ICOP Technology Inc.

User Manual

QEC-RXXHU

EtherCAT Slave HID Module

With RS232/485, MPG, LCM and Keypad

(Revision 3.0)

REVISION

DATE	VERSION	DESCRIPTION
2022/09/01	**Version 1.0**	**New Release.**
2023/09/24	**Version 2.0**	**Updated Product Specifications.**
2024/1/11	**Version 3.0**	**Add Getting Started**

COPYRIGHT

Ver.3 January, 2024

TRADEMARKS ACKNOWLEDGMENT

For more detailed information or if you are interested in other ICOP products, please visit our official websites at:

- Global: www.icop.com.tw
- USA: www.icoptech.com
- Japan: www.icop.co.jp
- Europe: www.icoptech.eu
- China: www.icop.com.cn

For technical support or drivers download, please visit our websites at:

- https://www.icop.com.tw/resource_entrance

For EtherCAT solution service, support or tutorials, 86Duino Coding IDE 500+ introduction, functions, languages, libraries, etc. Please visit the QEC website:

- QEC: https://www.qec.tw/

This Manual is for the QEC series.

SAFETY INFORMATION

- Read these safety instructions carefully.
- Please carry the unit with both hands and handle it with caution.
- Power Input voltage +19 to +50VDC Power Input (Typ. +24VDC)
- Make sure the voltage of the power source is appropriate before connecting the equipment to the power outlet.
- To prevent the QEC device from shock or fire hazards, please keep it dry and away from water and humidity.
- Operating temperature between -20 to +70°C/-40 to +85°C (Option).
- When using external storage as the main operating system storage, ensure the device's power is off before connecting and removing it.
- Never touch un-insulated terminals or wire unless your power adaptor is disconnected.
- Locate your QEC device as close as possible to the socket outline for easy access and avoid force caused by the entangling of your arms with surrounding cables from the QEC device.
- If your QEC device will not be used for a period of time, make sure it is disconnected from the power source to avoid transient overvoltage damage.

WARNING!

DO NOT ATTEMPT TO OPEN OR TO DISASSEMBLE THE CHASSIS (ENCASING) OF THIS PRODUCT. PLEASE CONTACT YOUR DEALER FOR SERVICING FROM QUALIFIED TECHNICIAN.

Content

Ch. 1

General Information

1.1 Introduction

QEC-RXXHU series is an industrial-grade EtherCAT slave human interface device (HID) module. Integrates rich EtherCAT gateway functions, including RS-232/485, MPG JOG, Keypad, and LCM, into the EtherCAT network for easy installation, communication, and data transmission with other devices. In addition, QEC-RXXHU has internal monitoring data capabilities to monitor the system's operating status for timely problem detection and repair.

QEC-RXXHU has passed the validation of consistency testing tools and is suitable for various traditional industrial automation applications, such as production line control, robot control, smart warehousing, etc. It can easily convert the communication method to the EtherCAT protocol to improve system stability and reliability.

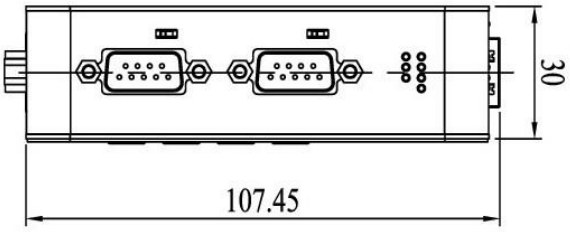
QEC-RXXHU features two standard UART interfaces that can connect to devices with RS232 or RS485 communication interfaces. With ESD protection, it is suitable for demanding industrial applications and supports overvoltage protection and hardware flow control. It can achieve a maximum speed of 115200 bps, making connecting the QEC-RXXHU to other traditional automation devices easy.
In addition to UART, QEC-RXXHU supports human-machine interface functions such as MPG JOG, Keypad, and LCM, improving the user's operating experience and making development more intuitive and convenient. For example, MPG JOG enables remote control of robotic arms or robots through the handwheel; Keypad allows parameter input through button presses; And LCM displays system operating status and data information. These human-machine interface functions make the QEC-RXXHU series suitable for a wider range of application scenarios, facilitating user operation and monitoring.

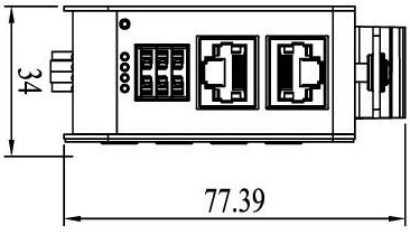
QEC-RXXHU has dimensions of 107.45 x 77.39 x 34 mm and can be conveniently installed in a system using a Din-Rail. It operates in a temperature range of -20°C to +70°C and features two network interfaces for EtherCAT network redundancy, which enhances system reliability and stability. If you are looking for a feature-rich, high-reliability, and easy-to-use industrial-grade automation product, the QEC-RXXHU series is a choice you cannot miss.

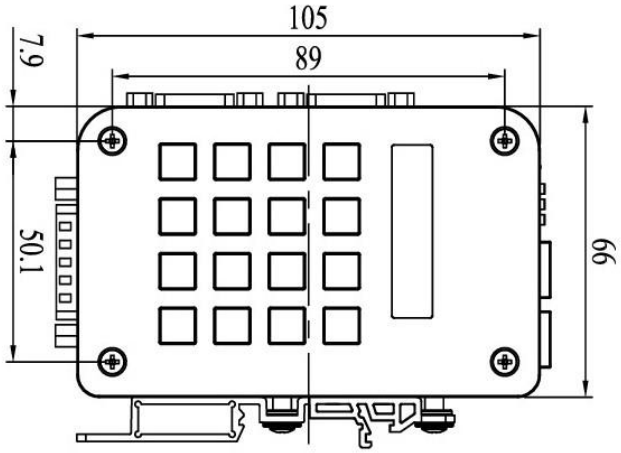
1.2 Specifications

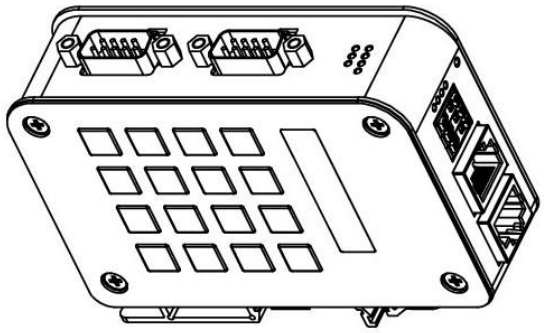
General	
Connector	UART port x2 (D-Sub 9-pin)/MPG Hand-wheel connecter
Interface	Keypad (4x4)/LCM (2 lines, 16 characters per line)
Protocol	EtherCAT
Ethernet Standard	IEEE 802.3
Transmission Rate	100Mbps
Power Connector	4-pins Power Input/Output & 2-pins FGND
Power Requirement	+19 to +50VDC Power Input (Typ. +24VDC@50mA)
Power Consumption	1.2W
LED Indicator	PWR, RUN, LINK, ERROR, TX/RX/MPG/KEYPAD/LCM status
Certifications	CE, FCC, VCCI
Hardware	
Operating Temperature	-20 to +70 °C
Dimension	107.45 x 66 x 30mm (Without DIN-Rail)
Weight	230g
Installation	DIN rail
Internal Monitoring	Temperature, Voltage, Current, Startup time
Serial Ports	
Channels	2
Interface Mode	RS232/RS485
Data transfer rate (bps)	2400,4800,9600,14400,19200,38400,57600,115200
Data width (bit)	5/6/7/8
Hardware Flow Control	CTS/RTS
Handwheel MPG	
Control Axes	Quadrature A/B Phase with axes selection
Input	Jog and federate override
Keypad	
Matrix	4x4
LCM	
LCD Module	2 lines, 16 characters per line

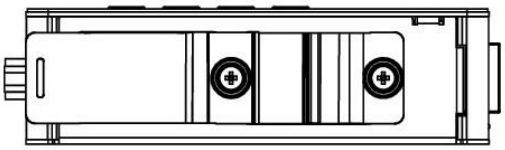
1.3 Dimension

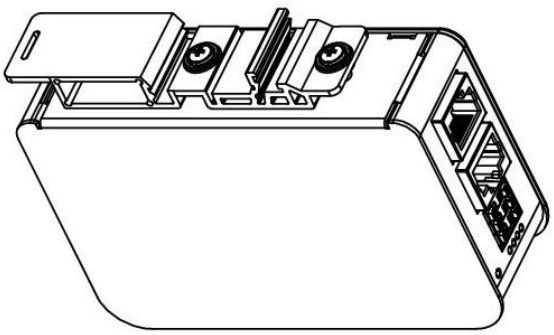










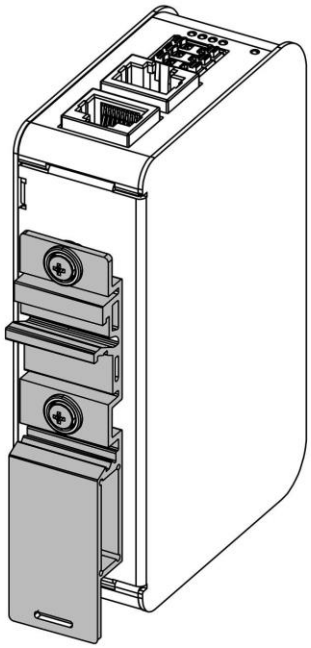


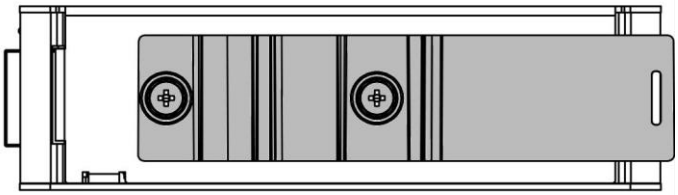
(Unit: mm)

1.4 Mounting Instruction

QEC-RXXHU series is an easy-install design to help you set-up your modules easily. Please refer to Ch.3.1 DIN-Rail installation.

- **DIN-Rill**





1.5 Ordering Information

Type	RJ45 power source		Functions		Feature		Coating
	Input	Output	HID	Functions	Standard	–	
QEC-R	X	X	HU	X	S		X

1. Type: Code 1~4

R: EtherCAT Slave

2. RJ45 Power source: Code 5~6

0: RJ45 In/Out w/o power

1: RJ45 In/Out – Power Device

3. Functions: Code 7~9

HU: HID

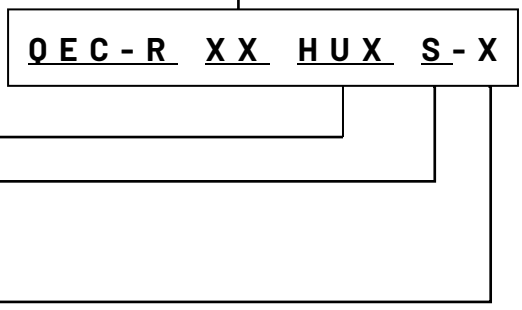
X: 0 ~ 9, with different functions

4. Feature: Code 10

S: Standard

5. Coating: Code 11

C: Yes / N: Normal



1.5.1 Ordering Part Number

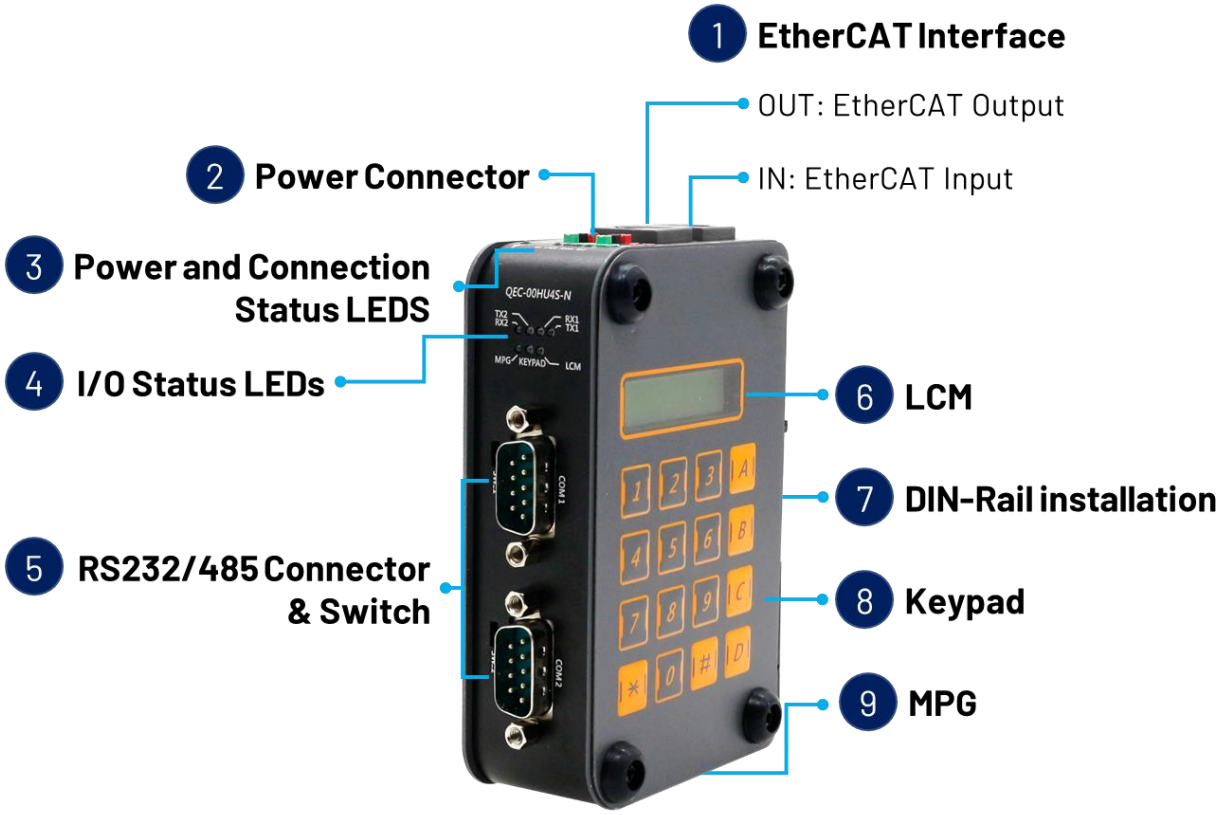
Above is the standard Part Number, please contact our sales if you need to order other part number.

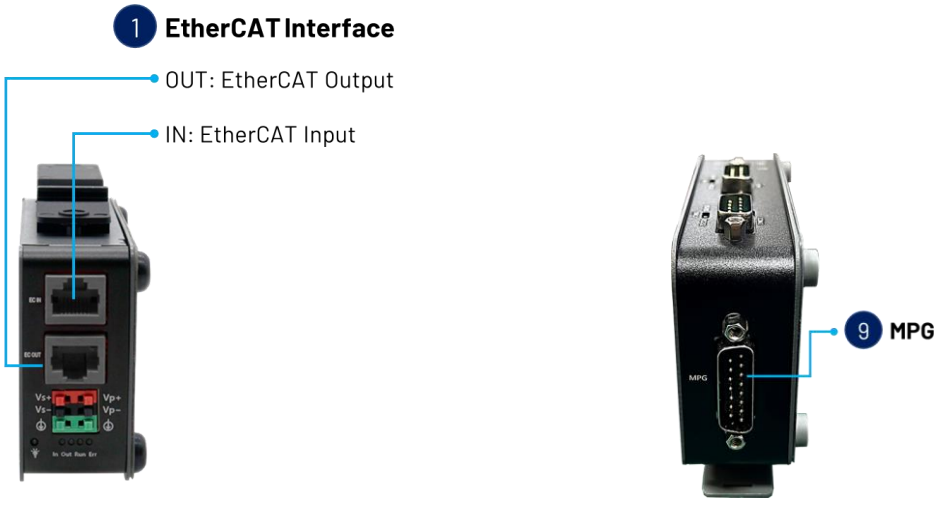
- **QEC-R00HU1S-N**: EtherCAT Slave HID module (MPG + RS232 x2)
- **QEC-R00HU2S-N**: EtherCAT Slave HID module (Keypad + LCM)
- **QEC-R00HU3S-N**: EtherCAT Slave HID module (MPG + Keypad + LCM)
- **QEC-R00HU4S-N**: EtherCAT Slave HID module (MPG + Keypad)
- **QEC-R00HU5S-N**: EtherCAT Slave HID module (RS232/485 x2)
- **QEC-R00HU6S-N**: EtherCAT Slave HID module (MPG + Keypad + RS232 x2)
- **QEC-R00HU7S-N**: EtherCAT Slave HID module (MPG + Keypad + LCM + RS232 x2)
- **QEC-R00HU9S-N**: EtherCAT Slave HID module (MPG + Keypad + LCM + RS232/485 x2)

Ch. 2

Hardware System

2.1 General Technical Data



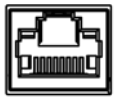


2.2 Connector Summary

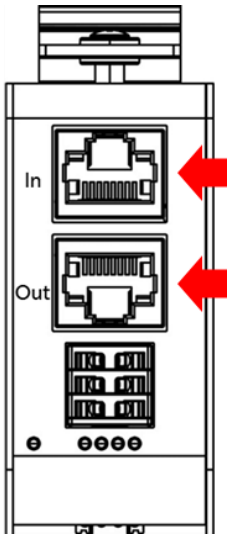
No.	Description		Type Narrative	Pin #
1	EtherCAT Interface	OUT	RJ45 Connector (Gold finger)	8-pin
		IN		8-pin
2	Power Connector		Power Socket	6-pin
3	Power and Connection Status LEDs		Status LEDs	-
4	I/O Status LEDs		Status LEDs	-
5	RS232/485 Connector & Switch		DB9 serial	9-pin
6	LCM		2 lines, 16 characters per line LCD	-
7	DIN-Rail		-	-
8	Keypad		Matrix 4x4	-
9	MPG		D-Sub Male 15P	15-pin

2.2.1 EtherCAT Interface

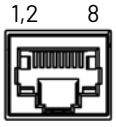
EC IN



Pin #	Signal Name	Pin #	Signal Name
1	LAN1_TX+	2	LAN1_TX-
3	LAN1_RX+	4	VS+
5	VP+	6	LAN1_RX-
7	VS-(GND)	8	VP-(GND)



EC OUT

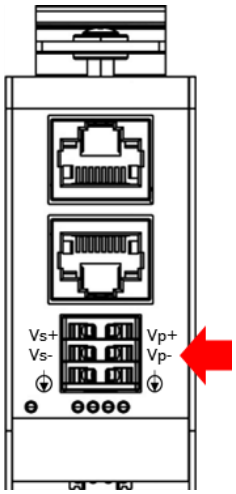


Pin #	Signal Name	Pin #	Signal Name
1	LAN2_TX+	2	LAN2_TX-
3	LAN2_RX+	4	VS+
5	VP+	6	LAN2_RX-
7	VS-(GND)	8	VP-(GND)

* PoE LAN with the Red Housing; Regular LAN with Black Housing.

* L4, L5, L7, L8 pins are option, for RJ45 Power IN/OUT.

2.2.2 Power Connector



Vs for system power; Vp for peripheral power and backup power.

Pin #	Signal Name	Pin #	Signal Name
1	Vs+	2	Vp+
3	Vs-(GND)	4	Vp-(GND)
5	F.G	6	F.G

* Power Input voltage +19 to +50VDC Power Input (Typ. +24VDC)

2.2.3 Power and Connection Status LEDs

Power Status LED

Power input is 24V (typical). The LED status provide high/low voltage warning.

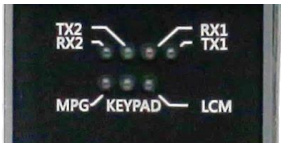
Notation	Color	States	Description
PWR 	Green / Red	Green LED On	Voltage <= 48V and Voltage >= 19V
		Green LED On Red LED On	1. Voltage < 50V and Voltage > 48V 2. Voltage < 19V and Voltage < 17V
		Red LED On	Voltage >= 50V and Voltage <= 17V

* Vs power status will be displayed first.

Connection Status LEDs

Notation	Color	States	Description
In	Green	Off	No link
		Blinking	Link and activity
		On	Link without activity
Out	Green	Off	No link
		Blinking	Link and activity
		On	Link without activity
Run	Green	Off	The device is in state INIT
		Blinking	The device is in state Pre-Operation
		Single Flash	The device is in state Safe-Operation
		On	The device is in state Operation
Err	Red	Off	No error
		Blinking	Invalid Configuration
		Single Flash	Local Error
		Double Flash	Process Data Watchdog Timeout EtherCAT Watchdog Timeout
		On	The device is in state Error

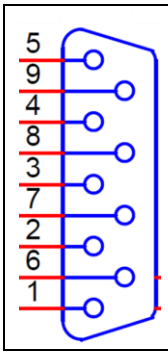
2.2.4 IO Status LEDs



Notation	Color	Description
TX1	Green	UART port1 TX signal
RX1	Green	UART port1 RX signal
TX2	Green	UART port2 TX signal
RX2	Green	UART port2 RX signal
MPG	Green	MPG signal
KEYPAD	Green	Keypad signal
LCM	Green	LCM signal

2.2.5 RS232/485 Connector & Switch

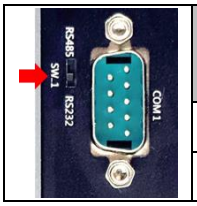
RS232/485 Connector



No.	Pin Assignment	No.	Pin Assignment
1	RS485-	6	DSR
2	RS485+/RXD	7	RTS
3	TXD	8	CTS
4	DTR	9	VCC
5	GND	-	-

* Note: RS232 and RS485 cannot be used simultaneously.

Switch



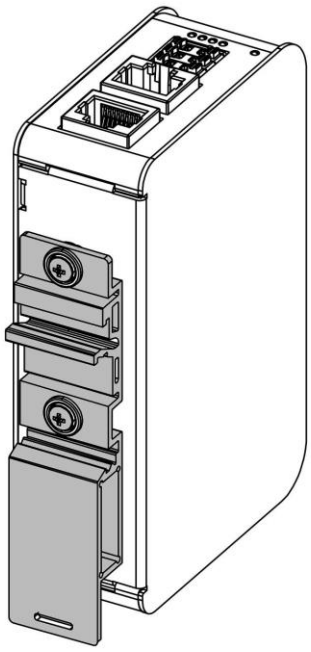
Notation	States
RS485	RS232 Enable
RS232	RS485 Enable

2.2.6 LCM

LCD Module: 2 lines, 16 characters per line.

2.2.7 DIN-Rail installation

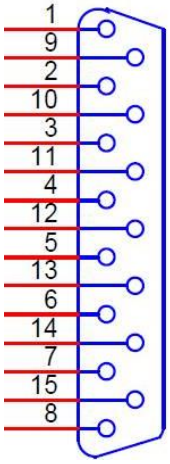
Please refer to Ch.3.1 DIN-Rail installation.



2.2.8 Keypad

Matrix: 4x4.

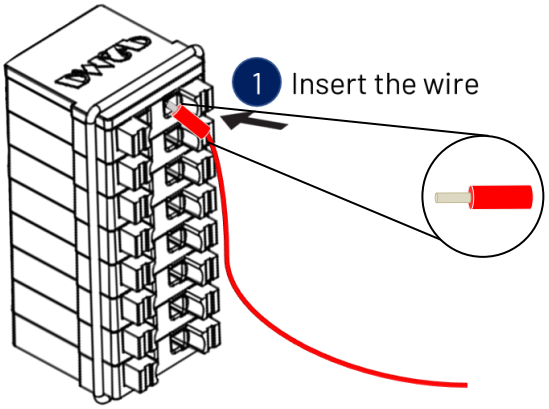
2.2.9 MPG

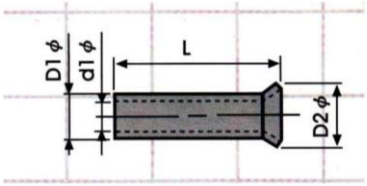


No.	Pin Assignment	No.	Pin Assignment
1	VCC	9	AXIS_B0
2	A	10	AXIS_B1
3	A-	11	AXIS_B2
4	B	12	MULTIPLE_B0
5	B-	13	MULTIPLE_B1
6	C	14	EMERGENCY
7	C-	15	LED
8	GND	-	-

2.3 Wiring to the Connector

2.3.1 Connecting the wire to the connector

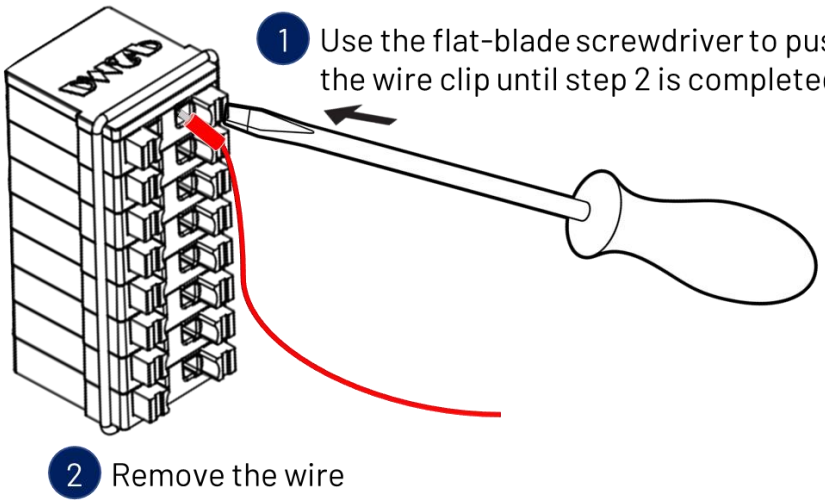




Insulated Terminals Dimensions (mm)

Position	L	ØD1	Ød1	ØD2
CN 0.5-6	6.0	1.3	1.0	1.9
CN 0.5-8	8.0	1.3	1.0	1.9
CN 0.5-10	10.0	1.3	1.0	1.9

2.3.2 Removing the wire from the connector



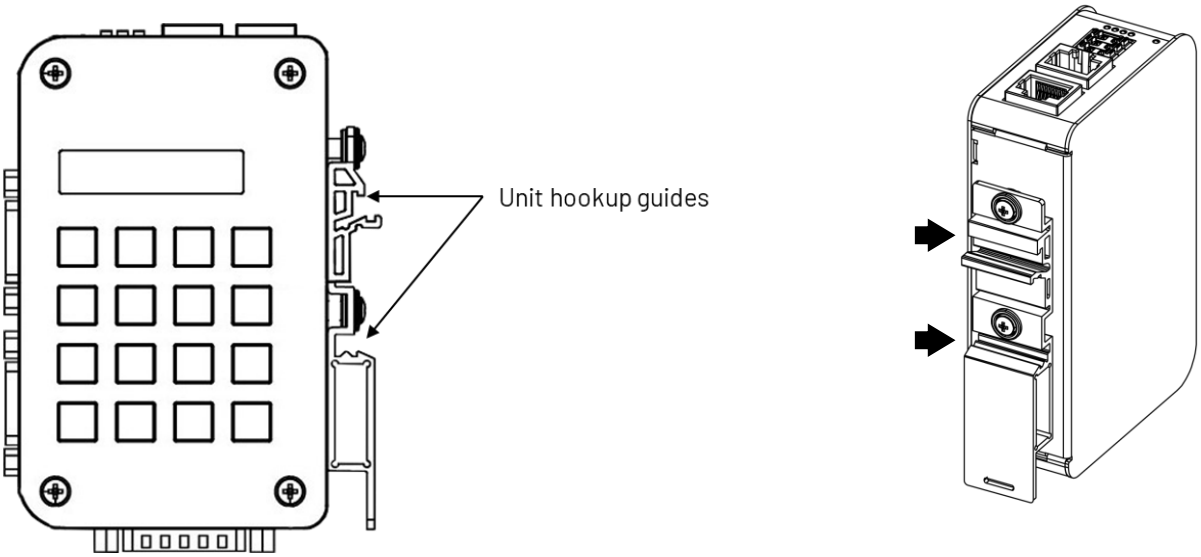
Ch. 3

Hardware Installation

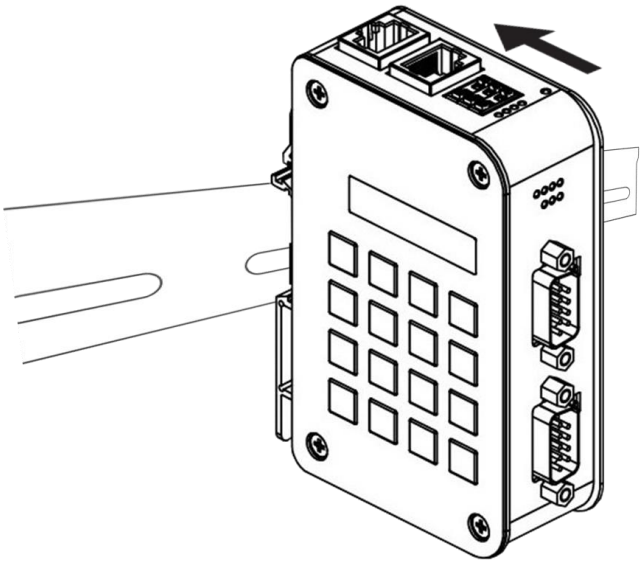
This section describes how to install QEC-RXXHU. Please turn OFF the power supply before you mount QEC-RXXHU. Always mount QEC-RXXHU one at a time.

3.1 DIN-Rail installation

Slide in the QEC-RXXHU on the hookup guides and press the QEC-RXXHU with a certain amount of force against the DIN track until the DIN Track mounting hook lock into place.



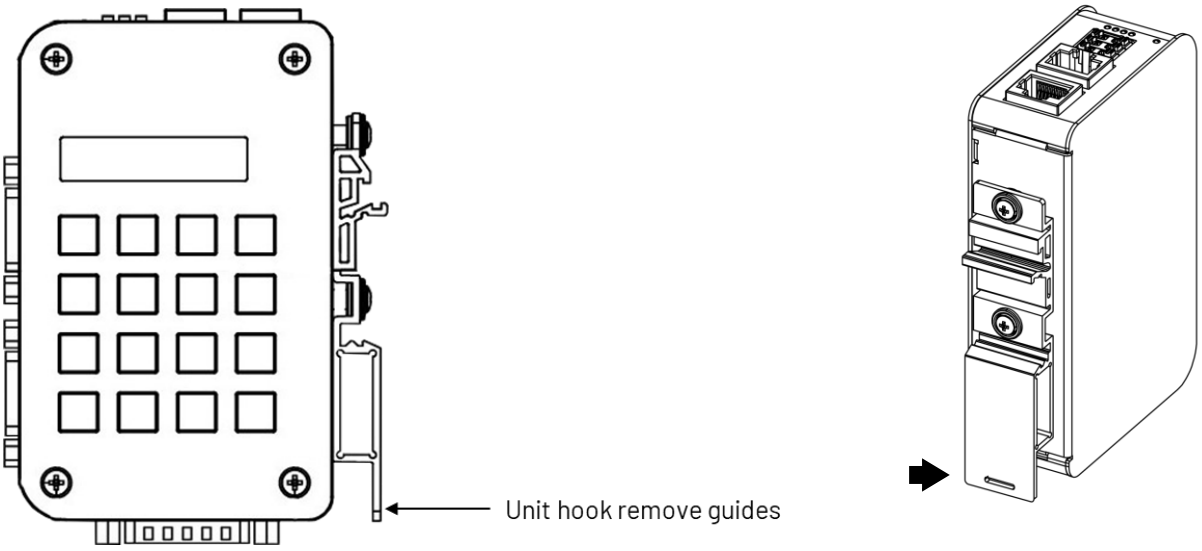
When you mount the QEC-RXXHU, releasing the DIN track mounting hook on the QEC-RXXHU is unnecessary. After you mount the QEC-RXXHU, make sure it is locked to the DIN Track.



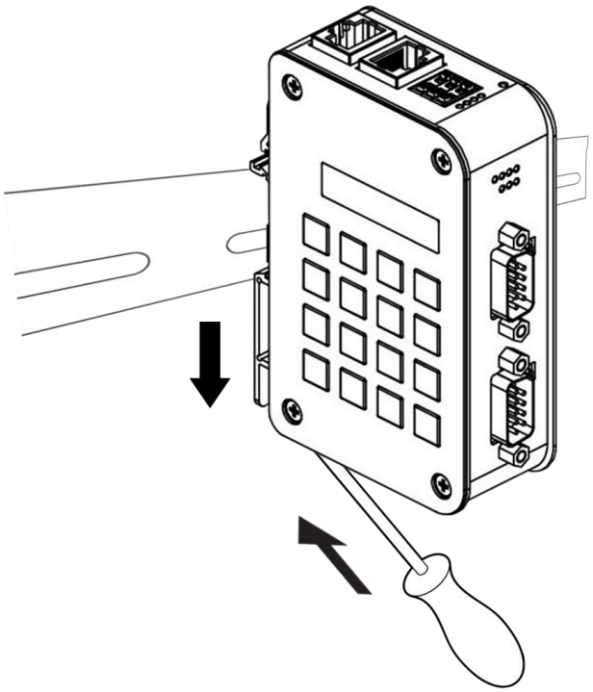
Note: Always turn OFF the Unit power supply and I/O power supply before connecting and removing the QEC-RXXHU.

3.2 Removing QEC-RXXHU Unit

Use a flat-blade screwdriver to remove the DIN Track mounting hook on the unit.



Pull down and out the flat-blade screwdriver with force against the DIN track until you hear the DIN Track remove the hook.



Ch. 4

Getting Started

This chapter explains how to access the QEC-RXXHU modules through the QEC-M-01 (EtherCAT Master) and its software, 86Duino Coding IDE. The parameter settings are easy to configure, shortening the system installation and evaluation time.

Note. QEC's PoE (Power over Ethernet)

In QEC product installations, users can easily distinguish between PoE and non-PoE: if the RJ45 house is red, it is PoE type, and if the RJ45 house is black, it is non-PoE type.

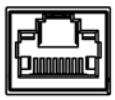
Non-PoE type

PoE type

PoE (Power over Ethernet) is a function that delivers power over the network. QEC can be equipped with an optional PoE function to reduce cabling. In practice, PoE is selected based on system equipment, so please pay attention to the following points while evaluating and testing:

1. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT master connects with a third-party EtherCAT slave).

2. The PoE function of QEC is different and incompatible with EtherCAT P, and the PoE function of QEC is based on PoE Type B, and the pin functions are as follows:



Pin #	Signal Name	Pin #	Signal Name
1	LAN1_TX+	2	LAN1_TX-
3	LAN1_RX+	4	VS+
5	VP+	6	LAN1_RX-
7	VS- (GND)	8	VP- (GND)

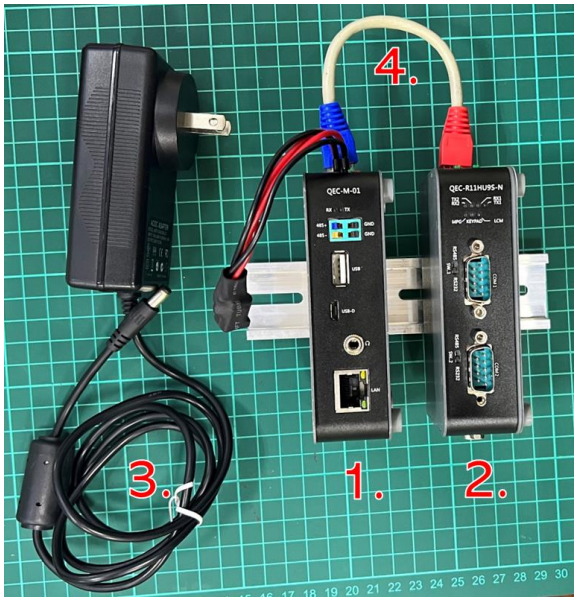
* PoE LAN with the Red Housing; Regular LAN with Black Housing.
* L4, L5, L7, L8 pins are option, for RJ45 Power IN/OUT.

3. QEC's PoE power supply is up to 24V/3A.

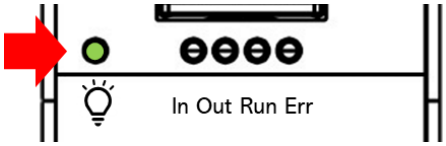
4.1 Hardware Preparation and Connection

The following devices are used here:

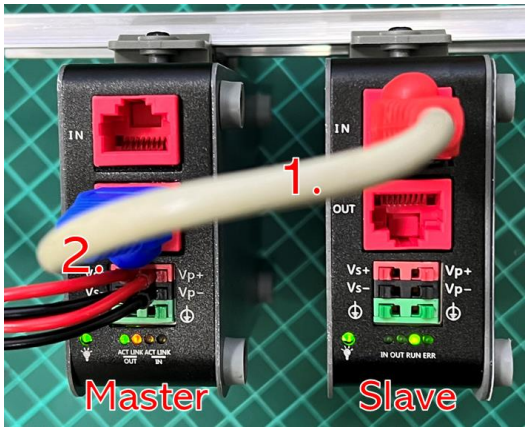
1. QEC-M-01P（EtherCAT Master/PoE）
2. QEC-R11HU9S-N (EtherCAT HID Slave, supports 2 UART, 1 MPG, 1 Keypad, 1 LCM)
3. 24V power supply & EU-type terminal cable
4. LAN cable



All QEC devices have PoE functions, so we only need to connect to Vs+/Vs and Vp+/Vp power pins (EU terminals) supplies for 19 to 50VDC power on QEC-M-01P, and then other devices will be powered by PoE. After powering on, you'll see the power LED light up and verify that the "PWR" LED indicators are ON (green).

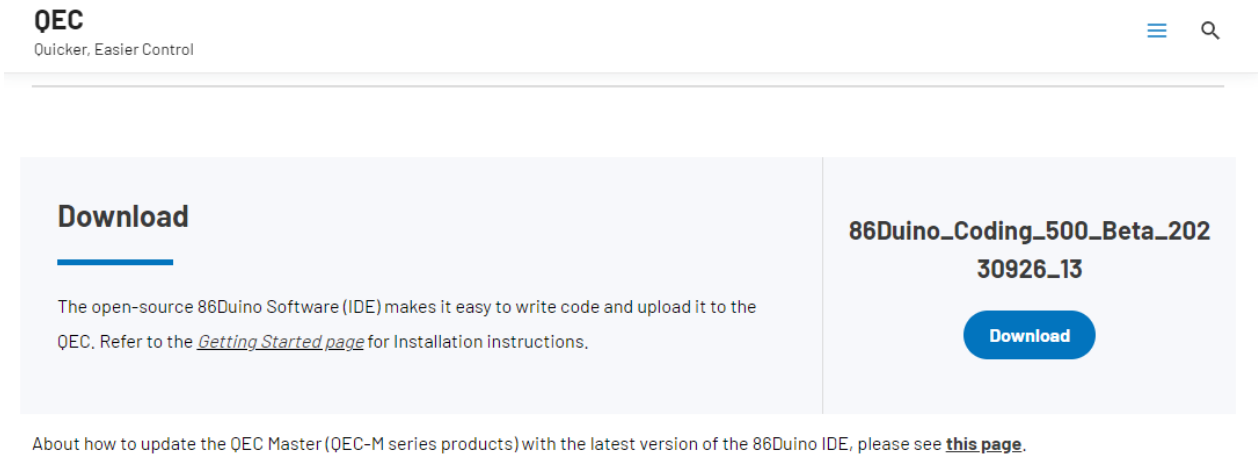


Using the EtherCAT Out port (top side) connected to the EtherCAT In port of QEC-R11HU9S via RJ45 cable (powered by PoE).



4.2 Software/Development Environment

Download 86duino IDE from https://www.qec.tw/software/.



After downloading, please unzip the downloaded zip file, no additional software installation is required, just double-click **86duino.exe** to start the IDE.

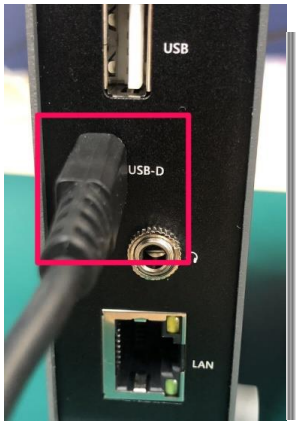
*Note: If Windows displays a warning, click Details once and then click the Continue Run button once.

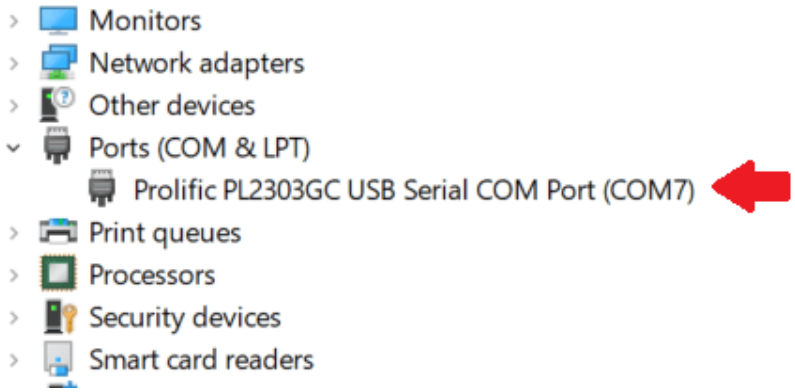
86Duino Coding IDE 500+ looks like below.

4.3 Connect to your PC and set up the environment

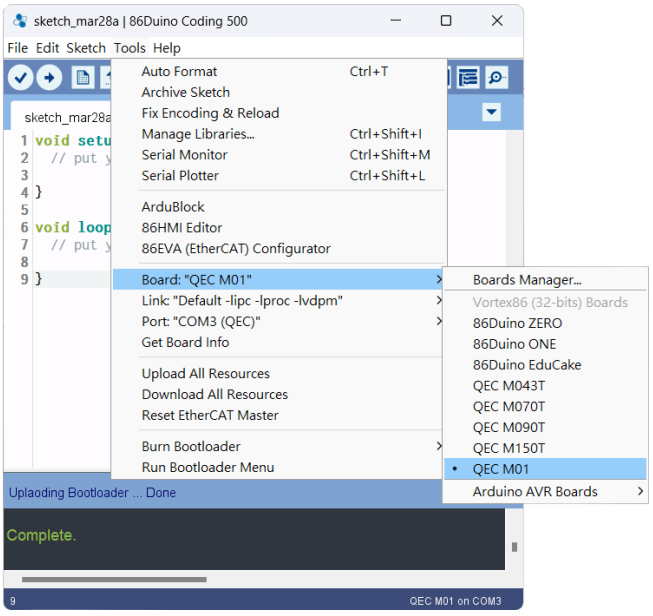
Follow the steps below to set up the environment:

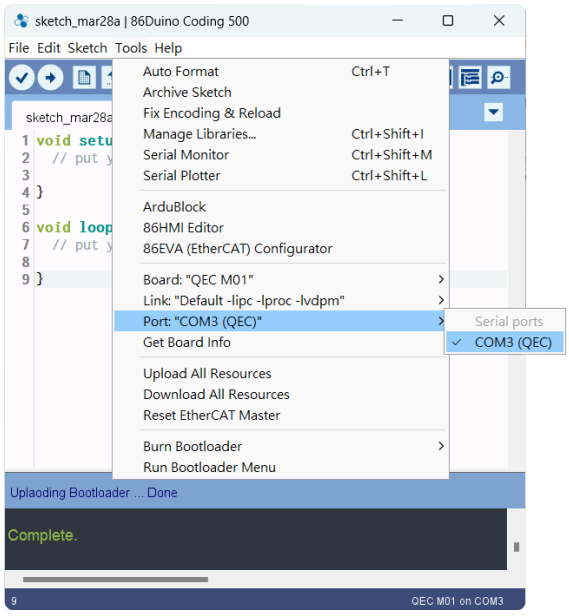
1. Connect the QEC-M-01P to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open "Device Manager" -> "Ports (COM & LPT)" in your PC and expand the ports; you should see that the "Prolific PL2303GC USB Serial COM Port (COMx)" is detected; if not, you will need to install the required drivers.
   (For Windows PL2303 driver, you can download here)





4. Open the 86Duino IDE.
5. Select the correct board: In the IDE's menu, select "Tools" ->" Board"- > QEC-M-01 (or the QEC-M master model you use).
6. Select Port: In the IDE's menu, select "Tools"- >" Port" and select the USB port to connect to the QEC-M master (in this case, COM3 (QEC)).



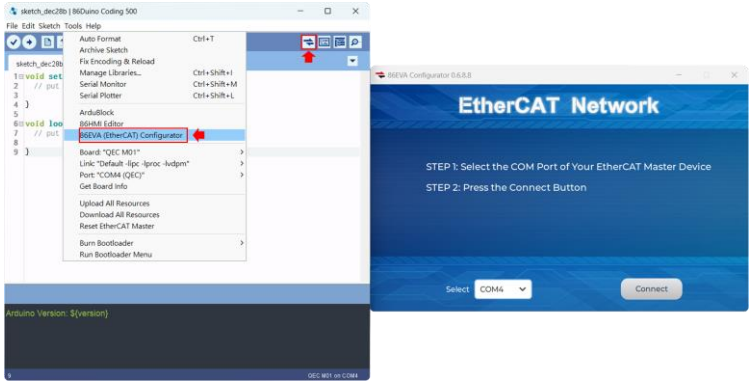


4.4 Configuration and Operation

86EVA is a graphical EtherCAT configurator based on the EtherCAT Library in the 86Duino IDE and is one of the development kits for 86Duino. The user can use it to configure the EtherCAT network quickly and start programming.

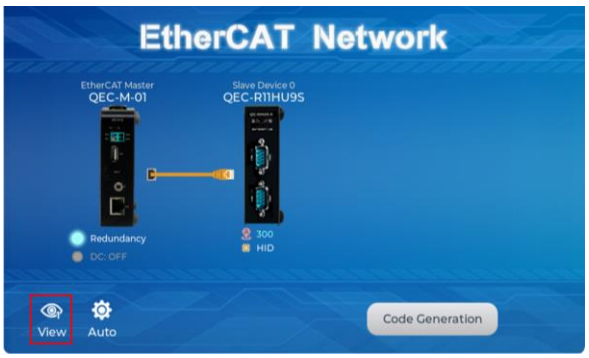
Step 1: Turn on 86EVA and scan

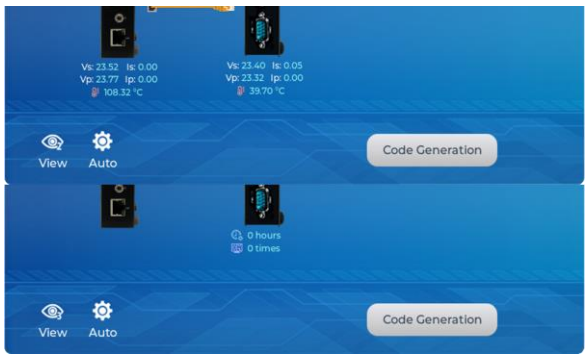
The 86EVA tool can be opened via the following buttons.



Once you have confirmed that the correct COM port has been selected of QEC-M-01P, press the Connect button to start scanning the EtherCAT network.

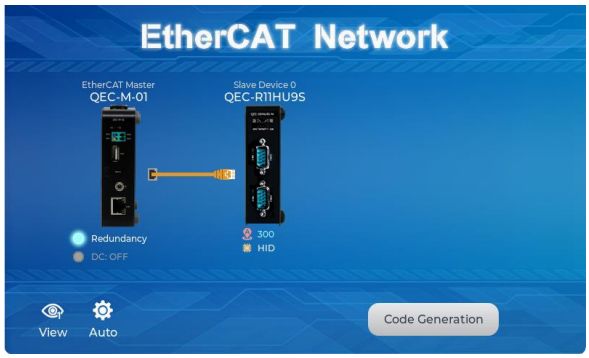
The connected devices will be displayed after the EtherCAT network has been scanned. Press the "View" button in the lower left corner to check the device's status (Voltage, Current, and Temperature; View2) and operating time (Hours; View3).





Step 2: Set the parameters

Press twice on the scanned device image to enter the corresponding parameter setting screen.



QEC-M-01

Press twice on the image of the QEC-M-01 to see the parameter settings.
This example will use the default settings and not change any settings; please click "Back" in the upper left corner to return.

QEC-R11HU9S-N

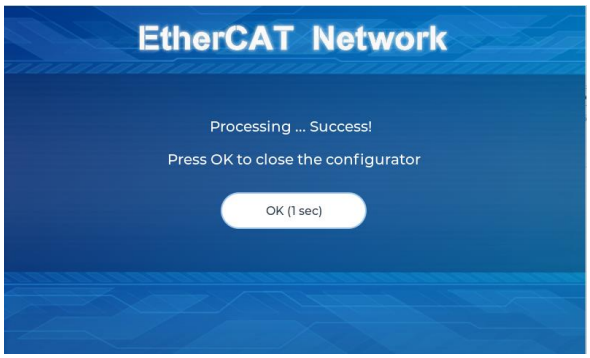
Press twice on the image of the QEC-R11HU9S to see the parameter settings.
This example will use the default settings and not change any settings; please click "Back" in the upper left corner to return.

For the Device Information, you can refer to 5.2.1 Standard Objects (0x1000-0x1FFF) and 5.2.2 Manufacturer Objects (0x5000-0x5FFF).

Step 3: Generation the code

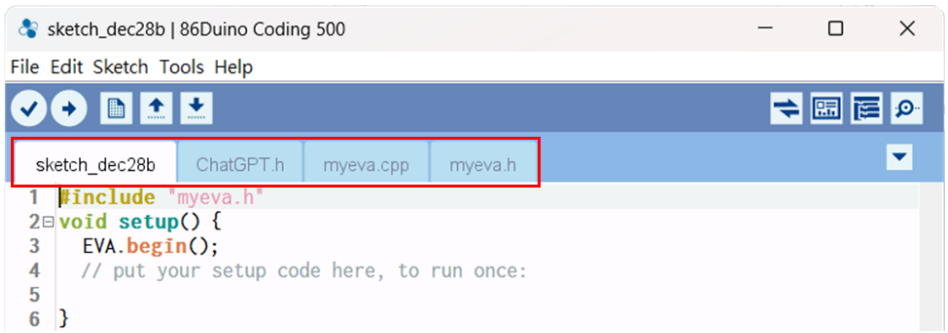
After configuring all settings, click the "Code Generation" button.

When you're done, double-click the OK button to turn off 86EVA, or it will close in 10 seconds.



The generated code and files are as follows:

- sketch_dec28b: Main Project (.ino, depending on your project name)
- ChatGPT.h: Parameters to provide to ChatGPT referred
- myeva.cpp: C++ program code of 86EVA
- myeva.h: Header file of 86EVA

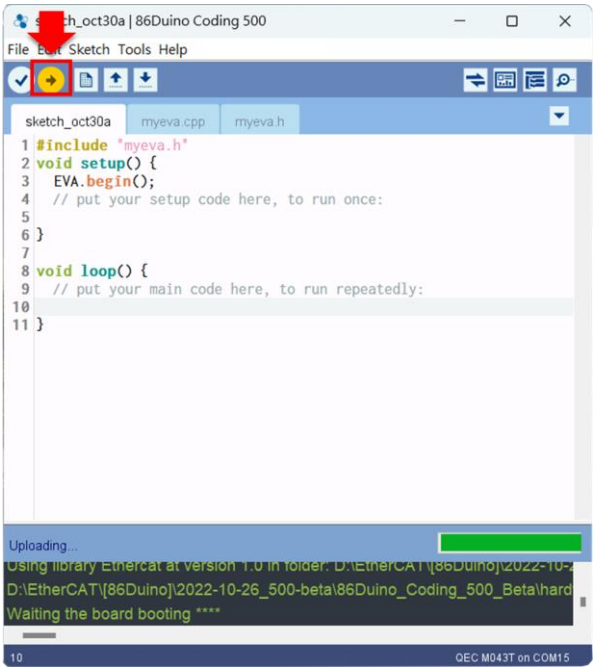


Additional note: After 86EVA generates code, the following code will be automatically generated in the main program (.ino), and any of them missing will cause 86EVA not to work.

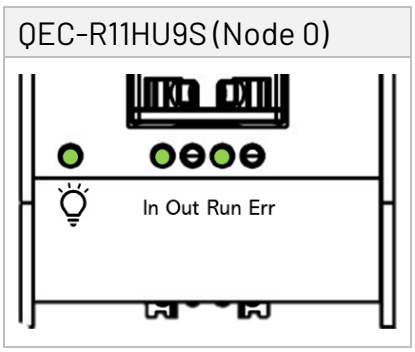
1. `#include “myeva.h”`: Include EVA Header file
2. `EVA.begin()` in `setup();` : Initialize the EVA function

Step 4: Upload the code

Once the code is generated, click on the toolbar to compile, and to confirm that the compilation is complete and error-free, you can click to upload. The program will run when the upload is complete.



After the upload, if the EtherCAT Network is running successfully, it will enter OPERATION mode.
You can confirm this by the RUN LED on your QEC slave device, which should light up. Additionally, the LED on the EtherCAT LAN ports will start blinking, indicating active operation.



EthercatDevice_QECRXXHU Class

This class within the EtherCAT Library specifically caters to QEC EtherCAT Slave Digital IO Modules. For comprehensive details on the *EthercatDevice_QECRXXHU Class*, please refer to EthercatDevice_QECRXXHU Class.

To obtain the full EtherCAT Master API User Manual, we encourage you to reach out to our sales team or email us directly at info@icop.com.tw. Our team is dedicated to providing you with comprehensive support and detailed information to enhance your experience with our products.

Practical Application Examples

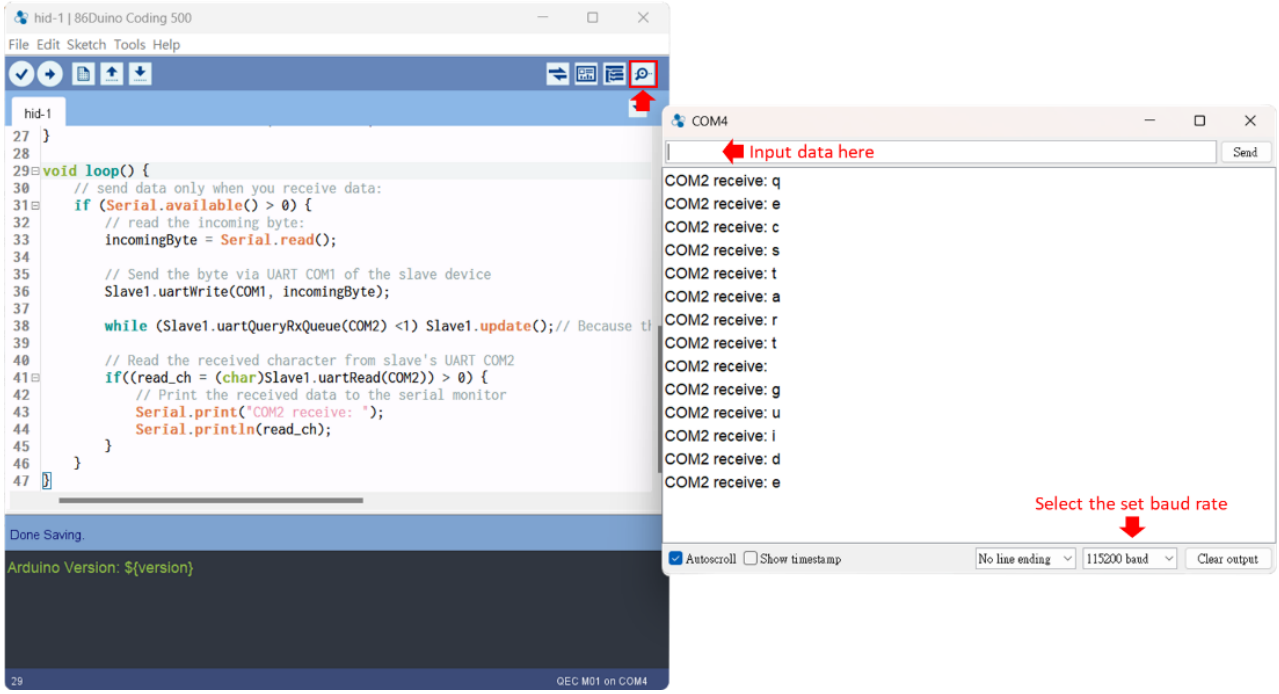
1. UART Port

The following example is reading the data from the Serial Monitor in 86Duino IDE and transferring it from COM1 to COM2. After COM2 receives the data, we print it on the Serial Monitor.

```
#include "Ethercat.h" // Include the EtherCAT Library

EthercatMaster EcatMaster; // Create an EtherCAT Master Object
EthercatDevice_QECR11HU9S Slave1; // Create an EtherCAT Slave Object for QECR11HU9S

int incomingByte = 0;  // Variable for incoming serial data
char read_ch; // Variable for read serial data (char)

void setup() {
    Serial.begin(115200); // Initialize serial communication at 115200 baud rate

    // Initialize the EtherCAT Master. If successful, all slaves enter PRE OPERATIONAL state
    EcatMaster.begin();

    // Attach QECR11HU9S slave device to the EtherCAT Master at position 0
    Slave1.attach(0, EcatMaster);

    // Start the EtherCAT Master. If successful, all slaves enter OPERATIONAL state
    // FreenRun Mode, and the parameter 1000000 sets the cycle time in nanoseconds
    EcatMaster.start(1000000, ECAT_FREERUN_AUTO);

    // Configure UART settings for two COM ports of the slave device
    Slave1.uartSetBaud(COM1, 115200); // Set baud rate for COM1
    Slave1.uartSetFormat(COM1, SERIAL_8N1); // Set data format for COM1
    Slave1.uartSetBaud(COM2, 115200); // Set baud rate for COM2
    Slave1.uartSetFormat(COM2, SERIAL_8N1); // Set data format for COM2
}

void loop() {
    // send data only when you receive data:
    if (Serial.available() > 0) {
        // read the incoming byte:
        incomingByte = Serial.read();

        // Send the byte via UART COM1 of the slave device
        Slave1.uartWrite(COM1, incomingByte);
        // Because the function is non-blocking, so we need to call update(); by ourselves
        while (Slave1.uartQueryRxQueue(COM2) <1) Slave1.update();

        // Read the received character from slave's UART COM2
        if((read_ch = (char)Slave1.uartRead(COM2)) > 0) {
            // Print the received data to the serial monitor
            Serial.print("COM2 receive: ");
            Serial.println(read_ch);
        }
    }
}
```

Note: Once the code is written, click on the toolbar to  compile, and to confirm that the compilation is complete and error-free, you can click  to upload. The program will run when the upload is complete.

After uploading, you can input a number or letter to the Serial Monitor in 86Duino IDE. All data will transfer from COM1 to COM2. After COM2 receives the data, we print it on the Serial Monitor, as in the image below.



2. Keypad + LCM + Buzzer

The following example is reading the keypad input data and printing on the specific positions of LCM according to it. Buzzer will buzz when the keypad is pressed; among them, '#' is a clear LCM display and sets the print position to the first row, and '*' is a clear LCM display and sets the print position to the second row.

```
#include "Ethercat.h"

EthercatMaster EcatMaster;
EthercatDevice_QECR11HU9S Slave1;

int lcmY = 1;

void myCallback() {
  Slave1.update();
}

void setup() {
  EcatMaster.begin();
  Slave1.attach(0, EcatMaster);
  EcatMaster.attachCyclicCallback(myCallback);
  EcatMaster.start(1000000, ECAT_SYNC);

  Slave1.keypadClear();
  Slave1.lcmClear();
}

void loop() {
  char keyPadInput = Slave1.keypadRead();

  if (keyPadInput >= '0' && keyPadInput <= '9') {
    Slave1.lcmGotoXY(keyPadInput - '0' + 1, lcmY);
    Slave1.lcmWrite(keyPadInput);
  } else if (keyPadInput >= 'A' && keyPadInput <= 'D') {
    Slave1.lcmGotoXY(keyPadInput - 'A' + 11, lcmY);
    Slave1.lcmWrite(keyPadInput);
  } else if (keyPadInput == '#') {
    lcmY = 1;
    Slave1.lcmClear();
  } else if (keyPadInput == '*') {
    lcmY = 2;
    Slave1.lcmClear();
  }

  if (keyPadInput != 0) {
    Slave1.buzzer(3000, 200);
  }
}
```

Note: Once the code is written, click on the toolbar to  compile, and to confirm that the compilation is complete and error-free, you can click  to upload. The program will run when the upload is complete.

After uploading, you can press the keypad on the QEC-R11HU9S-N's side and printing on the specific positions of LCM according to it. Buzzer will buzz when the keypad is pressed; among them, '#' is a clear LCM display and sets the print position to the first row, and '*' is a clear LCM display and sets the print position to the second row.

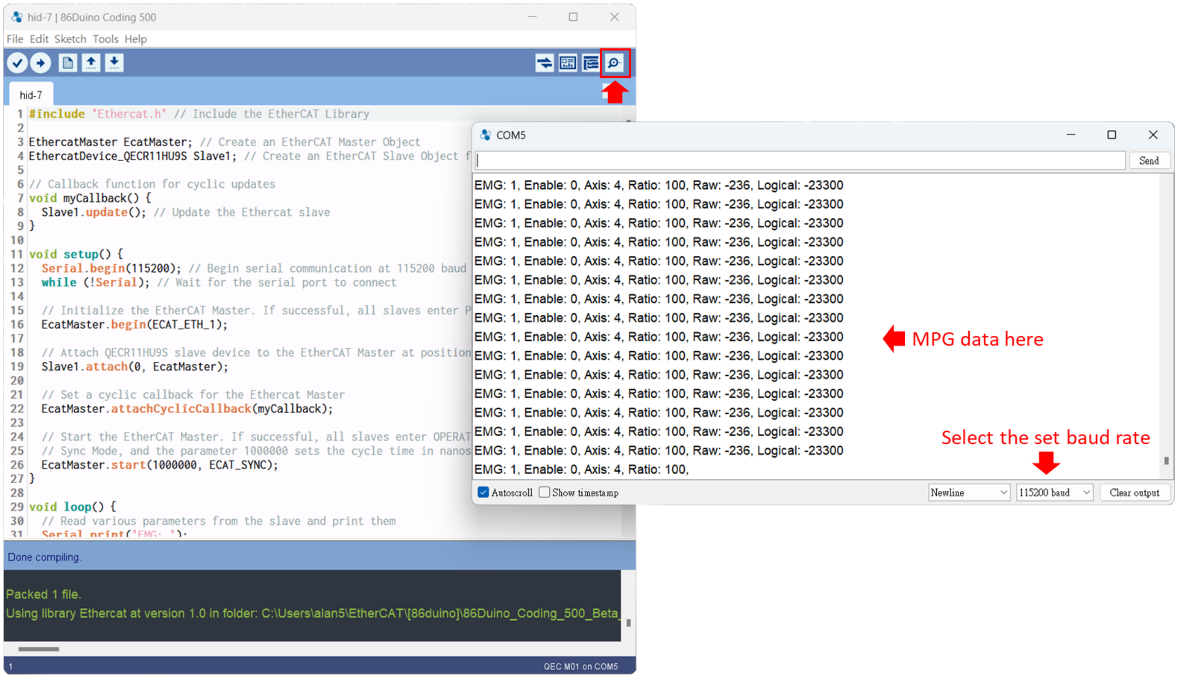
3. MPG

In the following example, we want to read the data and status of the MPG of QEC-R11HU9S, and print out EMG, Enable, Axis, Ratio, Raw, Logical data through the Serial Monitor of 86Duino IDE.

```
#include "Ethercat.h"

EthercatMaster EcatMaster;
EthercatDevice_QECR11HU9S Slave1;

void myCallback() {
  Slave1.update();
}

void setup() {
  Serial.begin(115200);
  while (!Serial);

  EcatMaster.begin(ECAT_ETH_1);
  Slave1.attach(0, EcatMaster);
  EcatMaster.attachCyclicCallback(myCallback);
  EcatMaster.start(1000000, ECAT_SYNC);
}

void loop() {
  Serial.print("EMG: ");
  Serial.print(Slave1.mpgReadEmergencyStop());
  Serial.print(", Enable: ");
  Serial.print(Slave1.mpgReadEnableSwitch());
  Serial.print(", Axis: ");
  Serial.print(Slave1.mpgReadAxis());
  Serial.print(", Ratio: ");
  Serial.print(Slave1.mpgReadRatio());
  Serial.print(", Raw: ");
  Serial.print(Slave1.mpgReadEncoderRaw());
  Serial.print(", Logical: ");
  Serial.println(Slave1.mpgReadEncoder());
}

```

Note: Once the code is written, click on the toolbar to  compile, and to confirm that the compilation is complete and error-free, you can click  to upload. The program will run when the upload is complete.

After the upload is completed, you can read the data and status of the MPG of QEC-R11HU9S, and view EMG, Enable, Axis, Ratio, Raw, Logical and other data through the Serial Monitor of 86Duino IDE.



4.5 Access Further Documentation

For those seeking comprehensive details about the EtherCAT Master API, we recommend referring to the EtherCAT Master API User Manual.This manual provides an in-depth exploration of the API, offering insights into more advanced features and capabilities.

Additionally, if you're interested in expanding your knowledge and exploring programming functions beyond the basic setup, the Language Reference Home and Libraries Reference Home are excellent resources. These sections contain valuable information and guides that cover a wide range of programming topics and libraries relevant to the EtherCAT technology.

For more info and sample request, please write to info@icop.com.tw, call your nearest ICOP Branch, or contact our Worldwide Official Distributor.

Ch. 5

Slave Information

5.1 ESI (EtherCAT Slave Information) file

The ESI files contain information unique to the EtherCAT Slave Terminals in XML format. You can load an ESI file into the Support Software to easily allocate Slave Terminal process data and other settings. The ESI files for QEC EtherCAT slaves are already installed in the Support Software.

Note. Ensuring Up-to-date Installation of the XML Device Description File (ESI)

To ensure smooth functioning, it is important to install the latest version of the XML device description file in the EtherCAT Master software. The latest version of the XML device description file can be downloaded from the QEC website.

https://www.qec.tw/

5.2 Object Dictionary

The object dictionary defined here shall be used complementary with ETG.5001 and ETG.1000.

- Device Profile: 5001
- Modul Profile: 0
- Modular Device Profile

Usage Notes:

- The PDO mapping object and SyncManager assignment object doesn't need to be defined. In that case they are created automatically.
- The following objects are fixed included in the SSC and shall not be defined in the file: 0x1000, 0x1001, 0x1008, 0x1009, 0x100a, 0x1010, 0x1011, 0x1018, 0x10F0, 0x10F1, 0x10F3, 0x1c00, 0x1c32, 0x1c33
- Entries less or equal one 8Bit shall not overlap byte borders.
- Entries greater 8Bit shall always start at an exact word border.

5.2.1 Standard Objects (0x1000-0x1FFF)

Index 1000 Device type

Index	Name	Data type	Flags	Default
1000	Device type	UINT32	RO	0x00001389 (5001)

Index 1001 Error register

Index	Name	Data type	Flags	Default
1001	Error register	UINT8	RO	0x00 (0)

Index 1008 Device name

Index	Name	Data type	Flags	Default
1008	Device name	STRING	RO	Refer to following table.

Table 4-1: Device Name

Type	Device Name	Type	Device Name
HID without PoE	QEC-R00HU1S	HID with PoE	QEC-R11HU1S
	QEC-R00HU2S		QEC-R11HU2S
	QEC-R00HU3S		QEC-R11HU3S
	QEC-R00HU4S		QEC-R11HU4S
	QEC-R00HU5S		QEC-R11HU5S
	QEC-R00HU6S		QEC-R11HU6S
	QEC-R00HU7S		QEC-R11HU7S
	QEC-R00HU8S		QEC-R11HU8S
	QEC-R00HU9S		QEC-R11HU9S

Index 1009 Hardware version

Index	Name	Data type	Flags	Default
1009	Hardware version	STRING	RO	DM406D

Index 100A

Index	Name	Data type	Flags	Default
100A	Software version	STRING	RO	1.00

Index 1018 Identity

Index	Name	Data type	Flags	Default
1018:0	Identity	UINT8	RO	> 4 <
1018:01	Vendor ID	UINT32	RO	0x00000BC3 (3011)
1018:02	Product code	UINT32	RO	Refer to following table.
1018:03	Revision	UINT32	RO	Depending by model.
1018:04	Serial number	UINT32	RO	0x00000000 (0)

Table 4-3: Product code & Revision Number

Model Name	Product code
QEC-R00HU1S	0x0086d404
QEC-R00HU5S	0x0086d400
QEC-R11HU5S	0x0086d403
QEC-R00HU9S	0x0086d401
QEC-R11HU9S	0x0086d402

Index 10F1 Error Settings

Index	Name	Data type	Flags	Default
10F1:0	Error Settings	UINT8	RO	> 2 <
10F1:01	Local Error Reaction	UINT32	RW	0x00000001 (1)
10F1:02	Sync Error Counter Limit	UINT32	RW	0x0004 (4)

Index 10F8 Timestamp Object

Index	Name	Data type	Flags	Default
10F8	Timestamp Object	UINT8	RW P	C0 1B A1 1A 36 00 00 00

Index 1A00 Input mapping

Index	Name	Data type	Flags	Default
1A00:0	Input mapping 0	UINT8	RO	> 6 <
1A00:01	SubIndex 001	UINT32	RO	0x6000:00, 32
1A00:02	SubIndex 002	UINT32	RO	0x6000:01, 1
1A00:03	SubIndex 003	UINT32	RO	0x6000:02, 1
1A00:04	SubIndex 004	UINT32	RO	0x6000:03, 1
1A00:05	SubIndex 005	UINT32	RO	0x6000:04, 1
1A00:06	SubIndex 006	UINT32	RO	0x6000:05, 1

Index 1C00 Sync manager type

Index	Name	Data type	Flags	Default
1C00:0	Sync manager type	UINT8	RO	> 4 <
1C00:01	SubIndex 001	UINT8	RO	0x01 (1)
1C00:02	SubIndex 002	UINT8	RO	0x02 (2)
1C00:03	SubIndex 003	UINT8	RO	0x03 (3)
1C00:04	SubIndex 004	UINT8	RO	0x04 (4)

Index 1C12 SyncManager 2 assignment

Index	Name	Data type	Flags	Default
1C12:0	SyncManager 2 assignment	UINT8	RO	> 0 <

Index 1C13 SyncManager 3 assignment

Index	Name	Data type	Flags	Default
1C13:0	SyncManager 3 assignment	UINT8	RO	> 1 <
1C13:01	SubIndex 001	UINT16	RO	0x1A00 (6656)

Index 1C33 SM input parameter

Index	Name	Data type	Flags	Default
1C33:0	SM input parameter	UINT8	RO	> 32 <
1C33:01	Synchronization Type	UINT16	RW	0x0001 (1)
1C33:02	Cycle Time	UINT32	RO	0x00000000 (0)
1C33:04	Synchronization Types supported	UINT16	RO	0x4003 (16387)
1C33:05	Minimum Cycle Time	UINT32	RO	0x000186A0 (100000)
1C33:06	Calc and Copy Time	UINT32	RO	0x00000000 (0)
1C33:08	Get Cycle Time	UINT16	RW	0x0000 (0)
1C33:09	Delay Time	UINT32	RO	0x00000000 (0)
1C33:0A	Sync0 Cycle Time	UINT32	RW	0x00000000 (0)
1C33:0B	SM-Event Missed	UINT16	RO	-
1C33:0C	Cycle Time Too Small	UINT16	RO	0x0057 (87)
1C33:20	Sync Error	BOOL	RO	-

5.2.2 Manufacturer Objects (0x5000-0x5FFF)

Index 0x5xxn Manufacturer Objects

Index	Object Code	DataType	Name	Default	Description
0x5000	VARIABLE	UINT16	SP_Voltage	0xXXXX	Read SP Voltage (RO)
0x5001	VARIABLE	UINT16	SP_Current	0xXXXX	Read SP Current (RO)
0x5002	VARIABLE	UINT16	PP_Voltage	0xXXXX	Read PP Voltage (RO)
0x5003	VARIABLE	UINT16	PP_Current	0xXXXX	Read PP Current (RO)
0x5004	VARIABLE	INT16	Temperature	0xXXXX	Read Temperature (RO)
0x5005	VARIABLE	UINT8	BoxStatus	0x00 (0)	(RO) NormalOperation 0 ESC_3p3_Power_NG 3 DIO_3p3_Power_NG 4 EXT_Xtal_Stop 5 EXT_Xtal_OverRang 6 PowerVoltageLowOrHigh 0x10 PowerVoltageTooLowOrTooOVer 0x11

Index 0x5010 Keypad

Index	Object Code	DataType	Name	Default	Description
0x5010	VARIABLE		Keypad	> 2 <	Read_KeyPad Write_KeyPad
		UINT8	ClearKeyPadString	0	(WO)
		STRING(16)	GetKeyPadString	0	(RO)

Index 0x5011 LCM

Index	Object Code	DataType	Name	Default	Description
0x5011	VARIABLE		LCM	> 3 <	Read_LCM Write_LCM
		UINT8	Row	0x2	(RO)
		UINT8	Column	0x10	(RO)
		STRING(32)	Buffer		(WO)

Index 0x5013 COM1

Index	Object Code	DataType	Name	Default	Description
0x5013	RECORD		COM1	> 15 <	Read_COM1 Write_COM1
		UINT32	Baud_Rate	0x1C200	(RW) 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200
		UINT8	Data_Width	0x08	(RW) 5, 6, 7, 8
		UINT8	StopBit	0	(RW) "0 = One "STOP bit" is generated in the transmitted data. 1 = When select 5-bit word length, 1.5 "STOP bit" is generated in the transmitted data.When select 6-, 7- and 8-bit word length, 2 "STOP bit" is generated in the transmitted data."
		UINT8	Parity	0	(RW) 0=none, 1=odd, 3=even, 5= mark, 7=space
		UINT16	BytesToWrite	0	(RO) Return How many bytes in Tx Buffer waiting to transmission.
		UINT16	BytesToRead	0	(RO) Return How many bytes in Rx Buffer waiting to Read.
		UINT8	EnableHardwareFlowControl	0	(RW) 0:disable, 1:Enable
		UINT8	RTS	0	(RW) only can work on hardware flow control is disabled.
		UINT8	CTS	0	(RO)
		UINT8	DTR	0	(RW)
		UINT8	DSR	0	(RO)
		STRING(256)	Tx	0	(WO)
		STRING(256)	Rx	0	(RO)
		UINT8	ClearFIFO	0	(WO)
		UINT8	Mode	1	(RO) 1=RS232 Mode ,0=RS485 Mode

Index 0x5014 COM2

Index	Object Code	DataType	Name	Default	Description
0x5014	RECORD		COM2	> 15 <	Read_COM2 Write_COM2
		UINT32	Baud_Rate	0x1C200	(RW) 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200
		UINT8	Data_Width	0x08	(RW) 5, 6, 7, 8
		UINT8	StopBit	0	(RW) "0 = One "STOP bit" is generated in the transmitted data. 1 = When select 5-bit word length, 1.5 "STOP bit" is generated in the transmitted data.When select 6-, 7- and 8-bit word length, 2 "STOP bit" is generated in the transmitted data."
		UINT8	Parity	0	(RW) 0=none, 1=odd, 3=even, 5= mark, 7=space
		UINT16	BytesToWrite	0	(RO) Return How many bytes in Tx Buffer waiting to transmission.
		UINT16	BytesToRead	0	(RO) Return How many bytes in Rx Buffer waiting to Read.
		UINT8	EnableHardwareFlowControl	0	(RW) 0:disable, 1:Enable
		UINT8	RTS	0	(RW) only can work on hardware flow control is disabled.
		UINT8	CTS	0	(RO)
		UINT8	DTR	0	(RW)
		UINT8	DSR	0	(RO)
		STRING(256)	Tx	0	(WO)
		STRING(256)	Rx	0	(RO)
		UINT8	ClearFIFO	0	(WO)
		UINT8	Mode	1	(RO) 1=RS232 Mode ,0=RS485 Mode

Index 0x5015 Hand Wheel Count Value

Index	Object Code	DataType	Name	Default	Description
0x5015	VARIABLE	INT32	HandWheelCountValue	0	(WO) Write Hand Wheel Count Value.

Index 0x5016 Invert Hand Wheel Count

Index	Object Code	DataType	Name	Default	Description
0x5016	VARIABLE	UINT8	InvertHandWheelCount	0	(RW) ReadInvertHandWheelState, WriteInvertHandWheelState 0=Forward, 1=Reverse

Index 0x5017 Switch Filter Value

Index	Object Code	DataType	Name	Default	Description
0x5017	VARIABLE	UINT32	SwitchFilterValue	0xfa0	(RW) ReadFilterValue, WriteFilterValue uint microSec ,Max 16000000uS.

Index 0x5018 Tone

Index	Object Code	DataType	Name	Default	Description
0x5018	RECORD		Tone		ReadTone, WriteTone
		UINT32	Frequency	0x01	(RW) The frequency of the tone in hertz. Min 1Hz Max 100KHz.
		UINT32	Duration	0x01	(RW) The duration of the tone in milliseconds. 0: keep playing.
		UINT8	Start	0	(RW) 0:stop play, 1:start Playing Or Playing.

5.2.3 Especial Objects (0x6000-0xFFFF)

Index 0x6nnx Input Data of the Module (0x6000 - 0x6FFF)

Index	Object Code	DataType	Name	Default	Description
0x6000	VARIABLE	INT32	HandWheelCount	0	(RO)
0x6001	RECORD		MPG_SwitchState	> 5 <	
		BOOL	EmergencyStop	0	(RO) 0: Release, 1:Push Down
		BOOL	MPG_Enalbed	0	(RO)
		BIT3	Axis	0	(RO) 0: OFF, 1:X, 2:Y, 3:Z, 4:4, 5:5, 6:6
		BIT2	Speed	0	(RO) 0:x1, 1:X10, 2:X100, 3:No use
		BIT1	PaddinBits	0	(RO)

Index 0x8nnx Configuration Data of Module (0x8000 – 0x8FFF)

Index	Object Code	DataType	Name	Default	Description
0x8FF0	VARIABLE	UINT8	OutputInputSetting		(RW)

Index 0xFxxx Device Object (0xF000 – 0xFFFF)

Index	Object Code	DataType	Name	Default	Description
0xF000	RECORD		Modular Device Profile	> 2 <	
		UINT16	Index distance	0x10	(RO)
		UINT16	Maximum number of modules		(RO)

Warranty

This product is warranted to be in good working order for a period of one year from the date of purchase. Should this product fail to be in good working order at any time during this period, we will, at our option, replace or repair it at no additional charge except as set forth in the following terms. This warranty does not apply to products damaged by misuse, modifications, accident or disaster. Vendor assumes no liability for any damages, lost profits, lost savings or any other incidental or consequential damage resulting from the use, misuse of, originality to use this product. Vendor will not be liable for any claim made by any other related party. Return authorization must be obtained from the vendor before returned merchandise will be accepted. Authorization can be obtained by calling or faxing the vendor and requesting a Return Merchandise Authorization (RMA) number. Returned goods should always be accompanied by a clear problem description.