Start Guide

M56S EtherCAT Servo Motor
CiA402 PP Mode (1-axis) with 86EVA

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

Date	Version	Description
2025/06/04	Version1.0	New Release.

Preface

In this guide, we will show you how to use the EtherCAT MDevice QEC-M-01 and the Moon's M56S EtherCAT Servo Motor.

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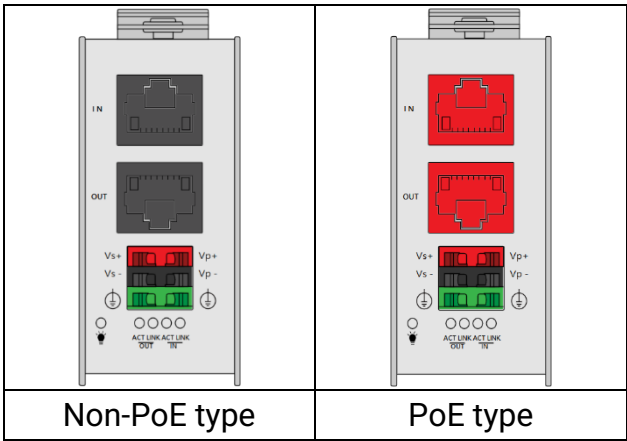
PoE (Power over Ethernet) is a function that delivers power over the network. QEC can be equipped with an optional PoE function to reduce cabling. In practice, PoE is selected based on system equipment, so please pay attention to the following points while evaluating and testing:

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-

T568B Color

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.   M56S (EtherCAT Servo Motor)
3.   24V power supply & EU-type terminal cable & LAN cable
4.   220V/400V AC input
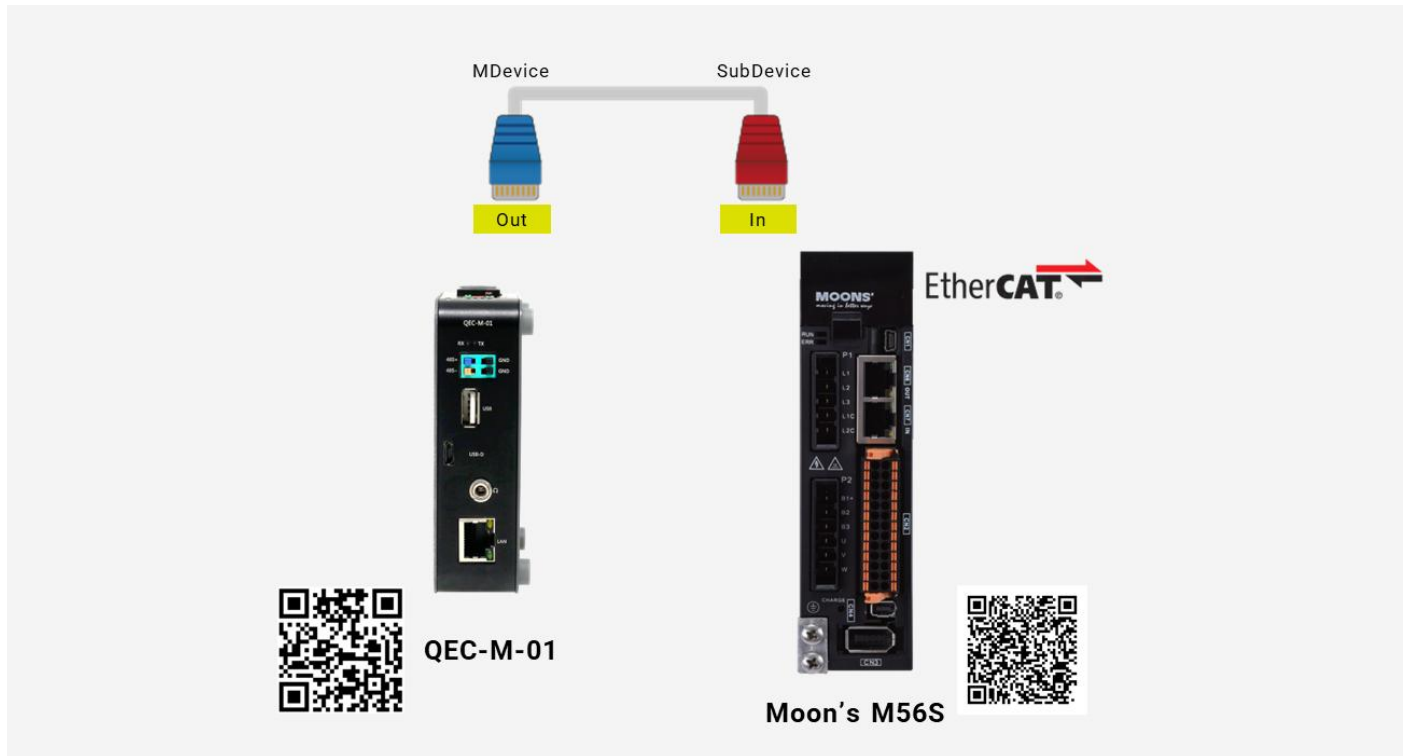
1.1 QEC-M-01

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.
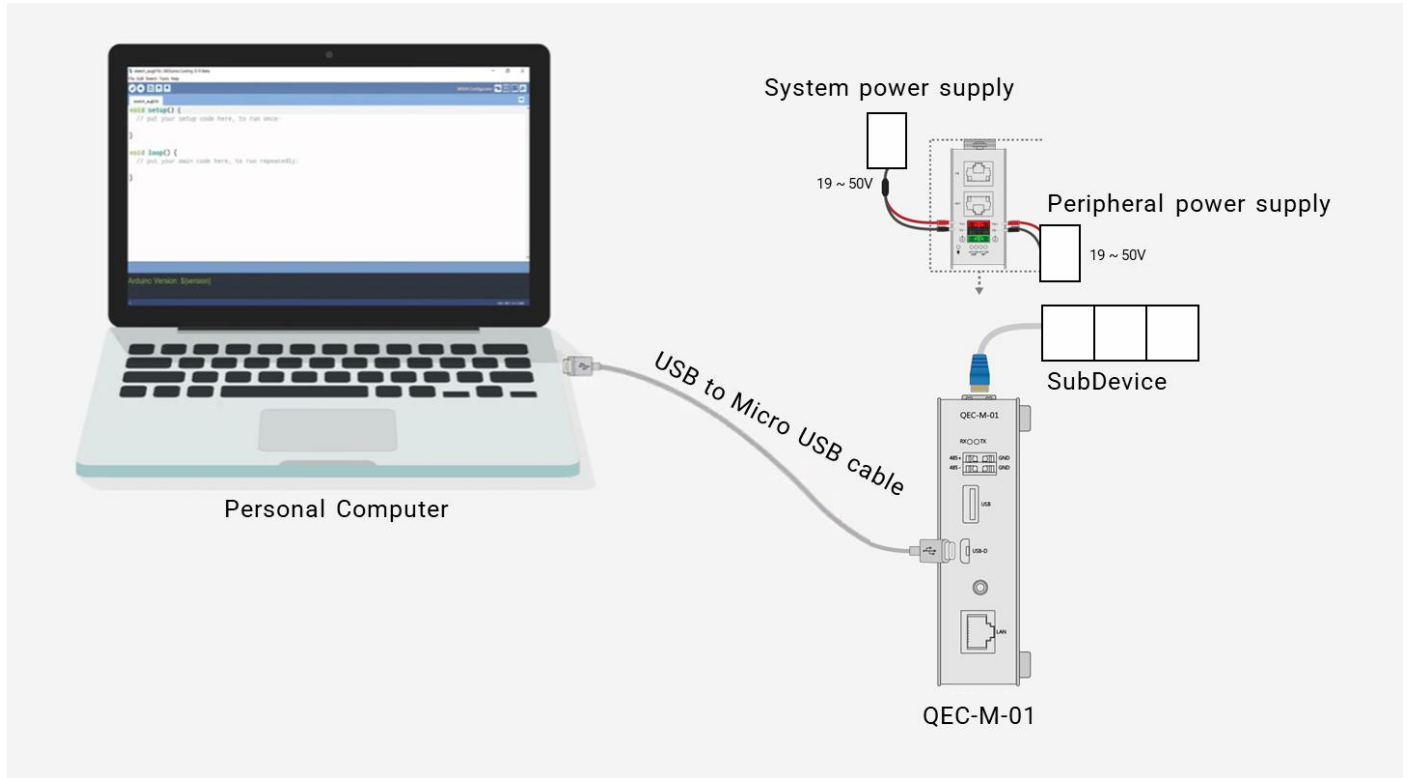
1.2 M56S

M56S series is a new generation of high-end 220V/400V AC input servo system, the power range is from 50W to 7.5KW. The drive adopts advanced motion control algorithms, coupled with a 26-bit high precision encoder motor, the velocity loop bandwidth is up to 3.5KHz. Efficient automatically adjusts the loop gain, superior vibration suppression and external disturbance compensation performance. The drive supports full closed-loop control, built-in regenerative energy absorption resistor, STO safety function and dynamic braking module (for certain models). It supports the control mode of position, velocity, torque and Q programming.
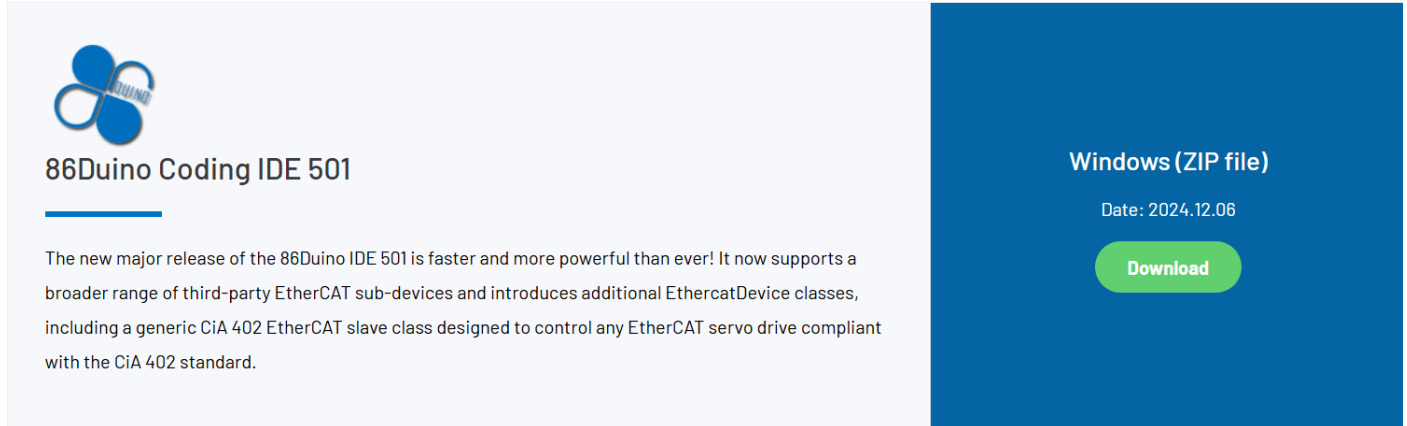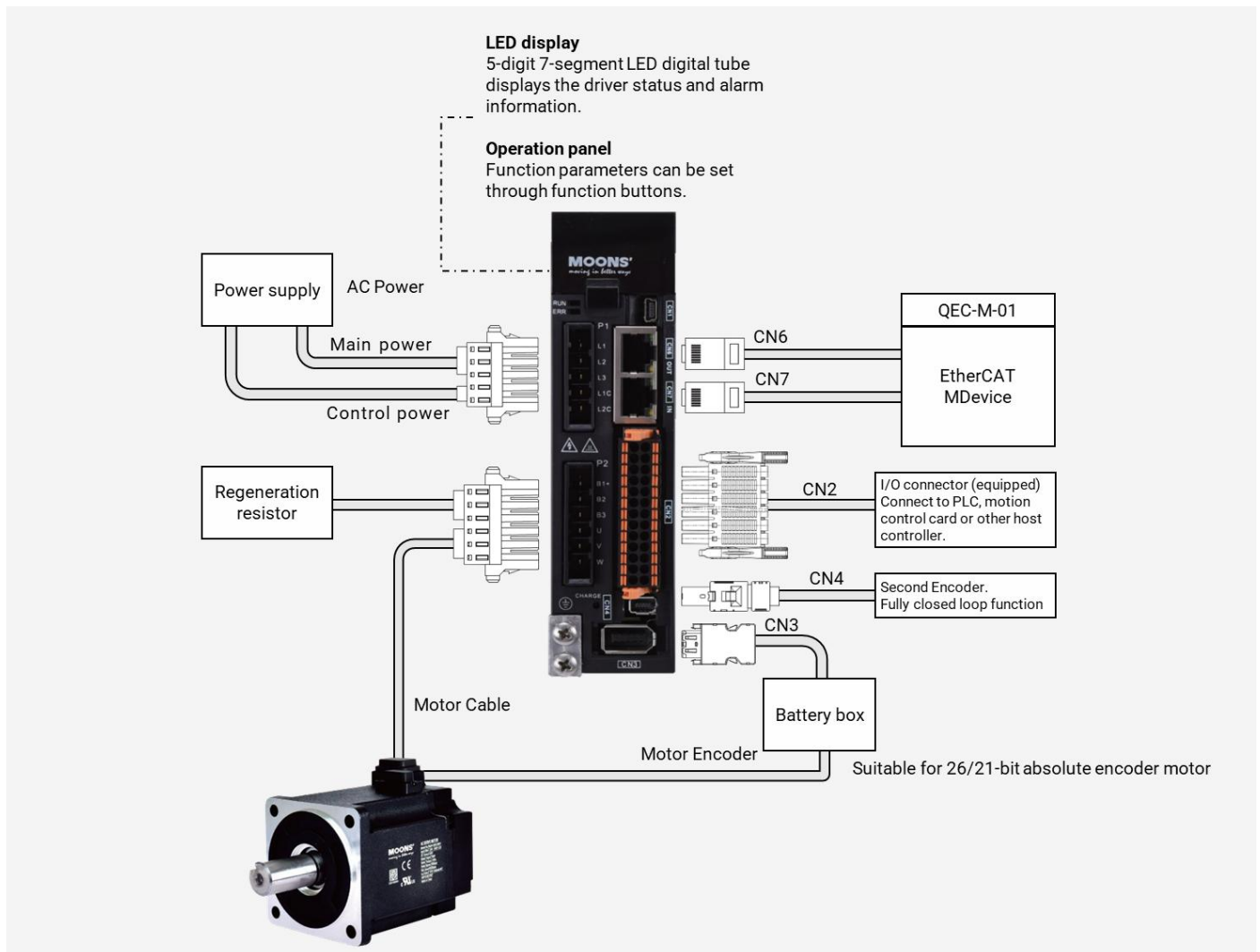

M56S Driver Wiring Overview
- **Power Supply**
  - AC input split into:
- **Communication**
  - CN6 / CN7: EtherCAT connection to QEC-M-01.
- **I/O Control**
  - CN2: Built-in I/O port.

- ▪ Connects to PLC or motion controller.
        - ▪ For start, stop, alarm reset, etc.
- **Encoder Feedback**
    - o CN3: Main encoder (supports 21/26-bit absolute encoders).
    - o CN4: Second encoder input for full-closed loop.
- **Front Panel**
    - o LED Display: Shows status and alarms.
    - o Function Keys: For parameter setting.

2. Software/Development Environment

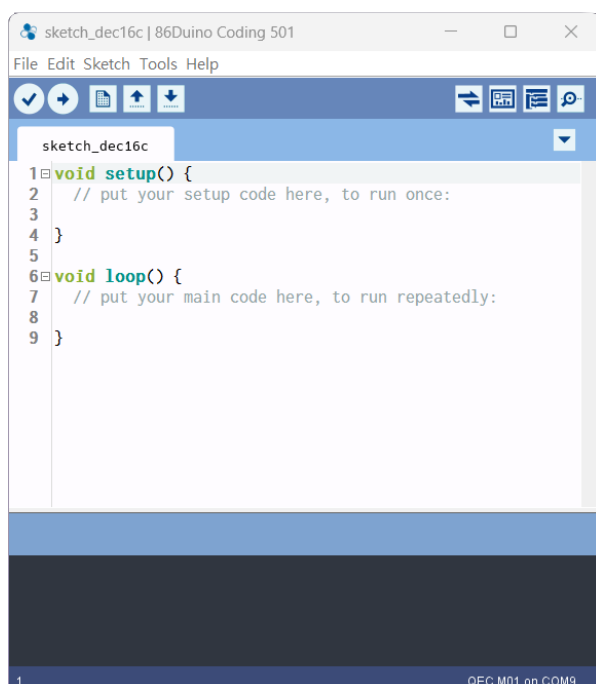Download 86duino IDE from https://www.qec.tw/software/.

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-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)
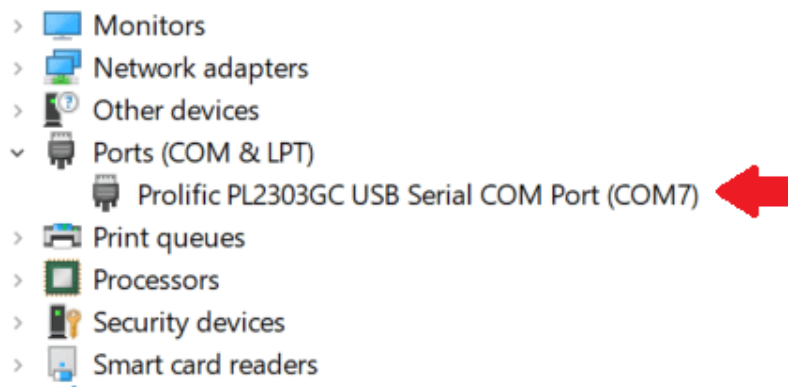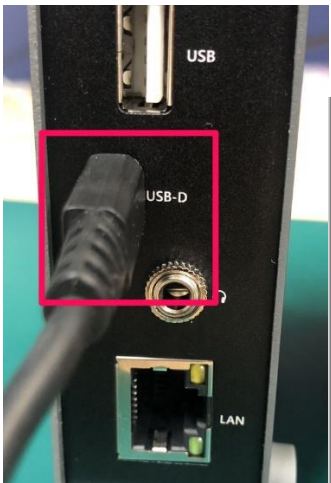
4. Open the 86Duino IDE.
5. Select the correct board: In the IDE's menu, select Tools> Board > QEC-M-01 (or the QEC MDevice model you use).
6. Select Port: In the IDE's menu, select Tools > Port and select the USB port to connect to the QEC MDevice (in this case, COM9 (QEC)).
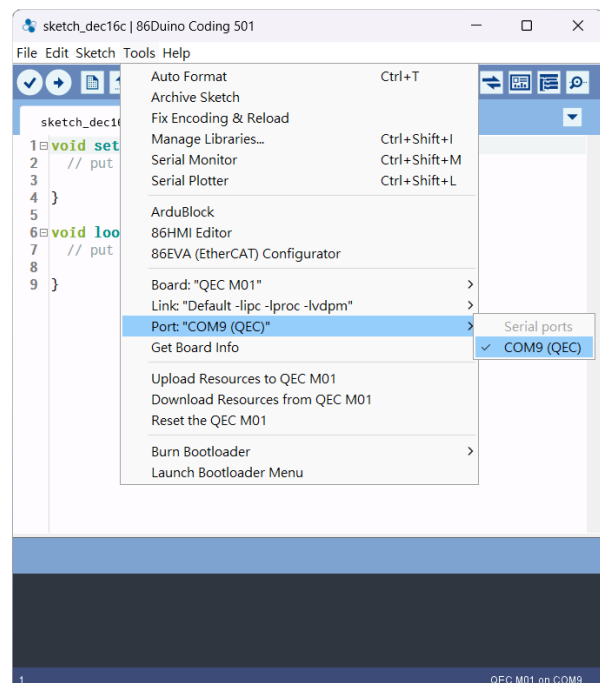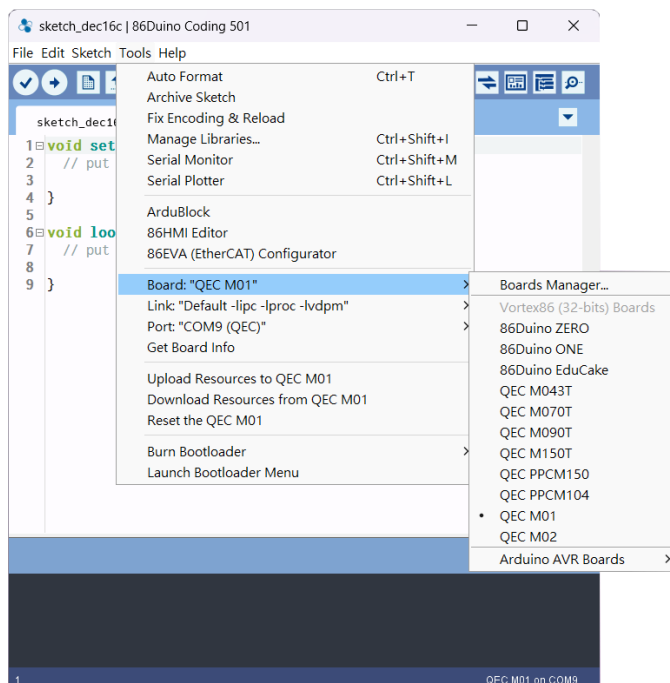
4.   Use 86EVA with code

This example shows how to operate the EtherCAT MDevice (QEC-M-01) and the Moon's M56S EtherCAT Servo Motor 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.

This code establishes EtherCAT communication and controls the M56S servo motor in Profile Position (PP) mode. The motor's position is updated cyclically, and the target position alternates between 100,000 and -100,000 units, simulating continuous forward and reverse movements.

Step 1: Turn on 86EVA and scan

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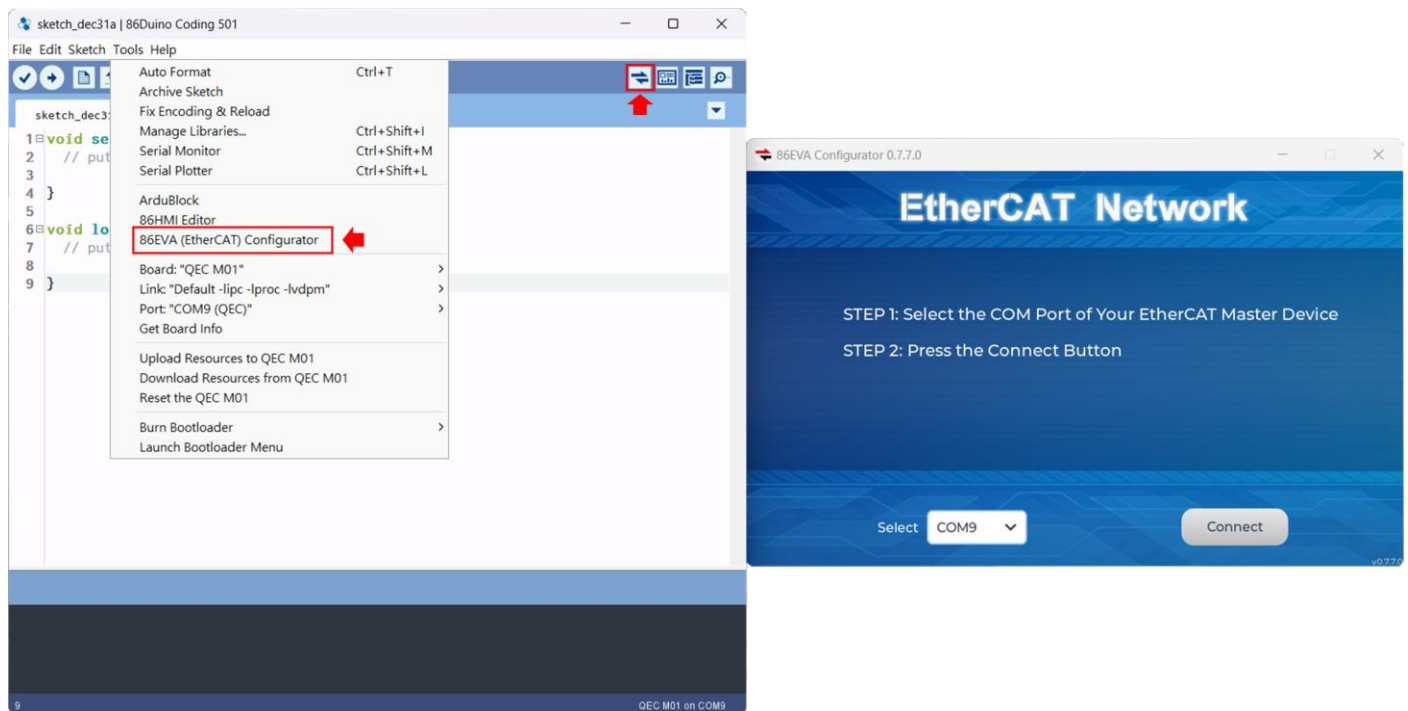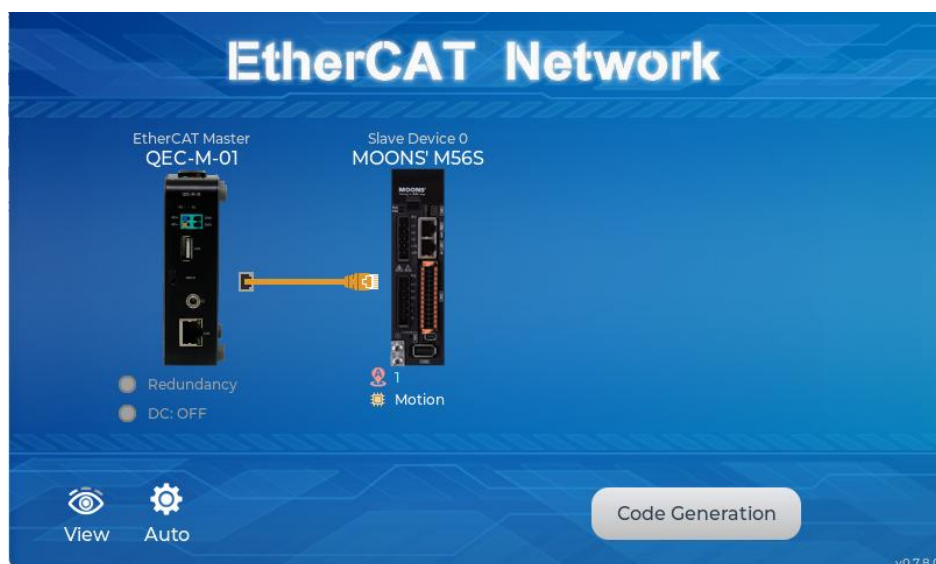

Please select the correct COM port and then click the "Connect" button.

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.
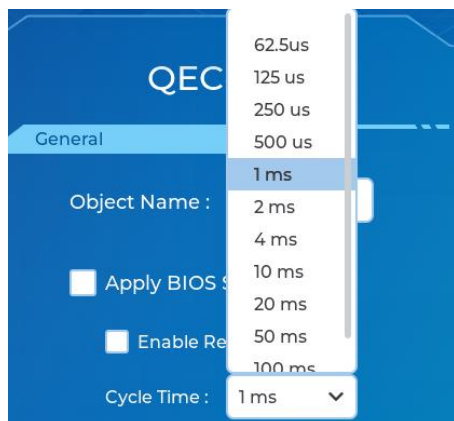
QEC-M-01

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.
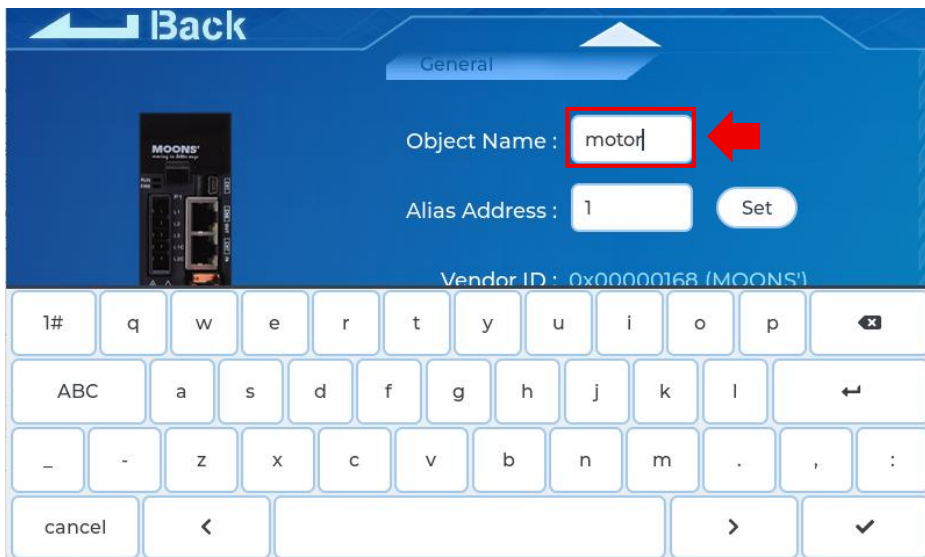
Moon's M56S

Press twice on the image of the M56S servo motor 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 "motor".
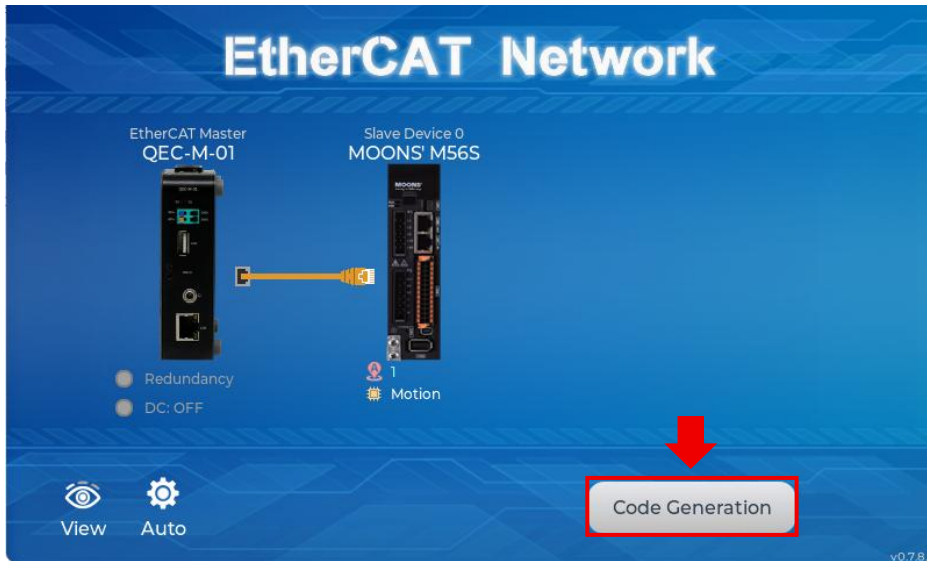It'll appear a keyboard after you click the Object Name.

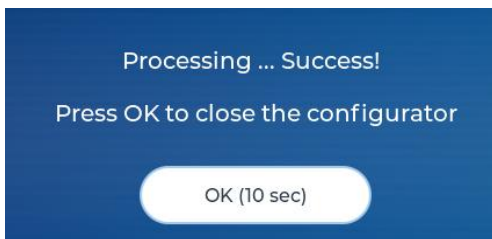Click "Back" in the upper left corner to return.

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

The programming code from 86EVA are set as the following by default:
- Moon's M56S EtherCAT Servo Motor: EthercatDevice_CiA402 object.
- CiA402 mode: Profile Position (PP) mode.
- EtherCAT mode: ECAT_SYNC.

And here is the setting by users:
- EtherCAT Cycle time: 1 millisecond.
- Device Object Name: QEC-M-01 is "EcatMaster", and M56S servo motor is "motor".

This code establishes EtherCAT communication and controls the M56S servo motor in Profile Position (PP) mode. The motor's position is updated cyclically, and the target position alternates between 100,000 and 100,000 units, simulating continuous forward and reverse movements.

A. In Setup Function:

In the setup() function initializes communication and configures the motor for CiA402 Profile Position (PP) 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.
3. Set Profile Position (PP) Mode
    - Configure the motor to PP mode using setCiA402Mode(CIA402_PP_MODE).
4. Enable the Motor
    - Use the enable() function to enable the motor and transition it to CIA402_OPERATION_ENABLED.
5. Configure Profile Parameters
    - Motion Profile Type: Linear Ramp, Profile Velocity: 100,000, Acceleration: 5,000, Deceleration: 5,000.

B. In Loop Function:

In the loop() function, the current position of the motor is displayed on the Serial Monitor, and the motor alternates its movement back and forth in a repeating cycle:
1. State Machine Logic
    - case 0:
      Start the motor and move to the target position (100,000 units). Once the command is successfully executed, transition to the next state.
    - case 1:
      Wait for the motor to reach the target position. Once the target is reached, proceed to the next state.

- case 2:

  Start the motor and move back to the original position (-100,000 units). Once the command is successfully executed, transition to the next state.

- case 3:

  Wait for the motor to return to the original position. Once the target is reached, reset the state machine back to case 0 to repeat the movement cycle.

2. Code Logic Summary

- Use the pp_Run() function to initiate position movement.
- Use the pp_IsTargetReached() function to confirm whether the target position has been reached.
- The state machine starts at case 0 and resets after completing case 3.

The example code is as follows:

```
#include "myeva.h"

int pp_state = 0;

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

  EVA.begin();
  motor.setCiA402Mode(CIA402_PP_MODE);
  Serial.print("Enable: "); Serial.println(motor.enable());
  motor.pp_SetMotionProfileType(0); // Linear ramp (trapezoidal profile)
  motor.pp_SetVelocity(100000);
  motor.pp_SetAcceleration(5000);
  motor.pp_SetDeceleration(5000);
}

void loop() {
  Serial.print("Pos: "); Serial.println(motor.getPositionActualValue());
  switch (pp_state)
  {
    case 0:
      if (motor.pp_Run(100000) == 0)
        pp_state++;
      break;
    case 1:
      if (motor.pp_IsTargetReached())
        pp_state++;
      break;
```
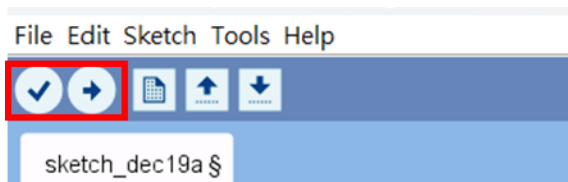
```
    case 2:
      if (motor.pp_Run(-100000) == 0)
        pp_state++;
      break;
    case 3:
      if (motor.pp_IsTargetReached())
        pp_state = 0;
      break;
  }
}
```
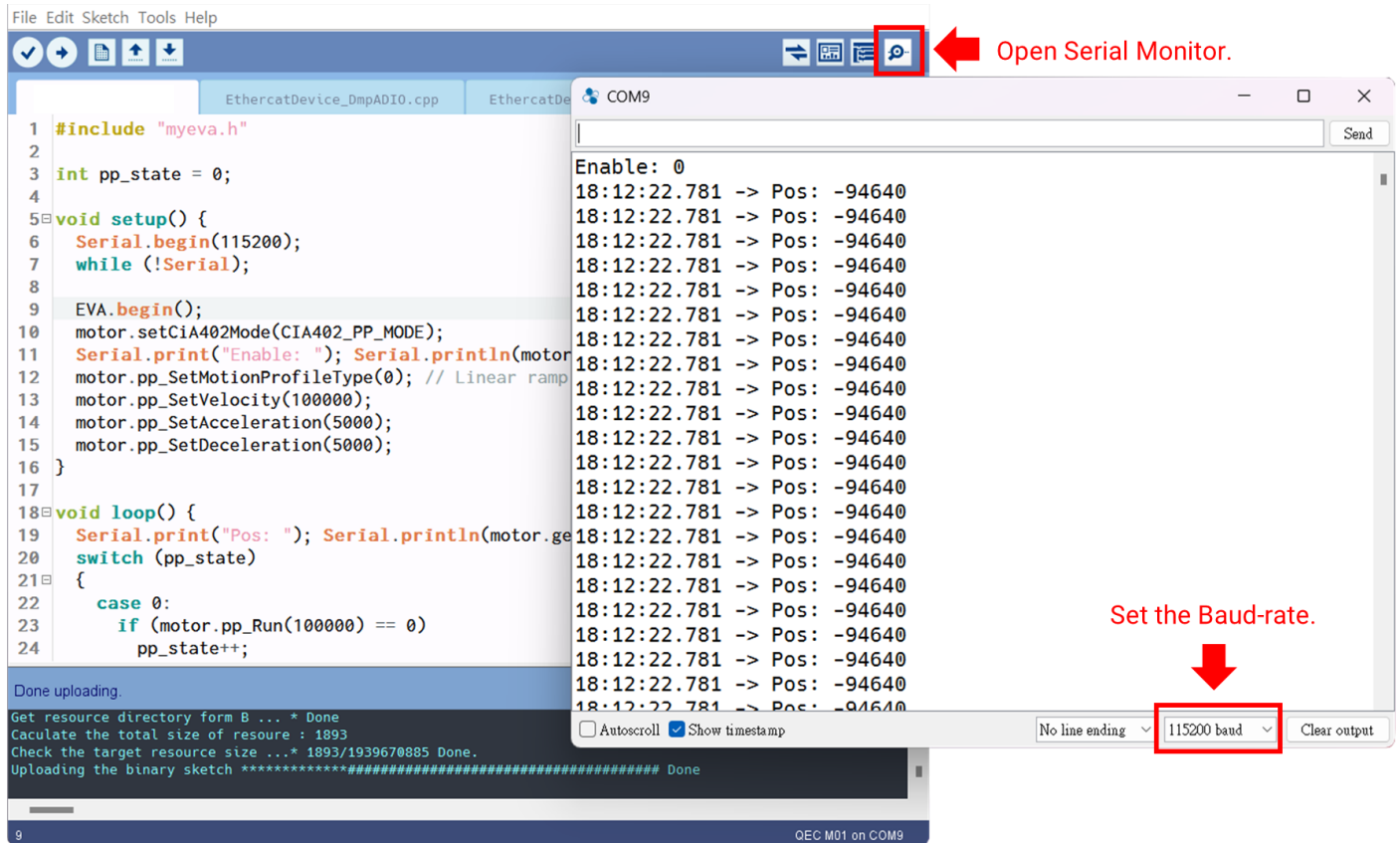
Note:

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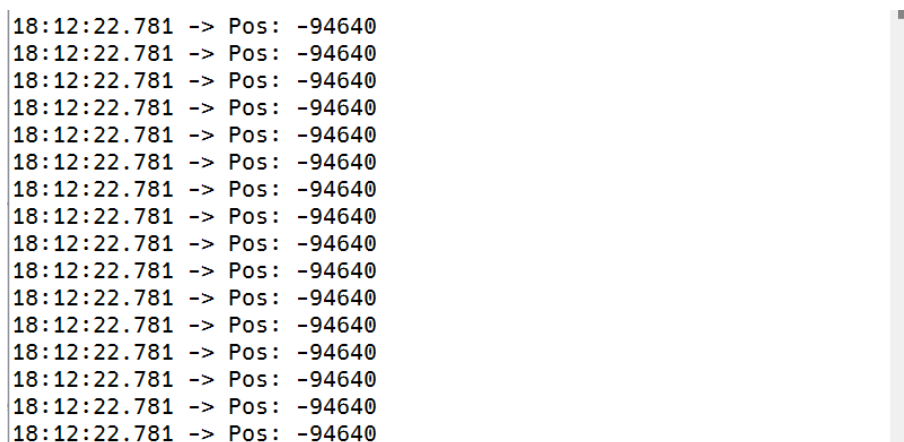
After you successfully upload the program to the QEC-M-01, you can open the Serial Monitor on 86Duino IDE. Please check the Serial baud rate is same as your setting.

If the EtherCAT communication config successful, Serial Monitor will print "Enable: 0", which means the CiA402 switch to Enable is successful.

It will print the motor's current position to the serial monitor.

```
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
18:12:22.781 -> Pos: -94640
```

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.
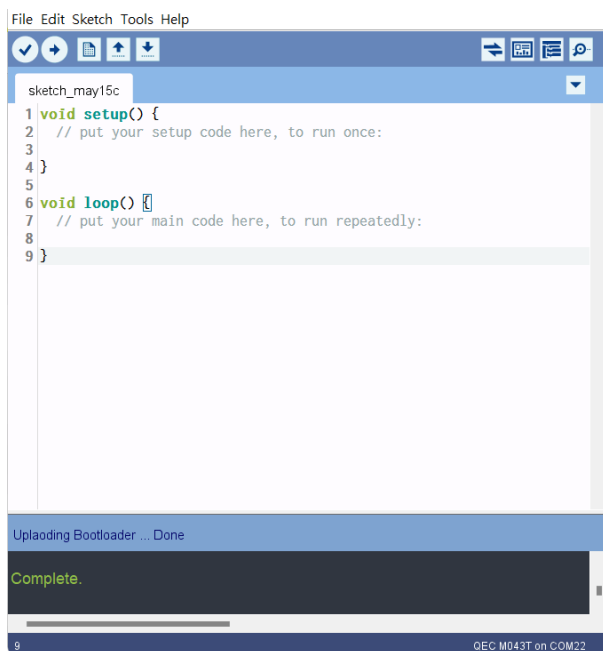
Step 2: Click "Burn Bootloader" button

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:
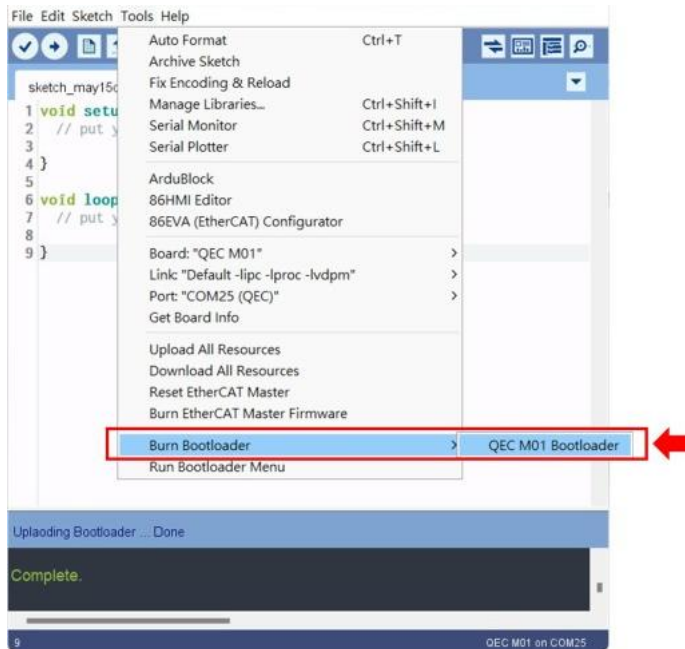
Step 3: Complete the Update

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.