Start Guide

Delta AC Servo Drive & Motor ASDA-A2/B3 Series (PP mode) with 86EVA

86Duino Coding IDE 500

EtherCAT Library

(Version 1.0)

Revision

Date	Version	Description
2024/11/05	VERSION1.0	NEW RELEASE.

Preface

In this guide, we will demonstrate how to use QEC-M-01 (EtherCAT master) to control Delta's ASDA-A2 and ASDA-B3 servo motors, taking Profile Position (PP) mode as an example.

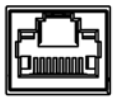
Notes QEC's PoE (Power over Ethernet)

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:

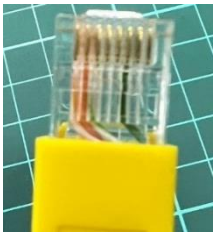


Pin #	Signal Name	Pin #	Signal Name
1	LAN1_TX+	2	LAN1_TX-
3	LAN1_RX+	4	VS+
5	VP+	6	LAN1_RX-
7	VS-(GND)	8	VP-(GND)

* 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).

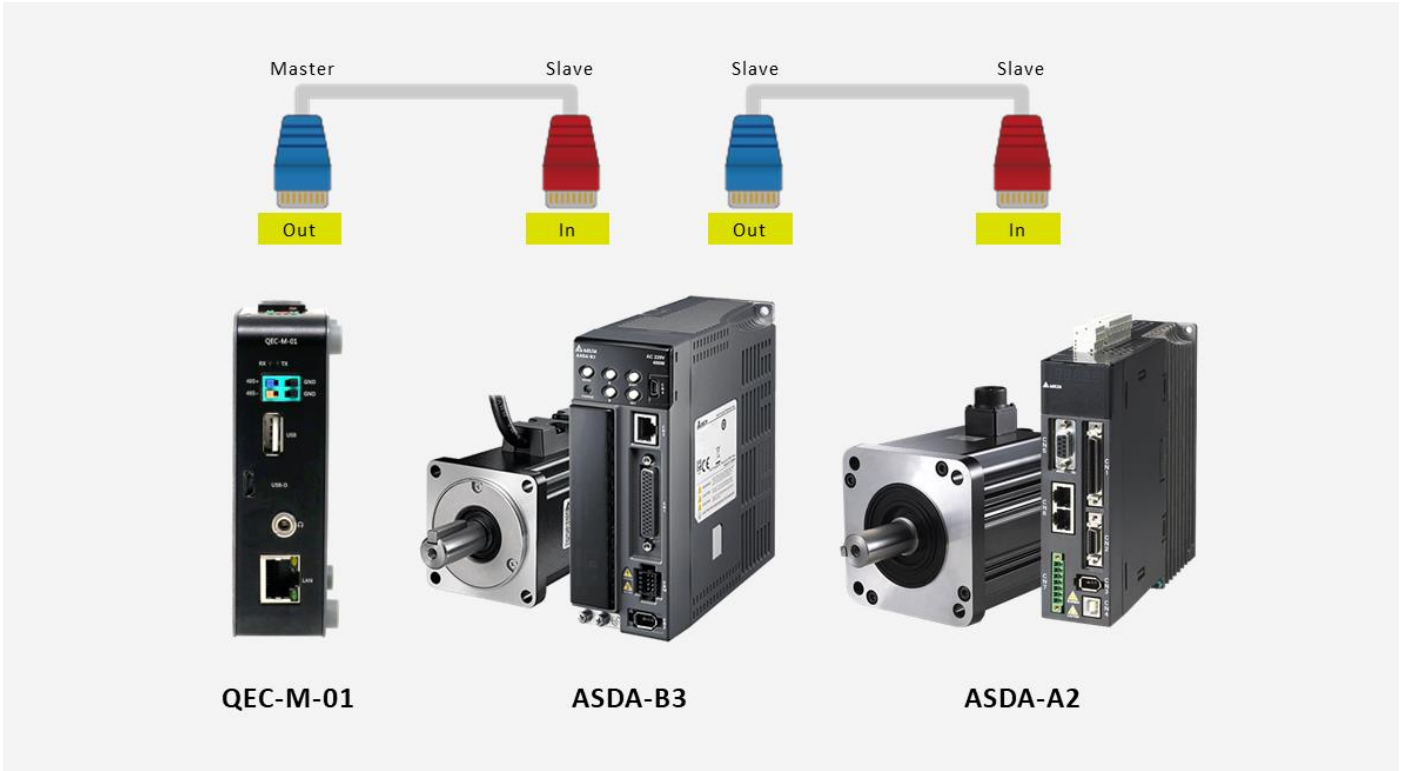


3. QEC's PoE power supply is up to 24V/3A.

1. Connection and wiring hardware

The following devices are used here:

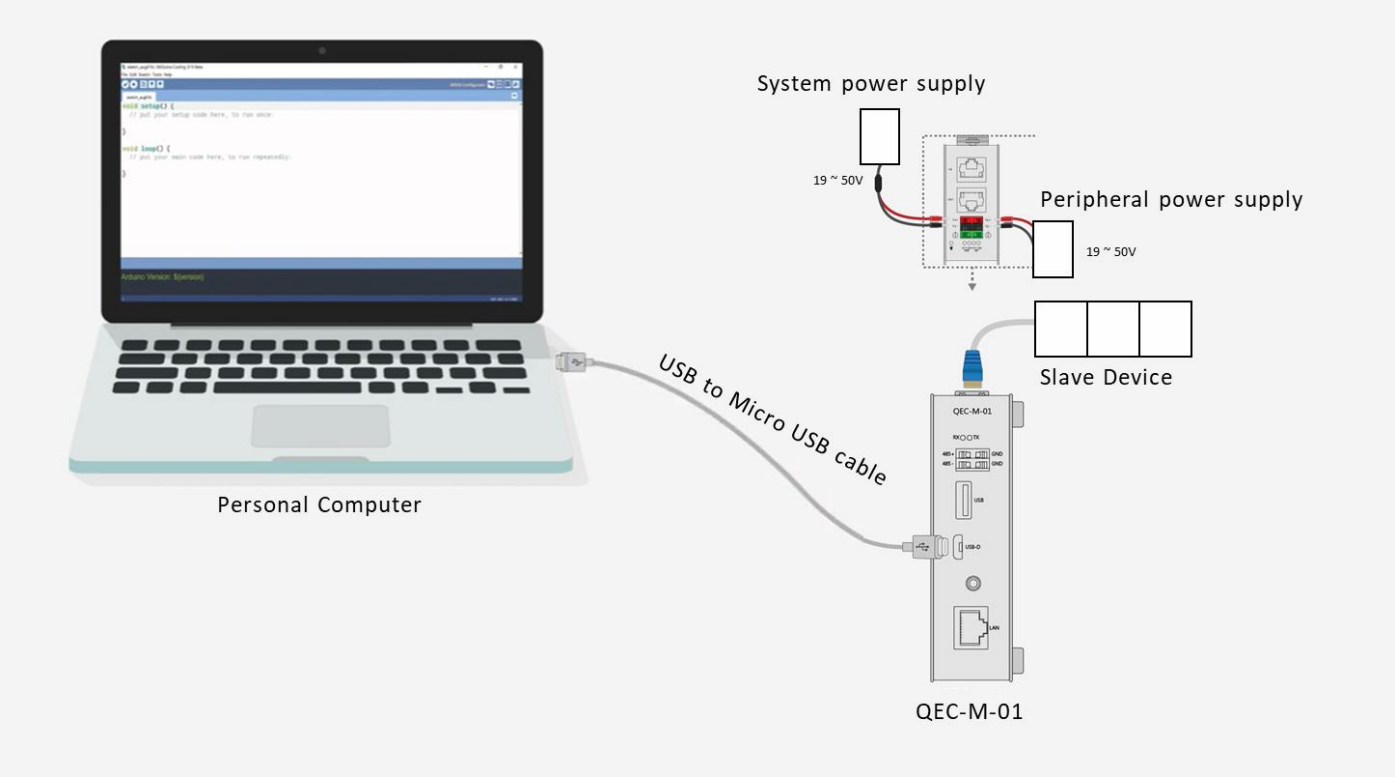
1. QEC-M-01（EtherCAT Master）
2. Delta AC Servo: ASDA-A2-E (EtherCAT)
3. Delta AC Servo: ASDA-B3 (EtherCAT)
4. 24V power supply & EU-type terminal cable
5. RJ45 Cables



1.1 QEC-M-01

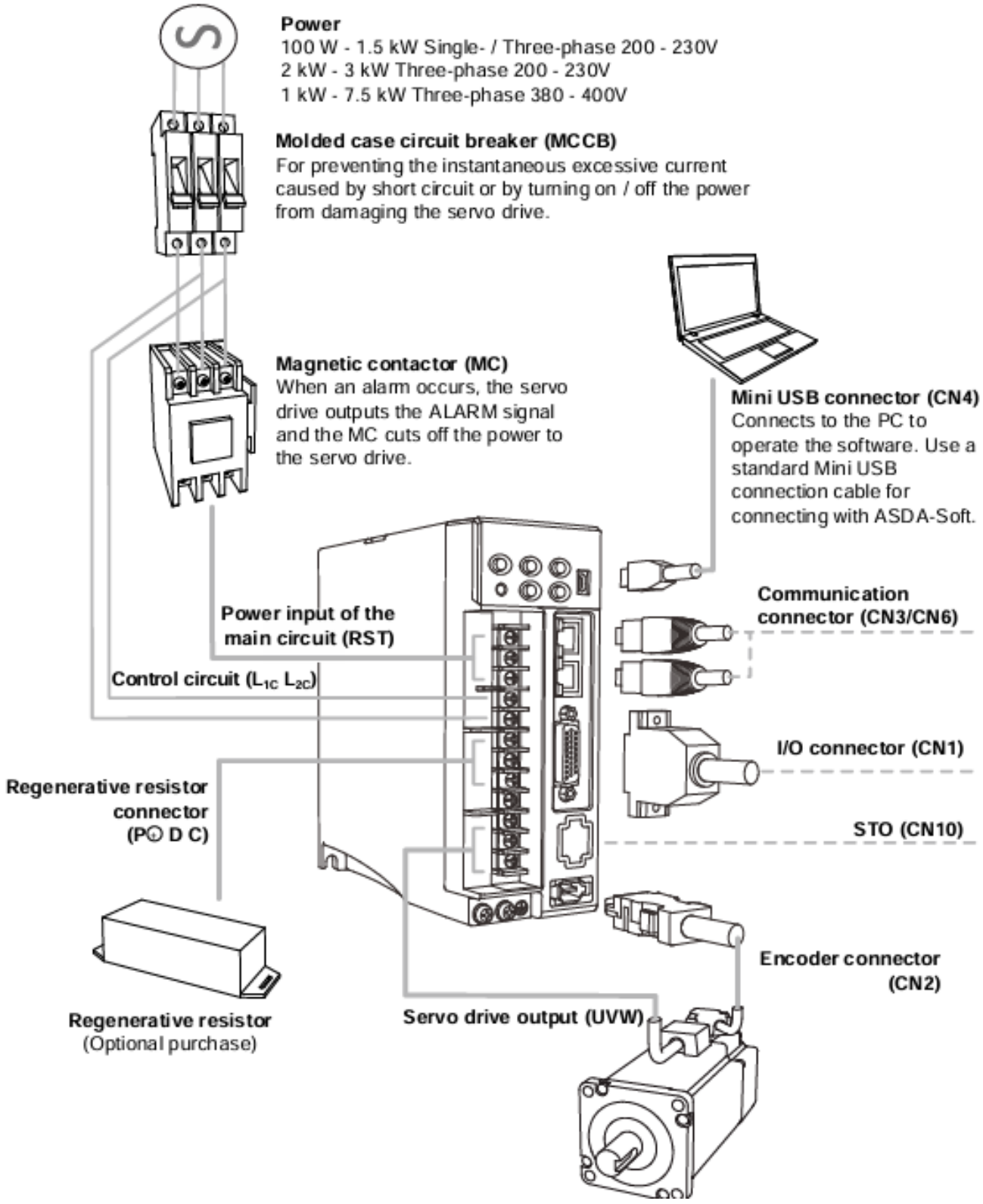
QEC EtherCAT Master.

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 the Delta EtherCAT Servos via RJ45 cable.



1.2 Delta EtherCAT AC Servo

To connect EtherCAT ports on Delta AC Servo, usually should be on CN6 ports.

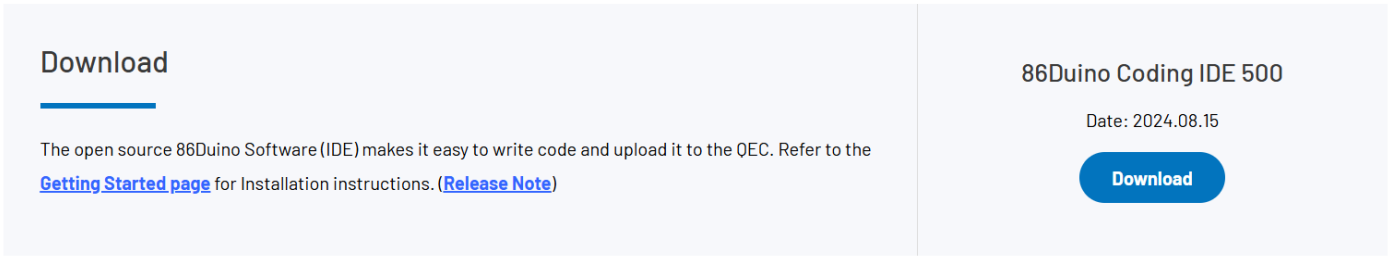


(Figure 1: Connecting to Delta communication type servo motor. From DELTA_IA_ASD_ASDA_B3 Operation User Manual)

ASDA-A2 and ASDA-B3: Provide a 220V power supply, for more details please refer to the Delta Website.

2. Software/Development Environment

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



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.

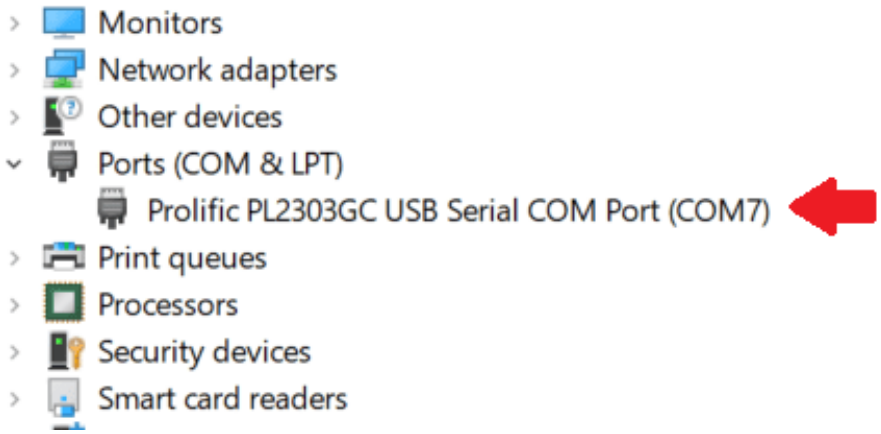
Note:
If Windows displays a warning, click Details once and then click the Continue Run button once.

86Duino Coding IDE 500+ looks like below.

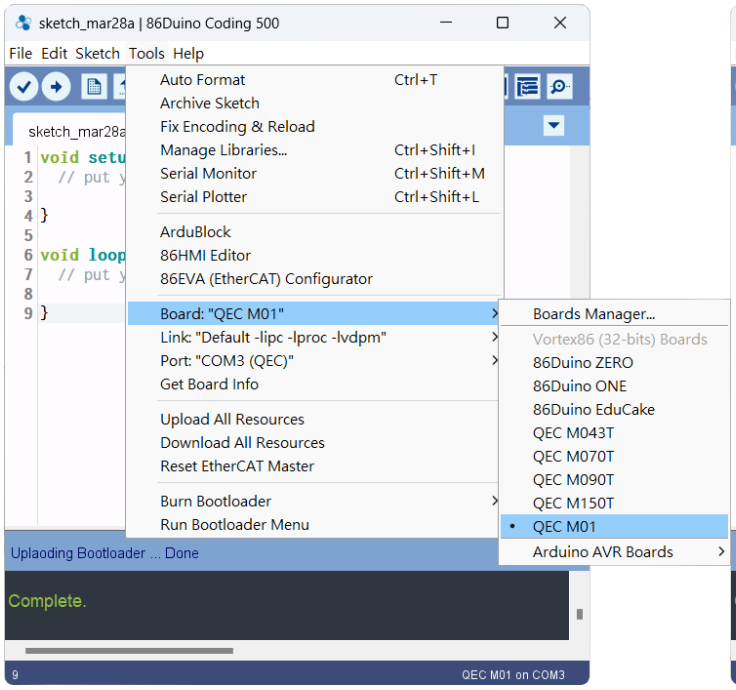
3. Connect to PC and set up the environment

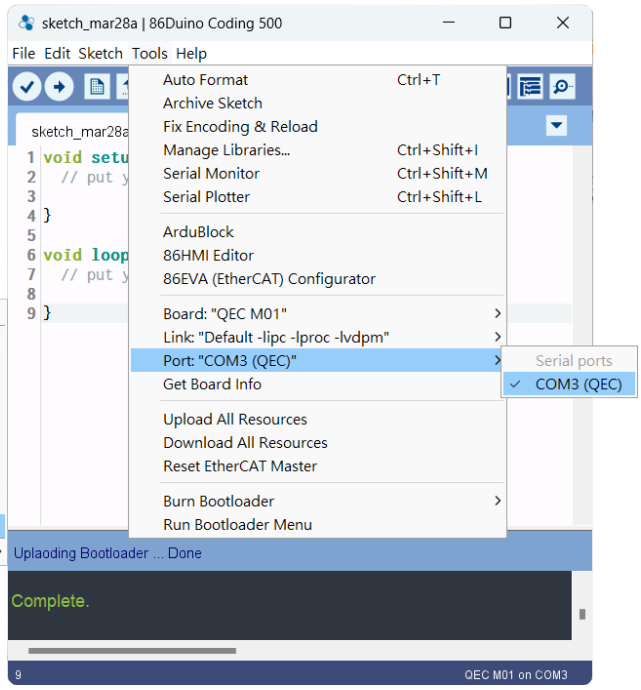
Follow the steps below to set up the environment:

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-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. Use 86EVA with code

This example shows how to operate the EtherCAT master (QEC-M-01) and the Delta EtherCAT AC-Servo Series: ASDA-A2-E and ASDA-B3 through the 86Duino IDE's graphical low-code programming tool, 86EVA.

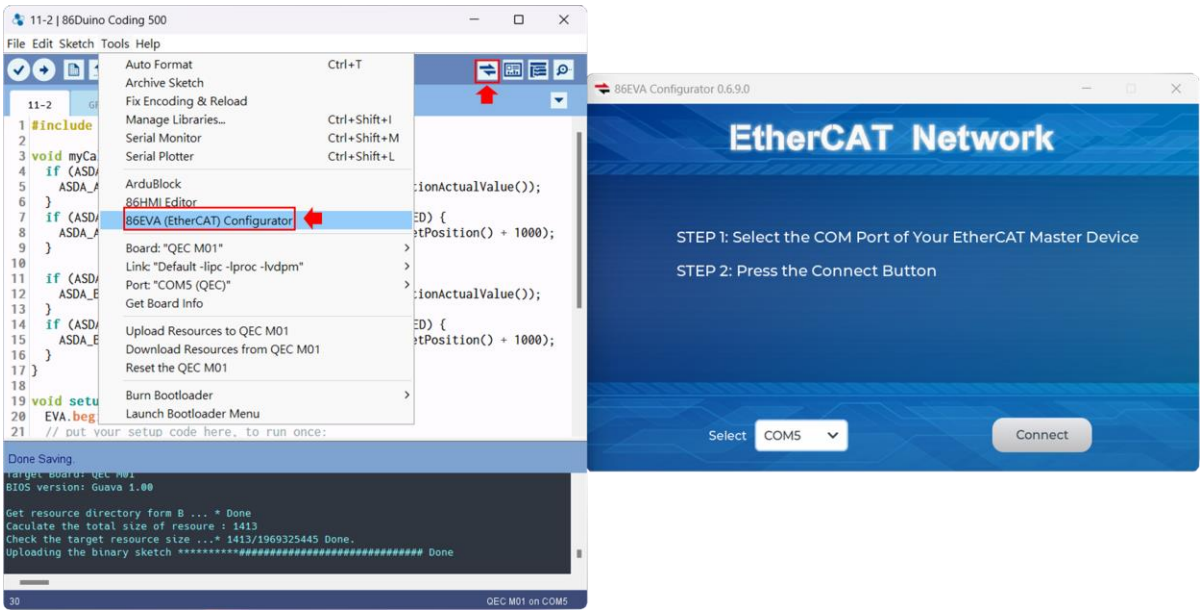
Software Tools Description:

- 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 program sets the Delta EtherCAT AC-Servo Series: ASDA-A2-E and ASDA-B3 to CiA402 Profile Position (CSP) mode. It continuously reads the current position of the motors and controls the motor using the Profile Position API function, which can set the position, velocity, and acceleration parameters.

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.

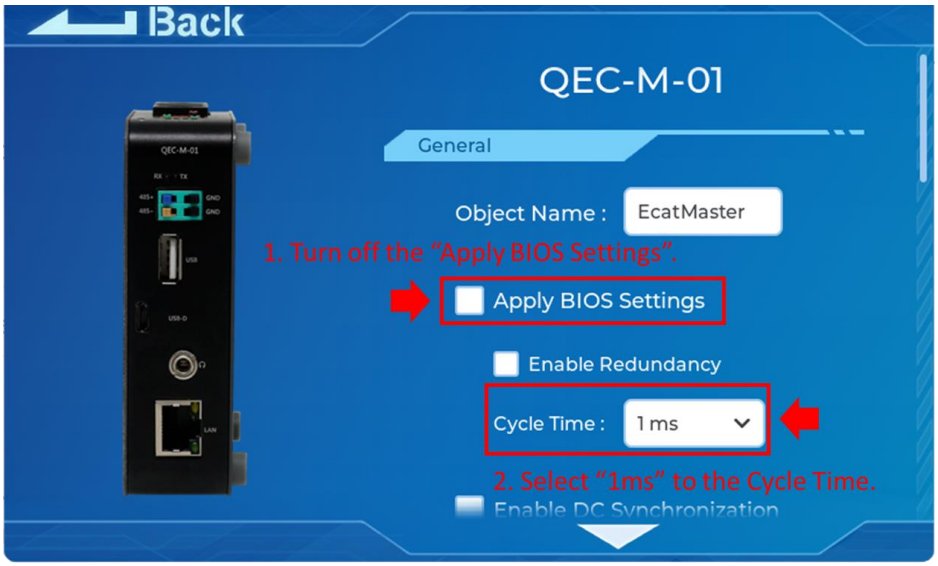
The connected devices will be displayed after the EtherCAT network has been scanned.

Step 2: Set the parameters

Press twice on the scanned device image to enter the corresponding parameter setting screen.

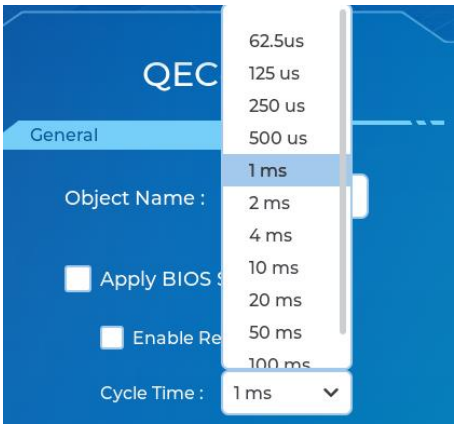
QEC-M-01

Press twice on the image of the QEC-M-01 to see the parameter settings.

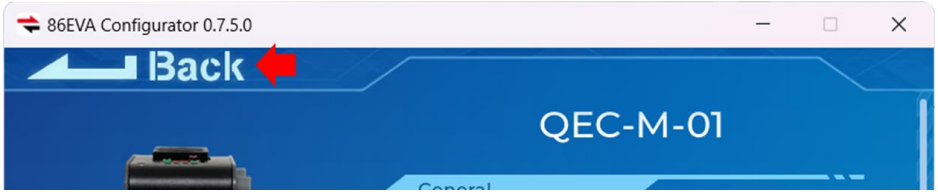


Please check the following configures.

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



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



Delta ASDA-B3

Double-click on the image of Delta ASDA-B3 to view the parameter settings.
This will display the Delta ASDA-B3's Object Name, Alias Address, Vendor ID, and Product Code.

Among these, we change the Object Name to "ASDA_B3" to facilitate identification in subsequent code.

After completion, click "Back" in the upper left corner to return.

Delta ASDA-A2

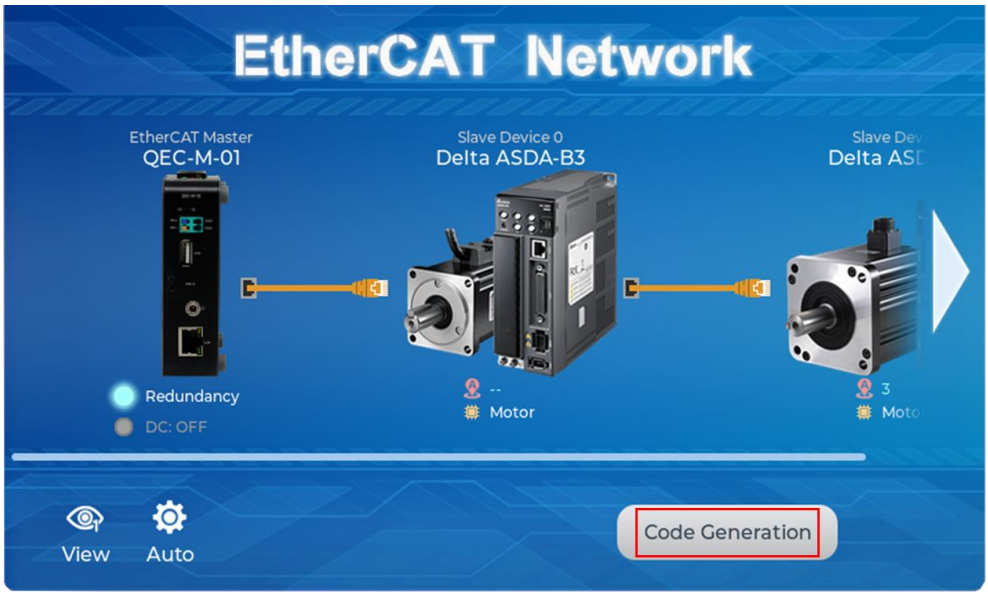
Double-click on the image of Delta ASDA-A2 to view the parameter settings.
This will display the Delta ASDA-A2's Object Name, Alias Address, Vendor ID, and Product Code.

Among these, we change the Object Name to "ASDA_A2" to facilitate identification in subsequent code.

After completion, click "Back" in the upper left corner to return.

Step 3: Generate the code

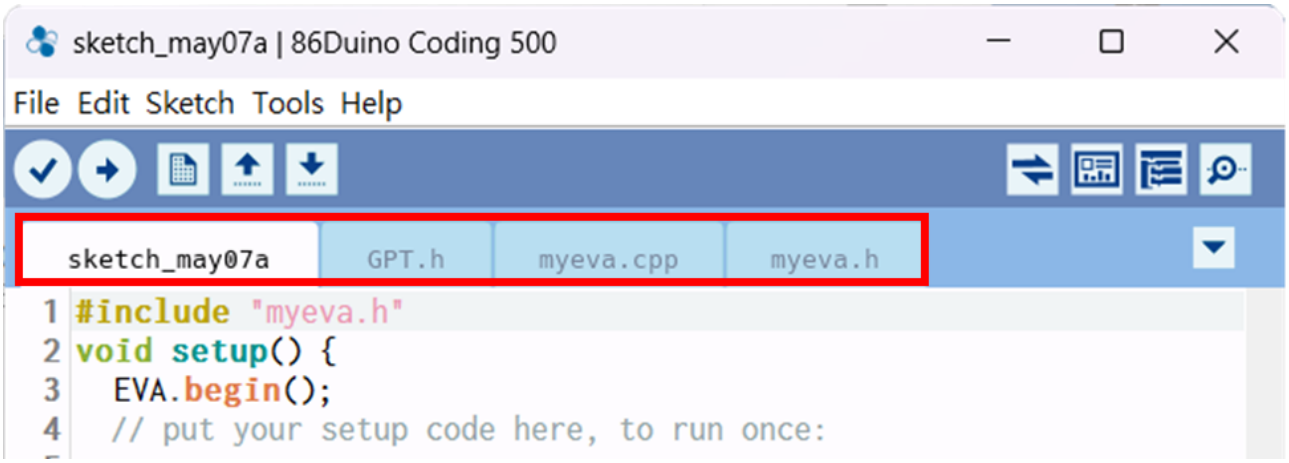
Once you've set your device's parameters, go back to the home screen and press the "Code Generation" button in the bottom right corner.



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_may07a: Main Project (.ino, depending on your project name)
- ChatGPT.h: Parameters to provide to ChatGPT referred
- 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

Note:

Delta's ASDA_A2 and ASDA_B3 servo motors will directly generate EthercatDevice_CiA402 objects.

Step 4: Write the code

The programming code from 86EVA are set as the following by default:

- Delta ASDA-A2-E and ASDA-B3: EthercatDevice_CiA402 object.
- 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”, ASDA-A2-E is “ASDA_A2”, and ASDA-B3 is “ASDA_B3”.

1. **In Setup Function:**
   - Initializes the serial (115200).
   - Register Cyclic Callback Function, which named “myCallback”, in EVA.begin() function.
   - Configure the ASDA_A2 and the ASDA_B3 in Profile Position (CSP) mode.
     Use delay(100) to wait for it to change successfully.
   - Enable the ASDA_A2 and the ASDA_B3, we change the CiA402 state to CIA402_OPERATION_ENABLED.
     Use delay(1000) to wait for it to change successfully.
2. **In Callback Function:**
   The motors’ positions are updated cyclically; we read the actual position value of each Delta Servos.
3. **In Loop Function:**
   The main loop prints the motors’ 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 and moves to a target position (Position: 10,000,000, velocity: 1,000,000, and acceleration: 1,000,000). Once the move begins, transition to the next state.
   - State 1: Waits for the motor to reach the target position. Once the target is reached, proceed to the next state.
   - State 2: Ends the current profile position movement.
   - State 3: Starts a profile position move back to the original position (Position: 0, velocity: 1,000,000, and acceleration: 1,000,000). When the move starts, transition to the next state.
   - State 4: Waits for the motor to reach the target position. Once the target is reached, proceed to the next state.
   - State 5: Ends the current profile position movement, and reset the state machine back to State 0 to repeat the cycle.
     Also, set the pp_done plus 1 to know whether both ASDA servos finish the actions.

   We use delay(100) to wait for the program to be processed successfully.

In summary, this code establishes EtherCAT communication, reads the ASDA_A2 and ASDA_B3 positions, and controls the motor using the Profile Position API function, which can set the position, velocity, and acceleration parameters.

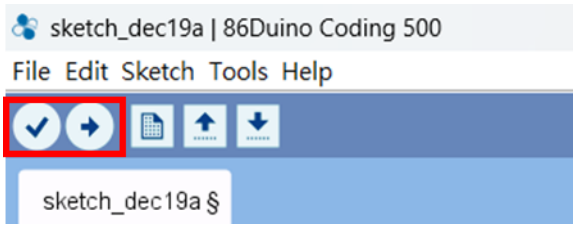
Here is the code:

```
#include "myeva.h"

int A2_pos = 0, B3_pos = 0, pp_state = 0;

void myCallback() {
  A2_pos = ASDA_A2.driveGetPositionActualValue();
  B3_pos = ASDA_B3.driveGetPositionActualValue();
}

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

  EVA.begin(myCallback);
  ASDA_A2.driveSetMode(CIA402_PP_MODE);
  ASDA_B3.driveSetMode(CIA402_PP_MODE);
  delay(100);

  ASDA_A2.driveEnable();
  ASDA_B3.driveEnable();
  delay(1000);
}

void loop() {
  Serial.print("A2 Pos: "); Serial.print(A2_pos);
  Serial.print("B3 Pos: "); Serial.println(B3_pos);

  switch (pp_state)
  {
    case 0:
      if (ASDA_A2.profilePositionBegin(10000000, 1000000, 1000000) == 0 &&
ASDA_B3.profilePositionBegin(10000000, 1000000, 1000000) == 0 )  // Start profile
position move
        pp_state++;
      break;
    case 1:
```

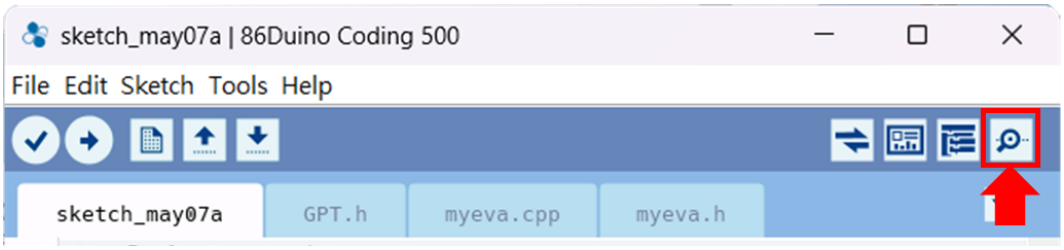
```
      if (ASDA_A2.driveIsTargetReached() && ASDA_B3.driveIsTargetReached())  //
Check if target position is reached
        pp_state++;
      break;
    case 2:
      ASDA_A2.profilePositionEnd();  // End profile move
      ASDA_B3.profilePositionEnd();
      pp_state++;
      break;
    case 3:
      if (ASDA_A2.profilePositionBegin(0, 1000000, 1000000) == 0 &&
ASDA_B3.profilePositionBegin(0, 1000000, 1000000) == 0 )
        pp_state++;
      break;
    case 4:
      if (ASDA_A2.driveIsTargetReached() && ASDA_B3.driveIsTargetReached())
        pp_state++;
      break;
    case 5:
      ASDA_A2.profilePositionEnd();  // End profile move
      ASDA_B3.profilePositionEnd();
      pp_state = 0;
      break;
  }
  delay(100);
}
```

Note:

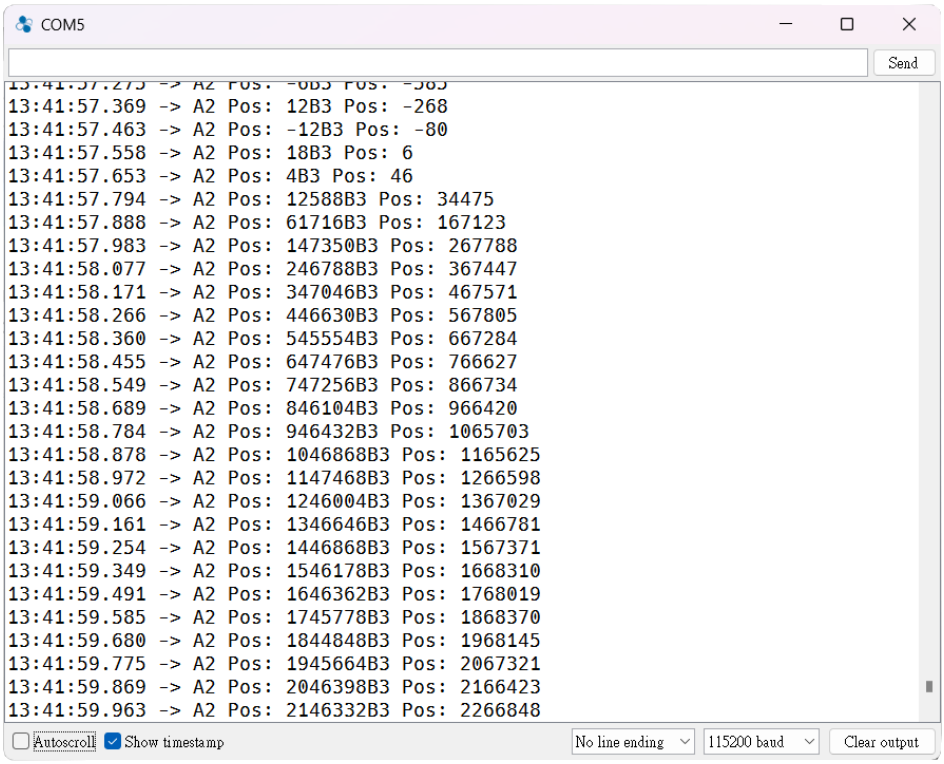
Once the code is written, click on the toolbar to compile, and to confirm that the compilation is complete and error-free, you can click to upload.



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.



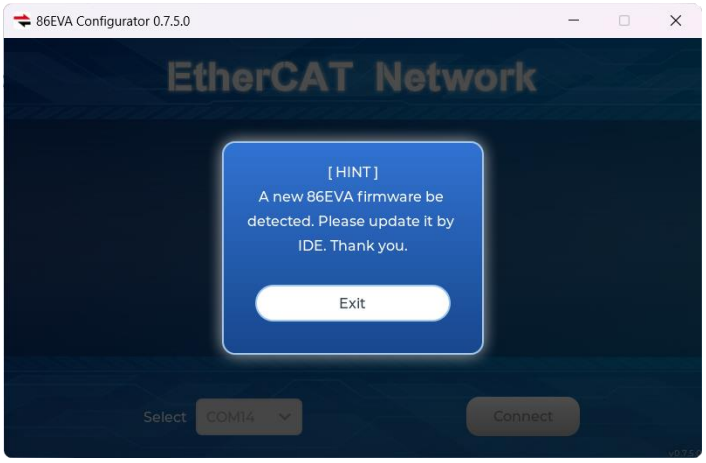
If the EtherCAT communication config successful, Serial Monitor will print “A2 Pos: “ and “B3 Pos: “, which means ASDA-A2 and ASDA-B3 positions.



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.



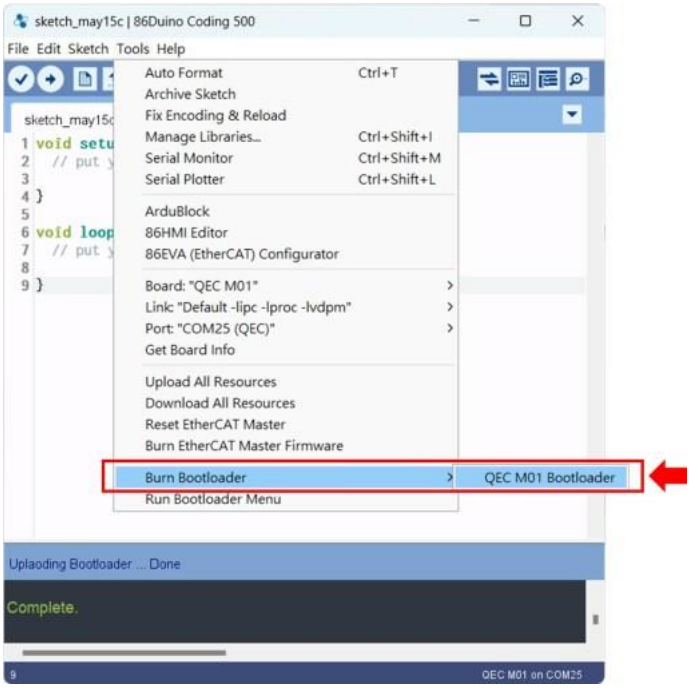
Now, we will further explain how to proceed with the update:

Step 1: Setting up QEC-M

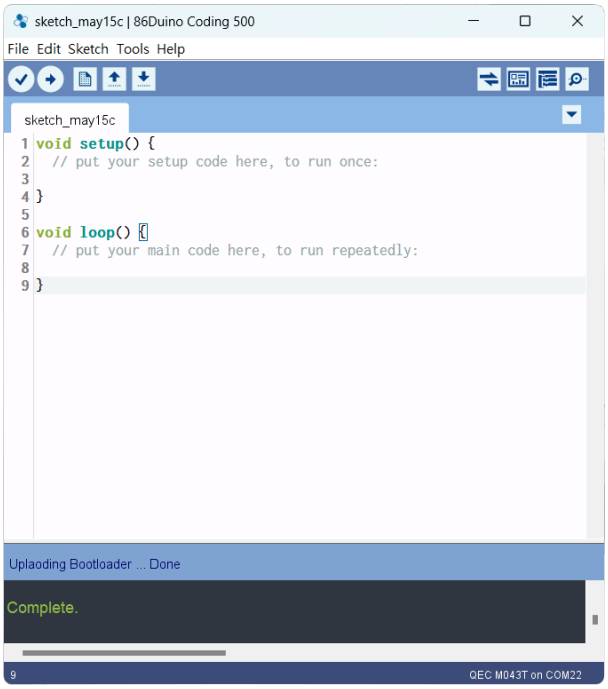
1. Download and install 86Duino IDE 500 (or a newer version): You can download it from Software.
2. Connect the QEC-M: Use a USB cable to connect the QEC-M to your computer.
3. Open 86Duino IDE: After the installation is complete, open the 86Duino IDE software.
4. Select Board: From the IDE menu, choose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