Q-Series PACs Training

To create a better life through our work

CONTENTS

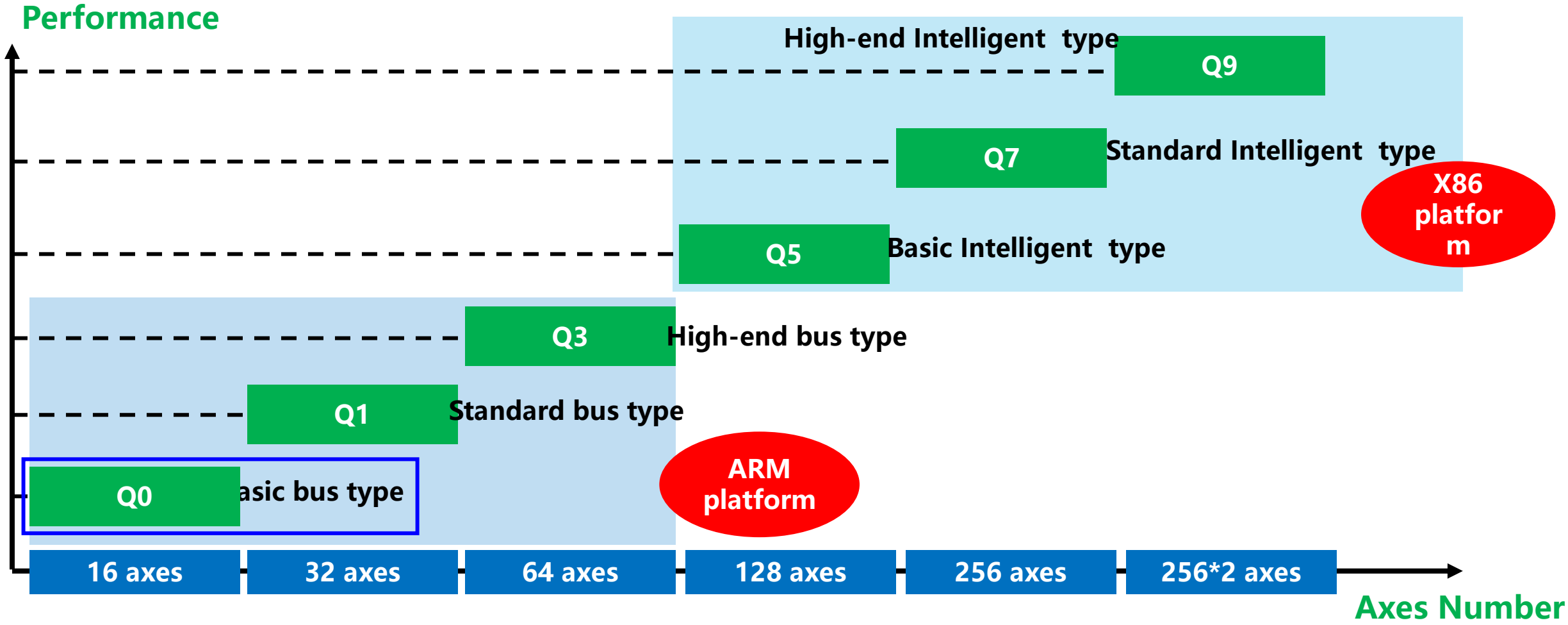
PART 01

Product Introduction

Fast processing of instructions
Quick Reaction
Rich bus Interfaces
multiple axes supported
Optional Extension
Unique Design

Q Series Family Product

Focus on high-end industrial automation to create a total solution



Remark: The actual number of axes is related to the scan cycle

Product Planing for Q-series

Q0	Q1	Q3	Q5	Q7	Q9
1024 points 16 axes	2048 points 32 axes	4095 points 64 axes	8192 points 128 axes	16384 points 256 axes	32768 points 256*2 axes
A7*2 650MHz 1G DDR+2G EMMC	A7*2 1.2GHz 1G DDR+4G EMMC	A53*2 1.3GHz 2G DDR+8G EMMC	X86-CE*4 2GHz 4G DDR+64G SSD	I5*2 4GHz 4G DDR+128G SSD	I7*4 4.5GHz 4G DDR+128G SSD

ARM Platform

x86 Platform

HIO*32 | CAN | IoT | RS485*2 | *Optional

USB*2 | SD | GigE*3 | BackplaneBUS | PCIE | HDMI

TSN | EtherNET/IP | OPC UA | EtherCAT | CANOPEN | MODBUS | MQTT

HWTOOLS | CODESYS | HCPACS | EDGE-COMPUTE | FA-IT

Logic Controll | Process Control | Motion Control | Robot Control | CNC Control

RTOS LINUX/WINDOWS RTE

Q-series Product List

Q-series products	Models	Function description	Recommended combination
CPU main unit	HCQ0-1 00-D	CPU unit，support RS485/RS232/CANOpen/EtherCAT	Can connect with Q-series extension modules, HCFA's EtherCAT/CanOpen servo drives and motors
CPU main unit	HCQ1-1 00-D	CPU unit，with high-speed I/O，support RS485/RS232/CANOpen/EtherCAT	Can connect with Q-series extension modules, EtherCAT/CanOpen servo drives and motors
CPU main unit	HCQ5-1 00-D	CPU unit，with high-speed I/O，support RS485/RS232/EtherCAT	Can connect with Q-series extension modules, HCFA's EtherCAT servo drives and motors
Coupler module	HCQX-EC01-D	Q-BUS interface，For connecting EtherCAT to Q series modules(HCQX-xxxx-D)coupler, converts transmission messages from EtherNet to E-bus signals	Can connect with Q-series extension module and host controller with EtherCAT protocol
Digital input module	HCQX-ID16-D	16-ch digital input module（support PNP/NPN）	Can connect with Q-series coupler and other extension modules
Digital output module	HCQX-OD16-D	6-ch digital output module	Can connect with Q-series coupler and other extension modules
Analog input module	HCQX-AD04-D	4-ch analog input module（support PNP/NPN）	Can connect with Q-series coupler and other extension modules
Analog output module	HCQX-DA04-D	4-ch analog output module	Can connect with Q-series coupler and other extension modules
Digital I/O module	HCQX-MD16-D	8-ch digital input+8-ch digital output	Can connect with Q-series coupler and other extension modules
Temperature measurement module	HCQX-TS04-D	4-ch temperature measurement module	Can connect with Q-series coupler and other extension modules
DC power module	HCQX-PW01-D	DC power supply module	To be released
AC power module	HCQX-PD01-A	AC power supply module	On the left-side of Q5
End unit	----------	Attached to the end side of the CPU unit or extension module, dustproof	Has been released

Naming Rule for Q-series PACs

CPU unit

HC Q1 X -1 3 0 0 - D - X X X X

Product name

SYM	Type
HC	HCFA controller

Product series

SYM	Type
Q0	Basic bus motion controller
Q1	Standard bus motion controller
Q3	High-end bus motion controller
Q5	Basic intelligent controller
Q7	Standard intelligent controller
Q9	High-end intelligent controller

Model name

SYM	Type
N/A	Standard
S	Basic type
J	Module type

Operating system

SYM	Type
1	Linux
2	Windows 10
3	Windows 7
4	QNX

Number of axes

SYM	Type
n(0-8)	2^(n+2)

Control software module

SYM	Type
0	CODESYS
1	HCPACS
2	ROBOT
3	CNC
4	MC
9	N/A

Additional software module

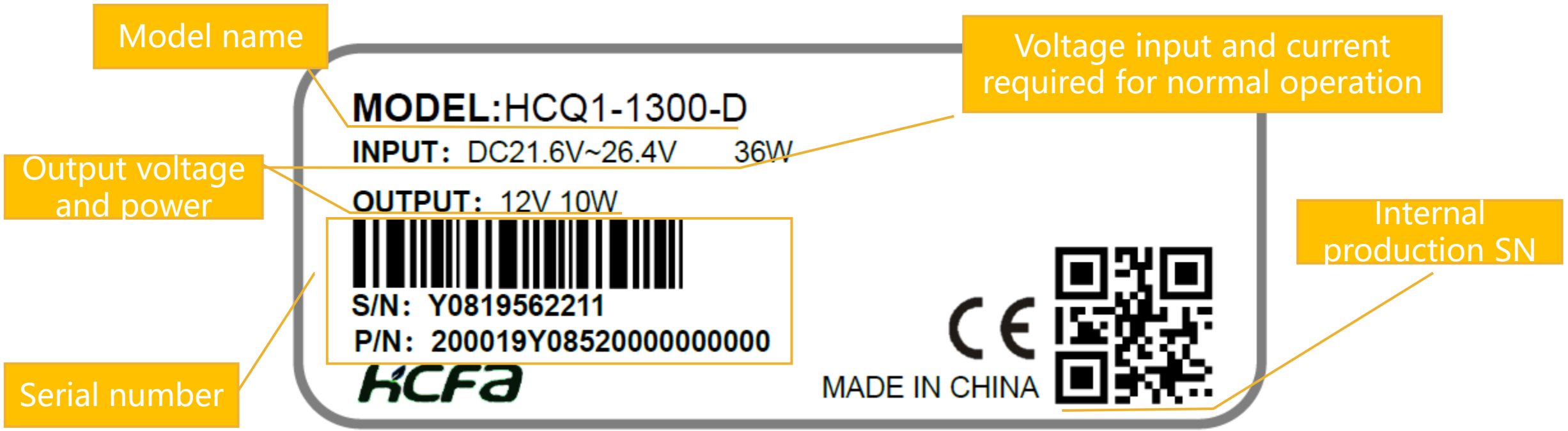
SYM	Type
0	Standard
1	Machine vision
2	Edge computing

Power type

SYM	Type
D	DC power
A	AC power

Control No. and customization

Nameplate description for CPU units

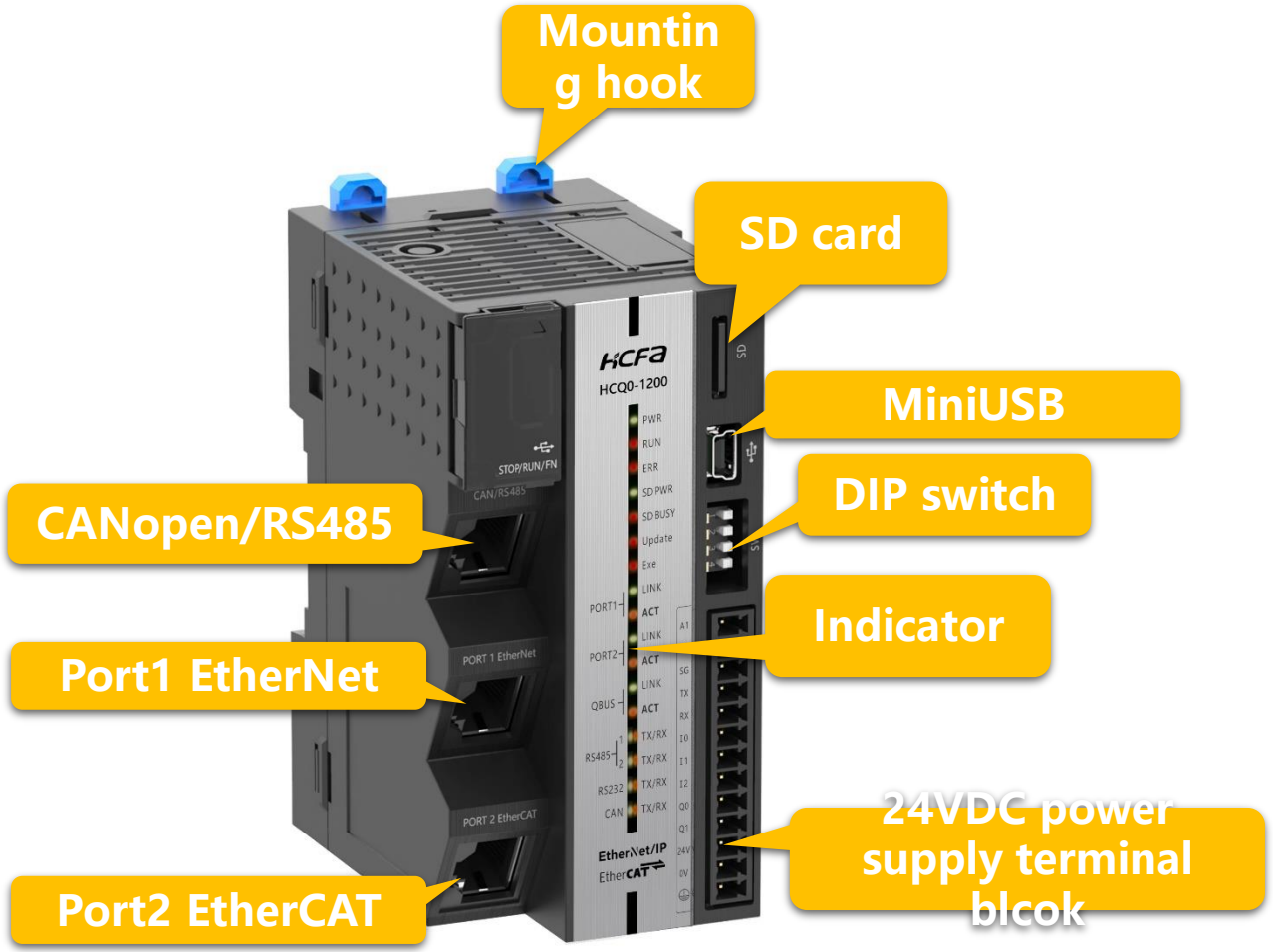


HCQ0

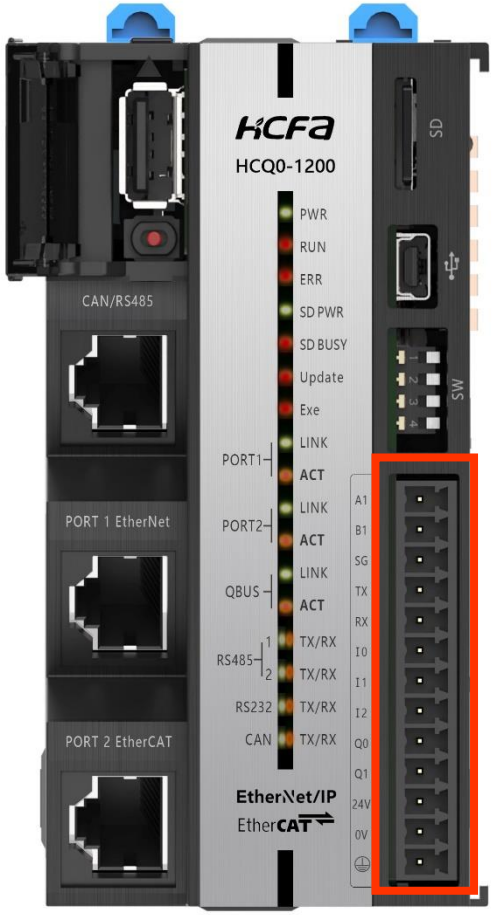
Basic Bus Controller

To create a better life through our work

Appearance & Interface for Q0 –series PAC

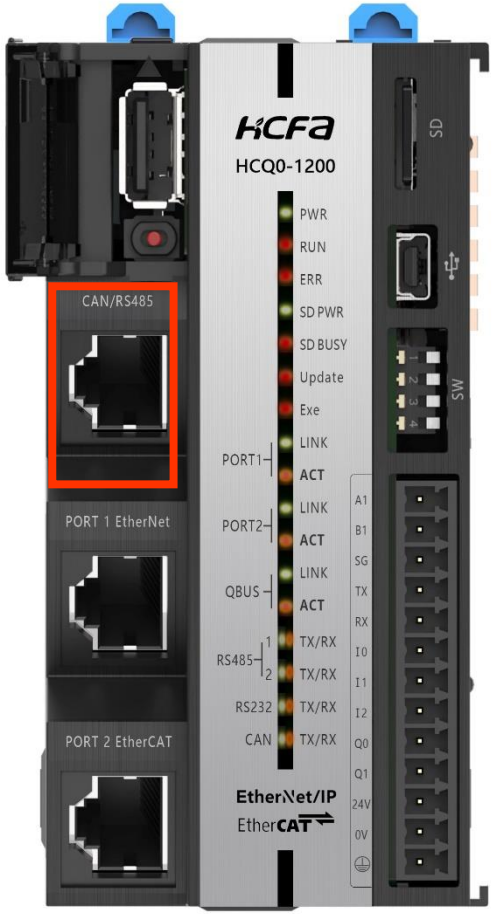


Appearance & Interface for Q0 –series PAC



Item	Name	Function
1	A1	RS485-A
2	B1	RS485-B
3	GND	GND for RS485 & 232
4	TX	RS232-Send
5	RX	RS232-Receive
6	I0	Input point 0，only support sink input
7	I1	Input point 1，only support sink input
8	I2	Input point 2，only support sink input
9	Q0	Output point 0，only support NPN output
10	Q1	Output point 1，only support NPN output
11	24V	DC power supply 24V input
12	0V	DC power supply 0V，IO terminal COM
13	FG	Gounding

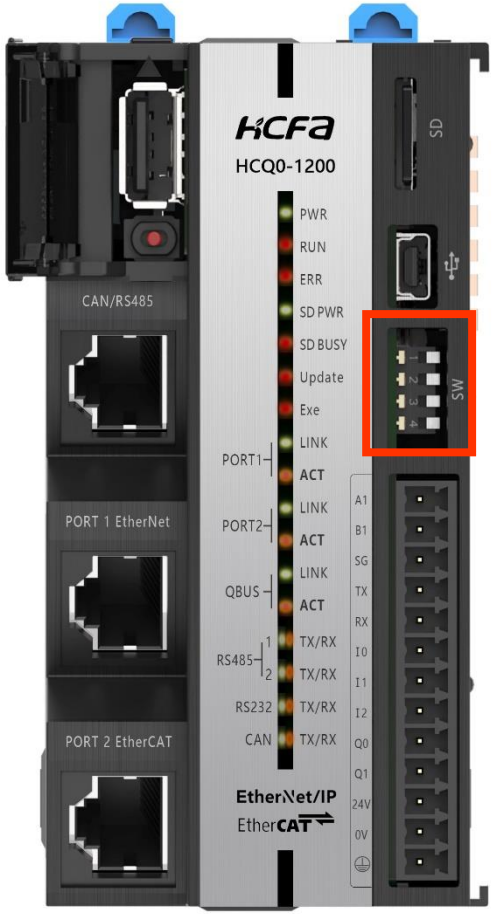
Appearance & Interface for Q0 –series PAC



Item	Description
1	CAN-H
2	CAN-L
3	Common grounding for RS485 master & CAN
4	RS485 master -A
5	RS485 master -B
6	n/c
7	n/c
8	n/c

- **RS485**: Corresponding to COM2 in the program, the port has built-in 120Ω terminal resistance, supports Modbus RTU master station, not support slave station, otherwise error occur and a red triangle displayed on the device
- **CAN**: Built-in 120Ω terminal resistance, supports CANopen master station, not support slave station, otherwise error occur and a red triangle displayed on the device

Appearance & Interface for Q0 –series PAC



SW2	SW1	Function
0	0	Long-press fn to remove U-disk /SD card
0	1	Long-press fn to reset IP address, restart after completion
1	0	Long-press fn to import PLC program, restart after completion
1	1	Reserved

- **4 DIP switches(4 digits)：** SW4: RS485 terminal resistance switch, SW3: reserved and not used, and the combination of SW1 and SW2 defines different DIP functions
- **After the DIP switches setting completed, long press the fn key for more than 2s to trigger the action to prevent accidental touch**

Technical parameters for Q0-series PAC

Item			Parameter
Programming	Total program capacity		16MBytes
	I-zone（%I）		128KBytes
	Q-zone（%Q）		128KBytes
	M-zone（%M）		512KBytes
	Power-failure holding zone		800KBytes
	Other variables		Undefined
Unit configuration	Number of extension modules	Digital module	Calculated from current consumption
		Analog module	
		External power supply	12V/16W
EtherCAT	Communication standard		IEC 61158 Type12
	EtherCAT master spec.		Class B (Compatible with Feature Pack Motion Control)
	Physical layer		100BASE-TX
	Modulation		Baseband
	Transmission speed		100Mbps (100Base-TX)
	Duplex mode		Full duplex
	Topology		Line, Bus and Star-type
	Transmission medium		Category 5 or higher twisted pair cable (aluminum foil + braided double shield cable)
	Max. transmission distance between nodes		100m
	Max. process data		Input: 5,736 bytes Output: 5,736 bytes (But the max. number of frames of process data is 4.)
	Communication cycle		Minimum 1ms

Technical parameters for Q0-series PAC

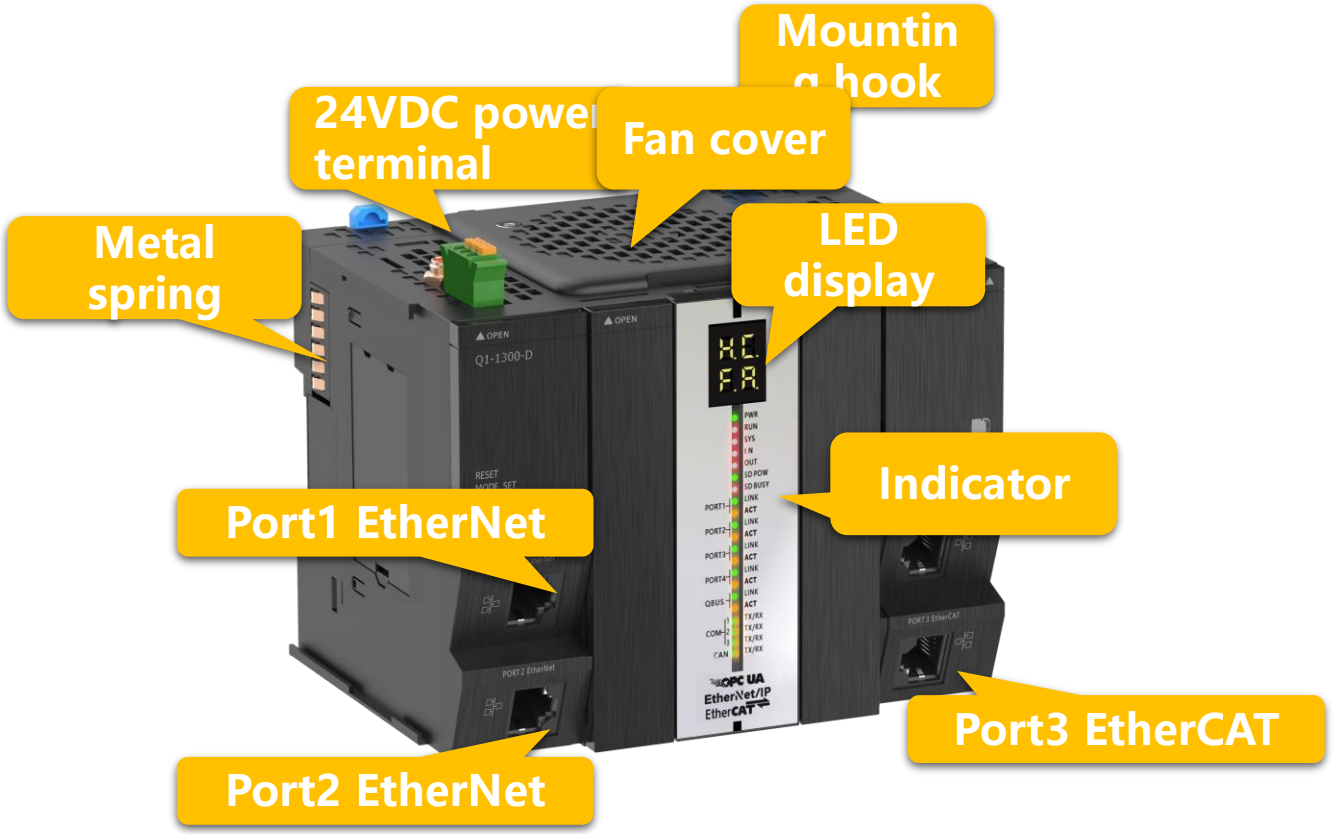
Item			Parameter
CNAOPEN master	Link layer		CAN2.0A
	Terminating resistor		Built-in 120Ω，disconnection is not supported
	Support baud rate bps		50K,100K,125K,250K,500K,800K和1M
	Transmission medium		Category 5 or higher twisted pair cable
	Max. communication distance		2500m（At 20kKbit/s）
	Max. number of slaves		32
	Communication cycle		Mini. 1ms
Serial port	Physical layer	COM1	RS485
		COM2	RS485, only support master
		COM3	RS232
	Terminating resistor	COM1	Support 120Ω, support DIP switching
		COM2	Built-in 120Ω，disconnection is not supported
	Baud rate bps		50K,100K,125K,250K,500K,800K和1M
	Max. communication distance	COM1,COM2	500m
		COM3	15m
	Topology	COM1,COM2	Line, Bus and Star-type
		COM3	Point to point
	Max. number of slaves	COM1,COM2	32
		COM3	1
	Transmission medium		Category 5 or higher twisted pair cable

HCQ1

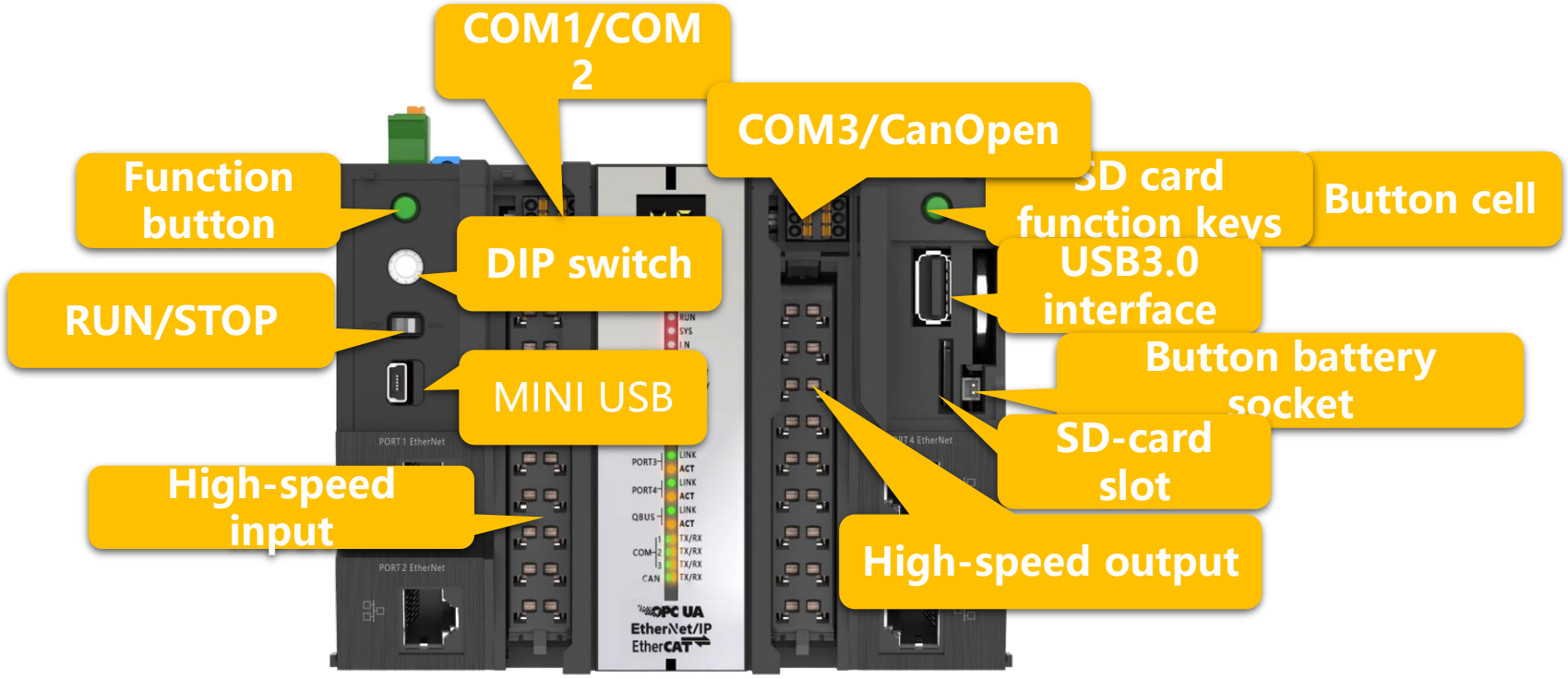
Standard Bus Controller

To create a better life through our work

Appearance & Interface for Q1 –series PAC



Appearance & Interface for Q1 –series PAC



COM1(RS485)	Modbus RTU (master/slave supported) Free Protocol
COM2(RS485)	
COM3(RS232)	
CAN	CANopen communication

Technical parameters for Q1-series PAC

Item			Parameters
Programming	Total program capacity		16MBytes
	I-zone（%I）		128KBytes
	Q-zone（%Q）		128KBytes
	M-zone（%M）		512KBytes
	Power-failure holding zone		800KBytes
	Other variables		Undefined
Unit configuration	Number of extension modules	Digital module	Calculated from current consumption
		Analog module	
		External power supply	12V/16W
EtherCAT	Communication standard		IEC 61158 Type12
	EtherCAT master spec.		Class B (Compatible with Feature Pack Motion Control)
	Physical layer		100BASE-TX
	Modulation		Baseband
	Transmission speed		100Mbps (100Base-TX)
	Duplex mode		Full duplex
	Topology		Line, Bus and Star-type
	Transmission medium		Category 5 or higher twisted pair cable (aluminum foil + braided double shield cable)
	Max. transmission distance between nodes		100m
	Max. process data		Input: 5,736 bytes Output: 5,736 bytes (But the max. number of frames of process data is 4.)
	Communication cycle		Mini. 500μs

Technical parameters for Q1-series PAC

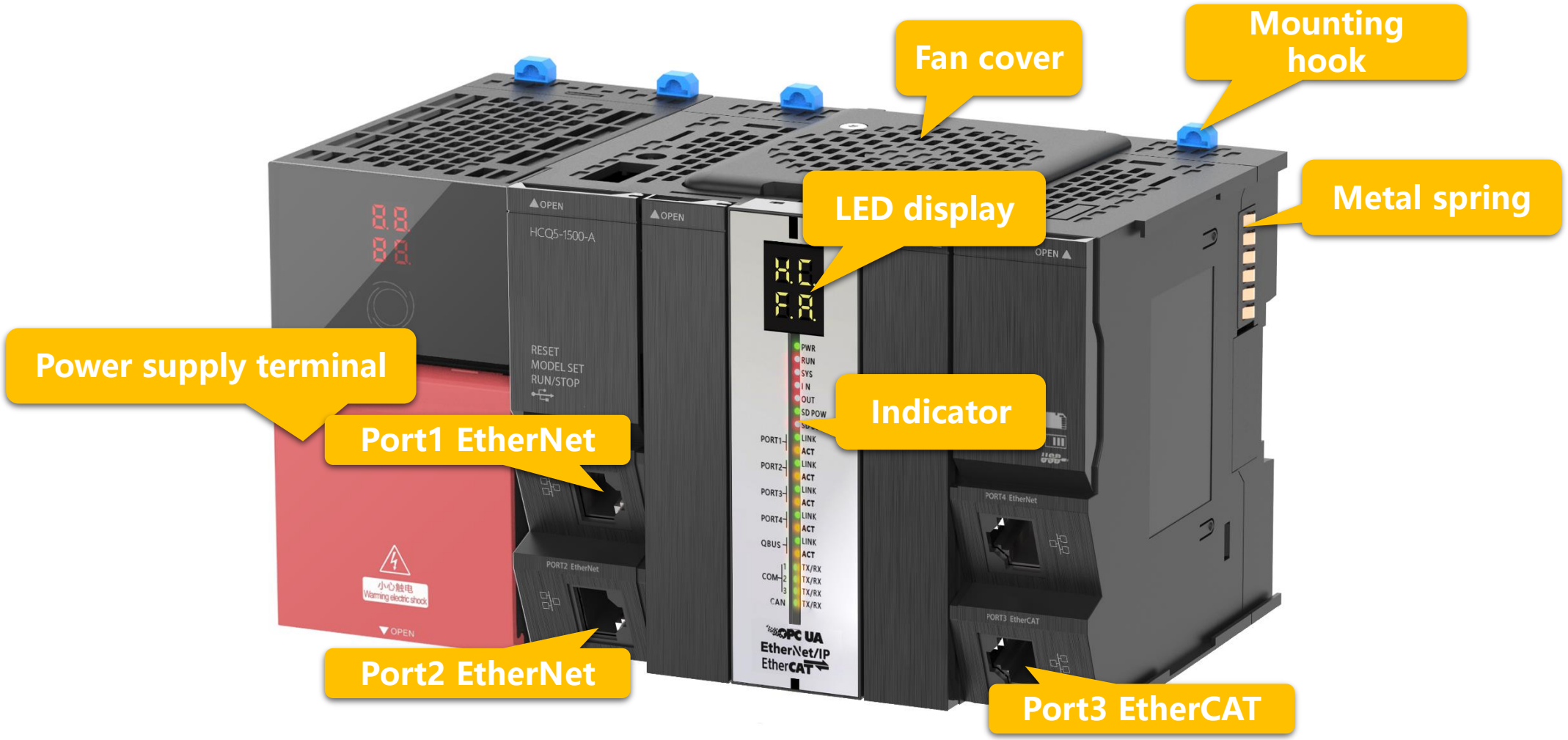
Item			Parameters
CNAOPEN master	Link layer		CAN2.0A
	Terminating resistor		Built-in 120Ω, support DIP switching
	Support baud rate bps		50K,100K,125K,250K,500K,800K和1M
	Transmission medium		Category 5 or higher twisted pair cable
	Max. communication distance		1000m（50Kbps）
	Max. number of slaves		32
	Communication cycle		Mini.1ms
Serial port	Physical layer	COM1,COM2	RS485
		COM3	RS232
		COM1,COM2	Support 120Ω，support DIP switching
	Terminating resistor		4800~115200
		COM1,COM2	500m
	Baud rate bps	COM3	15m
	Max. communication distance	COM1,COM2	Line, Bus and Star-type
			Point to point
	Topology	COM1,COM2	32
		COM3	1
	Max. number of slaves		Category 5 or higher twisted pair cable

HCQ5

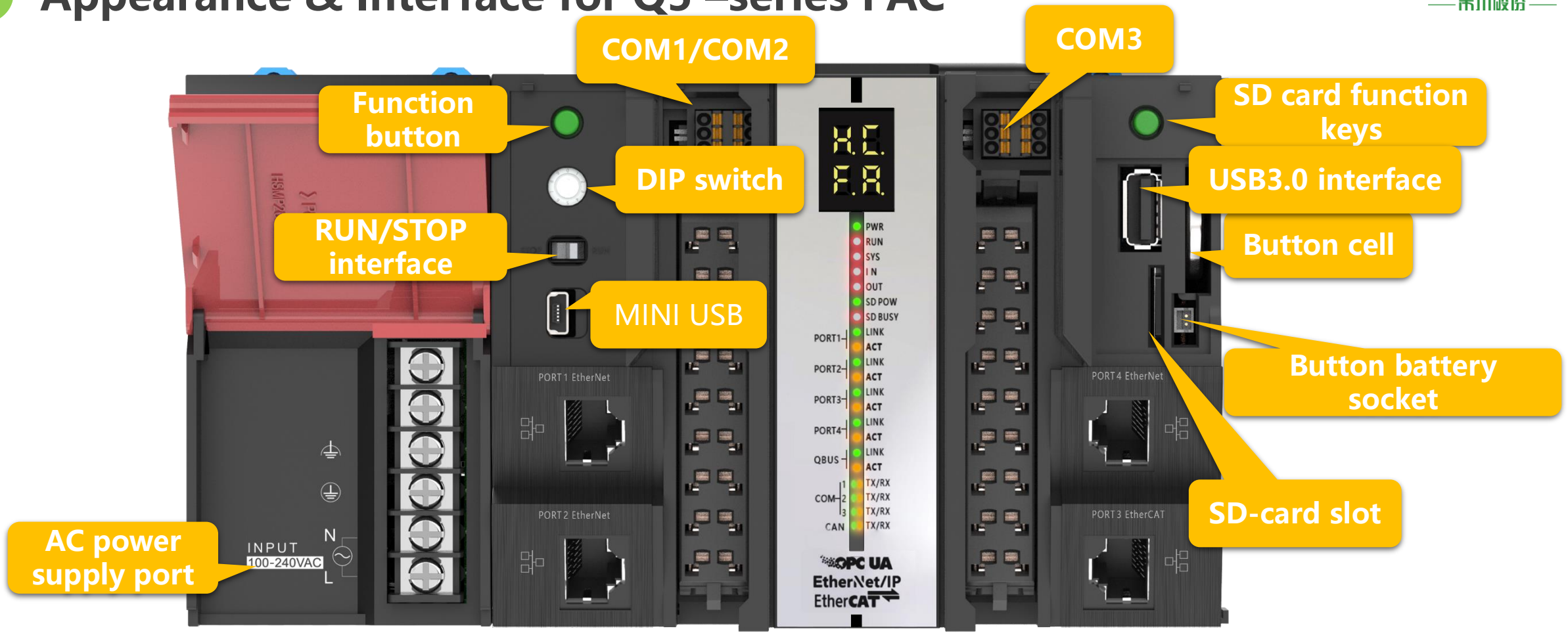
Basic Intelligent Controller

To create a better life through our work

Appearance & Interface for Q5 –series PAC

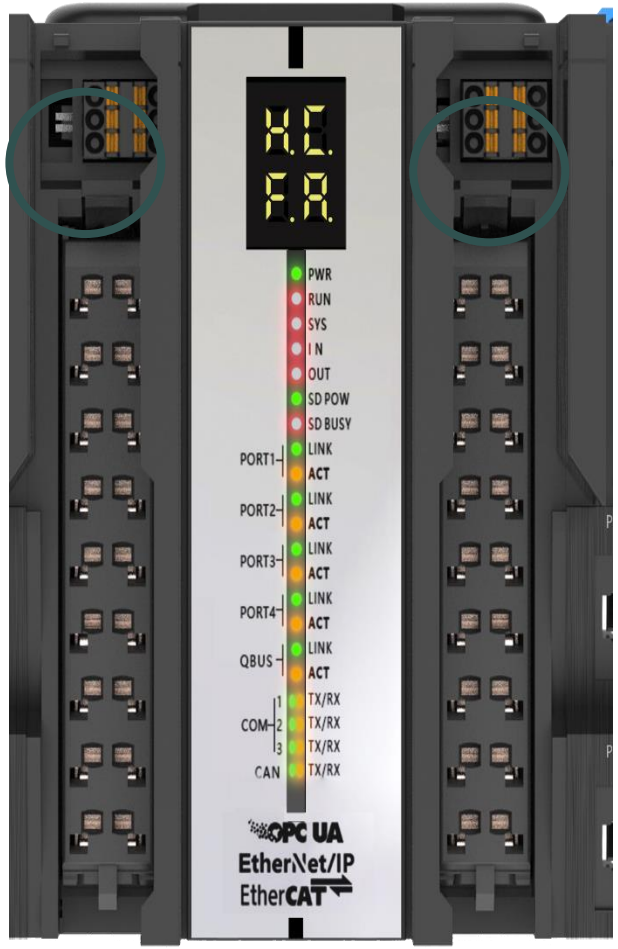


Appearance & Interface for Q5 –series PAC



COM1(RS485)	Modbus RTU (master/slave supported) Free Protocol for serial port
COM2(RS485)	
COM3(RS232)	

Appearance & Interface for Q5 –series PAC



Name	Function
COM1	Hardware RS485
COM2	Hardware RS485
COM3	Hardware RS232

- COM1~COM3
- Baud rate： 4800-115200bps
- Communication format: support parity, stop bit: 1 or 2 bits
- Communication protocol: Modbus RTU master-slave and non-protocol communication

Appearance & Interface for Q5 –series PAC

Terminal	Description
N/A	Do not connect
N/A	Do not connect
Signal ground	Shield ground
Grounding	Grounding
N	220V AC neutral wire input
L	220V AC live wire input

Power spec.

Q1 power spec.	
Voltage	AC100~240V
Voltage fluctuation range	-15%~20%
Power input	100W
Undervoltage level	AC80V
Output voltage	12V
Voltage fluctuation	±5%
Output Power	60W

Technical parameters for Q5-series PAC

Item			Parameters
Programming	Total program capacity		16MBytes
	I-zone（%I）		128KBytes
	Q-zone（%Q）		128KBytes
	M-zone（%M）		512KBytes
	Power-failure holding zone		800KBytes
	Other variables		Undefined
Unit configuration	Number of extension modules	Digital module	Calculated from current consumption
		Analog module	
		External power supply	12V/30W
EtherCAT	Communication standard		IEC 61158 Type12
	EtherCAT master spec.		Class B (Compatible with Feature Pack Motion Control)
	Physical layer		100BASE-TX
	Modulation		Baseband
	Transmission speed		100Mbps (100Base-TX)
	Duplex mode		Full duplex
	Topology		Line, Bus and Star-type
	Transmission medium		Category 5 or higher twisted pair cable (aluminum foil + braided double shield cable)
	Max. transmission distance between nodes		100m
	Max. process data		Input: 5,736 bytes Output: 5,736 bytes (But the max. number of frames of process data is 4.)
	Communication cycle		Mini. 250μs

Naming Rule for Q-series Modules

Coupler module

HC QX - EC - D -X X X X

Product name

SYM	Type
HC	HCFA controller

Series name

SYM	Type
QX	QX slave module
RX	RX slave module

Function module

SYM	Type
ID	Digital input
OD/OC	Digital output
MD	Digital I/O
AD	Analog input
DA	Analog output
MA	Analog I/O
TS	Temperature input
TC	Temperature control
HB	Disconnection detection
HC	High-speed input
LC	Load cell
PG	Pulse output
GM	Motion control
RS	Serial communication
IL	IO-LINK
AW	Any-wire
EC	Ethercat coupler
PD	Power extension
PF	IO power extension
TX	Grounding terminal extension

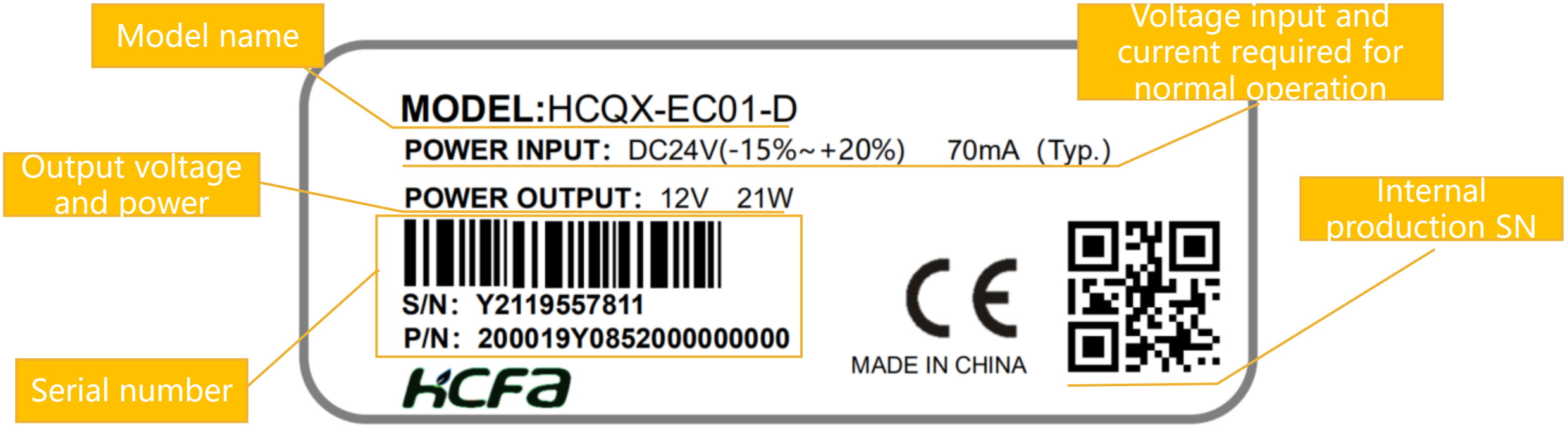
Power type

SYM	Type
D	DC power
A	AC power

Control No. and customization

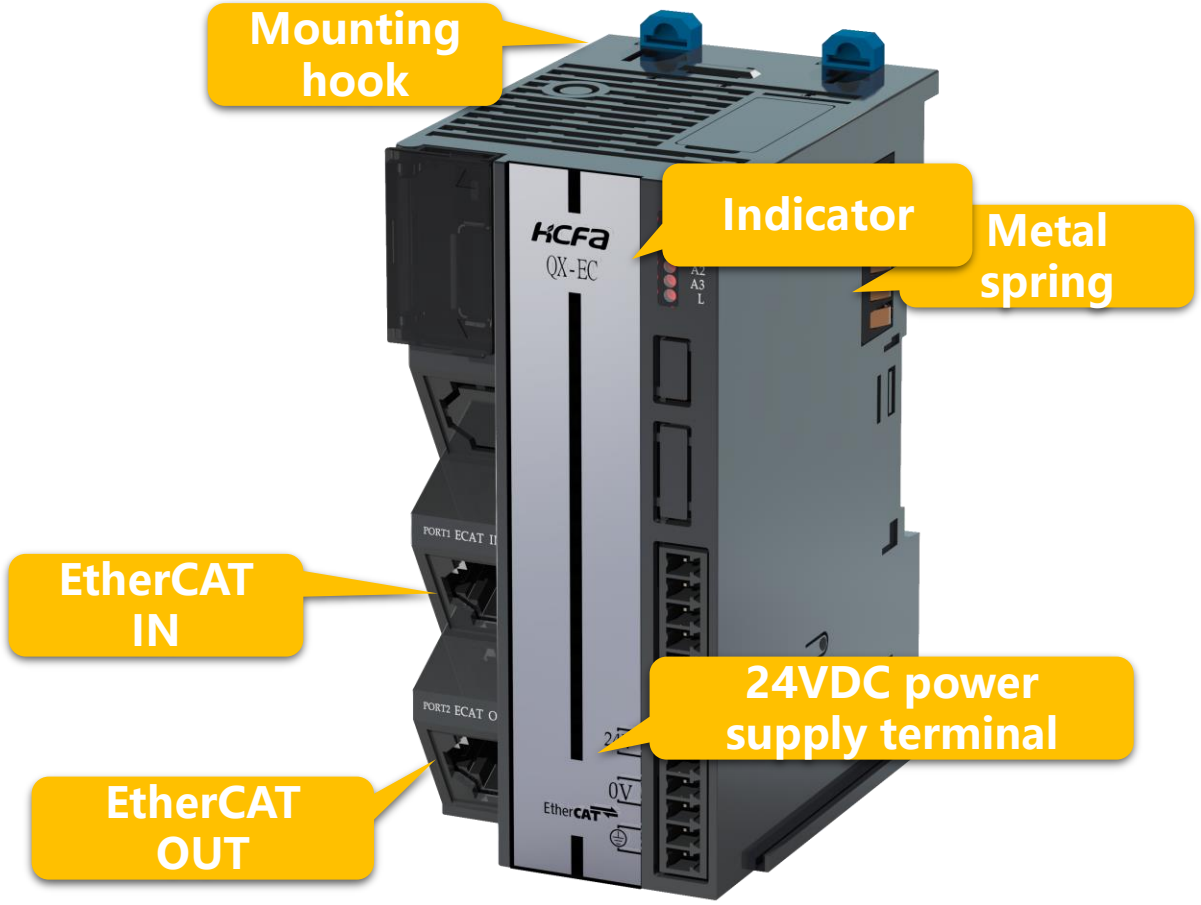
Remark: Other function modules are under development, except ID OD MD AD DA EC TS modules and OD is transistor output and OC is relay output

Nameplate Description for Coupler Module



Appearance & Interface for Coupler Module

EC coupler



Naming Rule for Q-series Modules

Q-series extension module

HC QX - OD 16 - DX X X X

Product name

SYM	Type
HC	HCFA controller

Series name

SYM	Type
QX	QX slave module
RX	RX slave module

Function module

SYM	Type
ID	Digital input
OD/OC	Digital output
MD	Digital I/O
AD	Analog input
DA	Analog output
MA	Analog I/O
TS	Temperature input
TC	Temperature control
HB	Disconnection detection
HC	High-speed input
LC	Load cell
PG	Pulse output
GM	Motion controller
RS	Serial communication
IL	IO-LINK
AW	Any-wire
EC	Ethercat coupler
PD	Power extension
PF	IO power extension
TX	Grounding terminal extension

Outputs

SYM	Type
16	16-ch output

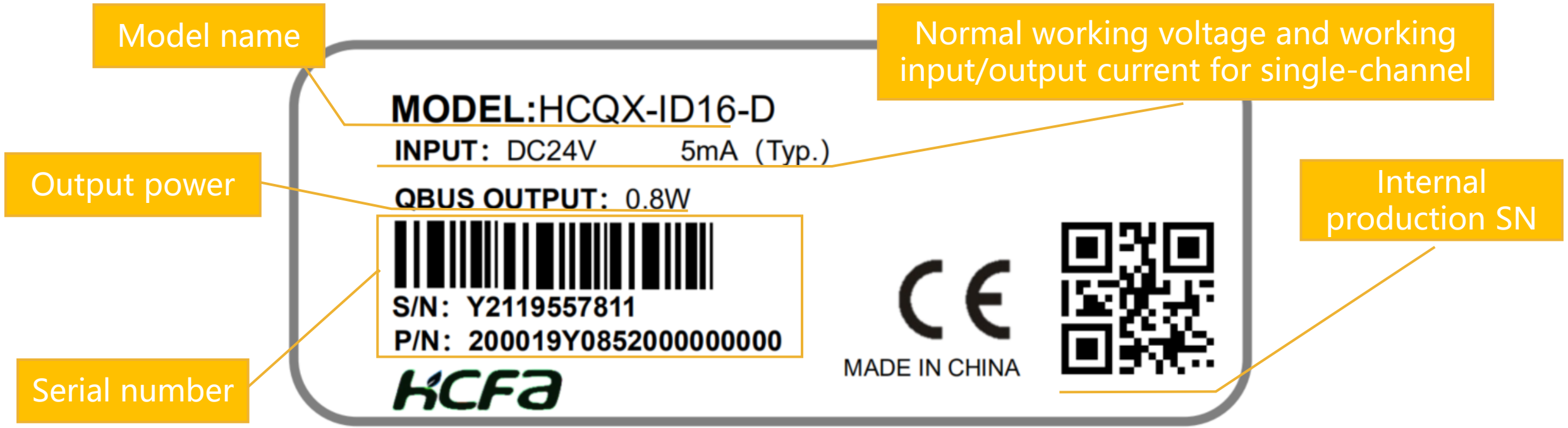
Power type

SYM	Type
D	DC power
A	AC power

Control No. and customization

Remark: Other function modules are under development, except ID OD MD AD DA EC modules and OD is transistor output and OC is relay output

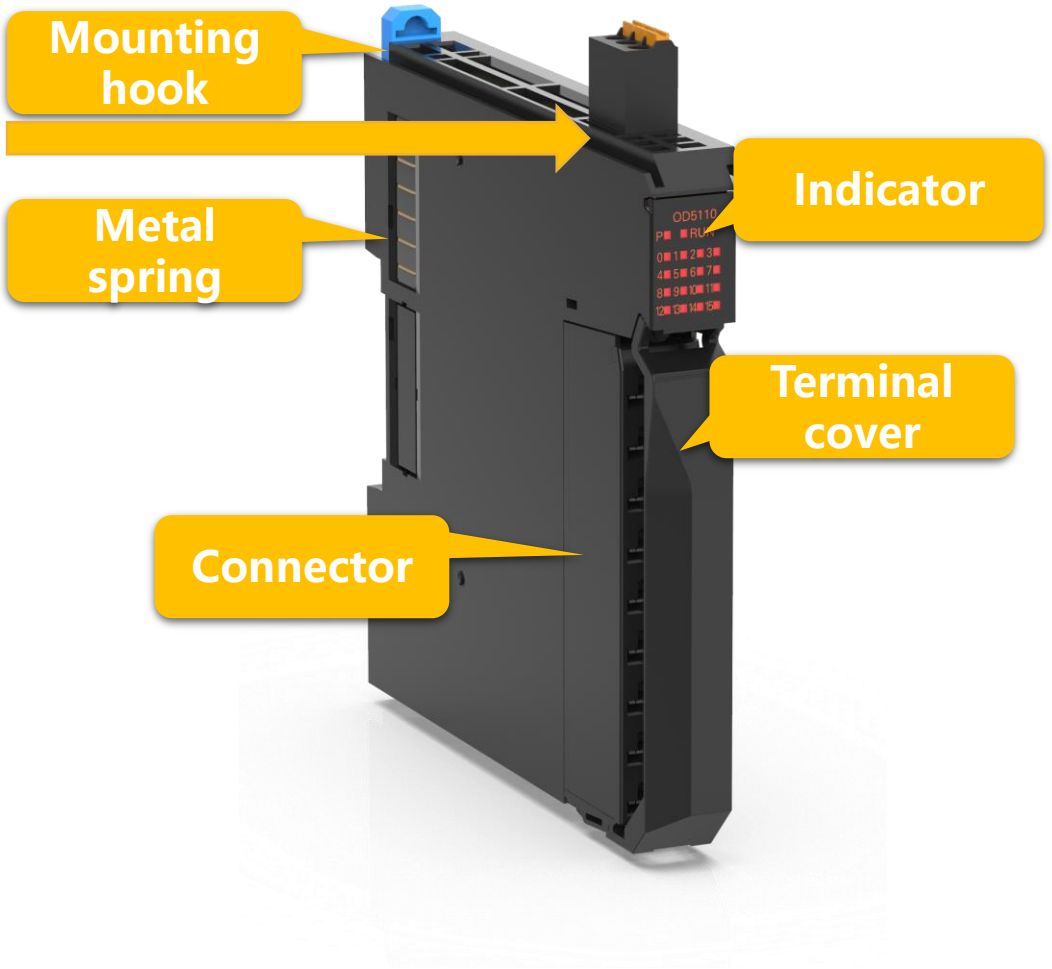
Nameplate Description for Q-series Extension Module



Appearance & Interface for Q-series Extension Modules

Note: On the top of the AD/DA/TS module, 24VDC needs to be supplied.

Remote Extension Module



PART 02

Software Function

Software Function

Motion control

High-speed pulse I/O（200KHZ）(Q1 supported，Q0 Q5 will support

Single axis positioning and constant speed

Electronic gear

Electronic Cam\Flying Shear\rotary Shear

Linear \Circular \Helical Interpolation

CNC G-code control\ Robot control

Protocol

CANopen

Modbus

RS232/RS485

EtherNet/IP

OPC UA

EtherCAT

Other functions

SD-card capacity 32G

- Program
- CPU parameter
- I/O configuration
- Recipe

Software Features

High speed & high config.

PAC

Flexible system config.

The number of slave stations is up to 65535

Various modules (digital，analog，temperature，positioning)

CODESYS platform

Based on standard PLC OPEN standard

Built-in 16-ch high-speed I/O（Q1）

Ultra-short synchronization period

Compatibility

Various bus interfaces

(MODBUS TCP

MODBUS RTU

CANOPEN

EtherCAT

OPC UA

EtherNet IP)

High ease of use

Modularization and standardization

6 programming languages (ST,LD,IL,SFC,CFC,FBD)

USB driver-free

12mm ultra-narrow module & can be pulled/pushed Terminal block

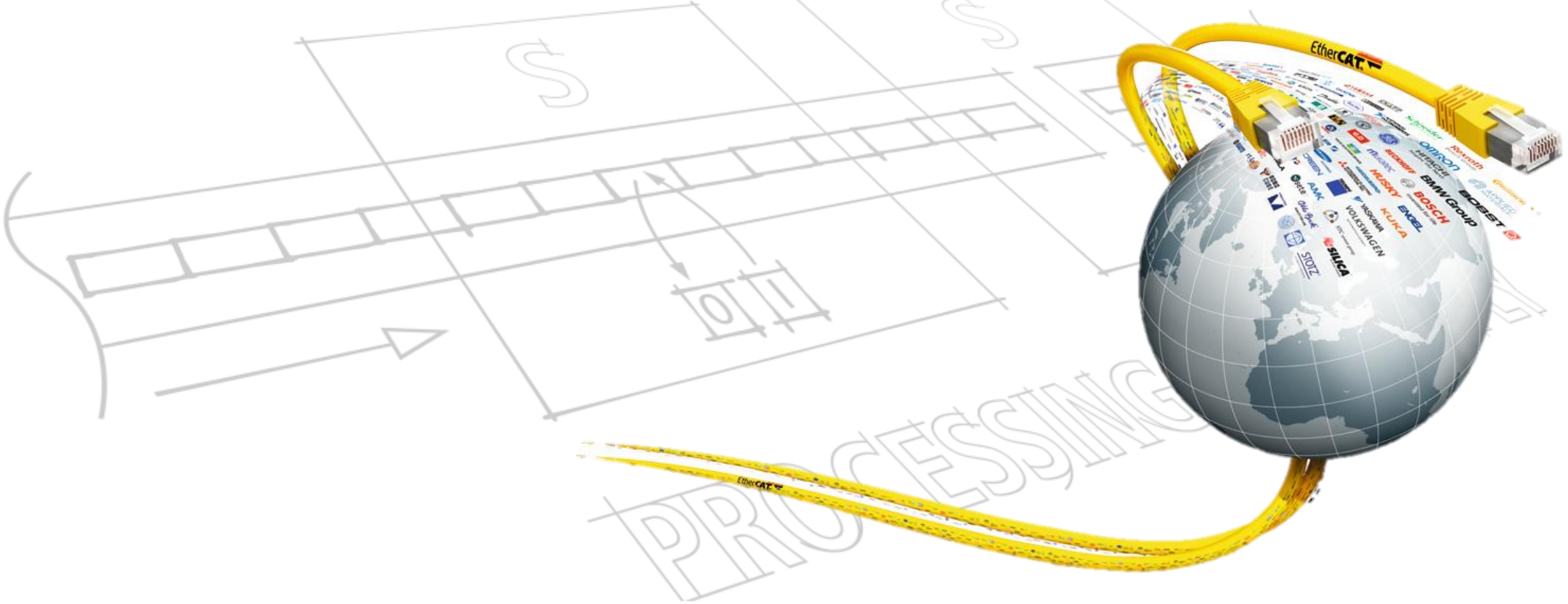
Cost-effective

Standard Ethernet interface , no additional hardware

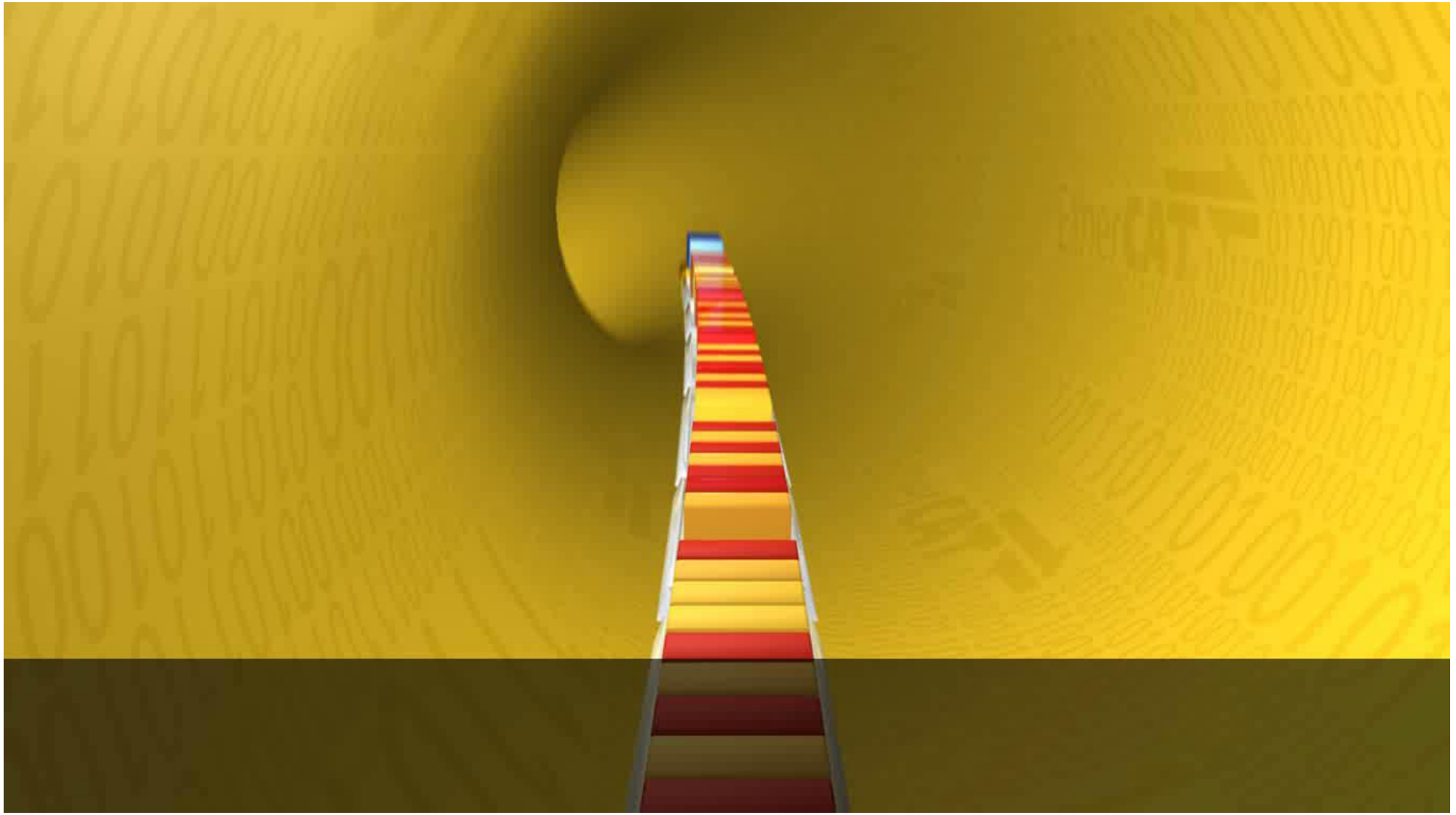
Standard development language and motion control library save development and maintenance costs

All hardware is designed by HCFA to save production costs and achieve higher cost performance

EtherCAT-Industrial Ethernet Fieldbus



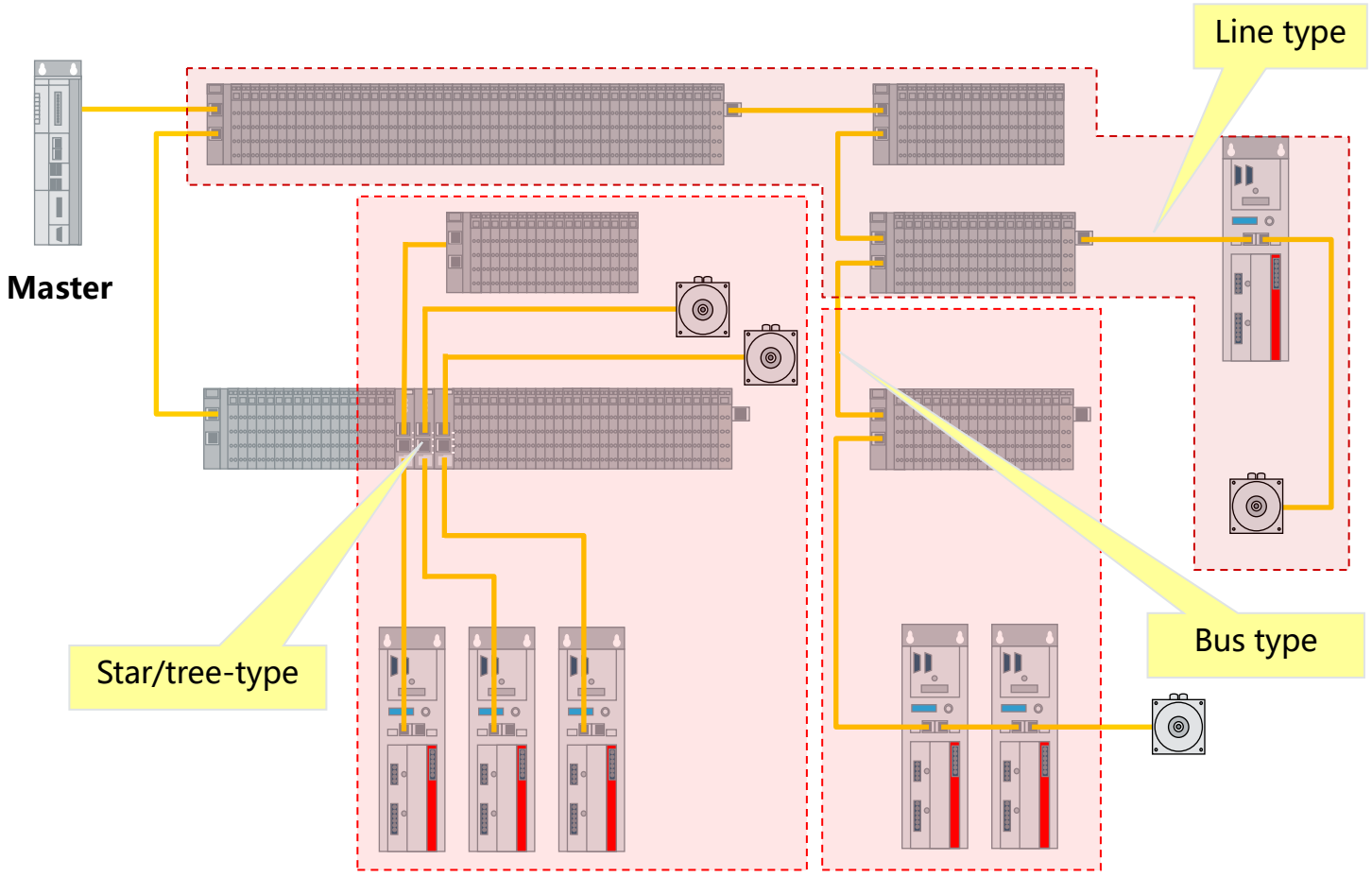
EtherCAT Principle



EthercAT on the fly

Topologies Realized by EtherCAT

Topology: Topologies that support any structures



EtherCAT max. number of nodes

Nodes：65535 theoretically

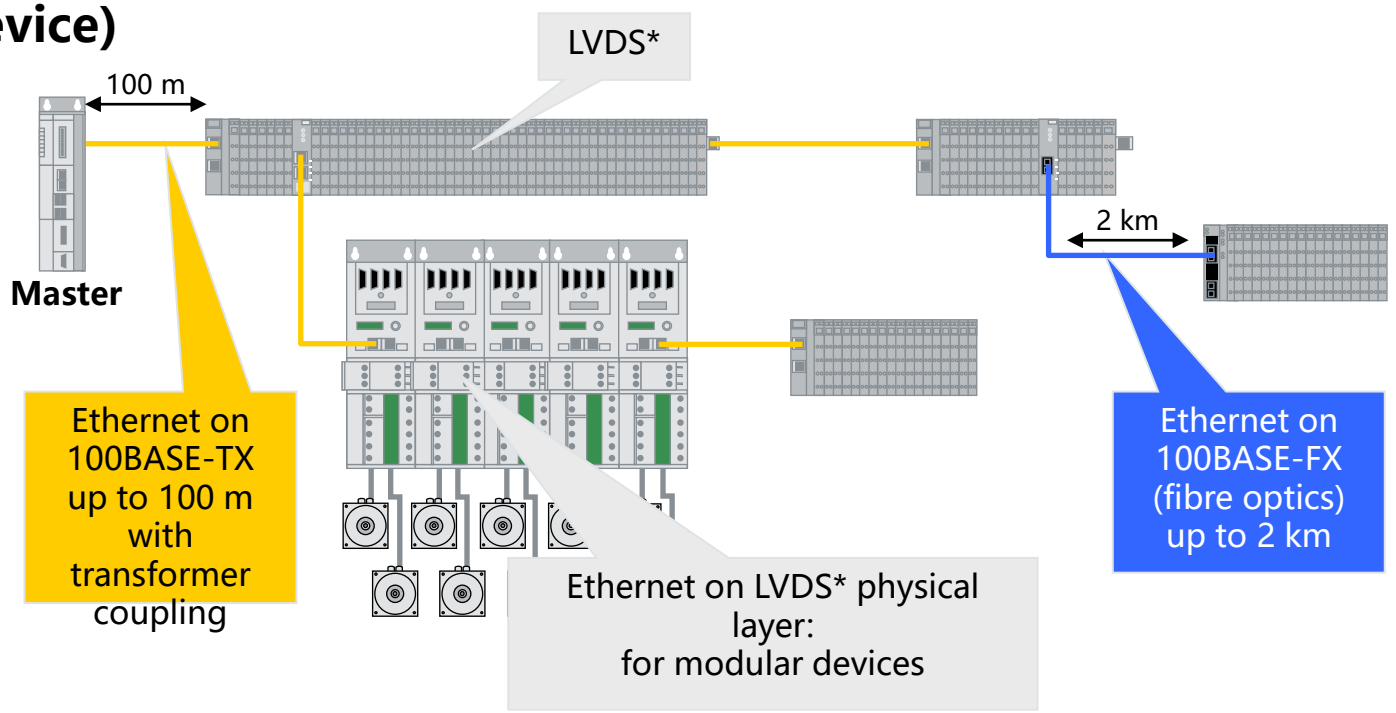
Defined by the EtherCAT slave address assignment
Slave power supply and network delay need to be considered

EtherCAT Communication Distance

Simple physical layer

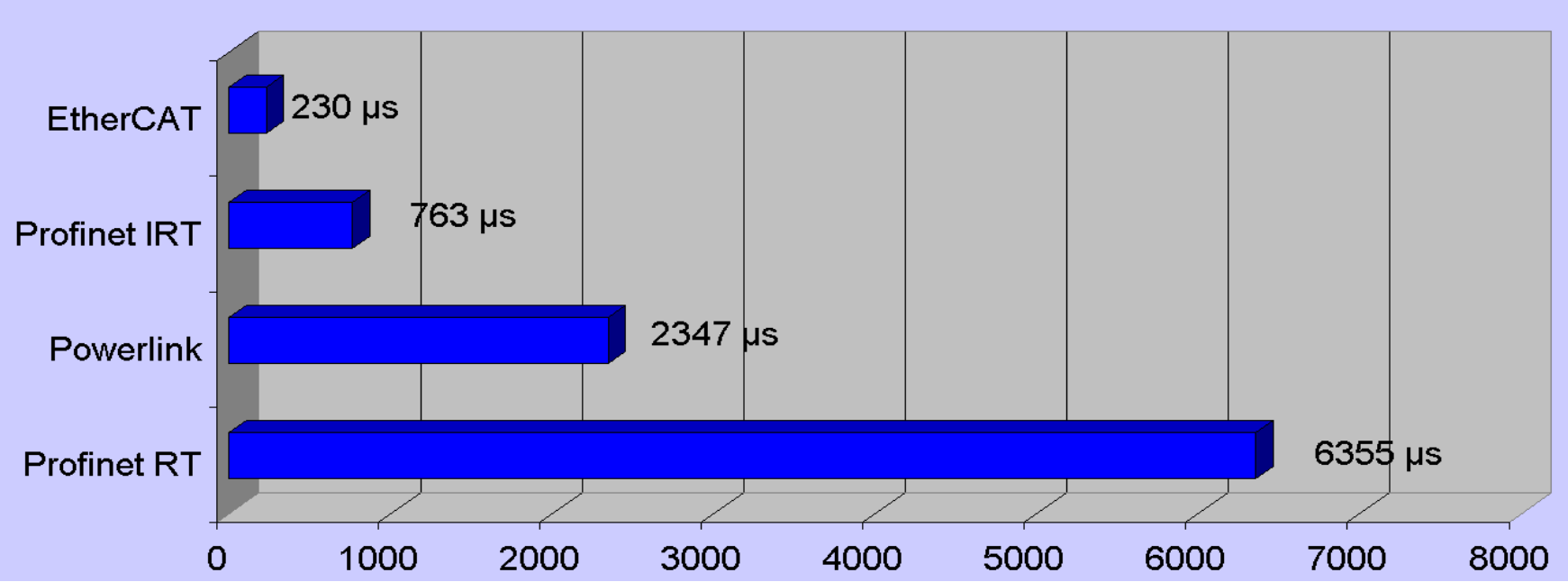
Physical layer：100 Base-Tx，100 Base-Fx，LVDS

EtherCAT supports multiple Ethernet physical layers:100BASE-TX (Max. distance between two nodes is 100 meters)100BASE-FX (Max. distance between two nodes is 2 km)LVDS (Modular Device)



EtherCAT Communication Speed

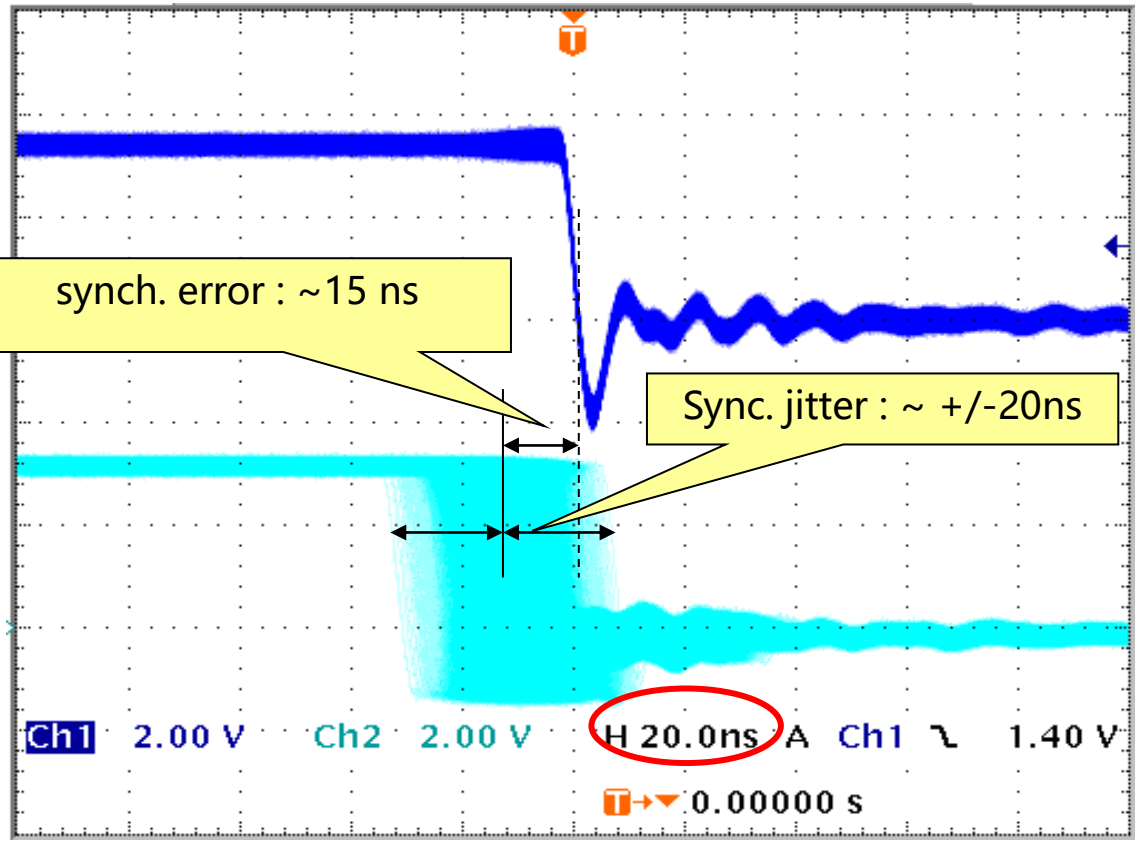
- 40 axes (6 bytes of input/output data per axis)
- 50 I/O stations for a total of 560 EtherCAT Bus Terminals
- 2000 digital + 200 analog I/O, bus length 500 m
- EtherCAT performance : Cycle time **230µs** at 33% bus load, message length 77µs



* Source: Ethernet Powerlink Spec V 2.0, App.3

EtherCAT Synchronization Accuracy

EtherCAT bus features - Precise synchronization



EtherCAT: Ethernet Control Automation Technology

- Long-term observation of two independent devices with an oscilloscope: 300 nodes with cable length of 120 meters ,

Blue: The synchronization signal of the reference node is used as the trigger signal of the oscilloscope
Cyan : Multiple sync signals for test nodes

Supported Vendors by EtherCAT

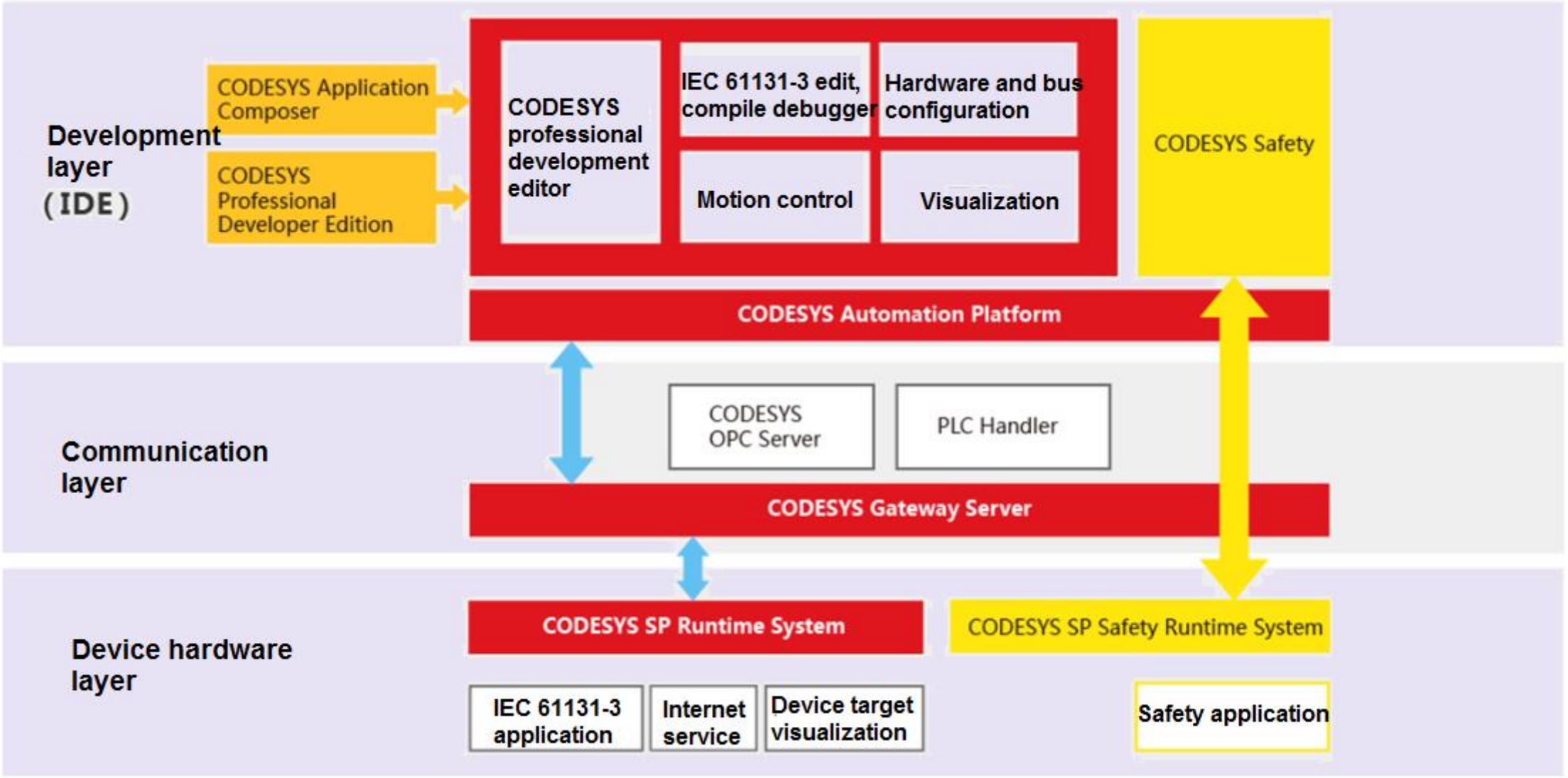
Software Function-CODESYS

Q series PLC chooses CODESYS as the software development platform.

As a powerful industrial automation software development platform, CODESYS has the following characteristics：

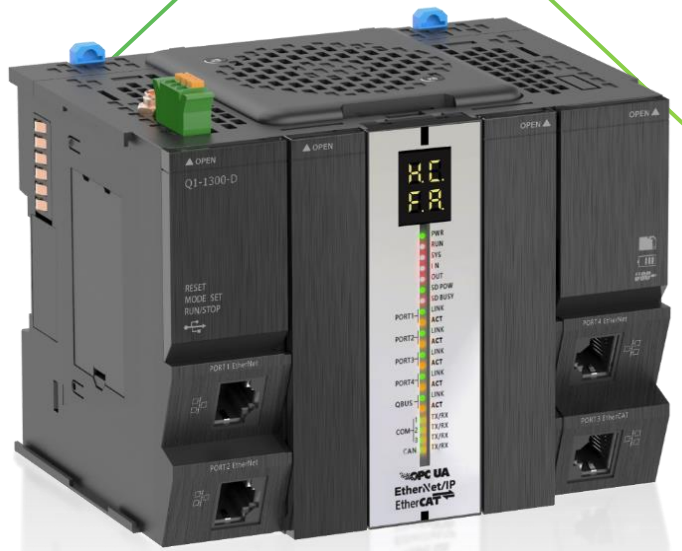
- Using IEC61131-3, it supports a variety of programming languages including ST\CFC\SFC\IL\LD\FBD. When programming projects in different industries and fields, users can freely choose the appropriate programming language
- Support data interaction with high-level languages
- Introducing the programming ideas, program frameworks and complex algorithms accumulated in the computer field into the field of industrial control, greatly enriching the flexibility and possibility of industrial control
- Provide HMI interface editing window, engineers can design more user-friendly screen
- Provide data collection, monitoring and analysis tools
- Provides error diagnosis and in-circuit emulation capabilities, allowing users to program and debug without actual device

Software Platform Architecture



PART 03

Application Training



Application training

-

01-Programming software installation

To create a better life through our work

01-Programming Software Installation

CODESYS installation and configuration requirements

Description	Mini. configuration	Recommended 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

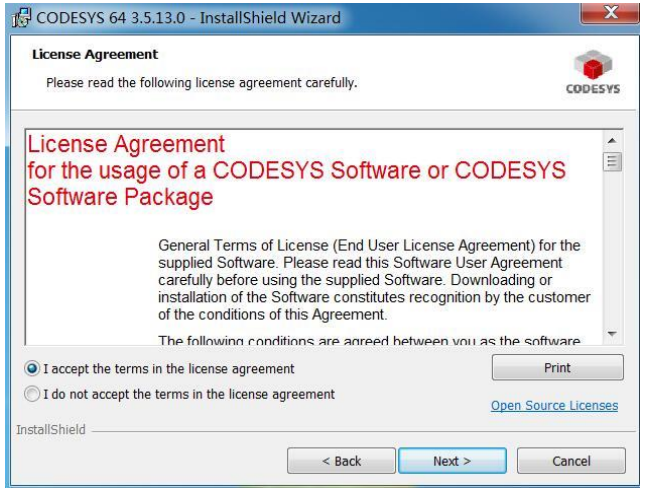
The software installation package can be downloaded from HCFA website, the download link is：

class.hcfa.cn

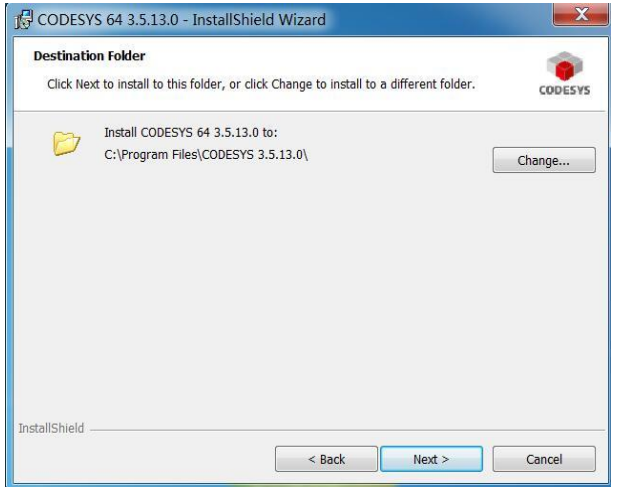
01-Programming software installation

1

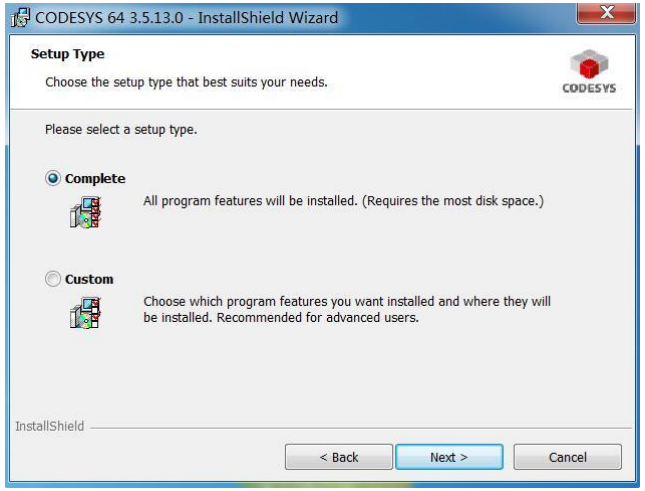
2



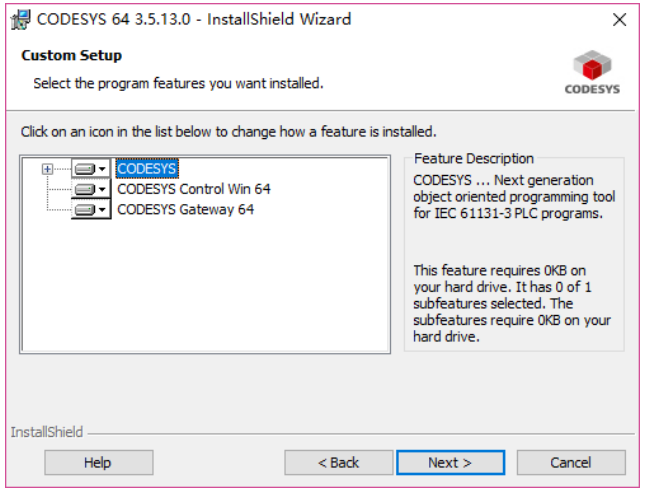
3



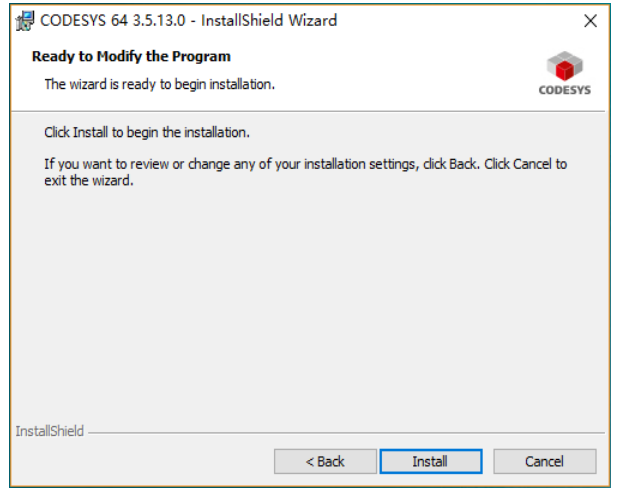
4

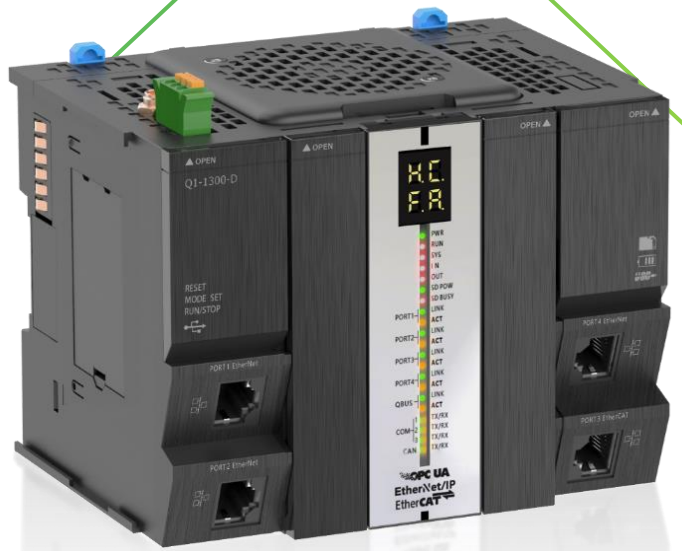


5



6





Application training

-

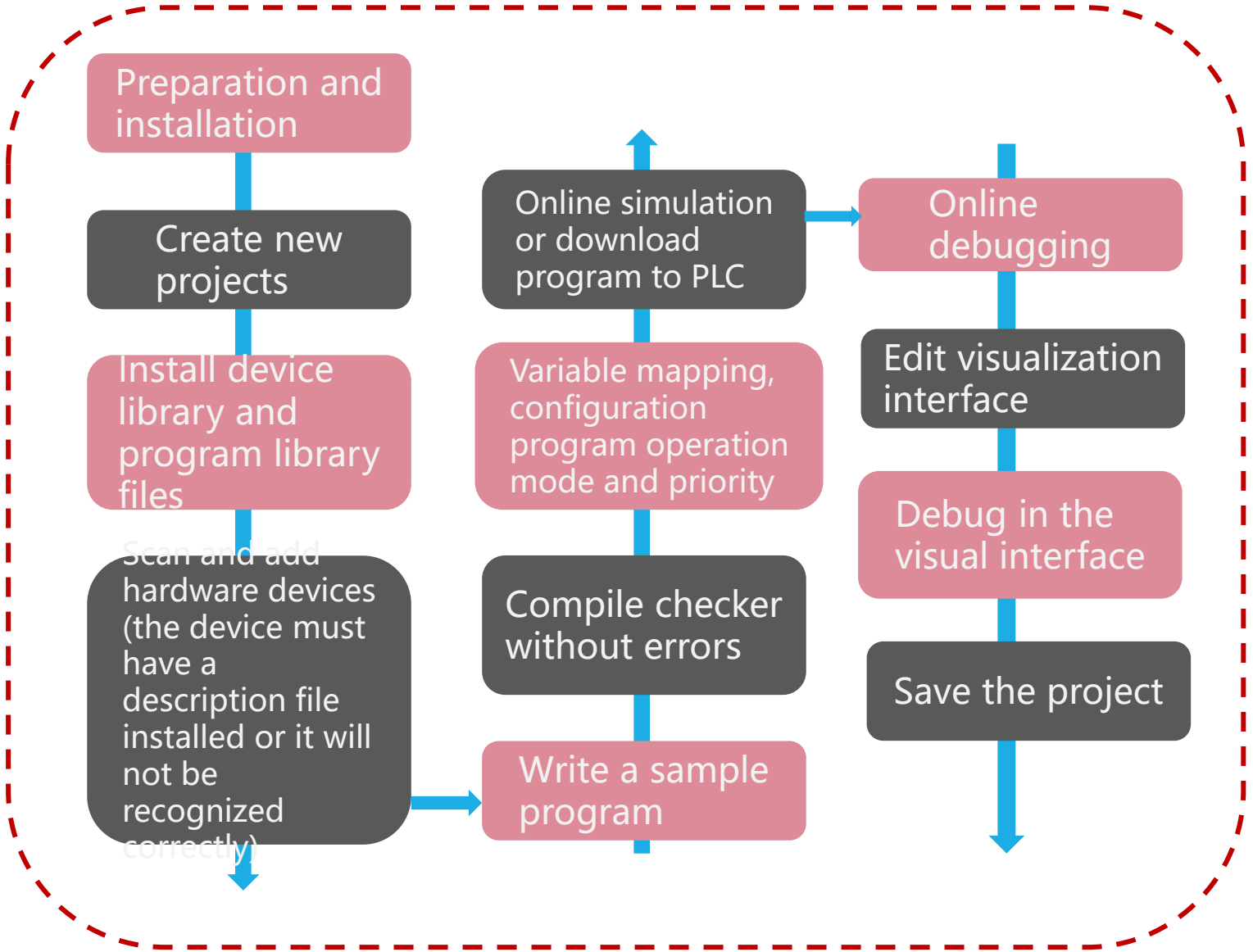
02-Create New Projects

To create a better life through our work

02-Create New Projects

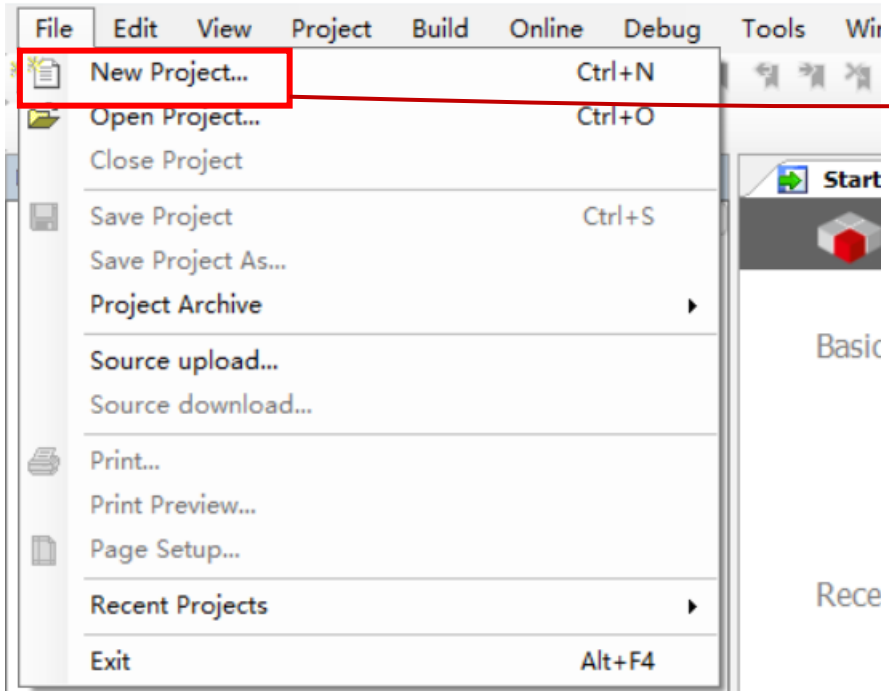
Use programming software to write a simple sample program to realize single-axis motion (such as absolute positioning, relative positioning, or jogging, etc.)

Create new projects

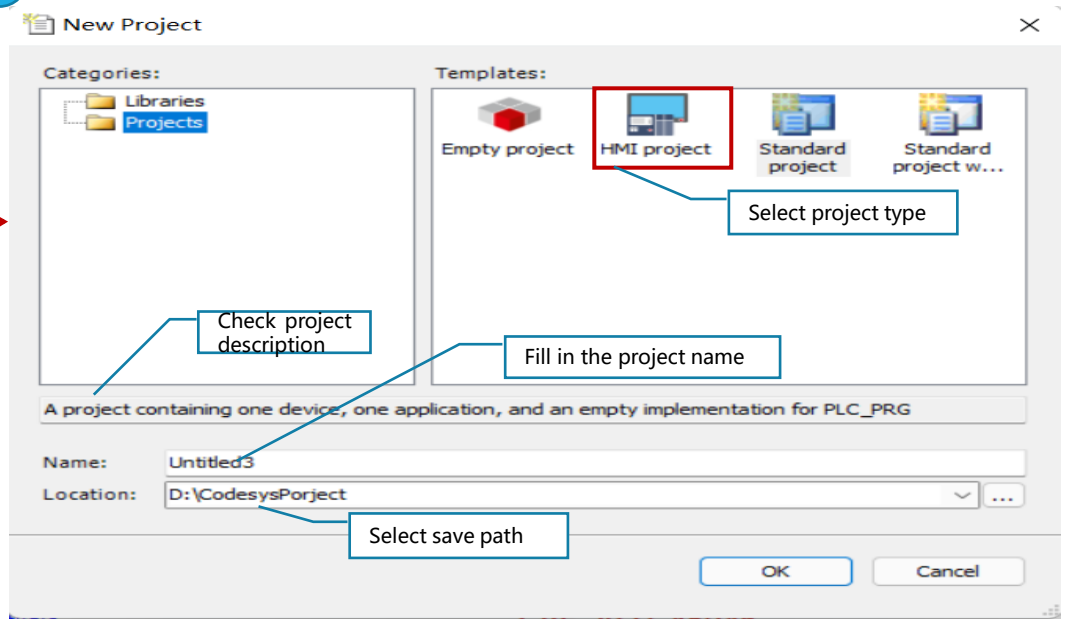


02-02-Create New Projects

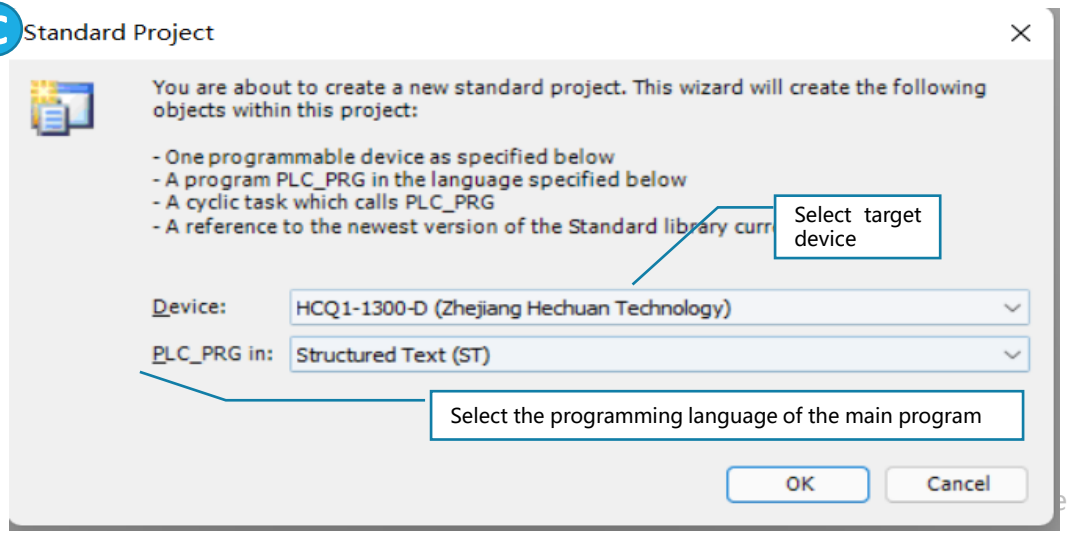
a

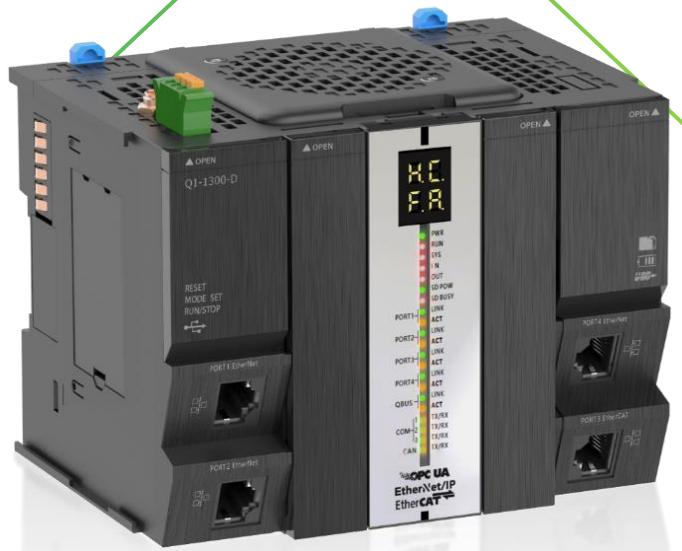


b



c





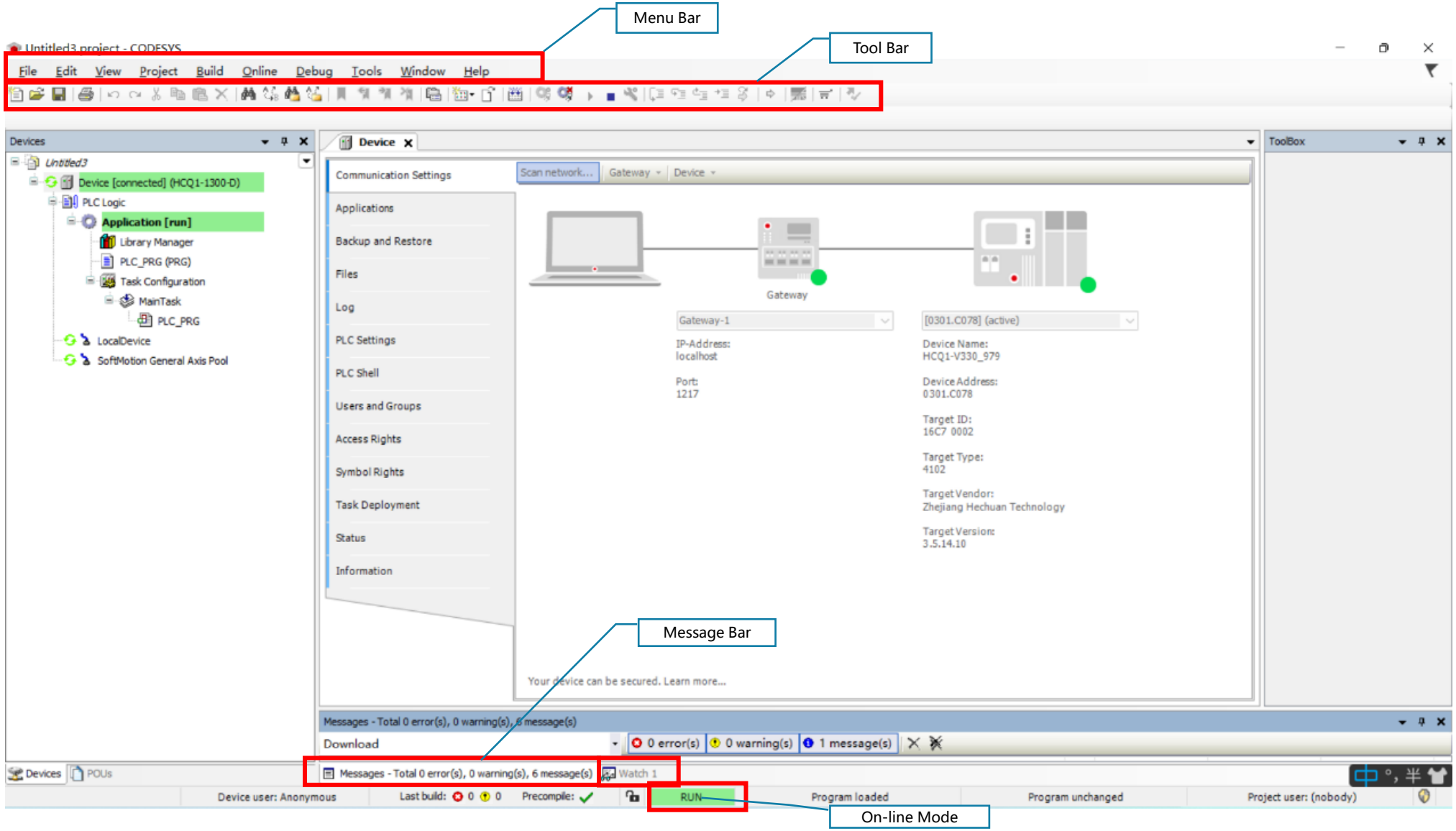
Application training

-

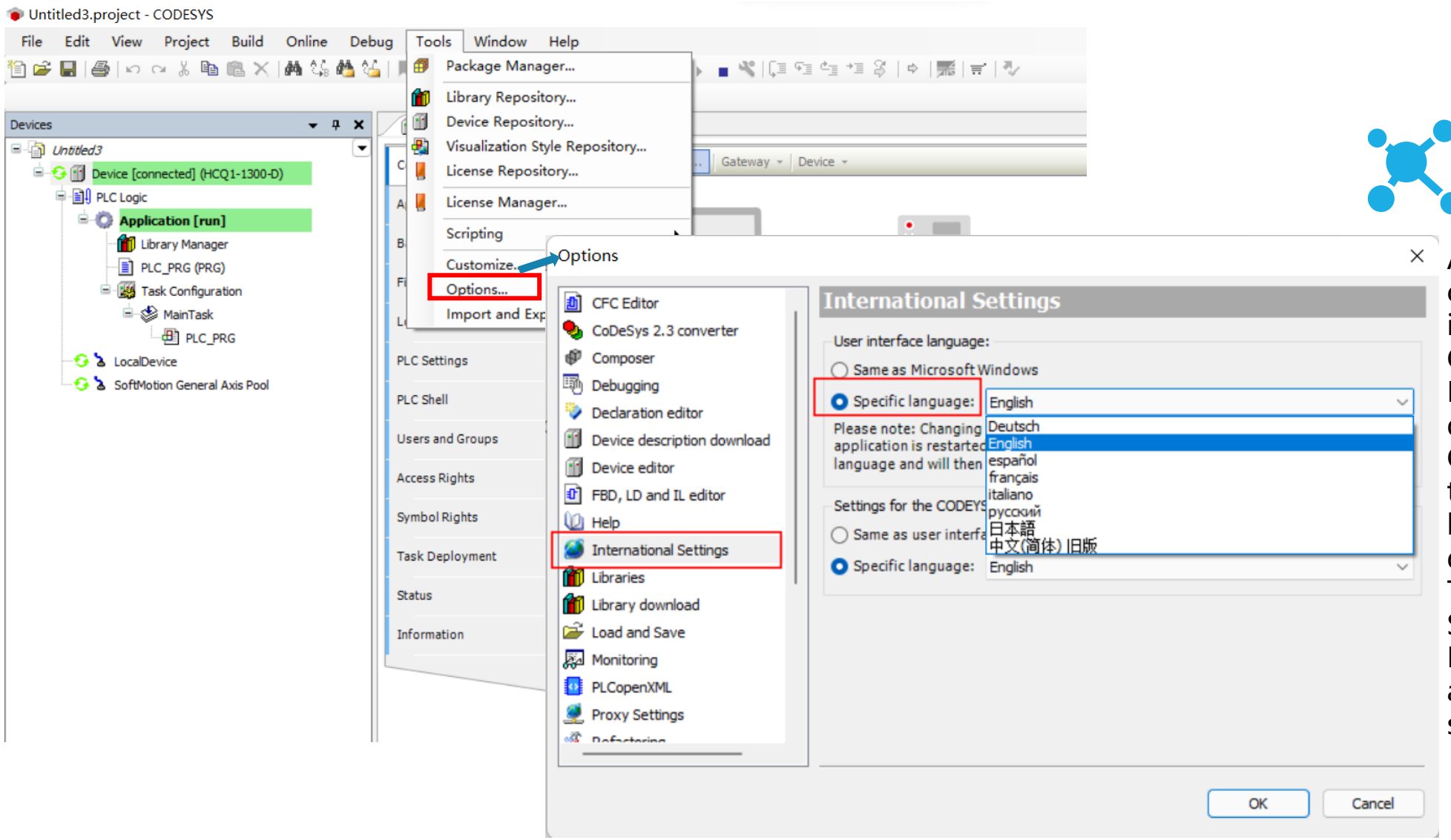
03-Programming Interface Introduction

To create a better life through our work

03-Programming Interface Introduction



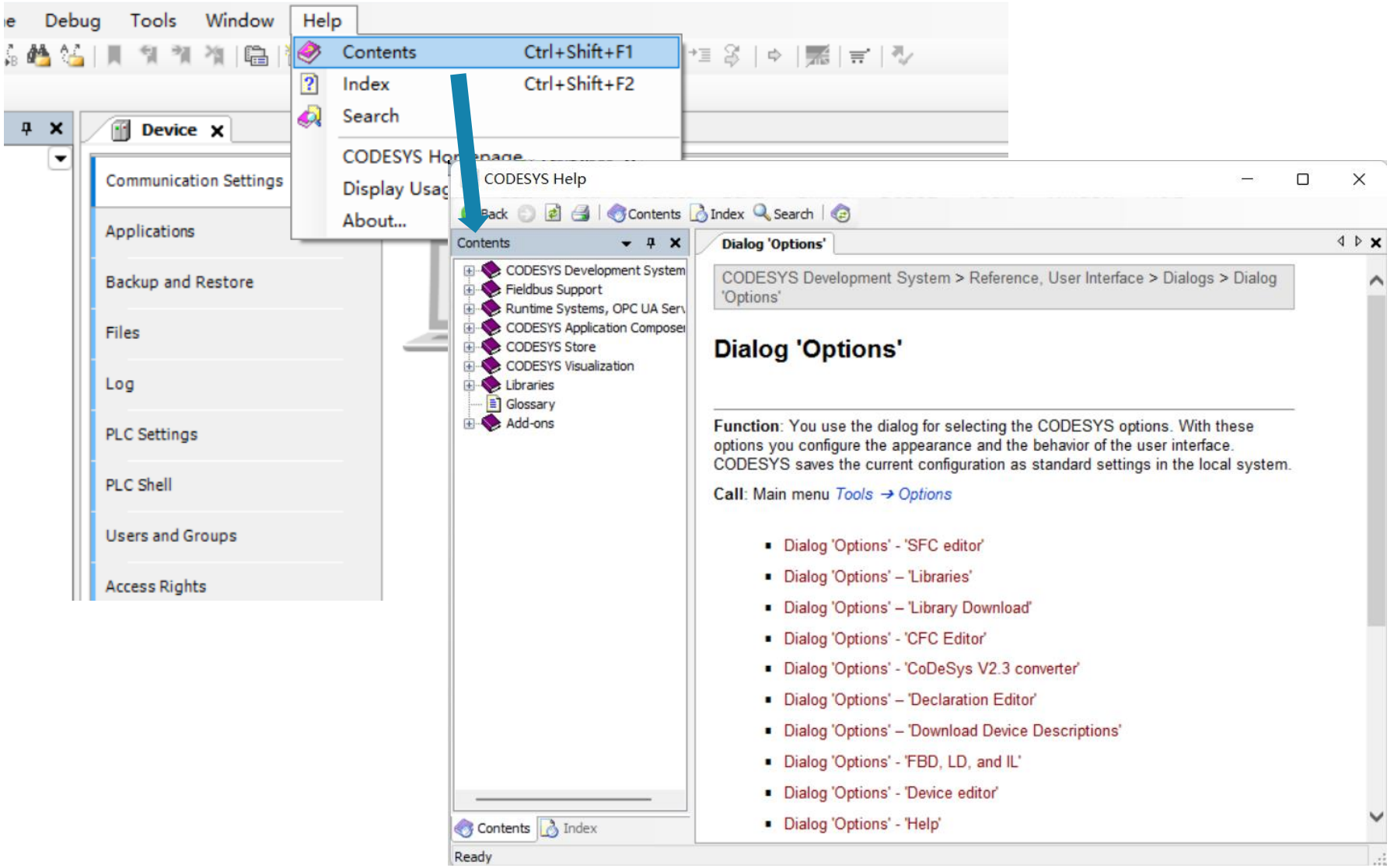
03-Programming Software Settings



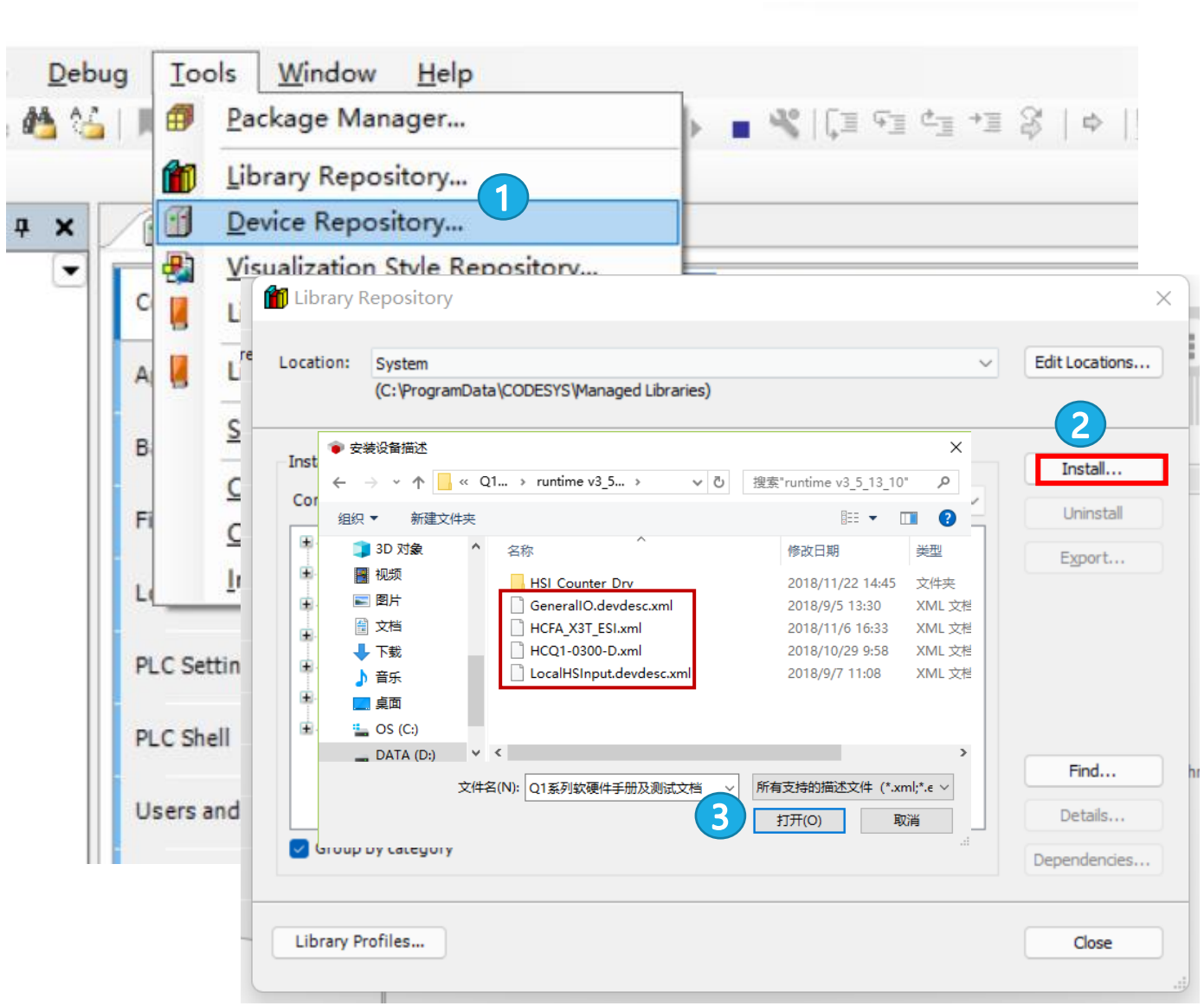
After the installation is completed, the default interface display language of CODESYS is English by default. If users need to switch the display language of the CODESYS software interface, they can switch the interface language through the drop-down tab in the menu bar Tools→Options→Language Settings→User Interface Language, and click Confirm and restart the CODESYS software to make the settings.

03-Programming Software Help System

After the CODESYS is installed, the help instructions for the software will be installed by default. After opening the software, users can find "Help" in the menu bar, and click "Contents" to open the online help. Users can quickly search based on index or search keywords.



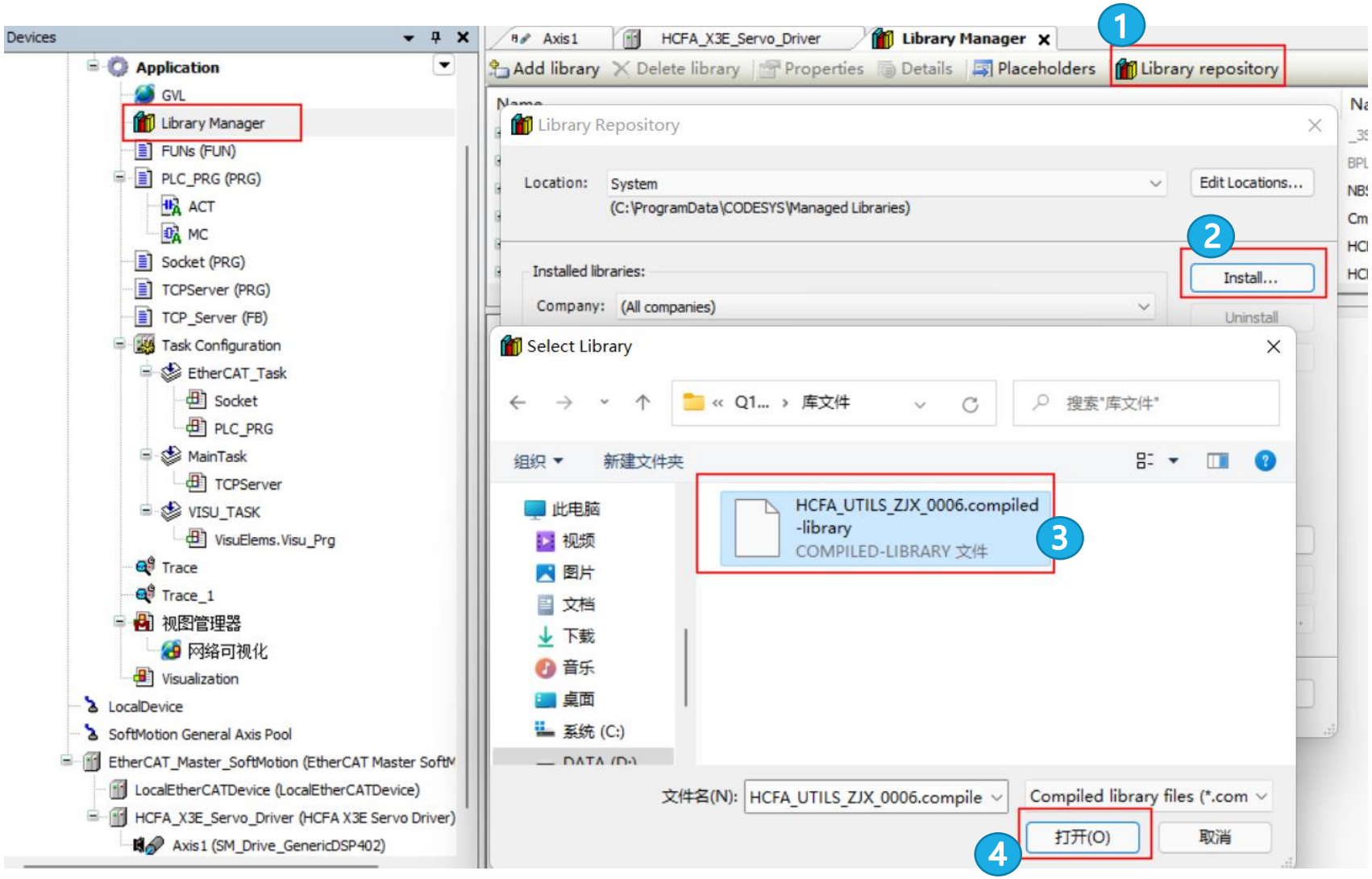
03- Installation of Device Description Files



The "Device Library" is for installing and uninstalling devices, and for users to view device information. Only after the subordinate device is added to the device library, the user can find the corresponding subordinate device in the "Add Device" tab, otherwise the subordinate device cannot be used. . Select Tools→Device Library, and select "Install" in the dialog box of the device library to import the corresponding device files. The devices that can be added include the supplier's PLC, SoftMotion motion control equipment (encoders, drives, etc.), fieldbus and special interface and other equipment.

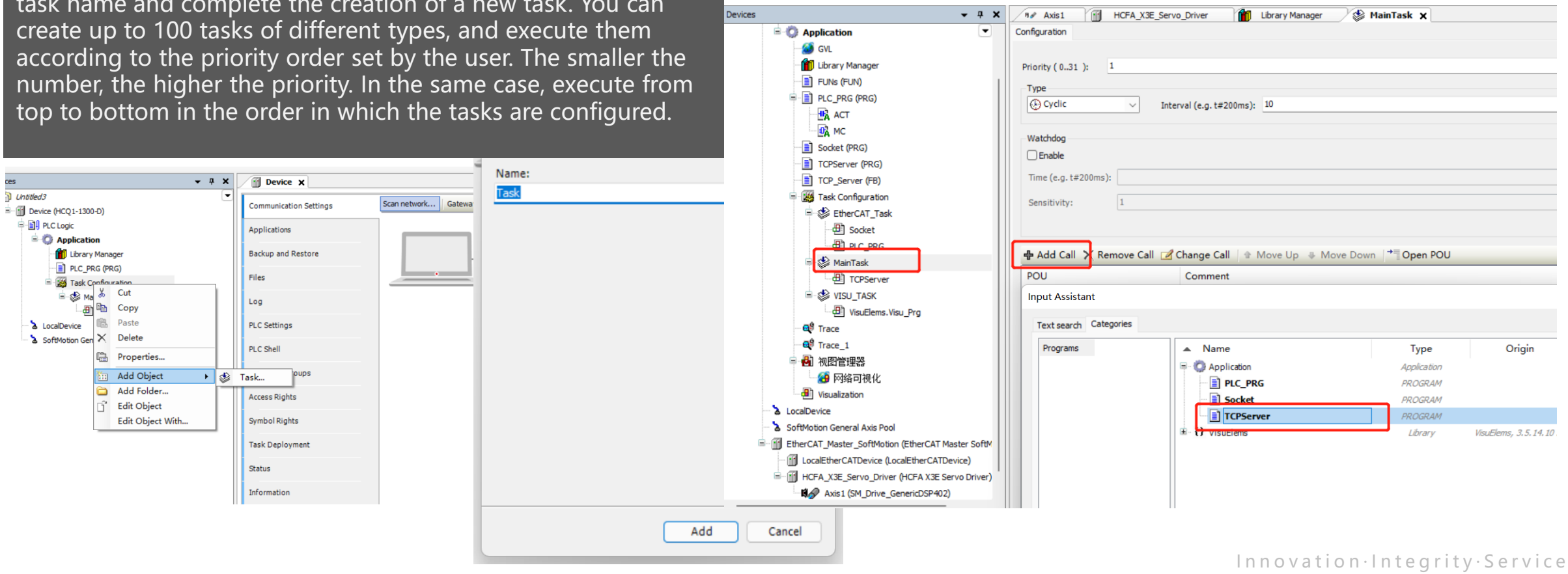
03-Installation of Library Files

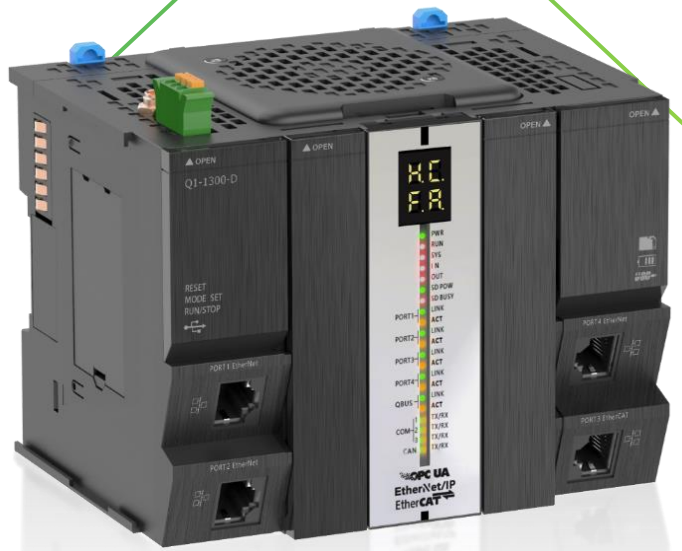
If a project needs to use an external library, the user needs to install the library file by himself, double-click the "Library Manager" in the left tree menu, enter the "Library File Manager", and find "Resource Library" → "Install" → Library file" need to be installed→" Open" to complete the installation of the library file.



03-Task Configuration and Priority

Task management can be carried out in the task configuration of the tree menu. A new standard PLC project will automatically generate a cyclically executed task, which is automatically associated with PLC_PRG. The default task period is 4ms and the priority is 1. The PLC program is only tasked by the task. The call will only participate in the compilation and actual execution. Right-click Task Configuration → Add Object → Task, define the task name and complete the creation of a new task. You can create up to 100 tasks of different types, and execute them according to the priority order set by the user. The smaller the number, the higher the priority. In the same case, execute from top to bottom in the order in which the tasks are configured.





Application training

-

04- Variable Data Type

To create a better life through our work

04-Variable Data Type

Q1-Supported data types (following IEC61131-3 specification)

Boolean variable

Boolean variables take values TRUE and FALSE
Reserve 8 bits of storage space, usually prefixed with x or b

Integer data

BYTE,WORD,DWORD,SINT,USINT,INT,UINT,DINT,UDINT,LINT,ULINT are all integer data types. Each of the different data types contains a different set of values. The following table lists the range for each integer data

Type	Lower limit	Upper limit	Memory space	Prefix
BYTE（byte）	0	255	8 bits	n
WORD（word）	0	65535	16 bits（2 bytes）	n
DWORD（double word）	0	4294967295	32 bits（4 bytes）	n
SINT（short integer）	-128	127	8 bits	n
USINT（short integer without sign）	0	255	8 bits	n
INT（Integer）	-32768	32767	16 bits	n
UINT（Integer without sign）	0	65535	16 bits	n
DINT（Double integer）	-2147483648	2147483647	32 bits	n
UDINT（Double integer without sign）	0	4294967295	32 bits	n
LINT（Long integer）	-2^{63}	-2^{63}-1	64 bits	n
ULINT（Long integer without sign）	0	$2^{64}-1$	64 bits	n

04-Variable Data Type

Date and time data types

CODESYS provides date and time data types by default, as shown in the following table:

Type	Lower limit	Upper limit	Memory space	Prefix
TIME_OF_DAY	TOD#0:0:0	TOD#23:59:59	32-bit	tod
DATE	D#1970-01-01	2106-02-07	32-bit	date
DATE_AND_TIME	DT#1970-01-01-00:00:00	DT#2106-02-07-06:28:15	32-bit	dt
TIME	T#0S	T#49D17H47S295MS	32-bit	tim/t

REAL/LREAL（Floating point）

REAL and LREAL are floating-point data types that refer to rational numbers. Among them, REAL occupies 32 bits of storage space, and LREAL occupies 64 bits. prefixed with f

REAL storage range： 1.175494351e-38F ~ 3.402823466e+38F

LREAL storage range： 2.2250738585072014e-308 ~ 1.7976931348623158e+308

Type	Lower limit	Upper limit	Memory space	Prefix
REAL	1.175494351e-38F	3.402823466e+38F	32bits	f
LREAL	2.2250738585072014e-308	1.7976931348623158e+308	64bits	f

04-Variable Data Type

STRING（character string)

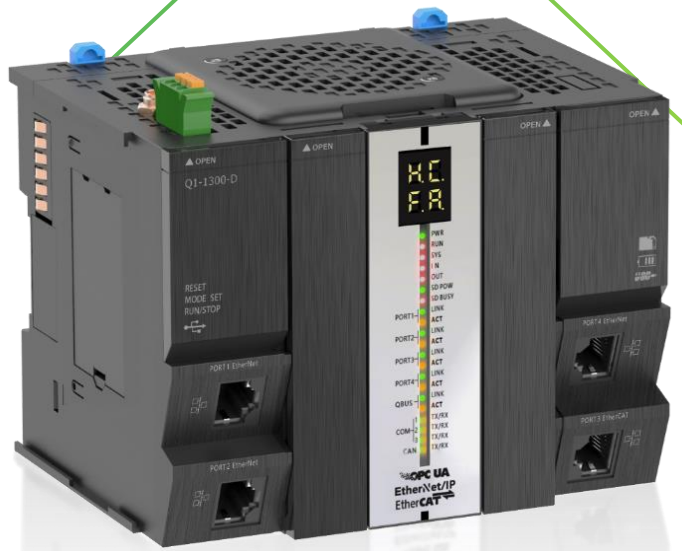
String type variables can contain any string of characters. The size of the variable at the time of declaration determines how much storage space is reserved for the variable, which involves the number of characters in the string and can fit inside parentheses and square brackets. If the size specification of the variable is not given, the default size is 80 characters. In IEC, the string type data is \0 terminated, so the actual position length of the string in the memory needs to be calculated as the string length + 1.

Basically, in CODESYS, there is no limit to the length of the string, but the string function can only handle 1-255 characters.

WSTRING（character string)

Since String can only handle characters in the ASCII code table, if the user needs to use the character type in the Unicode encoding, he needs to choose to use Wstring to implement it. A character occupies two bytes in Wstring.

Type	Memory space	Declaration example	SIZEOF result	LEN result	prefix
STRING	Default: 80, max. 255 (defined by the string processing function of STRING)	sVar: string; sVar: string (1) ; sVar: string (255) : = 'ABC' ;	81 2 256	3 1 3	s
WSTRING	Undefined	wsVar:wstring:= "Welcome" ;	6	4	ws



Application training

-

05-Naming Rule for Variables

To create a better life through our work

05-Naming Rule for Variables

Naming rule for Variables in IEC needs to follow the following rules:

1. The first character of a variable can be a letter (abcd....) or an underscore (_)

2. It can be followed by numbers (123...), letters (abc...) and underscores (_)

3. Variables are not case-sensitive (abc and ABC represent the same variable)

4. No special characters in variable （E.g.: ！，＂，$ etc)

5. No space （a　b）and continuous underscore （a____b） in variable

Keywords in IEC61131-3 cannot be used as variable names.

For example:

Logical operation keyword : AND, OR, NOT...

Data type keyword : BOOL, INT, REAL...

Type and Structure Definition Keywords : TYPE, STRUCT

Block or program keyword : FUNCTION, FUNCTION_BLOCK, PROGRAM

cannot be the same as the function block name

Keywords are automatically capitalized in blue in the program

05-Naming Rule for Variables

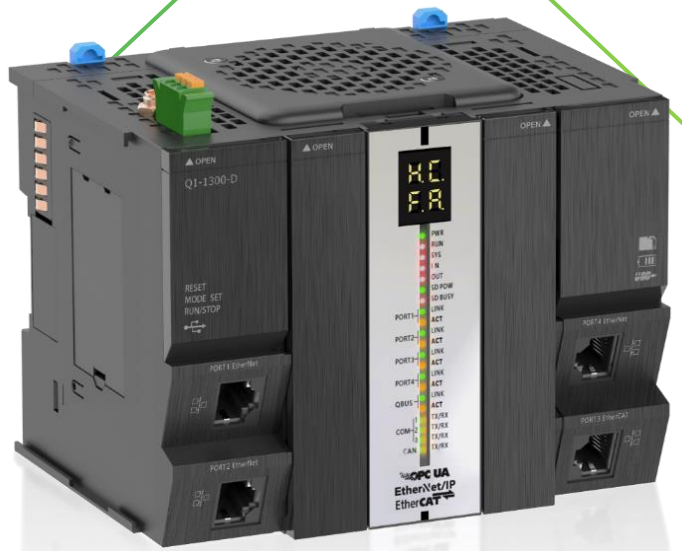
Naming rule（ no mandatory requirement ）

Range+Attribute
_Function Description

Range	Abbr.	Example
Input variable	i	Var_INput
Output variable	o	Var_output
Global variable	g	
Hardware input	I	
Hardware output	Q	
HMI commu.	h	

Attributes	Abbr.	Example
Floating points No.	f/r	LREAL/REAL
Integer	d/i	DINT/INT/SINT
Word	w	DWORD/WORD
Byte	by	BYTE
Word	b	BOOL
Time	T	TIME

Functions	Length	Example
Level 1 (optional)	4	For classification; b_main axis_jog button
Level 2	8	Functions: hf_jogging speed; gf_main axis_jogging speed



Application training

-

06-Simple Programming

To create a better life through our work

06-Simple Programming

Example 1 Q1 Platform - Square Wave Program with Adjustable Duty Cycle

①.Creation of new projects

②. POU creation

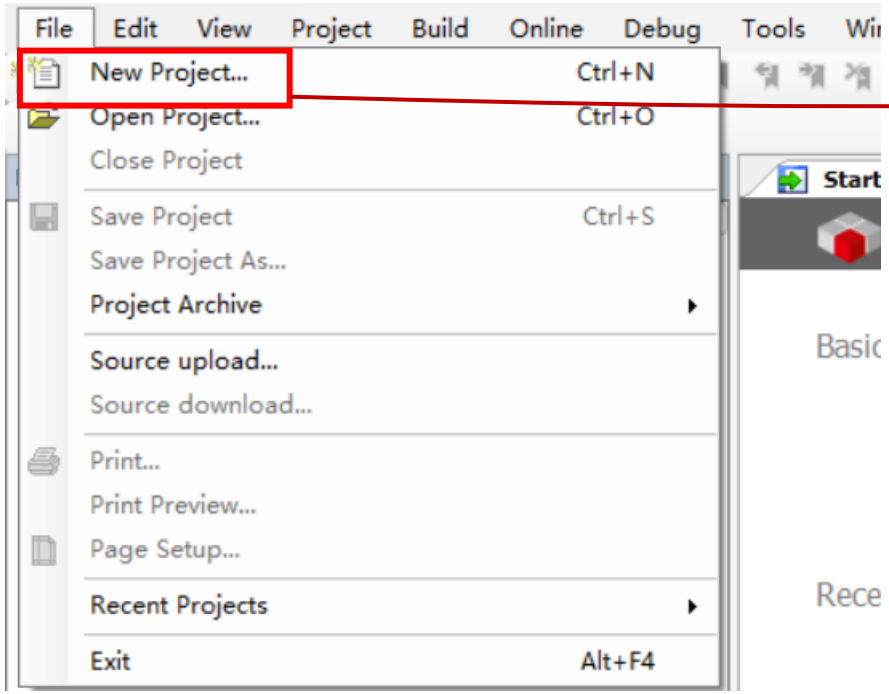
③. Variable declaration

④. Programming

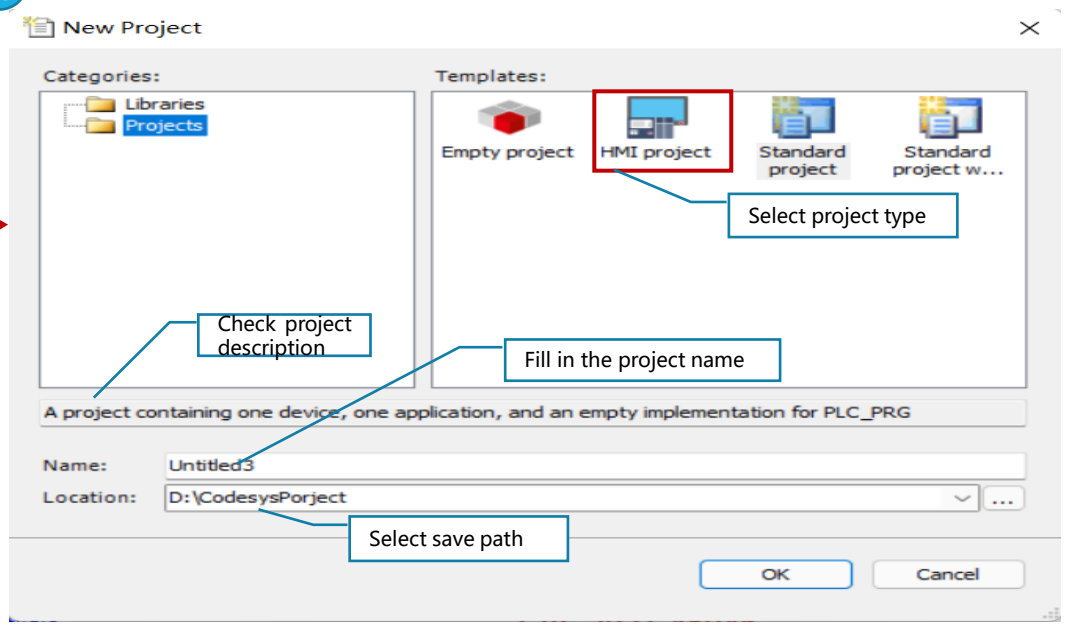
⑤. Use of the debug function

06-Simple Programming – New Projects

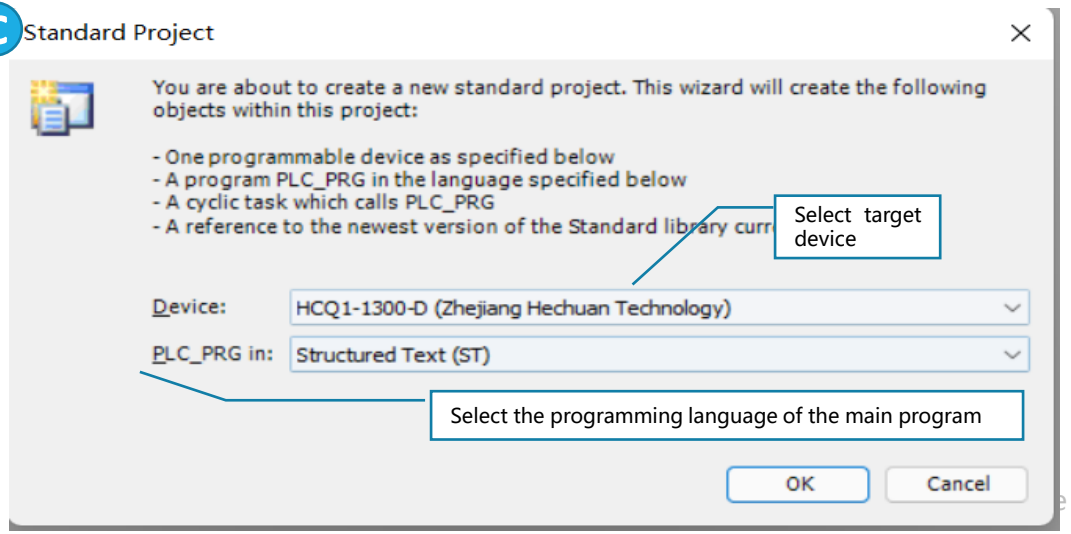
a



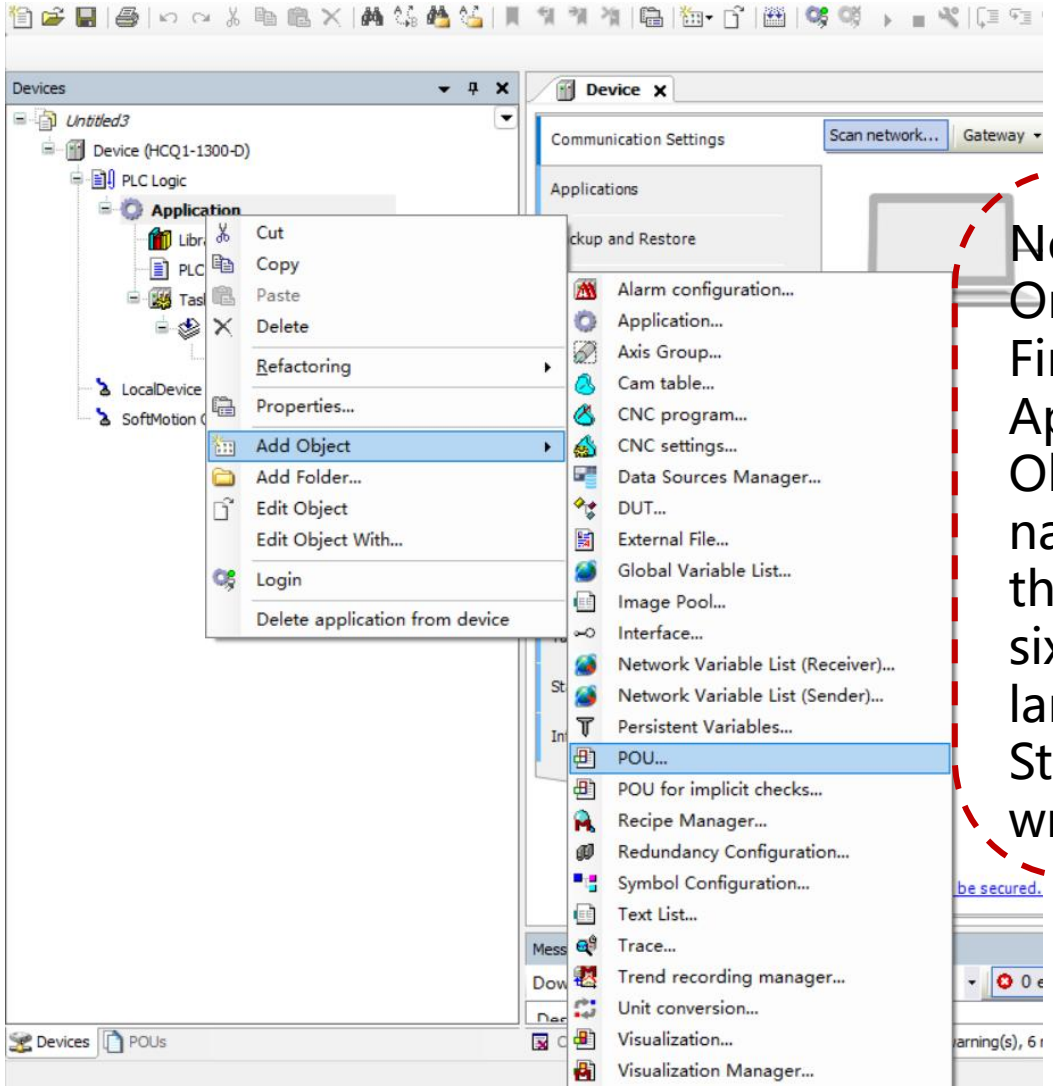
b



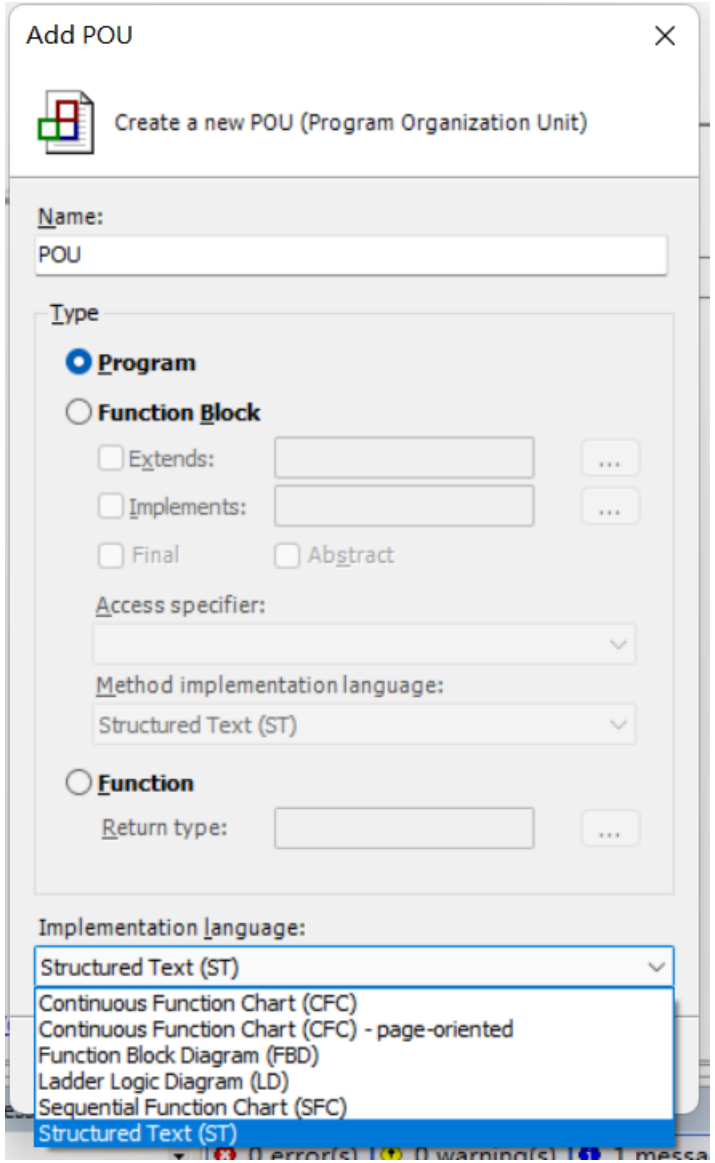
c



06-Simple Programming – POU Creation



New POU（Program Organization Unit），First, right-click Application→Add Object→POU, fill in the name of the POU, select the block type, support six programming languages, and select Structured Text (ST) to write a sample program.



06-Simple Programming – Program Example

Declare the following variables in the variable

PLC_PRG

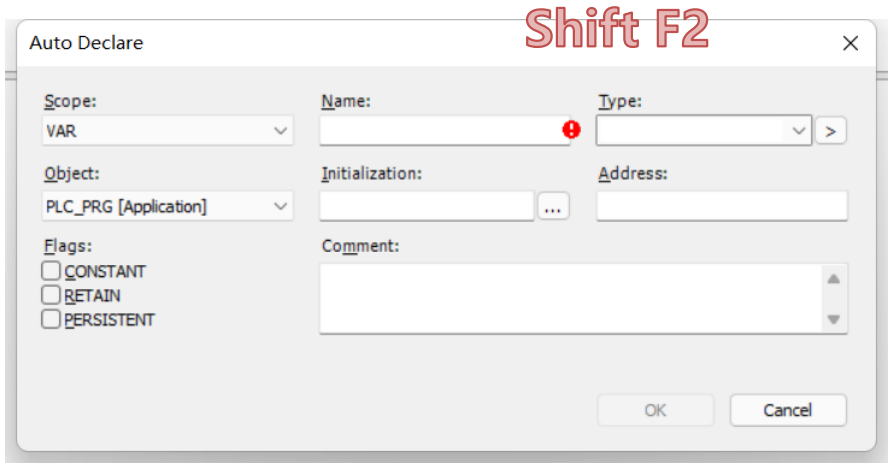
```
1  PROGRAM PLC_PRG
2  VAR
3      fbTimer1        :TON;//定时器
4      index           :INT:=10;
5      iVar            :INT;//中间变量
6      iRectangle      :INT; //方波信号，幅值可调
7      iRatio1         :INT:=5;//"1"信号占比
8      iRatio2         :INT:=5;//"0"信号占比
9  END_VAR
```

Variable declaration statement

Comments

Write a program in the main program window

```
1  (*示例程序是一个可调占空比的方波，通过给定常数Ratio1和Ratio2可以设置方波iRectangle的"1""0"的周期长度和所占比例*)
2      fbTimer1(IN:=NOT fbTimer1.Q, PT:=T#500MS  );
3  CASE index OF
4      10:
5      irectangle:=1;
6      IF iVar<iRatio1 AND fbTimer1.Q THEN
7          iVar:=iVar+1;
8      ELSIF iVar>iRatio1 OR iVar=iRatio1 THEN
9          index:=20;
10         iVar:=0;
11     END_IF
12
13     20:
14     irectangle:=0;
15     IF iVar<iRatio2 AND fbTimer1.Q THEN
16         iVar:=iVar+1;
17     ELSIF iVar>iRatio2 OR iVar=iRatio2 THEN
18         index:=10;
19         iVar:=0;
20     END_IF
21 END_CASE
```

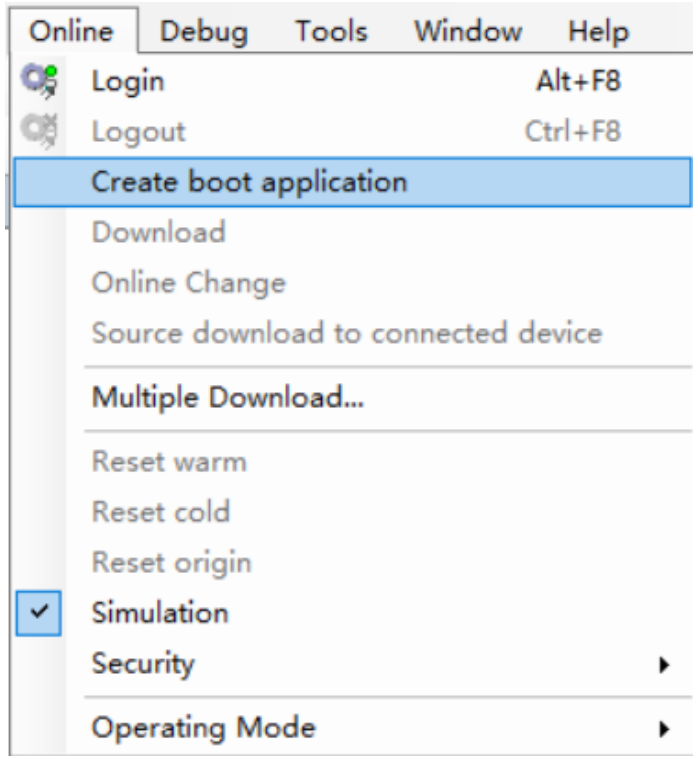


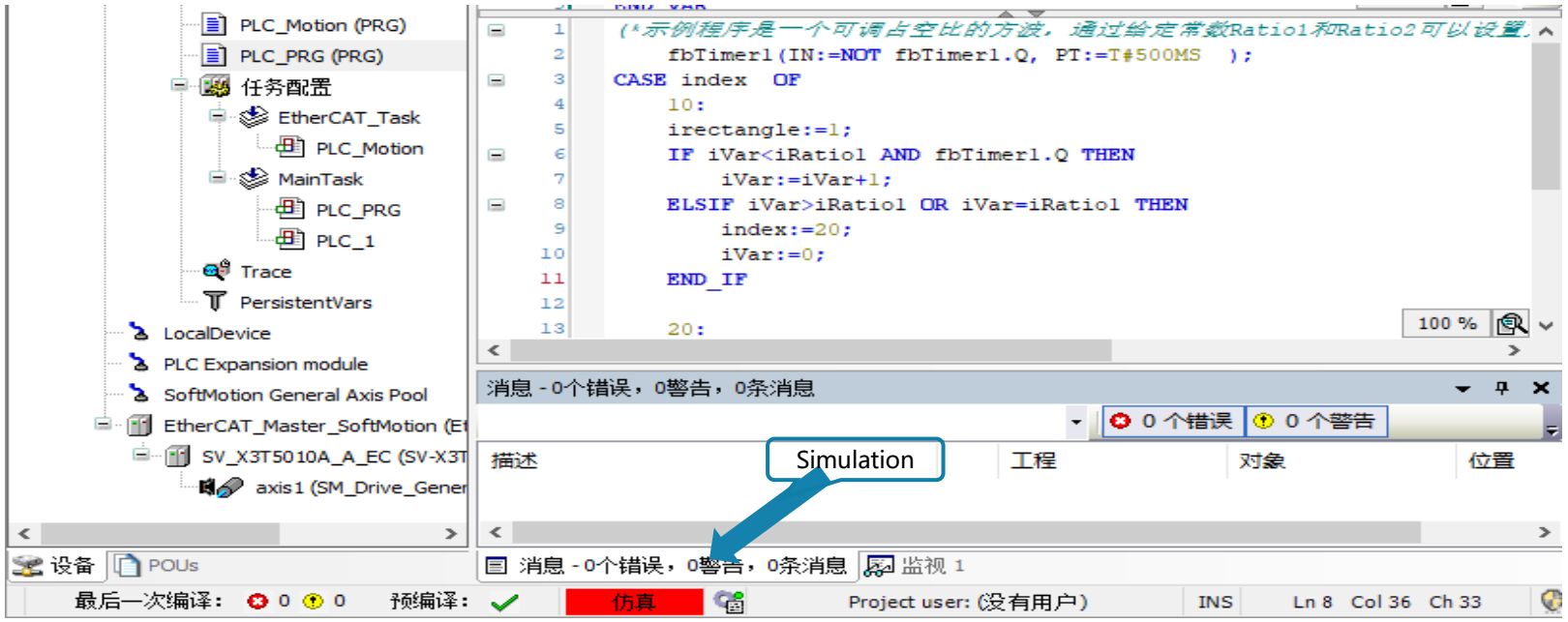
Scope: variable scope
Name: variable name
Type: Variable data type
Object: The application where the variable is located
Initialization: variable initial value
Address: Mapping relationship between variables and external hardware points
Flags: Variable types can be set to constant, held and persistent
Comments: Comments in the format (*comment content*)

06-Simple Programming - Simulation Function

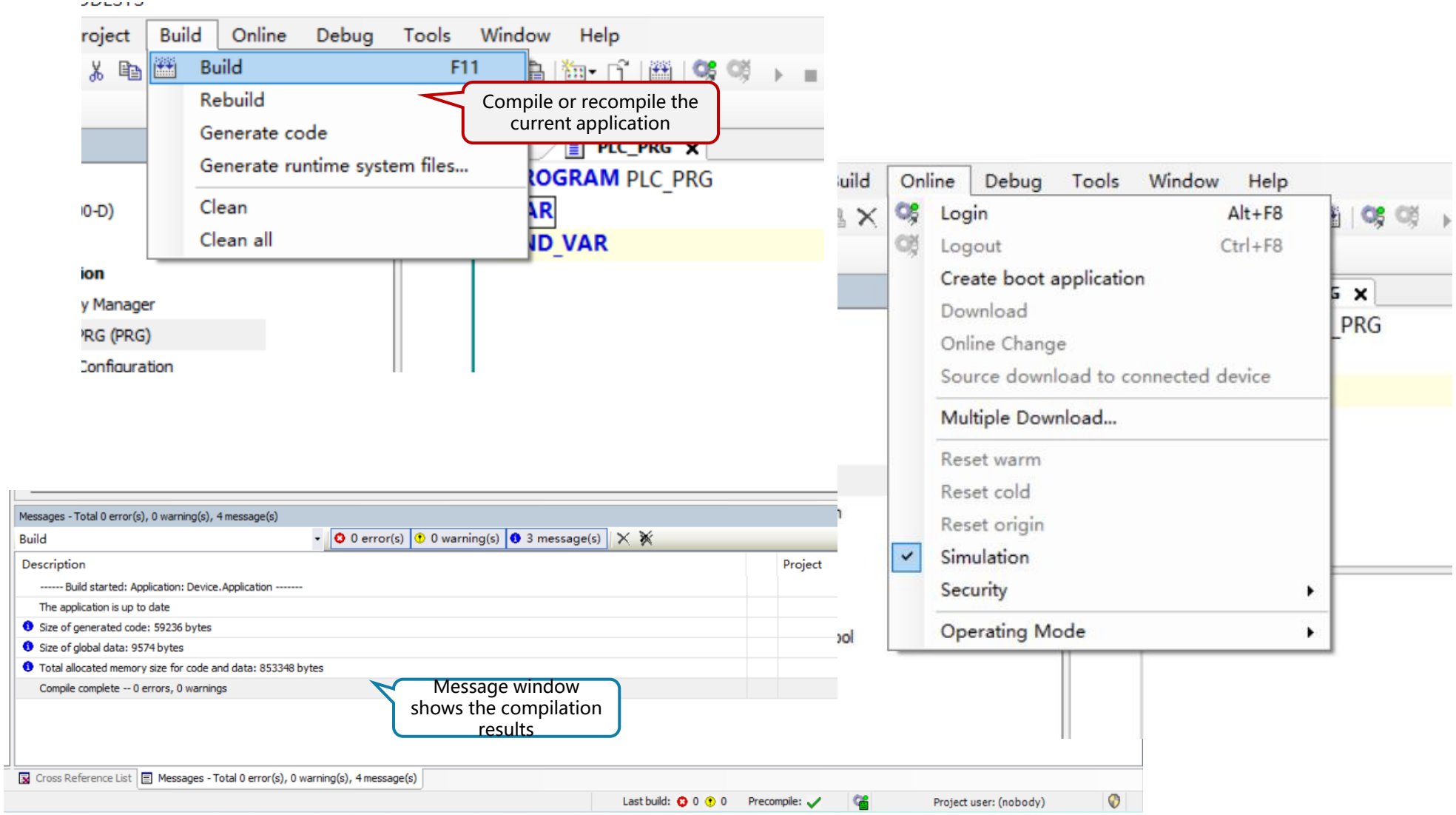
In addition to downloading the program directly to the PLC for online operation and debugging, CODESYS also provides online simulation function without additional hardware

If the gateway is not turned on, find "Online" in the menu bar and select "Simulation" to enter the simulation mode. After checking it, will appear on the left side of the simulation. Follow the steps to log in and run the program directly to perform program simulation debugging.

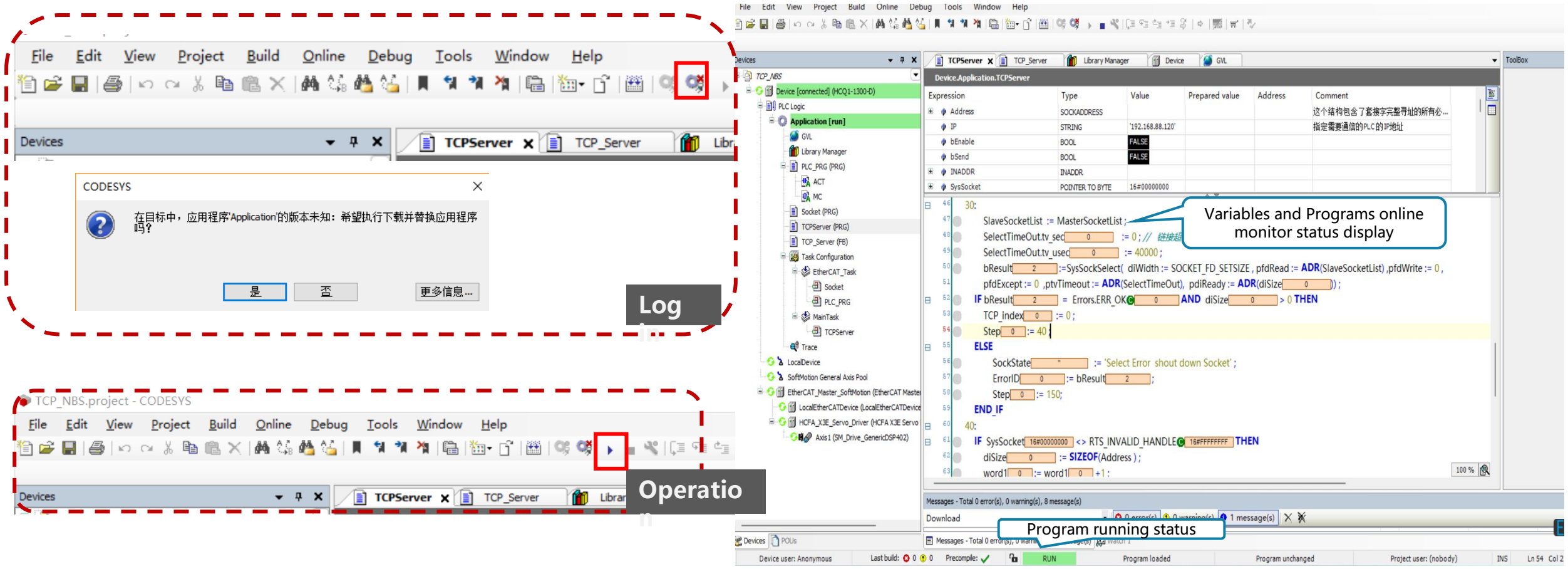


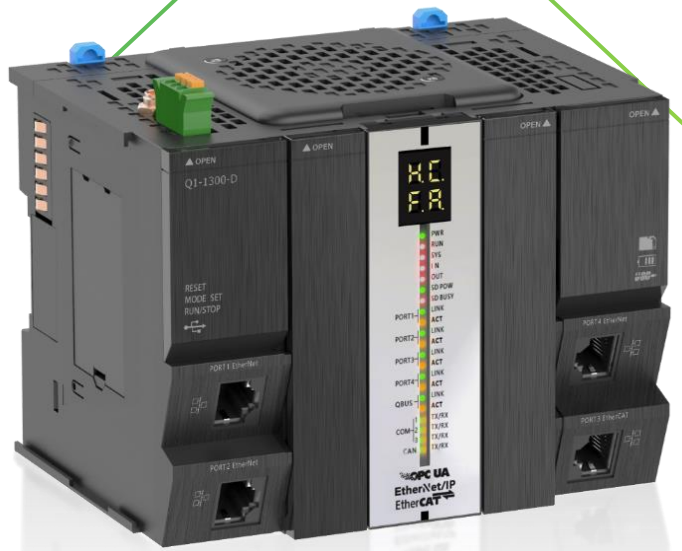


06-Simple Programming - Program Compilation and Download



06-Simple Programming - Program Compilation and Download





Application training

-

07-POU Type Introduction

To create a better life through our work

07-POU Type Introduction - Introduction to Block Types

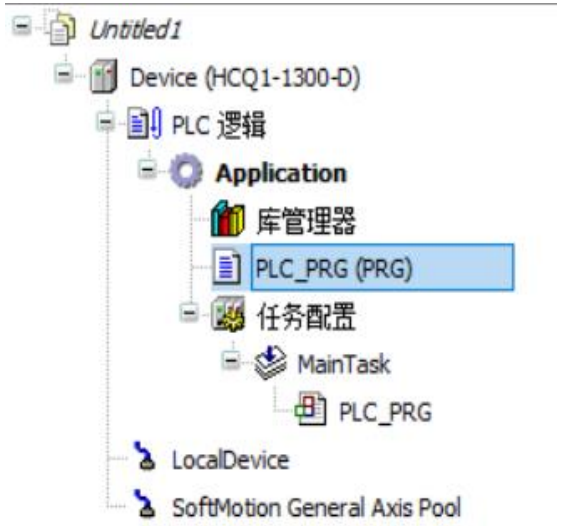
Three types of blocks are included in the POU (PROGRAM ORGANISATION UNIT) of IEC61131-3:

- Program
- Function Block
- Function

07-POU Type Introduction - PRG

Program： PRG（program）

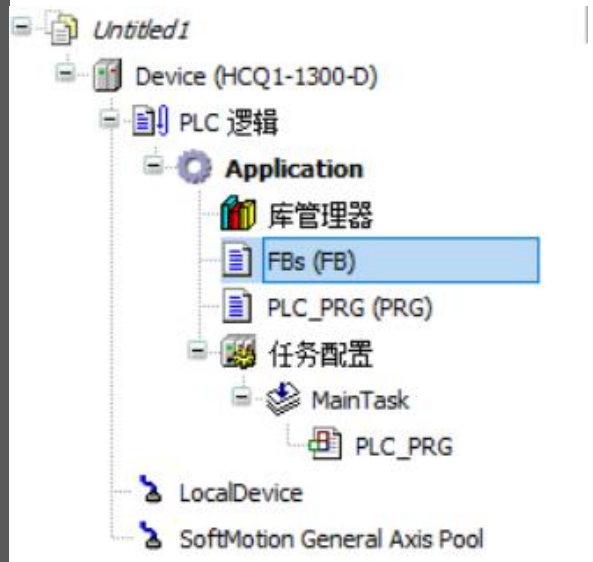
- Called by tasks (a program can call another program)
- Can be called: FBs, functions, (programs)
- Local variables: static, that is, the data of local variables can be used in the next cycle.
- Input: By default, there is no variable of input type, and input variables can be defined in VAR_INPUT.
- Output: The default is no output variable, and the output volume can be defined in VAR_OUTPUT
- Input and output variables can be defined in VAR_IN_OUT.
- Monitor: The value of the variable in the online control state is visible in real time.



07-POU Type Introduction - FB

Function block： FB

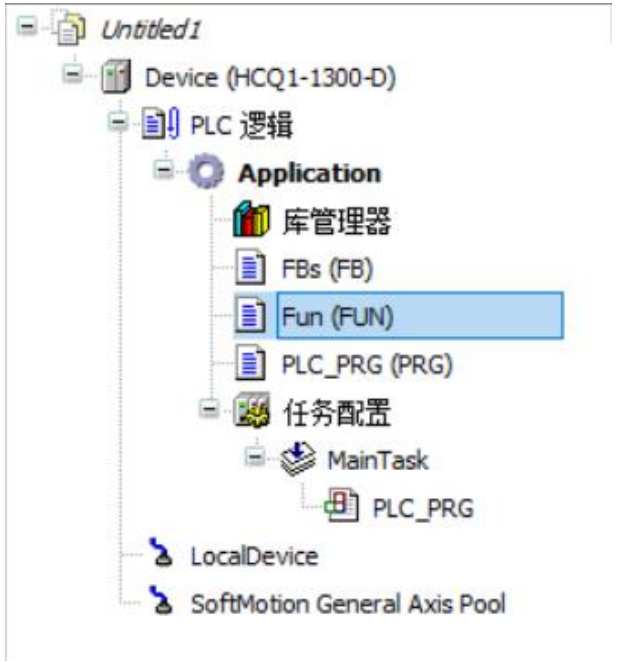
- Called by programs or other FBs
- Can be called: FBs, functions,
- Local variables: Static, that is, local variable data can be used in the next running cycle. In multiple function blocks, each FB has its own local variables.
- Input: 0,1,2,3…VAR_INPUT
- Output: 0,1,2,3.. VAR_OUTPUT
- Input/output: 0,1,2,3.. VAR_IN_OUT
- Monitor:Local variables are visible for each specified function block under online control.

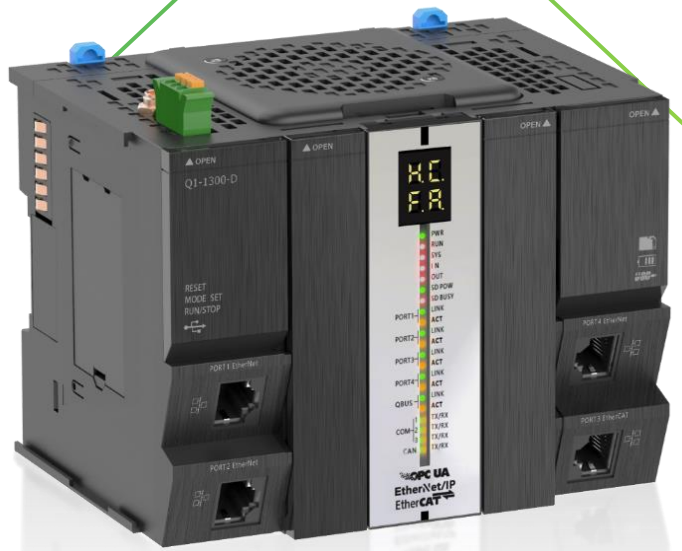


07-POU Type Introduction - FC

Function : FC

- **Called by programs, function blocks and other functions**
- **Can be called: Functions**
- **Local variables: Temporary, i.e. local variable data is only available when the current function executes. This data area is then used for other functions.**
- **Input: 1,2,3..... VAR_INPUT**
- **Output: Only one!**
- **Input/output: 1,2,3..... VAR_IN_OUT ,**
- **Monitor: Only "???" can be seen when online. Use breakpoints.**





Application training

-

08-Memory Addressing

To create a better life through our work

08-Memory Addressing- Memory Definition for Q1-seires

Classified by memory storage medium:

Flash、DDR、SDRAM。。。

Classified by usage:

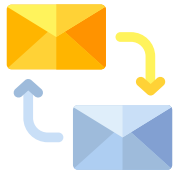
System code memory
System operating memory
User program storage
User program running data buffer
User variable storage area

The above-mentioned memory areas are invisible and inoperable for users. Only the "User variable storage area" is directly related to users. In this section, we only introduce the rules for its use.

08-Memory Addressing- Memory Definition for Q1-seires

Q1-Composition of system memory:
Input I, Output Q, Internal M

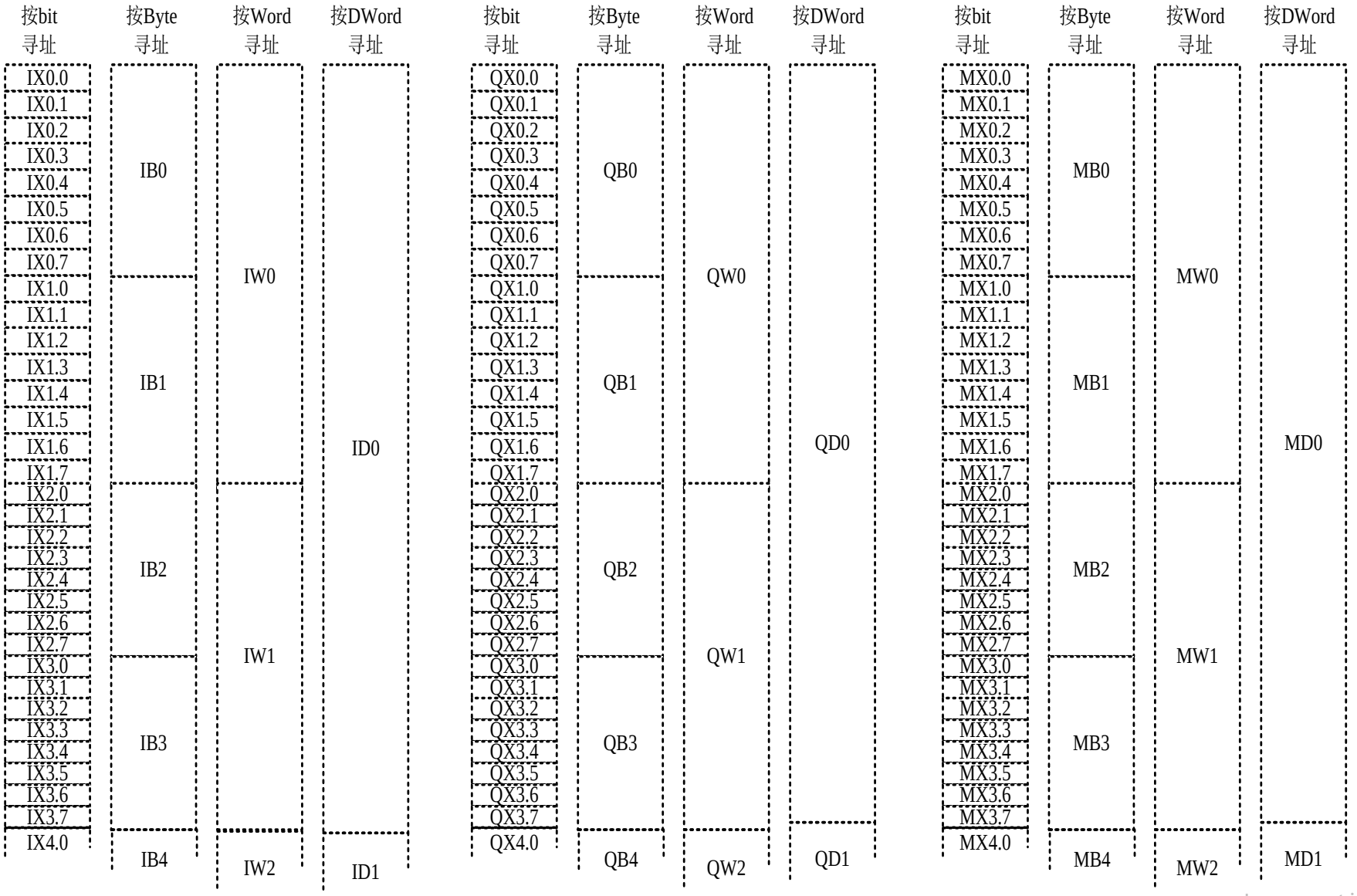
Zone	Usage	Size	Address range
I-zone (%I)	For user	64KWords	%IW0-%IW65535
Q-zone (%Q)	For user	64KWords	%QW0-%QW65535
M-zone (%M)	For user	240KWords	%MW0-%MW245759
	Persistent variable area	800KBytes	Inaccessible for users
	Retain variable area	2768Bytes	Inaccessible for users



- The starting address of the Word-type register follows that the starting address is an even-numbered Byte address; The Byte is the two times of Word addressing index numbers
- The starting address of the DWord type register follows that the starting address is an even word address; The word is two times of Dword addressing index numbers

Even alignment rules for address calculation

08-Memory Addressing- Rules Examples

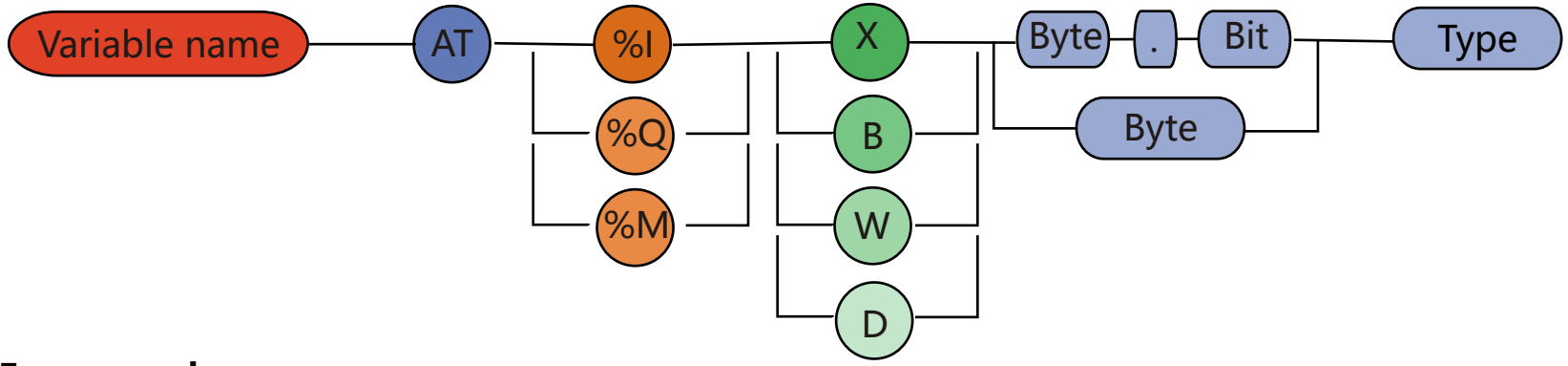


08-Memory Addressing- Variable Address Specification

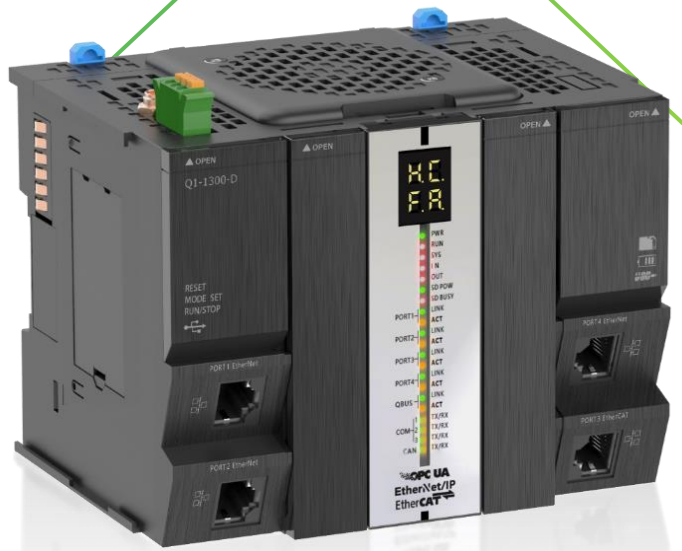
Variable names can be connected with addresses

For determining input and output fixed address variables, you can use AT %I or AT %Q or AT %M to declare the variable type.

For variables whose full address is known:



For example:
w marquee AT %MW10 :WORD;
b start AT %MX0.0 :BOOL;



Application training

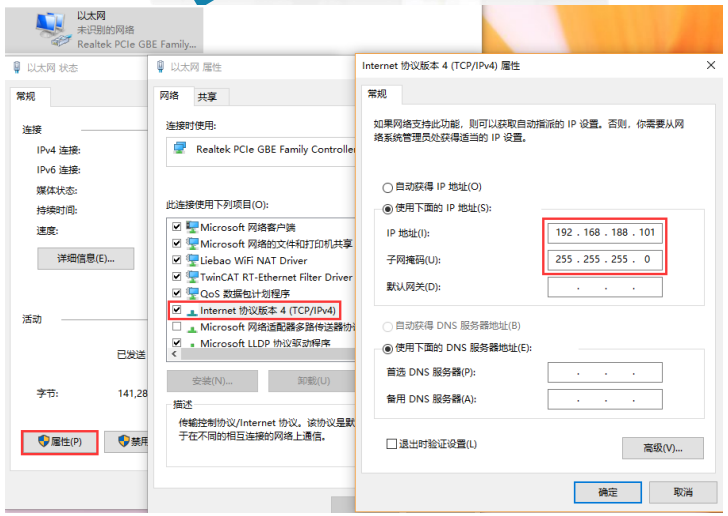
-

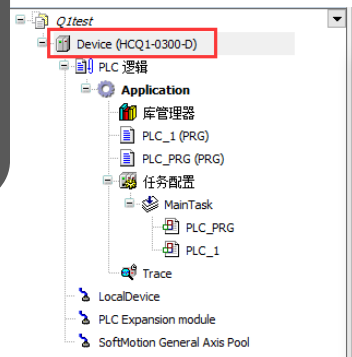
09-PAC Connection

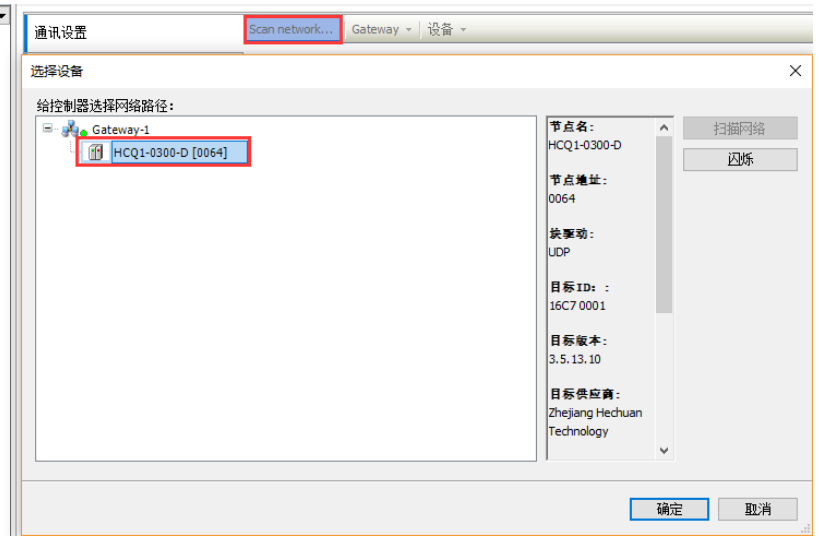
To create a better life through our work

09-PAC Connection- Network Settings

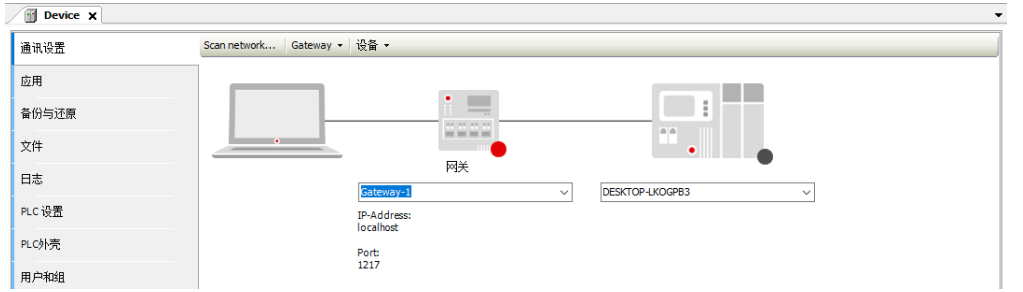
Modify the IP address to the same network segment
PLC LAN1: 192.168.188.100(default)
PLC LAN2: 192.168.88.100(default)



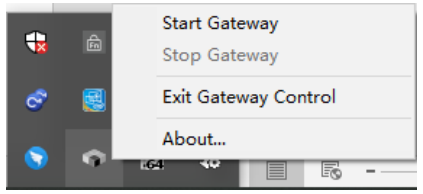


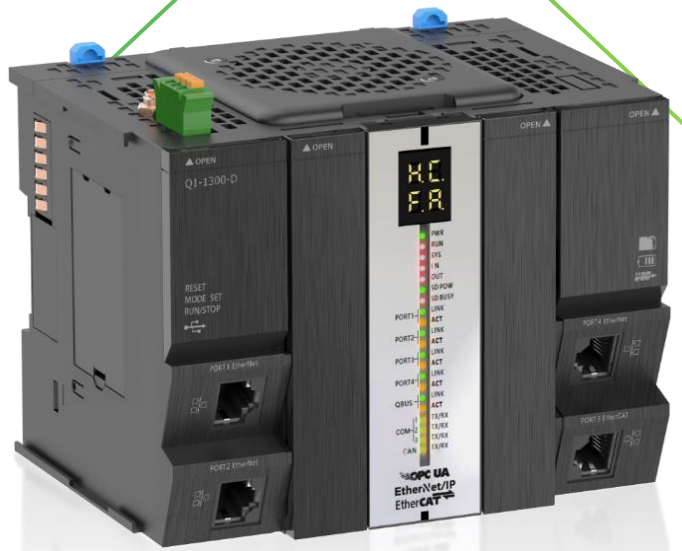


Gateway not turned on



Find the CODESYS on the PC, right-click, select "StartGateway", start the gateway, then perform scanning and adding





Application training

-

10-ST Language Introduction

To create a better life through our work

10-ST Language Introduction

Structured text(ST)

Structured Text is a textual high-level language, similar to PASCAL or C. Program code consists of instructions, and instructions consist of keywords and expressions. Unlike IL languages, ST statement loops can contain numerous statements, thus allowing the development of complex structures.

For example:

```
IF value < 7 THEN
  WHILE value < 8 DO
        value := value +1;
  END_WHILE;
END_IF;
```

Operator

When evaluating an expression, the operands are evaluated in order of operator precedence, from high to low.;

Operators with equal precedence shall be processed in the left-to-right order as written in the expression

For example:

If A, B, C, and D have values 1, 2, 3, and 4, respectively, then

Expression formula： A+B-C*ABS(D)　　should be equal to -9,

Expression formula： (A+B-C)*ABS(D)　　should be equal to 0

10-ST Language Introduction- ST Operator

Operation	Symbol	Priority
Brackets	(Expression formula)	Highest
Function call	Function name (parameter list, separated by commas)	
Exponentiation	EXPT	
Negative value To complement	- NOT	
Multiply Divide Remainder	* / MOD	
Plus Subtract	+ -	
Compare	<,>,<=,>=	
Be equal Not equal	= <>	
Logical and	AND	
Logical XOR	XOR	
Logical or	OR	Lowest

10-ST Language Introduction- ST Instructions and Keywords

The ST program consists of instructions, which are separated by a semicolon ";". These directives consist of keywords and expressions

Keyword	Description
:=, S=, R=	Assign, set, reset
	Function block calls and outputs
RETURN	Return (exit the current POU)
IF	Select
CASE	Multiple selection
FOR	FOR cycle
WHILE	WHILE cycle
REPEAT	REPEAT cycle
EXIT	Exit the cycle
CONTINUE	Continue the cycle for the next execution
JMP	Jump
;	Empty statement

10-ST Language Introduction- Common Operators and Comparisons

Operation	ST language	C language	Priority
Brackets	(Expression formula)	(Expression formula)	Highest
Function call	Function name (parameter list, separated by commas)	Function name (parameter list, separated by commas))	
Exponentiation	EXPT	POW	
Negative value Logical negation Bitwise negation	- NOT NOT	- ! ~	
Multiply, divide, take remainder	*、/、MOD	*、/、%	
Plus, minus	+、-	+、-	
Compare	<,>,<=,>=	<,>,<=,>=	
Equal to , not equal to	=、<>	==、!=	
Logical and Bitwise and	AND AND	&& &	
Exclusive OR	XOR	^	
Logical or Bitwise or	OR OR	|| |	Lowest

10-ST Language Introduction- ST Common Instructions – Assignment and Conditional Judgment

Instruction	ST language	C-language
Assignment operator	:=	=
Comments	Assign the value on the right to the variable on the left of the equal sign	

Instruction	ST language	C-language

(IF) conditional judgment execution statement	If (variable 1=variable 2) then
	Operation expression1;
	Operation expression2;
	ELSIF (Judge conditions 2) then
	Operation expression3;
	Operation expression4;
	Else
	Operation expression5;
	Operation expression6;
	END_IF
		If (variable 1==variable 2)
		{
		Operation expression1;
		Operation expression2;
		}
		Else if (Judge conditions 2)
		{
		Operation expression3;
		Operation expression4;
		}
		else
		{
		Operation expression5;
		Operation expression6;
		}

10-ST Language Introduction- ST Common Instructions - Cycle

Instruction	ST language	C-language

(FOR) cycle execution statement	FOR i:=initial value TO end value BY step size DO
	Operation expression1;
	Operation expression2;
	END_FOR
		FOR (i =initial value ; i<end value ; i++)
		{
		Operation expression1;
		Operation expression2;
		}

Comments		

(WHILE) cycle execution statement	WHILE (conditional expression) DO
	Operation expression1;
	Operation expression2;
	END_ WHILE
		WHILE (conditional expression)
		{
		Operation expression1;
		Operation expression2;
		}

Comments	The cycle executes the statement only when the condition is true	

(REPEAT) cycle execution statement	REPEAT
	Operation expression1;
	Operation expression2;
	UNTIL (conditional expression)
	END_ WHILE
		DO
		{
		Operation expression1;
		Operation expression2;
		}
		WHILE (conditional expression) ;

Comments	Execute first, then judge the cycle statement	

10-ST Language Introduction- ST Common Instructions - Branch Judgment

Instruction	ST language	C-language

（CASE）cycle execution statement	CASE （conditional expression） OF
	Constant1:
	
	Operation expression1;
	Operation expression2;
	
	Constant2:
	Operation expression3;
	Operation expression4;
	ELSE
	Operation expression5;
	END_ CASE
		SWITCH （conditional expression）
		{
		CASE constant 1:
		Operation expression1;
		Operation expression2;
		Break;
		CASE constant2:
		Operation expression3;
		Operation expression4;
		Break;
		Default： Operation expression5;
		}

Comments		

10-ST Language Introduction- ST Common Instructions - Jump out

Instruction	ST language	C-language
Jump out of the loop or conditional action	EXIT;	BREAK;

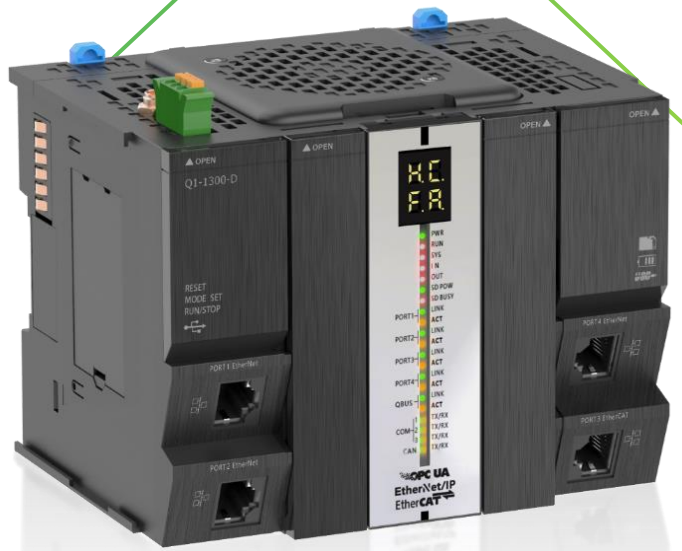
Instruction	ST language	C-language
Jump instruction	JMP;	Goto;
Comments	//jump instruction	

Instruction	ST language	C-language
(RETURN) Return instruction	RETURN;	return;
Comments	//jump out of the executor	

Instruction	ST language	C-language
Jump out of the loop or conditional action	continue;	continue;
Comments	//jump out of FOR、WHILE、REPEAT cycle	

10-ST Language Introduction- ST Common Instructions - Comments

	ST language	C-language
Comments	Single line comment: //Comments Multi-line comment: (* Comments *)	Single line comment: //Comments Multi-line comment: /* Comments */



Application training

-

11-Single-axis Motion Control

To create a better life through our work

11-Single-axis Motion Control– Common Instructions

1、The use of basic motion control instructions: (For details, please check the CODESYS help document first)

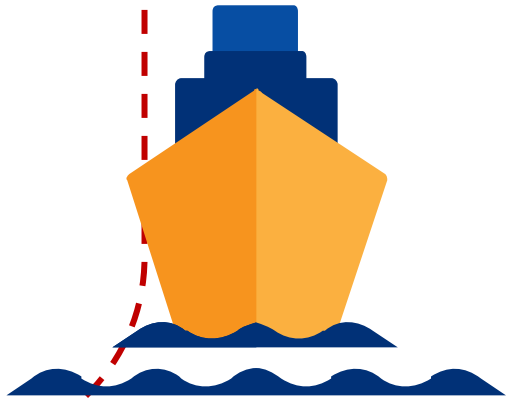
1.MC_Power
2.MC_Jog
3.MC_Stop
4.MC_Halt
5.MC_Reset
6.MC_ReadStatus
6.MC_MoveVelocity
7.MC_MoveAbsolute
8.MC_MoveRelative
9.MC_SetPostion

2、Use of Tracking Features:

Monitor the motion status of the axes with the tracking function

3、 Axis State Machine :

Determine the current state of the axis according to the axis state machine

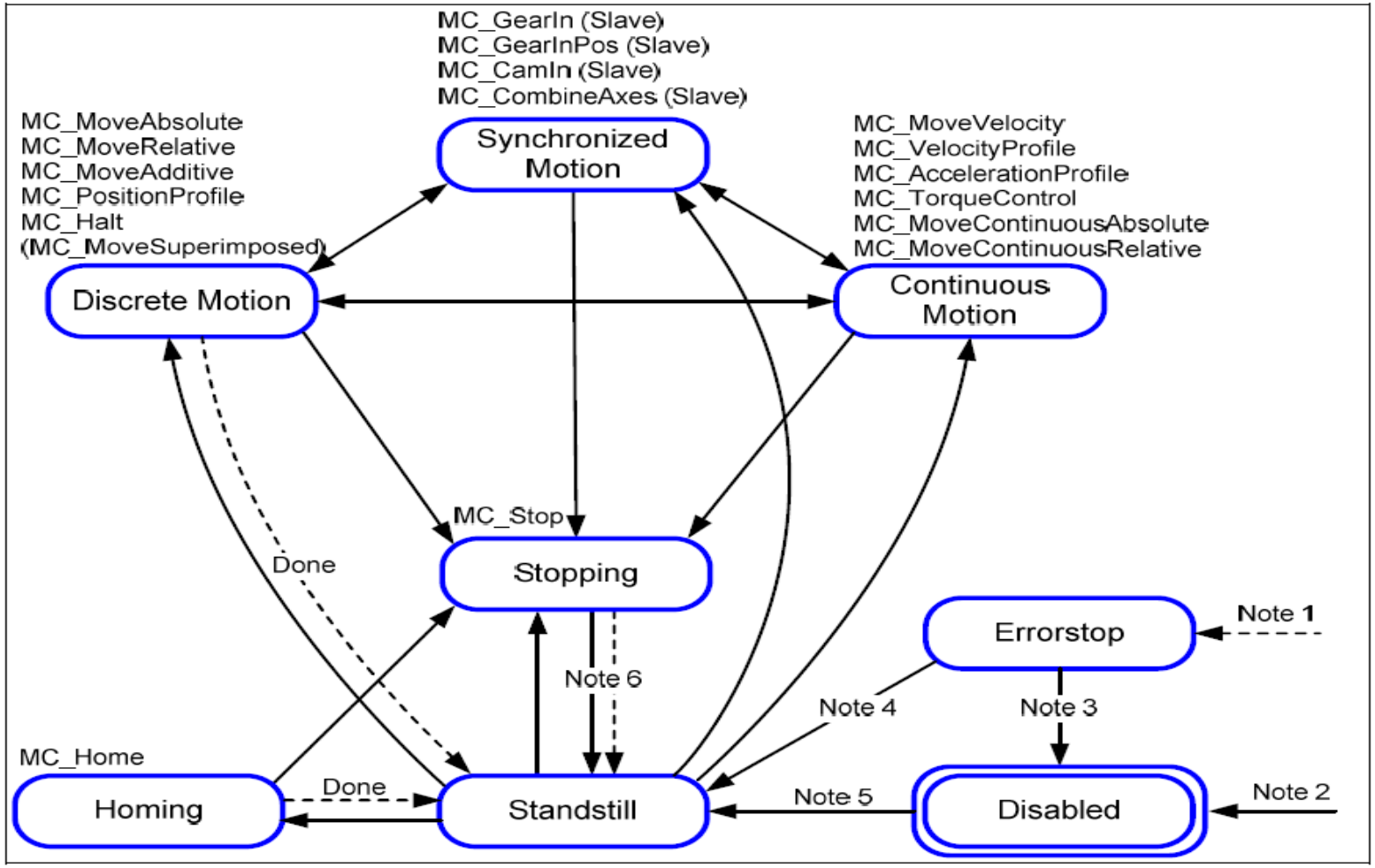


11-Single-axis Motion Control- State Machine

Axis data structure variable (Axis.nAxisState) to indicate the current running state of the axis, the variable Axis.nAxisState is an enumeration variable, and there are 8 possible states as follows:

0: Power_off (Disabled) : The axis is not powered on or enabled, the MC_Power instruction needs to be executed
1: Errorstop;---------- Execute the MC_Reset/MC_Power command first
2: Stopping;---------- Wait for the stop
3: Standstill;---------- Axis stopped, out of synchronization
4: Discrete_Motion;--------- Axis is in discrete operating state
5: Continuous_Motion;----- Axes are in continuous operation
6: Synchronized_Motion;--- Axes are in synchronous operation
7: Homing;---------- The axis is in zero return operation, waiting for the completion of the zero return operation.

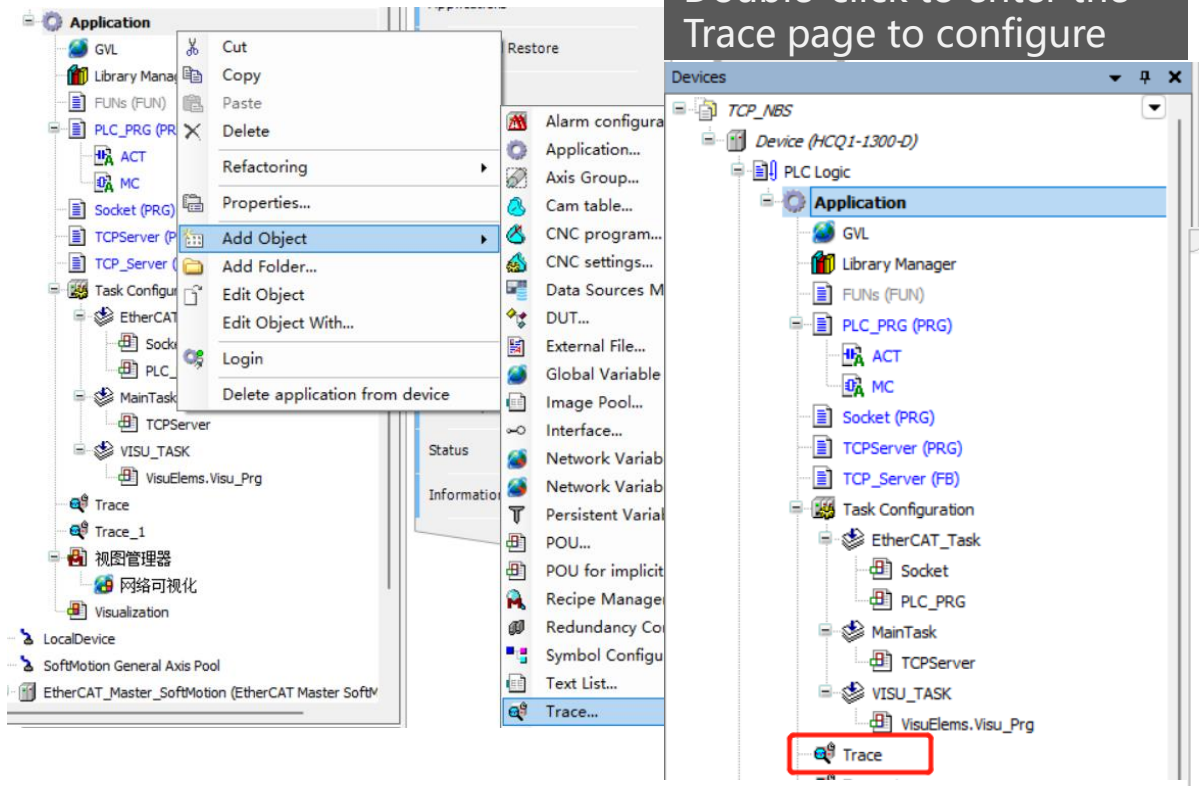
11-Single-axis Motion Control- State Machine Transition



Note 1:	From any state. An error in the axis occurred.
Note 2:	From any state. MC_Power.Enable = FALSE and there is no error in the axis.
Note 3:	MC_Reset AND MC_Power.Status = FALSE
Note 4:	MC_Reset AND MC_Power.Status = TRUE AND MC_Power.Enable = TRUE
Note 5:	MC_Power.Enable = TRUE AND MC_Power.Status = TRUE
Note 6:	MC_Stop.Done = TRUE AND MC_Stop.Execute = FALSE

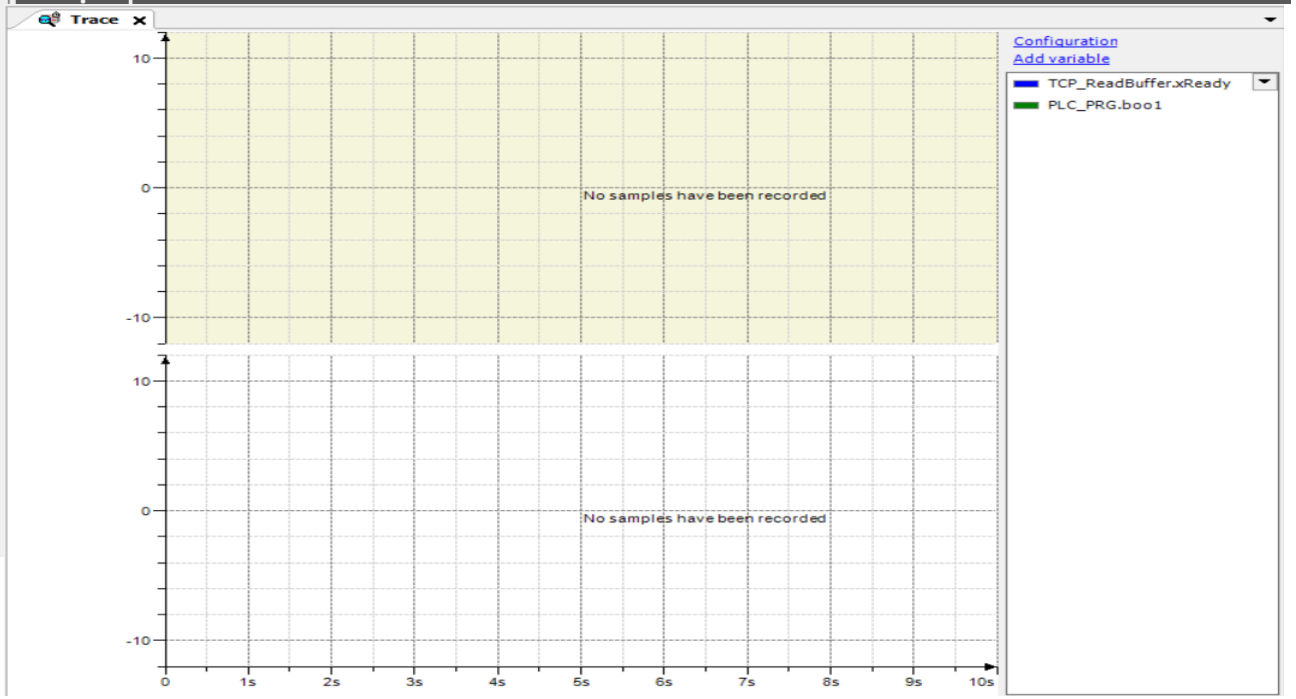
11-Single-axis Motion Control- Trace Online Monitoring

Right-click Application→Add Object→Trace. After successful addition, Trace will appear in the left tree menu



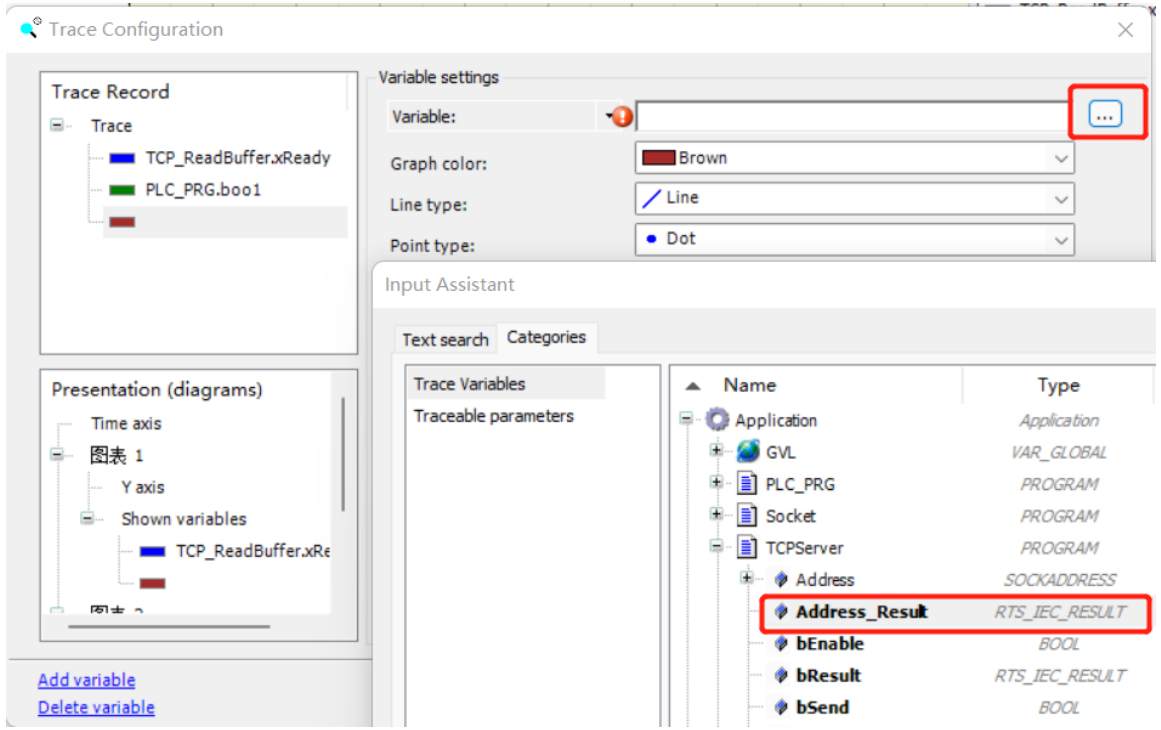
Double-click to enter the Trace page to configure

Click the configuration button in the upper right corner of the Trace interface to add and delete monitoring variables and set the sampling

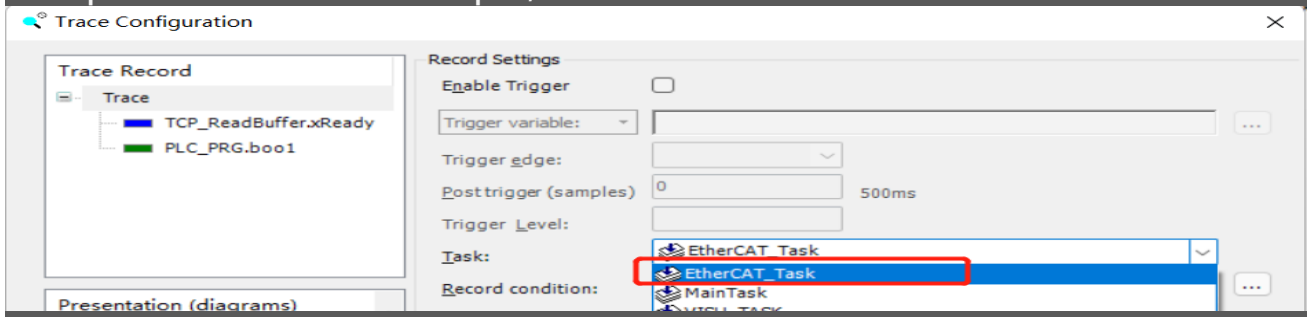


11-Single-axis Motion Control- Trace Online Monitoring

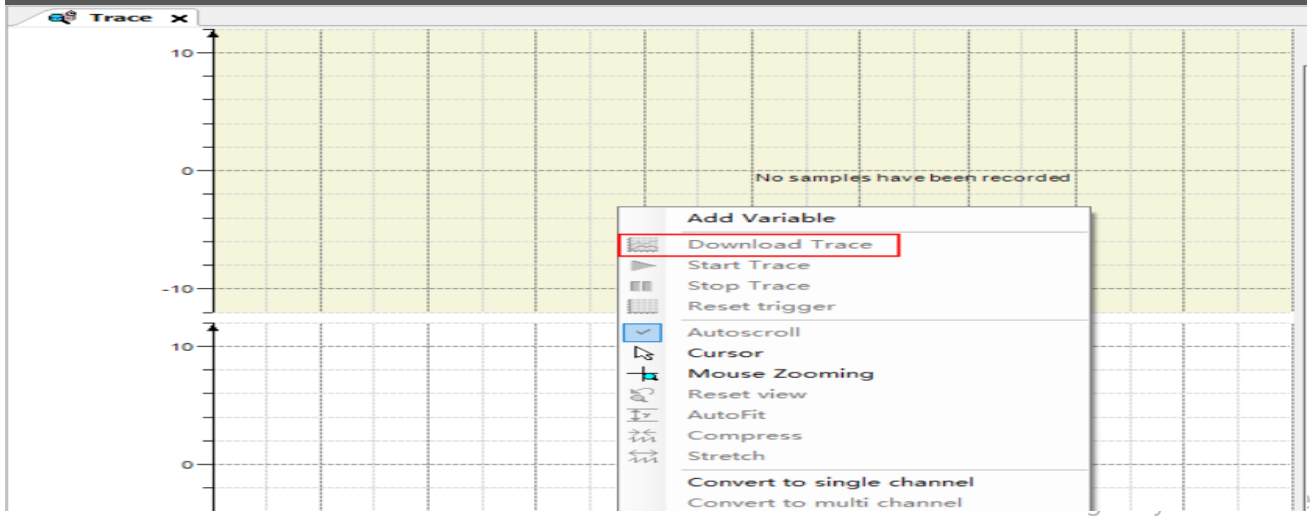
After opening the tracking configuration page, add and delete variables through the Add/Remove Variable button



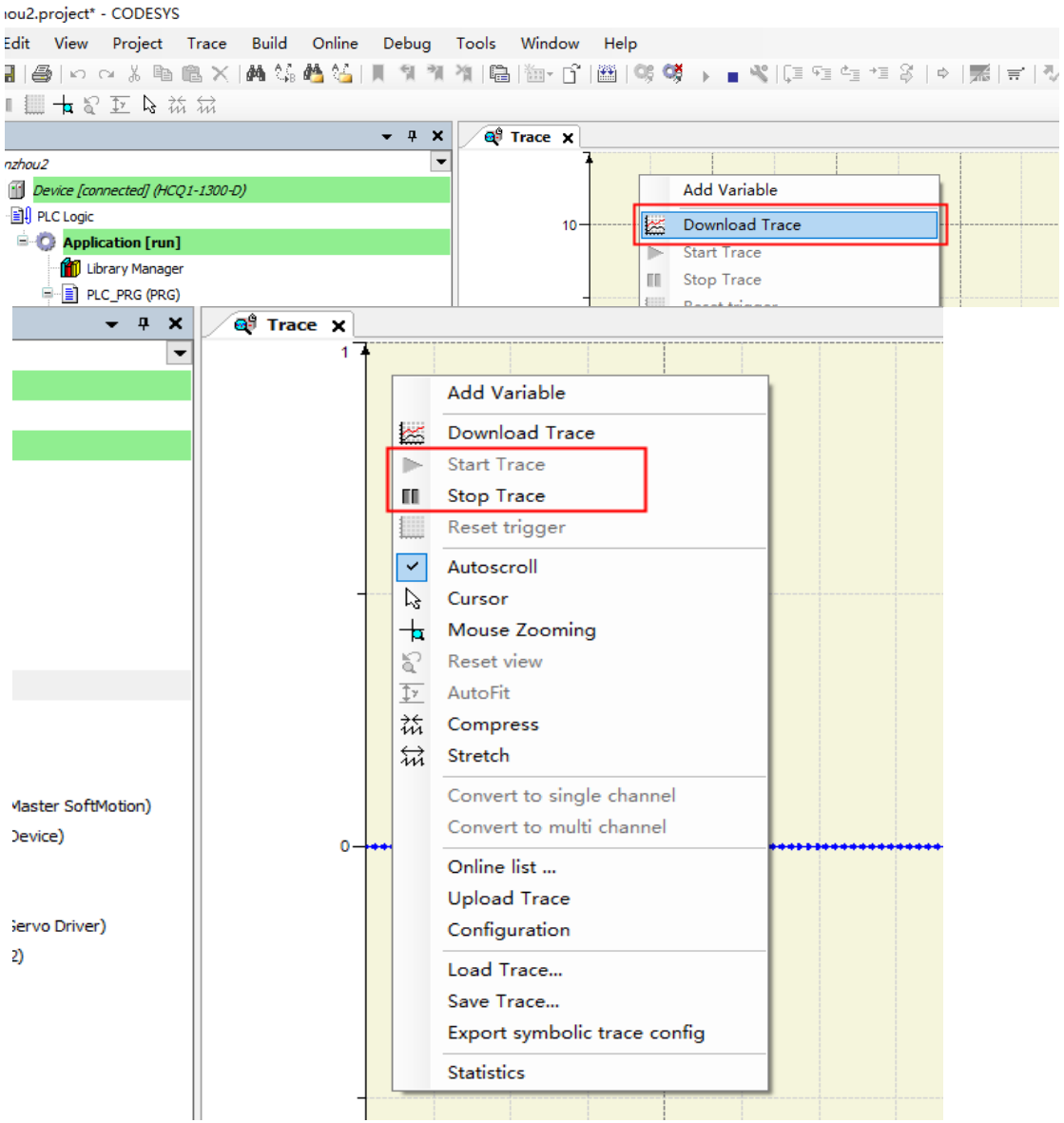
Configure the task followed by the sampling cycle by clicking the tree menu Trace on the left side of the dialog box, and select it from the drop-down list. For example, select MainTask



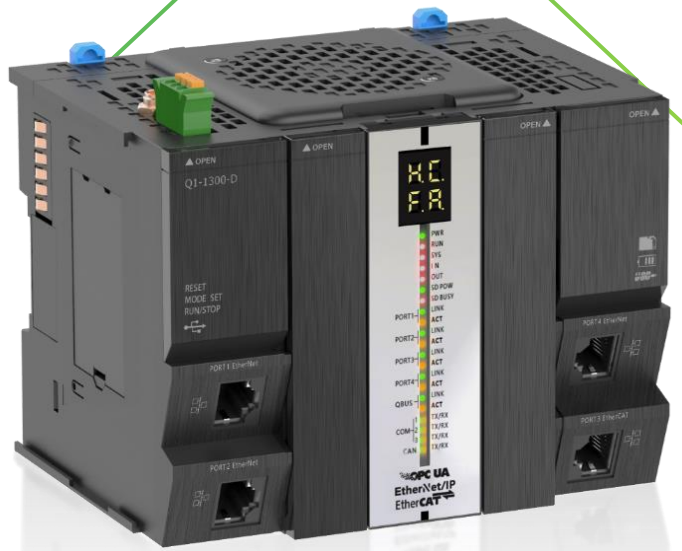
After completing the basic configuration of Trace, you need to log in the program and run it normally before you can download and start the trace (Start Trace) anywhere on the Trace interface and observe the running trace of the variables. Otherwise, this option is grayed out by default and cannot be selected.



11-Single-axis Motion Control- Trace Online Monitoring



Right-click anywhere on the Trace page to pause or start the trace, move the mouse to observe the historical trace, and scroll the wheel to zoom the trace waveform



Application training

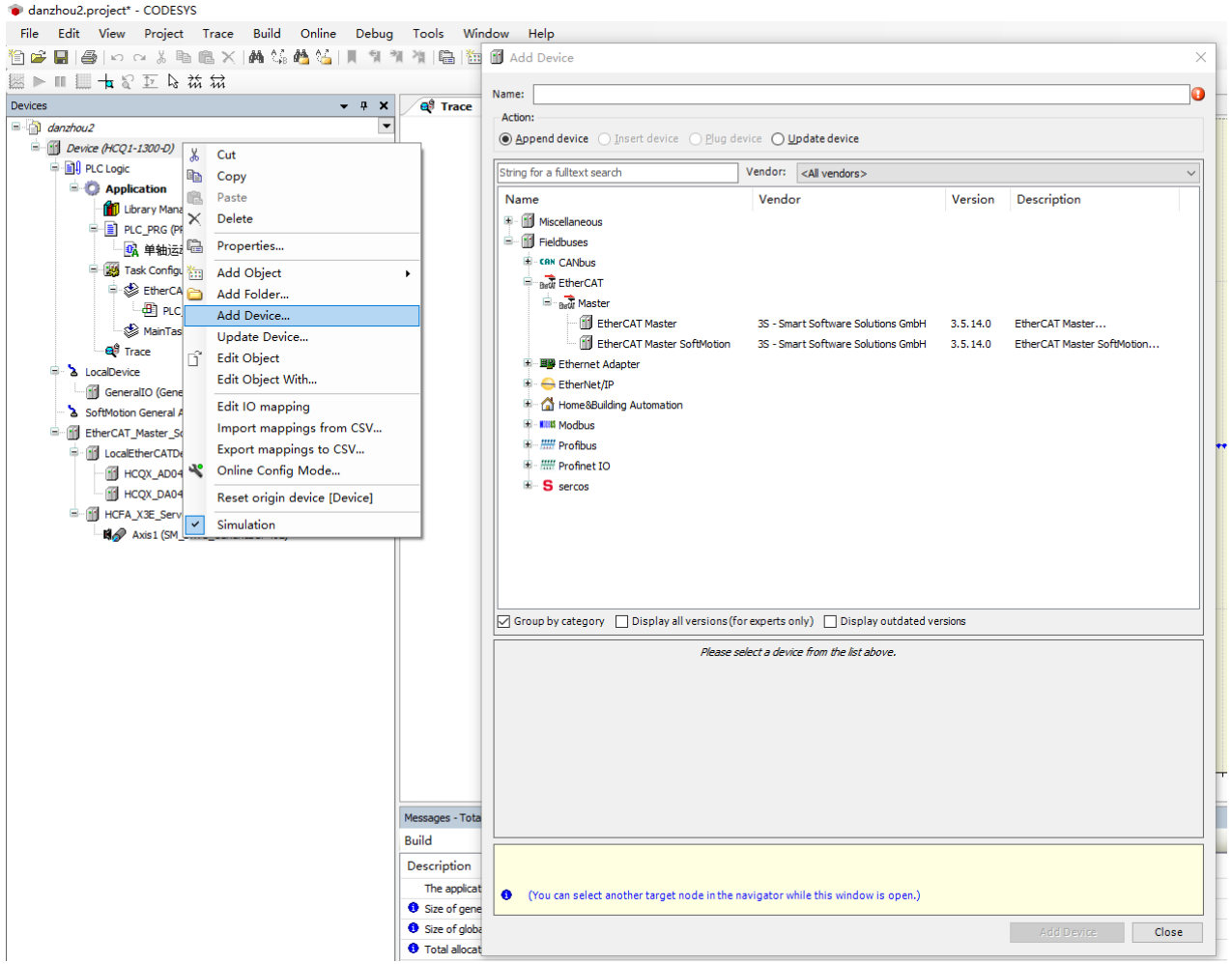
-

12- Connect HCFA Servos

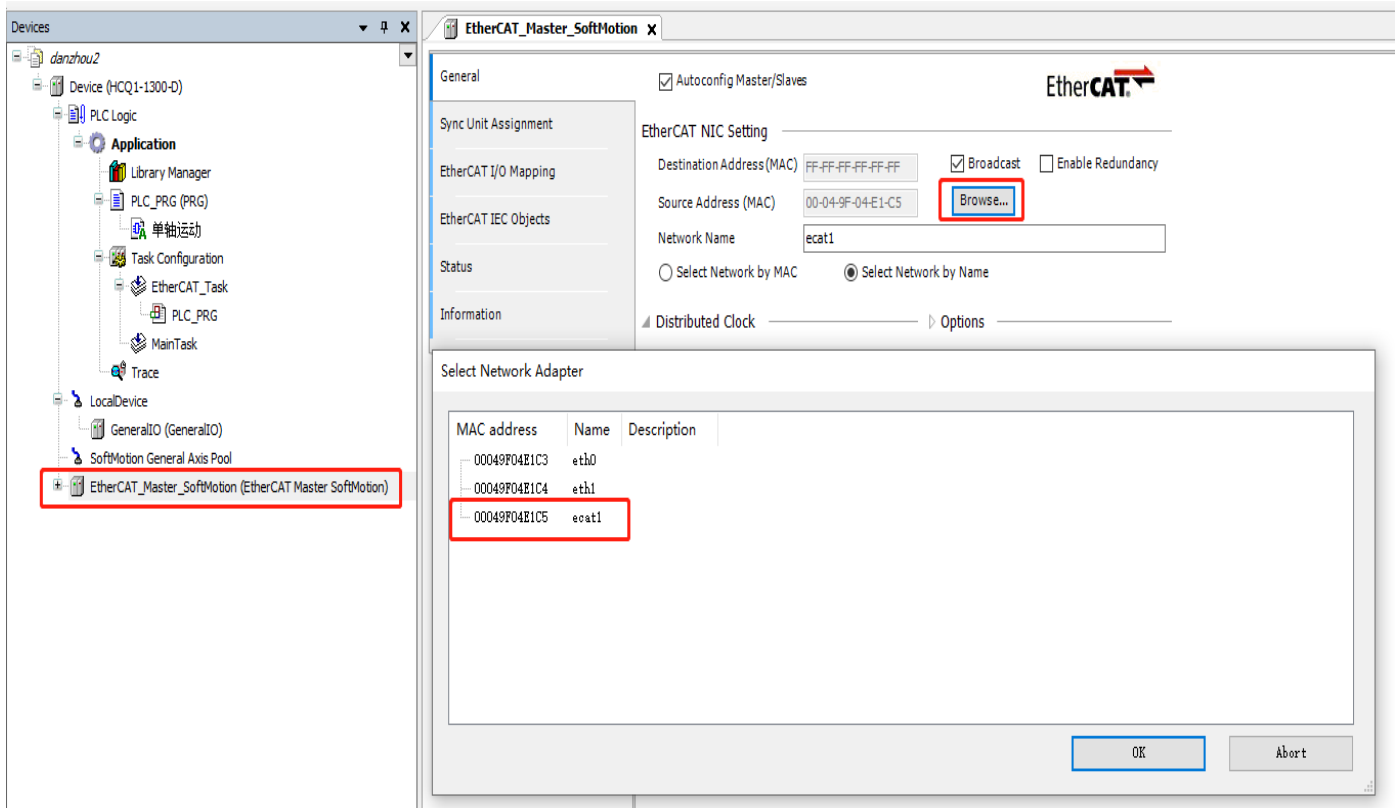
To create a better life through our work

12-Connect HCFA Servos

Add or scan the hardware, users can manually add the hardware they need for simulation or scan the actual hardware for debugging, take adding a device as an example

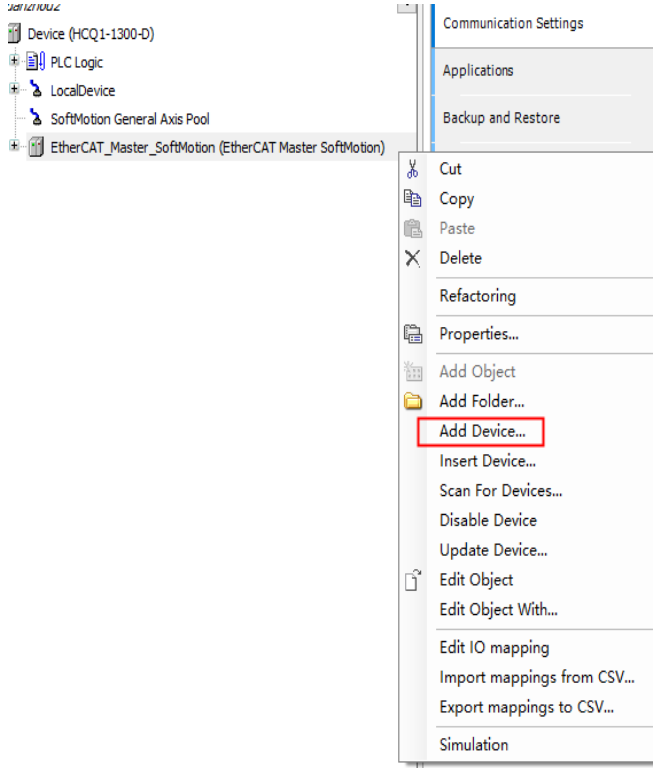


12-Connect HCFA Servos

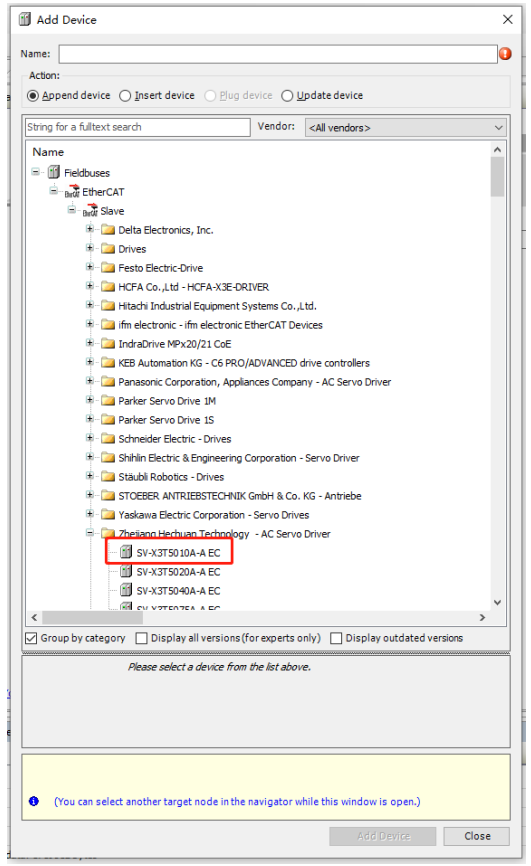


12-Connect HCFA Servos

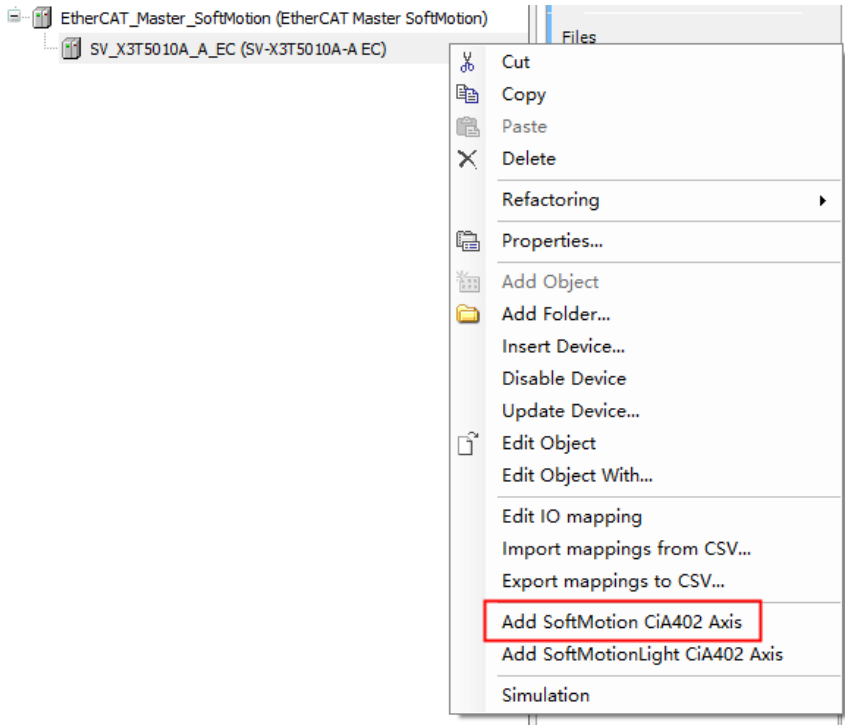
1 Continue to add X3T to the configuration interface



2

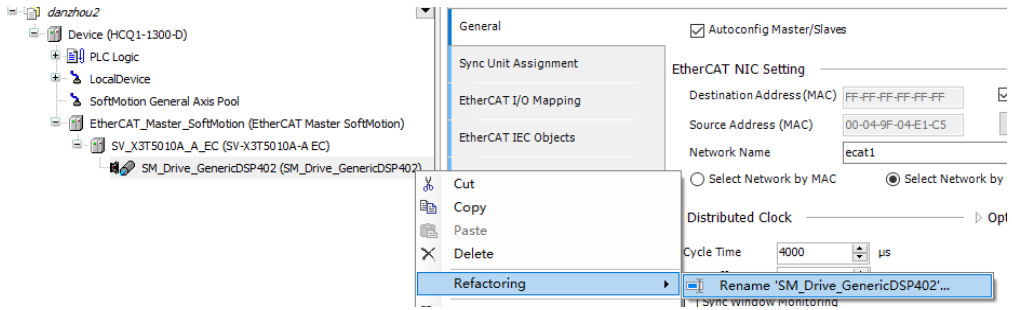


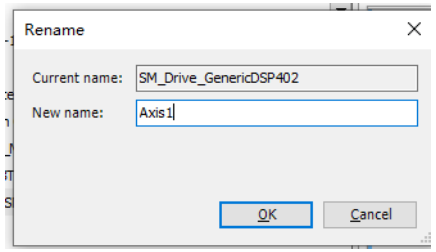
3

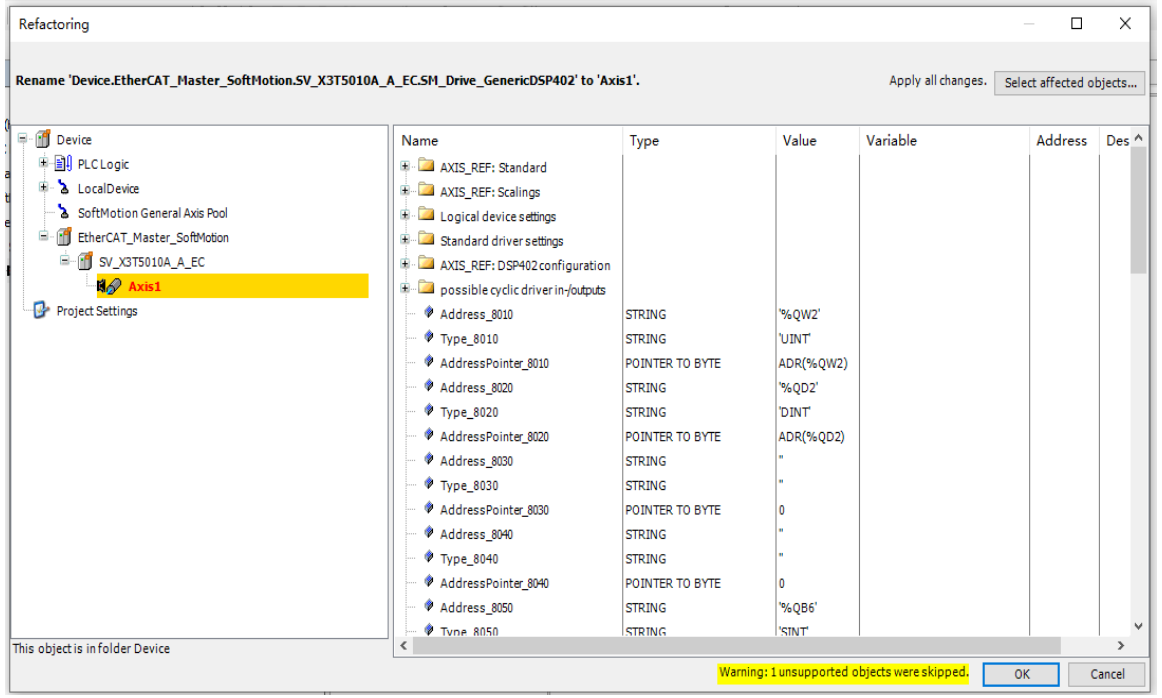


12-Connect HCFA Servos

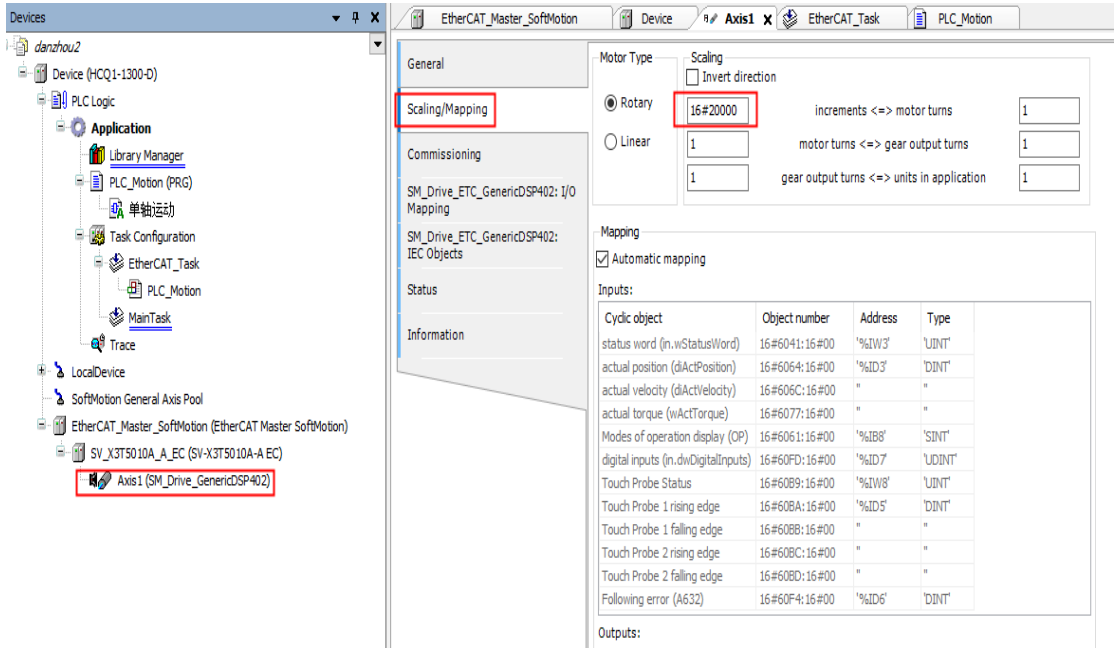
Rename the axis to complete the mapping with the axis variable in the program, and the added axis name can be used directly in the program





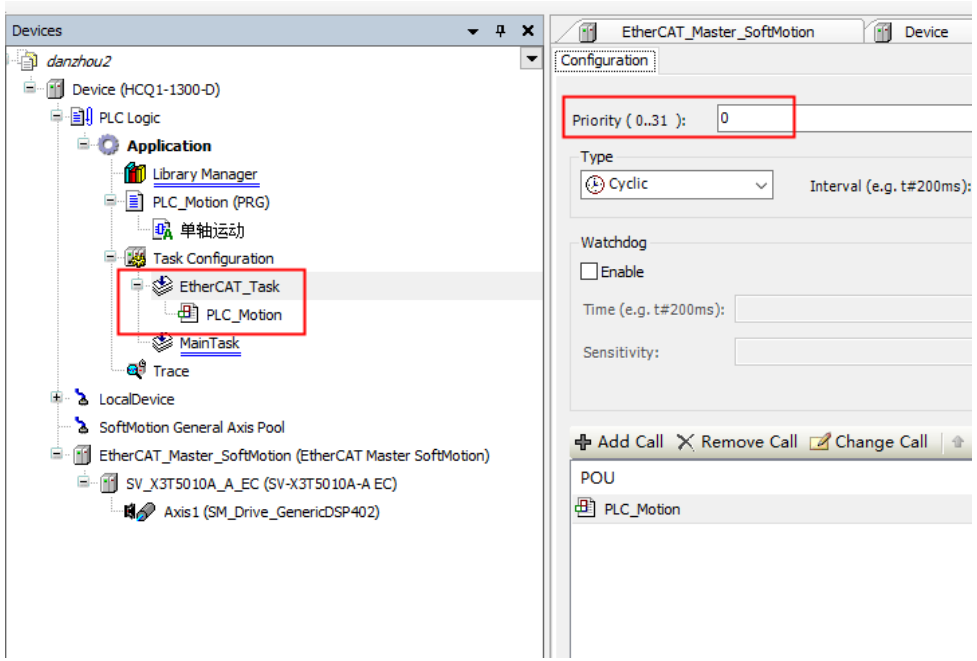


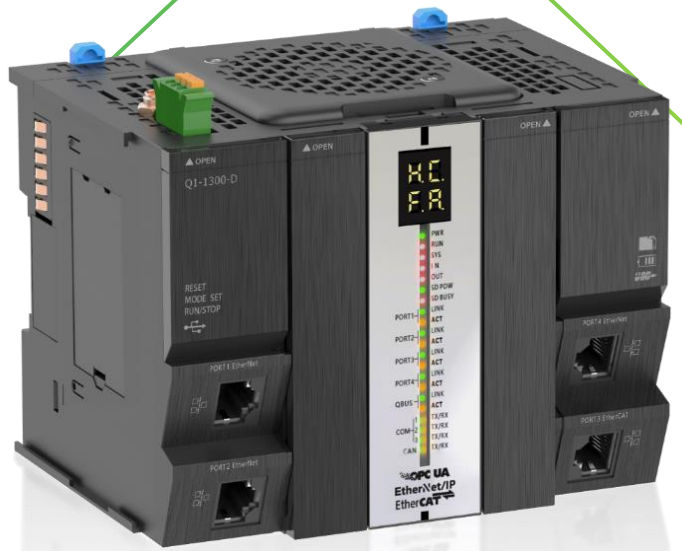
12-Connect HCFA Servos



Modify encoder parameters

Set Motion Control Program Priority





Application training

-

13-Multi-axis Motion Control

To create a better life through our work

13-Multi-axis Motion Control

2. The use of multi-axis synchronous motion control instructions: (For details, please check the CODESYS help document first)

1.MC_GearIn
2.MC_GearOut
3.MC_CamTableSelect
4.MC_CamIn
5.MC_CamOut


Control
Drive
Sensor
M&E
Info.
KCFA
To be the most valuable automation core -
component and solution provider