# Start Guide

## HID: Keypad + LCM + Buzzer

86Duino Coding IDE 500

EtherCAT Library

(Version 1.1)

# Revision

| Date | Version | Description |
|---|---|---|
| 2024/1/8 | VERSION1.0 | NEW RELEASE. |
| 2024/11/20 | VERSION1.1 | SPLIT THE DEVELOPMENT STEPS INTO TWO DOCUMENTS. |

# Preface

In this guide, we will show you how to use the EtherCAT Master QEC-M-01P and the QEC-RXXHUX Series (EtherCAT HID Slave).

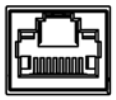
## Notes QEC's PoE (Power over Ethernet)

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. 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:

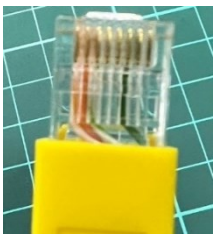



| Pin # | Signal Name | Pin # | Signal Name |
|---|---|---|---|
| 1 | LAN1_TX+ | 2 | LAN1_TX- |
| 3 | LAN1_RX+ | 4 | VS+ |
| 5 | VP+ | 6 | LAN1_RX- |
| 7 | VS-(GND) | 8 | VP-(GND) |

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

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

2. 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).

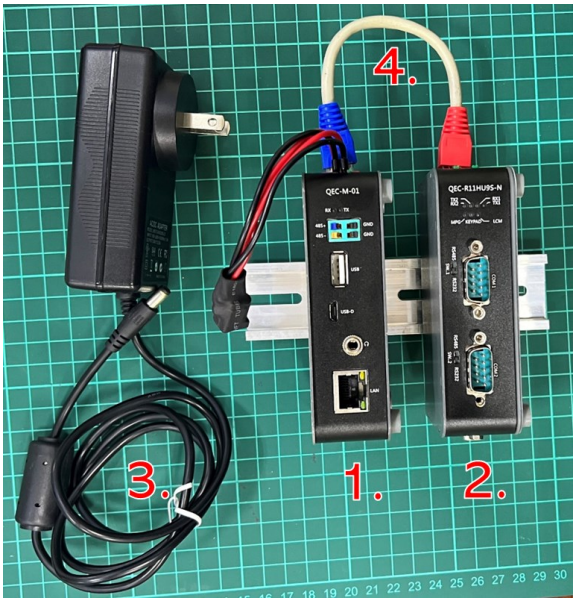


3. QEC's PoE power supply is up to 24V/3A.

# 1. Connection and wiring hardware

The following devices are used here:

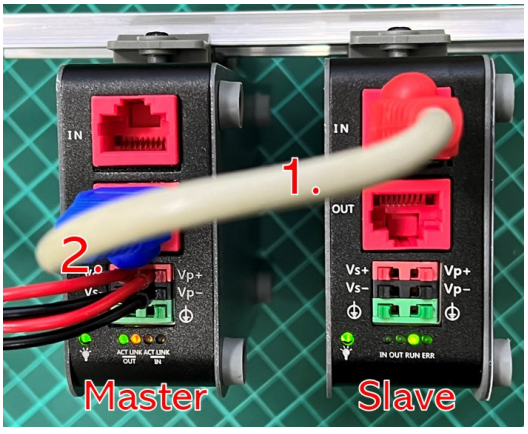
1. QEC-M-01P（EtherCAT Master/PoE）
2. QEC-R11HU9S-N (EtherCAT HID Slave, supports 2 UART, 1 MPG, 1 Keypad, 1 LCM)
3. 24V power supply & EU-type terminal cable
4. RS232 cable & LAN cable




## 1.1 QEC-M-01P & QEC-R11HU9S-N

QEC EtherCAT master with PoE function.

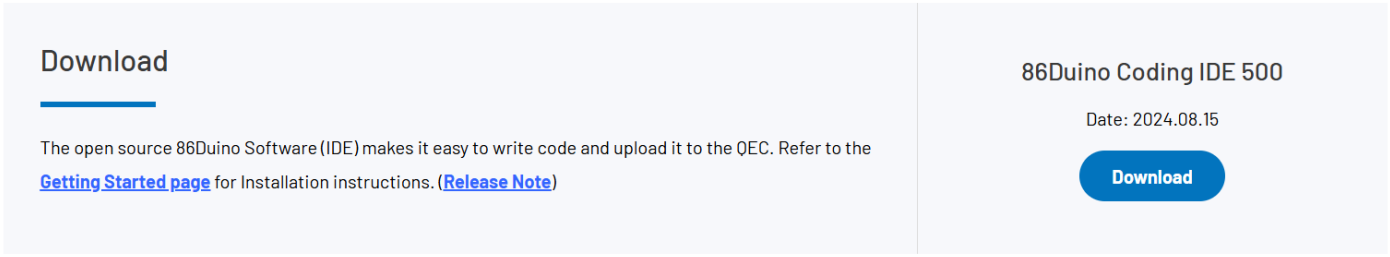
1. Using the EtherCAT Out port (top side) connected to the EtherCAT In port of QEC-R11HU9S via RJ45 cable (powered by PoE).
2. Connect to Vs+/Vs- and Vp+/Vp- power supplies via EU terminals for 24V power.



We will use the keypad, LCM, and Buzzer features onboard the QEC-R11HU9S-N device.

# 2. Software/Development Environment

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




About how to update the QEC Master (QEC-M series products) with the latest version of the 86Duino IDE, please see this page.

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

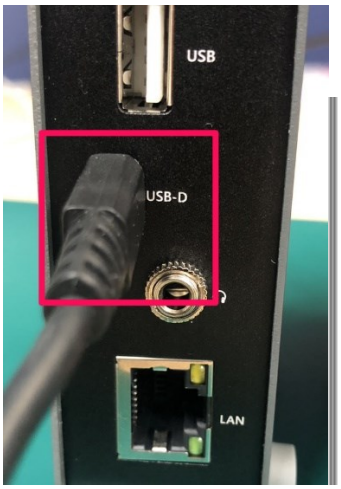

**Note**:
If Windows displays a warning, click Details once and then click the Continue Run button once.

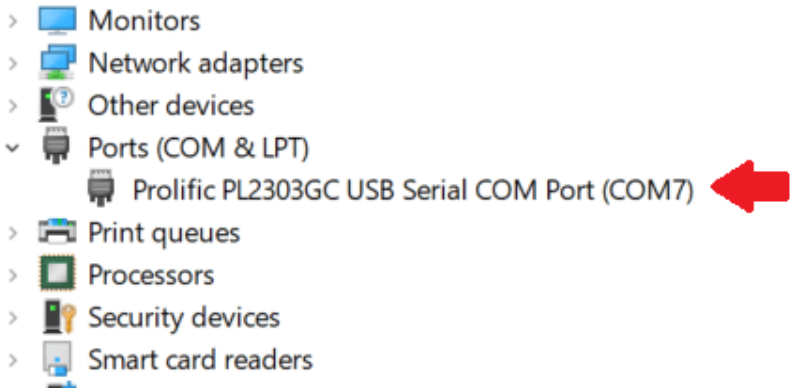
86Duino Coding IDE 500+ looks like below.

# 3. Connect to PC and set up the environment

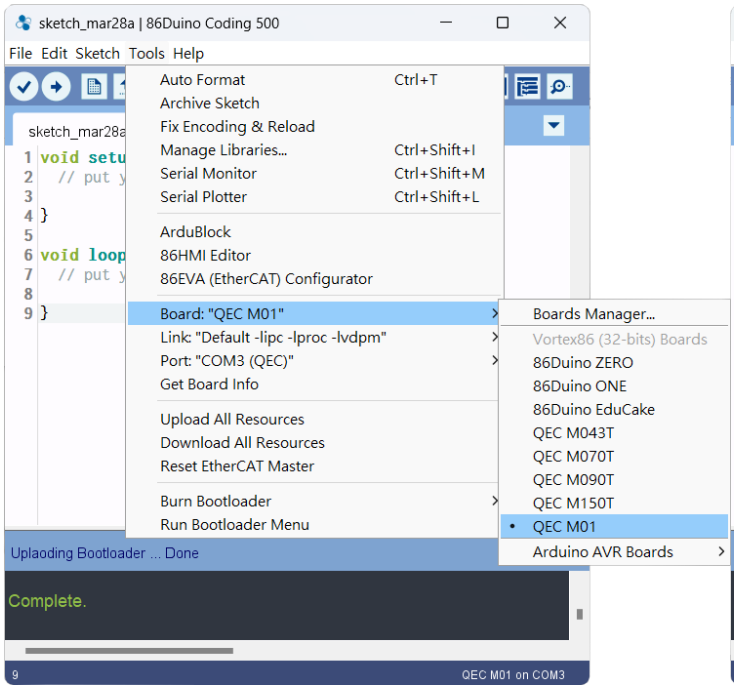
Follow the steps below to set up the environment:

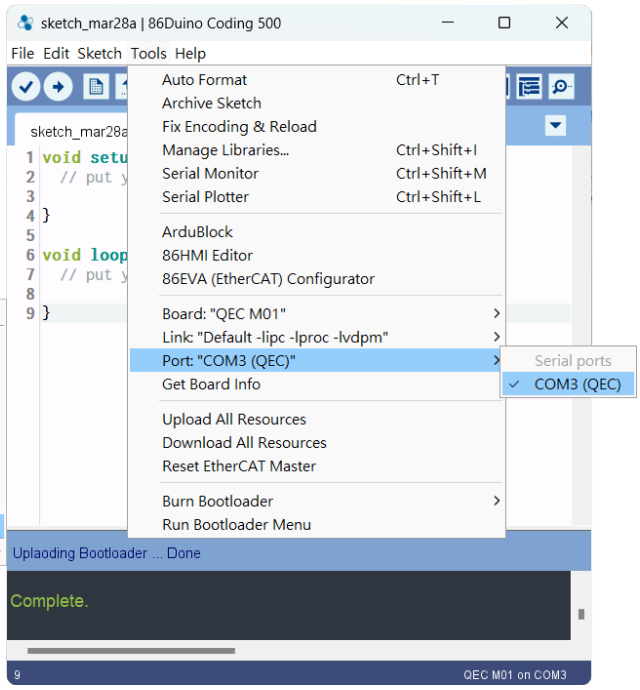
1. Connect the QEC-M-01P to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “Device Manager” (select in the menu after pressing Win+X) ->” Ports (COM & LPT)” in your PC and expand the ports; you should see that the “Prolific PL2303GC USB Serial COM Port (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. Write code

The EtherCAT master (QEC-M-01P) and the HID slave (QEC-R11HU9S-N) can be configured and programmed via the EtherCAT library in the 86Duino IDE. The Arduino development environment has two main parts: setup() and loop(), which correspond to initialization and main programs. Before operating the EtherCAT network, you must configure it once. The process should be from Pre-OP to OP mode in EtherCAT devices.

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.

When using QEC Slave, you can use the dedicated QEC Ethercat Slave Library. For example, QEC-R11HU9S can be used EthercatDevice_QECRXXHU Class.

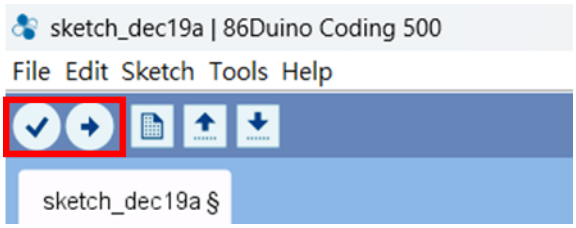
The example code is as follows:

```
#include "Ethercat.h"

EthercatMaster EcatMaster;
EthercatDevice_QECR11HU9S Slave1;

void myCallback() {
  Slave1.update(); // Update the Ethercat slave every frame
}

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

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

int lcmY = 1;
void loop() {
  // Read input from the Slave's keypad
  char keyPadInput = Slave1.keypadRead();
```

```
  // Check if the input is a digit
  if (keyPadInput >= '0' && keyPadInput <= '9') {
    // Display the digit on the LCM at the specified position
    Slave1.lcmGotoXY(keyPadInput - '0' + 1, lcmY);
    Slave1.lcmWrite(keyPadInput);
  }
  // Check if the input is a letter from A to D
  else if (keyPadInput >= 'A' && keyPadInput <= 'D') {
    // Display the letter on the LCM at the specified position
    Slave1.lcmGotoXY(keyPadInput - 'A' + 11, lcmY);
    Slave1.lcmWrite(keyPadInput);
  }
  // Check if the '#' key is pressed
  else if (keyPadInput == '#') {
    // Reset to the first line on LCM and clear the screen
    lcmY = 1;
    Slave1.lcmClear();
  }
  // Check if the '*' key is pressed
  else if (keyPadInput == '*') {
    // Change to the second line on LCM and clear the screen
    lcmY = 2;
    Slave1.lcmClear();
  }

  // If any key was pressed, activate the buzzer
  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.

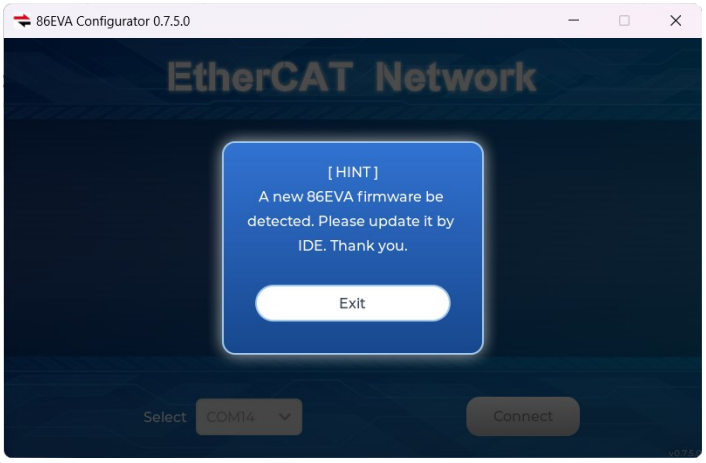


After uploading, you can press the keypad on the QEC-R11HU9S-N's side and printing on the specific positions of LCM according to it. Buzzer will buzz when the keypad is pressed; among them, '#' is a clear LCM display and sets the print position to the first row, and '*' is a clear LCM display and sets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

# Troubleshooting

## QEC-M-01 cannot successfully upload code

When you are unable to successfully upload code, please open 86EVA to check if your QEC EtherCAT Master's environment is abnormal. As shown in the figure below, please try updating your QEC EtherCAT Master's environment, which will include the following three items: Bootloader, EtherCAT firmware, and EtherCAT tool.



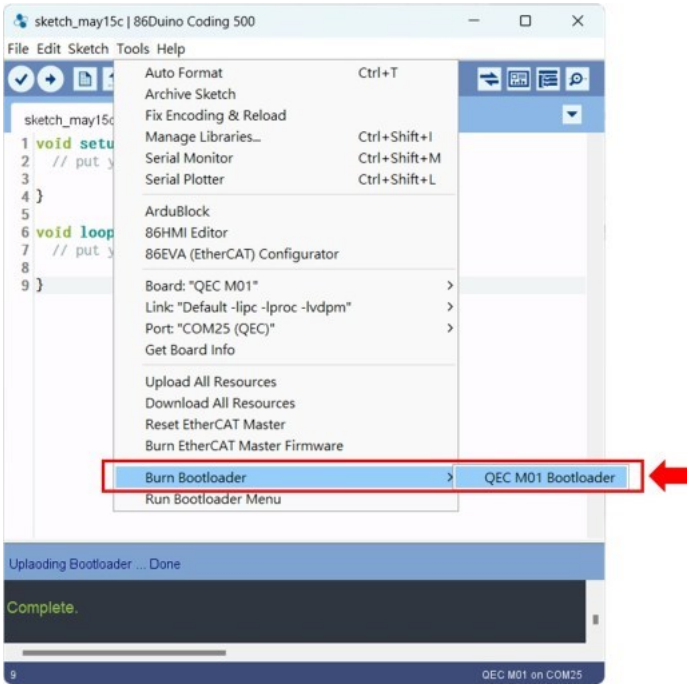

Now, we will further explain how to proceed with the update:

### Step 1: Setting up QEC-M

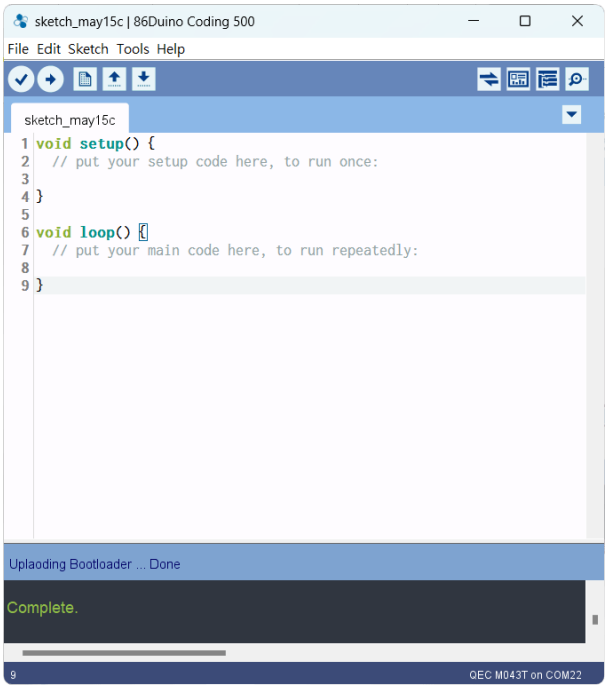
1. Download and install 86Duino IDE 500 (or a newer version): You can download it from Software.
2. Connect the QEC-M: Use a USB cable to connect the QEC-M to your computer.
3. Open 86Duino IDE: After the installation is complete, open the 86Duino IDE software.
4. Select Board: From the IDE menu, choose “Tools” > “Board” > “QEC-M-01” (or the specific model of QEC-M you are using).
5. Select Port: From the IDE menu, choose “Tools” > “Port” and select the USB port to which the QEC-M is connected.

## Step 2: Click “Burn Bootloader” button

After connecting to your QEC-M product, go to “Tools”> “Burn Bootloader”. The currently selected QEC-M name will appear. Clicking on it will start the update process, which will take approximately 5-20 minutes.
QEC-M-01:




## Step 3: Complete the Update




After completing the above steps, your QEC-M has been successfully updated to the latest version of the development environment.

# 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.