Start Guide

CM-TE66NA3G-N + CM-TE66PA3G-P
EtherCAT DI/DO

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

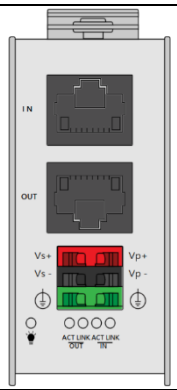
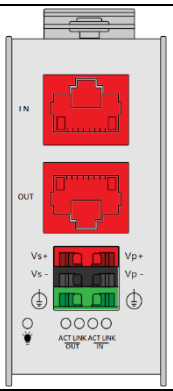
Date	Version	Description
2026/3/27	Version1.0	New release.

Preface

In this guide, we will show you how to use the EtherCAT MDevice **QEC-M-01**, the **CM-TE66NA3G-N** (EtherCAT 32 DO) and the **CM-TE66PA3G-P** (EtherCAT 32 DI).

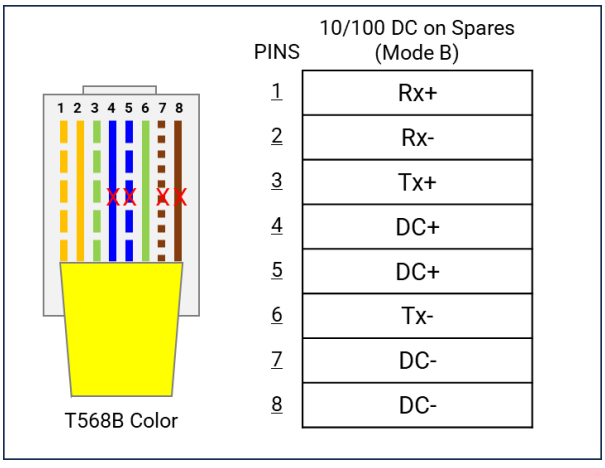
Notes QEC's PoE (Power over Ethernet)

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:



PINS	10/100 DC on Spares (Mode B)
1	Rx+
2	Rx-
3	Tx+
4	DC+
5	DC+
6	Tx-
7	DC-
8	DC-

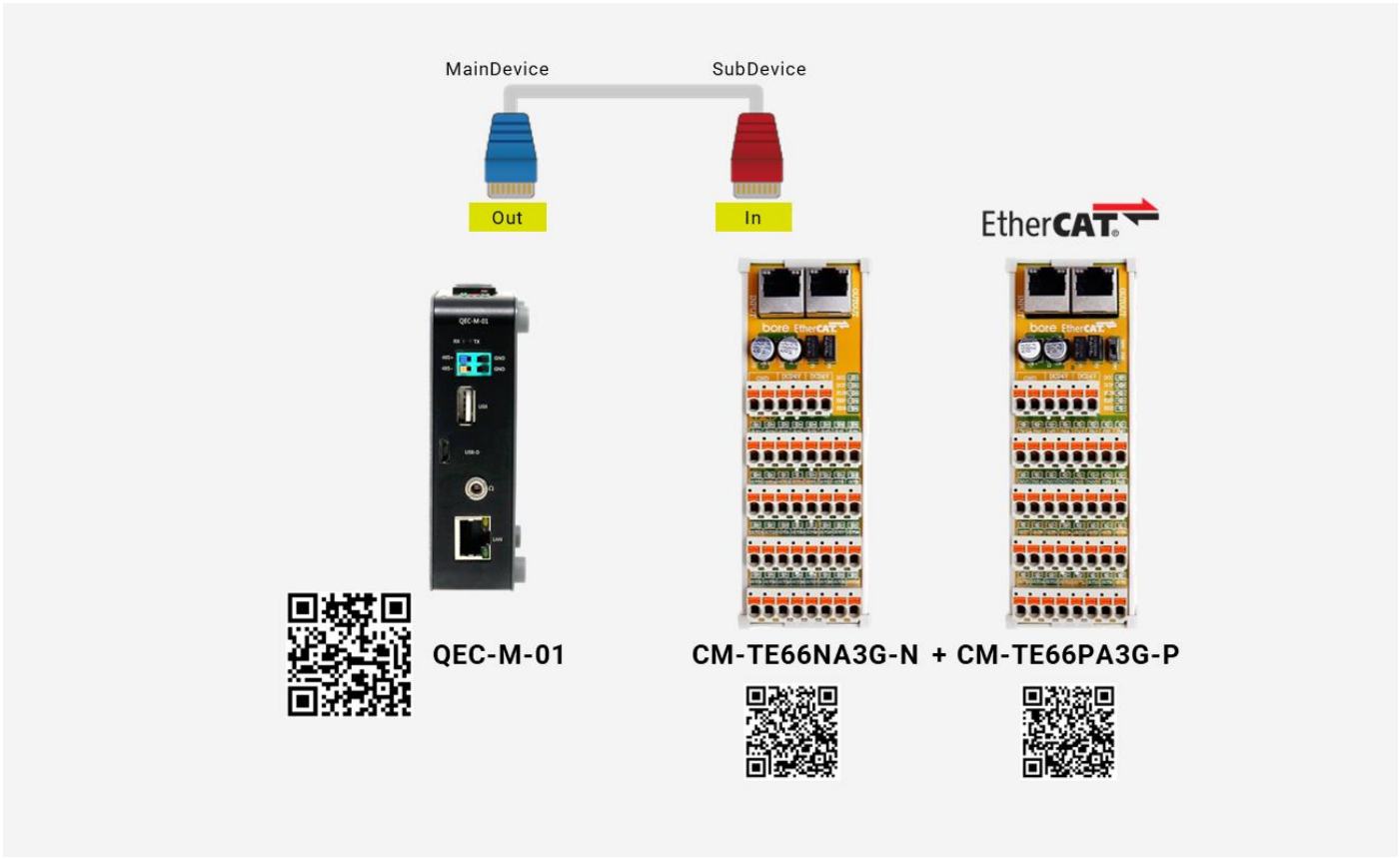
2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).

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

1. Connection and wiring hardware

The following devices are used here:

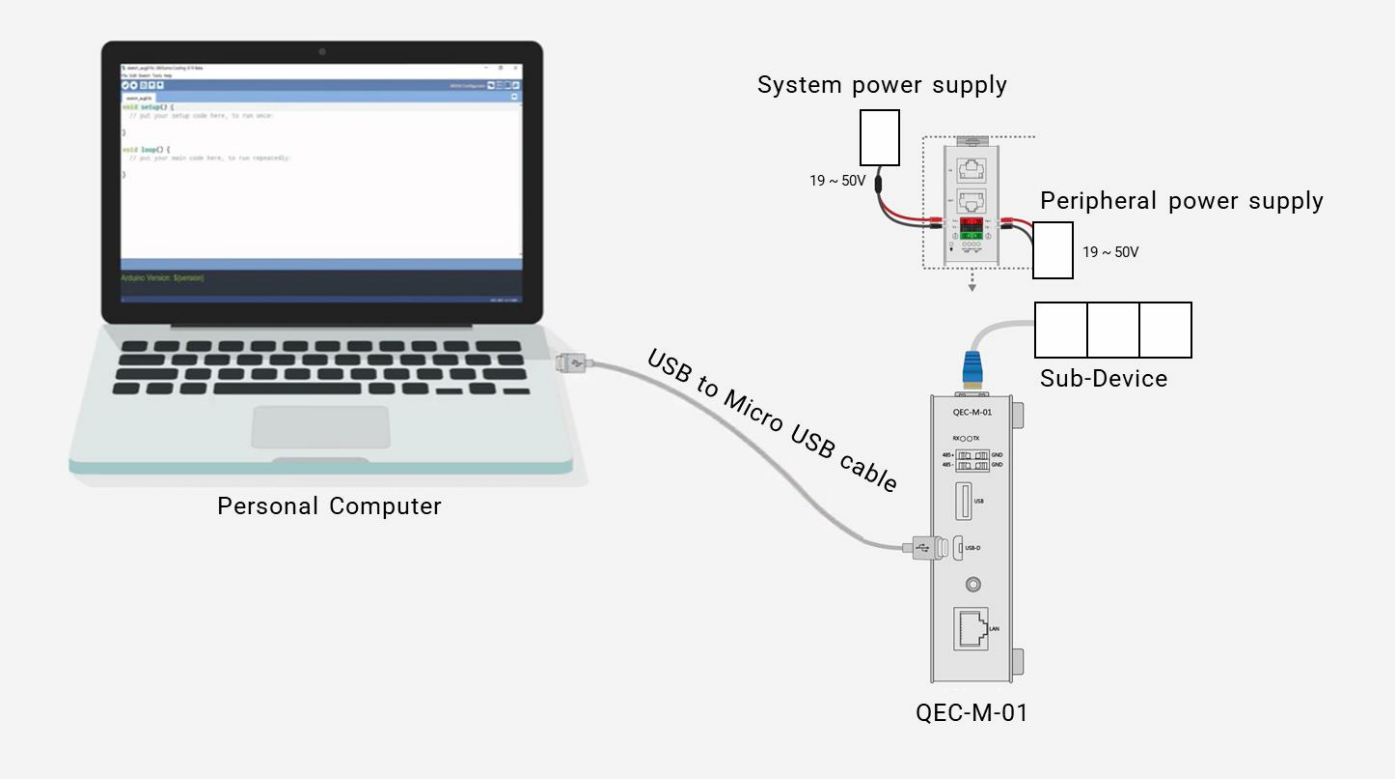
1. QEC-M-01 (EtherCAT MDevice)
2. Bore CM-TE66NA3G-N (EtherCAT 32 DO)
3. Bore CM-TE66PA3G-P (EtherCAT 32 DI)
4. 24VDC power supply & EU-type terminal cable & LAN cable



1.1 QEC-M-01

QEC EtherCAT MDevice, **QEC-M-01**.

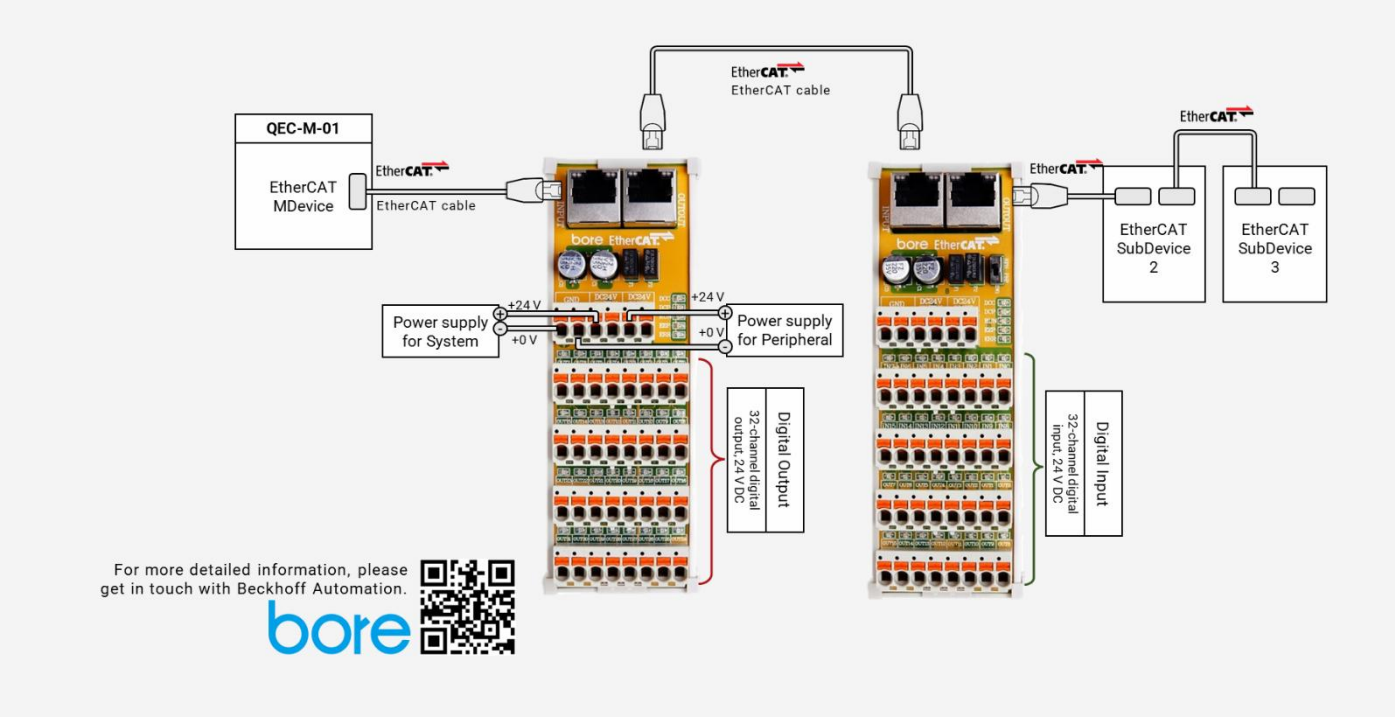
1. Power Supply: Connect to Vs+/- and Vp+/- power supplies via EU terminals for 24V power.
2. EtherCAT Connection: Using the EtherCAT Out port (On the top side) connected to the EtherCAT In port of EtherCAT SubDevice via RJ45 cable.



1.2 Bore 32 DI/DO

The **Bore CM-TE66NA3G-N** (EtherCAT 32 DO) and **CM-TE66PA3G-P** (EtherCAT 32 DI) are 32-point EtherCAT digital I/O modules. They can be used as EtherCAT SubDevices in a QEC EtherCAT network for basic digital input sensing and output control.

The diagram below shows a typical wiring example using a **QEC MDevice** (for example, **QEC-M-01**) together with **Bore EtherCAT DI/DO modules**.

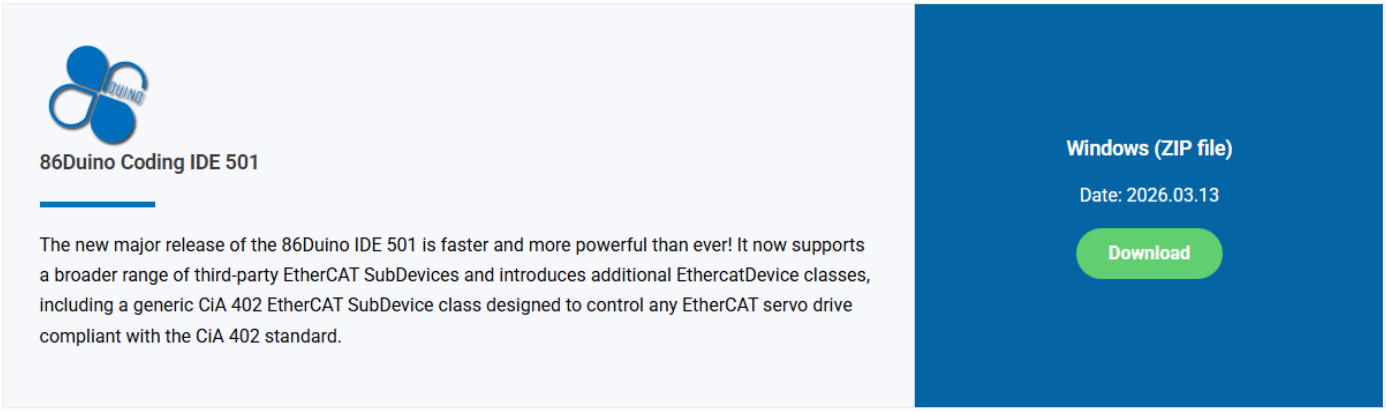


Connections are grouped by function:

- EtherCAT
  - IN: Connect to the EtherCAT port of the QEC MDevice or the previous SubDevice.
  - OUT: Connect to the next EtherCAT SubDevice, if any.
- Power & Grounding
  - Provide 24 VDC power for module operation and field I/O.
  - Connect 0 V (GND) properly to ensure stable communication.
- Digital Inputs (CM-TE66PA3G-P, DI)
  - Input channels receive external digital signals from field devices.
  - Input signals should be referenced to the module input common or 0 V (GND).
- Digital Outputs (CM-TE66NA3G-N, DO)
  - Output channels are used to drive external loads.
  - A suitable external 24 VDC power supply is required for output operation.
  - Ensure the load wiring matches the module's output type and electrical rating.
- Indicators / LEDs
  - EtherCAT LEDs indicate power status and EtherCAT communication activity.
  - Channel LEDs indicate the ON/OFF status of each DI or DO point.

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.

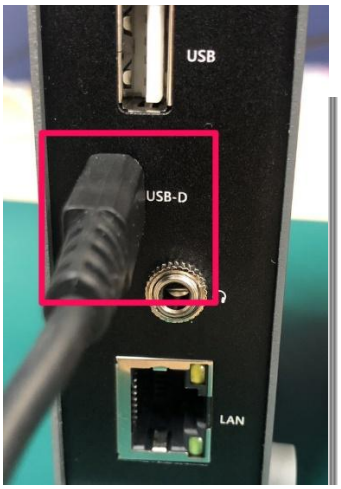
*__Note__: If Windows displays a warning, click Details once and then click the Continue Run button.

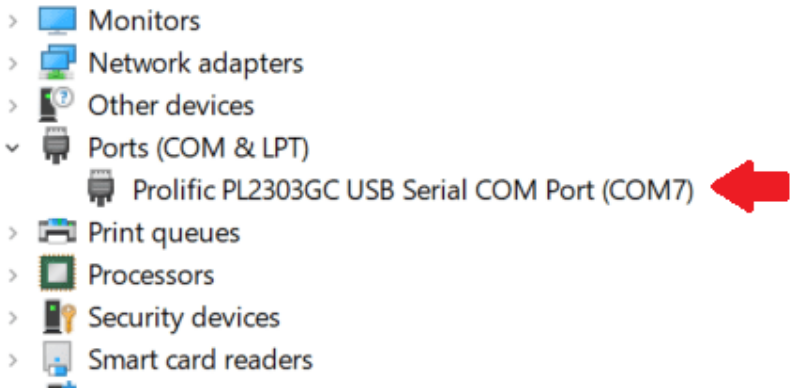
86Duino Coding IDE 501+ looks like below.

3. Connect to PC and set up the environment

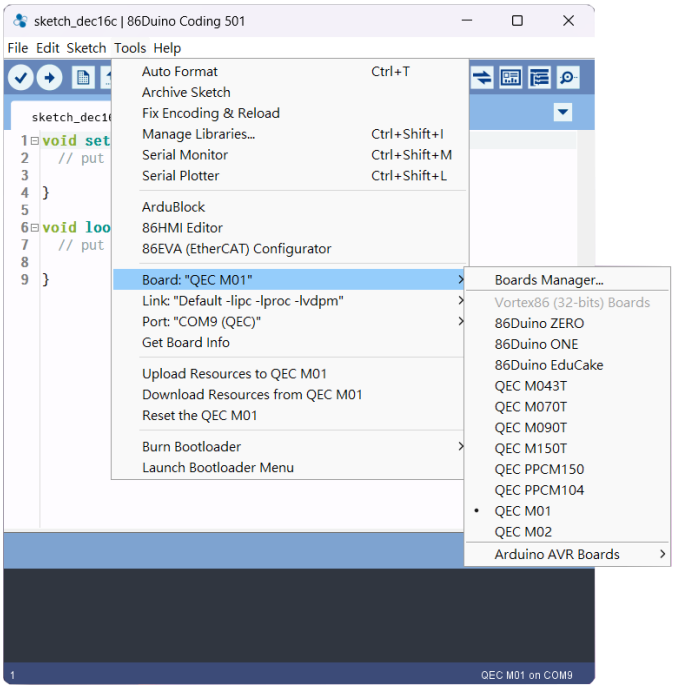
Follow the steps below to set up the environment:

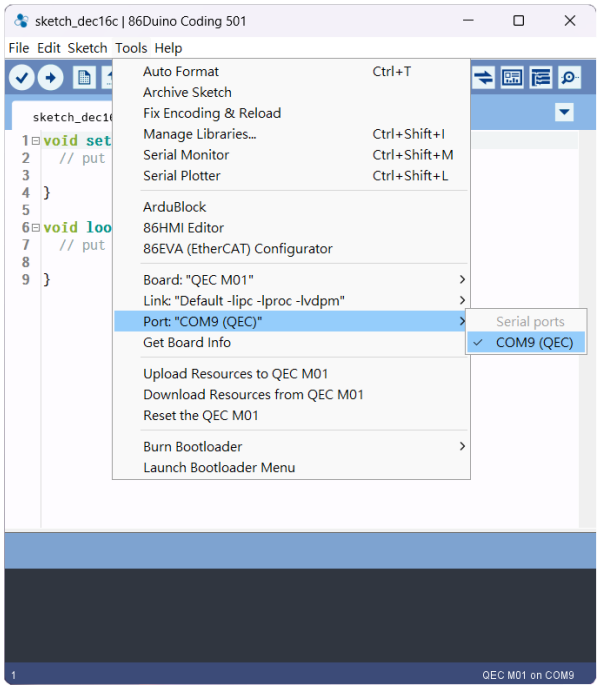
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “**Device Manager**” (select in the menu after pressing Win+X) ->” **Ports (COM & LPT)**” in your PC and expand the ports; you should see that the “**Prolific PL2303GC USB Serial 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-M01**” (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 **Bore CM-TE66NA3G-N** (EtherCAT 32 DO) and **CM-TE66PA3G-P** (EtherCAT 32 DI) can be configured and programmed using the EtherCAT library in the **86Duino IDE**, together with a **QEC MDevice** such as **QEC-M-01**.

An Arduino-based program is mainly divided into two parts: `setup()` and `loop()`.
The `setup()` function is executed once after power-on or reset and is used for initialization. The `loop()` function runs continuously after initialization and is used for cyclic control logic.

Before cyclic I/O access starts, the EtherCAT network should be initialized correctly and brought to the OP (Operational) state. In general, the EtherCAT state transition follows this sequence:
Pre-OP → Safe-OP → OP

The following example demonstrates a basic EtherCAT DI/DO test:

- EtherCAT cycle time: 1 ms
- EtherCAT mode: ECAT_SYNC

The `EthercatMaster` object (`master`) represents the QEC MDevice.
The `EthercatDevice_Generic` objects represent the connected EtherCAT SubDevices, `Slave0` represents the Bore CM-TE66NA3G-N (32 DO), and `Slave1` represents the Bore CM-TE66PA3G-P (32 DI).

In this example, the program periodically toggles one output bit of the CM-TE66NA3G-N, then reads back one input bit of the CM-TE66PA3G-P after a short delay. This is used to verify that the input state changes correctly in response to the output signal.
For testing, the selected DO channel and DI channel can be connected by external wiring, or the DI channel can be driven by an external input signal.

A. In Setup Function:
The `setup()` function is used to initialize Serial communication, start the EtherCAT master, attach the EtherCAT SubDevices, and begin cyclic EtherCAT communication.

1. Initialize Serial Communication
    - Start serial communication at a baud rate of 115200 for debug messages.
2. Start the EtherCAT MDevice (QEC-M-01)
    - Call `master.begin()` to initialize the EtherCAT MDevice and enter the EtherCAT state machine (typically to PRE-OP / initialization stage).
3. Attach the SubDevice (`EthercatDevice_Generic` Class)
    - Use `attach(node, master)` to attach the first EtherCAT SubDevice:
      `Slave0.attach(0, master)` → SubDevice 0: Bore DO
      `Slave1.attach(1, master)` → SubDevice 1: Bore DI

4. Start EtherCAT and Switch to OPERATIONAL
    - Call `master.start(1000000, ECAT_SYNC)` to start cyclic communication and switch to OPERATIONAL state.
    - This sets the cycle time to 1 ms (1,000,000 ns) and enables ECAT_SYNC mode.

B. In Loop Function:

The `loop()` function runs continuously after initialization.
In this example, it periodically turns one DO bit ON and OFF, and reads one DI bit to verify the result.

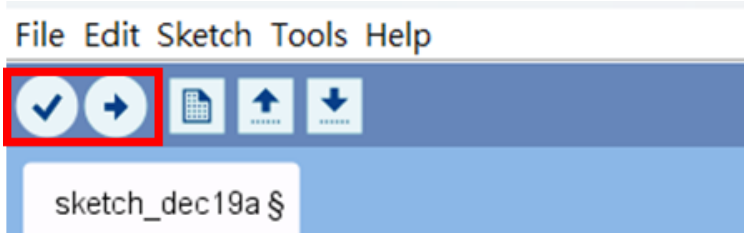
1. Write a PDO Output Bit (ON)
    - Use `pdoBitWrite(i, HIGH)` to set PDO bit i (DO 0 to 31) to HIGH.
    - Wait 10 ms (`delay(10)`) to allow the DI to reflect the DO change and the EtherCAT input PDO data to refresh before reading.
2. Read a PDO Input Bit and Print Result
    - Use `pdoBitRead(i)` to read PDO bit i (DI 0 to 31).
    - Print the value to the Serial Monitor: “i: <value>”.
3. Hold Time
    - Wait 1000 ms (`delay(1000)`) before changing the state.
4. Write a PDO Output Bit (OFF) and Repeat
    - Use `pdoBitWrite(i, LOW)` to set PDO bit i (DO 0 to 31) to LOW.
    - Use `delay(10)` to wait again for the DI update.
    - Use `pdoBitRead(i)` to read PDO bit i (DI 0 to 31).
5. Code Logic Summary
    - `pdoBitWrite(n, HIGH/LOW)`: drives PDO bit n (commonly used for DO mapping).
    - `pdoBitRead(n)`: reads PDO bit n (commonly used for DI mapping).
    - The example toggles the output every second and logs the corresponding input bit state.

The example code is as follows:

```
#include "Ethercat.h"

EthercatMaster master;
EthercatDevice_Generic Slave0;
EthercatDevice_Generic Slave1;

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

  Serial.print("master begin: ");
  Serial.println(master.begin());
```

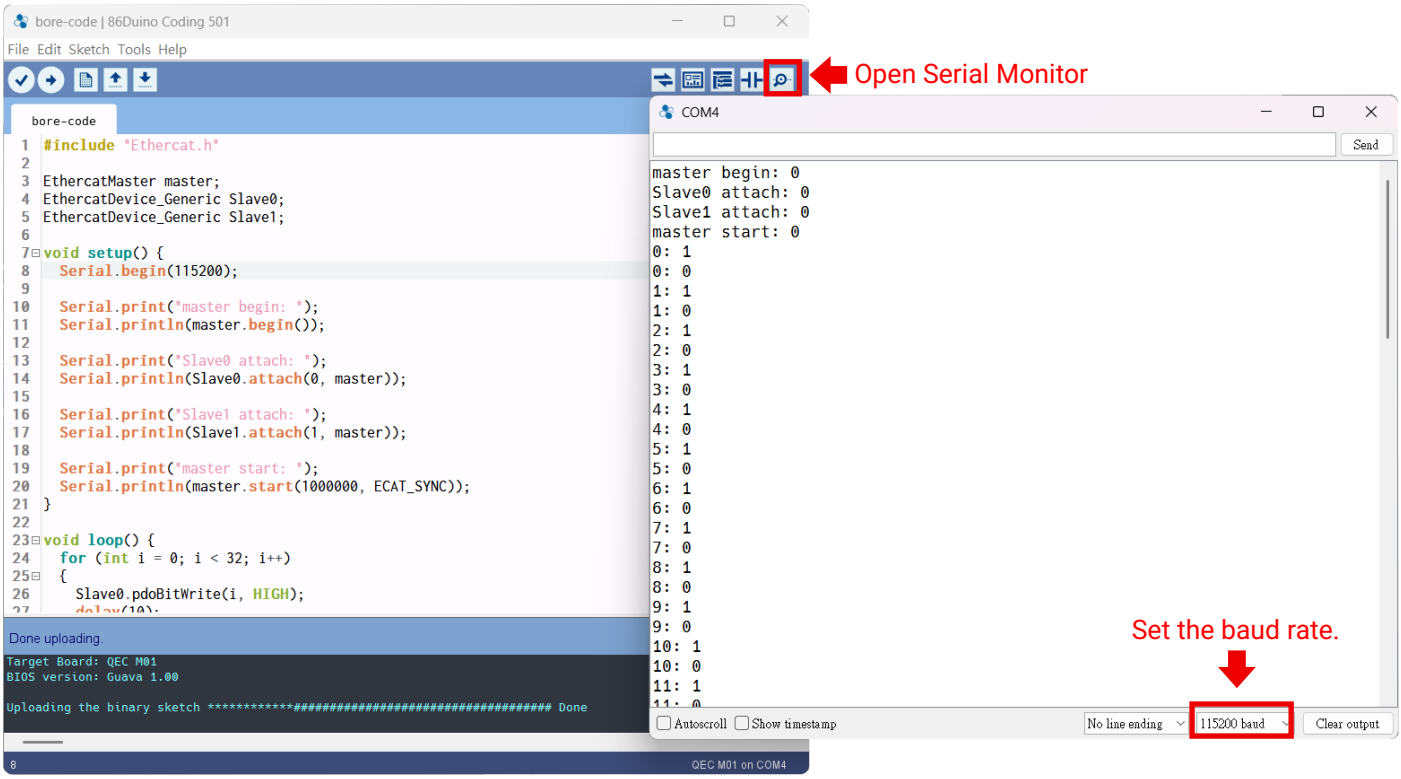
```
  Serial.print("Slave0 attach: ");
  Serial.println(Slave0.attach(0, master));

  Serial.print("Slave1 attach: ");
  Serial.println(Slave1.attach(1, master));

  Serial.print("master start: ");
  Serial.println(master.start(1000000, ECAT_SYNC));
}

void loop() {
  for (int i = 0; i < 32; i++)
  {
    Slave0.pdoBitWrite(i, HIGH);
    delay(10);
    Serial.print(i); Serial.print(": ");
    Serial.println(Slave1.pdoBitRead(i));
    delay(1000);

    Slave0.pdoBitWrite(i, LOW);
    delay(10);
    Serial.print(i); Serial.print(": ");
    Serial.println(Slave1.pdoBitRead(i));
  }
}
```

***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.

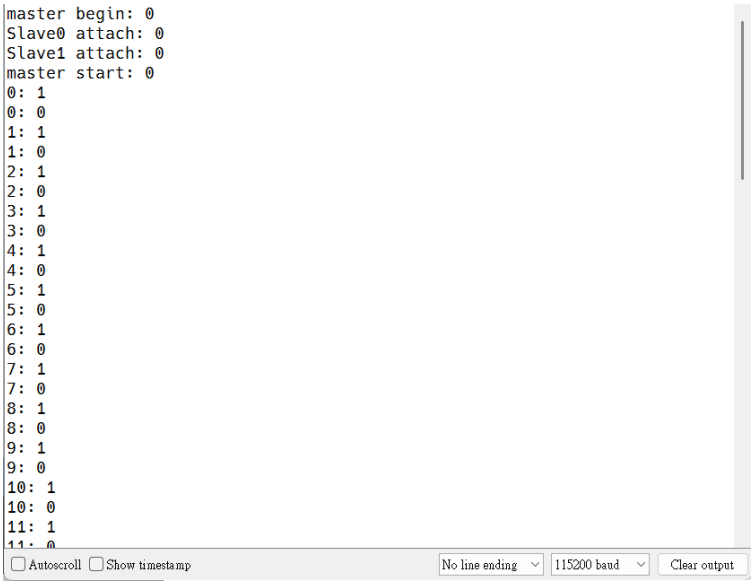
Please make sure that the Serial baud rate matches your program setting.



If EtherCAT initialization is successful, the Serial Monitor will print 0 as the return value for each initialization step.

```
master begin: 0
Slave0 attach: 0
Slave1 attach: 0
master start: 0
```

The Serial Monitor will also print the Bore DI values.



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:

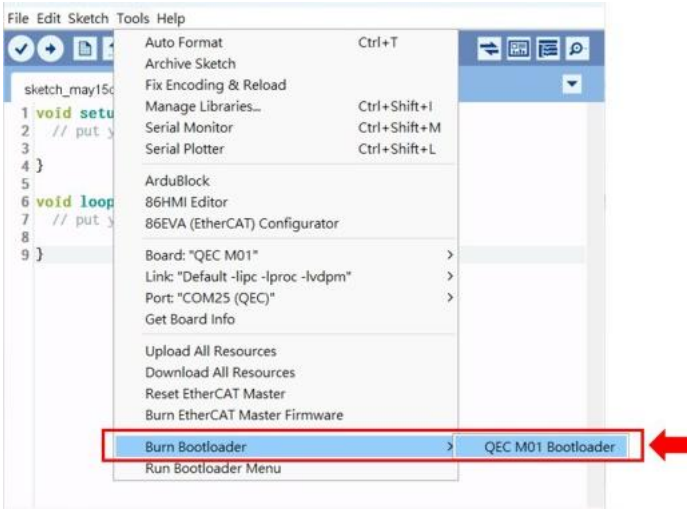
Step 1: Setting up QEC-M

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.

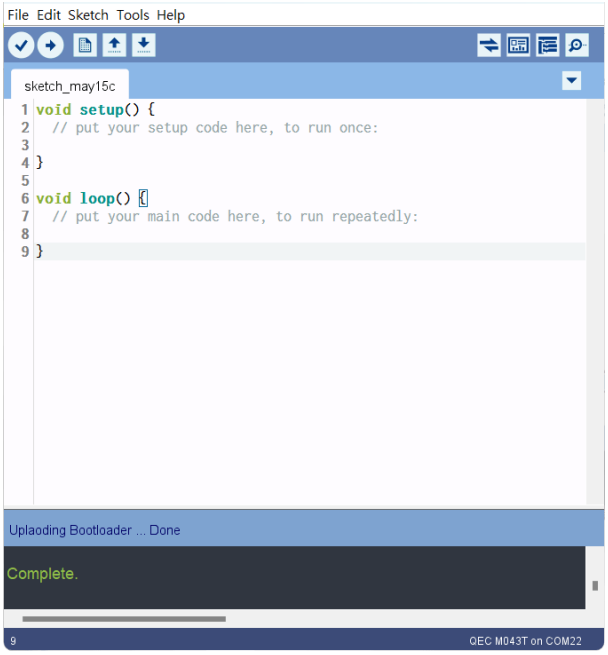
Step 2: Click “Burn Bootloader” button

After connecting to your QEC-M device, go to “**Tools**”> “**Burn Bootloader**”.
The name of the currently selected QEC-M device will be displayed.
Clicking on it will start the update process, which will take approximately 5 to 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.