Start Guide

OrientalMotor AZ-mini EtherCAT Slave (PP mode)

86Duino Coding IDE 500

EtherCAT Library

(Version 1.0)

Revision

Date	Version	Description
2024/9/11	VERSION1.0	NEW RELEASE.

Preface

In this guide, we will show you how to use the EtherCAT Master QEC-M-01 and OrientalMotor AZ-mini EtherCAT Slave.

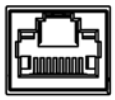
Notes QEC's PoE (Power over Ethernet)

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:

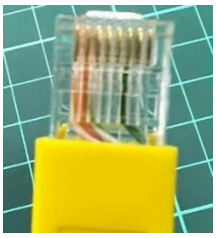


Pin #	Signal Name	Pin #	Signal Name
1	LAN1_TX+	2	LAN1_TX-
3	LAN1_RX+	4	VS+
5	VP+	6	LAN1_RX-
7	VS-(GND)	8	VP-(GND)

* 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

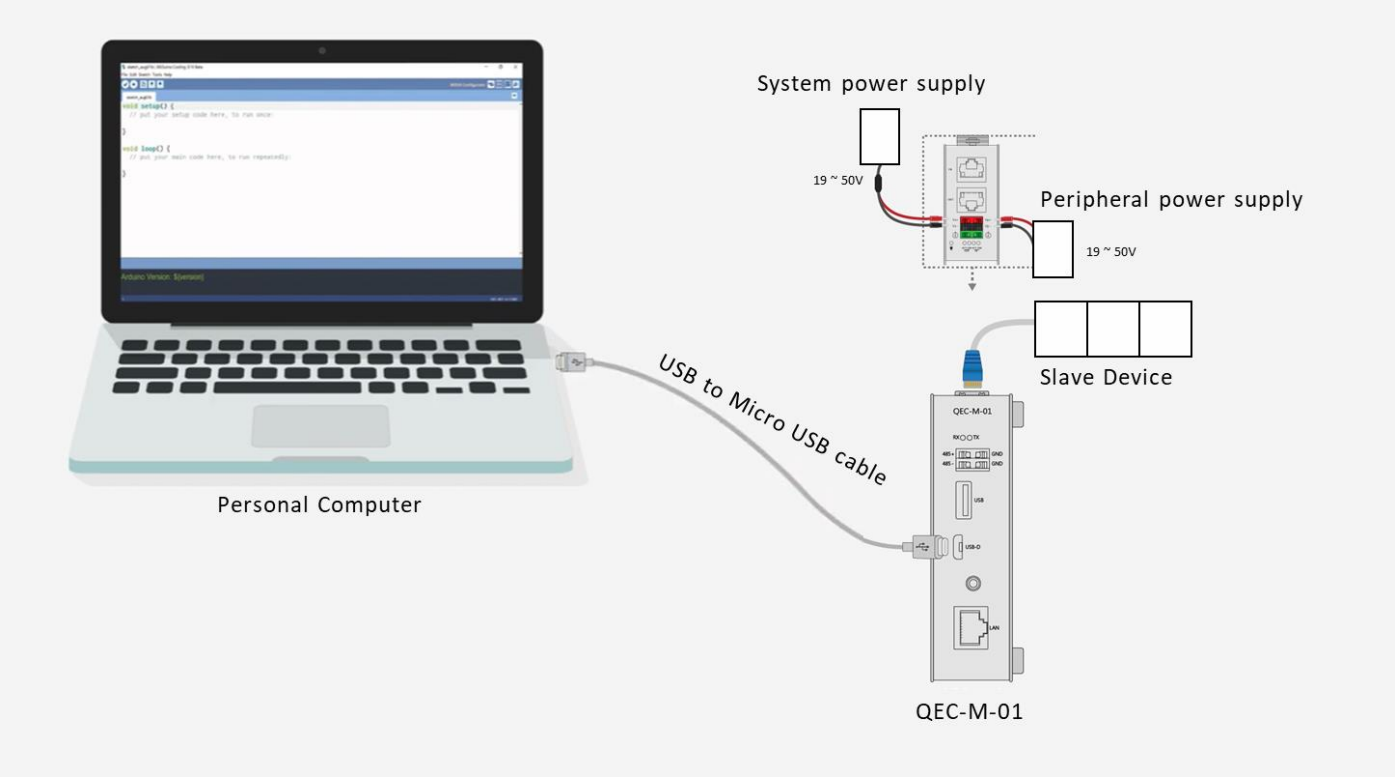
The following devices are used here:

1. QEC-M-01 (EtherCAT Master)
2. AZD-KRED, AZ Series mini EtherCAT Driver (OrientalMotor Step-Servo Driver)
3. AZM48AK, 1.65 in. (42 mm) AZ Series Stepper Motor with Absolute Mechanical Encoder (DC Input)
4. 24V power supply

1.1 QEC-M-01

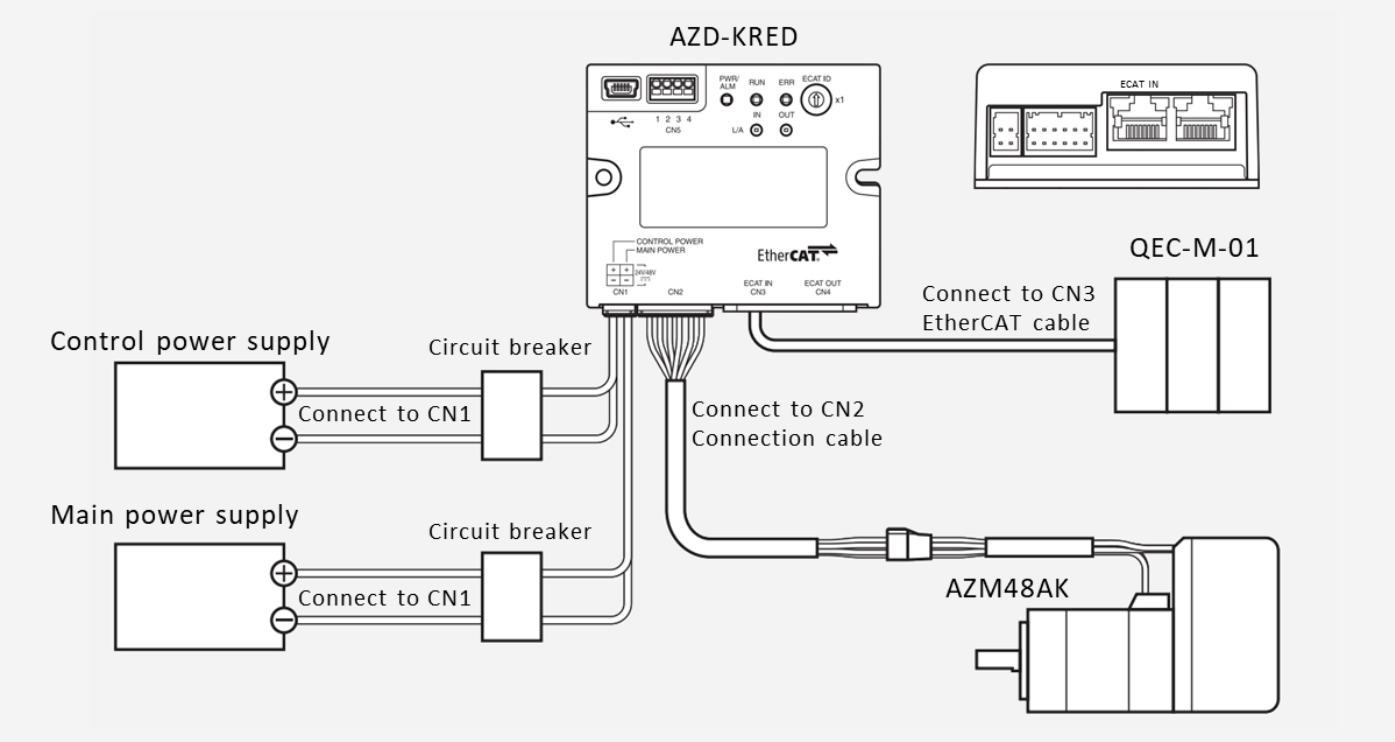
QEC EtherCAT Master.

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 AZD-KRED via RJ45 cable.



1.2 AZD-KRED

AZD-KRED, an AZ Series mini EtherCAT Driver (OrientalMotor Step-Servo Driver).
This figure shows an example when the **AZM48AK** motor is connected.



1. It is an OrientalMotor cable. Purchase is required separately.
2. Connecting the control power supply allows you to continue monitoring even if the main power supply is shut off. Connect it as necessary.
3. It is recommended that a circuit breaker or a circuit protector is connected because incorrect wiring may cause the internal input circuit to short-circuit.

Note

- Connect the connectors securely. Insecure connections may cause malfunction or damage to the motor or the driver.
- When connecting the cables, secure them so that no load is applied to the connectors. Applying a load to the connector may result in a connection failure, causing the driver to malfunction.
- Keep **10 m (32.8 ft.)** or less for the wiring distance between a motor and a driver. Exceeding **10 m (32.8 ft.)** in the wiring distance may result in an increase of the electrical noise emitted from the driver.
- Keep **2 m (6.6 ft.)** or less for the cable length of the main power supply and control power supply.

Memo

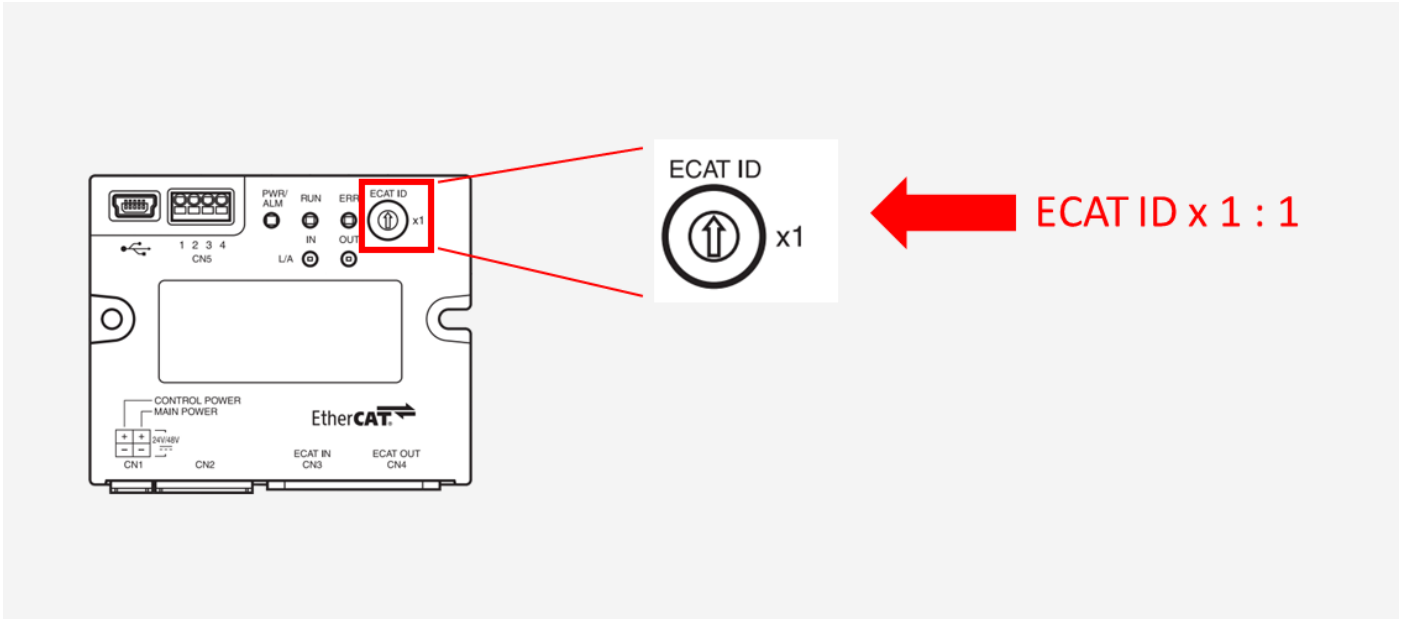
- Before connecting or disconnecting a connector, turn off the main power supply and the control power supply, and check the **PWR/ALM LED** has been turned off.
- When disconnecting the connector, pull out while pressing the latches on the connector with fingers.

Node Address Configuration for AZ-mini Driver:

The node address can be set using the **ECAT ID ×1** switch. The switch allows configuration of node addresses ranging from 0 to 15 in hexadecimal.

- The **node address switch** can be set to 16 steps, where addresses 0 to 15 correspond to hexadecimal values 0x0 to 0xF.
- When connecting multiple devices in the EtherCAT network, ensure that no two devices have the same node address to avoid conflicts.

Initial Shipment Setting: 0 (×1:0)

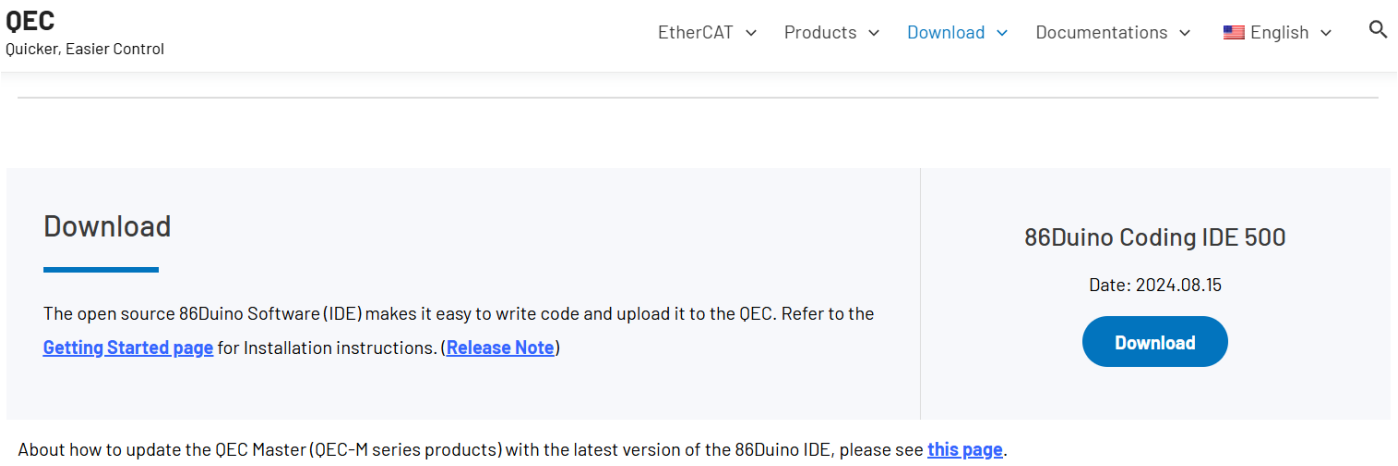


Setting Range	Description
0 (00h)	The master’s settings will be applied.
1-15 (01h-0Fh)	The driver's settings will be applied.

* Note: When adjusting the switch settings, ensure that both the main power and control power are turned off. The settings will not take effect if the switch is adjusted while the power is still on.

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.

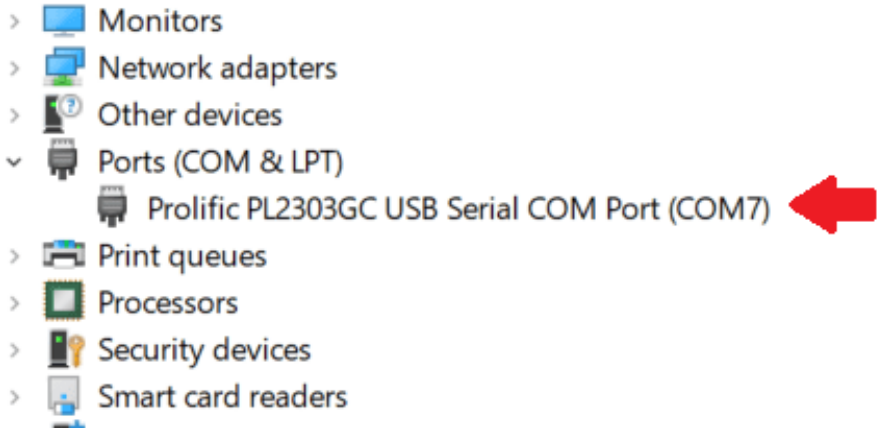
Note: If Windows displays a warning, click Details once and then click the Continue Run button once.

86Duino Coding IDE 500+ looks like below.

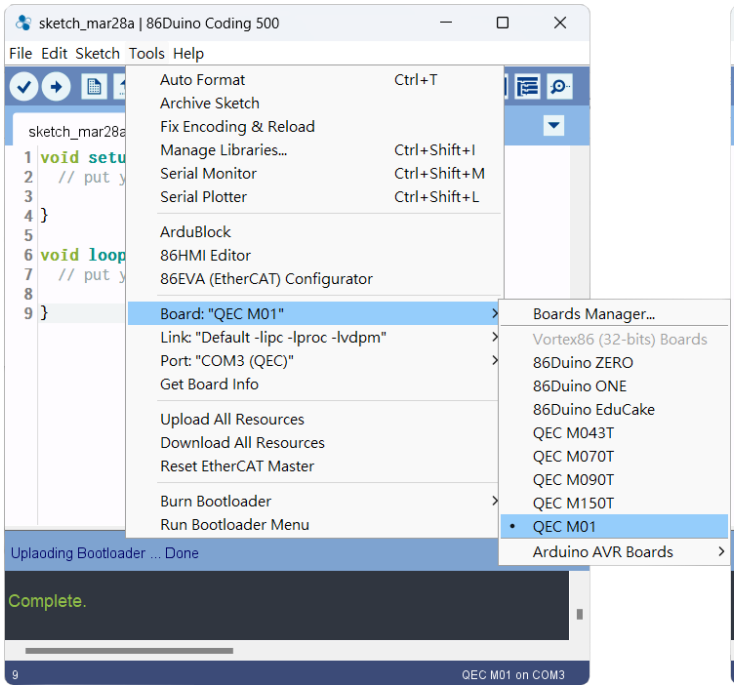
3. Connect to your PC and set up the environment

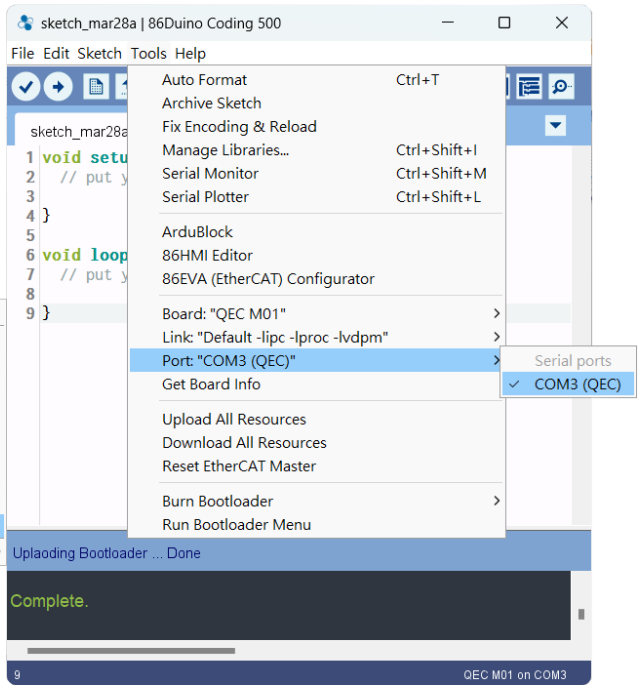
Follow the steps below to set up the environment:

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-01) and the OrientalMotor AZ-mini EtherCAT Driver (AZD-KRED) 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 AZ-mini driver into CiA402 Profile Position (PP) mode:

- EtherCAT Cycle Time: 1 millisecond.
- EtherCAT Mode: ECAT_SYNC.
- Distributed Clock: Enabled, following the cycle time.

The `EthercatMaster` object (master) represents the QEC-M-01, while the `EthercatDevice_CiA402` object (az_mini) represents the AZ-mini driver.

1. **In Setup Function:**
   - Initialize serial communication at a baud rate of 115200.
   - Start the EtherCAT master and switch the EtherCAT state machine to the PRE-OPERATIONAL state.
   - Configure the AZ-mini in Profile Position (PP) mode.
   - Set the AZ-mini to Distributed Clock (DC) mode and synchronize the cycle time with EtherCAT communication.
   - Set the maximum motor speed to 10,000 (units depending on system configuration).
   - Register the cyclic callback function myCallback for periodic position updates.
   - Start the EtherCAT master, setting the cycle time and mode in the `start()` function. Switch the EtherCAT state machine to the OPERATIONAL state.
   - Enable the AZ-mini and transition its CiA402 state to CIA402_OPERATION_ENABLED. Use `delay(1000)` to ensure the state change completes successfully.
2. **In Callback Function:**
   The motor's actual position is read cyclically, allowing real-time updates of the motor's current position.
3. **In Loop Function:**
   The main loop prints the motor's current position to the serial monitor.
   And it manages the Profile Position (PP) mode state machine. The motor moves to a specific position, waits for the target to be reached, and then reverses back to the starting position:
   - State 0: Starts the profile position move to a target position (10,000 units). Once the move begins, transition to the next state.
   - State 1: Ends the current profile position movement.
   - State 2: Waits for the motor to reach the target position. Once the target is reached, proceed to the next state.
   - State 3: Starts a profile position move back to the original position (0 units). When the move starts, transition to the next state.
   - State 4: Ends the current profile position movement.

- State 5: Waits for the motor to reach the target position. If the target is reached, reset the state machine back to State 0 to repeat the cycle.

Each state introduces a delay of 100 milliseconds to ensure smooth transitions.

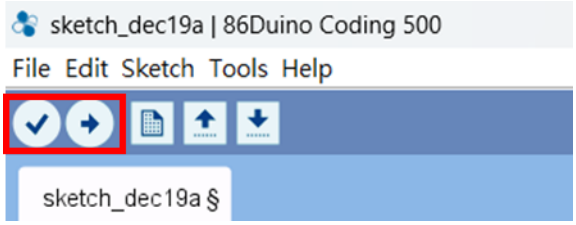
This code establishes EtherCAT communication and controls the AZ-mini driver in Profile Position (PP) mode. The motor's position is updated cyclically, and the target position alternates between 0 and 10,000 units, simulating continuous forward and reverse movements.

Here is the code:

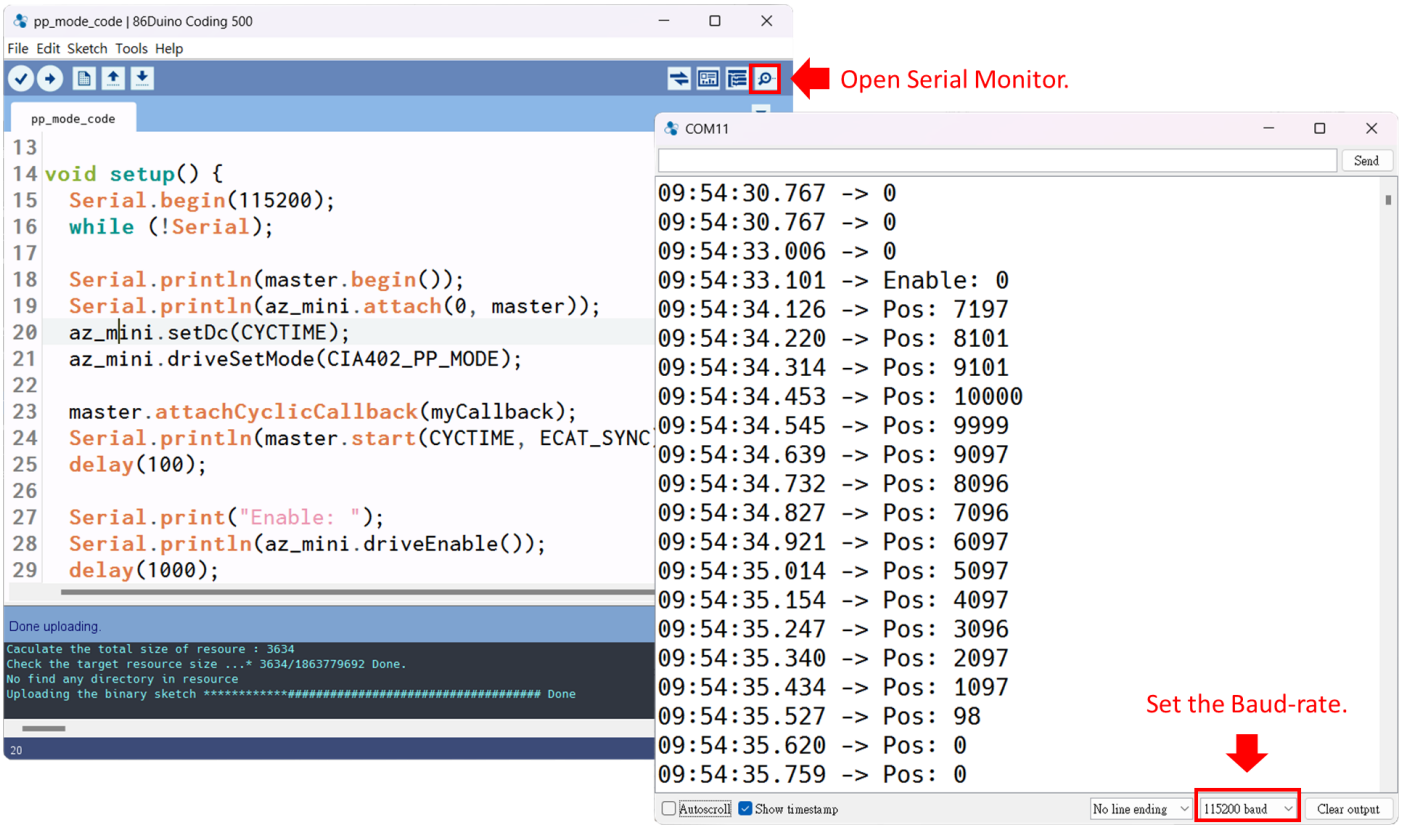
```
#include "Ethercat.h"

#define CYCTIME 1000000

EthercatMaster master;
EthercatDevice_CiA402 az_mini;

int pp_state = 0, pos = 0;

void myCallback() {
  pos = az_mini.driveGetPositionActualValue();
}

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

  Serial.println(master.begin());
  Serial.println(az_mini.attach(0, master));
  az_mini.setDc(CYCTIME);
  az_mini.driveSetMode(CIA402_PP_MODE);
  az_mini.driveSetMaxMotorSpeed(10000);

  master.attachCyclicCallback(myCallback);
  Serial.println(master.start(CYCTIME, ECAT_SYNC));
  delay(100);

  Serial.print("Enable: ");
  Serial.println(az_mini.driveEnable());
  delay(1000);
}
```

```
void loop() {
  Serial.print("Pos: ");
  Serial.println(pos);

  switch (pp_state) {
    case 0:
      if (az_mini.profilePositionBegin(10000, 10000, 10000) == 0)
        pp_state++;
      break;
    case 1:
      az_mini.profilePositionEnd();
      pp_state++;
      break;
    case 2:
      if (az_mini.driveIsTargetReached())
        pp_state++;
      break;
    case 3:
      if (az_mini.profilePositionBegin(0, 10000, 10000) == 0)
        pp_state++;
      break;
    case 4:
      az_mini.profilePositionEnd();
      pp_state++;
      break;
    case 5:
      if (az_mini.driveIsTargetReached()) {
        pp_state = 0;
      }
      break;
  }

  delay(100);
}
```

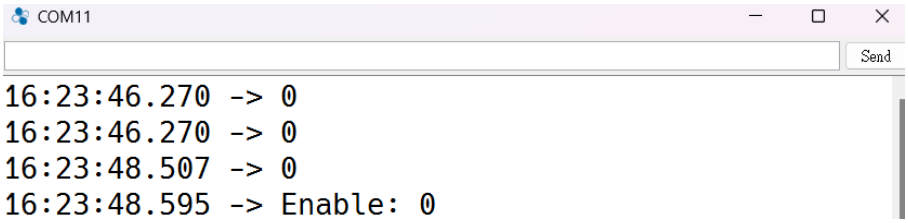
Note: Once the code is written, click on the toolbar to compile, and to confirm that the compilation is complete and error-free, you can click to upload.



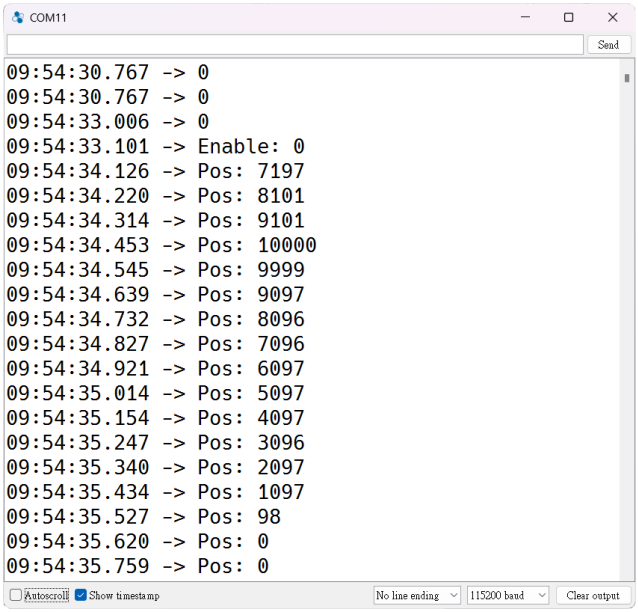
After you successfully upload the program to the QEC-M-01, you can open the Serial Monitor on 86Duino IDE. Please check the Serial baud rate is same as your setting.



If the EtherCAT communication config successful, Serial Monitor will print “0” and “Enable: 0”.



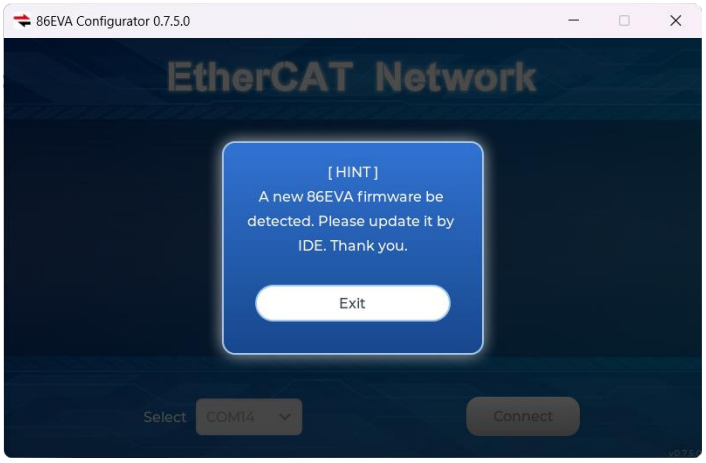
It will print the motor’s current position to the serial monitor.



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 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.



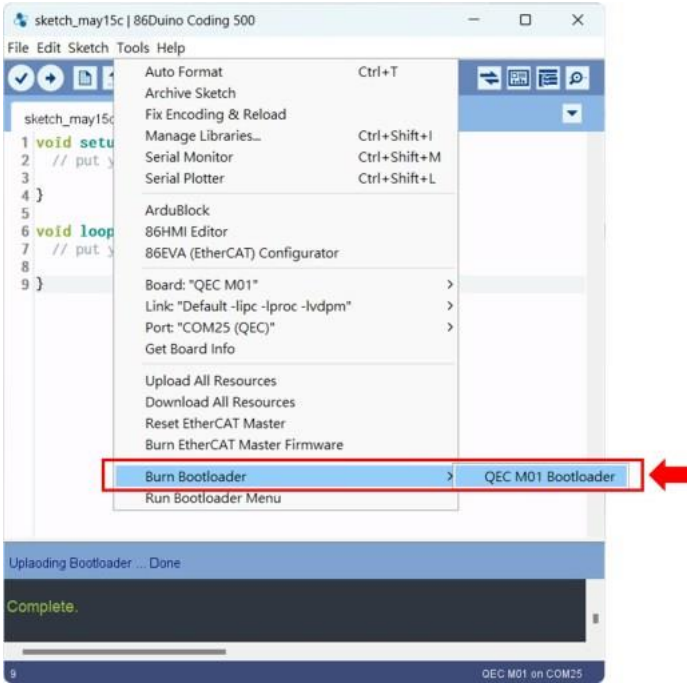
Now, we will further explain how to proceed with the update:

Step 1: Setting up QEC-M

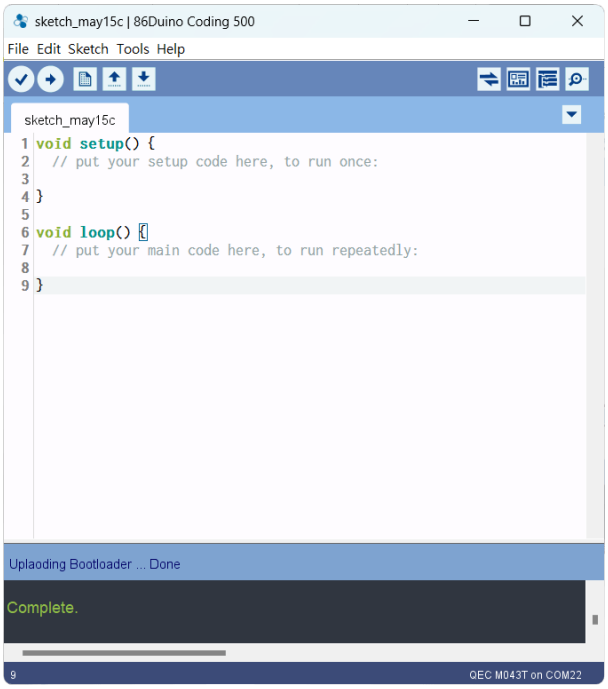
1. Download and install 86Duino IDE 500 (or a newer version): You can download it from Software.
2. Connect the QEC-M: Use a USB cable to connect the QEC-M to your computer.
3. Open 86Duino IDE: After the installation is complete, open the 86Duino IDE software.
4. Select Board: From the IDE menu, choose “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.