Start Guide

Delta AC Servo Motors and Drives
ASDA-B3/ASDA-A2 Series (PP mode)

86Duino Coding IDE 501

EtherCAT Library

(Version 2.0)

Revision

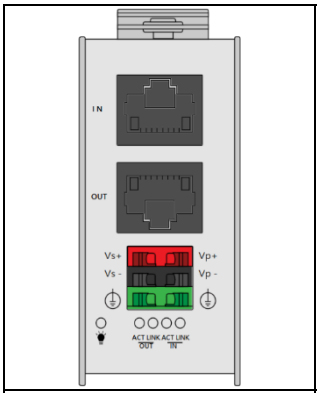
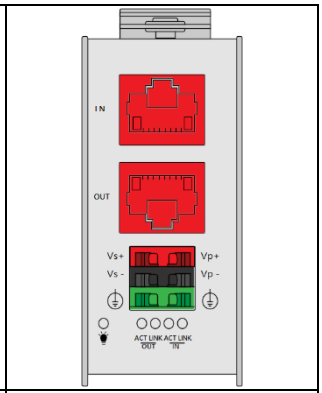
Date	Version	Description
2024/11/04	Version1.0	New release.
2026/4/9	Version2.0	• Use 86duino IDE 501 control. • Change Main-Device to MDevice, and Sub-Device or slave to SubDevice

Preface

In this guide, we will show you how to use the EtherCAT MDevice QEC-M-01 and Delta's ASDA-B3 and ASDA-A2 AC Servo Motors and Drives, taking Profile Position (PP) mode as an example.

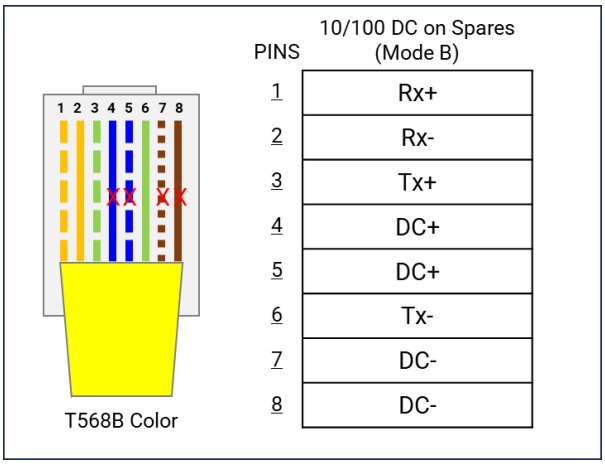
Notes QEC's PoE (Power over Ethernet)

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

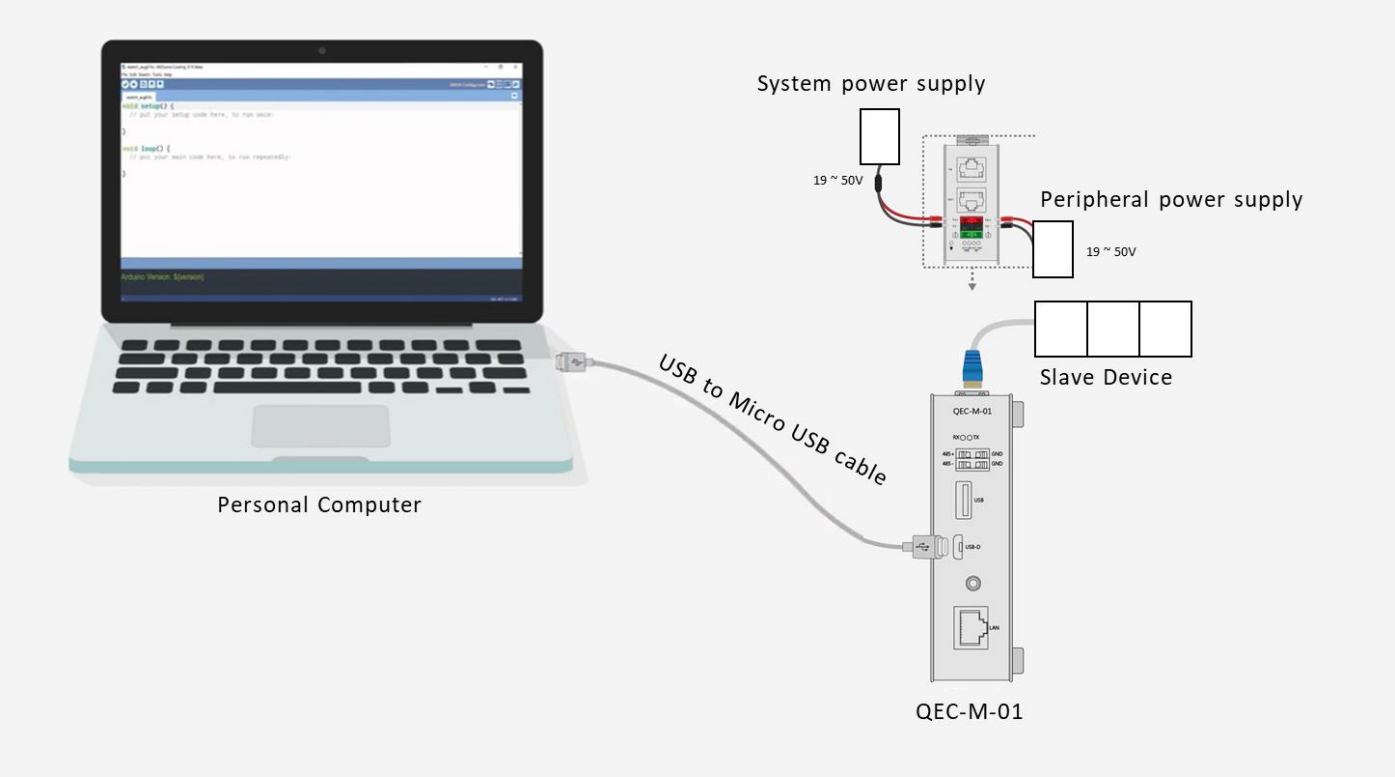
The following devices are used here:

1. QEC-M-01 (EtherCAT MDevice)
2. Delta ASDA-B3 (EtherCAT AC Servo Motor and Drives)
3. Delta ASDA-A2 (EtherCAT AC Servo Motor and Drives)
4. 24V power supply & EU-type terminal cable
5. 220V power supply for Delta AC Servo.
6. RJ45 Cables

1.1 QEC-M-01

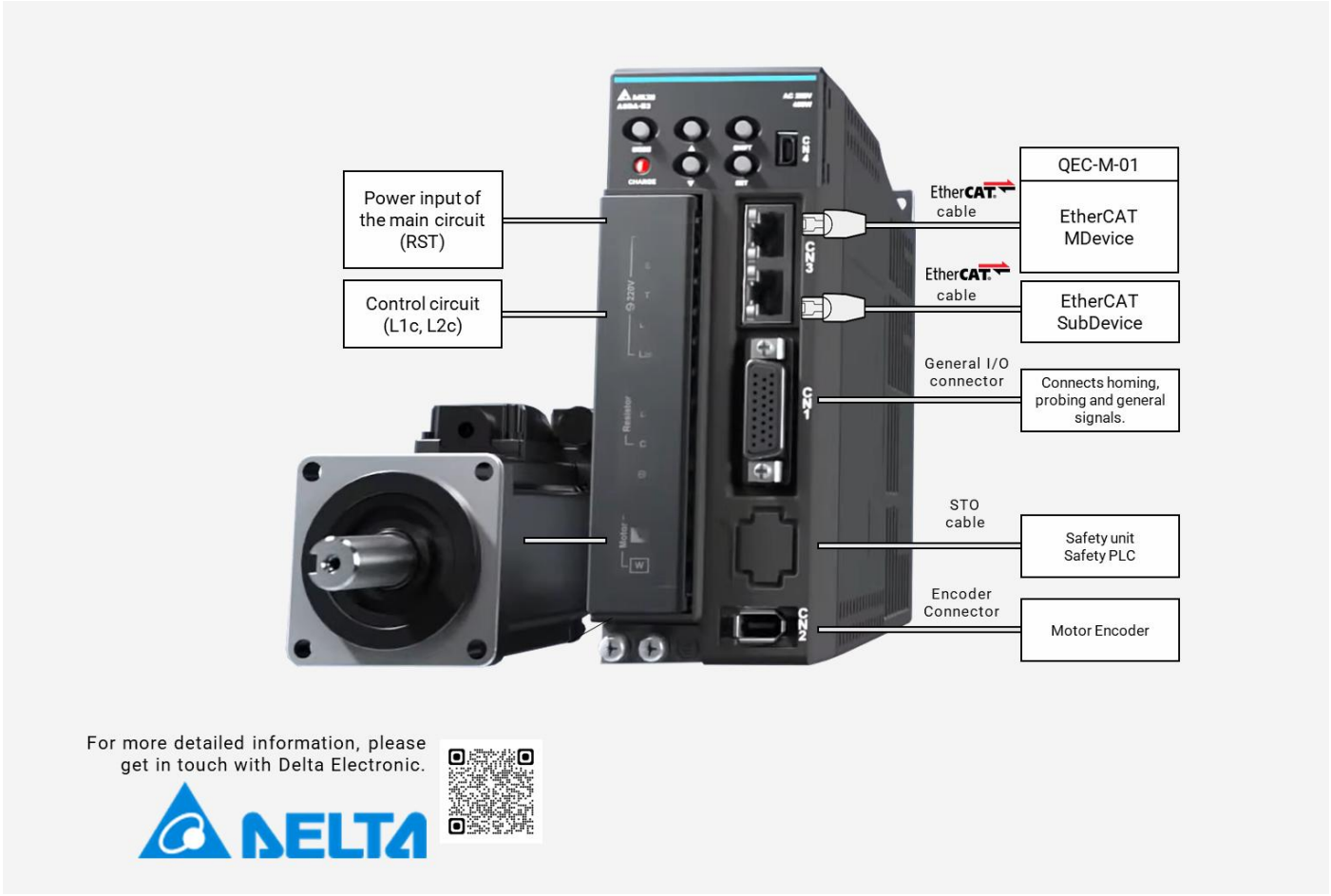
QEC EtherCAT MDevice.

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 the Delta EtherCAT Servos via RJ45 cable.



1.2 Delta ASDA-B3

The **Delta ASDA-B3** is an AC Servo motor and drive, with a built-in EtherCAT interface compliant with CiA402 motion profiles. This figure shows an example of when the ASDA-B3 motor is connected.



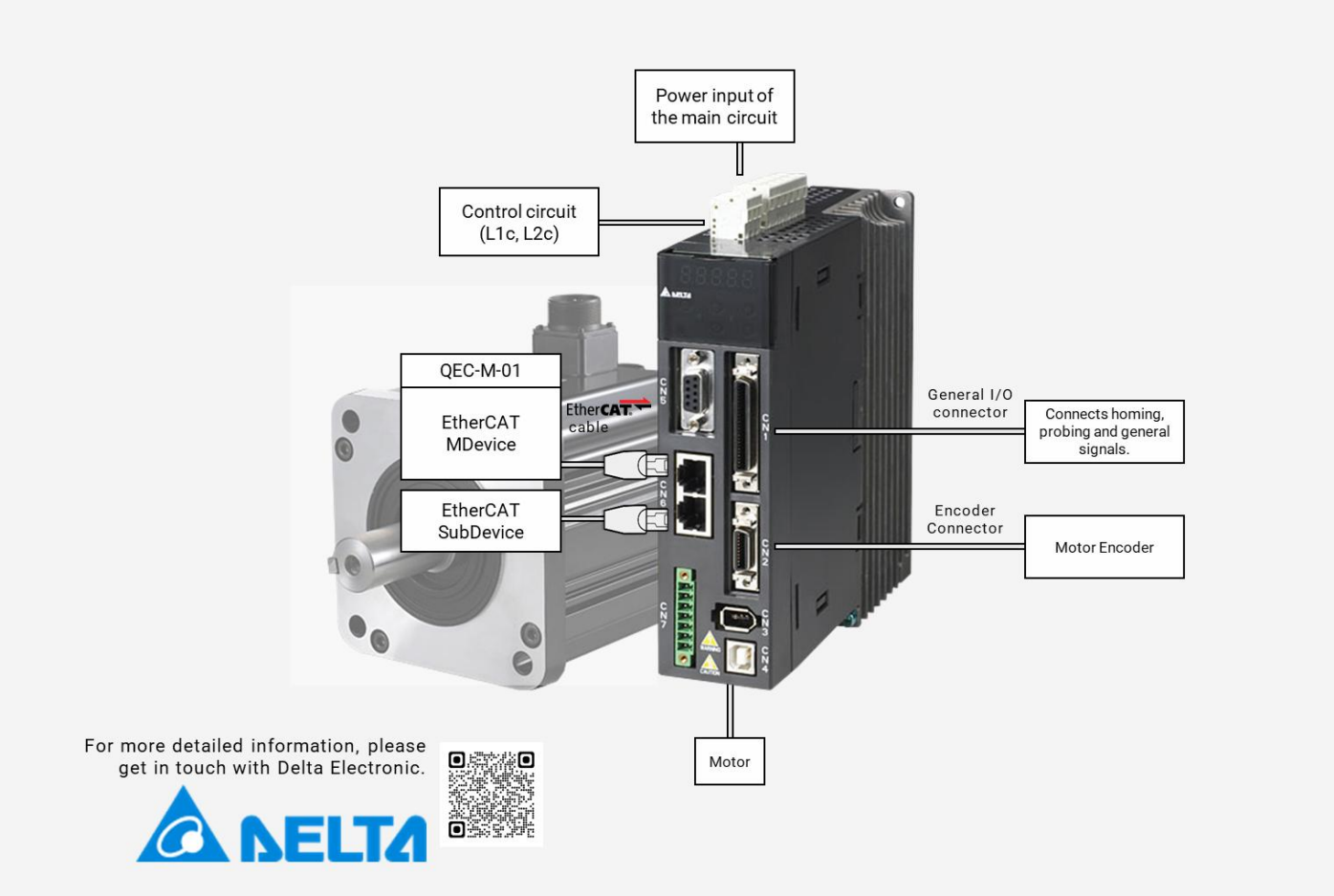
The connections are grouped by function:

1. EtherCAT connectivity
    - QEC-M-01 → Drive EtherCAT IN; EtherCAT OUT → next SubDevice (if any).
    - Put the network to OP before issuing CiA402 commands.
2. Main circuit power (RST)
    - Connect AC power to R/S/T. Install an MCCB and Magnetic Contactor upstream.
    - Supported: single/three-phase 200–230 V (≤3 kW);
      three-phase 380–400 V (1–7.5 kW).
3. Control power (L1C / L2C)
    - Supply control circuitry at L1C/L2C (AC 200–240 V).
4. Servo motor output (UVW)
    - Connect U/V/W to the servo motor power input.
5. Regenerative resistor (P⊕ D C)
    - Built-in resistor is sufficient for most applications.
      Connect external resistor for high-inertia loads.

6. I/O connector (CN1)
    - Connects ORG, POT/NOT limit switches, and general DI/DO signals for QEC homing and motion protection.
7. STO (CN10)
    - Must be active to enable the drive. Bridge the connector during development if no Safety PLC is used.
8. Encoder connector (CN2)
    - Use the manufacturer-supplied encoder cable.

1.3 Delta ASDA-A2

The **Delta ASDA-A2** is an AC Servo motor and drive, with a built-in EtherCAT interface compliant with CiA402 motion profiles. This figure shows an example of when the ASDA-A2 motor is connected.



The connections are grouped by function:

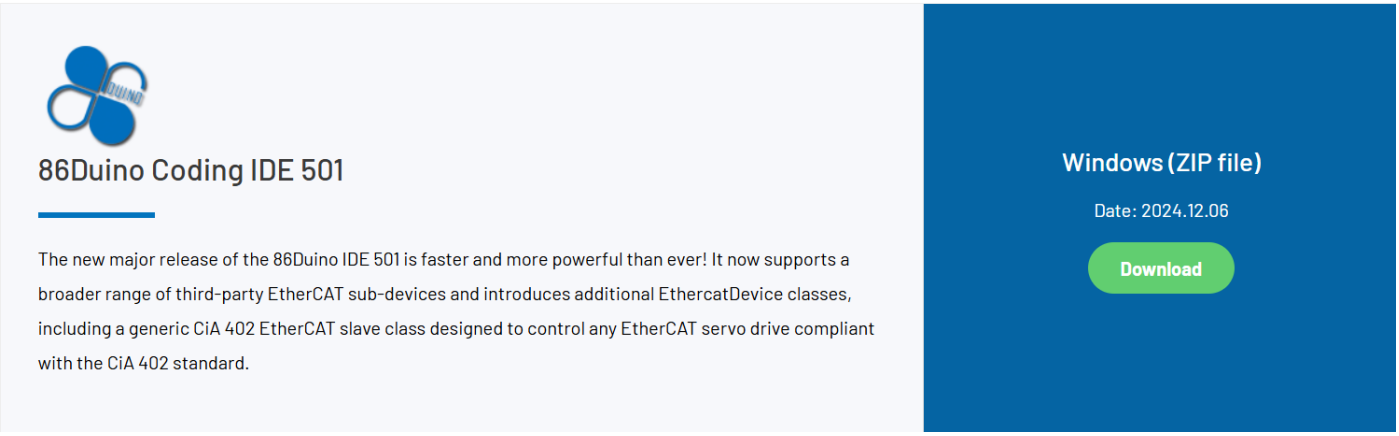
1. EtherCAT connectivity
    - QEC-M-01 → Drive EtherCAT IN; EtherCAT OUT → next SubDevice (if any).
    - Put the network to OP before issuing CiA402 commands.
2. Main circuit power (RST)
    - Connect AC power to R/S/T. Install an NFB (No Fuse Breaker) and Electromagnetic Contactor upstream.
    - Supported: single/three-phase 200–230 V (200 W–1.5 kW); three-phase 200–230 V (2–3 kW).
3. Control power (L1C / L2C)
    - Supply control circuitry at L1C/L2C (AC 200–230 V).
4. Servo motor output (UVW)
    - Connect U/V/W to the servo motor power input.
5. Regenerative resistor (P⊕ / C / D)
    - Built-in resistor is sufficient for most applications.

- For external resistor: connect between P⊕ and C, and open the circuit between P⊕ and D.

6. I/O connector (CN1)
   - Connects ORG, POT/NOT limit switches, and general DI/DO signals for QEC homing and motion protection.
   - An optional terminal block module (ASD-BM-50A) can be used for easier wiring.
7. Encoder connector (CN2)
   - Use the manufacturer-supplied encoder cable.

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.

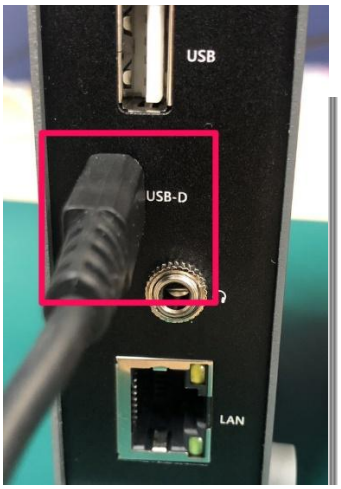
*__Note__: If Windows displays a warning, click Details once and then click the Continue Run button once.

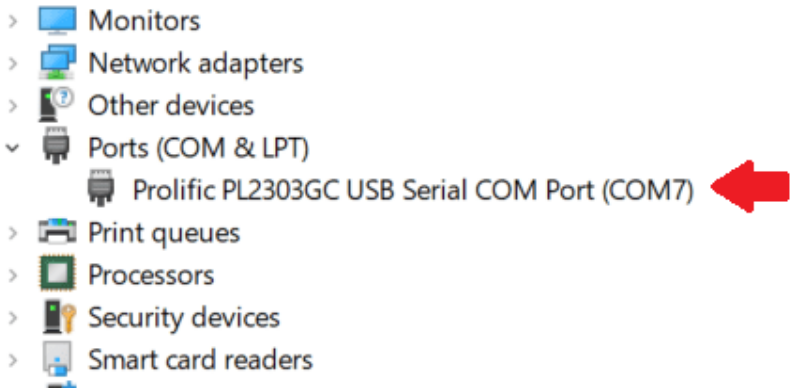
86Duino Coding IDE 501+ looks like below.

3. Connect to PC and set up the environment

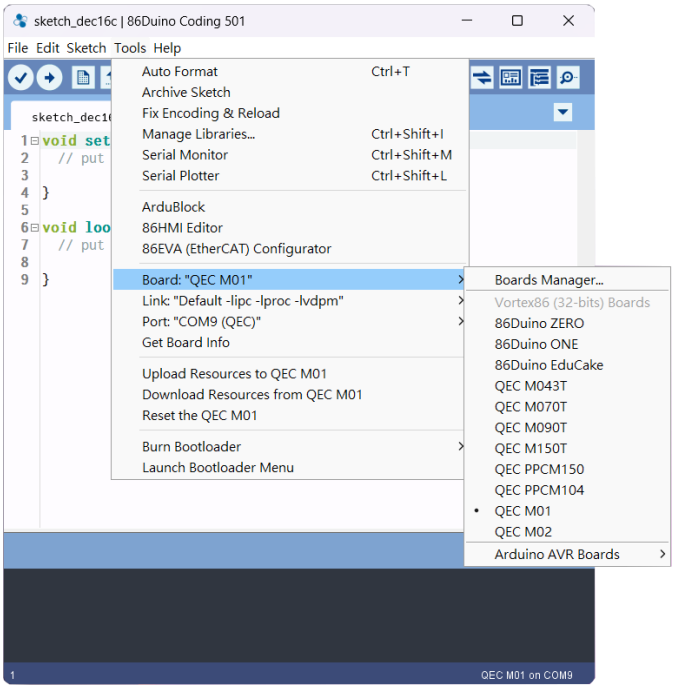
Follow the steps below to set up the environment:

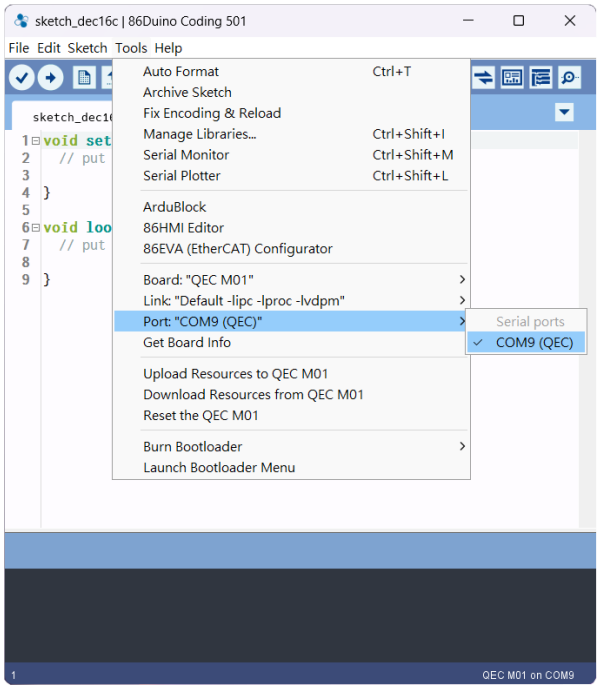
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “**Device Manager**” (select in the menu after pressing Win+X) ->” **Ports (COM & LPT)**” in your PC and expand the ports; you should see that the “**Prolific PL2303GC USB Serial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

The EtherCAT MDevice (QEC-M-01) and the Delta ASDA drives 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 program. 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 two ASDA drives into **CiA402 Profile Position (PP) mode**:

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

The `EthercatMaster` object (`master`) represents the QEC-M-01, while the `EthercatDevice_CiA402` array (`motor[2]`) represents the two ASDA-B3 and ASDA-A2 drives.

A. In Setup Function

In the `setup()` function, initialize communication and configure the motor for CiA402 Profile Position (PP) mode. Follow the steps below:

1. Initialize Serial Communication
    - Use `Serial.begin(115200)` to initialize serial communication.
2. Start the EtherCAT MDevice
    - Use `begin()` to initialize the EtherCAT MDevice and set the state machine to PRE-OPERATIONAL.
3. Attach the ASDA EtherCAT SubDevices
    - Use `attach(i, master)` to register each drive by its SubDevice index. Index 0 = first drive (ASDA-B3), Index 1 = second drive (ASDA-A2).
4. Enable Distributed Clock
    - Use `setDc(1000000)` on each drive to enable DC sync with a 1 ms cycle time.
5. Set CiA402 mode to Profile Position (PP)
    - Configure each drive using `setCiA402Mode(CIA402_PP_MODE)`.
6. Start the EtherCAT MDevice
    - Use `start(1000000, ECAT_SYNC)` to transition the network to OPERATIONAL.
7. Enable the Motors
    - Use `enable()` on each drive to transition to CIA402_OPERATION_ENABLED.
8. Configure Profile Parameters
    - Set motion profile type, velocity, acceleration, and deceleration for each drive.

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:

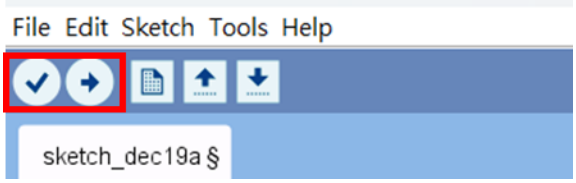
1. State Machine Logic
    - case 0: Move to +10,000,000 counts. Once the command is successfully executed, transition to the next state.
    - case 1: Wait for the motor to reach the target position. Once the target is reached, proceed to the next state.
    - case 2: Move to −10,000,000 counts. Once 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:

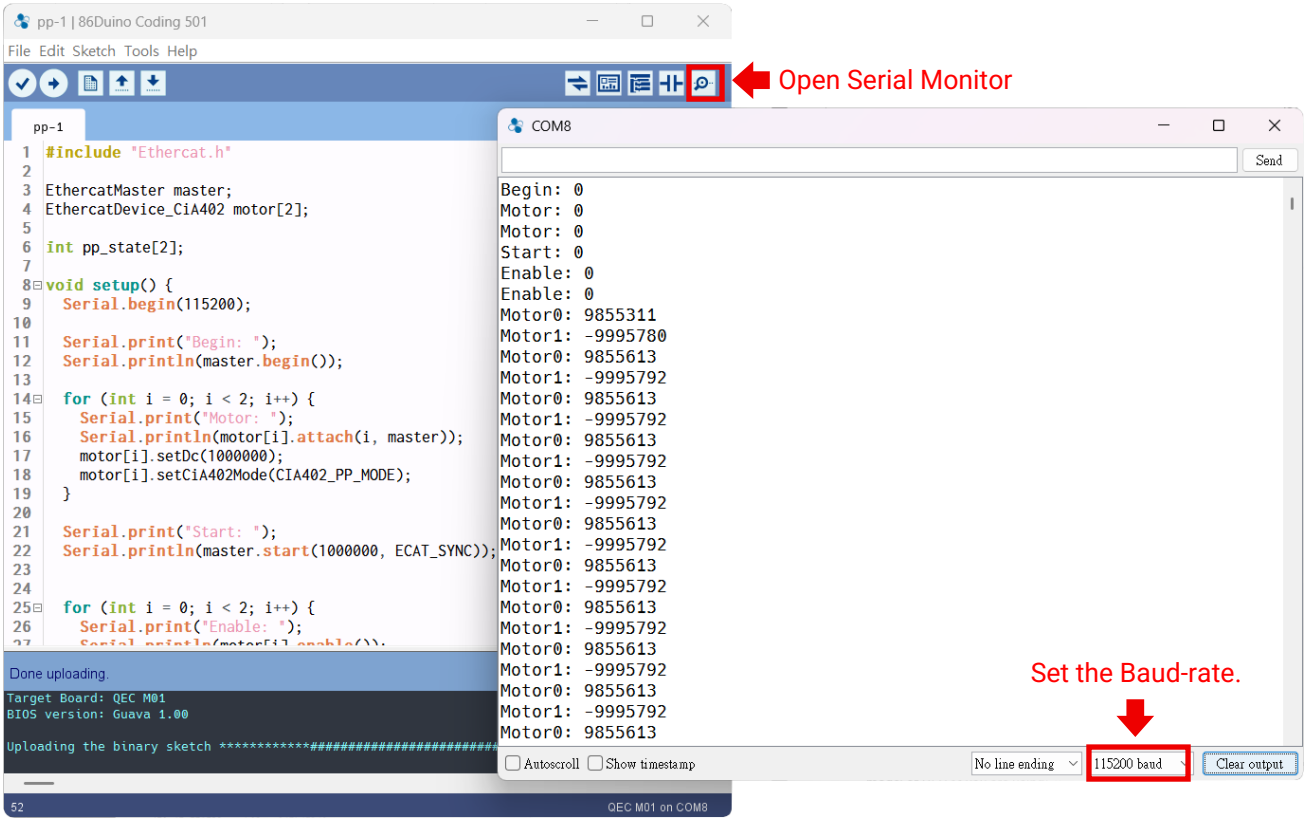
```
#include "Ethercat.h"

EthercatMaster master;
EthercatDevice_CiA402 motor[2];

int pp_state[2];

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

  Serial.print("Begin: ");
  Serial.println(master.begin());
  for (int i = 0; i < 2; i++) {
    Serial.print("Motor: ");
    Serial.println(motor[i].attach(i, master));
    motor[i].setDc(1000000);
    motor[i].setCiA402Mode(CIA402_PP_MODE);
  }
  Serial.print("Start: ");
  Serial.println(master.start(1000000, ECAT_SYNC));


  for (int i = 0; i < 2; i++) {
```

```
    Serial.print("Enable: ");
    Serial.println(motor[i].enable());
    motor[i].pp_SetMotionProfileType(0);
    motor[i].pp_SetVelocity(1000000);
    motor[i].pp_SetAcceleration(1000000);
    motor[i].pp_SetDeceleration(1000000);
  }
}

void loop() {
  for (int i = 0; i < 2; i++) {
    Serial.print("Motor"); Serial.print(i); Serial.print(": ");
    Serial.println(motor[i].getPositionActualValue());

    switch (pp_state[i]) {
      case 0:
        if (motor[i].pp_Run(10000000) == 0)
          pp_state[i]++;
        break;
      case 1:
        if (motor[i].pp_IsTargetReached())
          pp_state[i]++;
        break;
      case 2:
        if (motor[i].pp_Run(-10000000) == 0)
          pp_state[i]++;
        break;
      case 3:
        if (motor[i].pp_IsTargetReached())
          pp_state[i] = 0;
        break;
    }
  }
}
```

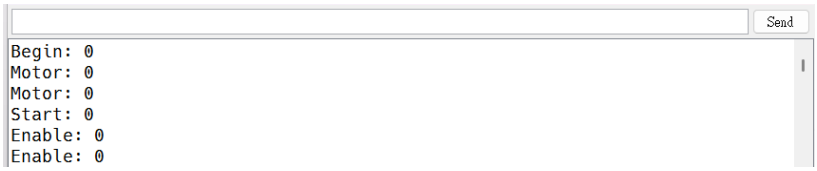
* **Note:** Once the code is written, click on the toolbar to compile, and to confirm that the compilation is complete and error-free, you can click to upload.



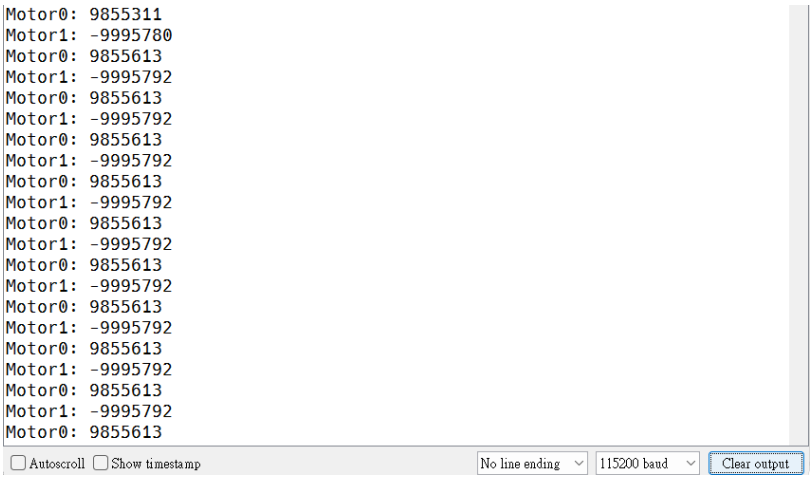
After you successfully upload the program to the QEC-M-01, you can open the Serial Monitor on the 86Duino IDE. Please check that the Serial baud rate is the same as your setting.



If the EtherCAT communication configuration is successful, the Serial Monitor will print "**0**" for each status for EtherCAT and "**Enable: 0**" for CiA402.



It will print the motor 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 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:

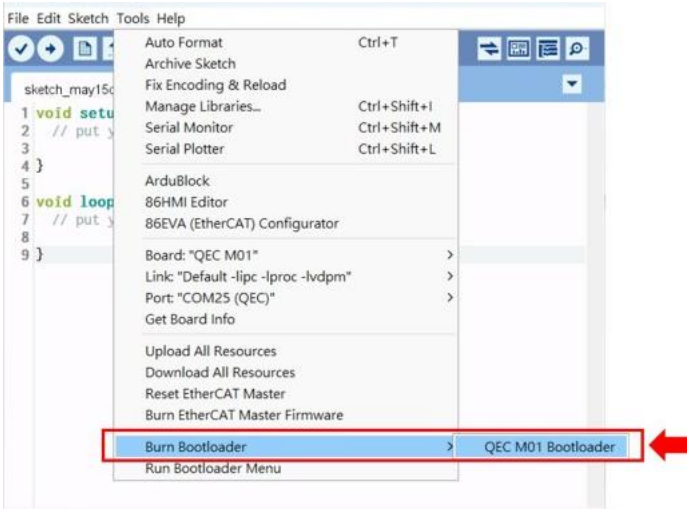
Step 1: Setting up QEC-M

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.

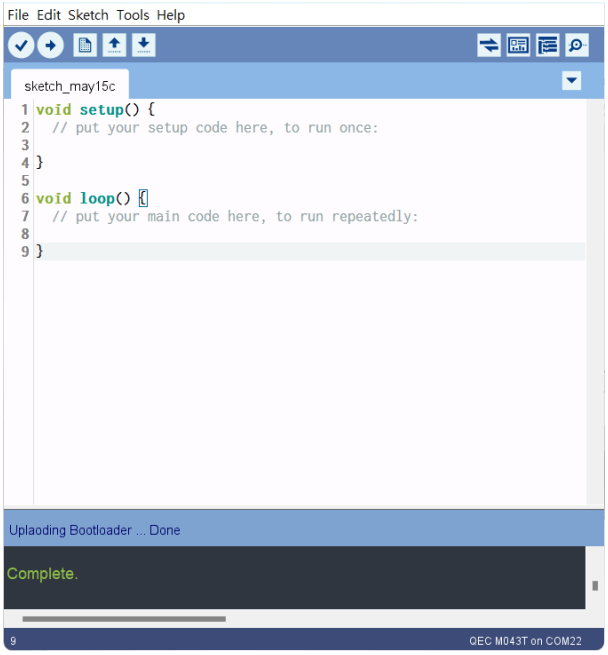
Step 2: Click “Burn Bootloader” button

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.