Start Guide

Ezi-IO EtherCAT Digital I/O Control

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

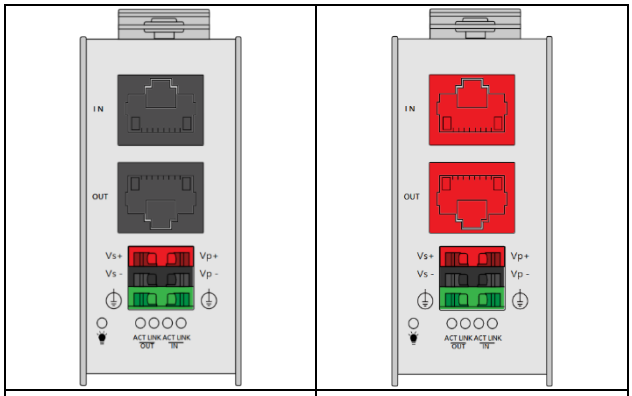
Date	Version	Description
2025/06/13	Version1.0	New Release.

Preface

In this guide, we will show you how to use the EtherCAT MDevice QEC-M-01 and the Ezi-IO (EtherCAT Input/Output Module).

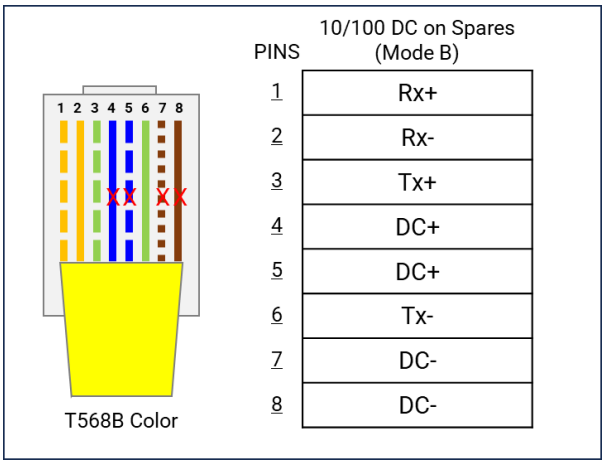
Notes QEC's PoE (Power over Ethernet)

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:



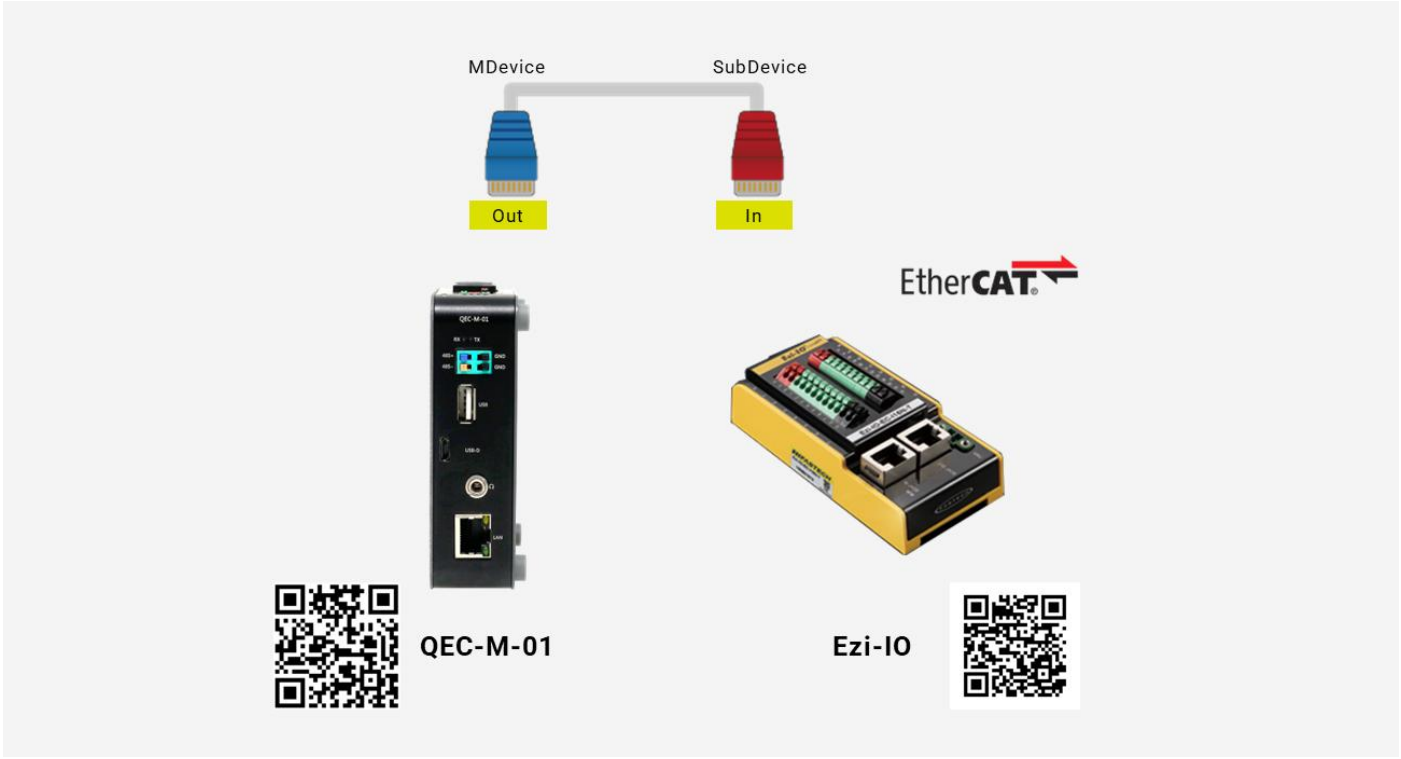
2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).

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

1. Connection and wiring hardware

The following devices are used here:

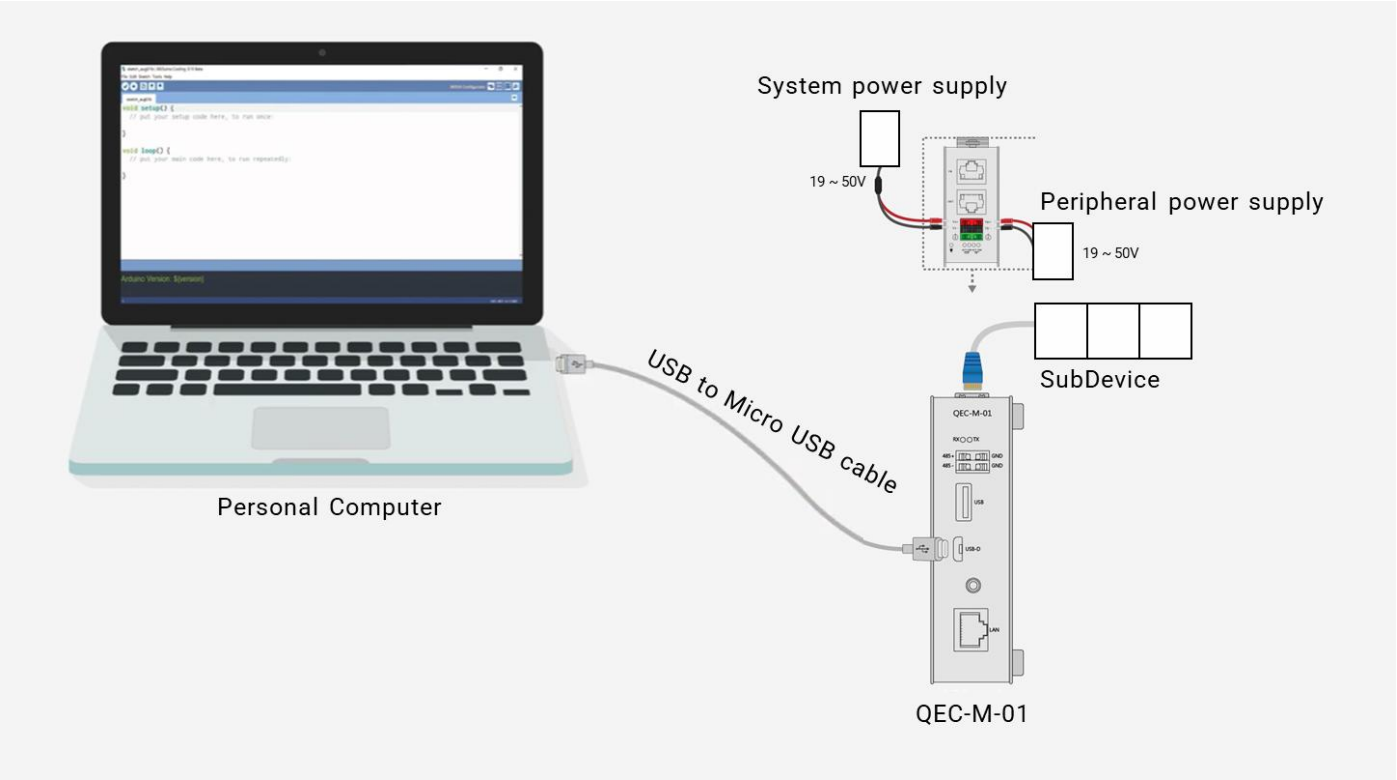
1. QEC-M-01 (EtherCAT MDevice)
2. Ezi-IO (EtherCAT Input/Output Module).
3. 24V power supply & EU-type terminal cable & LAN cable



1.1 QEC-M-01

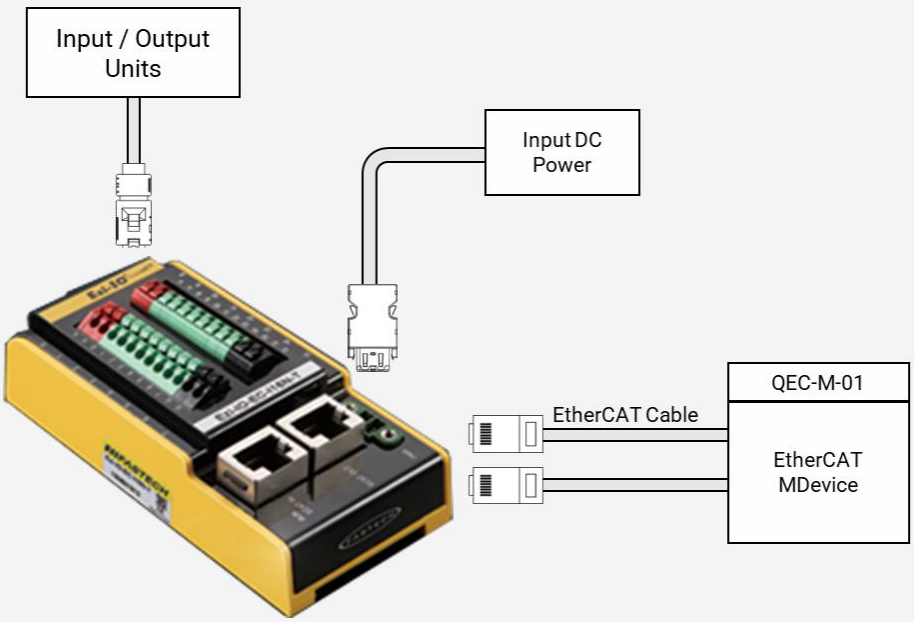
QEC EtherCAT MDevice.

1. Power Supply: Connect to Vs+/Vs- and Vp+/Vp- power supplies via EU terminals for 24V power.
2. EtherCAT Connection: Using the EtherCAT Out port (On the top side) connected to the EtherCAT In port of EtherCAT SubDevice via RJ45 cable.



1.2 Ezi-IO

Ezi-IO is a Fully Digital I/O control Unit from FASTECH mounted with an EtherCAT SubDevice Controller chip. This figure shows an example of an Ezi-IO connection.

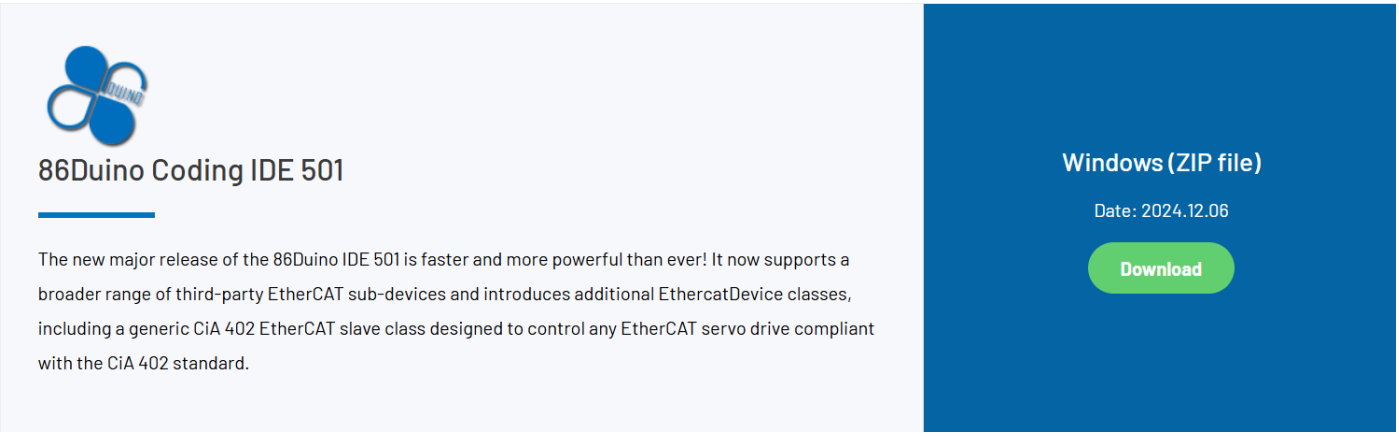


Ezi-IO

1. EtherCAT Connectivity
   - (EtherCAT Cable) Two RJ45 EtherCAT ports for real-time network communication.
   - Connects to the QEC-M-01 EtherCAT MDevice.
2. I/O Signal Connection
   - (Top Connector) Used to connect external input and output units such as sensors, switches, or relays.
3. Power Supply
   - (24V DC Input) Supplies operating power to the module through the power connector.

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.

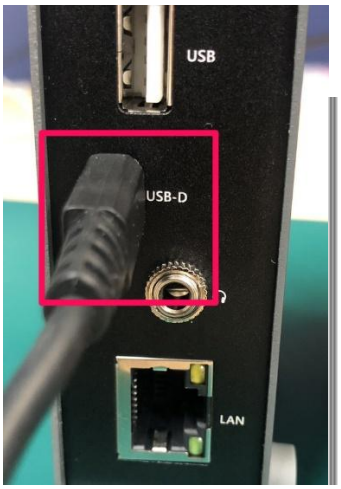
Note:
If Windows displays a warning, click Details once and then click the Continue Run button once.

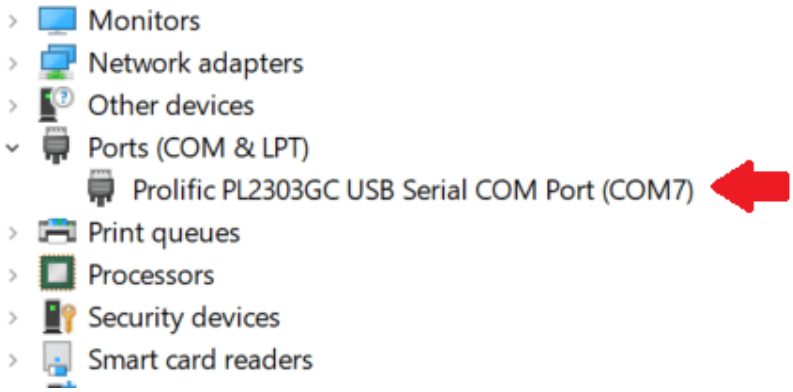
86Duino Coding IDE 501+ looks like below.

3. Connect to PC and set up the environment

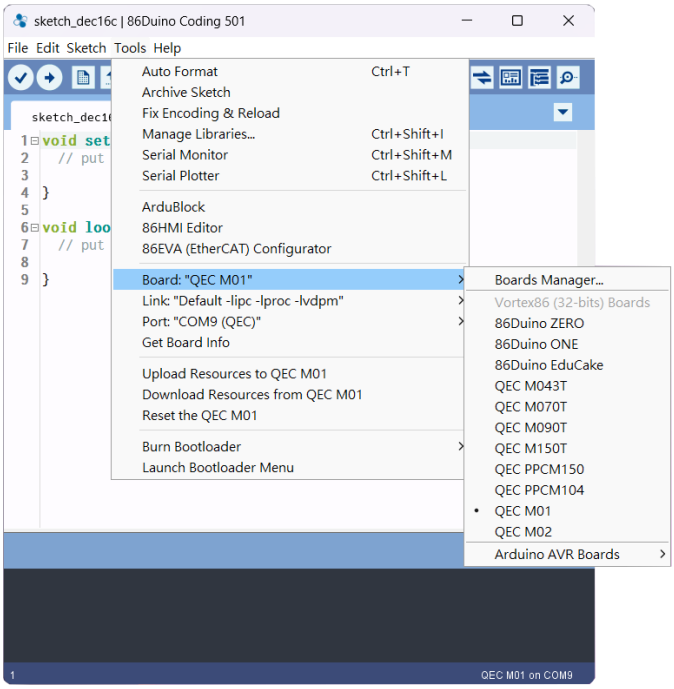
Follow the steps below to set up the environment:

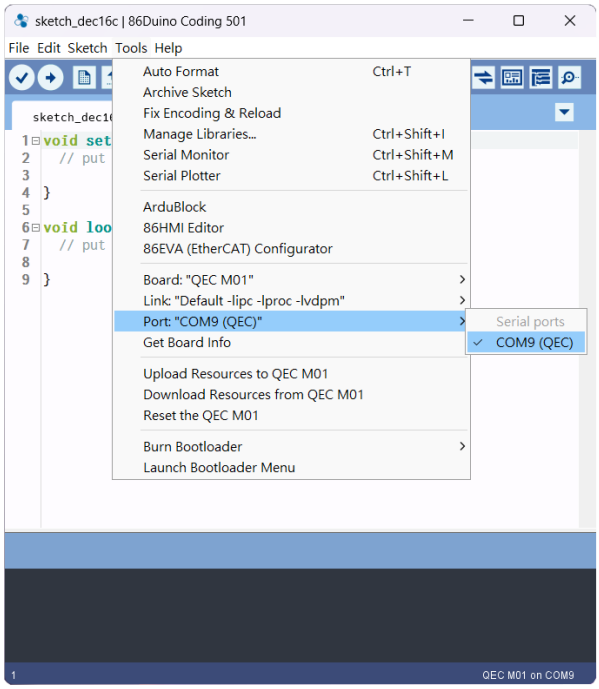
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “Device Manager” (select in the menu after pressing Win+X) ->” Ports (COM & LPT)” in your PC and expand the ports; you should see that the “Prolific PL2303GC USB Serial COM Port (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 MDevice model you use).
6. Select Port: In the IDE's menu, select Tools > Port and select the USB port to connect to the QEC MDevice (in this case, COM9 (QEC)).





4. Write code

The EtherCAT MDevice (QEC-M-01) and the Ezi-IO (EtherCAT Input/Output Module) 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 program sets the Ezi-IO:

- EtherCAT Cycle Time: 1 millisecond.
- EtherCAT Mode: ECAT_SYNC.

The EthercatMaster object (“mdevice”) represents the QEC-M-01, while the EthercatDevice_Generic object (“ezi_io”) represents the Ezi-IO module.

In this example, we connect the Ezi-IO’s Digital Output channel 0 to Digital Input channel 7, DO ch1 to DI ch6, DO ch2 to DI ch5, and so on.

A. In Setup Function:

In the setup() function, the system initializes serial communication, starts the EtherCAT MDevice, and attaches the I/O SubDevice.

Follow the steps below:

1. Initialize Serial Communication at a baud rate of 115200.
2. Start the EtherCAT MDevice. Call mdevice.begin() to initialize the QEC-M-01 to enter the PRE-OPERATIONAL state.
3. Attach the I/O Device. Call ezi_io.attach(0, mdevice) to bind the first EtherCAT SubDevice.
4. Start the EtherCAT MDevice. Use mdevice.start(1000000, ECAT_SYNC) to set the communication cycle to 1 ms (1,000,000 ns) and enter the OPERATIONAL state.

B. In Loop Function:

In the loop() function, the I/O device performs a basic blinking task by toggling output and monitoring the input state in a 500 ms cycle.

1. Output and Input Logic
    - Set **DO0** (Digital Output 0) to HIGH using pdoBitWrite(0, HIGH).
    - Read **DI7** (Digital Input 7) and display its status using pdoBitRead(7).
    - Wait 500 ms.
    - Set **DO0** to LOW, read **DI7** again, and repeat the process.

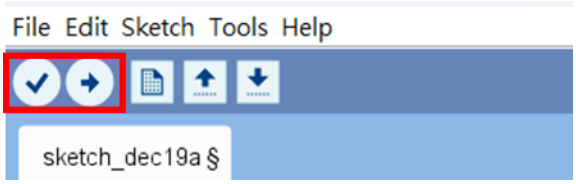
This demonstrates basic digital I/O control and monitoring using the QEC EtherCAT platform.

The example code is as follows:

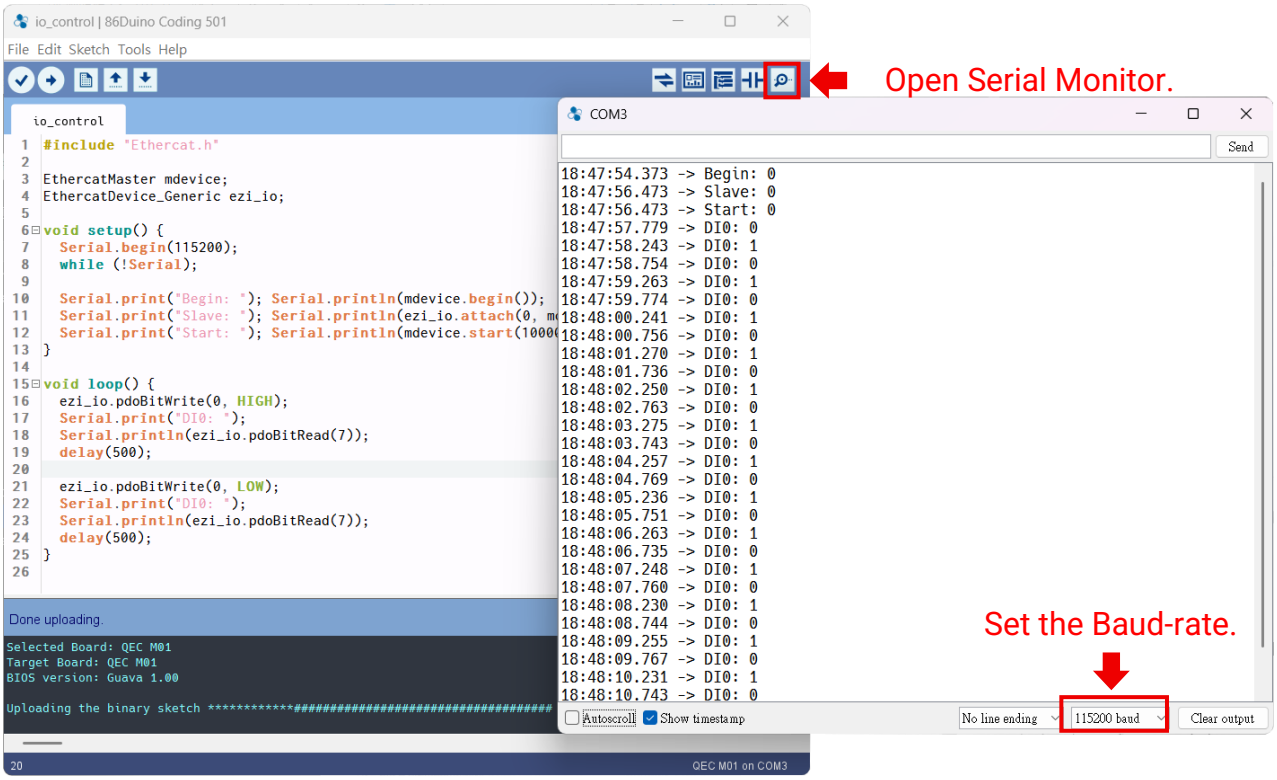
```
#include "Ethercat.h"

EthercatMaster mdevice;
EthercatDevice_Generic ezi_io;

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

  Serial.print("Begin: "); Serial.println(mdevice.begin());
  Serial.print("Slave: "); Serial.println(ezi_io.attach(0, mdevice));
  Serial.print("Start: "); Serial.println(mdevice.start(1000000, ECAT_SYNC));
}

void loop() {
  ezi_io.pdoBitWrite(0, HIGH);
  Serial.print("DI0: ");
  Serial.println(ezi_io.pdoBitRead(7));
  delay(500);

  ezi_io.pdoBitWrite(0, LOW);
  Serial.print("DI0: ");
  Serial.println(ezi_io.pdoBitRead(7));
  delay(500);
}
```

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.

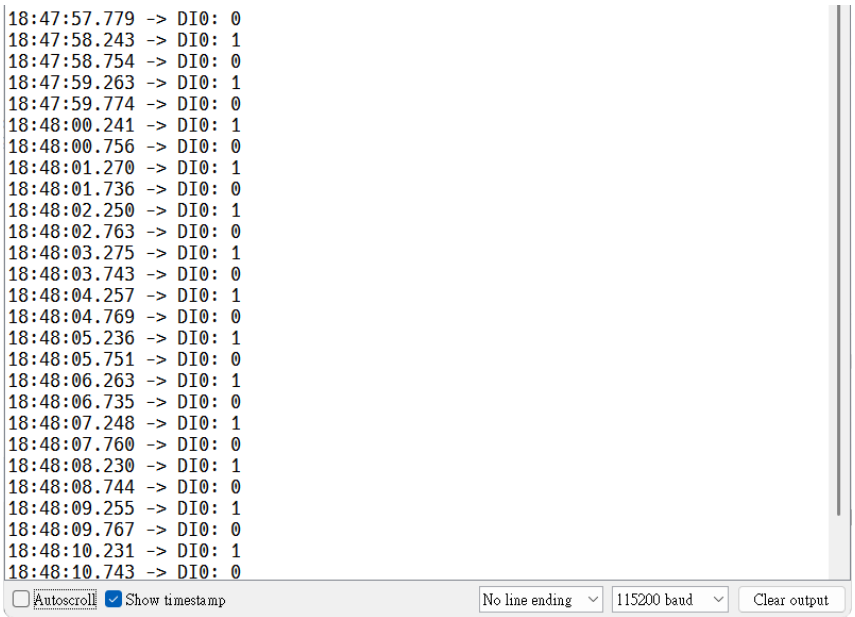


After you successfully upload the program to the QEC-M-01, you can open the Serial Monitor on 86Duino IDE. Please check the Serial baud rate is same as your setting.



If the EtherCAT communication config successful, Serial Monitor will print “Begin: 0”, “Slave: 0”, and “Start:0”, which mean all configurations are successful.

It will print the DI ch7 state to the serial monitor.



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 MDevice's environment is abnormal. As shown in the figure below, please try updating your QEC EtherCAT MDevice'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:

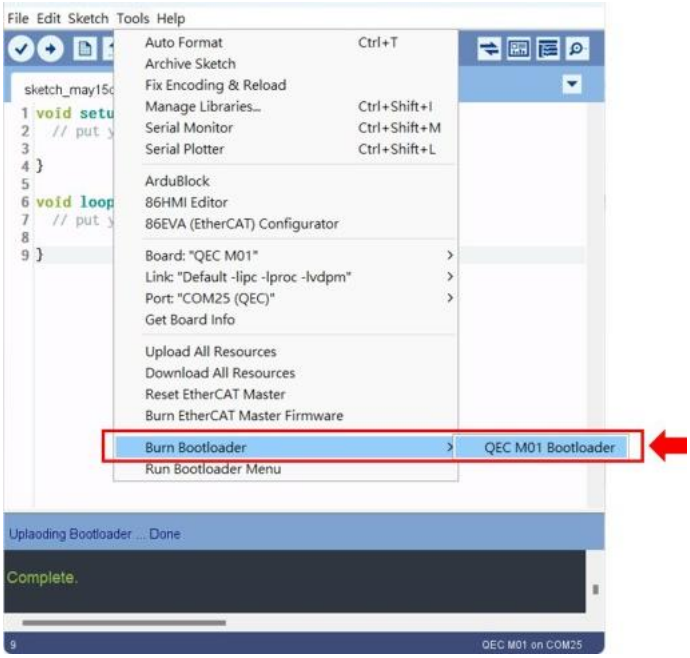
Step 1: Setting up QEC-M

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.

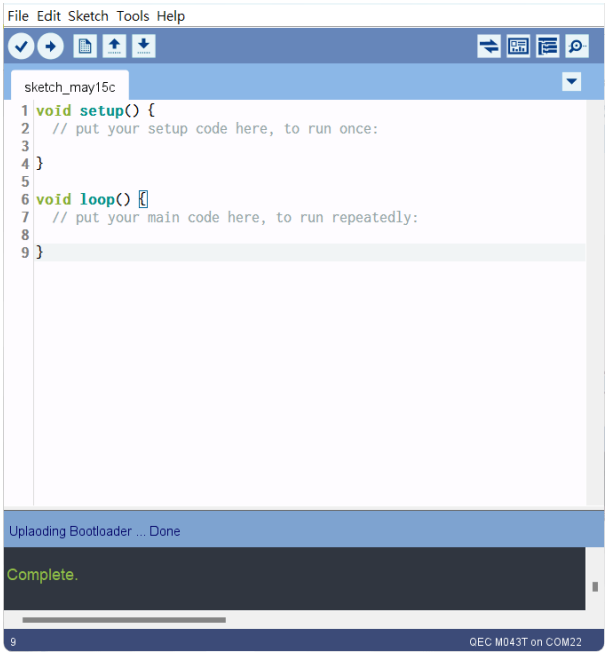
Step 2: Click “Burn Bootloader” button

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.