Start Guide

OrientalMotor AZ-mini EtherCAT Slave (PP mode) with 86EVA

86Duino Coding IDE 500

EtherCAT Library

(Version 1.0)

Revision

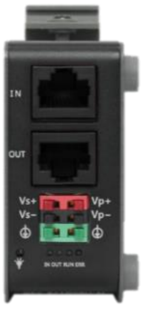
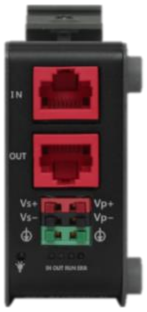
Date	Version	Description
2024/9/11	VERSION1.0	NEW RELEASE.

Preface

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

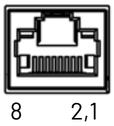
Notes QEC’s PoE (Power over Ethernet)

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:

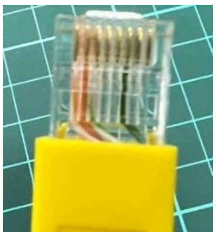
1. The PoE function of QEC is different and incompatible with EtherCAT P, and the PoE function of QEC is based on PoE Type B, and the pin functions are as follows:

	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).

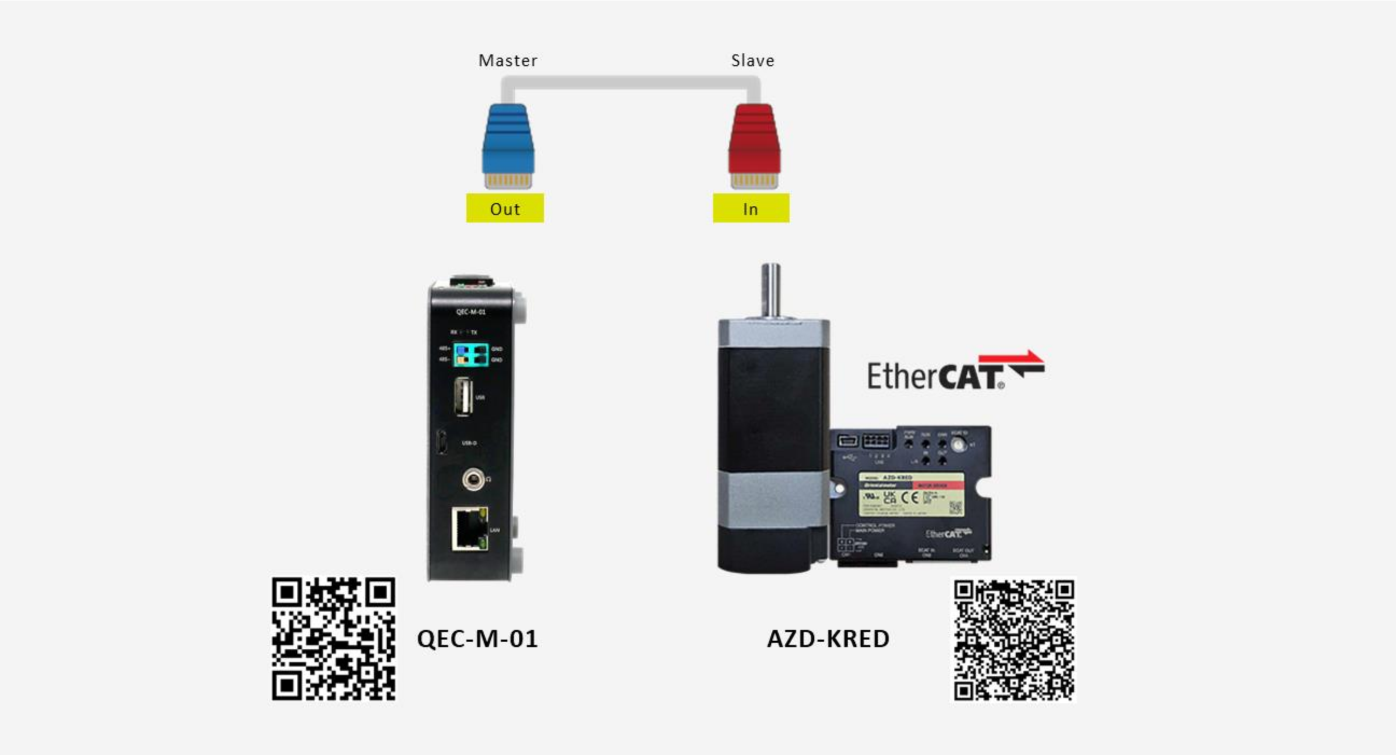


3. QEC’s PoE power supply is up to 24V/3A.

1. Connection and wiring hardware

The following devices are used here:

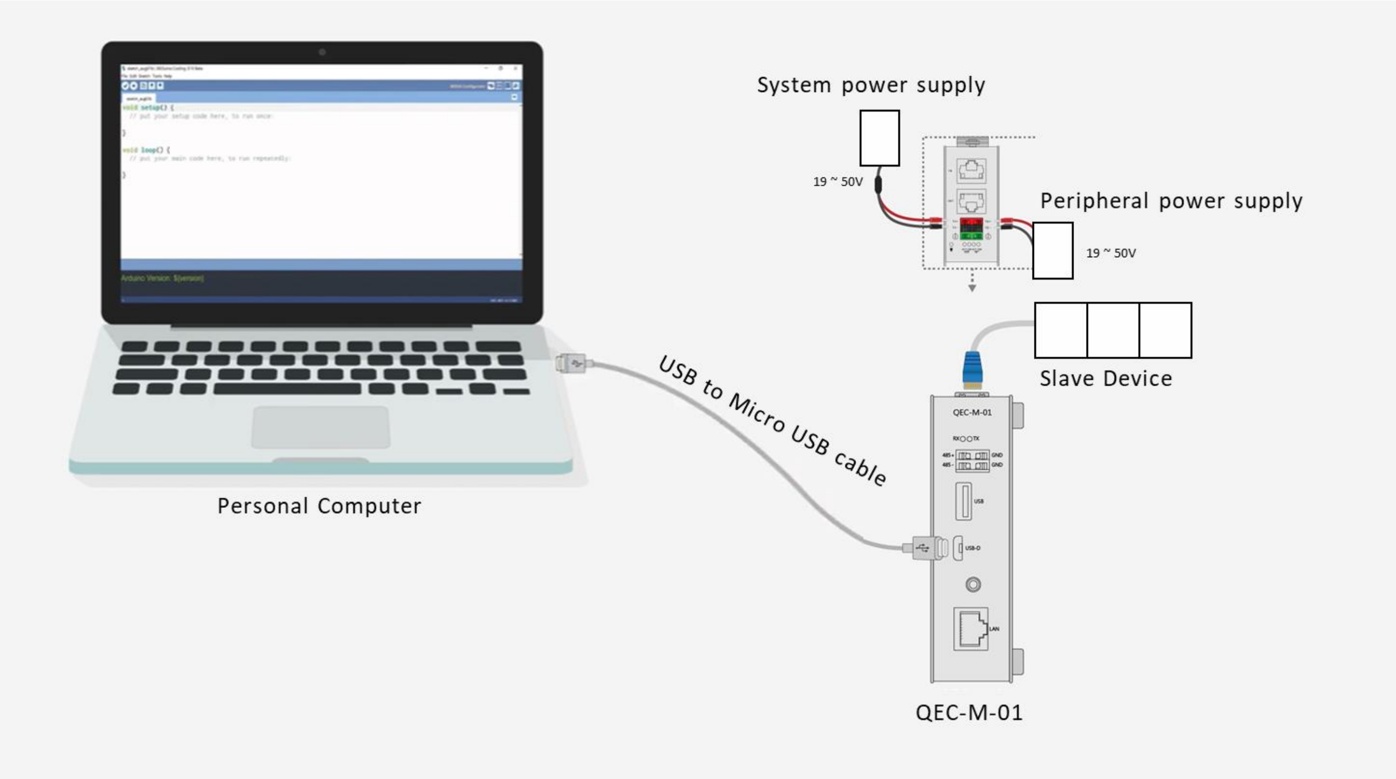
1. QEC-M-01 (EtherCAT Master)
2. AZD-KRED, AZ Series mini EtherCAT Driver (OrientalMotor Step-Servo Driver)
3. AZM48AK, 1.65 in. (42 mm) AZ Series Stepper Motor with Absolute Mechanical Encoder (DC Input)
4. 24V power supply



1.1 QEC-M-01

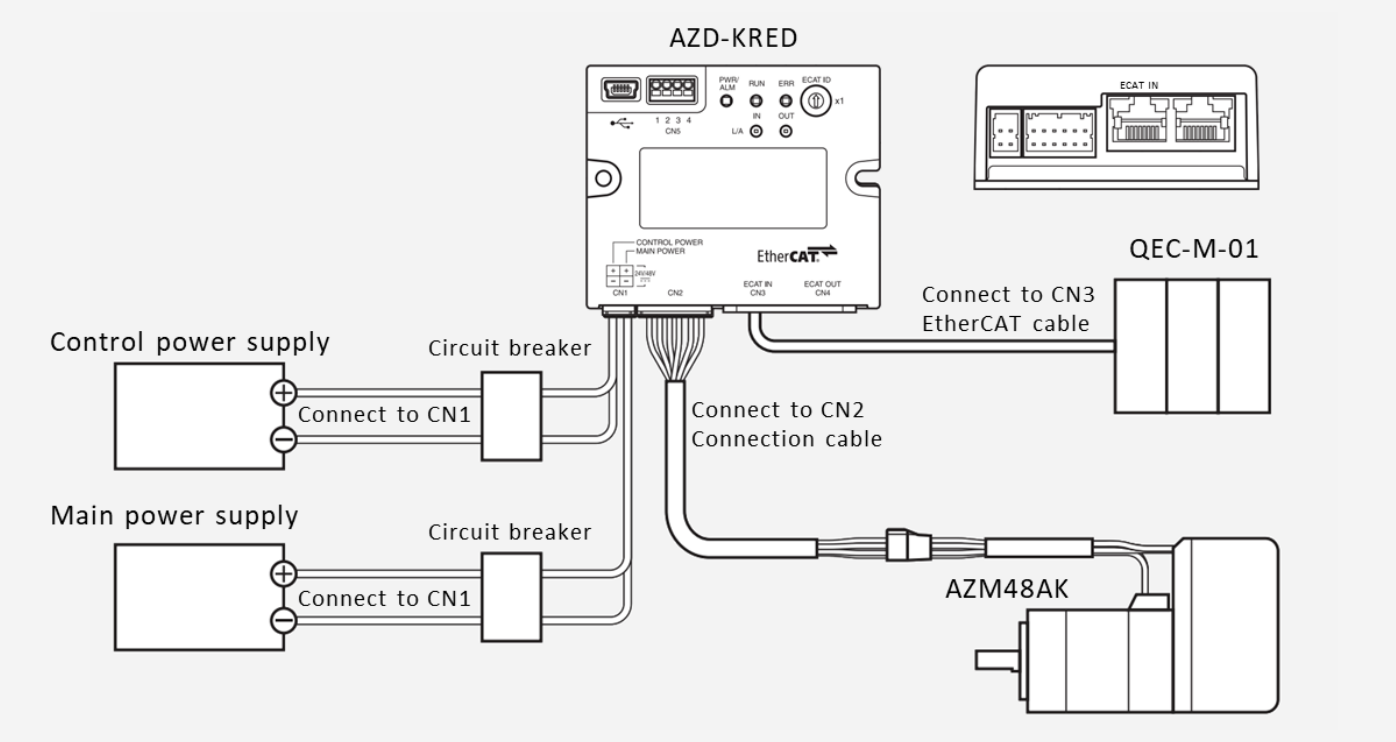
QEC EtherCAT Master.

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

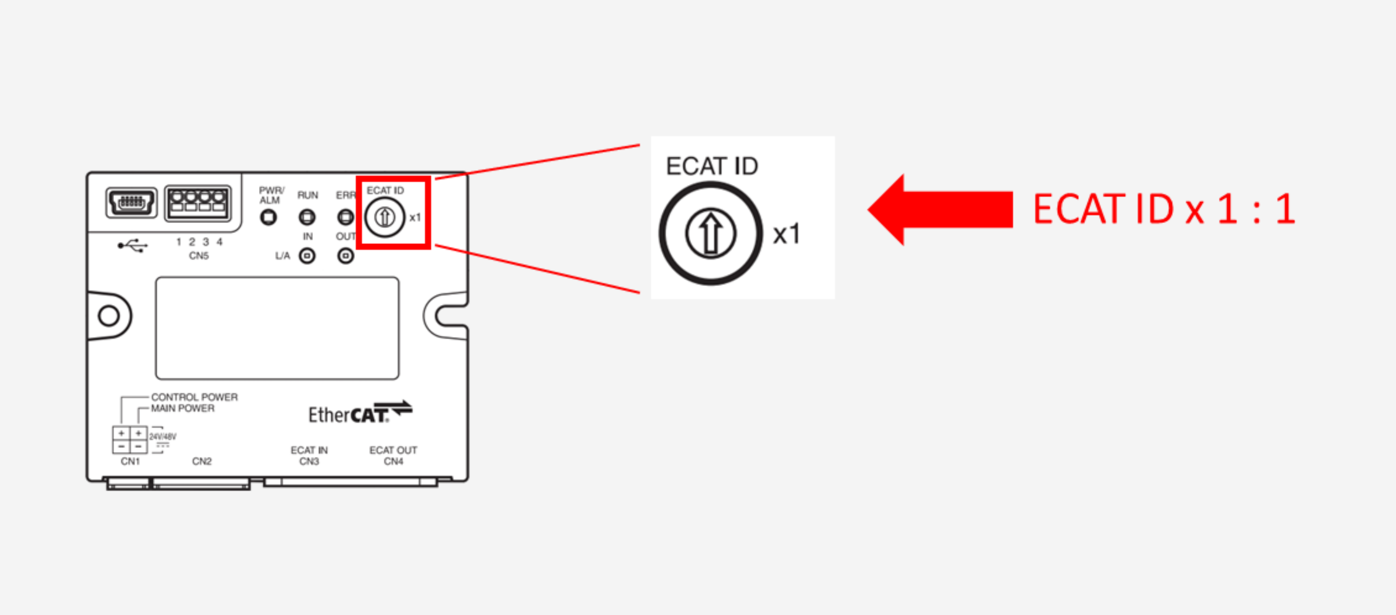
- Before connecting or disconnecting a connector, turn off the main power supply and the control power supply, and check the **PWR/ALM LED** has been turned off.
- When disconnecting the connector, pull out while pressing the latches on the connector with fingers.

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)

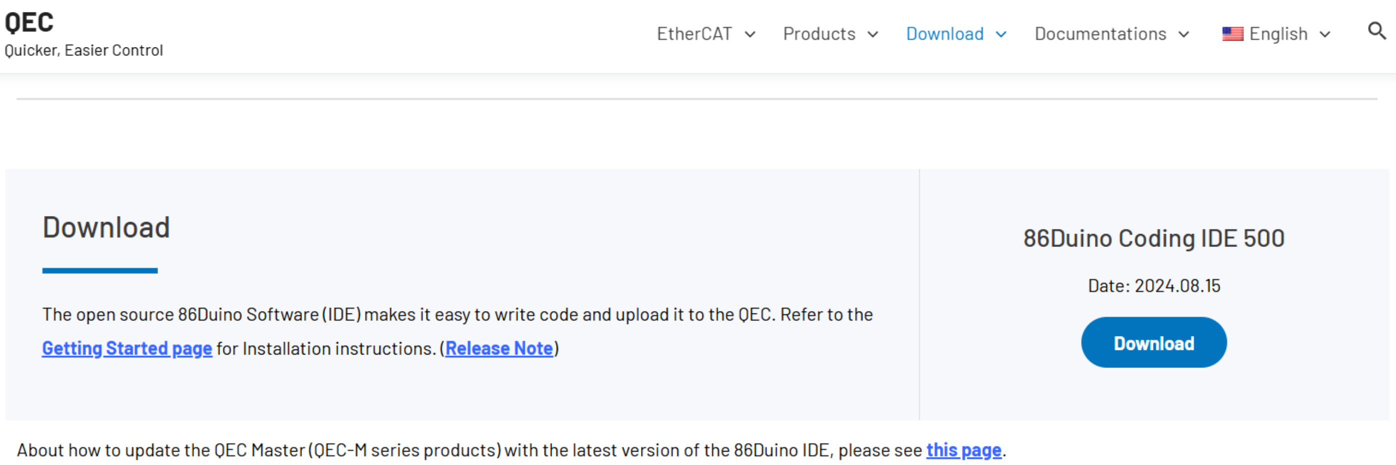


Setting Range	Description
0 (00h)	The master’s settings will be applied.
1-15 (01h-0Fh)	The driver's settings will be applied.

* 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/.



Download

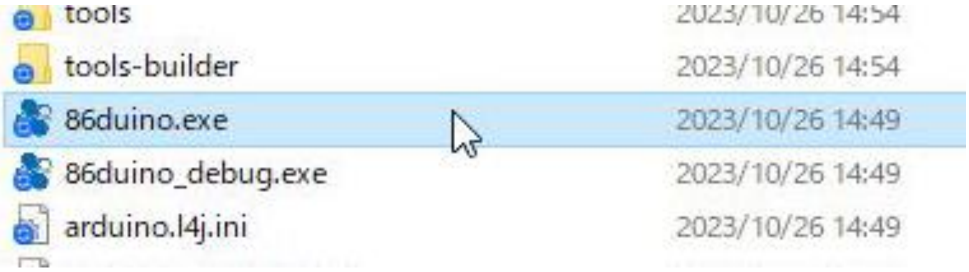
The open source 86Duino Software (IDE) makes it easy to write code and upload it to the QEC. Refer to the Getting Started page for Installation instructions. (Release Note)

86Duino Coding IDE 500

Date: 2024.08.15

About how to update the QEC Master (QEC-M series products) with the latest version of the 86Duino IDE, please see this page.

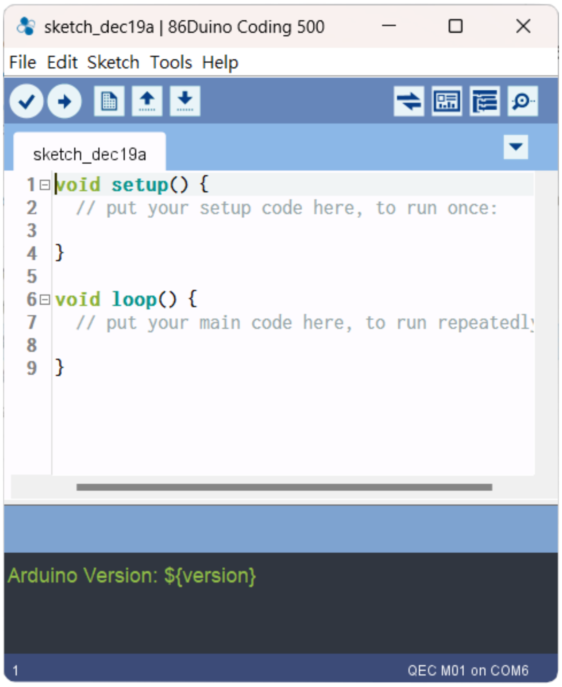
After downloading, please unzip the downloaded zip file, no additional software installation is required, just double-click 86duino.exe to start the IDE.



tools	2023/10/26 14:54
tools-builder	2023/10/26 14:54
86duino.exe	2023/10/26 14:49
86duino_debug.exe	2023/10/26 14:49
arduino.l4j.ini	2023/10/26 14:49

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

86Duino Coding IDE 500+ looks like below.



sketch_dec19a | 86Duino Coding 500
File Edit Sketch Tools Help
sketch_dec19a

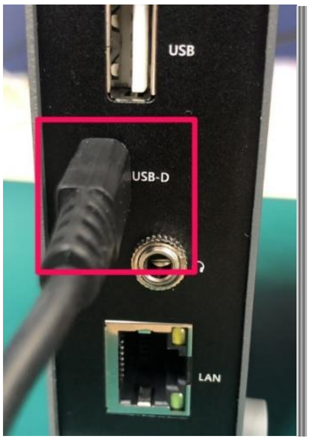
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedl
  
}
```

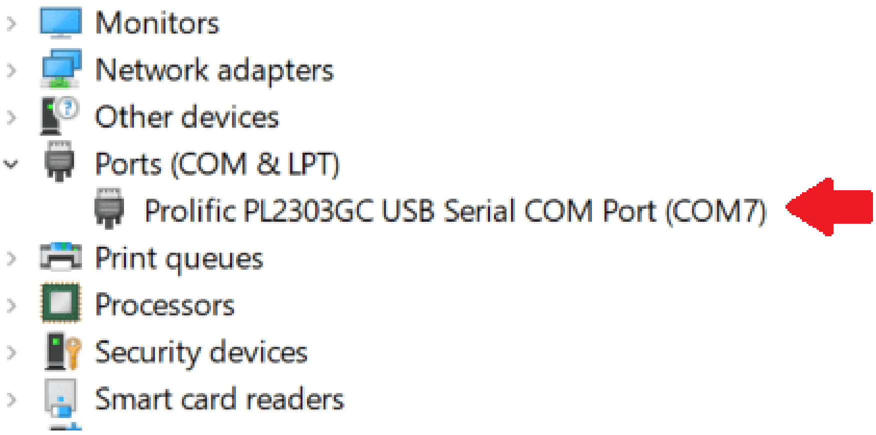
Arduino Version: ${version}

3. Connect to your PC and set up the environment

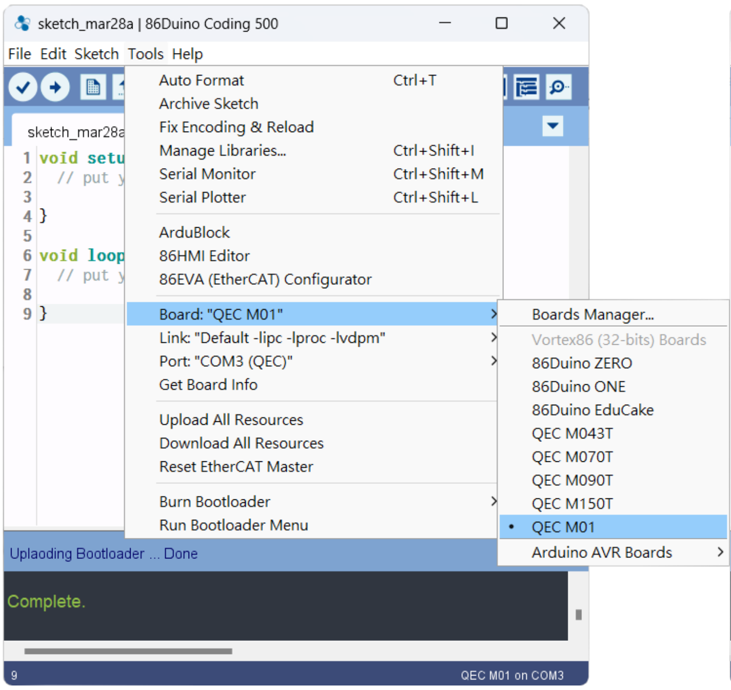
Follow the steps below to set up the environment:

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)





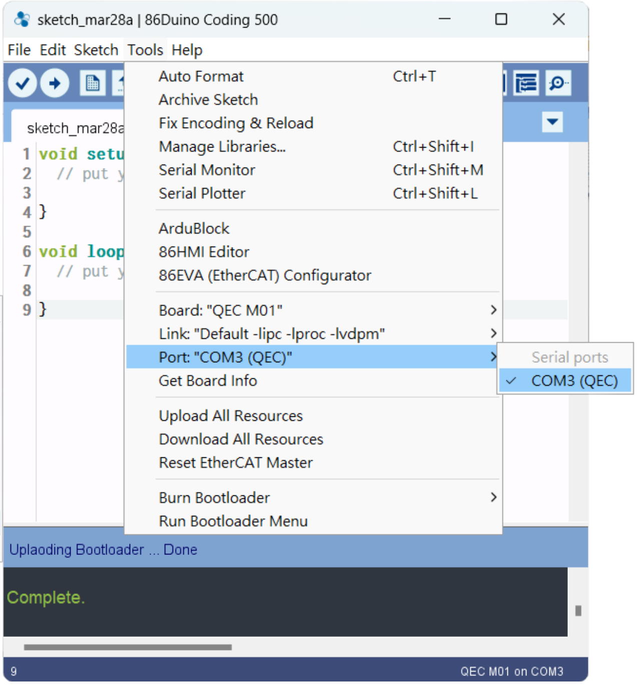
4. Open the 86Duino IDE.
5. Select the correct board: In the IDE's menu, select Tools> Board > QEC-M-01 (or the QEC-M master model you use).
6. Select Port: In the IDE's menu, select Tools > Port and select the USB port to connect to the QEC-M master (in this case, COM3 (QEC)).



Uplaoding Bootloader ... Done

Complete.

QEC M01 on COM3



Uplaoding Bootloader ... Done

Complete.

QEC M01 on COM3

4. Use 86EVA with code

This example shows how to operate the EtherCAT master (QEC-M-01) and the OrientalMotor AZ-mini EtherCAT Driver through the 86Duino IDE's graphical low-code programming tool, 86EVA.

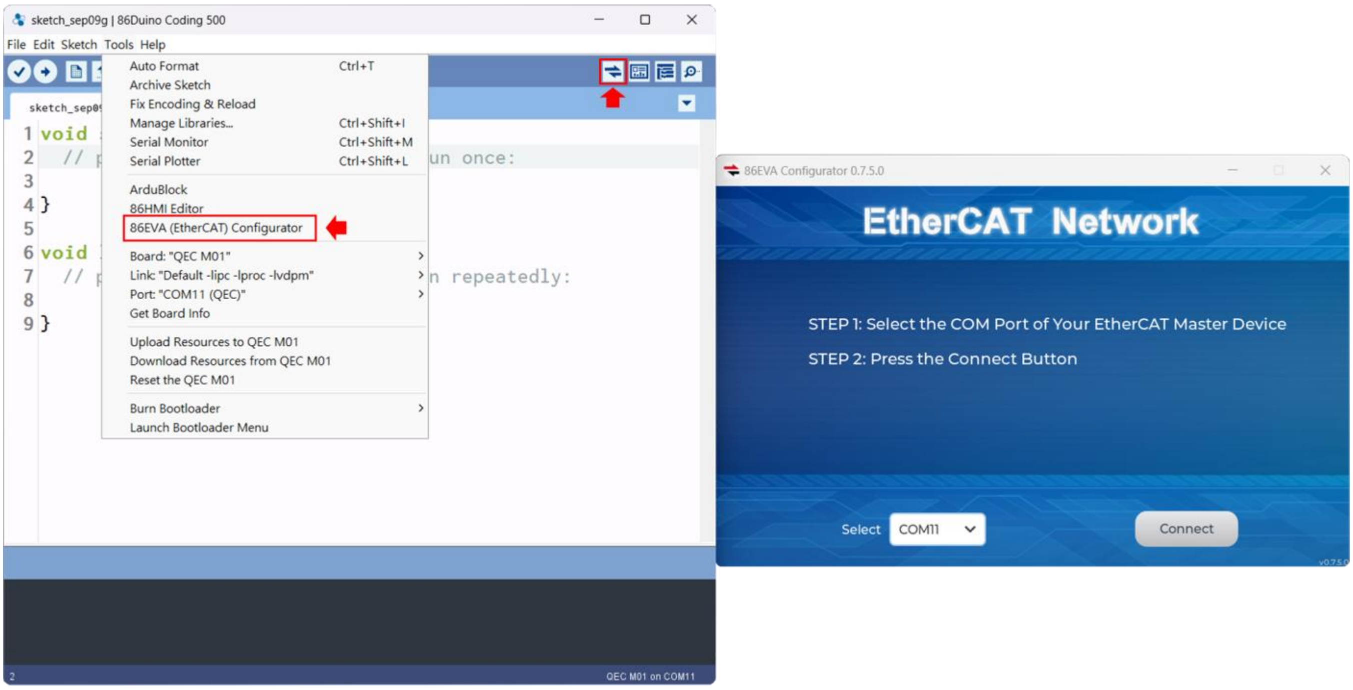
Software Tools Description:

- 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 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.

Step 1: Turn on 86EVA and scan

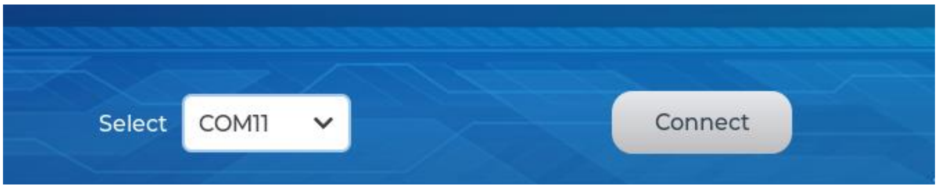
The 86EVA tool can be opened via the following buttons.


sketch_sep09g | 86Duino Coding 500
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
ArduBlock
86HMI Editor
86EVA (EtherCAT) Configurator
Board: "QEC M01"
Link: "Default -lipc -lproc -lvdpm"
Port: "COM11 (QEC)"
Get Board Info
Upload Resources to QEC M01
Download Resources from QEC M01
Reset the QEC M01
Burn Bootloader
Launch Bootloader Menu
QEC M01 on COM11
86EVA Configurator 0.7.5.0
EtherCAT Network
Select
COM11
Connect

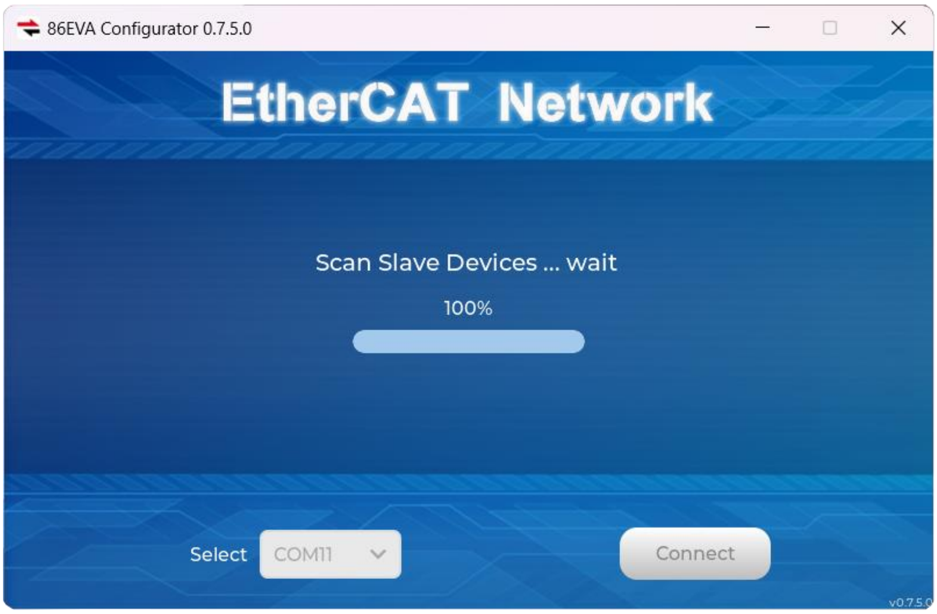
STEP 1: Select the COM Port of Your EtherCAT Master Device

STEP 2: Press the Connect Button

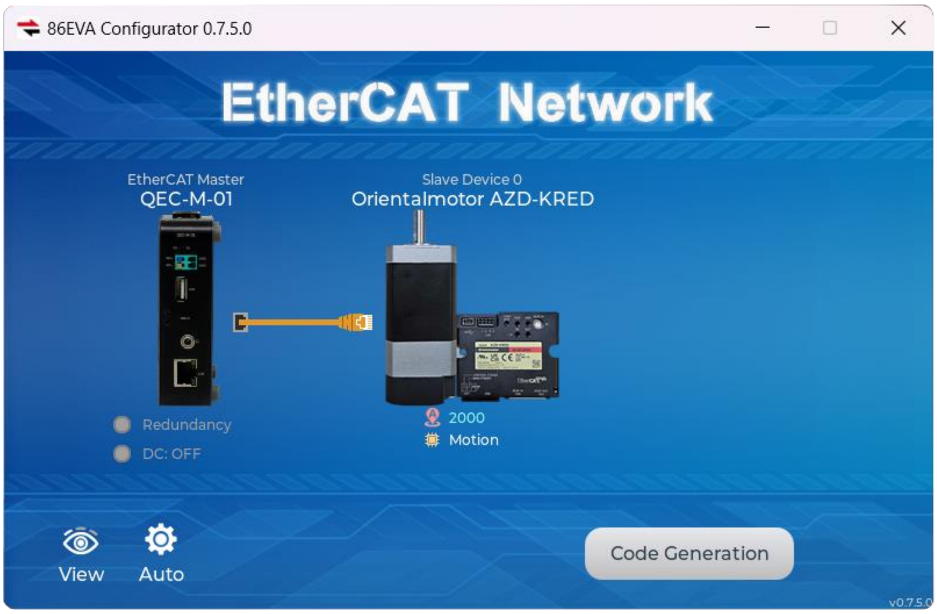
Please select the correct COM port and then click the “Connect” button.



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



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

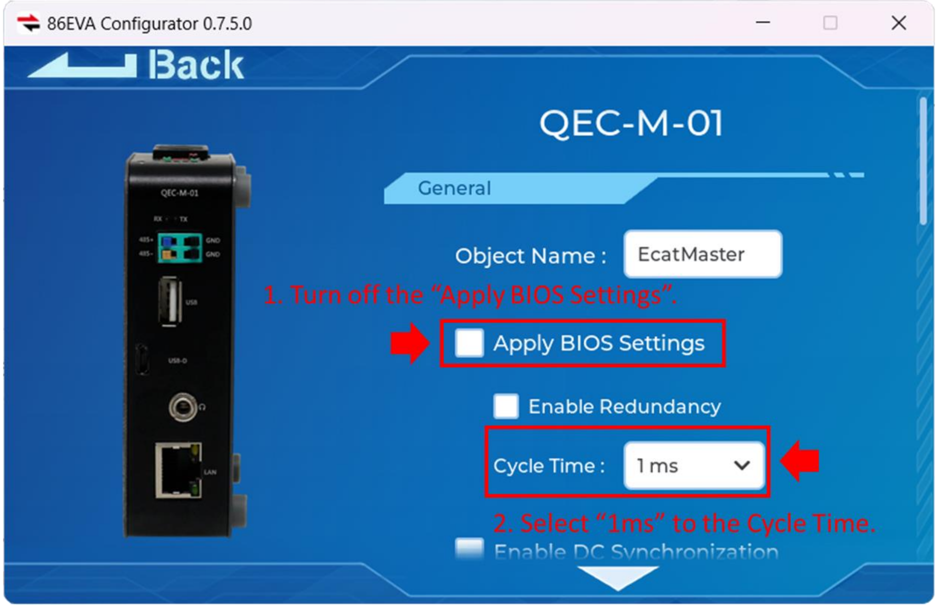


Step 2: Set the parameters

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.



86EVA Configurator 0.7.5.0

QEC-M-01

General

Object Name : EcatMaster

1. Turn off the "Apply BIOS Settings".

□ Apply BIOS Settings

□ Enable Redundancy

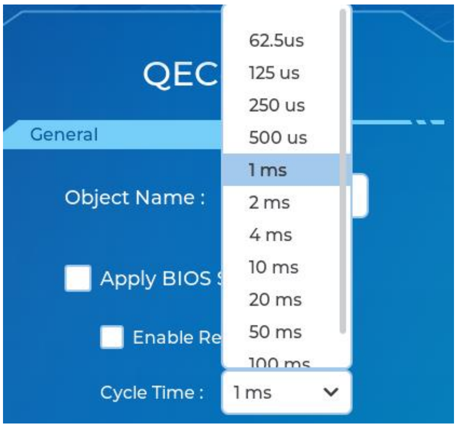
Cycle Time : 1 ms

2. Select "1ms" to the Cycle Time.

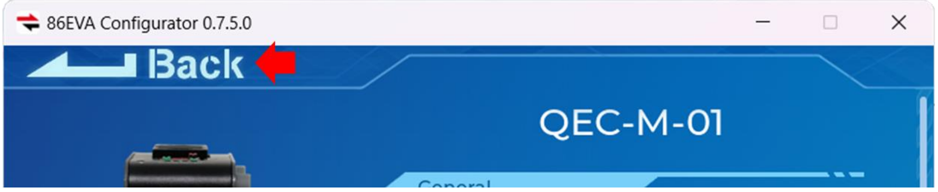
□ Enable DC Synchronization

Please check the following configures.

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

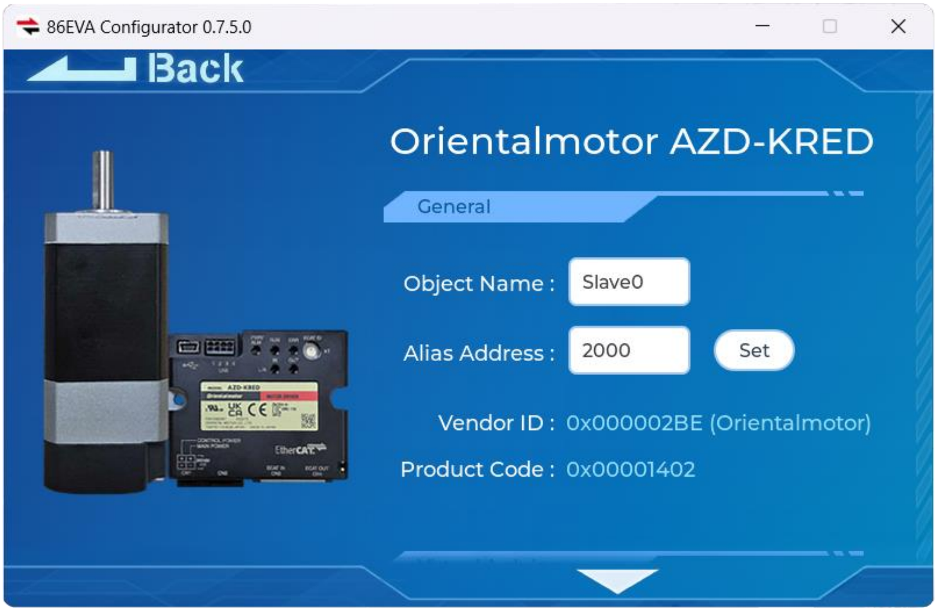


Click "Back" in the upper left corner to return.



AZD-KRED

Press twice on the image of the OrientalMotor AZD-KRED to see the parameter settings.



86EVA Configurator 0.7.5.0

Orientalmotor AZD-KRED

General

Object Name : Slave0

Alias Address : 2000

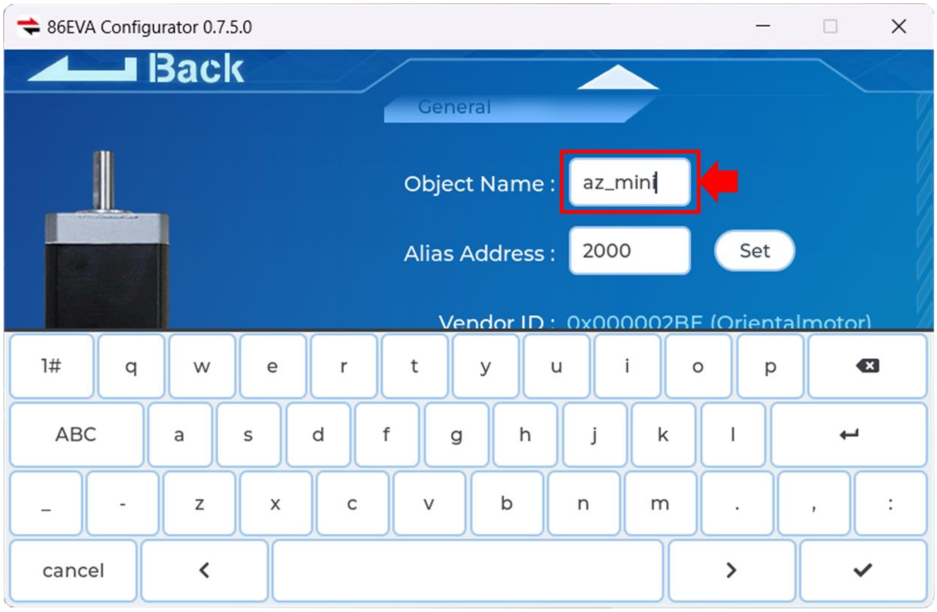
Vendor ID : 0x000002BE (Orientalmotor)

Product Code : 0x00001402

The page will show the Object Name, Alias Address, Vendor ID, and Product Code.

Please change the Object Name to “AZD-KRED”.

It’ll appear a keyboard after you click the Object Name.



86EVA Configurator 0.7.5.0

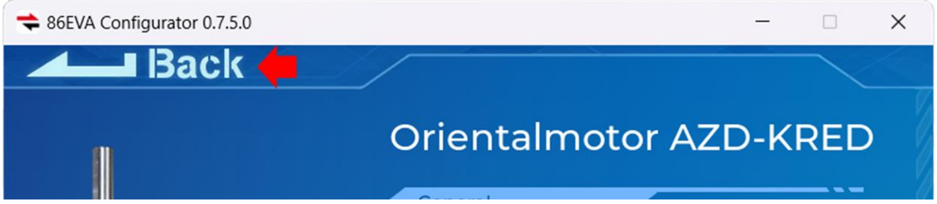
General

Object Name : az_mini

Alias Address : 2000

Vendor ID : 0x000002BE (Orientalmotor)

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

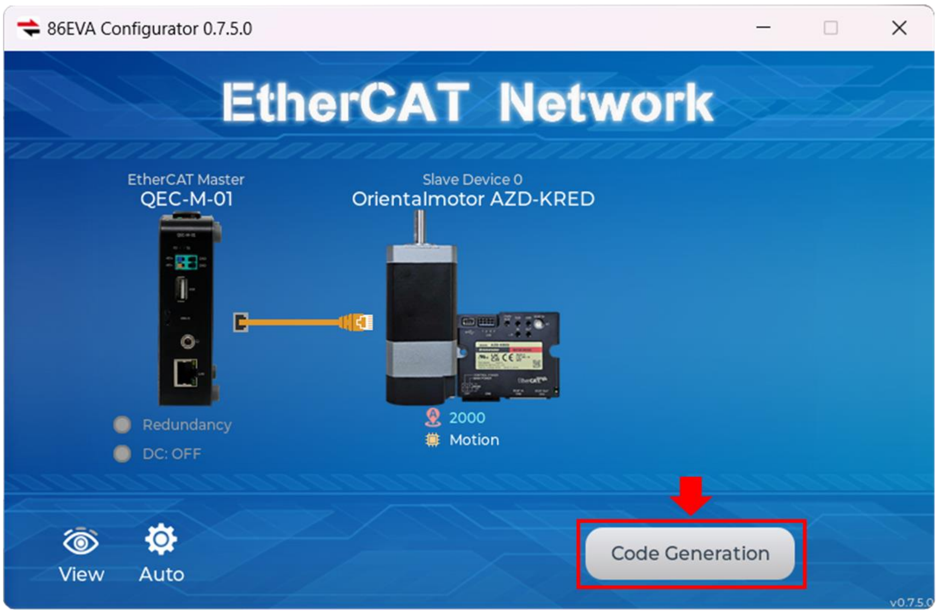


86EVA Configurator 0.7.5.0

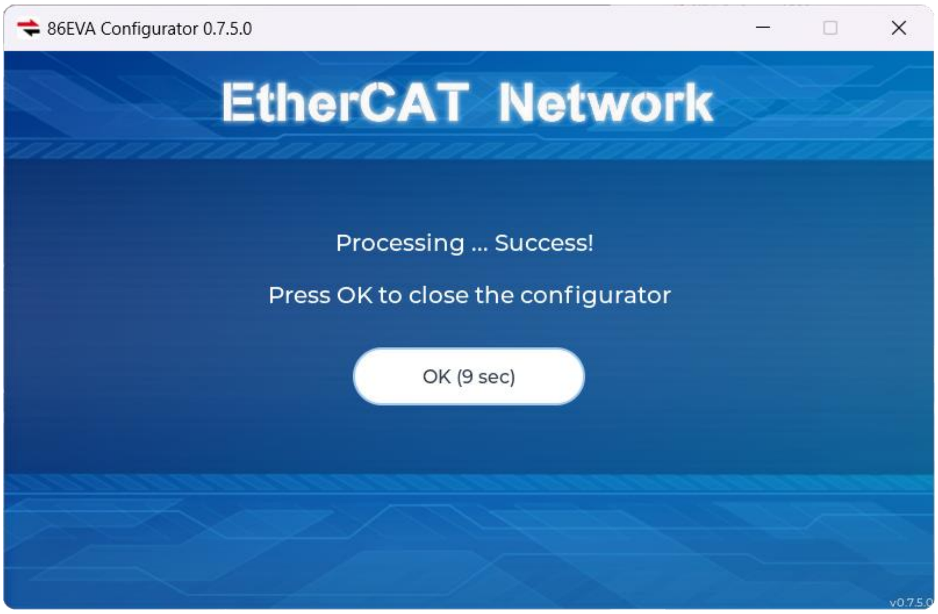
Orientalmotor AZD-KRED

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.

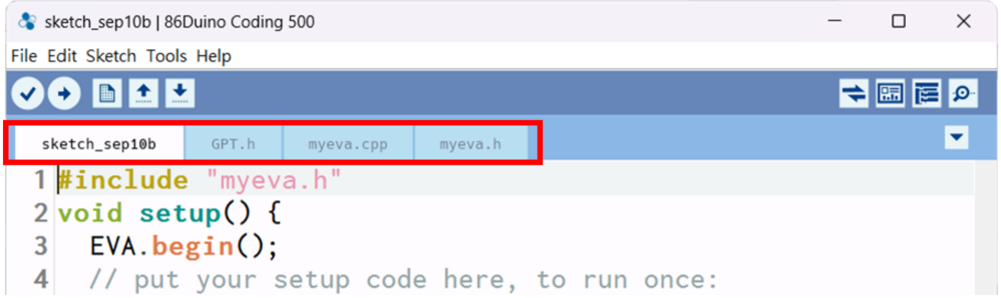


Processing ... Success!

Press OK to close the configurator

The generated code and files are as follows:

- sketch_sep10b: Main Project (.ino, depending on your project name)
- GPT.h: Parameters to provide to ChatGPT referred
- myeva.cpp: C++ program code of 86EVA
- myeva.h: Header file of 86EVA


sketch_sep10b | 86Duino Coding 500
File Edit Sketch Tools Help
sketch_sep10b
GPT.h
myeva.cpp
myeva.h

```
#include "myeva.h"
void setup() {
  EVA.begin();
  // put your setup code here, to run once:
```

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:

- OrientalMotor AZ-KRED: `EthercatDevice_CiA402` object.
- CiA402 mode: Profile Position (PP) mode.
- EtherCAT mode: ECAT_SYNC.
- Distributed Clock: Open. Follow the cycle time.

And here is the setting by users:

- EtherCAT Cycle time: 1 millisecond.
- Device Object Name: QEC-M-01 is “EcatMaster”, and AZD-KRED is “az-mini”.
-

1. **In Setup Function:**
   - Initializes the serial (115200).
   - Register Cyclic Callback Function, which named “myCallback”, in `EVA.begin()` function.
   - Enable the AZ-mini. Change the CiA402 state to CIA402_OPERATION_ENABLED.
     Use `delay(1000)` to wait for it to change 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). 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). 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.

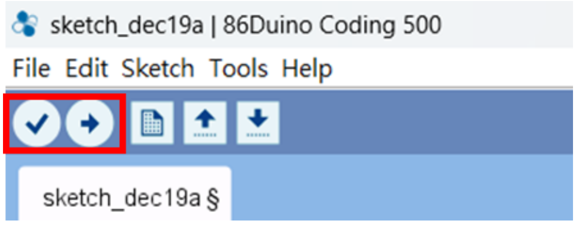
Each state introduces a delay of 100 milliseconds to ensure smooth transitions.

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.

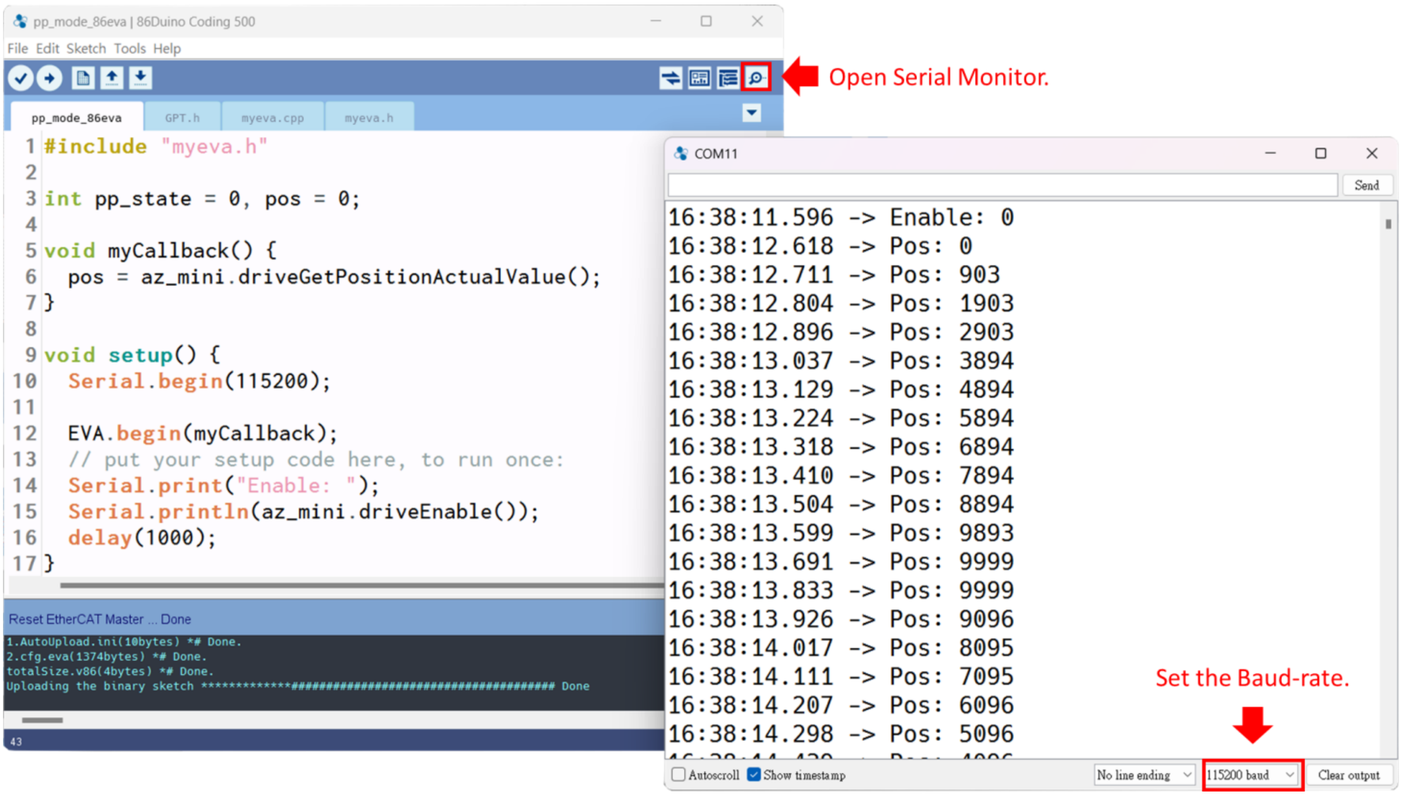
Here is the code:

```
#include "myeva.h"

int pp_state = 0, pos = 0;

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

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

  EVA.begin(myCallback);
  // put your setup code here, to run once:
  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.


sketch_dec19a | 86Duino Coding 500
File Edit Sketch Tools Help
sketch_dec19a §

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.


pp_mode_86eva | 86Duino Coding 500
File Edit Sketch Tools Help
pp_mode_86eva
GPT.h
myeva.cpp
myeva.h
COM11
Send
43

```
#include "myeva.h"

int pp_state = 0, pos = 0;

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

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

  EVA.begin(myCallback);
  // put your setup code here, to run once:
  Serial.print("Enable: ");
  Serial.println(az_mini.driveEnable());
  delay(1000);
}
```

```
Reset EtherCAT Master ... Done
1.AutoUpload.ini(10bytes) *# Done.
2.cfg.eva(1374bytes) *# Done.
totalSize.v86(4bytes) *# Done.
Uploading the binary sketch ************#################################### Done
```

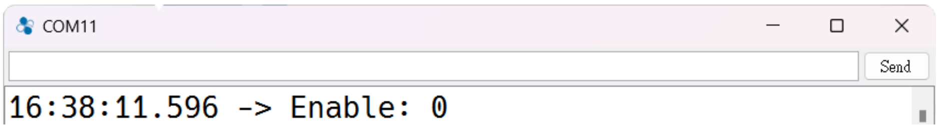
```
16:38:11.596 -> Enable: 0
16:38:12.618 -> Pos: 0
16:38:12.711 -> Pos: 903
16:38:12.804 -> Pos: 1903
16:38:12.896 -> Pos: 2903
16:38:13.037 -> Pos: 3894
16:38:13.129 -> Pos: 4894
16:38:13.224 -> Pos: 5894
16:38:13.318 -> Pos: 6894
16:38:13.410 -> Pos: 7894
16:38:13.504 -> Pos: 8894
16:38:13.599 -> Pos: 9893
16:38:13.691 -> Pos: 9999
16:38:13.833 -> Pos: 9999
16:38:13.926 -> Pos: 9096
16:38:14.017 -> Pos: 8095
16:38:14.111 -> Pos: 7095
16:38:14.207 -> Pos: 6096
16:38:14.298 -> Pos: 5096
```

☐ Autoscroll ☑ Show timestamp   No line ending   115200 baud   Clear output

Open Serial Monitor.

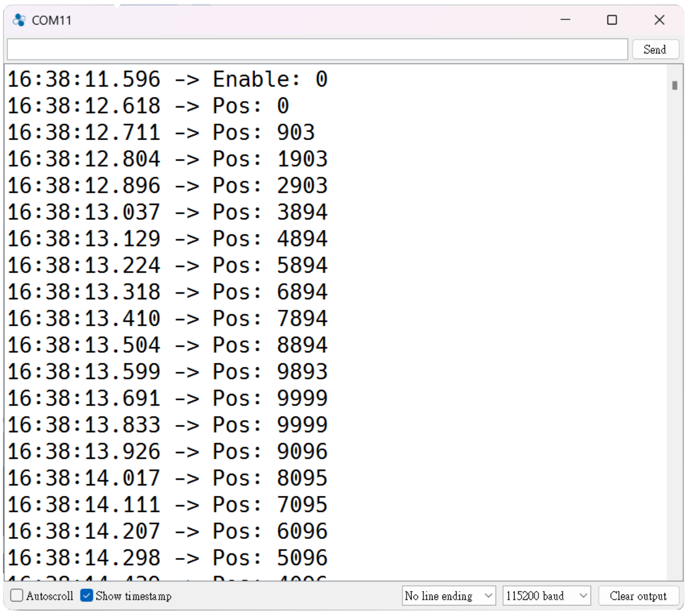
Set the Baud-rate.

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



```
16:38:11.596 -> Enable: 0
```

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



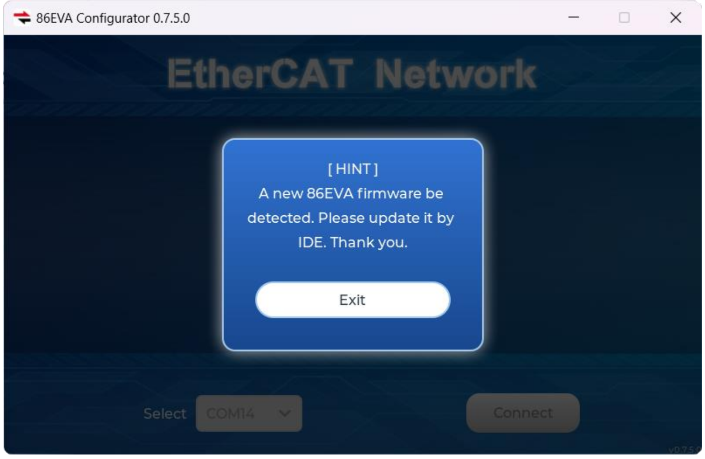
```
16:38:11.596 -> Enable: 0
16:38:12.618 -> Pos: 0
16:38:12.711 -> Pos: 903
16:38:12.804 -> Pos: 1903
16:38:12.896 -> Pos: 2903
16:38:13.037 -> Pos: 3894
16:38:13.129 -> Pos: 4894
16:38:13.224 -> Pos: 5894
16:38:13.318 -> Pos: 6894
16:38:13.410 -> Pos: 7894
16:38:13.504 -> Pos: 8894
16:38:13.599 -> Pos: 9893
16:38:13.691 -> Pos: 9999
16:38:13.833 -> Pos: 9999
16:38:13.926 -> Pos: 9096
16:38:14.017 -> Pos: 8095
16:38:14.111 -> Pos: 7095
16:38:14.207 -> Pos: 6096
16:38:14.298 -> Pos: 5096
```

☐ Autoscroll ☑ Show timestamp   No line ending   115200 baud   Clear output

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.



[HINT]
A new 86EVA firmware be detected. Please update it by IDE. Thank you.

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

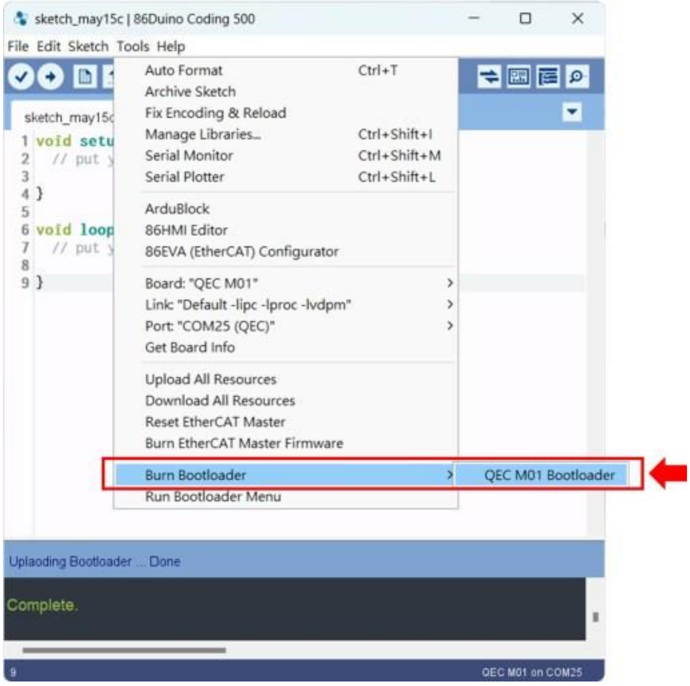
Step 1: Setting up QEC-M

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

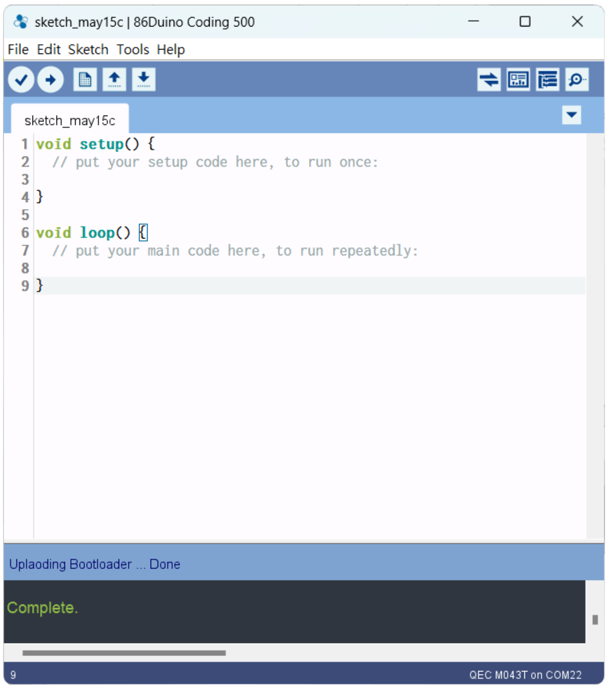
After connecting to your QEC-M product, go to “Tools”> “Burn Bootloader”. The currently selected QEC-M name will appear. Clicking on it will start the update process, which will take approximately 5-20 minutes.

QEC-M-01:



```
Uplaoding Bootloader ... Done
Complete.
```

Step 3: Complete the Update


sketch_may15c | 86Duino Coding 500
File Edit Sketch Tools Help
sketch_may15c
9
QEC M043T on COM22

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
Uplaoding Bootloader ... Done
Complete.
```

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.