Start Guide

SANMOTION G - EtherCAT AC Servo

CiA402 CSP Mode (1-axis)

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

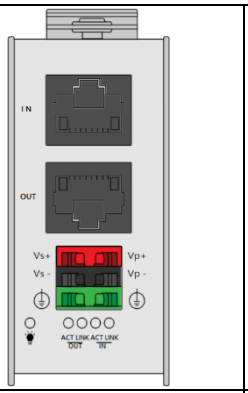
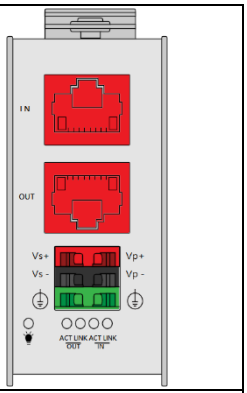
Date	Version	Description
2025/9/12	Version1.0	New Release.

Preface

In this guide, we will show you how to use the EtherCAT MDevice QEC-M-01 and the SANYO DENKI SANMOTION G Series **GADSA01AH24** (AC Servo Amplifier, EtherCAT Driver).

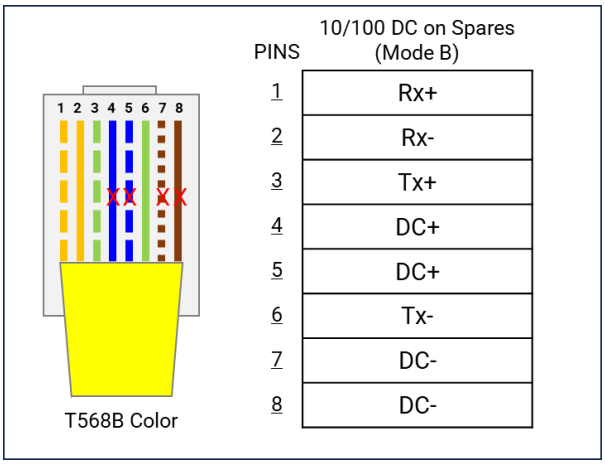
Notes QEC's PoE (Power over Ethernet)

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

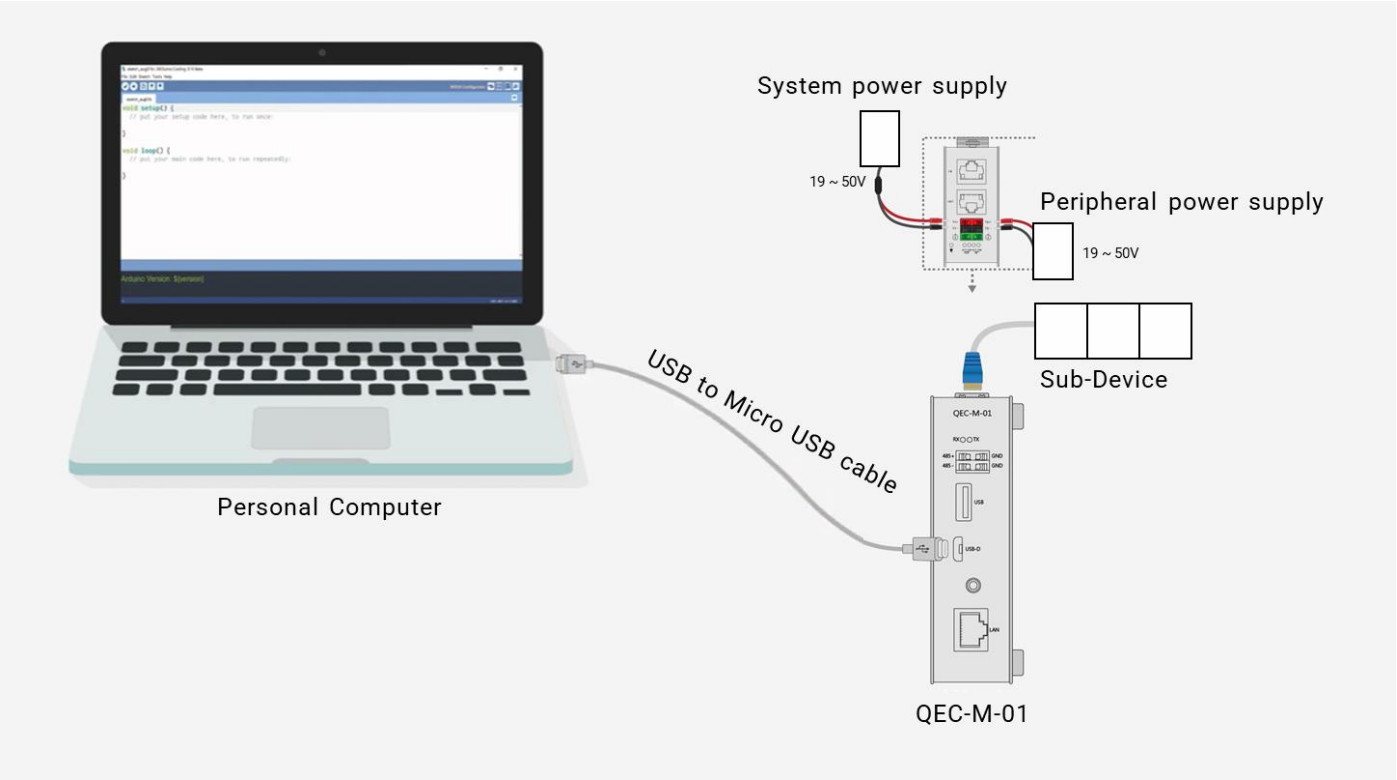
The following devices are used here:

1. QEC-M-01 (EtherCAT MDevice)
2. SANYO DENKI SANMOTION G Series GADSA01AH24 (AC Servo Amplifier, EtherCAT Driver)
3. 220 AC power & 24VDC power supply & EU-type terminal cable & LAN cable
4. GAM2A4005 (AC Servo Motor, flange size 40 mm square)

1.1 QEC-M-01

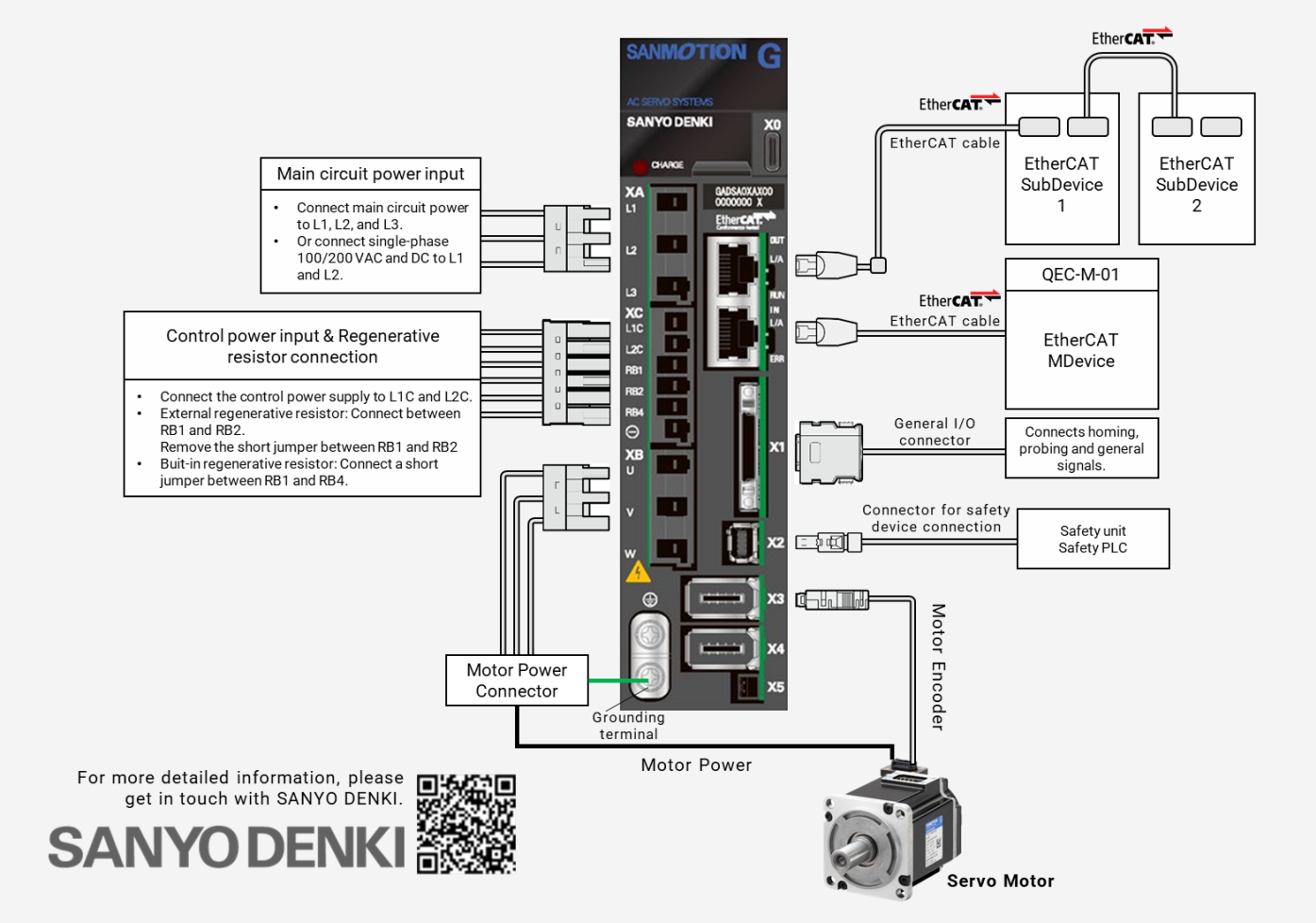
QEC EtherCAT MDevice.

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 GADSA01AH24

The **GADSA01AH24** is a SANMOTION G-series single-axis AC servo amplifier with a built-in EtherCAT interface compliant with CiA402 motion profiles. This figure shows an example of when the **GAM2A4005F0XRK0** motor is connected.



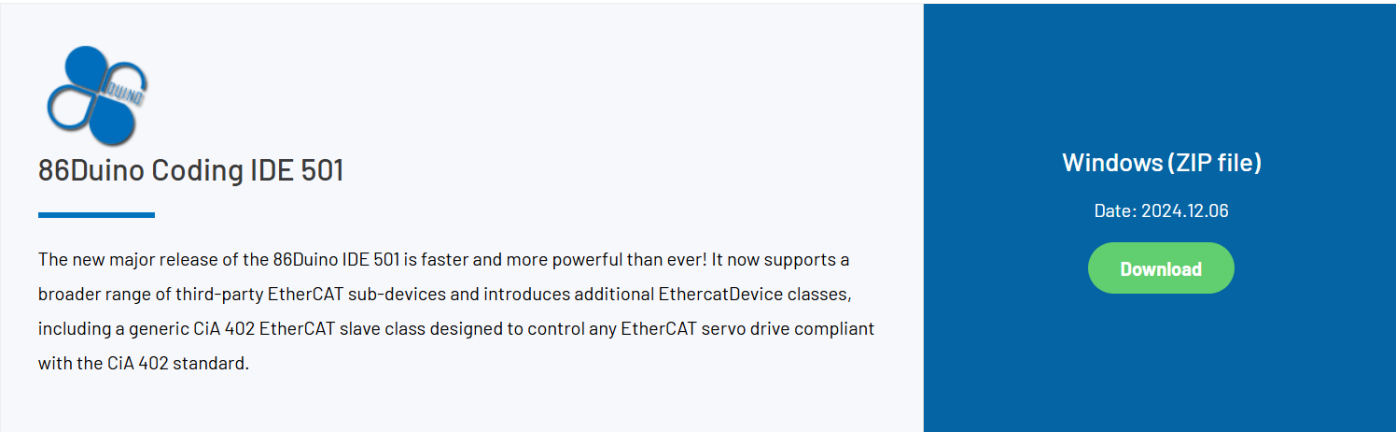
The connections are grouped by function:

1. EtherCAT connectivity
   - QEC-M01 → Drive EtherCAT IN with an EtherCAT cable.
   - Optional daisy-chain from drive EtherCAT OUT to other EtherCAT SubDevices.
   - Put the network to OP before issuing CiA402 commands.
2. Main circuit power (motor power stage)
   - Connect AC power to L1/L2/L3; for single-phase use L1/L2 (AC 200–240 V).
   - Ground the chassis at the PE terminal.
   - *Single-phase note:* set System1 ID01 = 01 (AC single-phase) to avoid phase-loss alarms (AL.63.0).
3. Control power & regenerative resistor
   - Supply the control circuitry at L1C/L2C (per model: AC 200–240 V or DC 24 V).
   - External regenerative resistor: connect RB1–RB2.
   - Built-in resistor (when applicable): short RB1–RB4 per manual.

4. Motor & encoder
   - U/V/W → servo motor power connector; keep motor PE bonded to drive PE.
   - X3/X4 → motor encoder (ES+/ES−, 5 V/SG, Shield).
   - X5 (optional) → encoder battery for absolute encoders.
5. Safety (STO) & general I/O
   - X2 → Safety PLC / safety relay for STO (both channels must be enabled; STO cannot be released by software).

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.

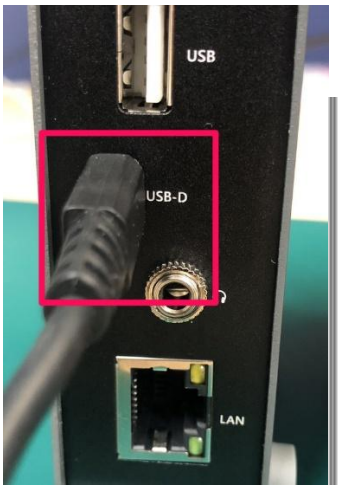
***Note**: If Windows displays a warning, click Details once and then click the Continue Run button once.

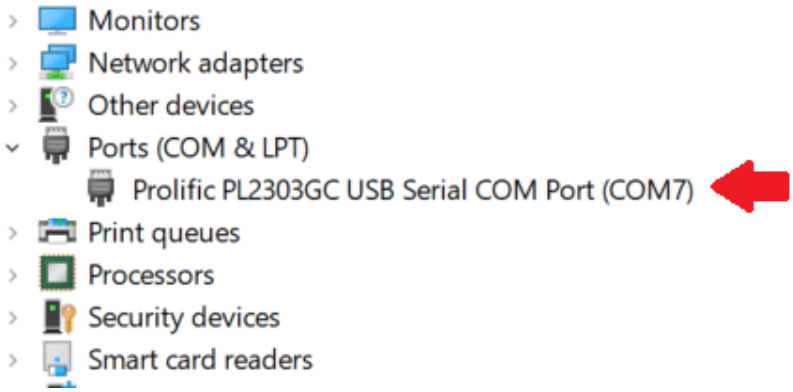
86Duino Coding IDE 501+ looks like below.

3. Connect to PC and set up the environment

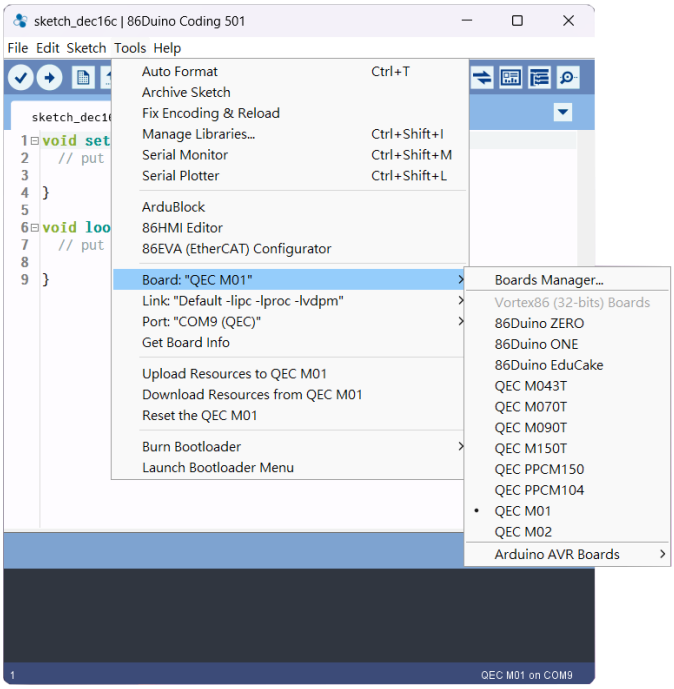
Follow the steps below to set up the environment:

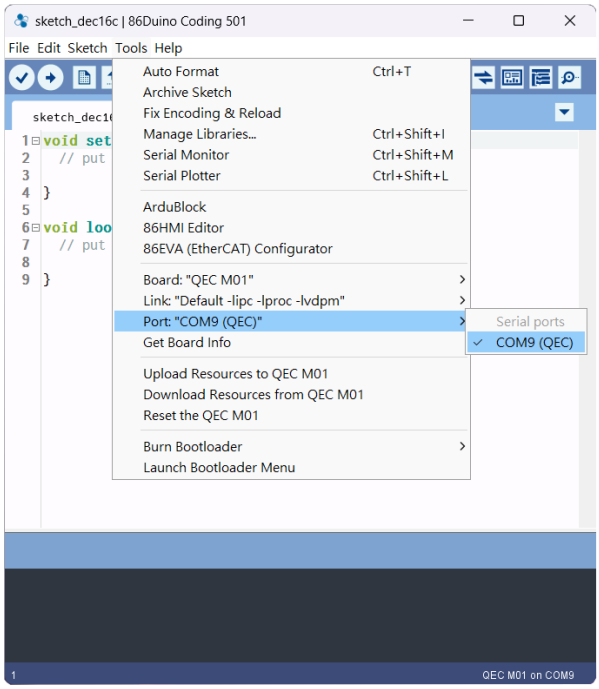
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “**Device Manager**” (select in the menu after pressing Win+X) ->” **Ports (COM & LPT)**” in your PC and expand the ports; you should see that the “**Prolific PL2303GC USB Serial 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 SANMOTION G Driver (GADSA01AH24) 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 GADSA01AH24 into CiA402 Cyclic-Synchronous Position (CSP) mode:

- EtherCAT Cycle Time: 1 millisecond.
- EtherCAT Mode: ECAT_SYNC.
- Motion is generated inside a 1 ms cyclic callback by stepping a command position at a fixed rate (constant velocity).

The `EthercatMaster` object (“master”) represents the QEC-M-01, while the `EthercatDevice_CiA402` object (“motor”) represents the GADSA01AH24 driver.

A. In Setup Function:

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

1. Initialize Serial Communication
    - Start serial communication at a baud rate of 115200.
2. Start the EtherCAT MDevice
    - Use the `begin()` function to begin the EtherCAT MDevice and set the EtherCAT state machine to the PRE-OPERATIONAL state.
3. Attach the GADSA01AH24 EtherCAT SubDevice
    - Use the `attach()` function to attach the EtherCAT SubDevice to the EtherCAT Network. Set the node number and the specific MDevice.
4. Configure Distributed Clocks (cycle = 1 ms)
    - `setDc(1000000)` to match a 1 ms control period.
5. Set the CiA402 mode to Cyclic-Synchronous Position (CSP) Mode
    - Configure the motor to CSP mode using `setCiA402Mode(CIA402_CSP_MODE)`.
6. Mapping the minimal PDO mapping that CSP mode needed to the GADSA01AH24 (No PDO mapping by default, needs to be filled out by users)
    - **RxPDO:**
      Use the `sdoDownload8(0x1C12, 0x00, 0)` and `sdoDownload8(0x1601, 0x00, 0)` to empty the RxPDO index. And use the `sdoDownload32()` to input the Controlword (0x6040) and Target position (0x607A) to map to the Index 0x1601, then use the `sdoDownload8()` to map Index 0x1601 to the Index 0x1C12.

- **TxPDO:**
  Use the `sdoDownload8(0x1C13, 0x00, 0)` and `sdoDownload8(0x1A01, 0x00, 0)` to empty the TxPDO index. And use the `sdoDownload32()` input the Statusword (0x6041) and Position actual value (0x6064) to map to the Index 0x1A01, then use the `sdoDownload8()` to map Index 0x1A01 to the Index 0x1C13.

7. Register the cyclic callback function (call this before `start()`)
   - Use the `attachCyclicCallback(MyCyclicCallback)` function to register the cyclic EtherCAT handler. It runs every cycle (e.g., 1 ms) in real time and must be PDO-only. Avoid SDO access, logging/Serial.print, delay, blocking calls, dynamic allocation, or heavy computation inside the callback.
8. Start the EtherCAT MDevice
   - Use the `start(1000000, ECAT_SYNC)` function to switch the EtherCAT state machine to the OPERATIONAL state. Set the cycle time to 1ms and ECAT_SYNC mode.
9. Initialize command position & Enable the motor
   - Set the first CSP target to the actual position for safety
     `motor.setTargetPosition(position = motor.getPositionActualValue());`
   - Use the `enable()` function to enable the motor and transition it to CiA402_OPERATION_ENABLED.

B. In Loop Function:

The `loop()` prints the actual position for monitoring. All motion logic runs in the cyclic callback.

C. In Callback Function

The `MyCyclicCallback()` function emits a new CSP target every cycle time (1 ms) to achieve constant-velocity motion, increasing 10000 units every cycle.
Follow the steps below:

1. State Check to ensure the CiA402 State is in CiA402_OPERATION_ENABLED
   - If `getCiA402State() != CIA402_OPERATION_ENABLED`, then return to do nothing.
2. Generate CSP Position Command
   - Increment a software position accumulator (e.g., position += 10000) and call `setTargetPosition(position)` each cycle.

This produces a constant position ramp. With a 1 ms cycle, +10,000 counts/cycle ≈ 10 M counts/s—adjust to match the drive/motor limits.

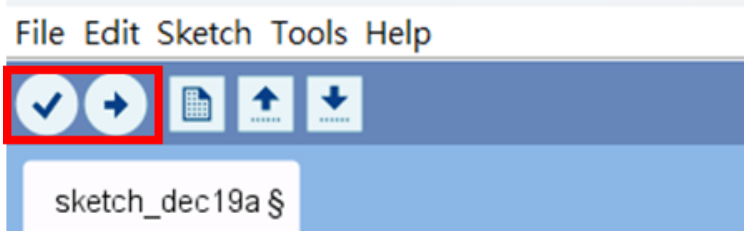
The example code is as follows:

```
#include "Ethercat.h"

EthercatMaster master;
EthercatDevice_CiA402 motor;
```

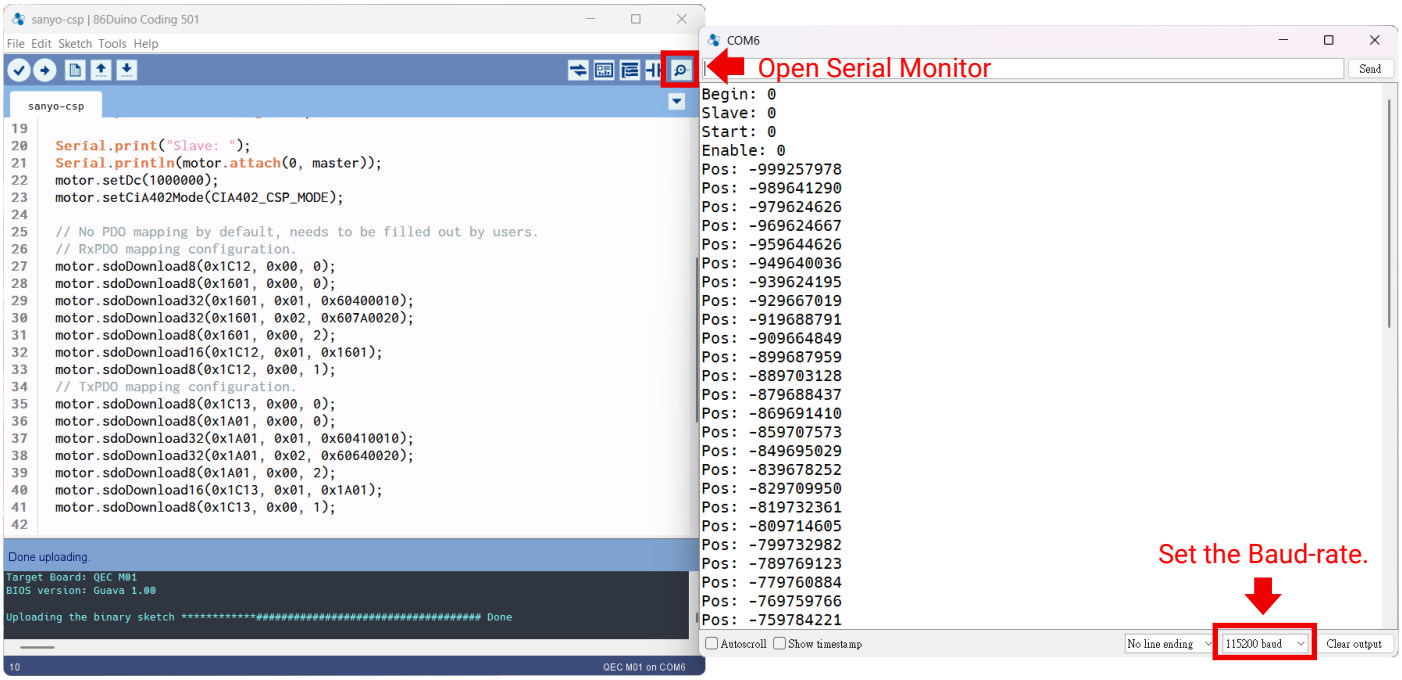
```
int32_t position = 0;

void MyCyclicCallback() {
  if (motor.getCiA402State() != CIA402_OPERATION_ENABLED)
    return;
  motor.setTargetPosition(position += 10000);
}

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

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

  Serial.print("Slave: ");
  Serial.println(motor.attach(0, master));
  motor.setDc(1000000);
  motor.setCiA402Mode(CIA402_CSP_MODE);

  // No PDO mapping by default, needs to be filled out by users.
  // RxPDO mapping configuration.
  motor.sdoDownload8(0x1C12, 0x00, 0);
  motor.sdoDownload8(0x1601, 0x00, 0);
  motor.sdoDownload32(0x1601, 0x01, 0x60400010);
  motor.sdoDownload32(0x1601, 0x02, 0x607A0020);
  motor.sdoDownload8(0x1601, 0x00, 2);
  motor.sdoDownload16(0x1C12, 0x01, 0x1601);
  motor.sdoDownload8(0x1C12, 0x00, 1);
  // TxPDO mapping configuration.
  motor.sdoDownload8(0x1C13, 0x00, 0);
  motor.sdoDownload8(0x1A01, 0x00, 0);
  motor.sdoDownload32(0x1A01, 0x01, 0x60410010);
  motor.sdoDownload32(0x1A01, 0x02, 0x60640020);
  motor.sdoDownload8(0x1A01, 0x00, 2);
  motor.sdoDownload16(0x1C13, 0x01, 0x1A01);
  motor.sdoDownload8(0x1C13, 0x00, 1);

  master.attachCyclicCallback(MyCyclicCallback);
  Serial.print("Start: ");
  Serial.println(master.start(1000000, ECAT_SYNC));

  motor.setTargetPosition(position = motor.getPositionActualValue());
```

```
  Serial.print("Enable: ");
  Serial.println(motor.enable());
}

void loop() {
  Serial.print("Pos: ");
  Serial.println(motor.getPositionActualValue());
  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.

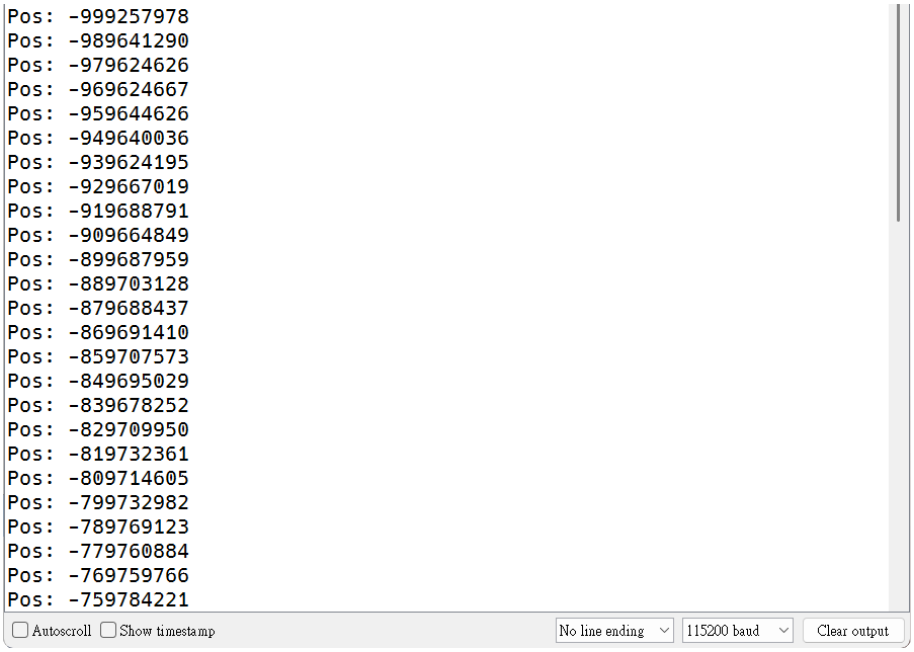


After you successfully upload the program to the QEC-M-01, you can open the Serial Monitor on the 86Duino IDE. Please check that the Serial baud rate is the same as your setting.



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’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 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:

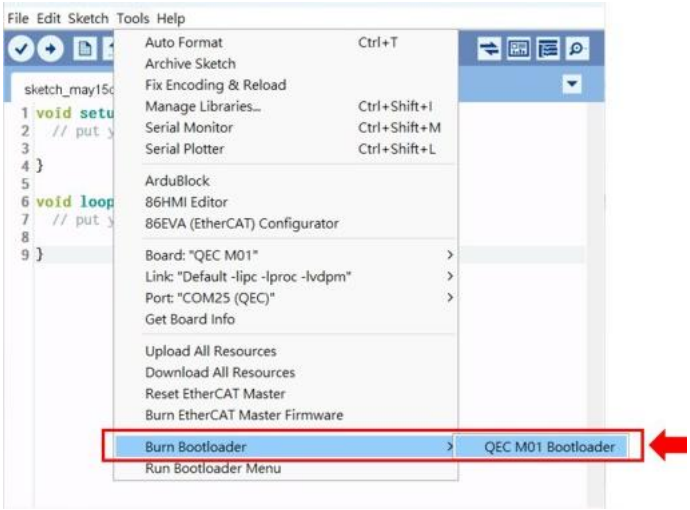
Step 1: Setting up QEC-M

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.

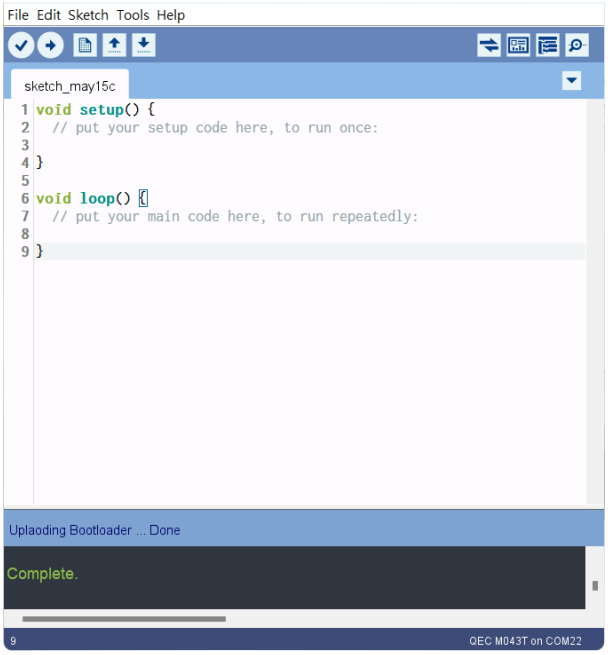
Step 2: Click “Burn Bootloader” button

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.