

Start Guide

M56S EtherCAT Servo Motor
CiA402 PP Mode (1-axis) with 86EVA
and ArduBlock



86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

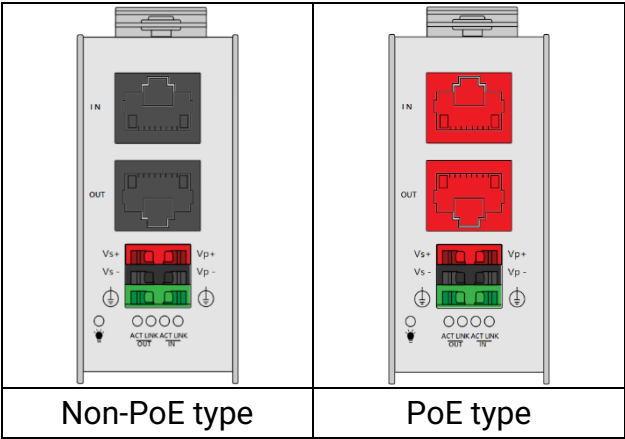
| Date | Version | Description |
|------------|------------|--------------|
| 2025/05/29 | Version1.0 | New Release. |

Preface

In this guide, we will show you how to use the EtherCAT MDevice QEC-M-01 and the Moon’s M56S EtherCAT Servo Motor.

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:

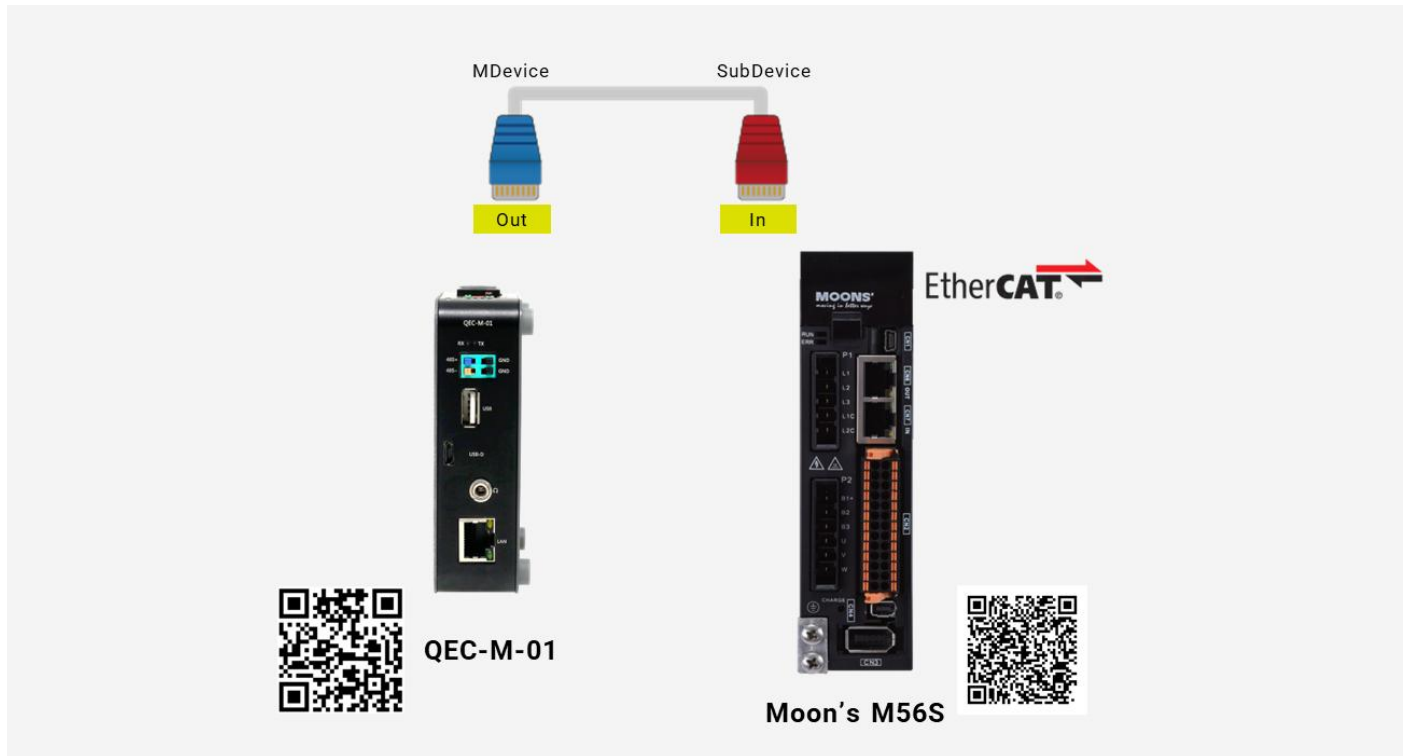


- 2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).
- 3. QEC’s PoE power supply is up to 24V/3A.

1. Connection and wiring hardware

The following devices are used here:

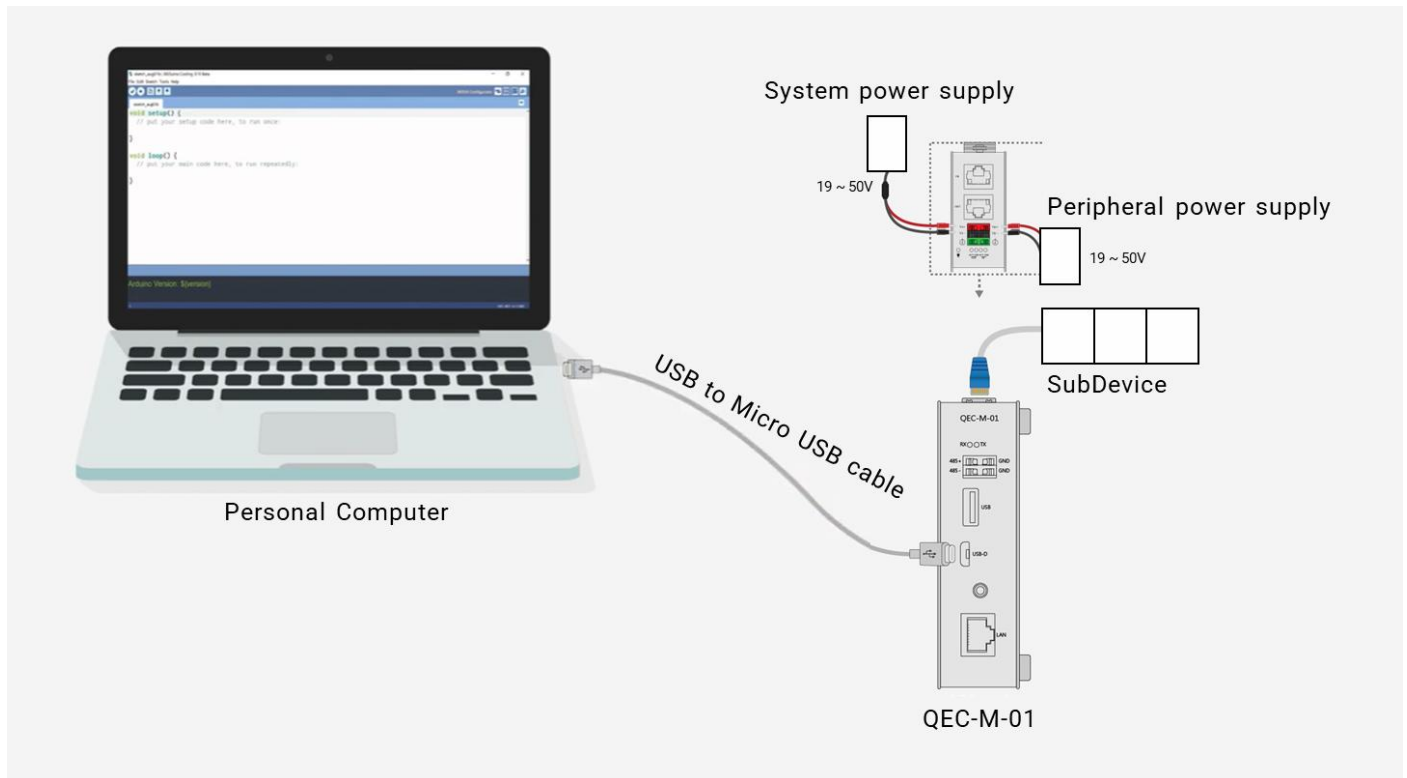
1. QEC-M-01 (EtherCAT MDevice)
2. M56S (EtherCAT Servo Motor)
3. 24V power supply & EU-type terminal cable & LAN cable
4. 220V/400V AC input



1.1 QEC-M-01

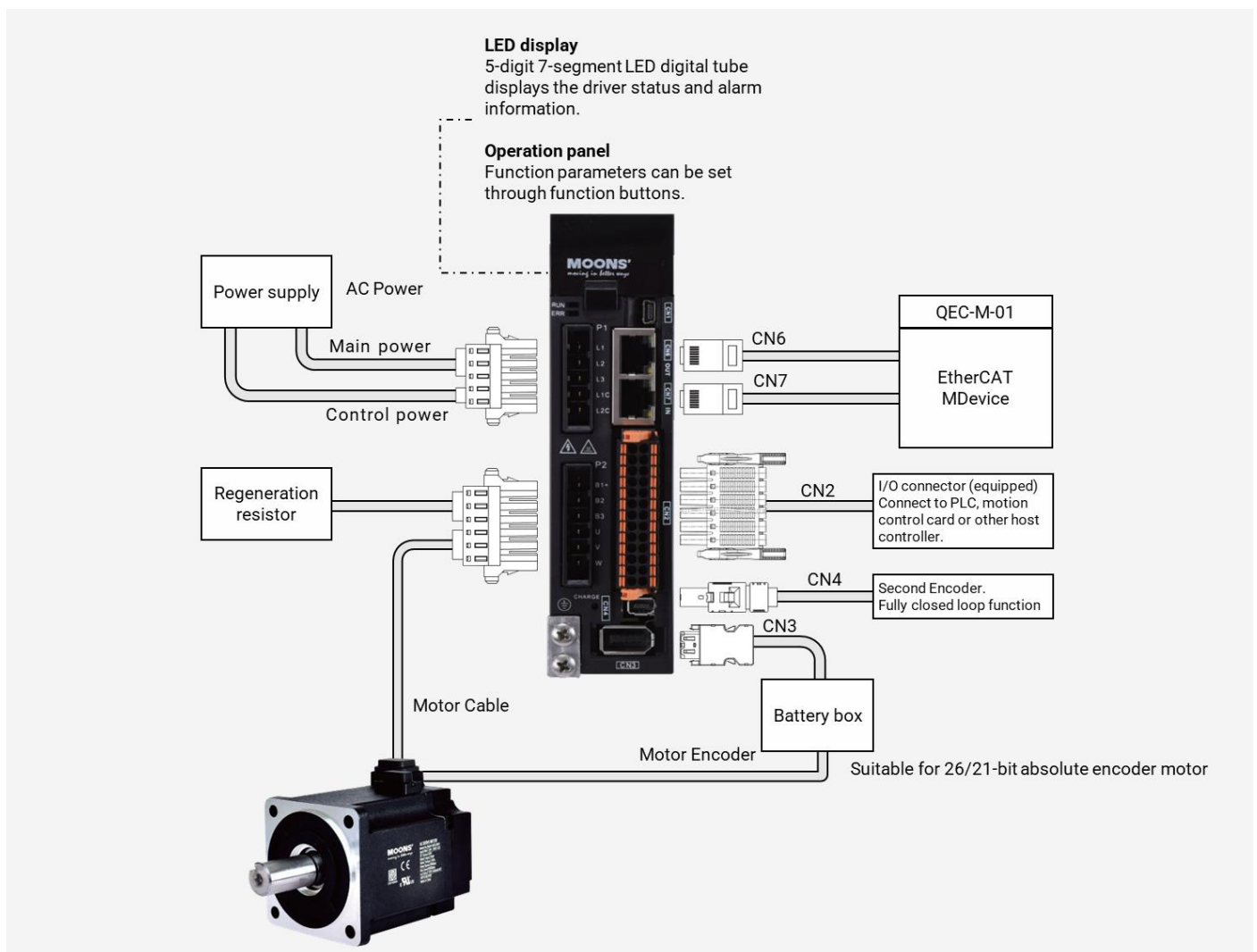
QEC EtherCAT MDevice.

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 M56S

M56S series is a new generation of high-end 220V/400V AC input servo system, the power range is from 50W to 7.5KW. The drive adopts advanced motion control algorithms, coupled with a 26-bit high precision encoder motor, the velocity loop bandwidth is up to 3.5KHz. Efficient automatically adjusts the loop gain, superior vibration suppression and external disturbance compensation performance. The drive supports full closed-loop control, built-in regenerative energy absorption resistor, STO safety function and dynamic braking module (for certain models). It supports the control mode of position, velocity, torque and Q programming.



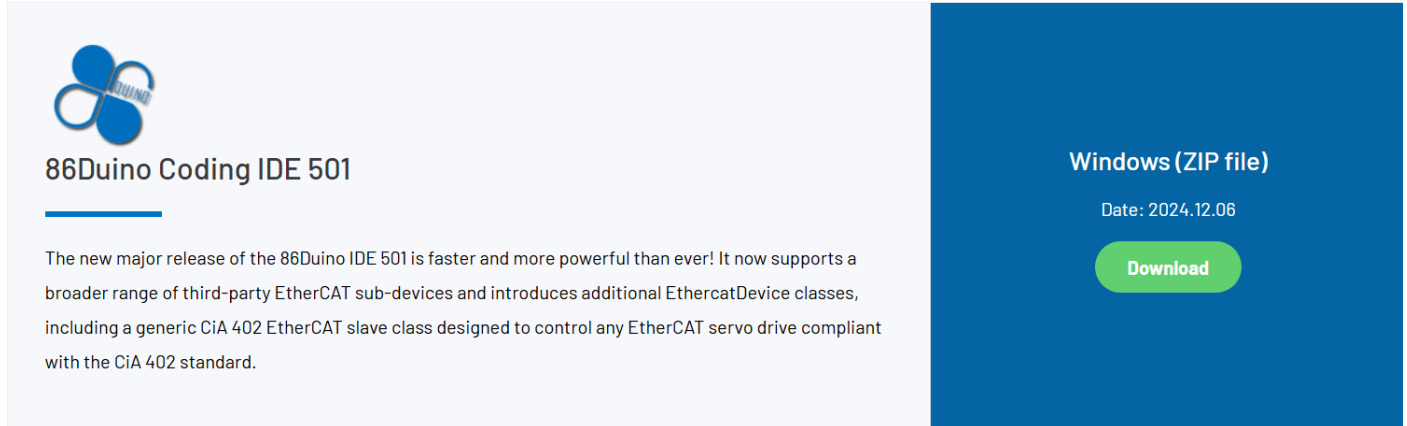
M56S Driver Wiring Overview

- **Power Supply**
 - AC input split into:
- **Communication**
 - CN6 / CN7: EtherCAT connection to QEC-M-01.
- **I/O Control**
 - CN2: Built-in I/O port.

- Connects to PLC or motion controller.
 - For start, stop, alarm reset, etc.
- **Encoder Feedback**
 - CN3: Main encoder (supports 21/26-bit absolute encoders).
 - CN4: Second encoder input for full-closed loop.
- **Front Panel**
 - LED Display: Shows status and alarms.
 - Function Keys: For parameter setting.

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

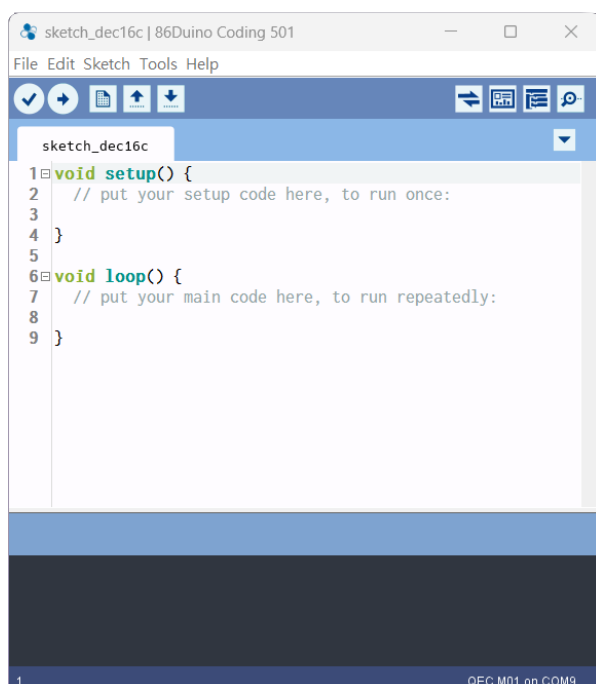
After downloading, please unzip the downloaded zip file, no additional software installation is required, just double-click 86duino.exe to start the IDE.



Note:

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

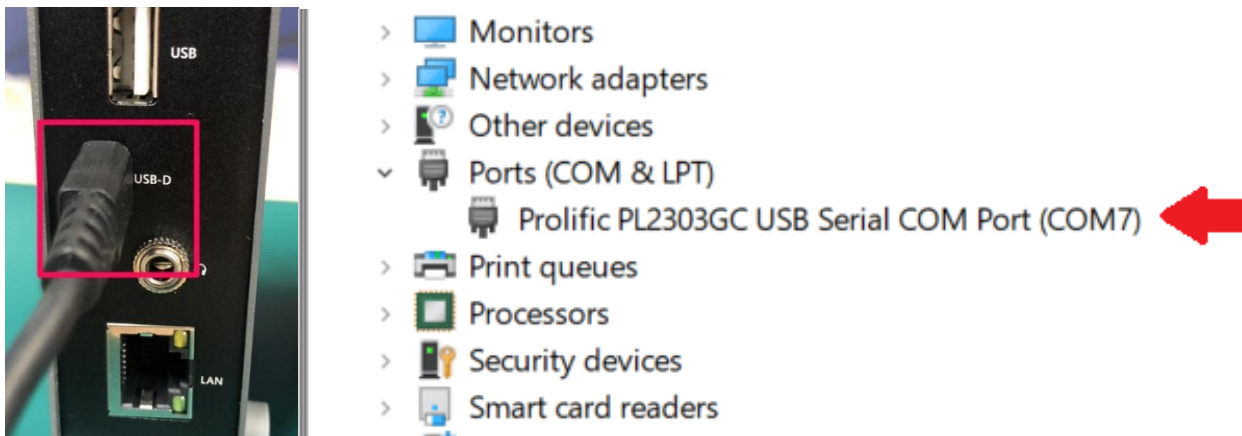
86duino Coding IDE 501+ looks like below.



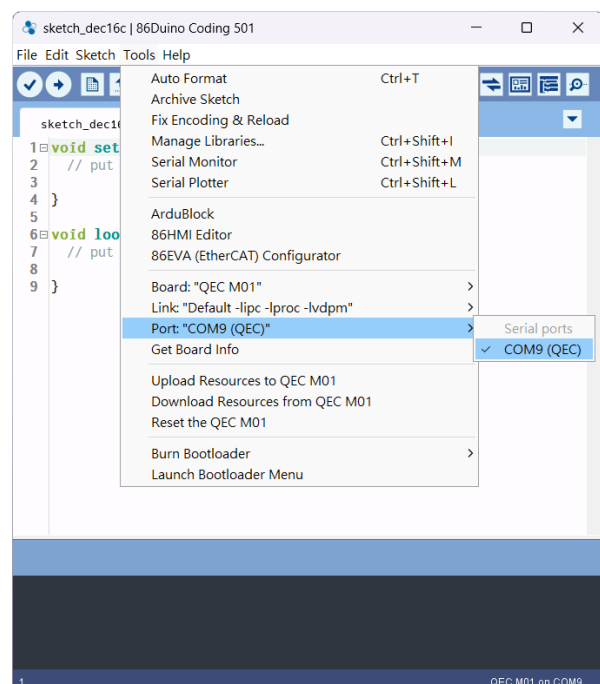
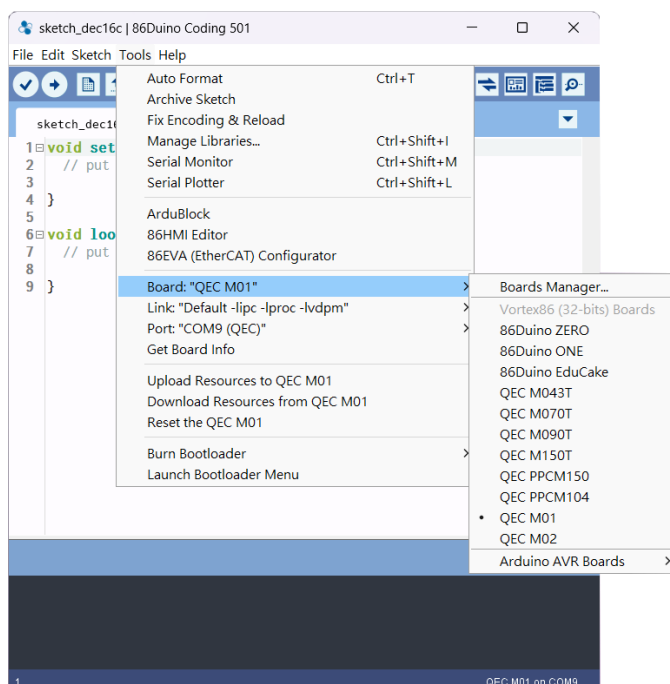
3. Connect to PC and set up the environment

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-M-01 (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 code

This example shows how to operate the EtherCAT MDevice (QEC-M-01) and the Moon's M56S EtherCAT Servo Motor through the 86Duino IDE's graphical low-code programming tool, 86EVA and ArduBlock.

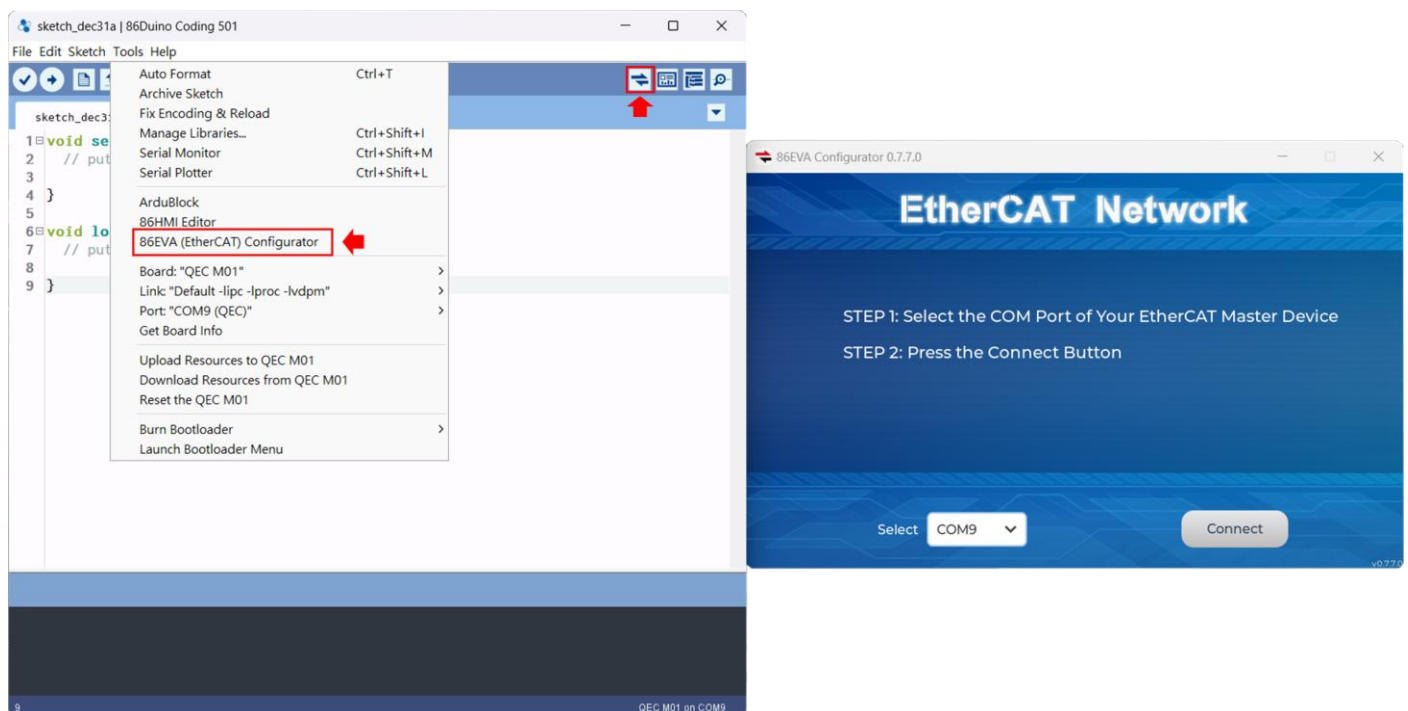
Software Tools Description:

- 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.

This program sets the M56S servo motor to CiA402 Profile Position (PP) mode. We'll controls the motor by updating its target position based on its operational state.

Step 1: Turn on 86EVA and scan

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



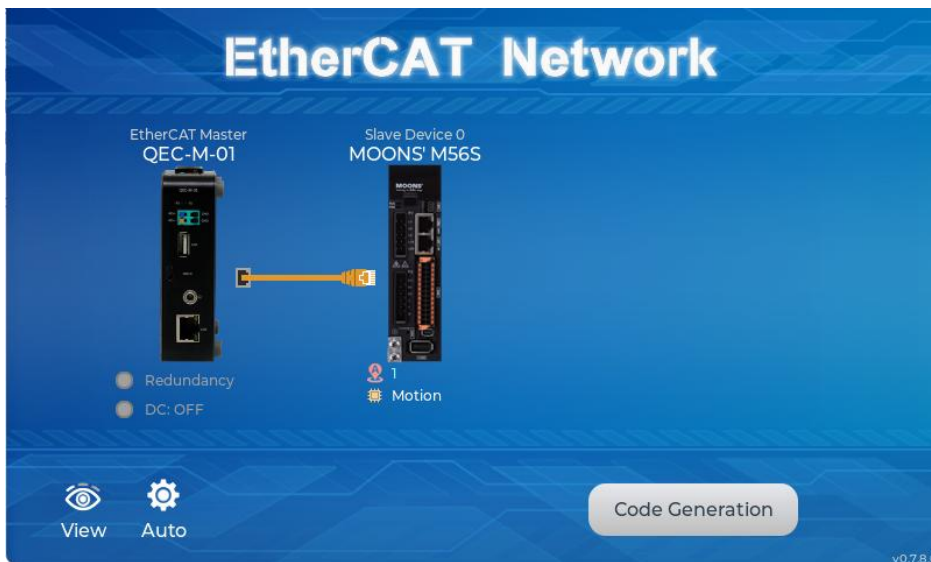
Please select the correct COM port and then click the “Connect” button.



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.

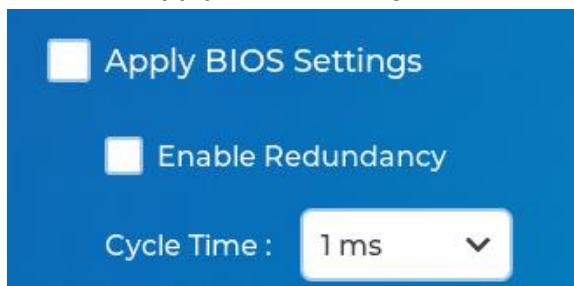
QEC-M-01

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

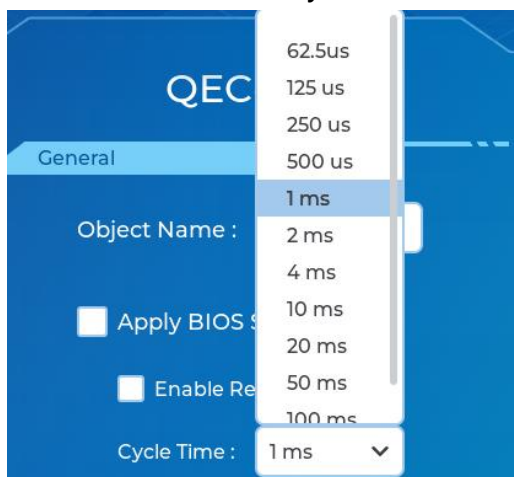


Please check the following configures.

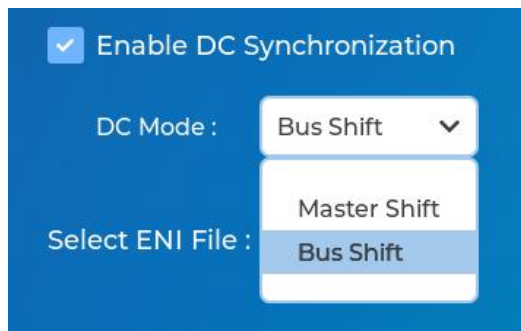
1. Turn off "Apply BIOS Settings".



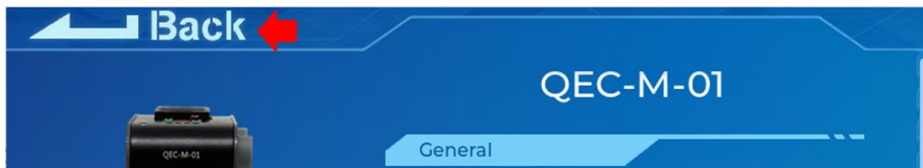
2. Select "1ms" to the Cycle Time.



3. Open DC Mode. Turn on "Enable DC Synchronization" and select the mode you need.



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



Moon's M56S

Press twice on the image of the M56S servo motor to see the parameter settings.

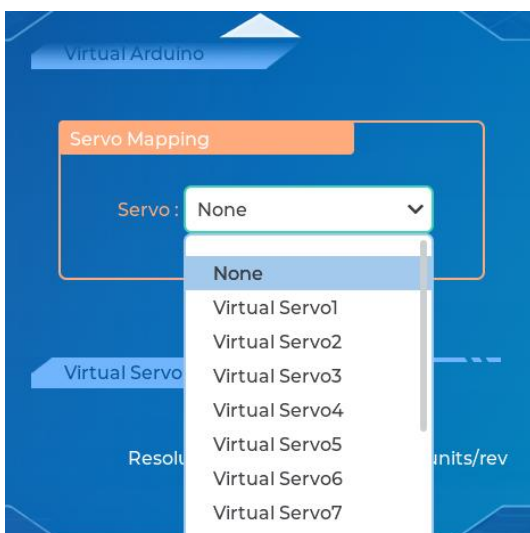


The page will show the Object Name, Alias Address, Vendor ID, Product Code, Virtual Arduino Mapping, and Virtual Servo Configuration parameters.

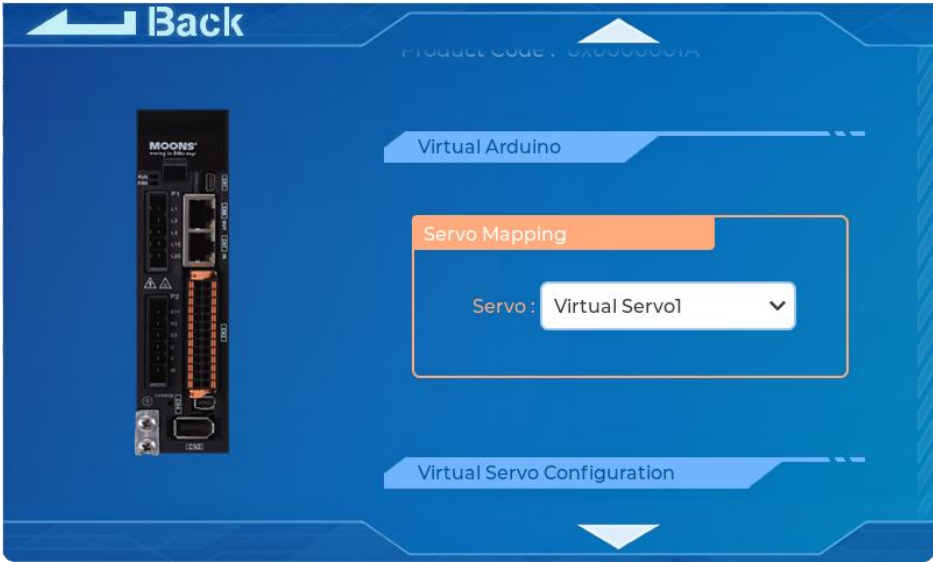
Continue down to the **"Virtual Arduino"** area.



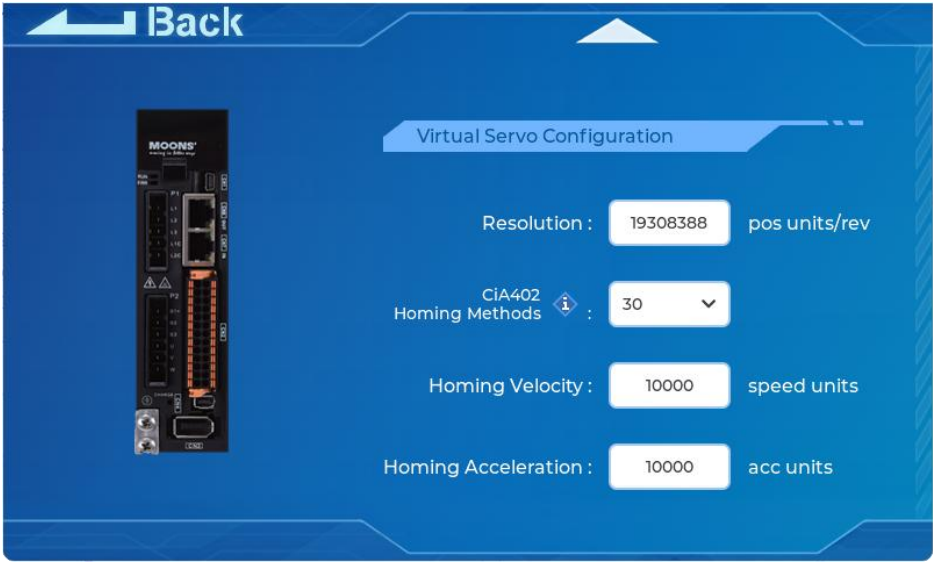
We select "Virtual Servo1" in the drop-down box of "Servo" in the **"Servo Mapping"** area.



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.

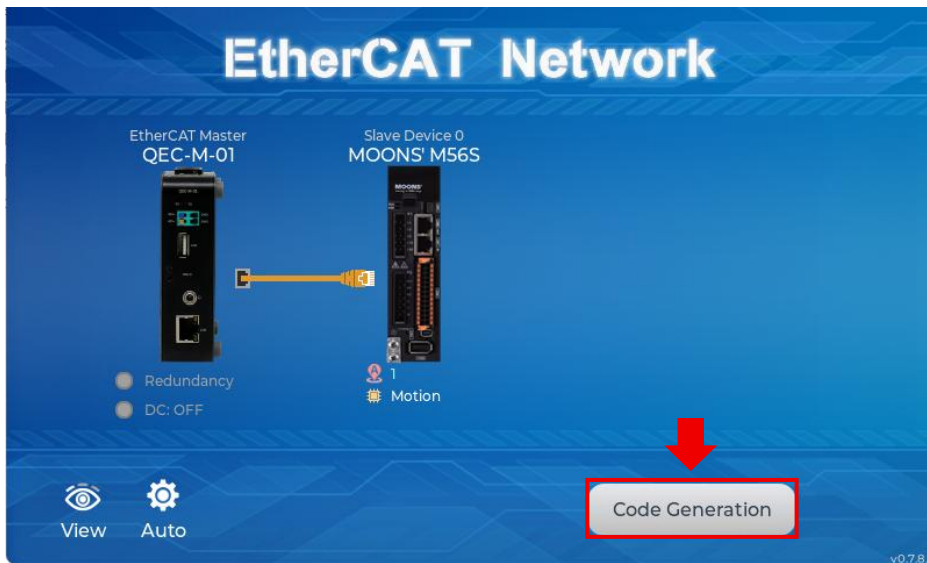


Click “**Back**” in the upper left corner to return.

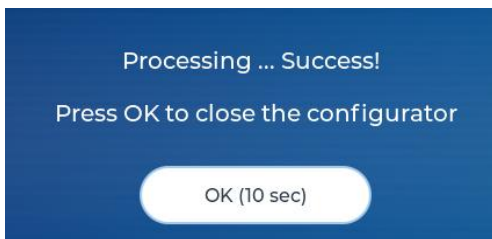


Step 3: Generate the code

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_sep10b: Main Project (.ino, depending on your project name)
- 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

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

- Moon's M56S EtherCAT Servo Motor: `EthercatDevice_CiA402` object.
- CiA402 mode: Profile Position (PP) mode.
- EtherCAT mode: ECAT_SYNC.
- Distributed Clock: Open. Follow the EtherCAT cycle time.
- DC Mode: Bus Shift.

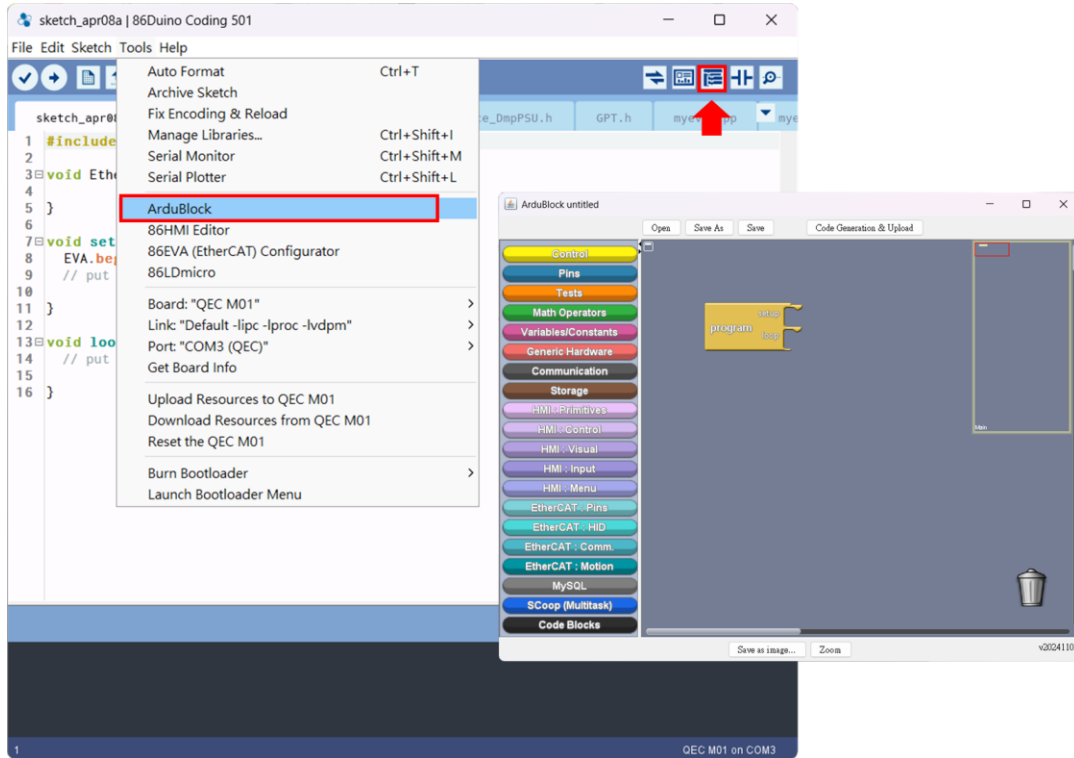
And here is the setting by users:

- EtherCAT Cycle time: 1 millisecond.
- Device Object Name: QEC-M-01 is "EcatMaster", and M56S servo motor is "Slave0".

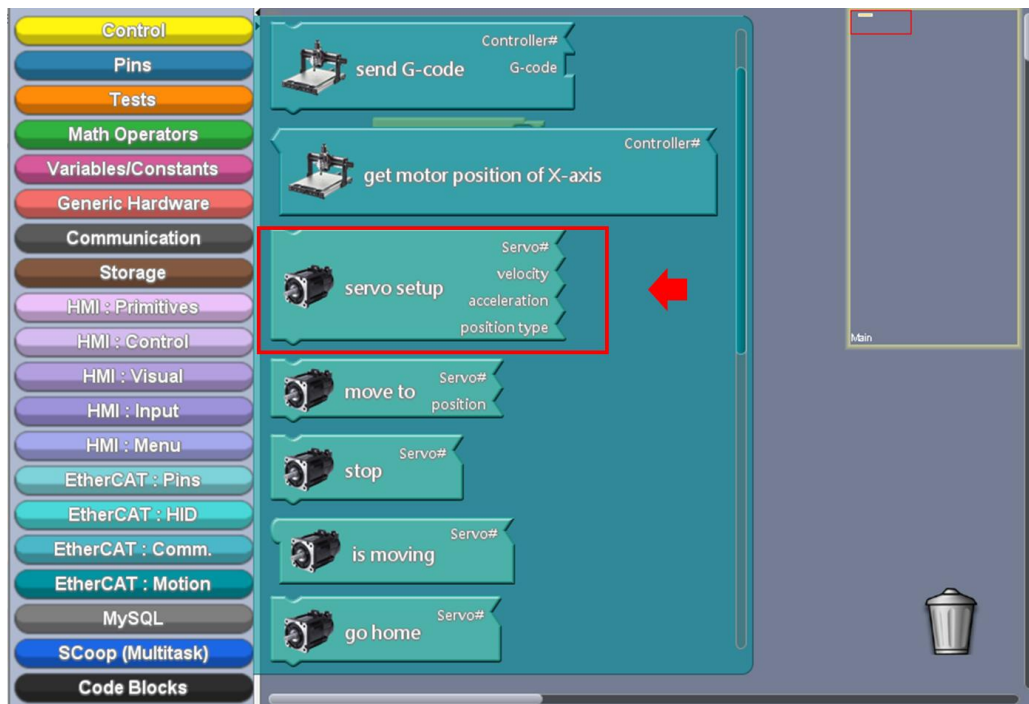
This program sets the M56S servo motor to CiA402 Profile Position (PP) mode. We'll control the motor by updating its target position based on its operational state.

Step 4: Turn on ArduBlock and setup

Next, after the 86EVA sets the Virtual Arduino Pins, we can open ArduBlock.



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



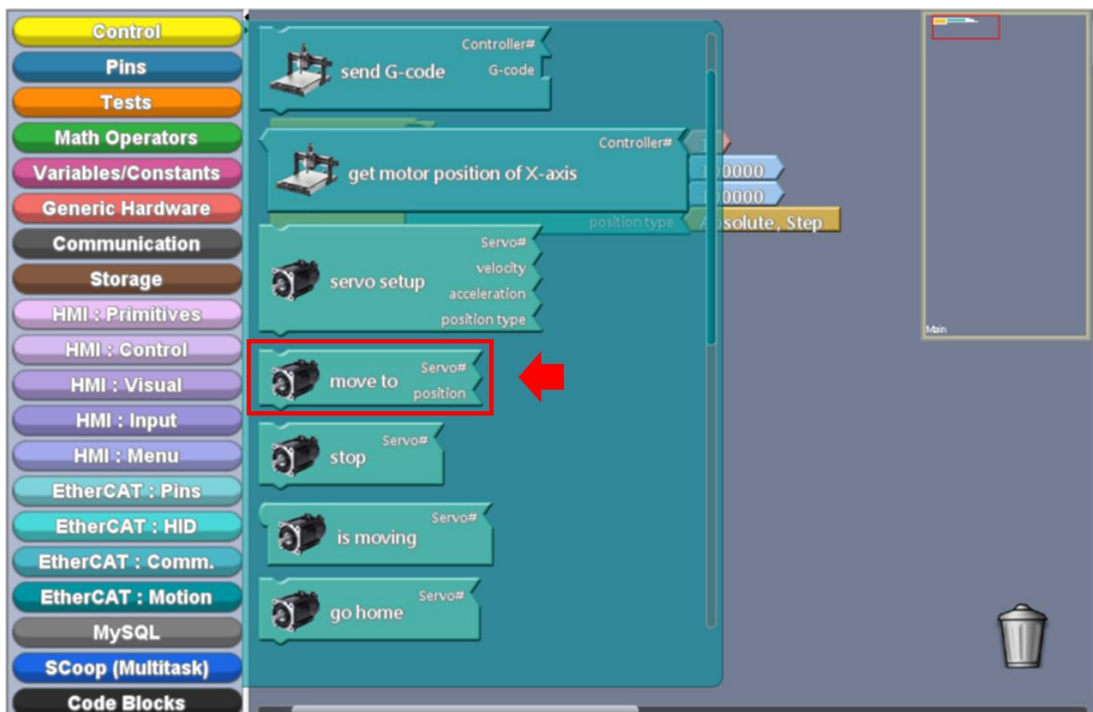
In this example, we set Servo# to "1", velocity to "100000", acceleration to "5000", and position type to "**Absolute, Step**".



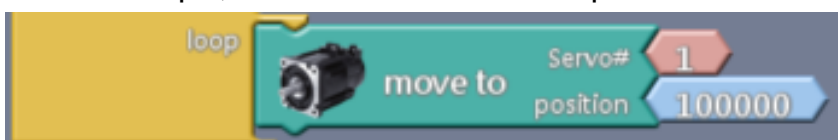
There are four types for position type: Absolute Angle, Absolute Step, Relative Angle, and Relative Step.



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

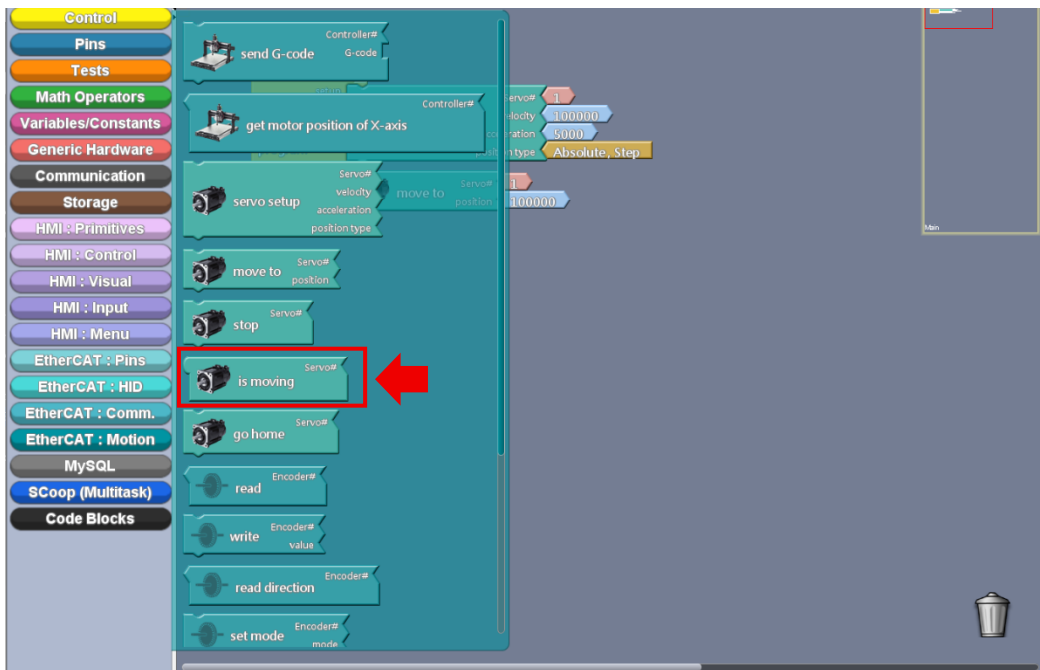


In this example, we set Servo# to "1" and position to "100000" and "-100000".

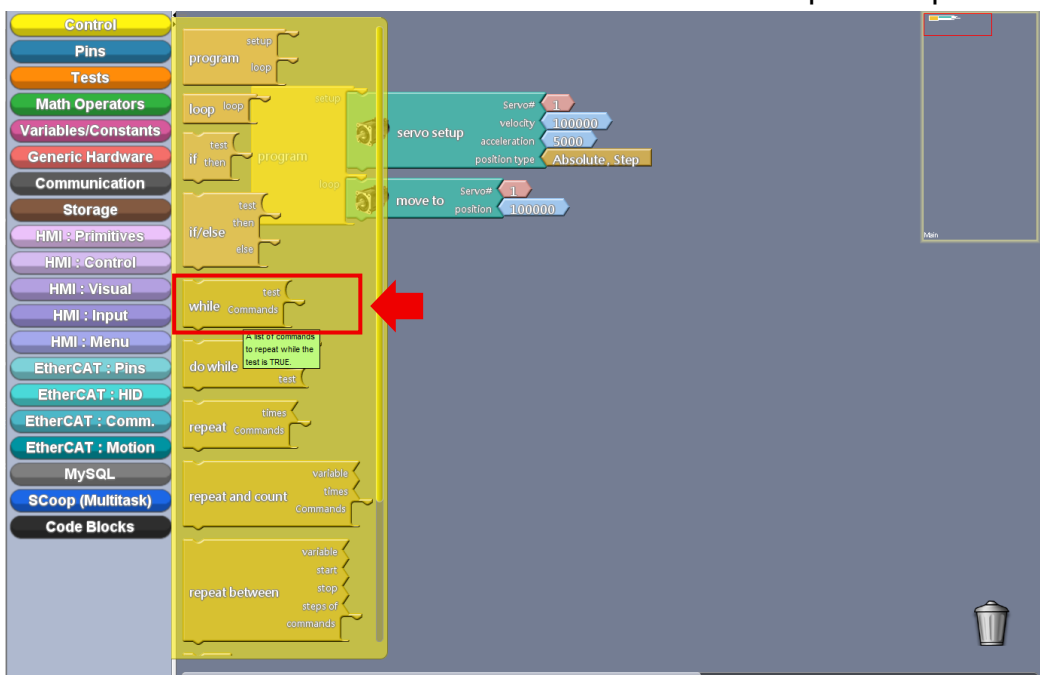


There needs to be enough moving time between the movements.

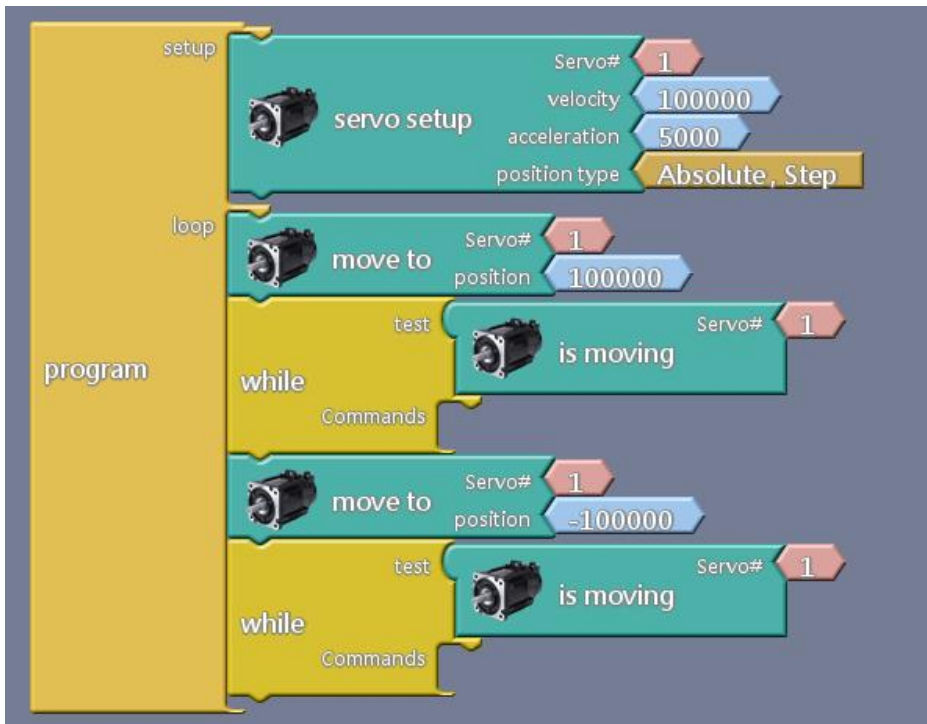
We can use the **"is moving"** block from the **"EtherCAT: Motion"** class to judge whether the motor is moving or not. It will return 0 if the motor is stopped.



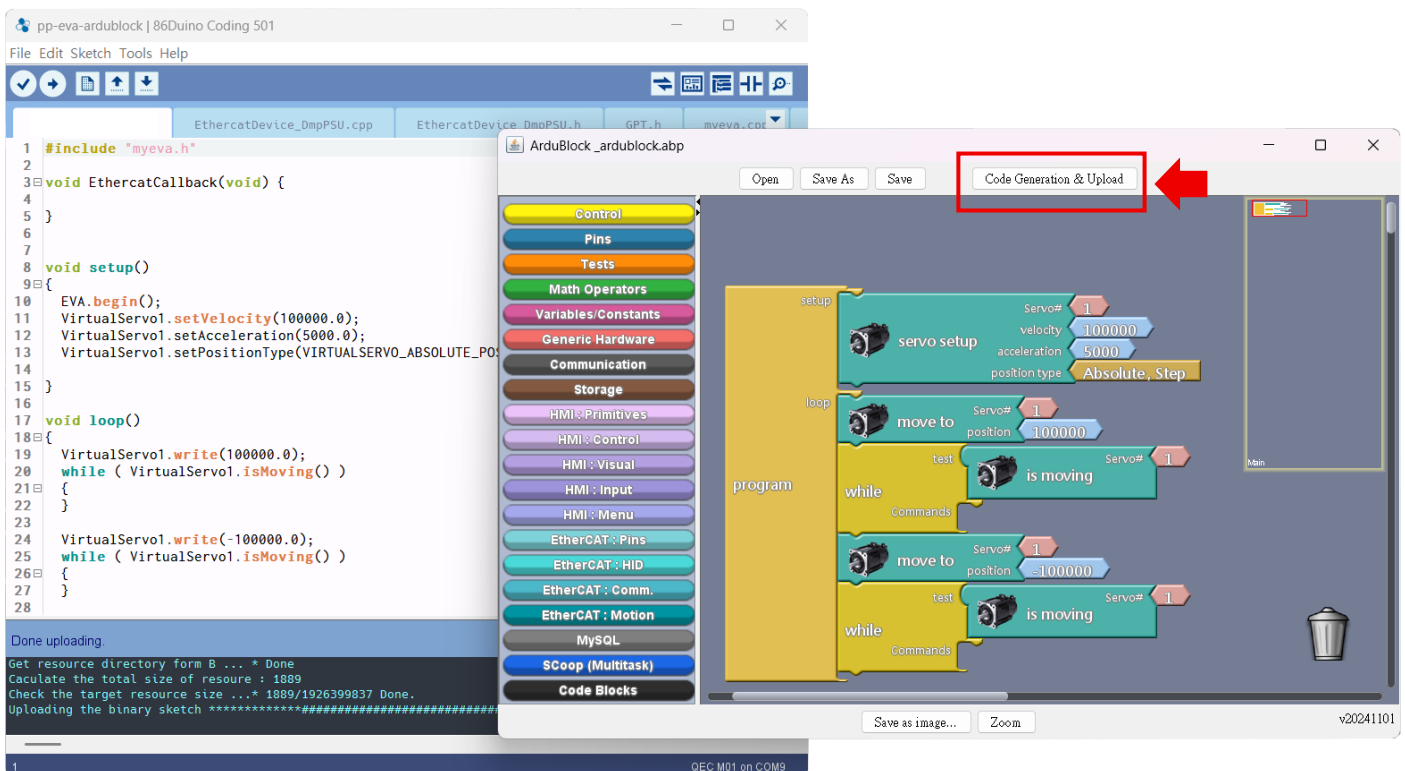
We also use **"while"** block from the **"Control"** class to keep the loop while the motor moves.



The finished block is as follows.



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

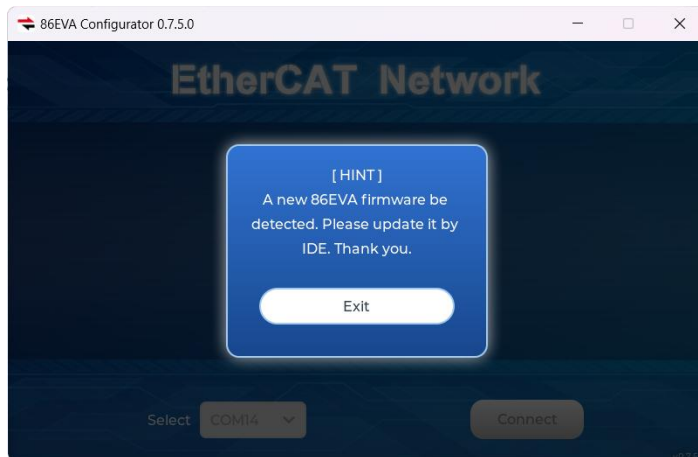


This block programming establishes EtherCAT communication, configures the M56S servo motor of velocity and acceleration, and controls the motor by updating its target position based on its operational state.

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:

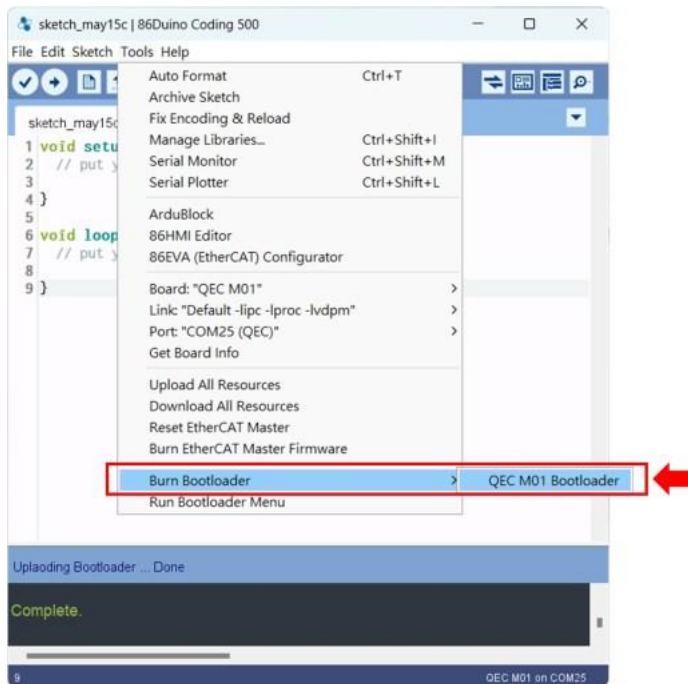
Step 1: Setting up QEC-M

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.

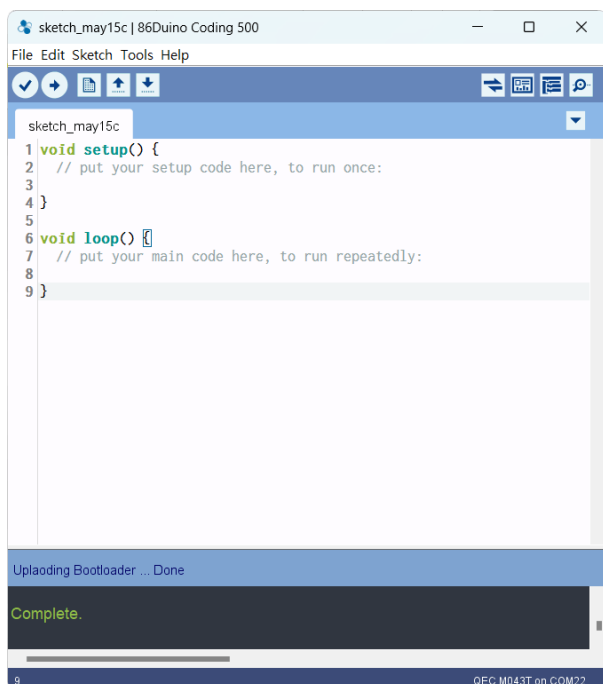
Step 2: Click “Burn Bootloader” button

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. 2025