Start Guide

QEC-R11CFFG : EtherCAT Remote Digital I/O, Analog I/O, and Serial with 86EVA

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

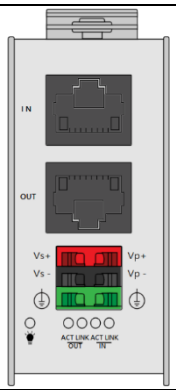
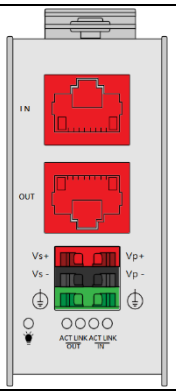
Date	Version	Description
2025/9/28	Version1.0	New Release.

Preface

In this guide, we will show you how to use the EtherCAT MDevice QEC-M-01 and the QEC-R11CFFG series (EtherCAT Compound I/O module, with DIO + AIO + Serial COM port).

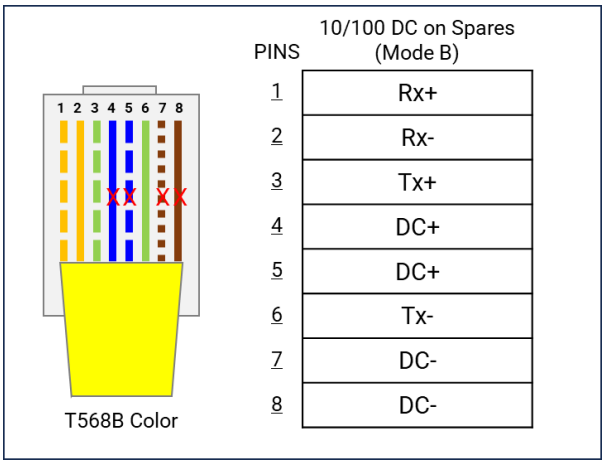
Notes QEC's PoE (Power over Ethernet)

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

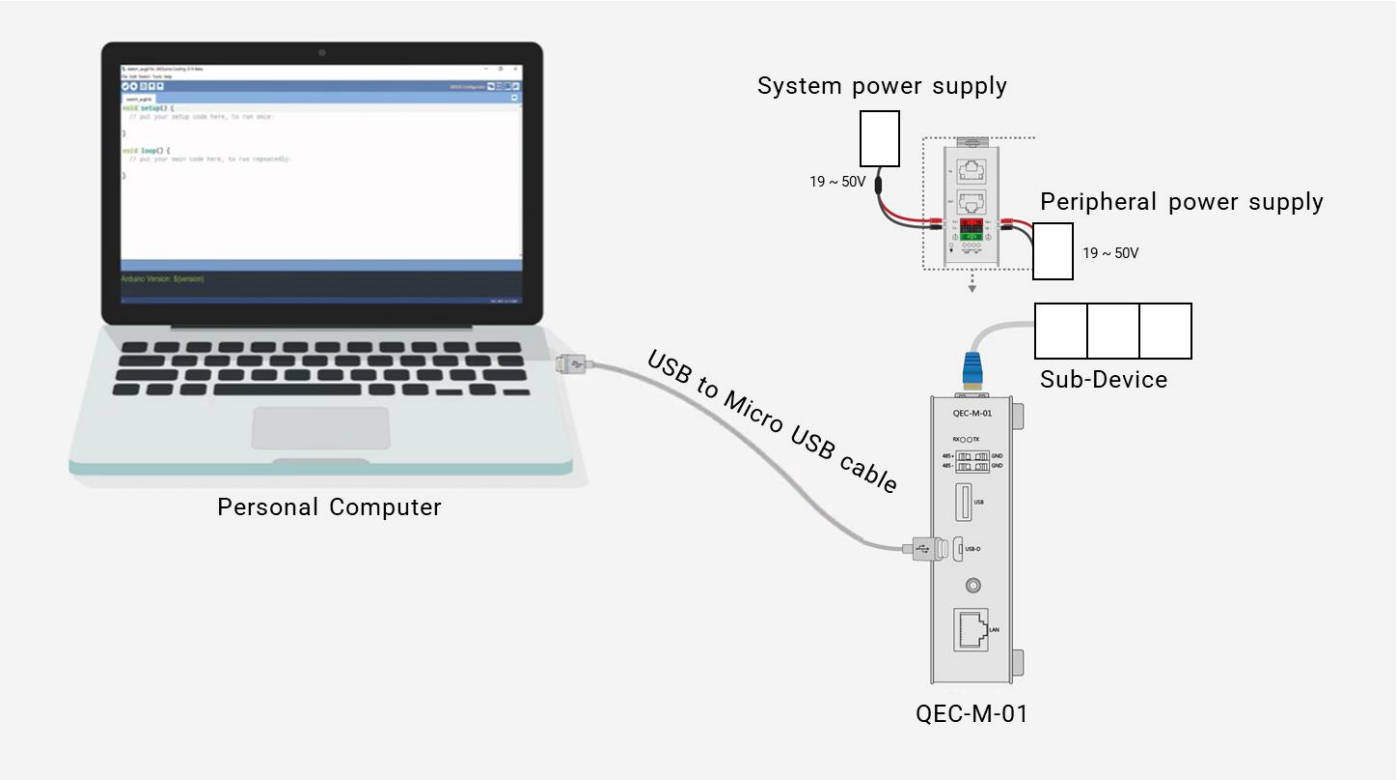
The following devices are used here:

1. QEC-M-01 (EtherCAT MDevice)
2. QEC-R11CFFG series (EtherCAT Compound I/O module, with DIO + AIO + Serial COM port)
3. 24VDC power supply & EU-type terminal cable & LAN cable

1.1 QEC-M-01

QEC EtherCAT MDevice.

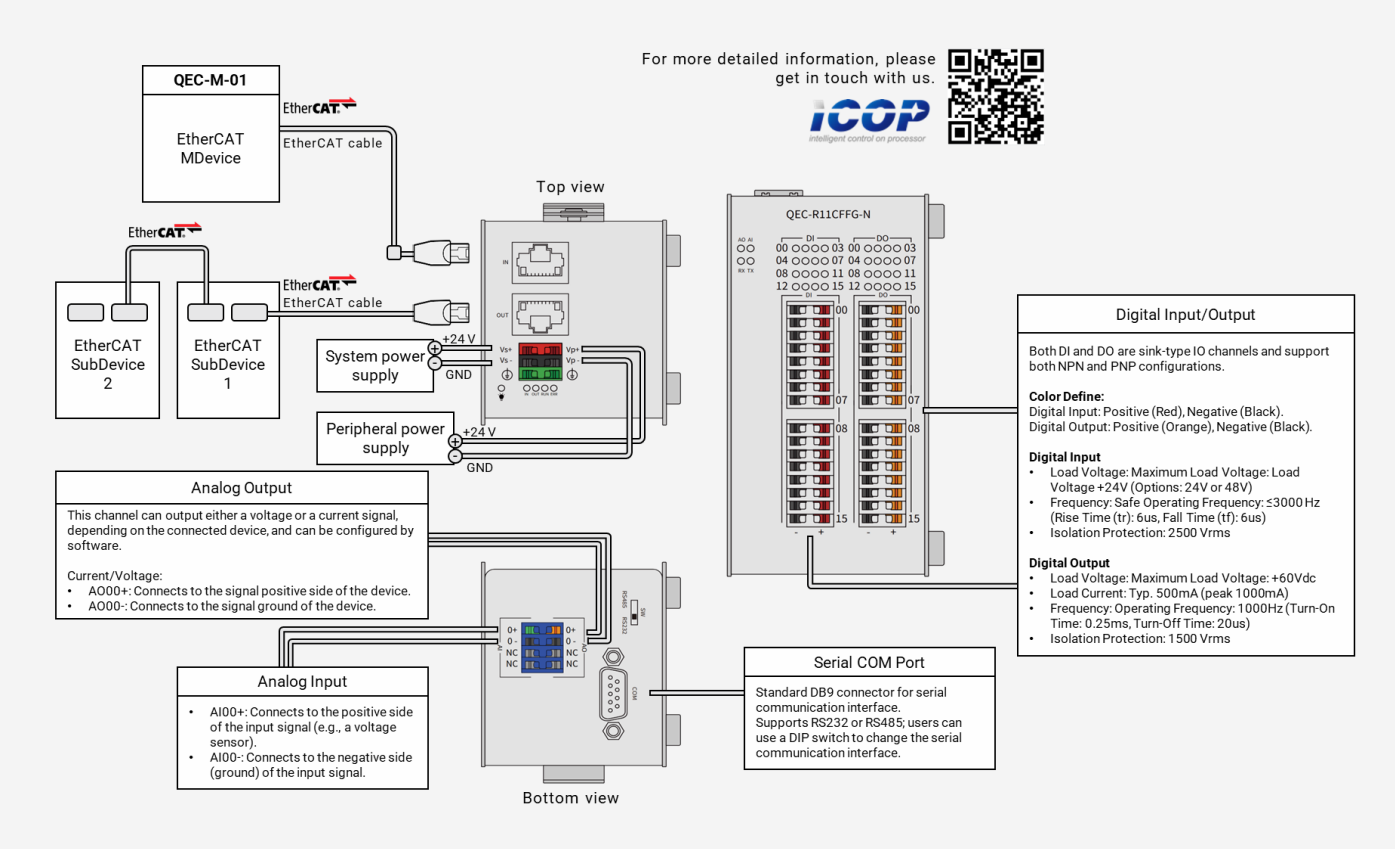
1. Power Supply: Connect to Vs+/Vs- and Vp+/Vp- power supplies via EU terminals for 24V power.
2. EtherCAT Connection: Using the EtherCAT Out port (On the top side) connected to the EtherCAT In port of EtherCAT SubDevice via RJ45 cable.



1.2 QEC-R11CFFG

The **QEC-R11CFFG** is an EtherCAT SubDevice that combines isolated 32-ch Digital I/O (DI16/DO16), analog I/O slots (voltage/current per configuration), and a serial COM port (RS-232/RS-485 selectable).

The diagram shows a typical wiring example with a QEC MDevice and an EtherCAT network.



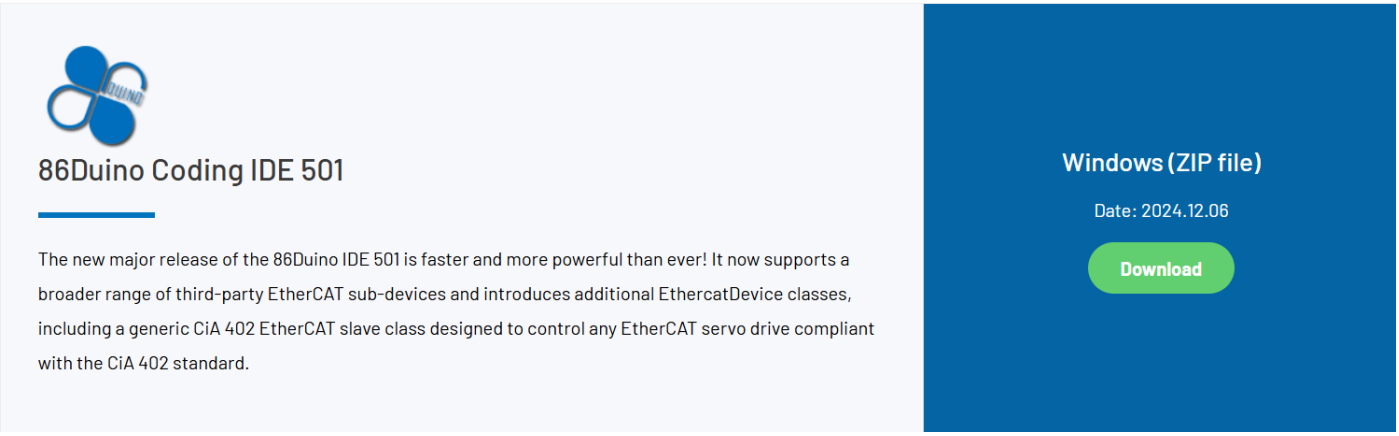
Connections are grouped by function:

- EtherCAT: MDevice → IN; OUT for daisy-chain.
- Power & Grounding:
  - VS+/VS−: system power +24 V/GND
  - VP+/VP−: field I/O power +24 V/GND
- Digital I/O:
  - DI 16 (sink type): supports NPN/PNP; safe op ≤ 3 kHz, tr/tf 6 µs; input +24 V (options 24/48 V).
  - DO 16: up to +60 Vdc, typ. 500 mA (peak 1 A); 1 kHz (Ton 0.25 ms / Toff 20 µs). Add flyback/snubber for inductive loads.
  - Terminal groups: DI00–07 / DI08–15, DO00–07 / DO08–15, each with + / −.
  - Isolation: DI 2500 Vrms; DO 1500 Vrms.
- Analog I/O:
  - AO voltage/current (software-selectable): AO00+ to device positive, AO00− to return.
  - AI: AI00+ to sensor positive, AI00− to sensor return.

- Serial COM: DB9; DIP switch selects RS-232/RS-485.
  For RS-485, use termination/bias and a shared reference.
- Indicators: PWR / RUN / ERR / L/A (see later section).
- Color legend: DI + Red / − Black; DO + Orange / − Black.

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.

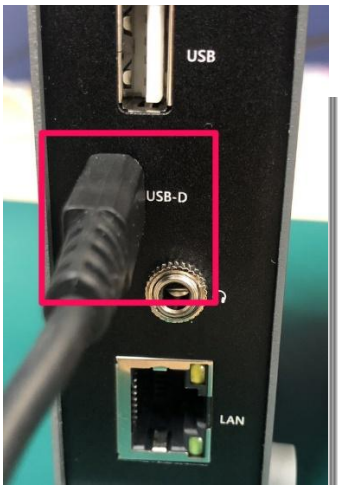
*__Note__: If Windows displays a warning, click Details once and then click the Continue Run button once.

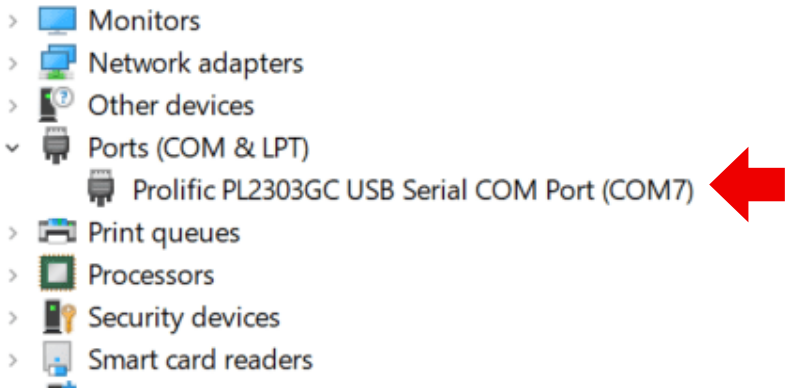
86Duino Coding IDE 501+ looks like below.

3. Connect to PC and set up the environment

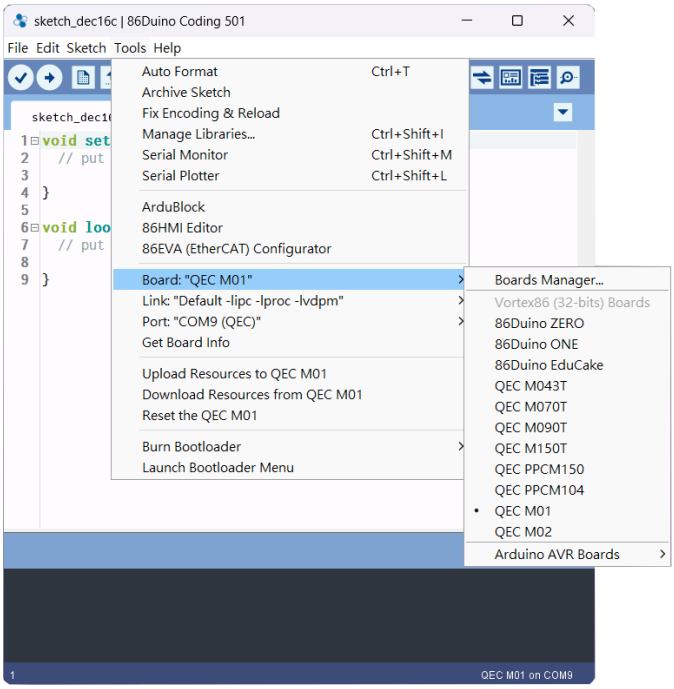
Follow the steps below to set up the environment:

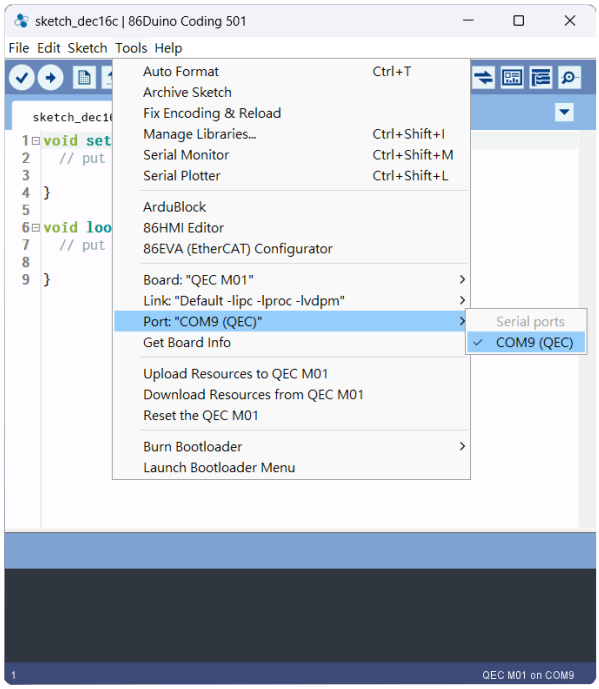
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “**Device Manager**” (select in the menu after pressing Win+X) ->” **Ports (COM & LPT)**” in your PC and expand the ports; you should see that the “**Prolific PL2303GC USB Serial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

This example shows how to operate the EtherCAT MDevice (QEC-M-01) and the QEC-R11CFFG (EtherCAT Compound I/O module, with DIO + AIO + Serial COM port) through the 86Duino IDE's graphical low-code programming tool, 86EVA.

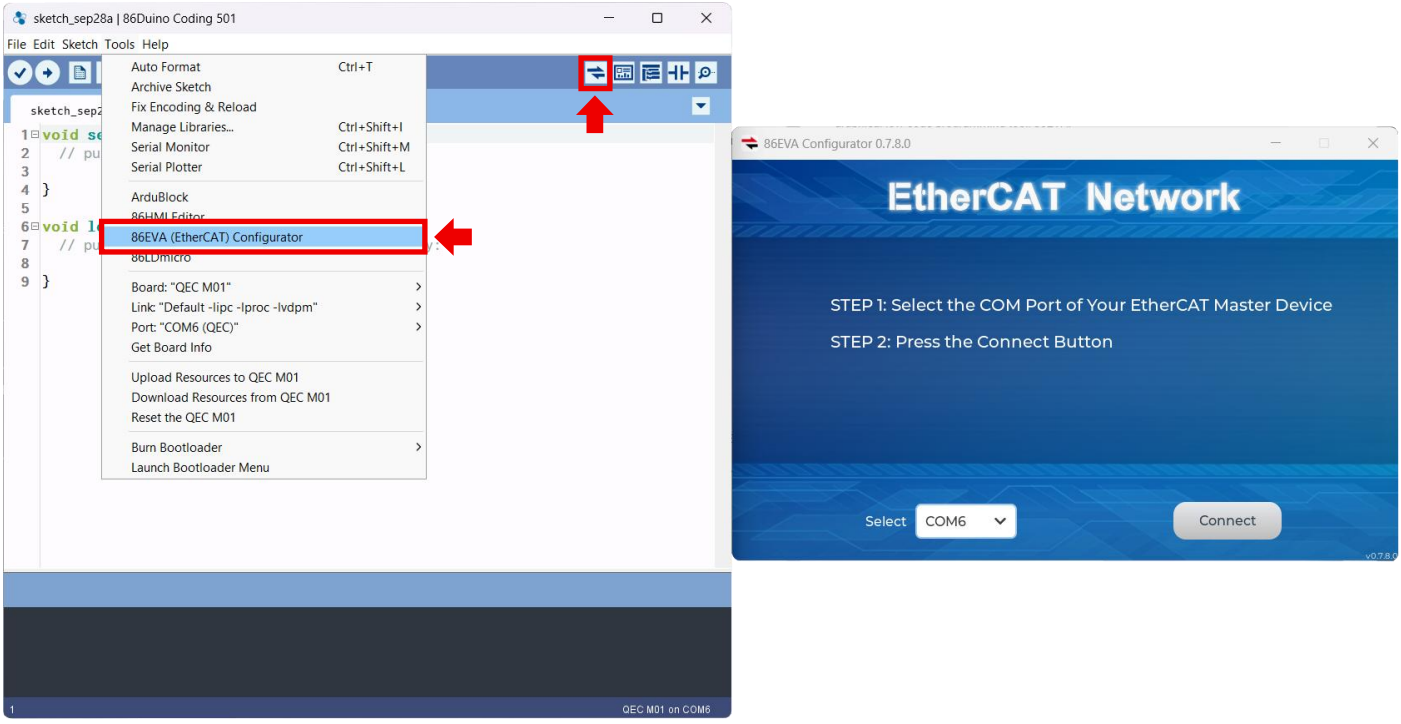
Software Tools Description:

- 86EVA (EVA, EtherCAT-Based Virtual Arduino):
  is a graphical EtherCAT configuration tool based on the EtherCAT Library in the 86Duino IDE and is one of the development kits for 86Duino.

We will present the Digital I/O, Analog I/O, and Serial COM Port (RS232/RS485) usage in the following section by steps.

Step 1: Turn on 86EVA and scan

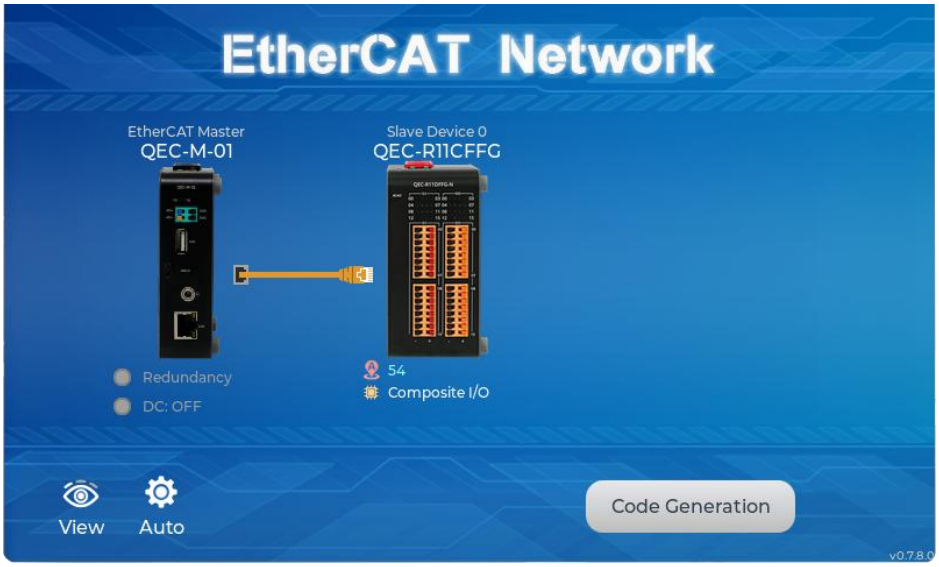
The 86EVA tool can be opened via the following buttons.



Please select the correct COM port and then click the "Connect" button.

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

The connected devices will be displayed after the EtherCAT network has been scanned.



Step 2: Set the parameters

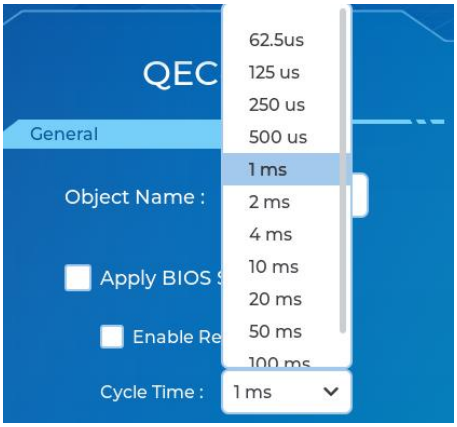
Press twice on the scanned device image to enter the corresponding parameter setting screen.

Step 2.1: QEC-M-01

Press twice on the image of the QEC-M-01 to see the parameter settings.

Please check the following configures.

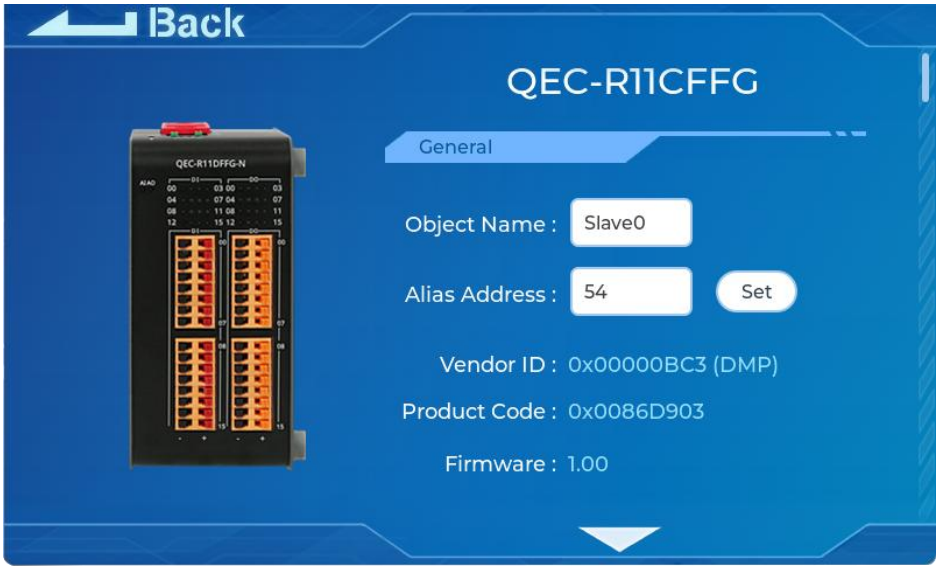
1. Turn off the "**Apply BIOS Settings**".
2. Select "**1ms**" to the Cycle Time.



Click "**Back**" in the upper left corner to return.

Step 2.2: QEC-R11CFFG

Press twice on the image of the QEC-R11CFFG to see the parameter settings.



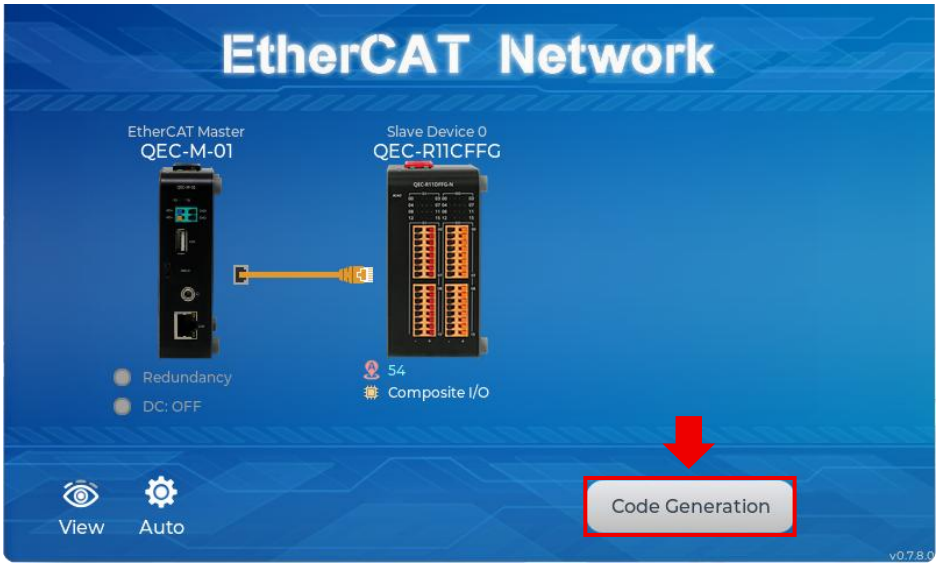
The page will show the Object Name, Alias Address, Vendor ID, Product Code, Virtual Arduino Mapping, and Virtual Servo Configuration parameters.

Please change the Object Name to “**slave0**”.
It’ll appear a keyboard after you click the Object Name.

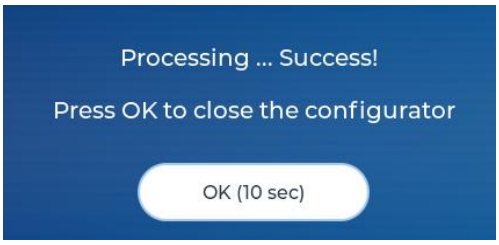
Click "**Back**" in the upper left corner to return.

Step 3: Generate the code

Once you've set your device's parameters, go back to the home screen and press the "Code Generation" button in the bottom right corner.

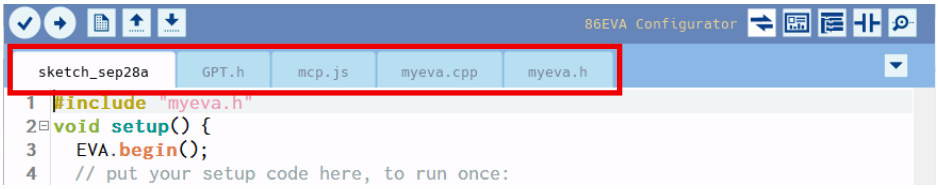


When you're done, double-click the OK button to turn off 86EVA, or it will close in 10 seconds.



The generated code and files are as follows:

- sketch_sep10b: Main Project (.ino, depending on your project name)
- 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: Write the code

Before operating the EtherCAT network, you must configure it once. The process should be from Pre-OP to OP mode in EtherCAT devices. 86EVA will automatically handle the EtherCAT State Machine in the background.

The programming code from 86EVA are set as the following by default:

- QEC-R11CFFG module: EthercatDevice_QECR11CFFG object.
- EtherCAT mode: ECAT_SYNC.

And here is the setting by users:

- EtherCAT Cycle time: 1 millisecond.
- The EthercatMaster object (“EcatMaster”) represents the QEC-M-01, while the EthercatDevice_QECR11CFFG object (“slave0”) represents the QEC-R11CFFG module.

We will present the Digital I/O, Analog I/O, and Serial COM Port (RS232/RS485) usage in the following section by steps.

Step 4.1: Digital I/O

In this section, we will read digital input DI00 (e.g., button) and mirror its state to digital output DO00 (e.g., LED/load).

A. In Setup Function:

In the `setup()` function, initialize communication and configure the bring the EtherCAT network up to OP mode. Follow the steps below:

1. Initialize Serial Communication
    - Start serial communication at a baud rate of 115200.
2. Start the 86EVA
    - Use the `EVA.begin()` function to start and initialize the EtherCAT network.

B. In Loop Function:

In the `loop()` function, read DI00 and mirror its state to DO00 continuously.

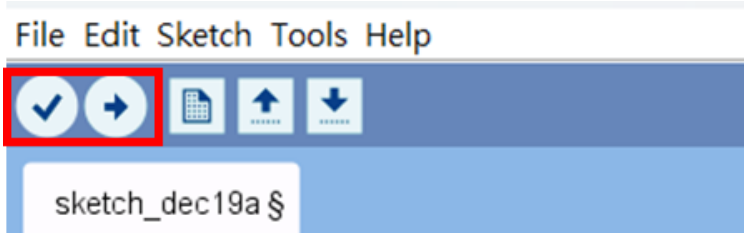
1. Logic
    - If `digitalRead(0)` is HIGH, set `digitalWrite(0, HIGH);` otherwise set `digitalWrite(0, LOW)`.
2. Code Logic Summary
    - Use `digitalRead(n)` to read DI channel n (0–15).
    - Use `digitalWrite(n, HIGH/LOW)` to drive DO channel n (0–15).

The example code is as follows:

```
#include "myeva.h"

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

void loop()
{
  if (slave0.digitalRead(0) == HIGH)
  {
    slave0.digitalWrite(0, HIGH);
  }
  else
  {
    slave0.digitalWrite(0, LOW);
  }
```

```
}
```

*__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.

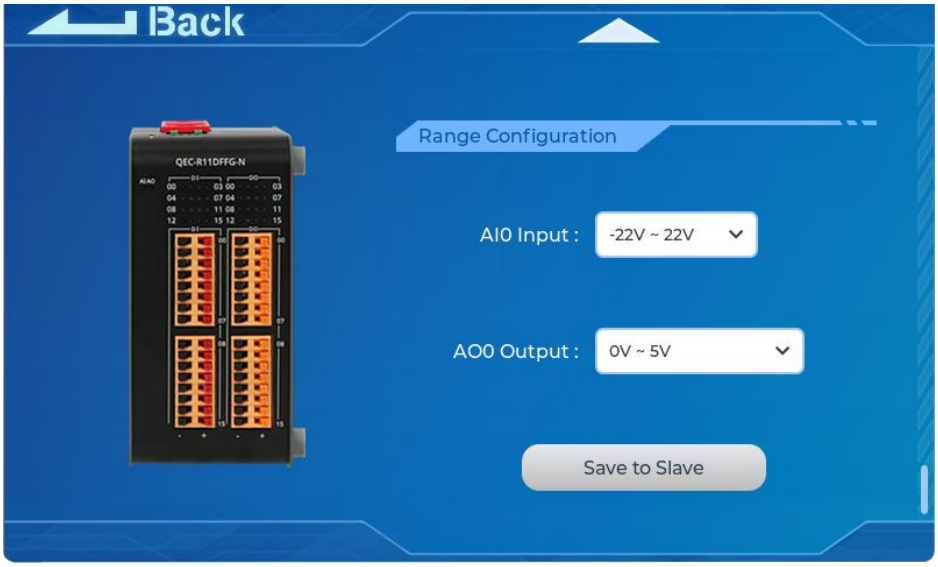


Step 4.2: Analog I/O

In this section, we will read analog input AI00 (voltage) and sweep analog output AO00 from 0 V → 5 V in 1-V steps every second.

Before we program, we need to configure the analog Input and output mode using the 86EVA.
It's an advanced setting of Step 2.2: QEC-R11CFFG.

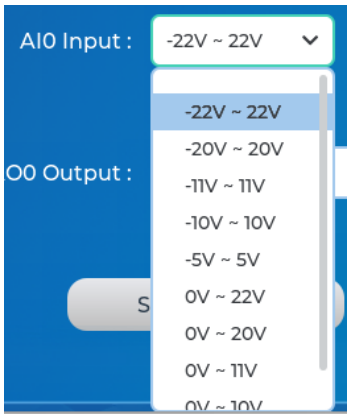
Go down to the "**Range Configuration**" area.



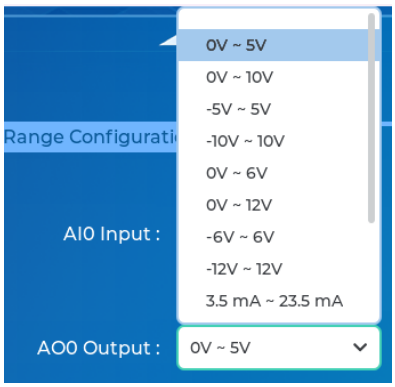
Users can choose different analog input and analog output ranges.
In this guide, please select AI0 Input's analog input range to "**-22V ~ 22V**" (default), and AO0 Output's analog output range to "**0V ~ 5V**" (default).

As shown in the picture below.

1. Analog Input (AI0 Input)



2. Analog Output (AO0 Output)



Then, click the "**Save to Slave**" button to save the range configuration to your QEC-R11CFFG.

After successfully configuring the SubDevice, it'll display a "**Success**" window, and ask you to reset your SubDevice to apply the new settings.

Now, we can begin writing the program.

A. In Setup Function:

In the `setup()` function, initialize communication, enter OP, and configure the analog ranges. Follow the steps below:

1. Initialize Serial Communication
    - Start serial communication at a baud rate of 115200.
2. Start the 86EVA
    - Use the `EVA.begin()` function to start and initialize the EtherCAT network.

B. In Loop Function:

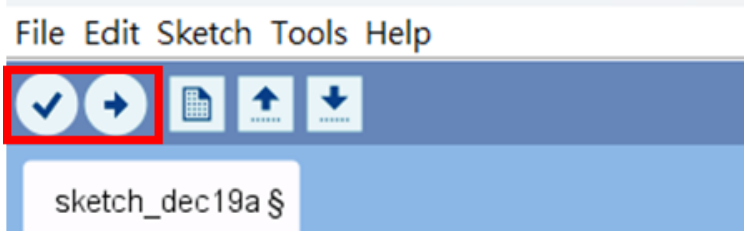
In the `loop()` function, print the measured AI voltage and step the AO voltage.

1. Logic
    - Read AI00 with `voltageRead(0)` and print the value.
    - Write AO00 with `voltageWrite(0, i)`, where i sweeps 0 → 5 → 0 (in volts), updating every 1 s.
2. Code Logic Summary
    - Use `voltageRead(ch)` to get the analog input voltage of channel ch.
    - Use `voltageWrite(ch, volts)` to set the analog output voltage of channel ch.

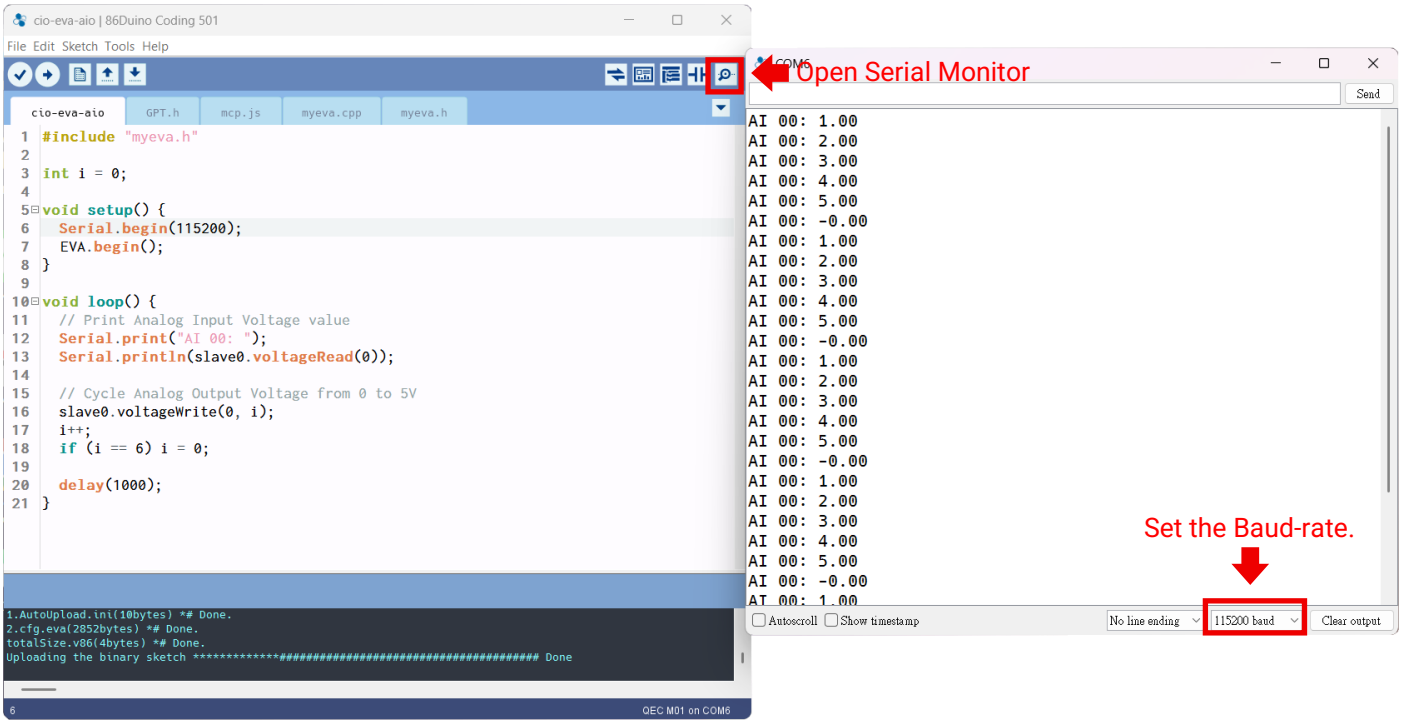
The example code is as follows:

```
#include "myeva.h"

int i = 0;

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

void loop() {
  // Print Analog Input Voltage value
  Serial.print("AI 00: ");
  Serial.println(slave0.voltageRead(0));

  // Cycle Analog Output Voltage from 0 to 5V
  slave0.voltageWrite(0, i);
  i++;
  if (i == 6) i = 0;

  delay(1000);
}
```

***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 the 86Duino IDE. Please check that the Serial baud rate is the same as your setting.



It will print the analog input AI00 value (Voltage) to the serial monitor.

```
AI 00: -0.00
AI 00: -0.00
AI 00: 1.00
AI 00: 2.00
AI 00: 3.00
AI 00: 4.00
AI 00: 5.00
AI 00: -0.00
AI 00: 1.00
```

Step 4.3: Serial COM Port

In this section, we forward bytes from the USB Serial Monitor to COM1 (RS-232) on QEC-R11CFFG. In this test, no data is expected to return from RS-232, so the code only transmits and optionally prints anything that happens to arrive.

Wiring

- Use the front DB9 connector (see diagram for pinout).
- Connect the external device's TX ↔ RX, RX ↔ TX, and GND ↔ GND.
- Keep cable length reasonable; share a common reference (GND) with the device.

A. In Setup Function:

In the `setup()` function, initialize communication, enter OP, and configure COM1. Follow the steps below:

1. Initialize Serial Communication
    - Start serial communication at a baud rate of 115200.
2. Start the 86EVA
    - Use the `EVA.begin()` function to start and initialize the EtherCAT network.
3. Configure the COM Port (RS-232)
    - Use the `uartSetBaud(COM1, 115200)` function to setup COM1 to baud rate 115200.
    - Use the `uartSetFormat(COM1, SERIAL_8N1)` function to set COM1 to SERIAL_8N1 format.

B. In Loop Function:

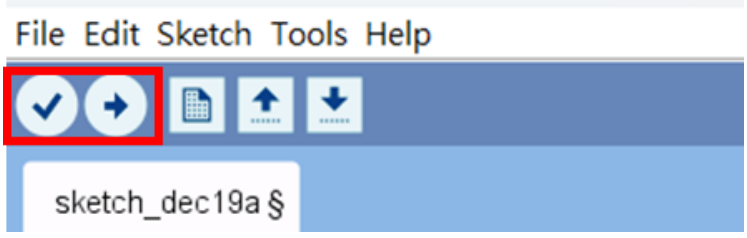
In the `loop()` function, keep EtherCAT/COM services running and forward any USB Serial byte from the Serial Monitor to COM1.

1. Logic
    - Call `update()` each cycle to update the UART queues on EtherCAT/mailbox.
    - If Serial Monitor has a byte, read it and send it to COM1 with `uartWrite(COM1, byte)`.
    - Since this test has no RS-232 reply, the code still tries a non-blocking `uartRead(COM1)` and prints only if a byte is present.
2. Code Logic Summary
    - Use `Serial.available()` to check the RX FIFO, and `Serial.read()` to get data from Serial.
    - Use `slave0.uartRead(COM1)` / `uartWrite(COM1)` to receive/transmit bytes to the Serial COM port on QEC-R11CFFG.
    - Keep remote serial I/O in `loop()` to do EtherCAT mailbox service.

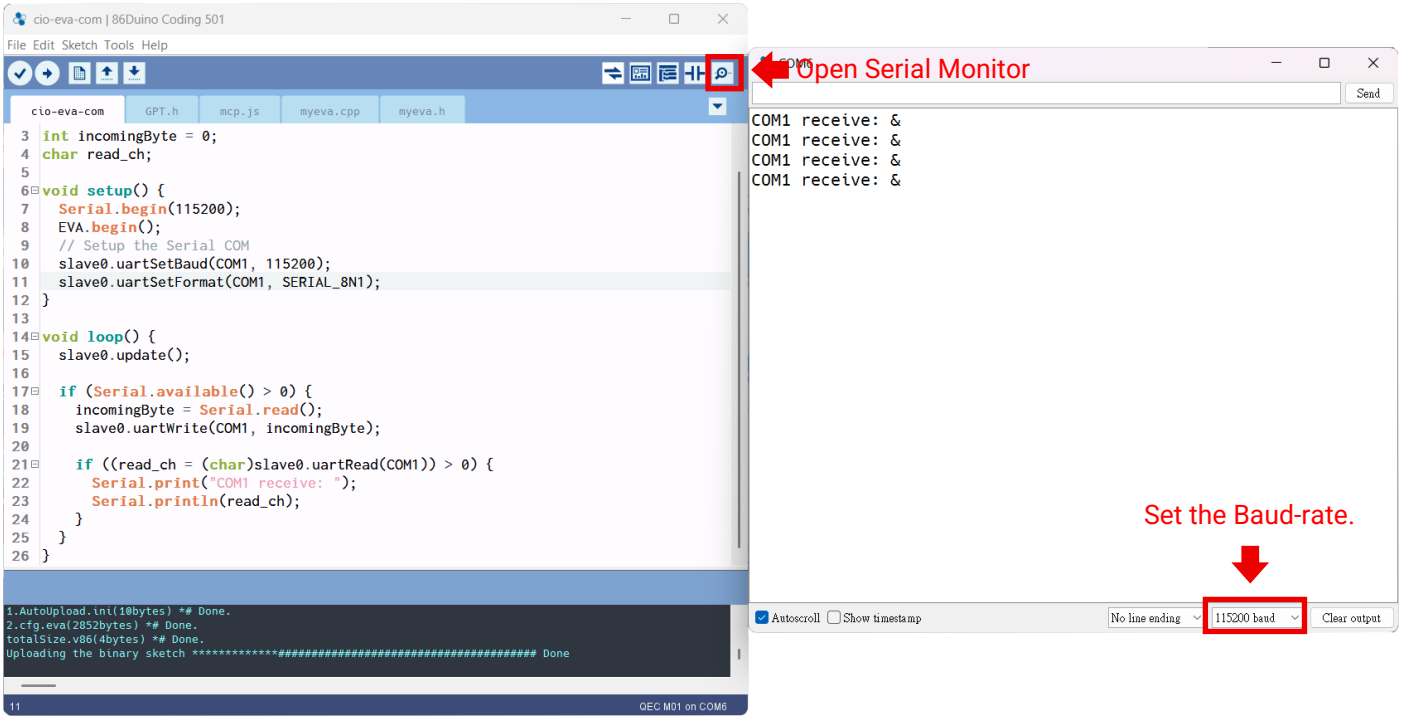
The example code is as follows:

```
#include "myeva.h"

int incomingByte = 0;
char read_ch;

void setup() {
  Serial.begin(115200);
  EVA.begin();
  // Setup the Serial COM
  slave0.uartSetBaud(COM1, 115200);
  slave0.uartSetFormat(COM1, SERIAL_8N1);
}

void loop() {
  slave0.update();

  if (Serial.available() > 0) {
    incomingByte = Serial.read();
    slave0.uartWrite(COM1, incomingByte);

    if ((read_ch = (char)slave0.uartRead(COM1)) > 0) {
      Serial.print("COM1 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.



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



It will print the value to the serial monitor if QEC-R11CFFG's COM port receives data.

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:

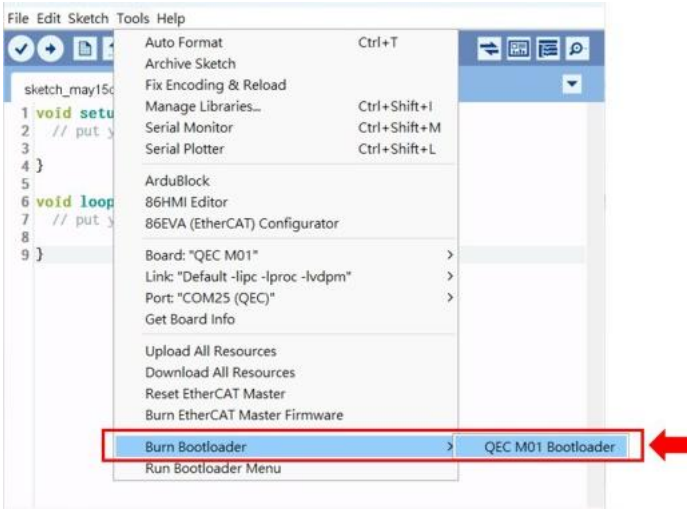
Step 1: Setting up QEC-M

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.

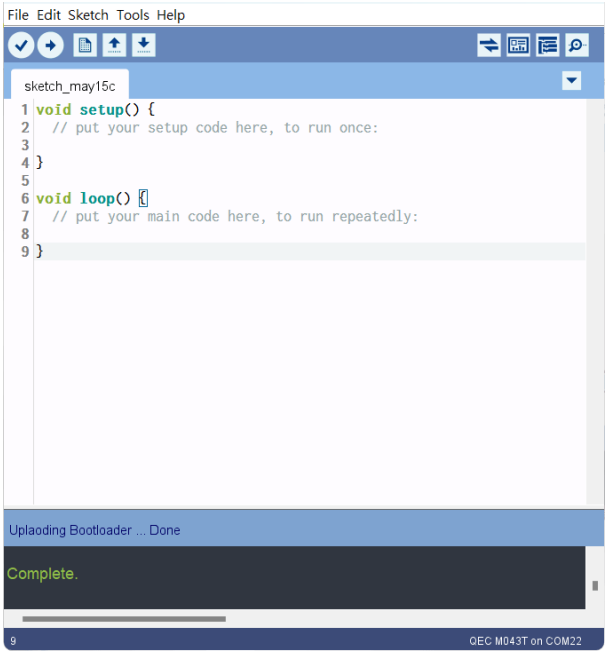
Step 2: Click “Burn Bootloader” button

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.