

Start Guide

Delta AC Servo Motors and Drives
ASDA-B3/ASDA-A2 Series (PP mode)
With 86EVA and ArduBlock



86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

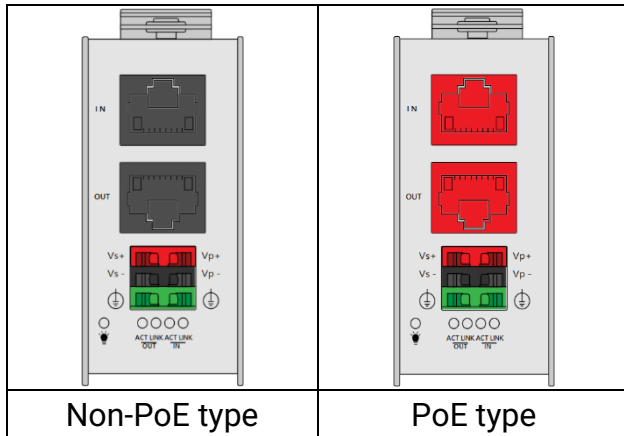
Date	Version	Description
2026/4/10	Version1.0	New release.

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.

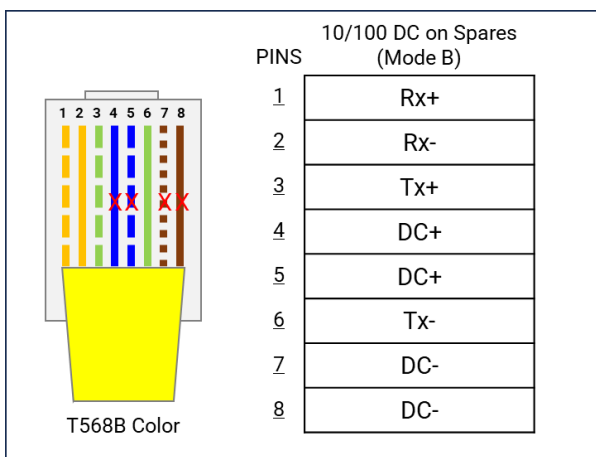
Notes QEC's PoE (Power over Ethernet)

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.



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:



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

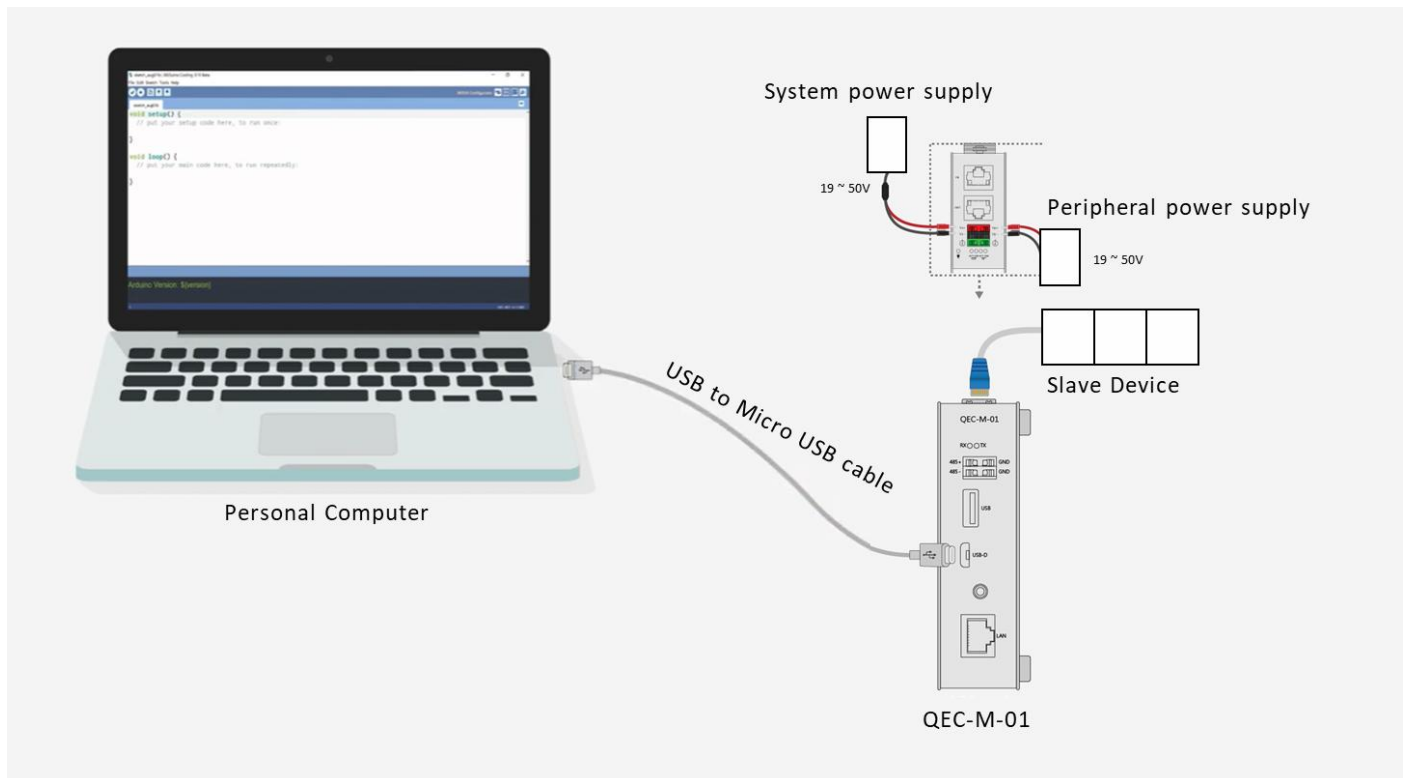
QEC EtherCAT MDevice.

1. Power Supply:

Connect to Vs+/Vs- and Vp+/Vp- power supplies via EU terminals for 24V power.

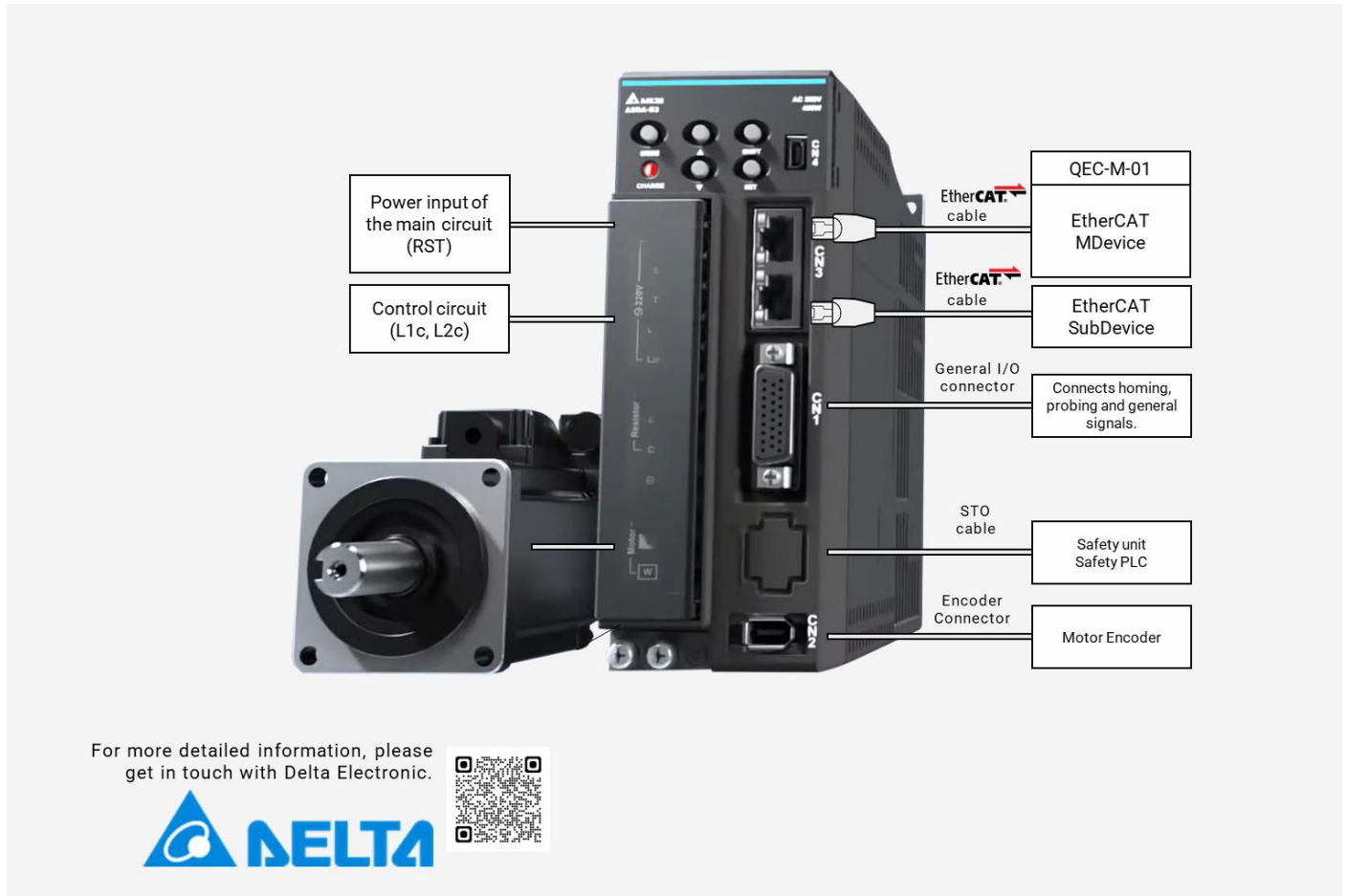
2. EtherCAT Connection:

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.



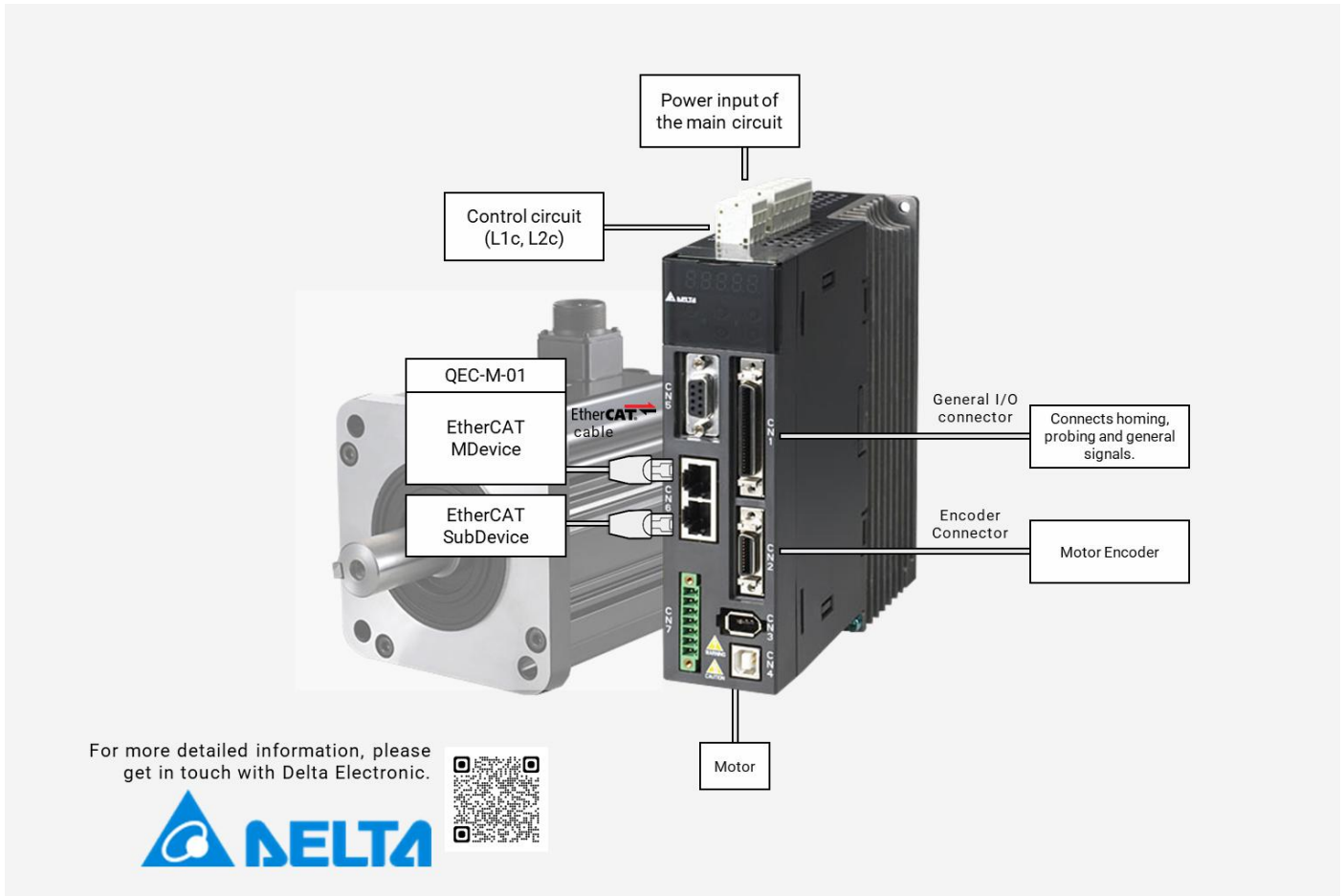
The connections are grouped by function:

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 ($P \oplus D C$)
 - 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.



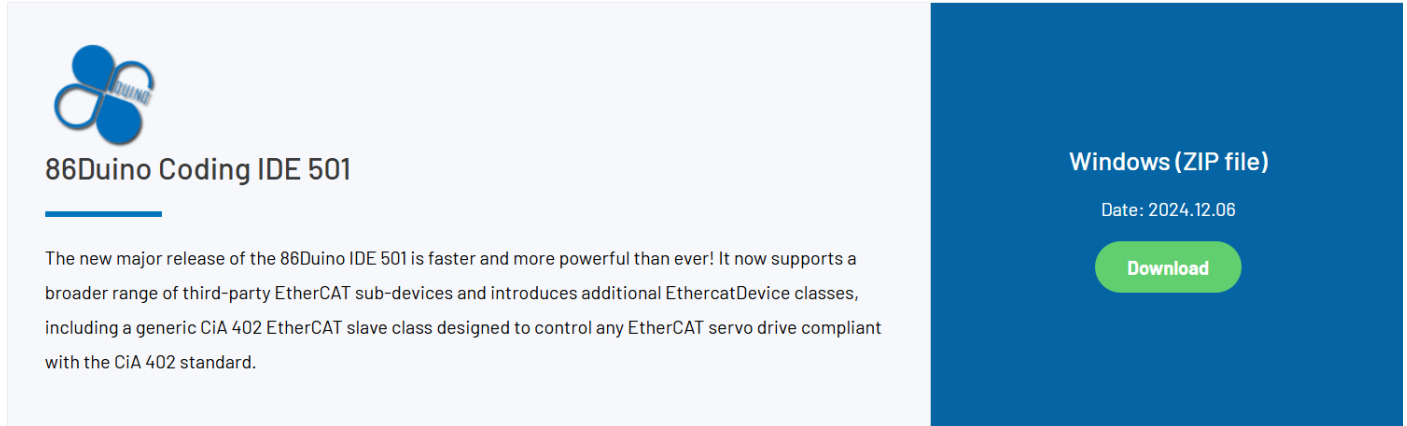
The connections are grouped by function:

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 \oplus and C, and open the circuit between P \oplus 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/>.



86Duino Coding IDE 501

The new major release of the 86Duino IDE 501 is faster and more powerful than ever! It now supports a broader range of third-party EtherCAT sub-devices and introduces additional EthercatDevice classes, including a generic CiA 402 EtherCAT slave class designed to control any EtherCAT servo drive compliant with the CiA 402 standard.

Windows (ZIP file)
Date: 2024.12.06
[Download](#)

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.

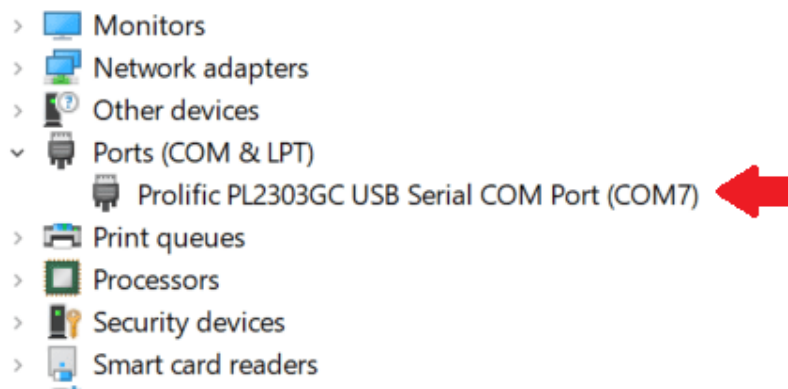
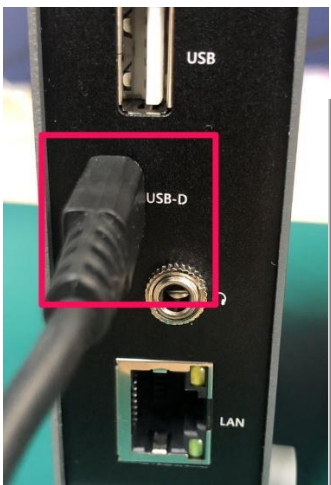
86Duino Coding IDE 501+ looks like below.



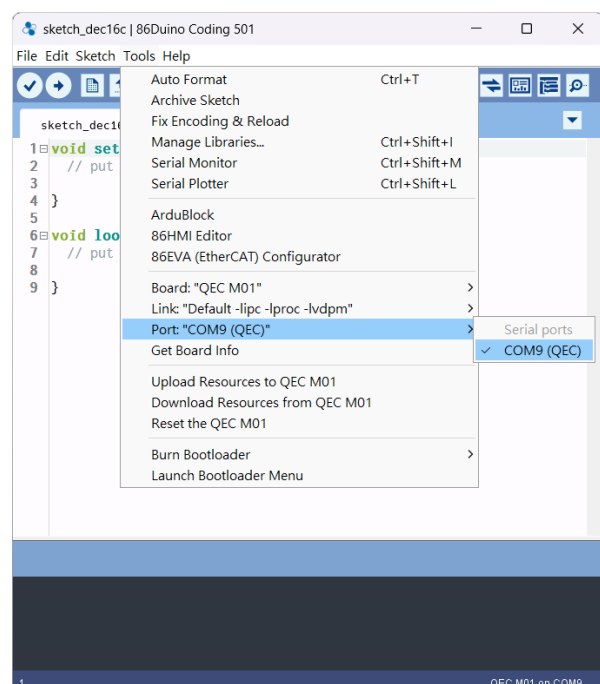
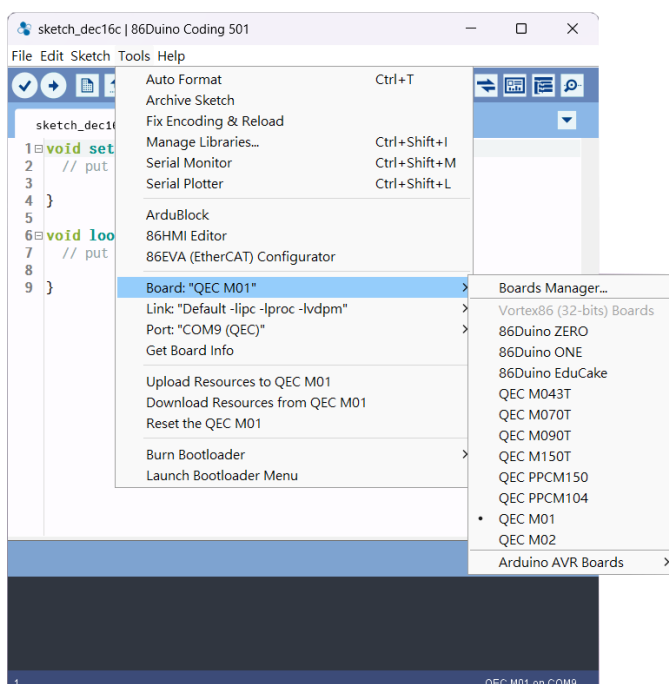
3. Connect to PC and set up the environment

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. Use 86EVA with ArduBlock

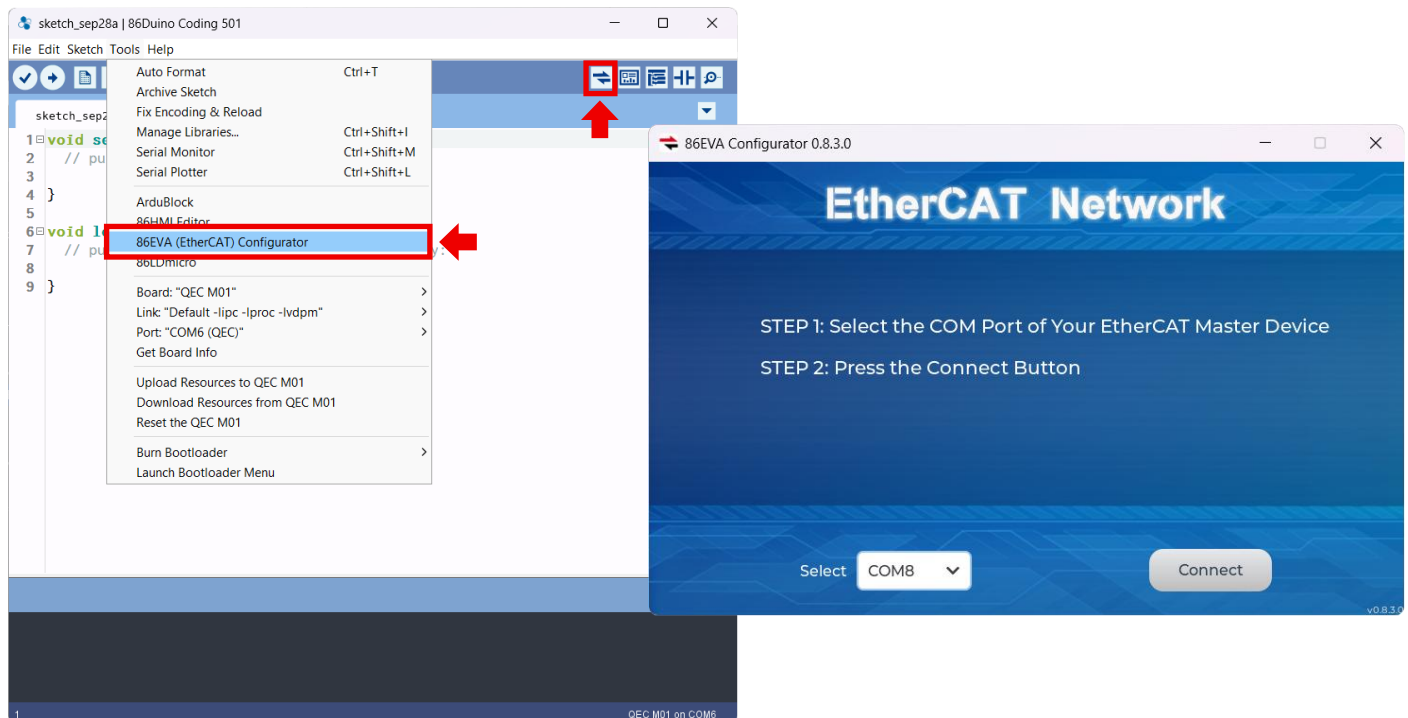
This example shows how to operate the EtherCAT MDevice (QEC-M-01) and the Delta ASDA drives (ASDA-B3 and ASDA-A2) through the 86Duino IDE's graphical low-code programming tool, 86EVA.

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.
- ArduBlock:**
 is a graphical interface for programming and IO control. It is third-party software that belongs to Arduino IDE, developed by David Li, a Shanghai-based creator, and must be attached to the IDE to operate. ArduBlock is a software that converts graphical blocks into code and eventually generates the main program to 86Duino Coding IDE, then compiles and uploads it.

4.1 Step 1: Turn on 86EVA and scan

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



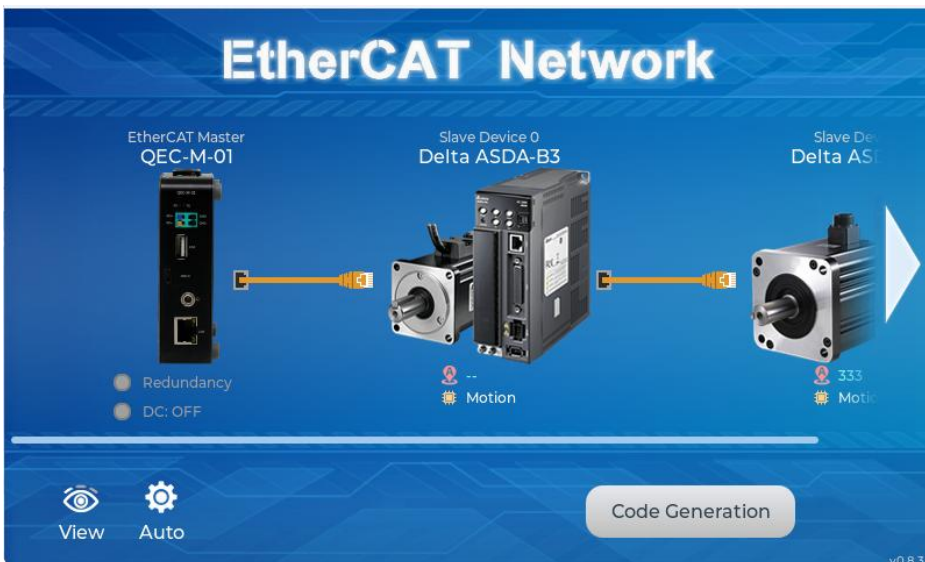
Please select the correct COM port and then click the **“Connect”** button.



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.



4.2 Step 2: Set the parameters

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

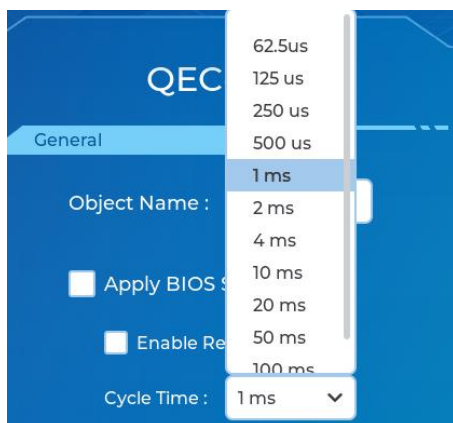
4.2.1 QEC-M-01

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



Please check the following configures.

1. Uncheck "**Apply BIOS Settings**".
2. Select "**1ms**" to the Cycle Time.



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



4.2.2 Delta ASDA-B3

Press twice on the image of the Delta ASDA-B3 to see the parameter settings.

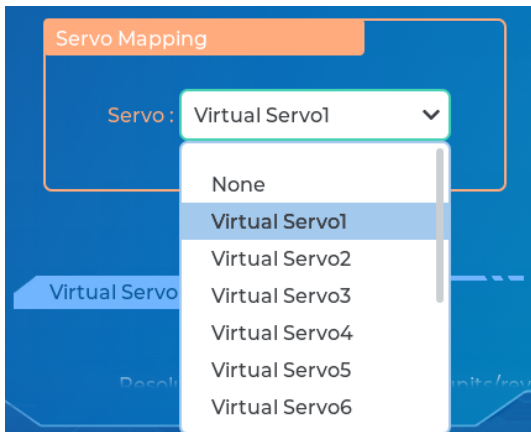


The page will show the Object Name, Alias Address, Vendor ID, Product Code, Virtual Arduino Mapping, and Virtual Servo Configuration parameters.

Continue down to the **“Virtual Arduino”** area.



We select "**Virtual Servo1**" in the drop-down box of "**Servo**" in the "**Servo Mapping**" area. This setting is for ArduBlock tool configuration.



Users can also view the servo configuration parameters through the "**Virtual Servo Configuration**" area, including resolution, max velocity, max acceleration, max deceleration, homing methods, homing velocity, and homing acceleration.



The resolution value shown here is read back from the drive via EtherCAT and reflects the actual encoder resolution of the connected motor.

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



4.2.3 Delta ASDA-A2

Press twice on the image of the Delta ASDA-A2 to see the parameter settings.

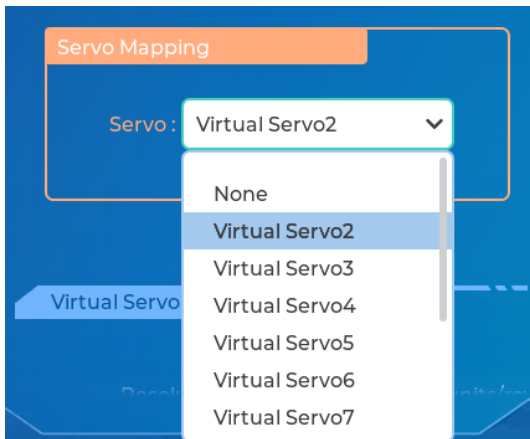


The page will show the Object Name, Alias Address, Vendor ID, Product Code, Virtual Arduino Mapping, and Virtual Servo Configuration parameters.

Continue down to the “**Virtual Arduino**” area.



We select "**Virtual Servo2**" in the drop-down box of "**Servo**" in the "**Servo Mapping**" area. This setting is for ArduBlock tool configuration.



Users can also view the servo configuration parameters through the "**Virtual Servo Configuration**" area, including resolution, max velocity, max acceleration, max deceleration, homing methods, homing velocity, and homing acceleration.



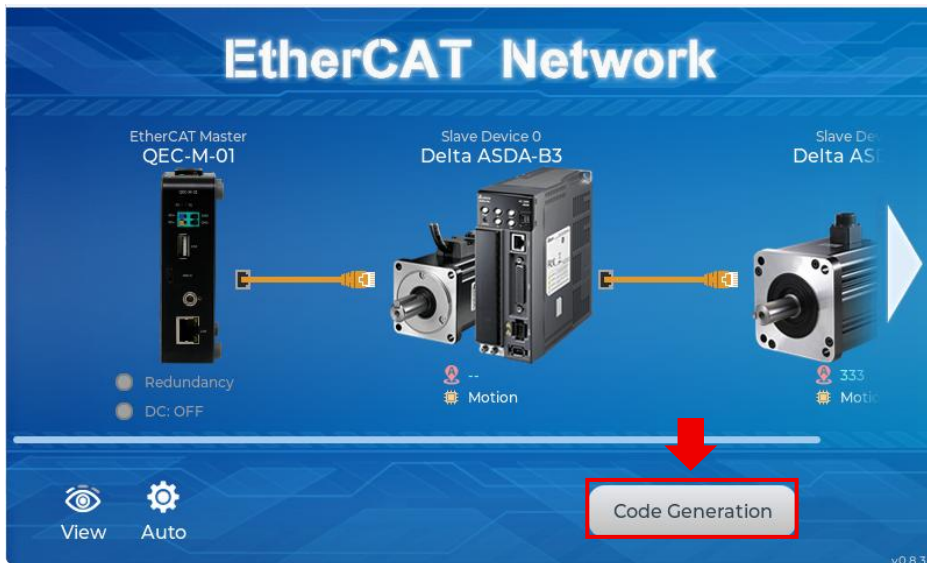
The resolution value shown here is read back from the drive via EtherCAT and reflects the actual encoder resolution of the connected motor.

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

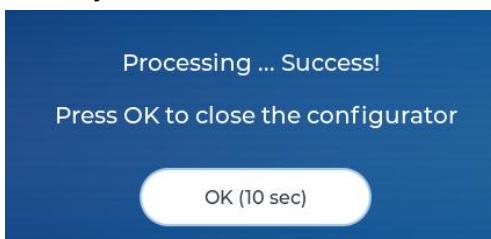


4.3 Step 3: Generate the code

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_sep10b: Main Project (.ino, depending on your project name)
- myeva.cpp: C++ program code of 86EVA
- myeva.h: Header file of 86EVA

```

1 #include "myeva.h"
2 void setup() {
3   EVA.begin();
4   // 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

4.4 Step 4: Turn on ArduBlock and setup

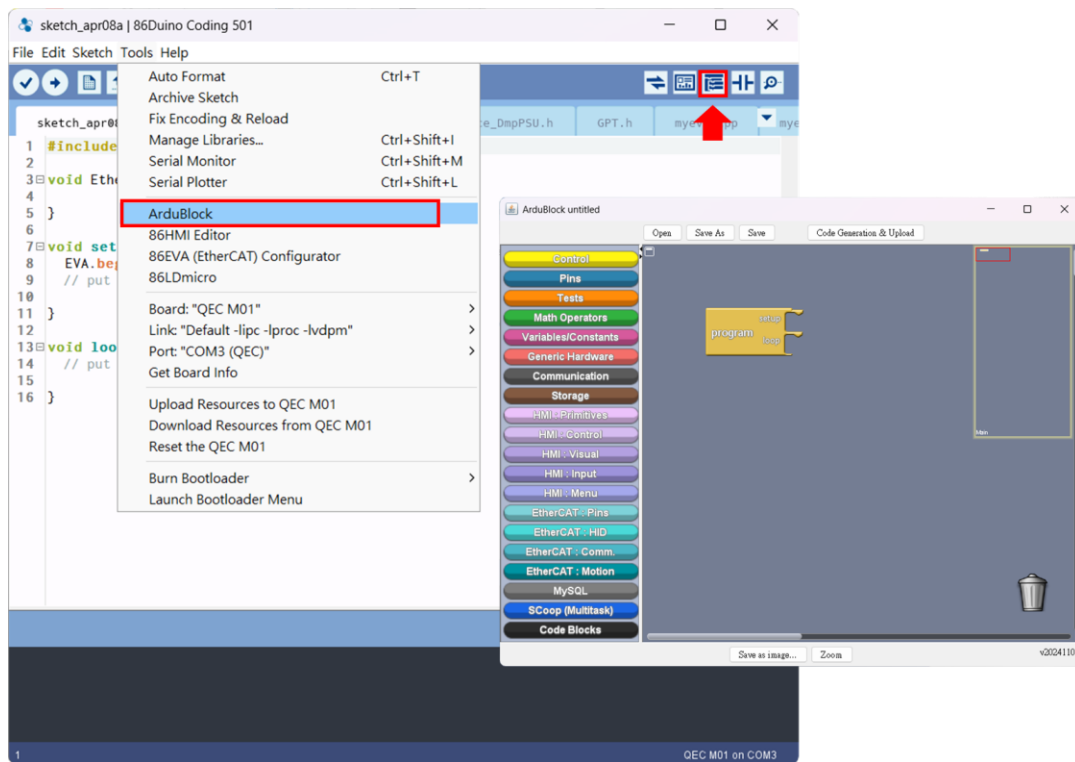
The programming code from 86EVA are set as the following by default:

- The EthercatMaster object (EcatMaster) represents the QEC-M-01, while the EthercatDevice_CiA402 represents the ASDA-B3 (motor0) and ASDA-A2 (motor1) drives.
- CiA402 mode: Profile Position (PP) mode.
- EtherCAT mode: ECAT_SYNC.

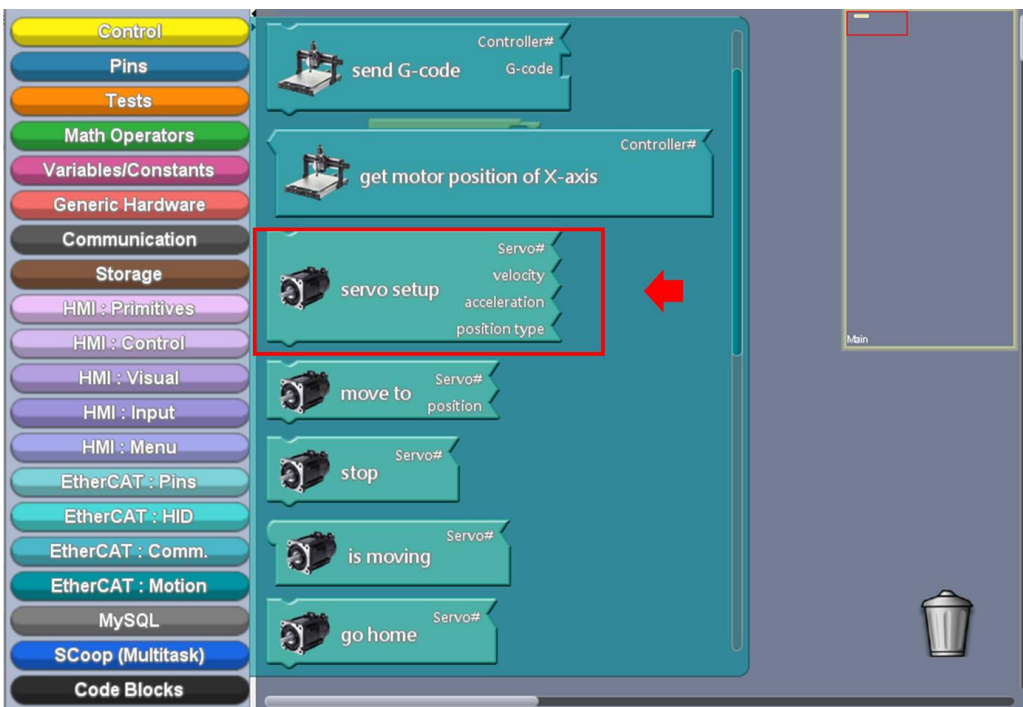
The following parameter is configured by the user in 86EVA:

- EtherCAT Cycle Time: 1 millisecond.
- ASDA-B3: Virtual Servo1
- ASDA-A2: Virtual Servo2

Open ArduBlock.



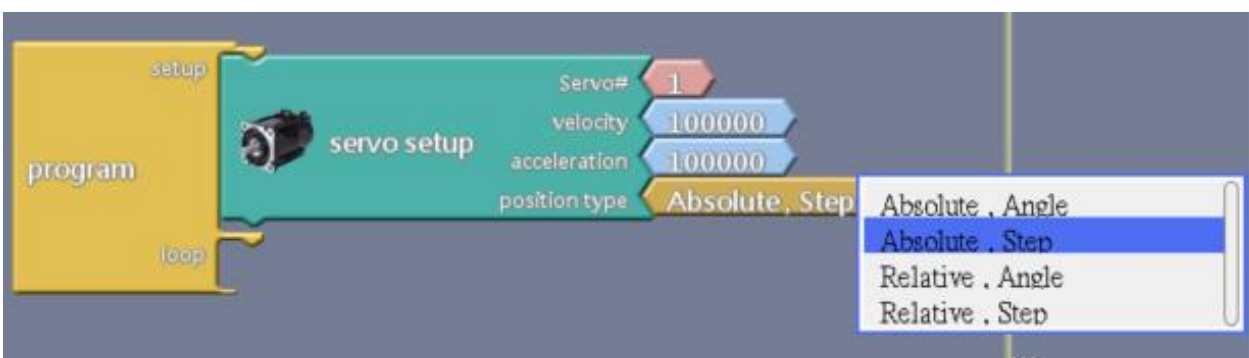
First, we need to initialize the servo parameters, including the Virtual Servo number, velocity, acceleration, and position type, so we drag the "servo setup" block from the "EtherCAT: Motion" class into the program's setup section.



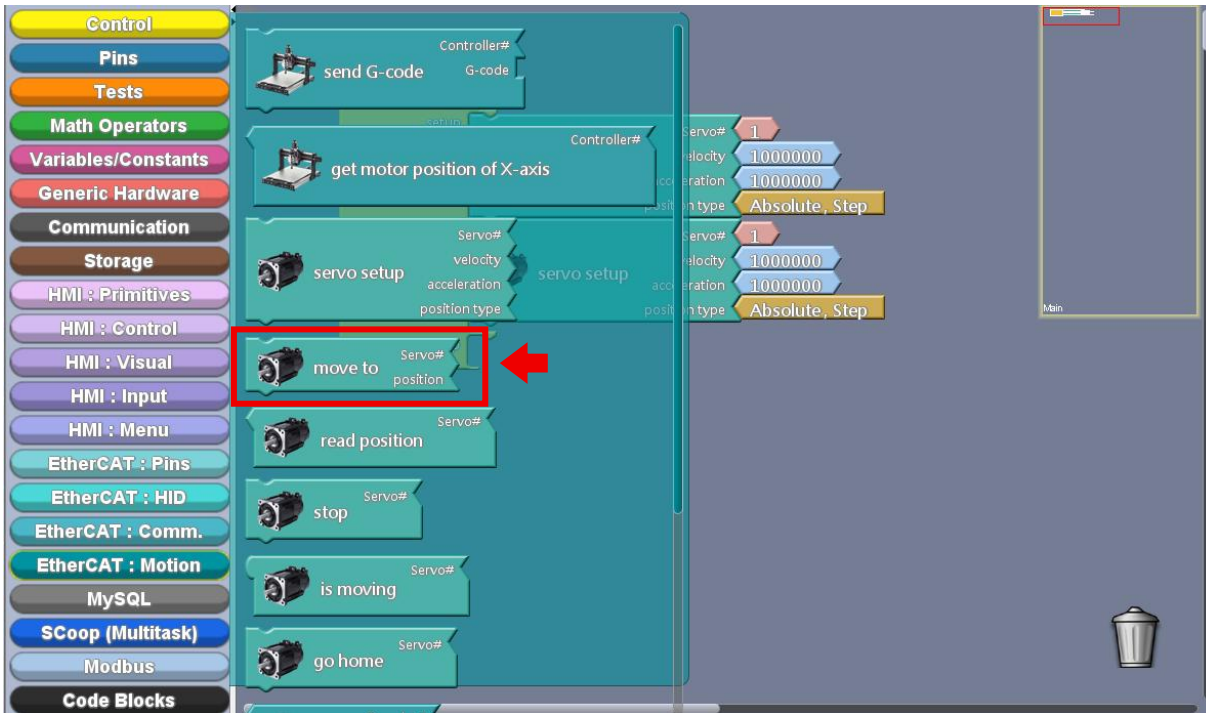
In this example, we use two "servo setup" blocks, and set Servo# to "1" and "2", velocity to "1000000", acceleration to "1000000", and position type to "Absolute, Step".



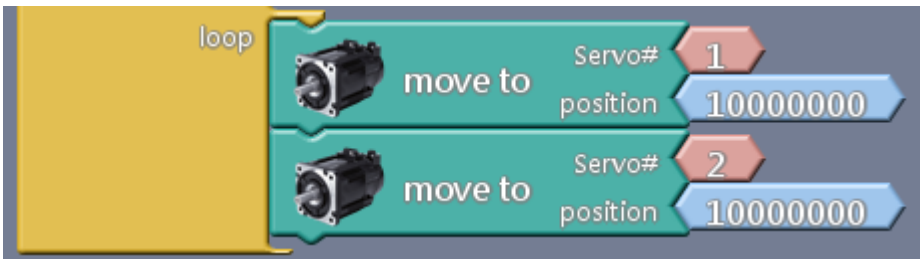
There are 4 position types: Absolute Angle, Absolute Step, Relative Angle, and Relative Step.



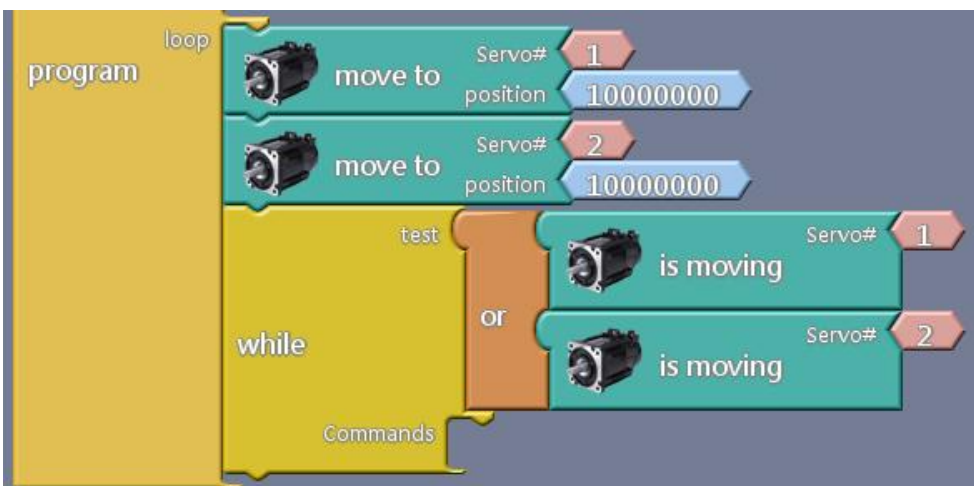
Second, we use the "move to" block from the "EtherCAT: Motion" class into the program's loop section.



In this example, we use two "move to" blocks, and set Servo# to "1" and "2", and each position to "1000000" and "-1000000".

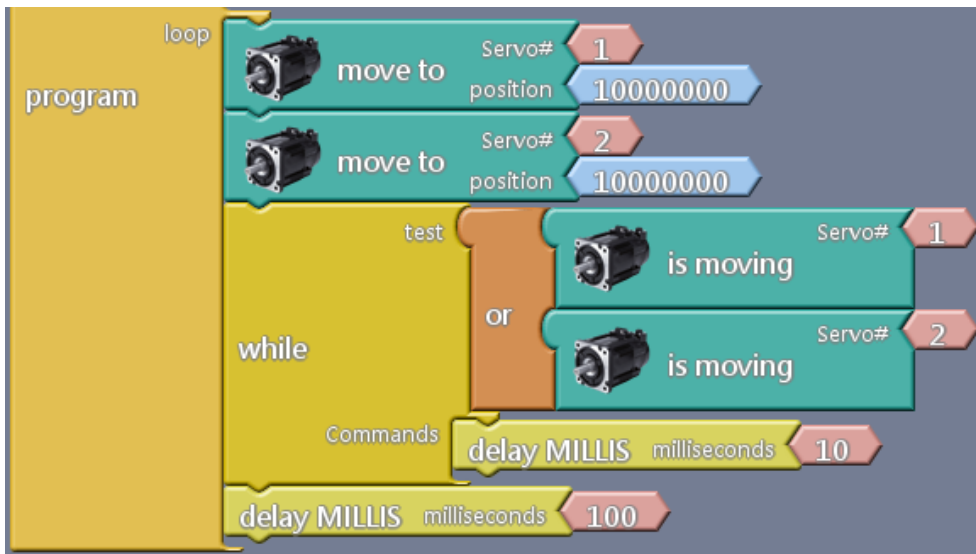


There needs to be enough moving time between the movements, so you need to stay if the motor is moving. We use a "while" block from the "Control" class, and put a "or" block (from the "Tests" class) to the test area of "while", and use 2 "is moving" blocks from the "EtherCAT: Motion" class, with Servo# "1" and "2", to check whether both motors have reached the target position.



And we use the **"delay Millis (milliseconds)"** block from the **"Control"** class after the **"while"** block, to wait for 100 milliseconds after the motors reach the position.

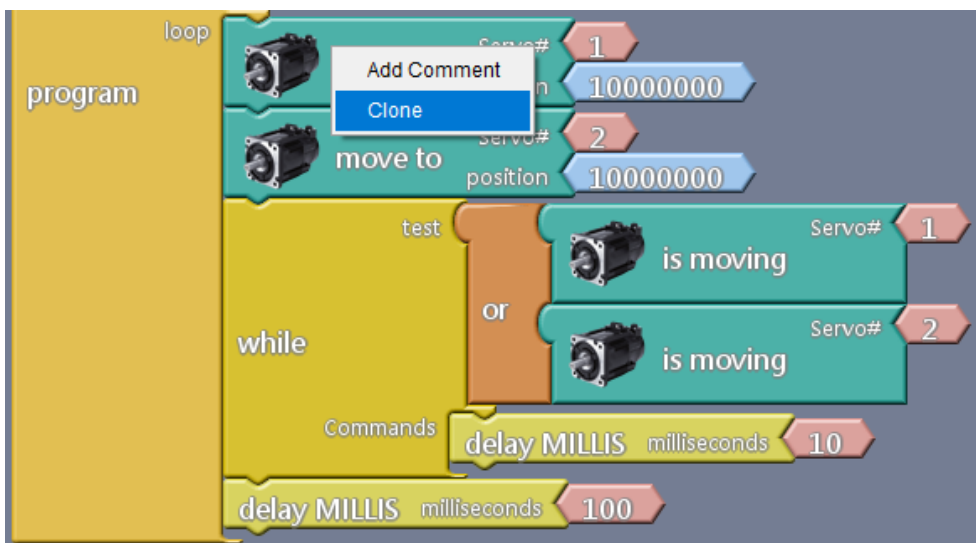
Because the **"while"** block will block the task, we also added a **"delay Millis (milliseconds)"** block for 10 ms to smooth the movement.



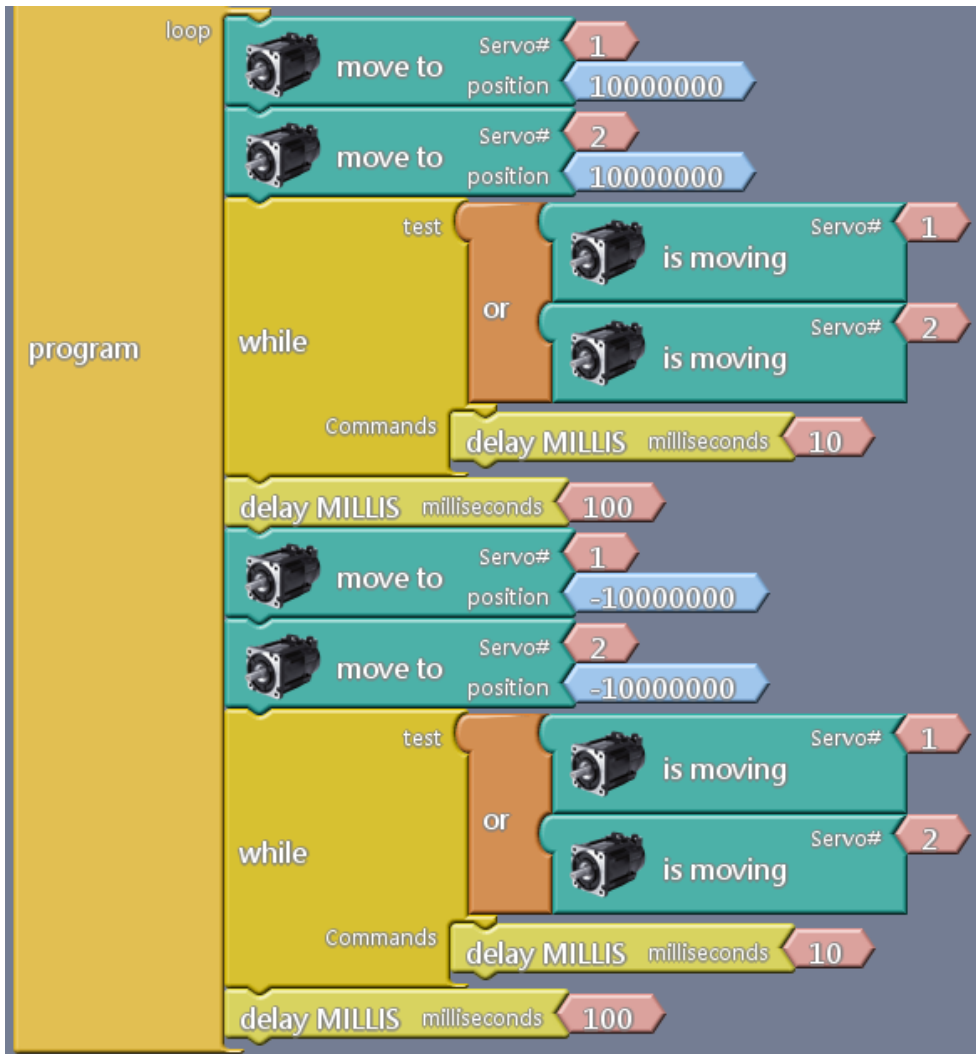
Now, both motors will move to position 10000000.

Next, we want the motors to move to position -10000000.

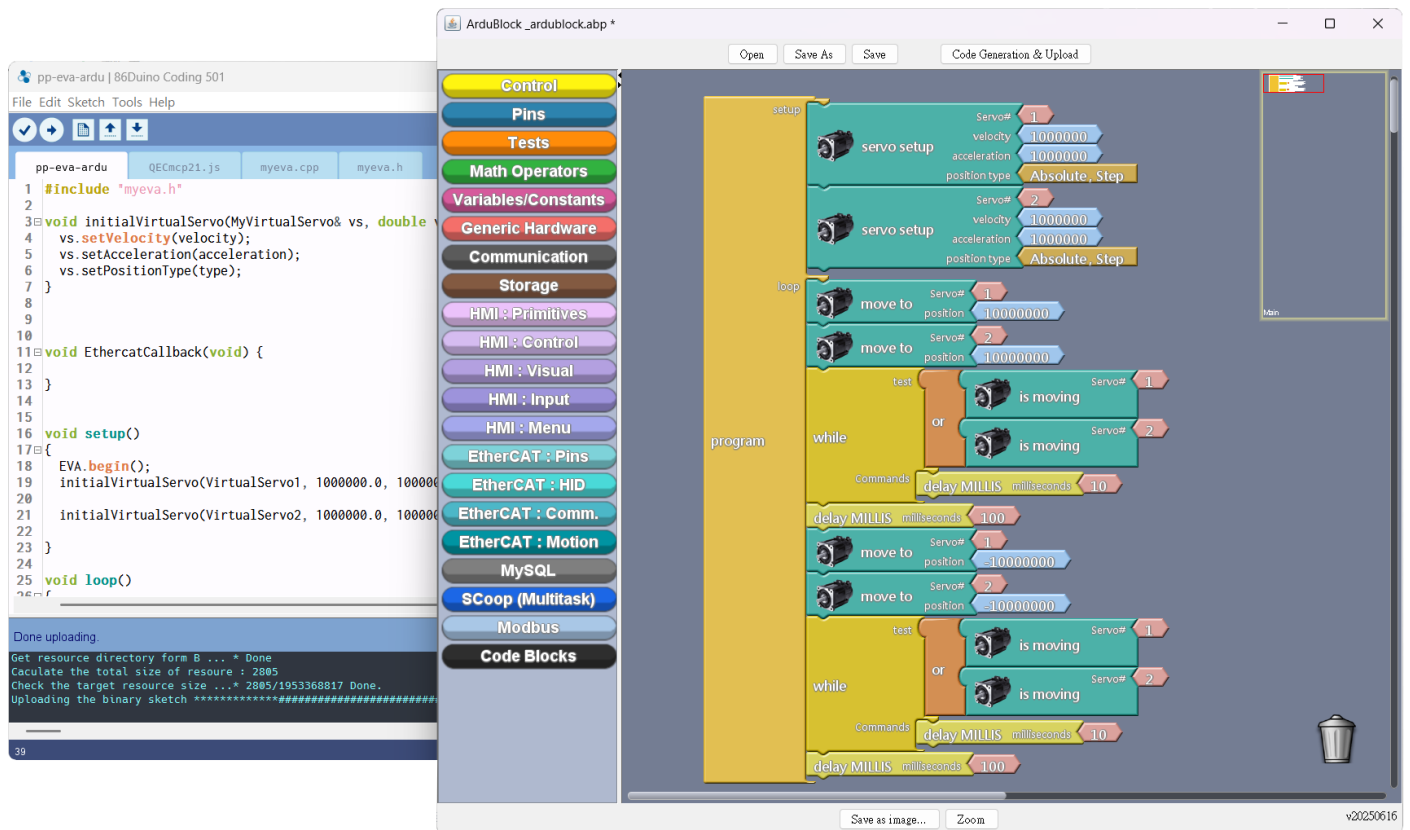
We can right-click the first block (**"move to"**) in the **"loop"** area and click **"Clone"**.



Then, put the clone blocks after the "delay Millis (milliseconds)" blocks, and change both positions to "-10000000" in the "move to" blocks.



After finishing, you can click the **"Code Generate & Upload"** button to generate and upload the code to the QEC-M-01.

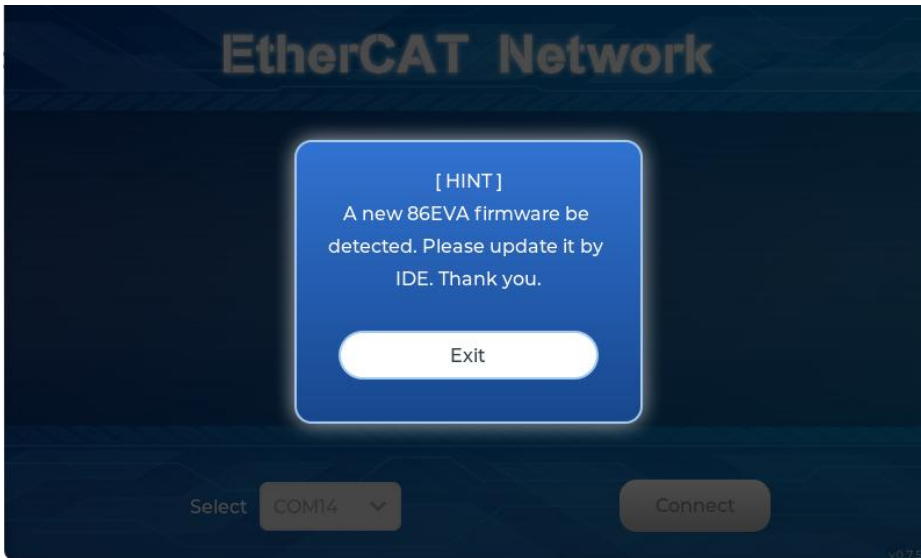


This program uses ArduBlock to configure two servo axes in Absolute Position Step mode, commanding both axes to move back and forth between +10,000,000 and -10,000,000 counts repeatedly.

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

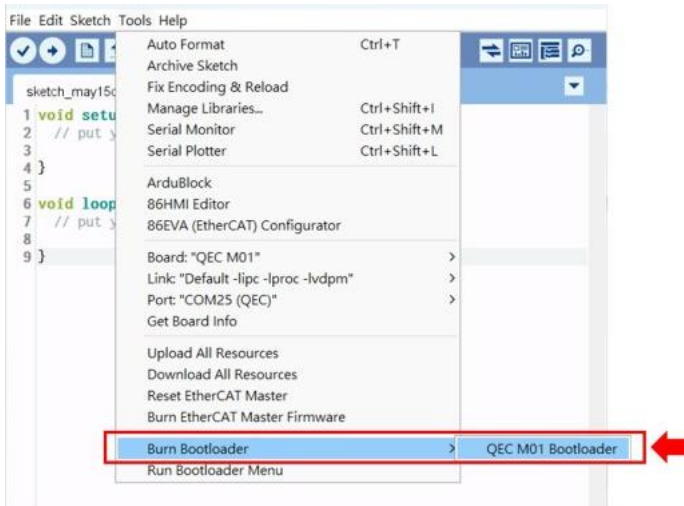
1. Download and install 86Duino IDE 500+ (or a newer version).
You can download it from [Software](#).
2. Connect the QEC-M: Use a USB cable to connect the QEC-M to your computer.
3. Open 86Duino IDE: After the installation is complete, open the 86Duino IDE software.
4. Select Board: From the IDE menu,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

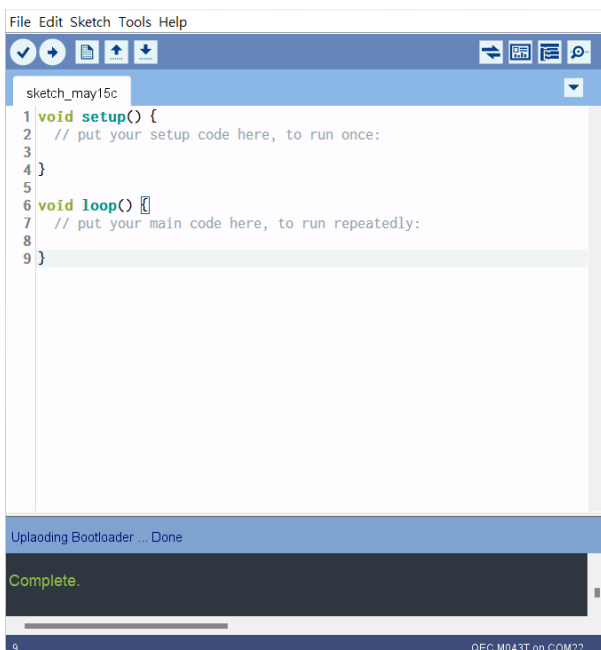
After connecting to your QEC-M product, go to “**Tools**”> “**Burn Bootloader**”.

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.

All Trademarks appearing in this manuscript are registered trademark of their respective owners. All Specifications are subject to change without notice.

©ICOP Technology Inc. 2026