	Traceable parameters	= 🔘 Application	Application	
Presentation (diagrams)		🖻 📄 PLC_PRG	PROGRAM	
- Time axis		🗄 🛷 fbTimer1	TON	
Diagram 1		🖉 🖗 index	INT	
Y axis		🔷 🕴 iRatio1	INT	
- Shown variables		🖉 iRatio2	INT	
- PLC PDC (Dectored)		🕴 iRectangle	INT	
PLC_PRG.iRectangle			INT	
PLC_PRG.IRectangle		🔷 🖗 iVar	2/1/	
		🕒 🖗 iVar 🗷 🕼 🎑 IoConfig_Globals	VAR_GLOBAL	

Single click task that follows Trace in left tree menu, select MainTask as example in drop down list. **Pic 0-4**

Trace Configuration		×
Trace Record	Record Settings	
□ Trace	Enable Trigger	
PLC_PRG.iRectangle	Trigger variable:	
	Trigger edge:	
	Posttrigger (samples)	
	Trigger Level:	
	Task:	~

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

🎓 Q1	test.pro	oject - C	CODESYS										
File	Edit	View	Project	Build	Online	Debug	Tools	Window	Help				
1		3	CH 🐰 🗎	ŝΧ	🐴 🕼	🐴 🚰 🔳	M M	×	🏪• 🗋	**	OŞ Oğ	ŀ	8

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

Pic 0	-2		
(ODESYS	3	\times
	?	Warning: The application 'Application' is currently in RUN mode on the PLC. However, do you want to download the latest code changes?	
		Yes No Details	

After lo	ogin, c	lick St	art logo	▶ to	run.							
Pic 0-3												
🍺 Q1	test.pro	oject - (CODESYS									
File	Edit	View	Project	Build	Online	Debug	Tools	Window	Help			
1		6	CH 🐰 🗎	$\mathbb{R}\times$	#1 🖓	🐴 🚰 📕	% %	M 🛱	‱ • ⊡`	🔛 Oş	¢	= 🖄

Running interface as below after correct login:

Q1test.project - CODESYS						- 🗆	>
File Edit View Project B	uild Online D	ebug Tools Wind	dow Help				
] ⊭≣ ∰ ∽∝% ⊫®®	× (# 35 #		🖷 i ዀ - ㎡ i 🕮 i	98 9 8 98 10	1 F1 C1 +1 S	→ 글 ∛	
						1 1 1 1	
)evices 🗸 🗸 🖡	× 🚭 Trace	PLC_PRG X					
Q1test	Device.Ap	plication.PLC_PRG					
😑 😏 🗊 Device [connected] (HCQ1-	1300 Expression	Туре	Value	Prepared value	Address	Comment	^
E B PLC Logic	🗄 🌶 fbTim					Timer	
Application [run]	index	INT	10				
Library Manager	🔷 iVar	INT	1			Intermediate variable	
PLC_PRG (PRG)	iRecta	ngle INT	1			Square wave signal, ampli	tuc
POU (PRG)	< iRatio	1 INT				N ratio	~
i≣ S MainTask	<						>
	E 1	*The example progr	am i Var	riable and prog	ram Online	stants Ratio	01 an
	⊟ 2	iRectangle"1" and	i "O'			I. I.	
🗬 🕄 Trace	3 🔴	fbTimerl(IN		monitor status	s display	MS);	
		CASE index 10	OE				
😔 🔁 SoftMotion General Axis	Pool 5	10: irectangle	1 .=1;				
				AND fbTimer1.0 FALSE	THEN		
	8		:= iVar 1 +				
	8 9 0			OR iVar 1 =i	Ratiol 5	THEN	
	> 10	index	10 :=20;			100	%
Cevices POUs	<	-)
lessages - Total 0 error(s), 0 warning(s), () message(s)						- 7
Juild	•	3 0 error(s) 😗 0 v	warning(s) 🐧 0 me	ssage(s) 🗙 💥			
Description				Project	Object	Position	
The application is up to date		_		.))			
Build complete 0 errors, 0 warnings :	ready for download!	Pro	gram run sta	tus			
	,		/				
Breakpoints 🗐 Messages - Total 0 err	or(s), 0 warning(s), 0	message(s) 🔝 Watch 📣	/				

Download Trace for sampling running trace of variable.

Pic 0-5

19111000000000000000000000000000000000		6 <u>4</u> % %	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	> • 4°
Devices 👻 🗸	× 🚭 Trace	e X		
Q1test G Original Connected] (HCQ1-130	▼ 0-D)	10	Add Variable	
PLC Logic		- 15	Download Trace	
Application [run]			Start Trace	
Library Manager		TO	Stop Trace	
PLC_PRG (PRG)			Reset trigger	
POU (PRG)			Autoscroll	

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 \rightarrow Watchr \rightarrow Watch 1, then list of Watch 1 will be created and 4 Watches can be created.

Pic 0-7

🐞 Q1	test.pro	oject	- CODESYS								
File	Edit	Viev	w Project	Build	Online	Debug	Tools	W	lindow	Help	
1		2	Devices			Alt	+0	1 2	1	🏧 - 🕤 🔮	9 Q
		٦	POUs			Alt	+1				
Devices		\$	Modules			Alt	+2				
0 O1te	ect		Messages			Alt	+3				
	Device (H		Element pro	perties							
	PLC I	33	ToolBox								
	= O .		Watch				•	F	Watch	1	
	1		Cross Refer	ence List				<u>ليکم</u>	Watch	12	
	-	\$2	Call Tree					K a	Watch	13	
	-[5	Bookmarks					1	Watch	14	
	= (Ð	Breakpoints					R	Watch	all Forces	
		阳	Call Stack								
		٩	Memory								
	- 6		Online Chan	ge Mem	ory Reser	ve Setting:	s				
	& Loca		Start Page								
	Soft	Ø	Security Scre	en							
<			Store								
😤 Devic			Choose per	pective			•				
Message	s - Total		Full Screen		Ct	trl+Shift+F	12				
Build		h	Properties					P	0 warn	ing(s) 🟮 0	mess

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**

Watch 1							→ ₽ X
Expression	Application	Туре	Value	Prepared value	Execution point	Address	Comment
PLC_PRG.iRectangle	Device.Application	INT	<not in="" logged=""></not>		Cyclic Monitoring		Square wave signal, amplit

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

Watch 1							- ₽ X
Expression	Application	Туре	Value	Prepared value	Execution point	Address	Comment
PLC_PRG.iRectangle	Device. Application	INT	<not lg<="" td=""><td></td><td>Cyclic Monitoring</td><td></td><td>Square wave signal, amplit</td></not>		Cyclic Monitoring		Square wave signal, amplit
I			On	line Watch			
🔊 Watch 1 📄 Messages - Total 0 err	ror(s), 0 warning(s), 0 message	(s)		·	;		
	Last build: 😗 0	• 0 Preco	mpile: 🧹 🛛 🎧	Project user:	(nobody)	INS Ln	1 Col 1 Ch 1 (🕖

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
on left and can be released via Release Value

Pic 0-10

File Edit View Project Build Onlin		bug Tools Window Help		-				
È 📽 🖬 ⊕ ∽ ∝ ½ 🖻 🖻 X M (i, s € →	Start Stop Single Cycle	F5 Shift+F8 Ctrl+F5	> •	n [= 4] n	*= \$ \$ # #	₹ % /	
levices		· · ·	Ctri+F5					
Q1test	1	New Breakpoint						
= 🈏 🔟 Device [connected] (HCQ1-1300-D)	5	Edit Breakpoint			Value	Prepared value	Address	Comment
PLC Logic		Toggle Breakpoint	F9					Timer
= 💮 Application [run]	0	Disable Breakpoint			20			
Library Manager	•	Enable Breakpoint			2	10		Intermediate variable
PLC_PRG (PRG)	ÇΞ	Step Over	F10		0			Square wave signal, amplitude is adjustable
POU (PRG)	9 <u>–</u>	Step Into	F8		5			"1"Signal ratio
E I Task Configuration	e_	Step Out	Shift+F10		5			"0"Signal ratio
	÷ <u>≡</u>	Run to Cursor						
	3	Set next Statement						
	\$	Show next Statement		<u> </u>				
SoftMotion General Axis Pool		Write values	Ctrl+F7		a square wav can be set*)	e with an adjustal	ble duty cycl	e,Give constants Ratio1 and Ratio2,
		Force values	F7			1.0 FALSE , PT	T#500ms	:=T#500MS);
		Force values Unforce values	F7 Alt+F7	OF				
		Toggle Flow Control mode	AIT+F7	0:	-1.			
	-			2 <10>	<iratiol< td=""><td>5 AND fbTimer</td><td>1.Q FALSE THE</td><td>N</td></iratiol<>	5 AND fbTimer	1.Q FALSE THE	N
		Core Dump	•	2 <10				
		Display Mode	•		<10> >iRat :=20;	iol 5 OR iVar	2 <10>	=iRatiol 5 THEN
		11 12 13 14 15	iVa END_IF 20: irectangle	r 2	<10> :=0;			

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	c 0-'	11				
On	line	Debug	Tools	Window	Help	
O Ş	Log	in			Alt+F8	
0Ş	Log	out		C	Ctrl+F8	
		ate boot ap vnload	plicatio	n		
	Onli	ine Change				
	Sou	rce downlo	ad to co	onnected d	evice	
	Mul	tiple Downl	oad			
	Res	et warm				
	Res	et cold				
	Res	et origin				
Γ	Sim	ulation				
	Secu	urity				•
	Ope	erating Mod	le			•

Notice: X: reserved value O: initial value

Online Command	VAR	VAR RETAIN	VAR RETAIN PERSISTENT
Reset warm	0	Х	x
Reset cold	0	0	х
Reset origin	0	0	0
Login with Download	0	0	х
Login with Online change	Х	Х	х

Pic 0-12

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 \rightarrow Simulation to enter simulation mode, tick and the logo \checkmark will appear on left of simulation.



Entering the mode and red logo of simulation will appear at bottom of the interface. **Pic 0-2**

		4.01				∇				_
PLC_PRG	B	1 (-	an adjustable	duty cycle,Give	constants H	latiol an	1d
🚭 Trace	B	2	iRectangle"	1" and "0" can	be set*)					
		3	fbTime	erl(IN:=NOT fb]	imerl.Q, PT:=T#	500MS);				
	B	4	CASE inde	x OF						
SoftMotion General Axis Pool		5	10:							
		6	irectar	ngle:=1;						
	B	7	IF iVa:	r <iratiol and="" f<="" td=""><td>bTimerl.Q THEN</td><td></td><td></td><td></td><td></td><td></td></iratiol>	bTimerl.Q THEN					
		8	iVa	ar:= iVar+1;						
	B	9	ELSIF	iVar>iRatiol (R iVar=iRatiol	THEN				
		10		index:=20;						
>		11		iVar:=0;					100 %	a .
Devices POUs	<	12	DIM TD							>
essages - Total 0 error(s), 0 warning(s), 3 me	ssage(s))							↓ ậ	,
uild			O error(s)	• 0 warning(s)	3 message(s)	\times ×				
escription						Project	Object	Position		1
Total allocated memory size for code and d	ata: 845	256 bytes								
Compile complete 0 errors, 0 warnings										
Watch 1 Messages - Total 0 error(s), 0	warning	1(s), 3 mes	sage(s)			+				
					_					

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

Q1test.project* - CODESYS						- 🗆	>
File Edit View Project Build Onli] 🎬 🗐 🚭 🗠 여 🐰 🖻 🛍 🗙 🏘 연	5	Window Help	OŞ Oği)	🔹 🖋 (ji ci ci	⊒ *⊒ & ¢]	£ ≓ ∛⁄	
evices 🗸 🕈 🗙	PLC_PRG X						
Q1test 🔹	Device.Application.	PLC_PRG					
▲		Type Val	ue	Prepared value	Address	Comment	^
🖹 🚫 Application [run]		INT 10					
Library Manager	🔹 iVar I	NT 4	i.	variable mo	odify Area	ntermediate variat	,
PLC_PRG (PRG)	iRectangle I	INT 1				quare wave signal	,
□ - 媛 Task Configuration □ - 愛 MainTask □ - 御 PLC_PRG	∲ iRatio1 I	INT 5	•	1722277		"1"Signal ratio	~
	⊟ 4 CASE	index 10 OF					
SoftMotion General Axis Pool	5 1 6 1 1 7 1 8	0: rectangle 1 :=1; F iVar 4 <iratiol iVar 4 := iVar LSIF iVar 4 >iRe index 10 :=2</iratiol 		line program	display are	ea 	
	5 1) 6 1: 1: 7 1: 8 1 8 1 8 1 8 1 8 1 8 1 8 1 8 1 8 1 1 1 1	0: rectangle <u>1</u> :=1; F iVar <u>4</u> <iratioj iVar<u>4</u>:= iVar LSIF iVar<u>4</u>>iRa</iratioj 		line program	display are	'	
Devices D	5 11 6 12 11 12 13 10 10 10 10 10 10 10 10 10 10	0: rectangle 1 :=1; F iVar 4 <iratiol iVar 4 := 1Var LSIF iVar 4 >iRe index 10 :=2</iratiol 	.tx		display are	100 %	, с т
Devices Devices POUs essages - Total 0 error(s), 0 warning(s), 0 message(s)	5 11 6 12 11 12 13 10 10 10 10 10 10 10 10 10 10	0: rectangle <u>1</u> :=1; F iVar <u>4</u> <iratioj iVar<u>4</u>:= iVar LSIF iVar<u>4</u>>iRa</iratioj 	.tx	line program	display are	100 %	3
Devices I POUs	5 11 6 12 11 12 13 10 10 10 10 10 10 10 10 10 10	0: rectangle 1:=1; F iVar 4 (:Ratio) iVar 4 := iVar LSIF iVar 4):R index 10 := 0 warning(s) 0 n	nessage(s)	××	display are	100 %	2
2 Devices 1 POUs essages - Total 0 error(s), 0 warning(s), 0 message(s) uild escription	5 11 6 12 10 7 0 11 8 0 8 10 0 8 10 0 error(s)	0: rectangle 1 :=1; F iVar 4 <iratiol iVar 4 := 1Var LSIF iVar 4 >iRe index 10 :=2</iratiol 	nessage(s)	××		100 %	2
Devices Device	s 11 c 11 l 7 11 s 0 s 0 s 0 s 0 c 10 s 0 c 11 s 0 c 11 s 0 c 11 c 11	0: rectangle 1:=1; F iVar 4 (ARatio) iVar 4:= iVar LSIF iVar 4):= index 10 := 0 warning(s) 0 n Status b	nessage(s)	××		100 %	2

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**

Q1test.project* - CODESYS				-		×
File Edit View Project Build	Online Debug Tools Wind	dow Help				₹
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	•		 • • • • • • • • • • • • • • • • • • •	도 속 제 않는 아 () \$ [하] \$		•
			P			
Devices 👻 🕈 🗙	Device X					•
Ottest Ottest Over (connected) (HCQ1-1300-D)	Communication Settings	 plcload				
드 · 페이 PLC Logic 드 · · · · · · · · · · · · · · · · · · ·	Applications	PLC load average:	3%			
Library Manager	Backup and Restore	CoreID:	0			
PLC_PRG (PRG) Task Configuration	Files	PLC Core load:	34			
B S MainTask	Log	 plcload				
Trace	PLC Settings	PLC load average:	3%			
- 😯 🍐 LocalDevice - 😯 🍐 SoftMotion General Axis Pool	PLC Shell	CoreID:	3 % 0			
	Users and Groups	PLC Core load:	38			
	Access Rights					
	Symbol Rights	plcload				
	Task Deployment	PLC load average:	38			
	Status	CoreID:	0	Command input box		
	Information	PLC Core load:	34			
		blcload				_
< >		picioad				~
2 Devices POUs	<					>
Messages - Total 0 error(s), 0 warning(s), 3 messag			_		-	ч х
Build	• 0 error(s) • 0	warning(s) 🜖 3 message(s)	X X			
Breakpoints Messages - Total 0 error(s), 0	warning(s), 3 message(s) 🔀 Watch 1					
Device user: Anonymous Last build:	🖸 0 🕐 0 🛛 Precompile: 🧹 🏫	RUN Prog	ram loaded	Program unchanged Project user:	(nobody)	Q

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

2	
Prints list of available commands.	
yetcmdlist	
Used internally to display all available commands.	
nem <address> [<size>]</size></address>	
Print Hexdump of spezified 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="">]</app></app>	
Dump Guids of one specific or all loaded applications.	
pinf [<app name=""> <app index="">]</app></app>	
Dump Project informations of one specific or all loaded applications.	
startprg [<app name=""> <app index="">]</app></app>	
Start one specific or all loaded applications.	
stopprg [<app name=""> <app index="">]</app></app>	
Stop one specific or all loaded applications.	
resetprg [<app name=""> <app index="">]</app></app>	
Reset one specific or all loaded applications.	
resetprgcold [<app name=""> <app index="">]</app></app>	
Perform a cold reset one specific or all loaded applications.	
reload [<app name=""> <app index="">]</app></app>	
Reload one specific or all loaded applications from their bootprojects.	
jetprgprop	
[not implemented, yet]	
getprgstat [<app name=""> <app index="">]</app></app>	
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>]</size></address>	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="">]</app></app>	A specific jump or all loaded apps			
pinf [<app name=""> <app index="">]</app></app>	A specific jump project information or all loaded apps			
startprg [<app name=""> <app index="">]</app></app>	Launch a specific or all apps			
stopprg [<app name=""> <app index="">]</app></app>	Stop a specific or all apps			
resetprg [<app name=""> <app index="">]</app></app>	Reset a specific or all apps			
resetprgcold [<app name=""> <app index="">]</app></app>	Perform a cold reset for a specific or loaded application			
reload [<app name=""> <app index="">]</app></app>	Reload a specific application or application loaded from the startup project			
getprgprop	Target has not been realized yet			
getprgstat [<app name=""> <app index="">]</app></app>	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			
	Set UTC by time string (obey format in "rtc-set YYYY-			
rtc-set	MM-DDThh:mm:ss[,sss]")			
saveretains [<applicationname>]</applicationname>	Save retain power down hold data to a file (specific application)			
restoreretains [<applicationname>]</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.

Communication Settings	plcload		
Applications	PLC load average:	38	
Backup and Restore	CoreID: PLC Core load:	0	
Files	Fic core road.		
Log			
PLC Settings			
PLC Shell			
Users and Groups			
Access Rights	Q1 feedback	k PLC Load I	rate
Symbol Rights			
Task Deployment			
Status			
Information			
	plcload		

[Example 3] Activate/stop applied program with PLC Shell

Pic 0-4

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

Devices 👻 🖬	¥ ¥	Device X	
□ 📋 Q1test	•		
🖃 🎲 Device [connected] (HCQ1-1300-D)		Communication Settings	startprg application
		Applications	Start Application:
Application [stop]		· · ·	Application: Application [OK]
- 📶 Library Manager		Backup and Restore	
PLC_PRG (PRG)		Files	Start Application
Task Configuration		rites	stopprg application
🖃 🥩 MainTask		Log	
PLC_PRG			Stop Application:
🚭 Trace		PLC Settings	Application: Application [OK]
🔂 👌 LocalDevice		PLC Shell	-
🖙 🚱 🍐 SoftMotion General Axis Pool			
		Users and Groups	
			Stop Application
		Access Rights	
		Symbol Rights	
		Task Deployment	
		Status	
		Status	
		Information	
			stopprg application
📽 Devices 👔 POUs		<	

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 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:



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:



In the program, the array Arr1 has only five elements of $1 \sim 5$, 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				
			Add Global Variable List	×
			Create a new global variable list	
Application Copy Cut Copy CredBounds Paste Copy CredBounds Poc_PRG (PRG Paste Poc_PRG Paste Paste Poc_PRG Paste Poc_PRG Paste Paste Poc_PRG Poc_PRG Paste Poc_PRG Paste Poc_PRG Poc_PRG Paste Poc_PRG Poc_PRG	ring	Alarm configuration Application Avis Group Cam table CNC program CNC settings Data Sources Manager DUT External File Global Variable List	Name:	
Cip Login Delete a	~	Image Pool Interface Network Variable List (Receiver)	Add	Cancel

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



Add status variable in CheckBounds as below:

Pic 0-6



Check online running:

Pic 0-7

Device.Application.PLC_PR	G1		
Expression	Туре	Value	Prepared value
🗏 🧳 arr1	ARRAY [15] OF INT		
🧼 arr1[1]	INT	0	66
arr1[2]	INT	0	
arr1[3]	INT	0	
arr1[4]	INT	0	
arr1[5]	INT	0	
< index	INT	0	
<] ??? :=66; <u>RETURN</u>	Boundar	y overflow in array
] ??? :=66; <u>RETURN</u>	Boundar	y overflow in array
] ??? :=66;RETURN Type	Boundar	y overflow in array Prepared valu

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

8	<pre>1 //Implicit build code:Here a 2 ③ IF index 0 < Lower</pre>	re : (suggestions for code implementation THEN				
	3 Chaol Rounde	low			Breakpoint Properti	es	×
Β	You can add a]	Condition Location	Execution point settings	
Β	the line with logging into				POU:	POU [Device: PLC Logic: Application]	~
	10 0	-	/		Position:	Line 6, Column 2 (Impl)	~
	11 O END_IPRETURN		Select All		Instances		
			Browse +				
			Advanced +				
		<u>0</u>	Input Assistant				
		1	New Breakpoint		2		
			Toggle Breakpoint		Instances selected	1: 0	
		÷≣	Run to Cursor				
		8	Set next Statement				
_			Unforce all values of 'Device.Application'			-	
4			Display Mode		🗹 Enable breakpoint	immediately	OK Cancel

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

vice.Application.CheckBounds			
ession	Туре	Value	Prepared value
index	DINT	0	
lower	DINT	1	
upper	DINT	5	
CheckBounds	DINT	1	
6 CheckBounds 1 :=upp	per 5;		
8 ELSE			
1 END IFRETURN	=FALSE;		
	index lower upper CheckBounds 1 //Implicit build code:Here are 2 IF index 0 < Lower 3 CheckBounds 1 :=lon 5 ELSIF index 0 > upper 6 CheckBounds 1 :=upp 7 GVL.bCheckBounds_StateFILS 8 ELSE 9 CheckBounds 1 :=ind	index DINT lower DINT upper DINT CheckBounds DINT 1 //Implicit build code:Here are suggestions for 2 IF index 0 < Lower 1 THEN 3 CheckBounds 1 :=lower 1 ;; 4 FUL.bCheckBounds_State[FAUSE]:=IRUE; 5 ELSIF index 0 > upper 5 THEN 6 CheckBounds 1 :=upper 5 ;; 7 GVL.bCheckBounds_State[FAUSE]:=TRUE; 8 ELSE 9 CheckBounds 1 :=index 0 ;;	index DINT 0 lower DINT 1 upper DINT 5 CheckBounds DINT 1 //Implicit build code:Here are suggestions for code implement 1 //Implicit build code:Here are suggestions for code implement 2 IF index 0 < Lower 1 THEN 3 CheckBounds 1 :=lower 1; 4 Jif index 0 > upper 5 THEN 6 CheckBounds 1 :=upper 5; 7 GVL.bCheckBounds_State[ALSE]:=TRUE; 8 ELSE 9 CheckBounds 1 :=index 0;

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

Devices	•	₽ X	🗂 Ad	dd Device			×
Untitled 10 U		Cut	Name Acti		evice O Update device		
E O Application		Сору	Strin	g for a fulltext search	Vendor: <all vendors=""></all>		\sim
- 👘 Library Mana	₿ X	Paste Delete	Na	me ⊨-and EtherCAT ⊨-and Master	Vendor	Version	Desi ^
🖹 🎆 Task Configu	G)	Properties		EtherCAT Master	3S - Smart Software Solutions GmbH	3.5.14.0	Ether
드 🥸 MainTask 니셴 PLC		Add Object Add Folder			3S - Smart Software Solutions GmbH	3.5.14.0	Ether
		Add Device	<	H-KIII Modbus			× 1
SoftMotion General A ^I	ſ	Update Device Edit Object Edit Object With Edit IO mapping		roup by category Display all versions (frd Name: EtherCAT Master SoftWotion Vendor: 35 - Smart Software Solutions Gm Categories: Master Version: 35.14.0 Order Number: Description: EtherCAT Master SoftWotion	ьн	ersions	
	4	Import mappings from CSV Export mappings to CSV Online Config Mode Reset origin device [Device]	App Dev	end selected device as last child of			
		Simulation			Add De	vice	Close

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**

Vevice (HCQ1-1300-D)	General		Autoconf	g Master/Slave	5			Ether CAT.
PLC Logic	Sync Unit Assignmer	it	EtherCAT NIC	Setting —				
Library Manager	EtherCAT I/O Mappir	g	Destination A	ddress(MAC)	FF-FF	-FF-FF-FF-FF	Broadcast	Enable Redundancy
PLC_PRG (PRG) Task Configuration	EtherCAT IEC Object	5	Source Addre		00-04	H9F-04-E1-C5	Browse	
EtherCAT_Task	Status		O Select Ne	twork by MAC	-	Select Netwo	ork by Name	
*	Select Network Ada	pter						
LocalDevice								
SoftMotion General Axis Pool	MAC address	Name	Description					
EtherCAT_Master_SoftMotion (EtherCA	00049F04E1C3	eth0						
LocalEtherCATDevice (LocalEtherCA	00049F04E1C4	eth1						
HCQX_OD16_D (HCQX-OD16-D	00049F04E1C5	ecat1						

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		
EtherCAT_Master_SoftMotion (EtherCAT Mass LocalEtherCATDevice (LocalEtherCATDev HCQX_OD 16_D (HCQX-OD 16-D) HCQX_EC_D (HCQX-EC-D) HCQX_MD 16_D (HCQX-MD 16-D) HCQX_AD04_D (HCQX-AD04-D)	×.	Cut Copy Paste Delete Refactoring
	i	Properties Add Object
		Add Object Add Folder Add Device
		Insert Device Scan For Devices

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

Pic5.10-5

Scan Devices			- o x
Scanned Devices			
Devicename	Devicetype	Alias Address	
- LocalEtherCATDevice	LocalEtherCATDevice_test	0	
MCFA_X3E_Servo_Driver	HEFA ISE Servo Driver	18	
			Local extension
			- Show Mitterences to
Assign Address			Project
Scan Device			Copy to project Close

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

	• 4	P X Device X	Add Device					
L Untited J1 Device (HCQ1-1300-0) Device (HCQ1-1300-0) Device (HCQ1-1300-0) Device (HCQ1-1300-0) Discrete (HCQ1-1300-0)				nsert device Plug de	Vendor: Vendo 35 - Sm	<all vendors=""></all>	Version 3.5.14.0 3.5.14.0	Des Ether Ether
LocaDevice SoftMotion General Axis Poo BetherCAT_Master_SoftMotio		Update Device Edit Object Edit Object With Edit IO mapping Import mappings from CSV Export mappings to CSV	Name: EtherCAT Vendor: 35 - Smi Categories: Mag Version: 3.5.14, Order Number:	Master SoftMotion art Software Solutions Gmb ster 0	ж	nly) 🔲 Display outdated ve	ersions	20
	4	Online Config Mode Reset origin device [Device] Simulation	Append selected devic Device (You can select and		avigator w	hile this window is open.)	vice	Close

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.



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

Pic 5.11-3					
E PLC_PKG					
- & LocalDevice	X	Cut			
🚡 SoftMotion General Axis Pool		Сору			
😑 🚮 EtherCAT_Master_SoftMotion (EtherCAT)	10.52	Paste			
SV_X3T5010A_A_EC (SV-X3T5010A-A	X	Delete			
		Refactoring +			
	6	Properties			
		Add Object			10.125
	6	Add Folder	CODESYS		\times
	Ď	Insert Device Disable Device Update Device Edit Object Edit Object With	1	Please note that this driver has not been verified or tested with the device instance you are trying to link it. All or some functions may not work as expected. This can include unexpected or uncontrolled movements of the device.	
		Edit IO mapping Import mappings from CSV Export mappings to CSV		确定	
		Add SoftMotion CiA402 Axis			
		Add SoftMotionLight CiA402 Axis			
rices POUs		Simulation			

Right click to add a good SM_ Drive_ Genericdsp402, select Refactoring \rightarrow 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				
🟅 LocalDevice			-	
🗠 🍐 SoftMotion General Axis Pool	Ж	Cut	1	
EtherCAT_Master_SoftMotion	Ē	Сору	I .	
SV_X3T5010A_A_EC (SV-X	Ē.	Paste	I .	
SM_Drive_GenericDS	×	Delete		
		Refactoring +	Ţ	Rename 'SM_Drive_GenericDSP402'
	h	Properties	-	
		Add Object		

Rename and select OK in following dialogue.

factoring							- 0	>
name 'Device.EtherCAT_Master_SoftMotion.SV_X3T501	0A_A_EC.SM_Drive_GenericDSP402' to 'Axis1'.					Apply all changes.	Select affected obj	jects
1 Device	Name	Туре	Value	Variable	Address	Description		
PLCLogic	🗷 🖾 AXIS_REF: Standard							
- 🟅 LocalDevice	AXIS_REF: Scalings							
SoftMotion General Axis Pool	💌 🗀 Logical device settings							
EtherCAT_Master_SoftMotion	🗷 🗀 Standard driver settings							
- 🚮 SV_X3T5010A_A_EC	🛞 🗀 AXIS_REF: DSP402 configurati	on						
🖬 🔗 Axis1	🔹 🖾 possible cyclic driver in-/outpu	ts						
Project Settings	Address_8010	STRING	'%QW0'					
	🌻 Type_8010	STRING	'UINT'					
	AddressPointer_8010	POINTER TO BYTE	ADR(%QW0)					
	Address_8020	STRING	'%QD1'					
	• • Type_8020	STRING	'DINT'					
	AddressPointer_8020	POINTER TO BYTE	ADR(%QD1)					
	Address_8030	STRING						
	Type_8030	STRING	-					
	AddressPointer_8030	POINTER TO BYTE	0					
	Address 8040	STRING	-					
	Type_8040	STRING	-					
	AddressPointer_8040	POINTER TO BYTE	0					
	Address 8050	STRING	'%Q82'					
	• Vpe_8050	STRING	'SINT'					
	AddressPointer_8050	POINTER TO BYTE	ADR(%QB2)					
	Address_8060	STRING	'%QW4'					
	Type_8060	STRING	'UINT'					
	AddressPointer_8060	POINTER TO BYTE	ADR(%QW4)					
	Address_8070	STRING						
	Type_8070	STRING	-					
	AddressPointer 8070	POINTER TO BYTE	0					

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 \rightarrow select DC \rightarrow DC SYNC0, select DC synchronous mode.



Devices 👻 🕈 🗙	Device Device PLC_PRG	EtherCAT_Master_	SoftMotion	SV_X3T5010A_A_EC X	-
Untitled11 Device (HCQ1-1300-D)	General	Address		— Additional ————	Ether CAT.
□	Process Data	AutoIncAddress EtherCAT Address	0 ÷	Enable Expert Settings	EulerCAL.
Library Manager	Startup Parameters	✓ Distributed Clock	•		
PLC_PRG (PRG) Task Configuration	EtherCAT I/O Mapping	Select DC	DC SYNC0	~	
EtherCAT_Task	EtherCAT IEC Objects	Enable	C SYNC0 FreeRUN		
Dec_PRG	Status	Sync0: Enable Sync 0			
SoftMotion General Axis Pool	Information	Sync Unit Cycle	x 1 🛛 🗸	4000 * Cycle Time (µs)	
EtherCAT_Master_SoftMotion (EtherCAT SV_X3T5010A_A_EC (SV-X3T5010A-)		 User Defined 		0 🔹 Shift Time (μs)	
Axis1 (SM_Drive_GenericDSP40		Sync1: Enable Sync 1			
		Sync Unit Cycle	x 1 🛛 🗸	4000 🔹 Cycle Time (µs)	
		O User Defined		0 🗘 Shift Time (μs)	
	¢				>
	`				,

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

Pic 5.13-3

Untitled11	General	Scaling				
Device (HCQ1-1300-D)	General	Invert direction				
🖻 🗐 🗍 PLC Logic	Scaling/Mapping	16#20000	increments <=> i	motor turns		1
- 💮 Application	21 11 2		otor turns <=> ge	ar output turn	-	1
Library Manager	Commissioning		-			1
PLC_PRG (PRG)	SM_Drive_ETC_GenericDSP402: I/O	1 gear	output turns <=> u	units in applic	ation	1
E - 🔣 Task Configuration	Mapping					
EtherCAT_Task	SM_Drive_ETC_GenericDSP402:	Mapping				
🗏 🍪 MainTask	IEC Objects	Automatic mapping				
PLC_PRG	Status	Inputs:				
		Cyclic object	Object number	Address	Type	^
SoftMotion General Axis Pool	Information	status word (in.wStatusWord)	16#6041:16#00	'%IW1'	'UINT'	
-		actual position (diActPosition)	16#6064:16#00	'%ID2'	'DINT'	
		actual velocity (diActVelocity)	16#606C:16#00		*	
SV_X3T5010A_A_EC (SV-X3T5010A-		actual torque (wActTorque)	16#6077:16#00		· · · · ·	
Axis1 (SM_Drive_GenericDSP40		Modes of operation display (OP)		'%IB4'	'SINT'	
		dioital inputs (in dwDioitalInputs)	16#60ED+16#00	'%ID6'	'UDINT'	*
		Outputs:				
		Cyclic object	Object number	Address	Туре	^
		ControlWord (out.wControlWord)	16#6040:16#00	'%QW0'	'UINT'	
		set position (diSetPosition)	16#607A:16#00	'%QD1'	'DINT'	
		set velocity (diSetVelocity)	16#60FF:16#00			_
		set torque (wSetTorque)	16#6071:16#00			
		Modes of operation (OP)	16#6060:16#00	'%QB2'	'SINT'	
		Touch Probe Euroction	16#60B8+16#00	'%OW4'	'LIINT'	· · · · · · · · · · · · · · · · · · ·

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 <u>4.3</u>. Write command as below:

Pic 5.14-1



1	fbPowerl (
2	Axis:=axis1,	
з	Enable:=TRUE,	
4	bRegulatorOn:=bpoweron,	
5	bDriveStart:=TRUE,);	
6	fbJogl (Main program
7	Axis:=axis1,	
8	JogForward:=bJogfw,	area
9	JogBackward:=bJogbw,	
10	Velocity:=Velocity,	
11	Acceleration:=Accelerat	ion ,
12	Deceleration:=Decelerat	ion ,
13	Jerk:= ,);	

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

Pic 5.14-2

Messages - Total 0 error(s), 0 warning(s), 1 message(s)							
Devices	-	🕴 0 error(s)	① warning(s)	 1 message(s) 	\times	⋇	
Description	Project	C	bject	Position			
Oreated task 'EtherCAT_Task'.							

Related parameters of EtherCAT Task need manual configuration, or loss error will be reported during the operation of subsequent shafts. Double click Application \rightarrow Task Configuration \rightarrow EtherCAT Task, call PLC_Motion in the task, steps refer to <u>4.2</u>, and set priority as "0".



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

😑 📀 🗊 Device [connected] (HCQ1-1300-D)	- O M Device [connected] (HCQ1-1300-D)	E 🚱 🛐 Device [connected] (HCQ1-1300-D)
유 회 PLC Logic 유 🕐 Application [run]	🖶 🗐 PLC Logic	PLC Logic Application [run]
Application (tin) Application (tin) Device (the construction of the construction (the construction of the constr	 Charles Manager Charles Manager Charles Manager Charles Configuration Charles	Library Manager PLC PRG (PRG) PLC PRG (PRG) WanTask PLC PRG ManTask PLC PRG ManTask Mormal operation of ManTask ManT

Enable the axis first, set bpoweron as TRUE.

	PLC_Motion	×		Start (15)				
De	evice.Applicatio	n.PLC_Motion						
xpr	ression	Туре	Value	Prepared value	Address	Comment	^	
1	fbPower1	MC_Power				axis1 Enable function block		
1	🖗 fbJog1	MC_Jog				axis1Jog function block		
1	bpoweron	BOOL	TRUE			Enable trigger		
	🖗 bJogfw	BOOL	FALSE			Positivejog	1	
,	< blogbw	BOOL	FALSE			Reverse jog		
-	Velocity	LREAL	20			Jog speed, default value is 0		
-	Acceleration	LREAL	50			Jog acceleration, default value is		1
1	Deceleration	IRFAI	50			Jog deceleration.default value i	~	
-		is:=axisl,				200000000000000000000000000000000000000		
	2 Ax: 3 End 4 bRe 5 bD 6 fbJog1 7 Ax:	is:=axisl, able TRUE :=TRUE egulatorOn TRUE riveStart TRUE (is:=axisl,	:=bpoweron TRU :=TRUE,);	E,				
	2 Ax: 3 End 4 bRe 5 bDD 6 fbJog1 7 Ax: 8 Jog	is:=axisl, able TRUE :=TRUI egulatorOn TRUE riveStart TRUE (:=bpoweron TRU :=TRUE,); =bJogfw FALSE,	. .				
	2 Ax: 3 End 4 bRe 5 bDD 6 fbJogl 7 Ax: 8 Jog 9 Jog	is:=axisl, able TRUE :=TRUI egulatorOn TRUE riveStart TRUE (is:=axisl, gForward FALSE ::	:=bpoweron TRU :=TRUE,); =bJogfw FALSE,	E, 20,				
	2 Ax: 3 End 4 bR 5 bD 6 fbJog1 7 Ax: 8 Jog 9 Jog 10 Ve: 11 Acc	is:=axis1, able TRUE :=TRUE egulatorOn TRUE riveStart TRUE (is:=axis1, gForwardFALSE :: gForwardFALSE locity 20 celeration	:=bpoweron TRU :=TRUE,); =bJogfw FALSE, :=bJogbw FALSE,	20 , ration 50 ,				

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: <u>https://help.CODESYS.com/</u>

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



未识别的网络 Realtek PCIe GbE Family Contr...

県 以太网 Statu	🏺 以太网 Properties	Internet 协议版本 4 (TCP/IPv4) Properties X
General	Networking Sharing	General
Connection IPv4 Connectiv IPv6 Connectiv Media State: Duration: Speed: Details Activity	Connect using: Reatek PCIe GbE Family Controller This connection uses the following items: Microsoft 网络客户端 Microsoft 网络的文件和打印机共 OS 数据包计划程序 Microsoft 网络适配器多路传送器 Microsoft 网络适配器多路传送器 Microsoft ULDP 协议驱动程序 Microsoft LLDP 协议驱动程序 Microsoft LLDP 协议驱动程序	Subnet mask: 255 . 255 . 255 . 0
Bytes:	Install Description 传输控制协议/Internet 协议。该协议是 协议,用于在不同的相互连接的网络」	Preferred DNS server: Alternate DNS server: Validate settings upon exit Advanced OK Cancel

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
-------	-----

Untitled11 Device (HCQ1-1300-D)	Communication Settings Scan network Gateway v Device v	
	Select Device Select the network path to the controller: Select the network path to the controller: Control of the controller: Device Name: HCQ1-1300D [0064] Device Address: 0064 Block driver: UDP Number of channels: 4 Serial number:	X Scan network Wink
	16C7 0002 V	Cancel

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 06 04 04 00 00 00 01							
	Example	Length	Instruction	Remark			
	0x97	1	transaction ID Hi	Client initiated, server replicated, for transaction			
	0x76	1	transaction ID Lo	pairing			
Map message head	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			
	0x0000	2	Starting address				
Data	0x 0001	2	number of registers				
Check	Check						

Response message:

97 76 00 00 05 04 04 02 12 34							
	Example	Length	Instruction	Remark			
0x97 1 transaction ID Hi		Client initiated, server replicated, for transaction					
	0x76	1	transaction ID Lo	pairing			
Map message head	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 of 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 popup 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.

Devices - 7 ×	Ethernet	×			
Untitled11 Device [connected] (HCQ1-1300-D)	General		Interface:		
PLC Logic	Status		IPAddress	192 . 168 . 0 . 1	
Library Manager	Ethernet Device I	I/O Mapping	Subnet Mask	255 . 255 . 255 . 0	
PLC_Motion (PRG) DLC_PRG (PRG)	Ethernet Device J		Default Gateway	0.0.0.	
=-∰ Task Configuration =-⊗ EtherCAT_Task	Network Adapt	ers			×
PLC_Motion			ldress		1
	10 eth0	127.0	. 0. 1 68. 188. 100		-
LocalDevice	eth1	192.1	68.88.100		
SoftMotion General Axis Pool EtherCAT Master SoftMotion (EtherCAT	ecat1	0.0.0	. 0		ł
Ethernet (Ethernet)	IP Address	192 . 168 .	88 . 100		
	Subnet Mask Default Gatews	255 . 255 . 2 av 192 . 168 .			
	MAC Address	00:04:9F:04:			
				OK Cancel	

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 \rightarrow 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**

Ethernet Modbus_TC	P_Slave X	
General	Modbus-TCP	
Modbus Slave Channel	Slave IP Address:	192 . 168 . 188 . 88
Modbus Slave Init	Response Timeout (ms):	1000
ModbusTCPSlave Parameters	Port:	502
ModbusTCPSlave IEC Objects		
Status		
Information		
	General Modbus Slave Channel Modbus Slave Init Modbus TCPSlave Parameters Modbus TCPSlave IEC Objects Status Information	General Modbus Slave Channel Modbus Slave Init Modbus TCPSlave Parameters Modbus TCPSlave IEC Objects Status Information

Click on Modbus slave channel→Add Channel

Ethernet X Modbus_TCP_	Slave X								
themet [Device]	Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length	Comment
Modbus Slave Channel									
Modbus Slave Init									
ModbusTCPSlave Parameters									
ModbusTCPSlave IEC Objects									
Status									
Information									
							,		
						Ad	d Channel	Delete	Edit

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.

ModbusChannel Channel Name Access Type	Channel 0	_	×		
Channel Name	-	_	×		
Channel Name	-	-	×		
Name	Channel 0	_			
	Channel 0				
	Read Holding Registers (Function Code 3)	~		
Trigger	Cydic	✓ Cycle Time (ms) 10	00		
Comment					
READ Register	1	7			
			~		
Error Handling	Keep last Value	~			
WRITE Register					
Offset	0×0000		~		
Length	1				
		ОК	Cancel		
L					
	READ Register Offset Length Error Handling WRITE Register Offset	Comment READ Register Offset Length Error Handling WRITE Register Offset Ox0000 Comment Ox0000 Comment Ox0000 Comment Co	Comment READ Register Offset Length Error Handling Keep last Value VRITE Register Offset Ox0000 Length 1	Comment READ Register Offset Dx0000 Length SI Error Handling Keep last Value WRITE Register Offset 0x0000 Length 1	Comment READ Register Offset Dx0000 Length S Error Handling Keep last Value WRITE Register Offset Dx0000 Length 1

ltem	Туре	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 ModbusTCPSIave configuration \rightarrow modify the unit ID (slave site address) to 1.

Pic6.2-9

General	Parameter	Туре	Value	Default Value	Unit	Description
	🖤 🖗 NewChannelConfig	BOOL	true	true		Use the new Channel-Config format
Modbus Slave Channel	🖗 Unit-ID	USINT	1	16#FF		Unit-ID of the Device
Modbus Slave Init	ResponseTimeout	DWORD	1000	1000		Maximum time for a Slave to respond in ms
	IPAddress	ARRAY[03] OF BYTE	[192, 168, 188,	[192, 168, 0, 1]		Configure IP Address of TCP SLave.
ModbusTCPSlave Parameters	🖗 Port	UINT	502	502		Port where the slave is listening
10dbusTCPSlave I/O Mapping	🐵 🖗 Channel 0					ChannelConfig
Modbus ICPSiave I/O Mapping	ConfigVersion	UDINT	16#03050B00	16#03050B00		
ModbusTCPSlave IEC Objects						
Status						
Information						
Information						
Information						
Information						
Information						
information						
information						
information						
nformation						
Information						
nformation						
nformation						
information						
information						
Information						

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



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

Pic6.2-11

Untitled11	General	Find	Filter Show a	1	• 1	🗣 Add FE	3 for IO channel	
Device (HCQ1-1300-D) Device (HCQ1-1300-D)	Modbus Slave Channel Modbus Slave Init	Variable Input Assistant	Mapping Channel Channel 0	Address %IW14	Type ARRAY [04] OF WORD		Description Read Holding Registers	
Library Manager PLC_Motion (PRG)	ModbusTCPSlave Parameters	Text search Categories						
PLC_PRG (PRG) Task Configuration	ModbusTCPSlave I/O Mapping	Variables	 Name 	Тур	e Address	5	Origin	
BetherCAT_Task Dec_Motion BetherCAT_Task Dec_Motion Dec_Motion Dec_PRG Dec_PRG LocaDevice	ModbusTCPSlave IEC Objects Status Information		Continue Continu	PROGA VAR GL ARRAY [15]	IAM DAM OF WORD			
SoffWoton General Avs Pool SoffWoton General Avs Pool General Avs Pool General EthercAT, Master SoffWoton (EtherCAT General Ethernet) Modbus_TCP_Master (Modbus TCP M Modbus_TCP_Slave (Modbus TC)			 IoConfig_Glob IoDrvEthercat IoDrvEthercat IoDrvEthernet ISM3_Basic ISM3_Math 	_	ry ry ry	10 5	00rvEtherCAT, 3.5 oDrvEthernet, 3.5.1 M3_Basic, 4.4.0.1 (M3_Math, 4.4.0.0 (

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

C	COM1	COM2	CC	SW3	0	OM4	Remote HMI
R	emote PLC	Ethernet	PLC	Serv	ervice Printer		Keyboard
ID	Device Alias	IP	1	Port	Device	Туре	Default Station
e E	thernet PLC(Or S	Service by Re	mote HN	VII)			
Manufacturer: Modbus Compatible 🔹							
	Device Type	e: Modbus_T	СР				T
	Device Alia	s: Alias canno	ot be nul	ll, Default	Device	3	
De	efault Station No	o.: Constant	•		1	Synchronize	e Station No.
E	Broadcast Statio	n:					
	Communication Setting The IP address of Ethernet PLC Fixed 192.168.88.200 Port No.: Constant 502 Keset Advance Instructions						
			/ance				
		structions			nenetw		DK Cancel
.oca		structions	rearreor		ough IP	address and	port directly.
.oca	Ins	structions	rearreor			address and	port directly.

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 <u>3.2</u>.

Pic6.2-13

Package Manager							×
Currently installed packages:							
Refresh			Sort by:	Name	\sim	Install	
Name	Version	Installation date	Update in	fo		Uninstall	
CODESYS SoftMotion	4.4.0.2	2022/4/2	Free version	4.10.0.0 available!		Details	
∰ HCQ1-1300-D-V300-Test版本扩展包	0.0.0.1	2022/4/19			1		
						Updates	
						Search updates	
						Download	
						CODESYS Store	
						Rating	
						CODESYS Store	
<					>		
Display versions 🗹 Search updates in ba	ckground					Close	

Shown as below after installation,

Pic6.2-14

Library Manager 🗙	
🔁 Add library 🗙 Delete library 🛛 😁 Properties 📷 Details 🛛 🚐 Placehold	ders 👔 Library repository
Name	Namespace
3SLicense = 3SLicense, 3.5.14.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE
🗄 📲 BreakpointLogging = Breakpoint Logging Functions, 3,5,5,0 (3S - Smart Software	Solutions GmbH) BPLog
HCFAModbusLib = HCFAModbusLib, 1.1.0.0 (HCFA)	HCFAModbusLib
E → 100 IoStandard = IoStandard, 3.5.13.0 (System)	IoStandard
🗈 📲 IoStandard = IoStandard, 3.5.13.0 (System)	IoStandard
SM3_Basic = SM3_Basic, 4.4.0.1 (3S - Smart Software Solutions GmbH)	SM3_Basic
SM3_CNC = SM3_CNC_4_4_0_0 (3S - Smart Software Solutions GmbH)	SM3_CNC

Modbus TCP FB and instruction as below:

	MODBUSTCP_DeviceDiag					
_	bRestart	BOOL	BO	OL bDone	-	
			BO	OL bErro	-	
			MODBUS_Erro	or wErrID	-	

Name	Туре	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 \rightarrow 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

	COM1	COM2	CC	M3	0	COM4	Remote HMI	
Re	emote PLC	Ethernet	PLC	Serv	/ice	Printer	Keyboard	
D	Device Alias	IP	F	ort	Device	Туре	Default Station	
P E	thernet PLC(Or	Service by Re	mote HN	11)				
	Manufactur	er: Modbus C	ompatib	le			•	
	Device Typ	pe: Modbus_1	ГСР				•	
	Device Ali	as: Alias cann	ot be nul	, Default	:Device	3		
De	efault Station N	o.: Constant	•		1	Synchronize	e Station No.	
e	Broadcast Static	on:						
	Communicatio	n Setting —			Compati	ble Model —		
	The IP address	of Ethernet P	LC		Aodbus ria TCP	Compatible E	xternal Device	
	Fixed •	192.168.88	3.200			Online Simula	tion	
	Port No.:		•					
	Constant 🔻		502 😴	, I				
	Reset	Adv	/ance					
	In	structions						
							OK Cance	
	I HMI can acces	s Ethernet PL	C register	data th	rough IP	address and	port directly.	
oca								

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

	MODBUSRTU_MasterDiag	
-bStop BOOL	BOOL bAllSlavesOk	⊢
-bReset BOOL	WORD wOfflineSlavesNum	⊢
	BOOL bDone	⊢
	BOOL bError	L
	MODBUS Error wErrID	L

Name	Туре	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

		M Add Device	×
		Name: ModbusRtuMaster Action: Action: Action: Action: Action: Act]
Devices		String for a fulltext search Vendor: <ali vendors=""> Name Vendor Version Description Image: Image</ali>	>
Application PLC_Logic Application Drary Mar PLC_RG (Same Config Same Config	Paste Delete		
LocalDevice SoftMotion General ModbusTcpDevice (Add Folder Add Device Update Device	Categories: Version: 1.0.0.2 Order Number: - Description:	
	Edit Object With Edit IO mapping Import mappings from CSV Export mappings to CSV	Append selected device as last child of Device ① (You can select another target node in the navigator while this window is open.) Add Device	

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

Pic6.3-3

- ModbusTcpDevice	1.1	TcoDexice)	Add Device × Name: ModbusktuSlave Action: Action: Action: Plug device O Update device
- 🗂 ModbusRtuMaster	≪ ×	Copy Paste Delete	String for a fulltext search Vendor: cAll vendors> Name Vendor Version Biscellaneous Image: ModbusRtuSlave HCFA
		Refactoring Properties Add Object Add Folder	
	ſ	Add Device Insert Device Disable Device Update Device Edit Object Edit Object With	Croup by category Display all versions (for experts only) Display outdated versions Name: ModbusRtuSlave Vendor: HCFA Categories: Version: 10.0.2 Order Number: - Description:
vices POUs		Edit IO mapping Import mappings from CSV Export mappings to CSV Simulation	Append selected device as last child of HodbusRtuMaster Image: Control of the select another target node in the navigator while this window is open.) Image: Control of the select another target node in the navigator while this window is open.) Image: Control of the select another target node in the navigator while this window is open.)

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



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

Pic6.3-5

```
ModbusRtuMaster (ModbusRtuMaster)

ModbusRtuSlave (ModbusRtuSlave)

ModbusSlaveReadChannel (ModbusSlaveReadChannel)

ModbusSlaveWriteChannel (ModbusSlaveWriteChannel)
```

Click Added ModbusRtuMaster, select ModbusRtuMaster configuration.

Pic6.3-6

4odbusRtuMasterParameters	Parameter	Туре	Value	Default Value	Unit	Description
1odbusRtuMaster I/O Mapping	COMID	Enumeration of BYTE	1	1		Number of communication com port
foodbusktumaster i/omapping	🔷 🖗 Baudrate	Enumeration of DWORD	9600	9600		Baudrate of the serial port
4odbusRtuMaster IEC Objects	🛷 Databit	Enumeration of BYTE	8	8		Databit of the serial port
	Stopbit	Enumeration of BYTE	1	1		Stopbit of the serial port
Status	🖤 < Parity	Enumeration of STRING	EVEN	EVEN		Parity for messages on the serial port.
information	FrameSpace	WORD	0	0	ms	time interval between frame transmissions

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

Parameter	Туре				
		Value	Default Value	Unit	Description
🖤 🖗 SlaveID	BYTE(0247)	1	1		The address of slave,set to 0 for broadcast
🖙 🖗 ResponseTimeout	WORD(265535)	100	100	ms	Slave response timeout
	ResponseTimeout	ResponseTimeout WORD(265535)	ResponseTimeout WORD(265533) 100	ResponseTimeout WORD(265535) 100 100	ResponseTimeout WORD(265535) 100 100 ms

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

10dbusReadChannelParameters	Parameter	Туре	Value	Default Value	Unit	Description
1odbusReadChannel I/O Mapping	FunctionCode	Enumeration of BYTE	Read Coils(1)	Read Coils(1)		Function code of modbus slave read channel
To obuskead channel 1/0 Happing	🗝 < addr	WORD	0	0		Read address of the function code
Status	🔷 < length	WORD(1160)	1	1		Read length of the function code
	🔷 🖗 Trigger	Enumeration of BOOL	Cyde	Cyde		Trigger the communication
Information	🖤 < CycleTime	WORD(1065535)	100	100	ms	only used when the cyclical trigger
	RetryTimes	BYTE	1	1		number of retries when communication failed

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

odbusWriteChannel Parameters	Parameter	Туре	Value	Default Value	Unit	Description
odbusWriteChannel I/O Mapping	FunctionCode	Enumeration of BYTE	Write Single C	Write Single C		Function code of modbus slave write channel
odbuswriteChannel I/O Mapping	🖉 🛷 addr	WORD	0	0		Write address of the function code
tatus	🖉 🖉 length	WORD(1160)	1	1		Write length of the function code
	🔷 🖗 Trigger	Enumeration of BOOL	Cycle	Cyde		Trigger the communication
formation	CycleTime	WORD(1065535)	100	100	ms	only used when the cyclical trigger
	RetryTimes	BYTE	1	1		number of retries when communication failed

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**

Remote PLC	Ethernet P	PLC	Service	Printer	Keyboard
COM1	COM2	СОМЗ		COM4	Remote HMI
Unused	© Co	onnect Devic	e(Master)	Provide	Service(Slave)
Device Ty	pe: Modbus_RT	U_Server			•
Device Al	ias: Service				
Server Station N	Io.: Constant 🔻				1 🛓
Enable	Address Mapping	g			
Communication	Setting			ble Model RTU Server	
Communicatio Baud Rate: Data Bit:	n Type: RS485-2 115200 8	•		Online Simulatio Iddr map B(n-1) B(n-1)	n
Stop Bit: Parity Bit:	1 None	•	3X_m>l 4X_m>l m:1,2,3,	.W(m-1) .W(m-1)	
Reset	Adva	nce	4X_k>R	W(k-10000) W(k-10000) 0001,10002,,6	5535
Inst	ructions				

6.3.3 Modbus RTU Slave Application

Modbus RTU slave FB and instruction as below: Pic6.3-11

	MODBUSRTU_DeviceDiag		
_	bRestart BOOL BOOL	bDone	┝
	BOOL	bError	H
	MODBUS_Error	wErrID	⊢

Name	Туре	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.





Open added ModbusRtuDevice, select ModbusRtuDevice configuration.

ModbusRtuDeviceParameters	Parameter	Туре	Value	Default Value	Unit	Description
ModbusRtuDevice I/O Mapping	COMID	Enumeration of BYTE	1	1		The comport used for communication
Hodbusktubevice () o Happing	🛷 Baudrate	Enumeration of DWORD	9600	9600		Baudrate of the serial port
ModbusRtuDevice IEC Objects	🛷 Databit	Enumeration of BYTE	8	8		Databit of the serial port
	Stopbit	Enumeration of BYTE	1	1		Stopbit of the serial port
Status	🔷 🖗 Parity	Enumeration of STRING	EVEN	EVEN		Parity for messages on the serial port
Information	FrameTimeout	WORD(1999)	100	100	ms	Timeout of the frame transmissions
	SlaveID	BYTE(1247)	1	1		Address of the modbus rtu device

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.

Creation of Simple PLC Projec

Communication Co	nnection			? >
Remote PLC	Ethernet PLC	Servic	e Printer	Keyboard
COM1	COM2	COM3	COM4	Remote HMI
Our Unused	Connect	t Device(Mas	ter) 💿 Provi	de Service(Slave)
Manufacture	: Modbus Compatil	ble		-
Device Type	: Modbus RTU			•
Device Alias	: Device			
Pre-set Station No	: Constant 🔻	0	Synchroni	ze Station No.
Broadcast Station	n:	Master 9	Station No.:	1 🔹
Communication Se Communication Baud Rate: Data Bit: Stop Bit: Parity Bit: Reset		Mode	oatible Model ous Compatible Ex ort Online Simulat	
Instru	ctions	,		
Help				OK Cancel

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



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 <u>5.12</u>.

用我们的工作

创造美好的生活

