Start Guide

Ezi-IO EtherCAT Digital I/O Control with 86EVA and ArduBlock

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

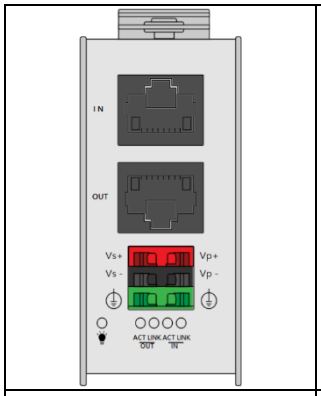
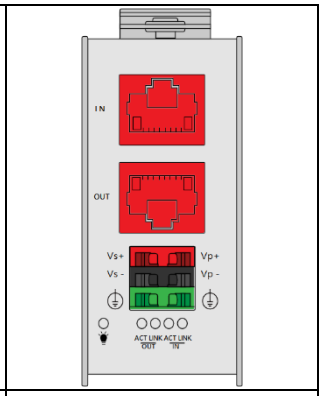
Date	Version	Description
2025/06/17	Version1.0	New Release.

Preface

In this guide, we will show you how to use the EtherCAT MDevice QEC-M-01 and the Ezi-IO (EtherCAT Input/Output Module).

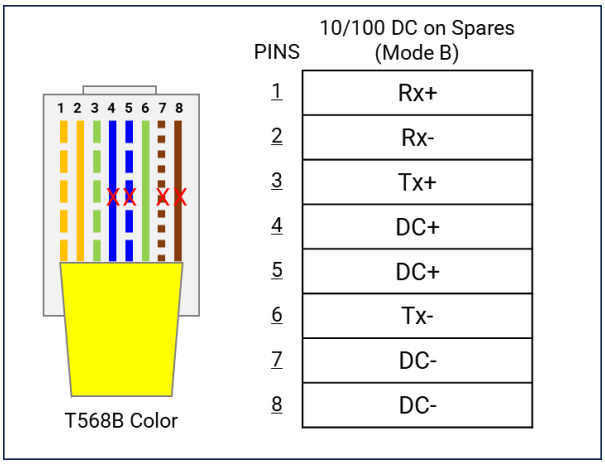
Notes QEC's PoE (Power over Ethernet)

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:



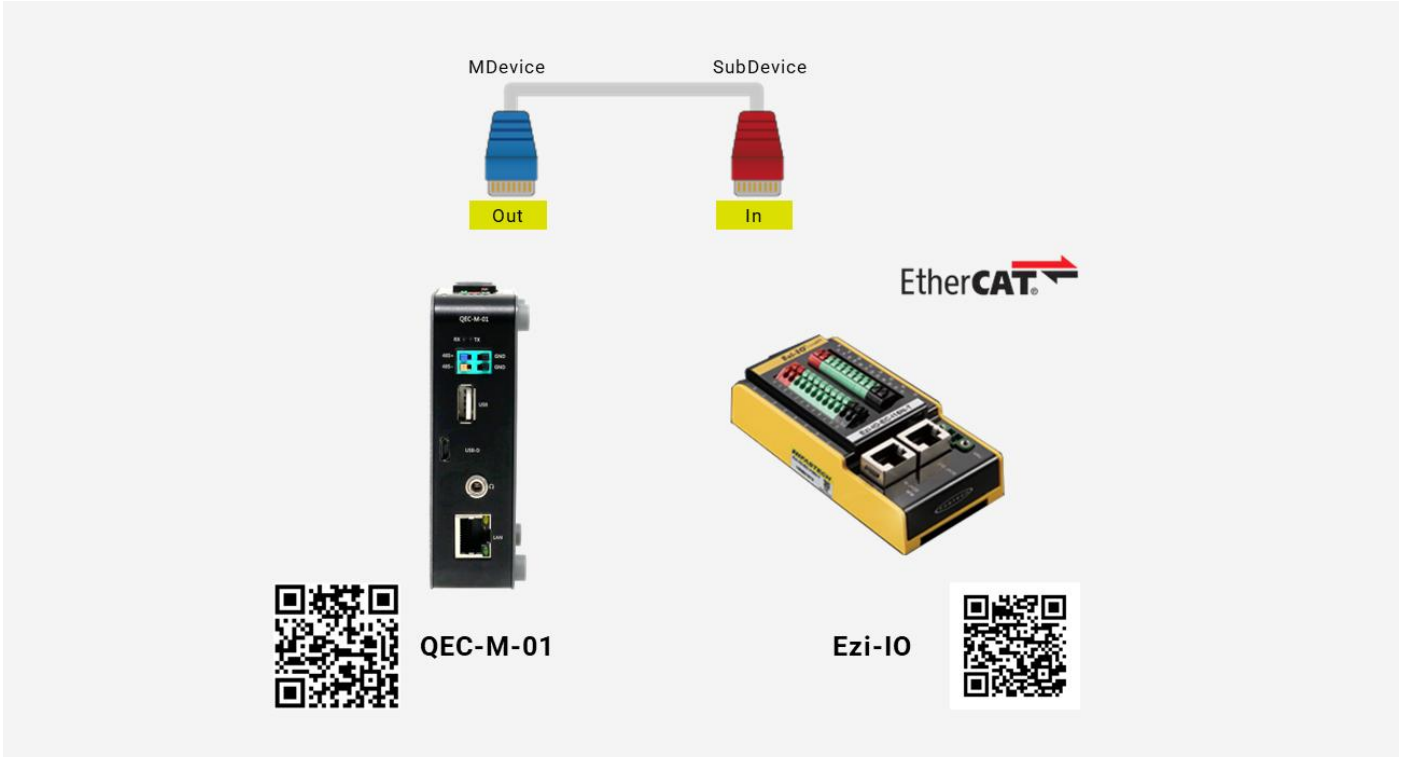
2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).

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

1. Connection and wiring hardware

The following devices are used here:

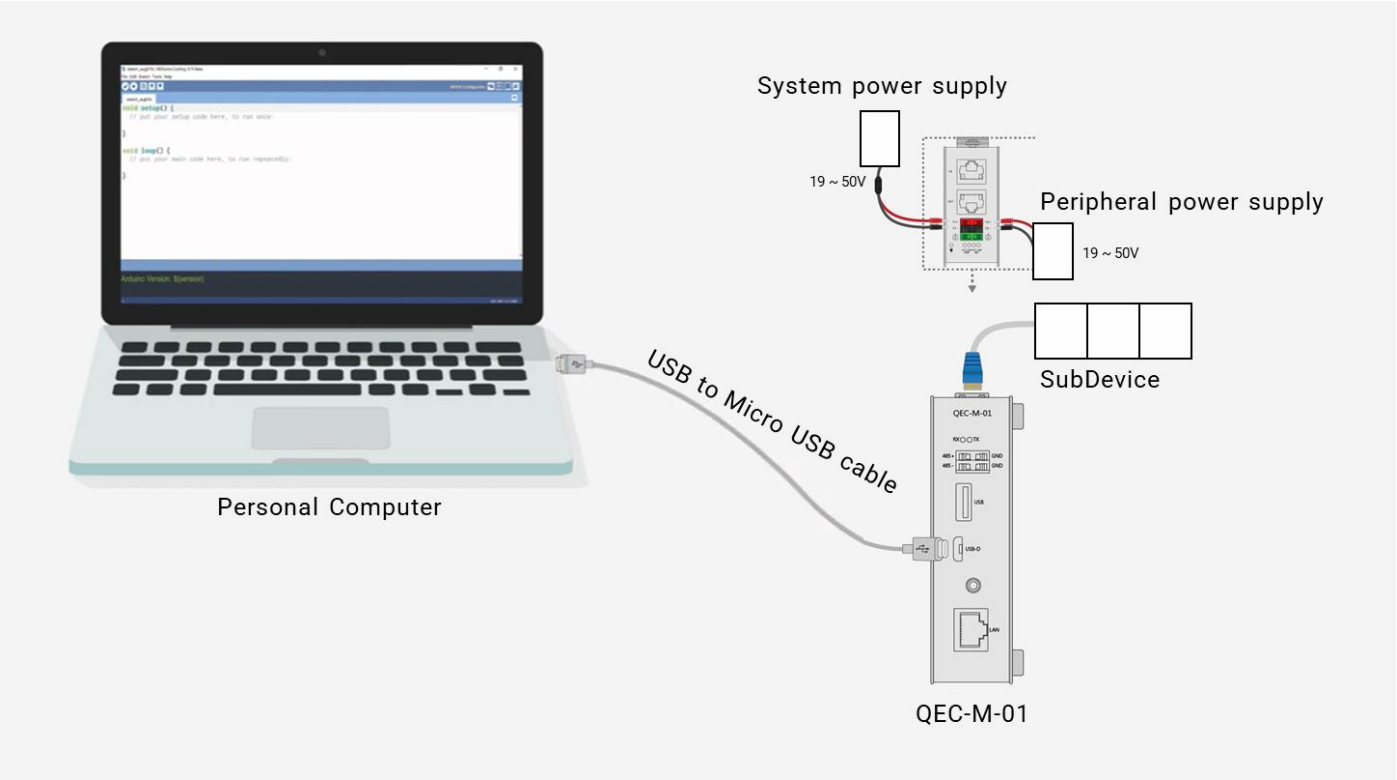
1. QEC-M-01 (EtherCAT MDevice)
2. Ezi-IO (EtherCAT Input/Output Module).
3. 24V power supply & EU-type terminal cable & LAN cable



1.1 QEC-M-01

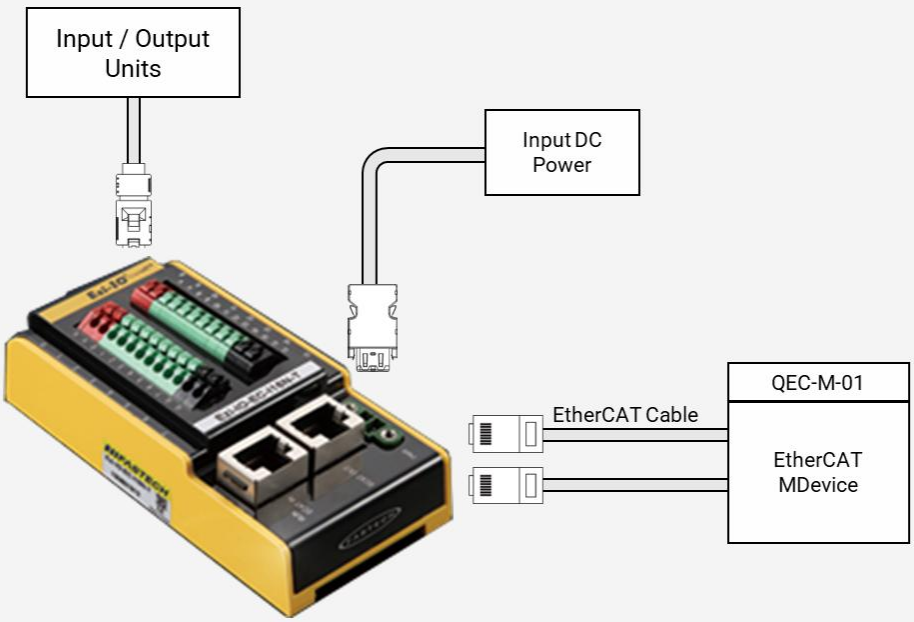
QEC EtherCAT MDevice.

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 Ezi-IO

Ezi-IO is a Fully Digital I/O control Unit from FASTECH mounted with an EtherCAT SubDevice Controller chip. This figure shows an example of an Ezi-IO connection.

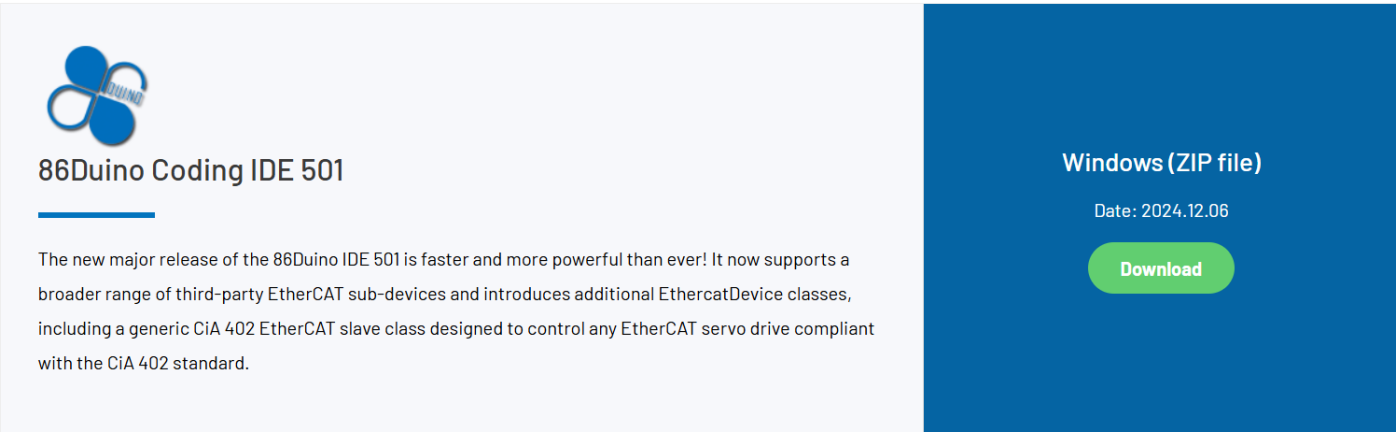


Ezi-IO

1. EtherCAT Connectivity
   - (EtherCAT Cable) Two RJ45 EtherCAT ports for real-time network communication.
   - Connects to the QEC-M-01 EtherCAT MDevice.
2. I/O Signal Connection
   - (Top Connector) Used to connect external input and output units such as sensors, switches, or relays.
3. Power Supply
   - (24V DC Input) Supplies operating power to the module through the power connector.

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.

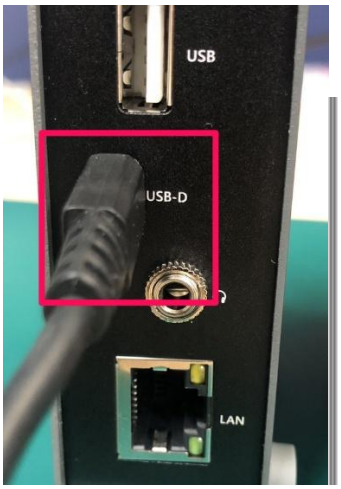
Note:
If Windows displays a warning, click Details once and then click the Continue Run button once.

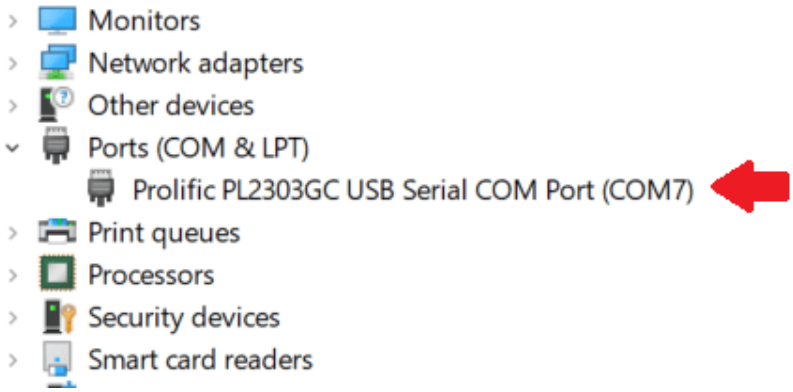
86Duino Coding IDE 501+ looks like below.

3. Connect to PC and set up the environment

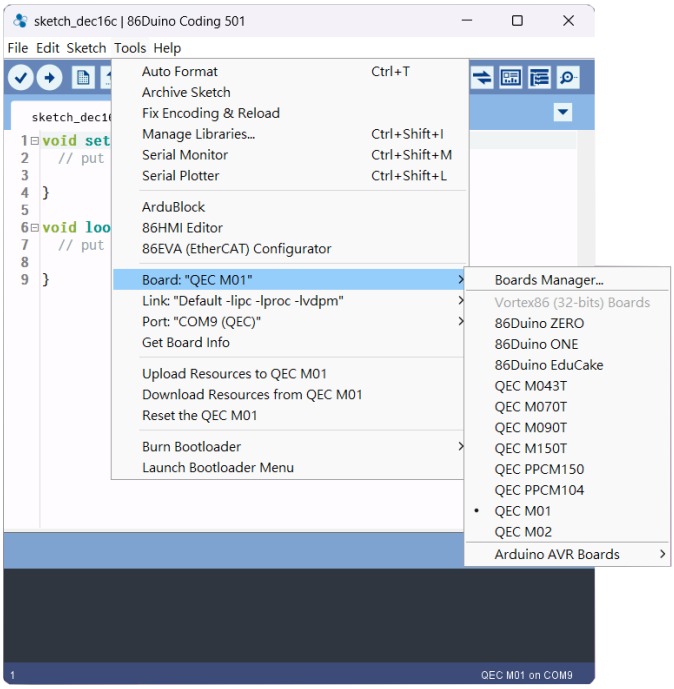
Follow the steps below to set up the environment:

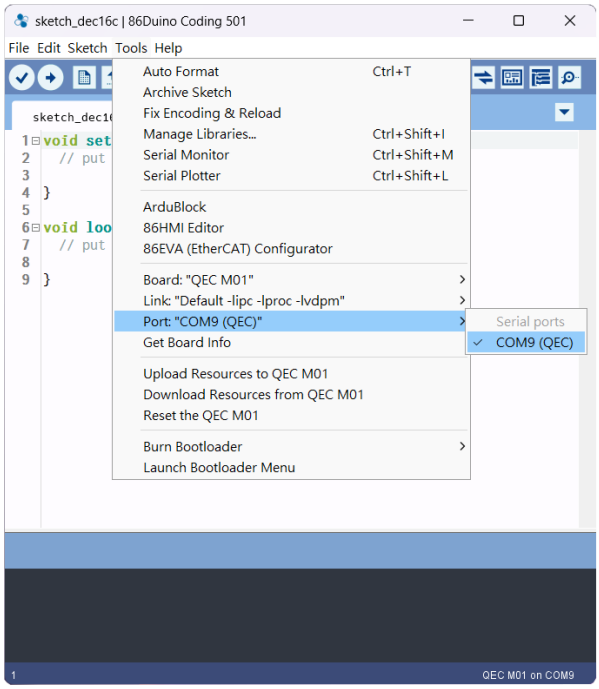
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “Device Manager” (select in the menu after pressing Win+X) ->” Ports (COM & LPT)” in your PC and expand the ports; you should see that the “Prolific PL2303GC USB Serial COM Port (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-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 ArduBlock

This example shows how to operate the EtherCAT MDevice (QEC-M-01) and the Ezi-IO (EtherCAT Input/Output Module) through the 86Duino IDE's graphical low-code programming tool, 86EVA and ArduBlock.

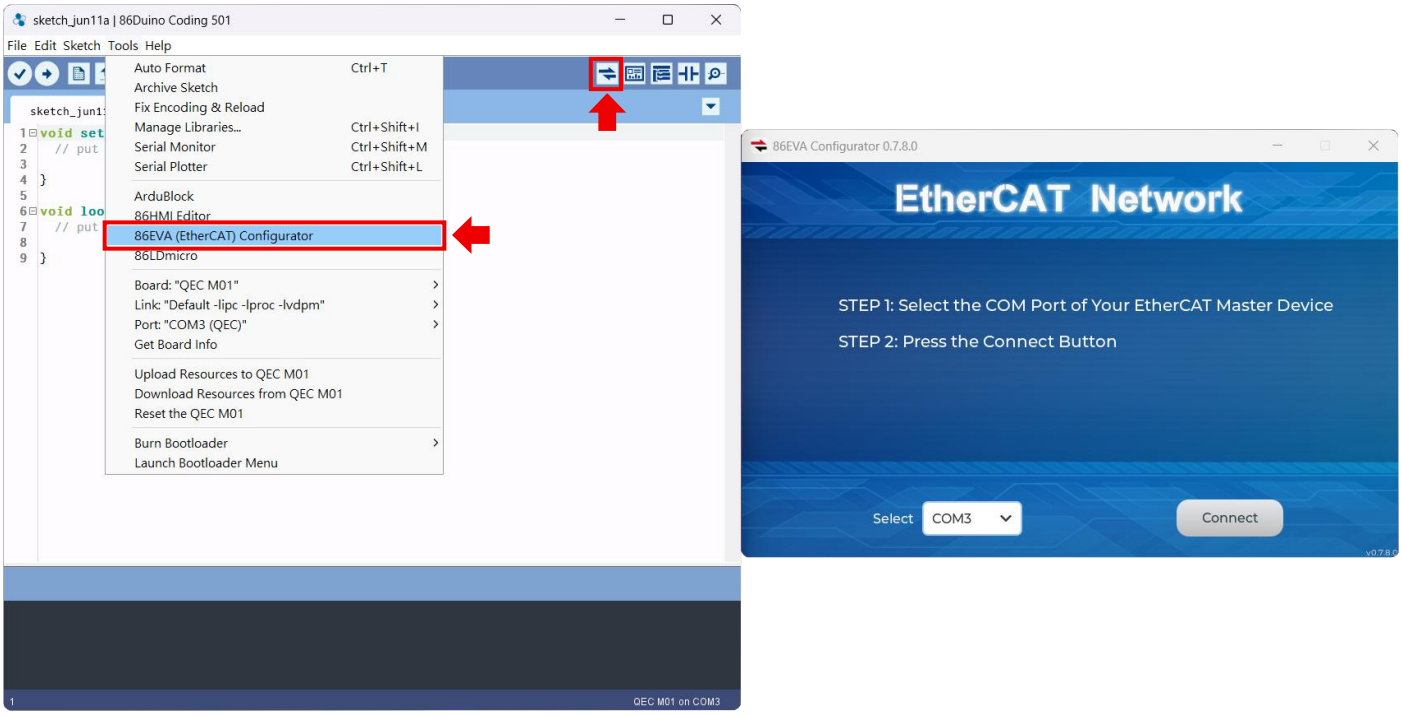
Software Tools Description:

- **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.

In this example, we connect the Ezi-IO's Digital Output channel 0 to Digital Input channel 7, DO ch1 to DI ch6, DO ch2 to DI ch5, and so on.

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.

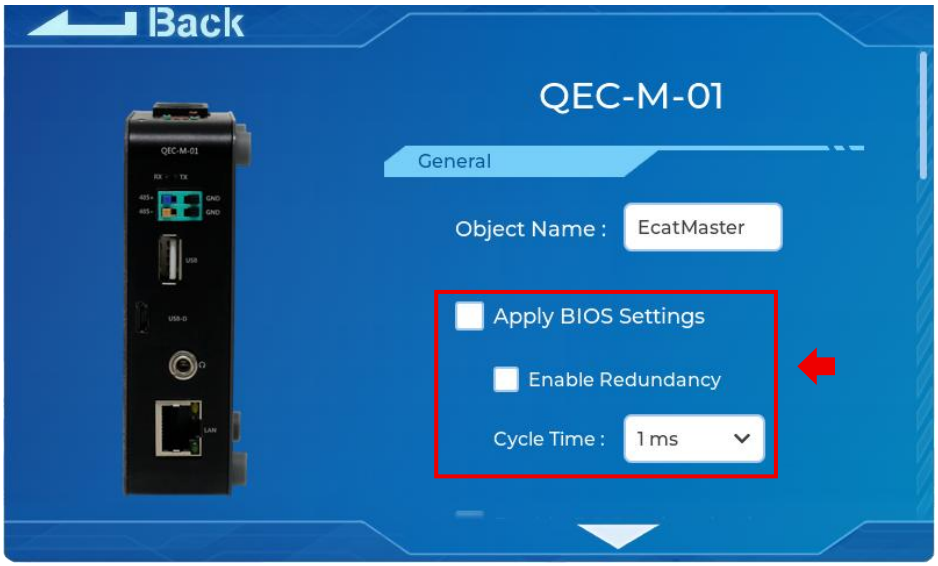
The connected devices will be displayed after the EtherCAT network has been scanned.

Step 2: Set the parameters

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

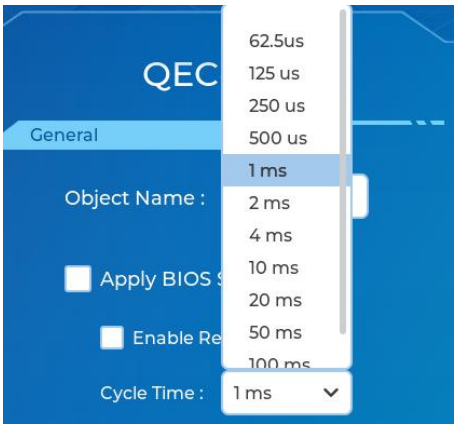
QEC-M-01

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.



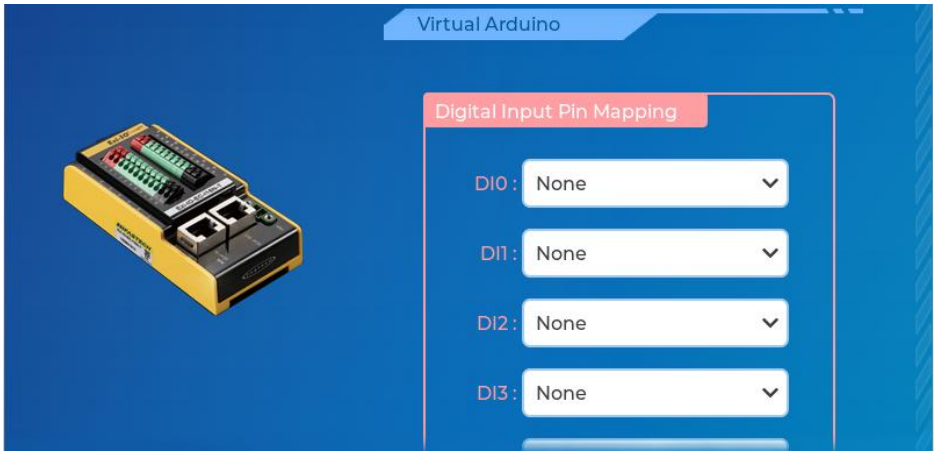
Click "**Back**" in the upper left corner to return.

Ezi-IO

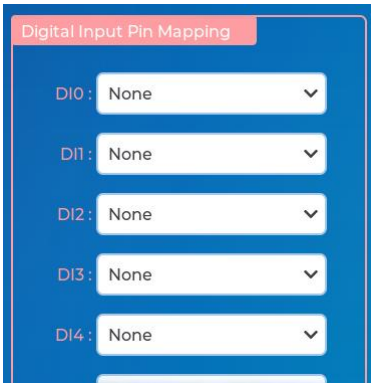
Press twice on the image of the Ezi-IO module to see the parameter settings.

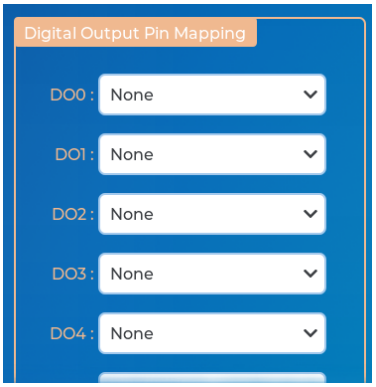
The page will show the Object Name, Alias Address, Vendor ID, Product Code, Virtual Arduino Mapping, and Virtual Servo Configuration parameters.

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

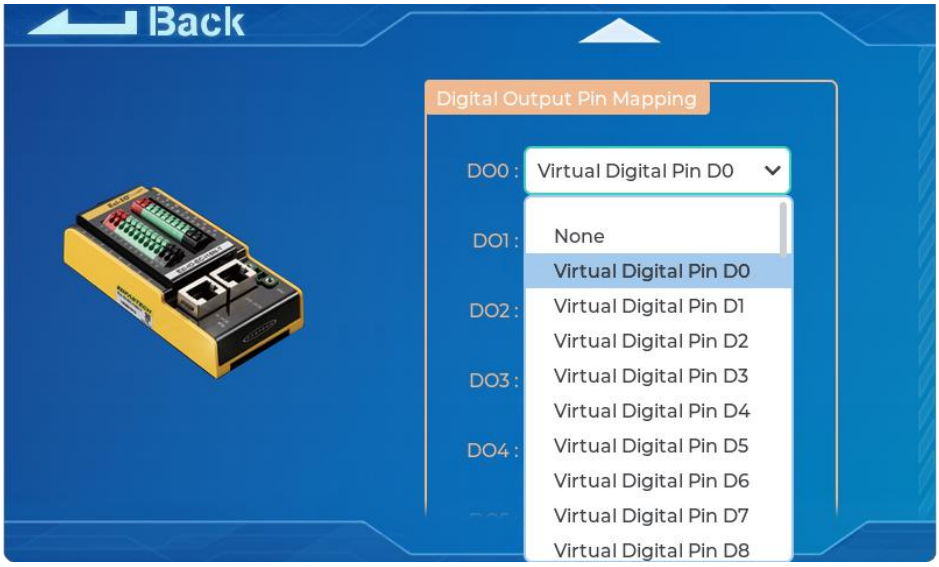


There are two sections for the Ezi-IO, "**Digital Input Pin Mapping**" and "**Digital Output Pin Mapping**", which correspond to the 8-channel digital input and 8-channel digital output on the Ezi-IO module.

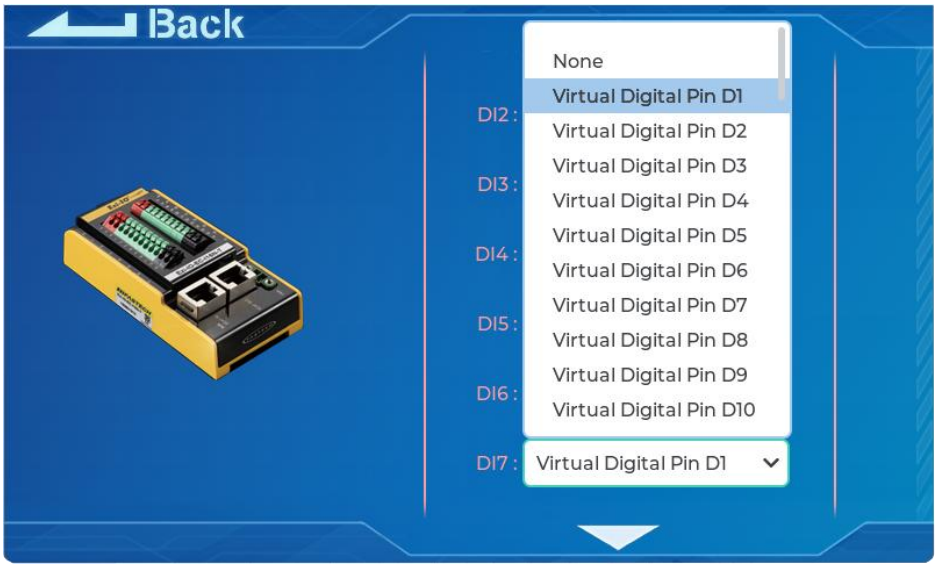




We select "Virtual Digital Pin D0" in the drop-down box of "**DO0**" in the "**Digital Output Pin Mapping**" area.



Then, we select “Virtual Digital Pin D1” in the drop-down box of “DI7” in the “**Digital Input Pin Mapping**” area.

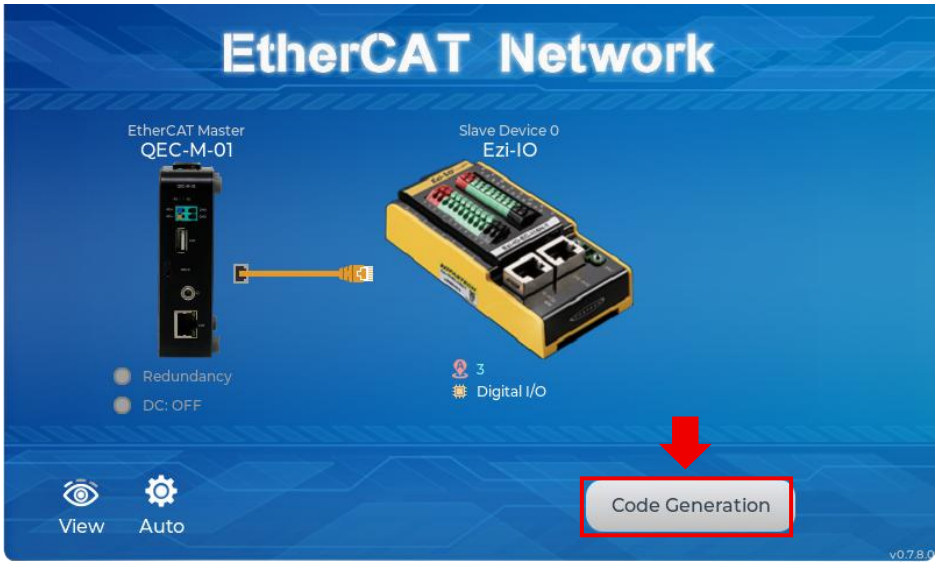


This setting is for ArduBlock tool configuration.

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

Step 3: Generate the code

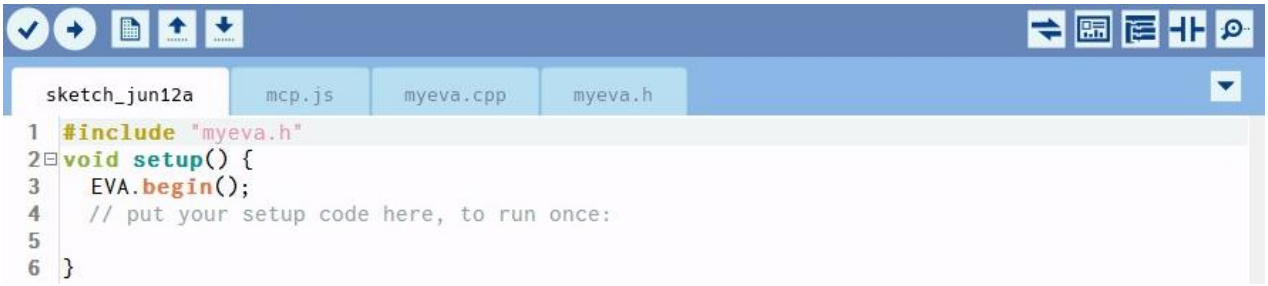
Once you've set your device's parameters, go back to the home screen and press the "**Code Generation**" button in the bottom right corner.



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:

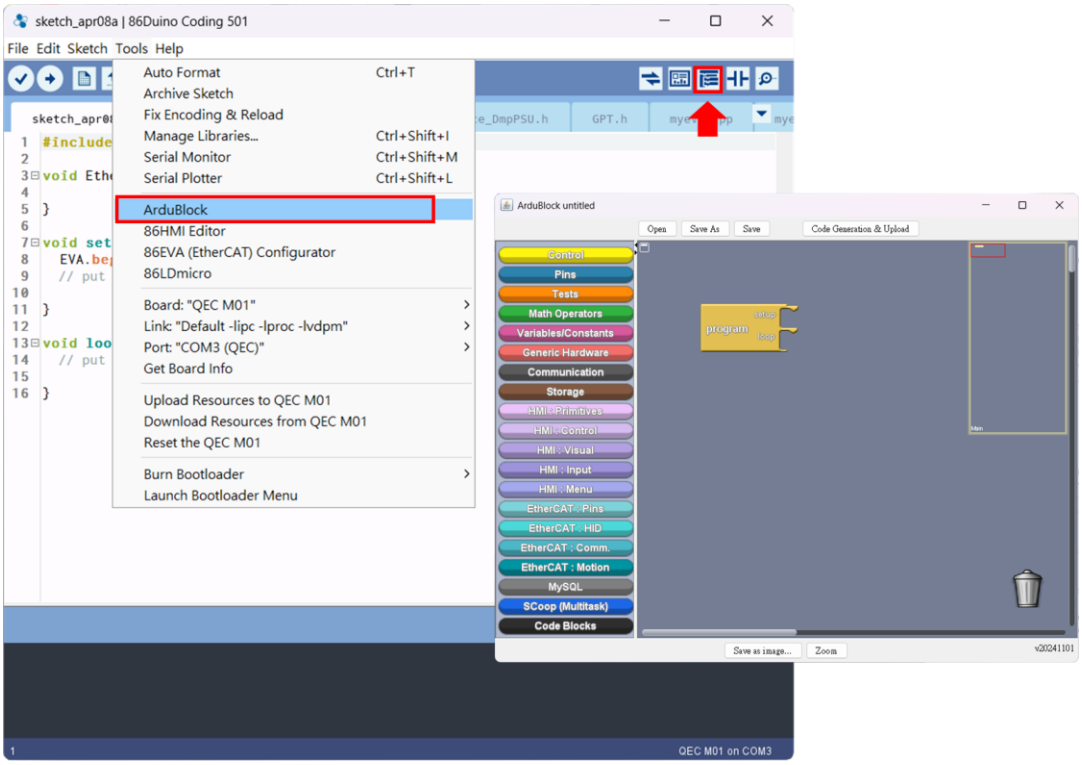
- Ezi-IO: EthercatDevice_Generic object.
- EtherCAT mode: ECAT_SYNC.

And here is the setting by users:

- EtherCAT Cycle time: 1 millisecond.
- Device Object Name: QEC-M-01 is “**EcatMaster**”, and Ezi-IO is “**Slave0**”.

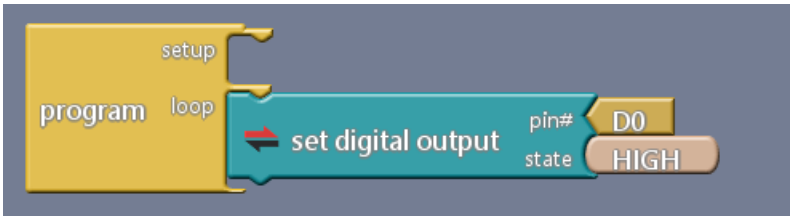
Step 4: Turn on ArduBlock and setup

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

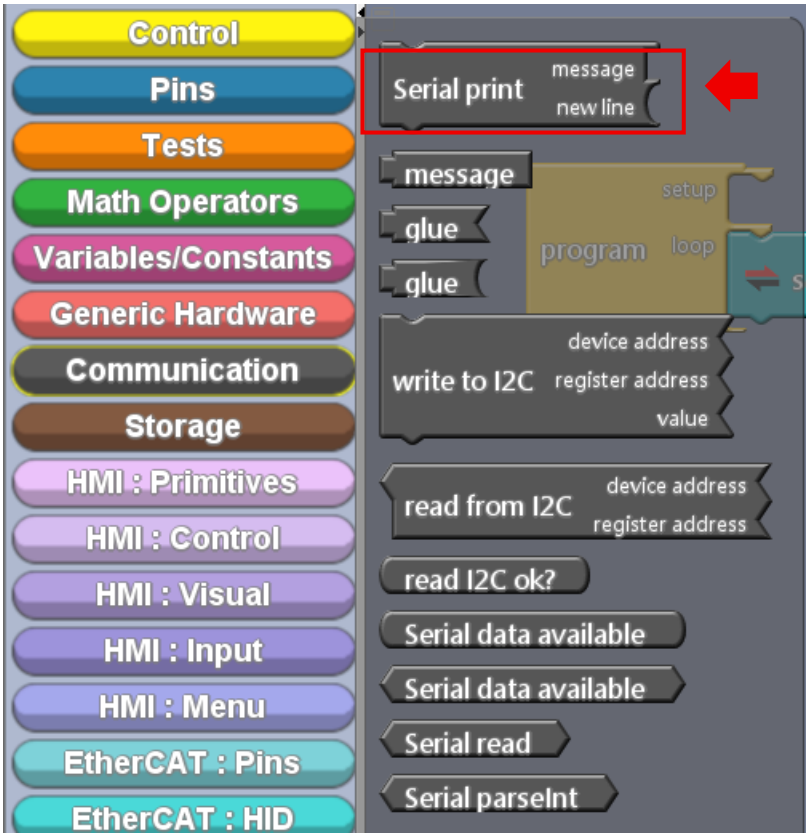


Since all configurations are done by 86EVA, we start with loop control.
First, we put the "**set digital output**" block from the "**EtherCAT: Pins**" class.

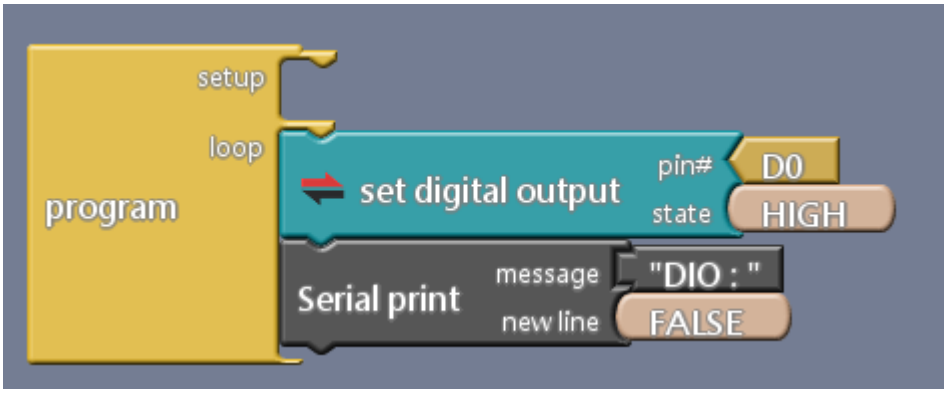
Then, we set pin# to "**D0**", which corresponds to the 86EVA "**Virtual Arduino**" configuration. And set state to "**HIGH**".



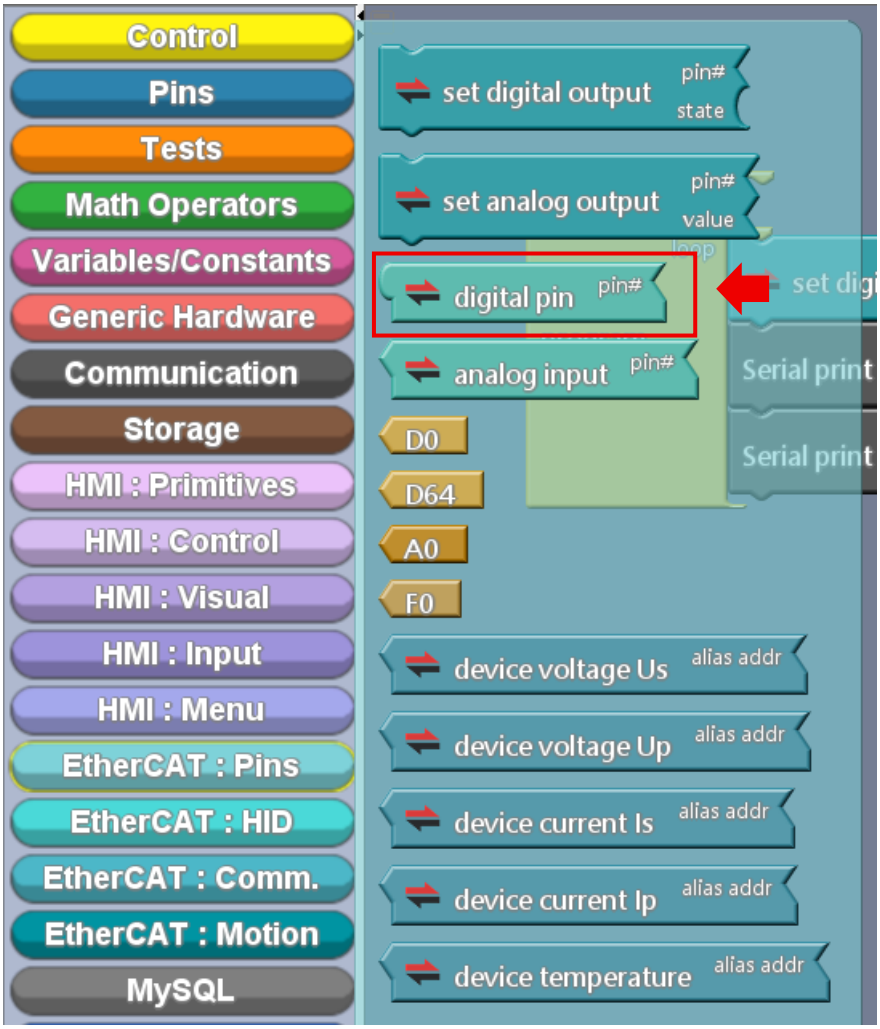
Next, we need to print the digital input state of the DI7 out to the Serial monitor. We put
We put two of the "**Serial print**" blocks from the "**Communication**" class.



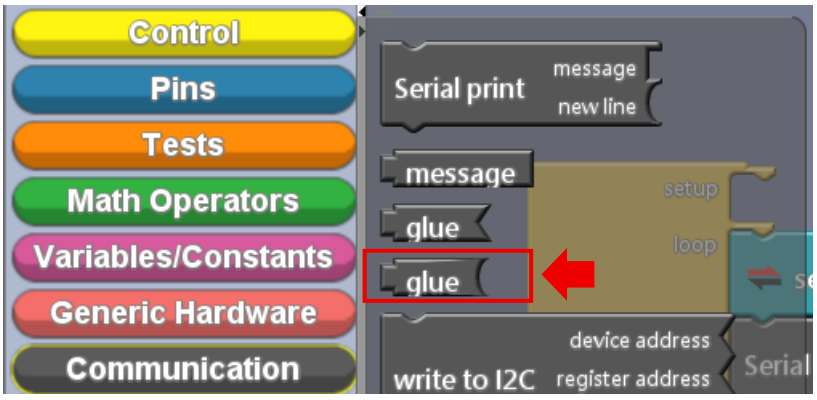
We set the first "**Serial print**" block's message to "**DIO :**", and new line to "**FALSE**".



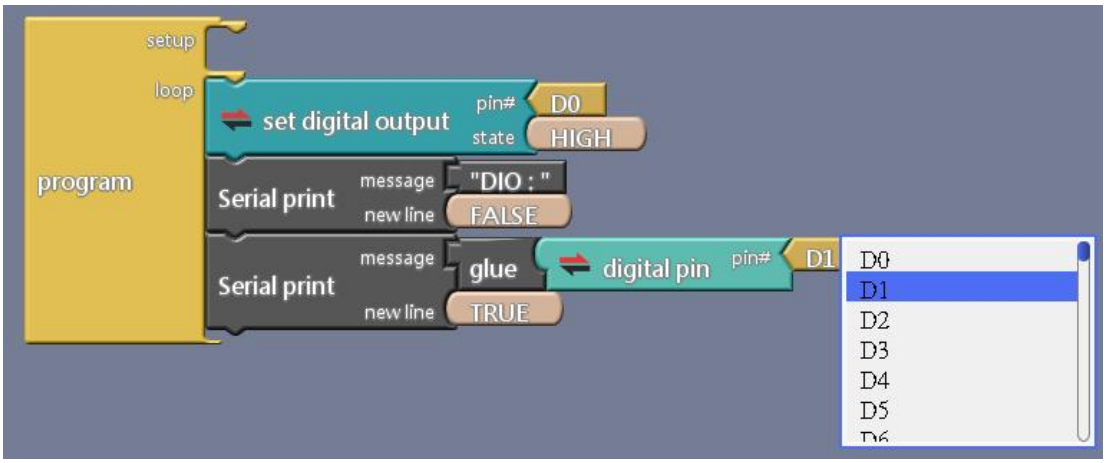
And we let the digital input state print out by using the second “**Serial print**” block.
We put the “**digital pin**” block from the “**EtherCAT : Pins**” class.



And use the “**glue**” block from the “**Communication**” class to glue the different data types.

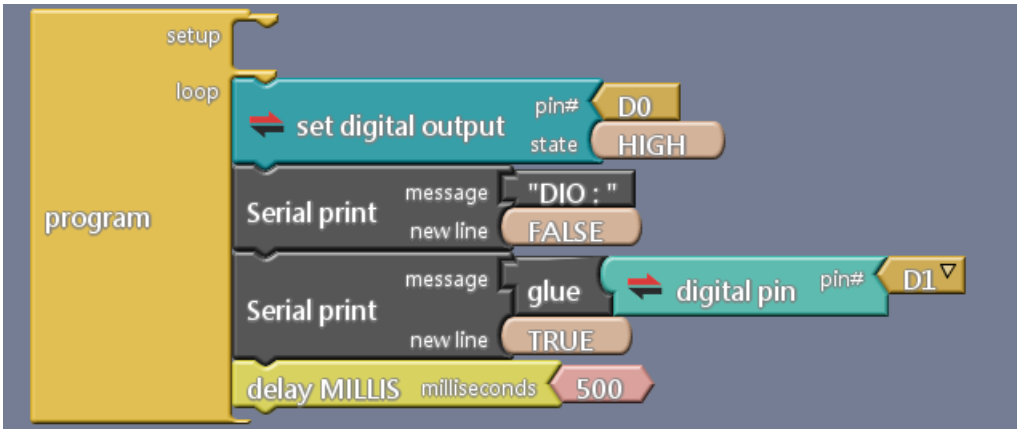


Like this.

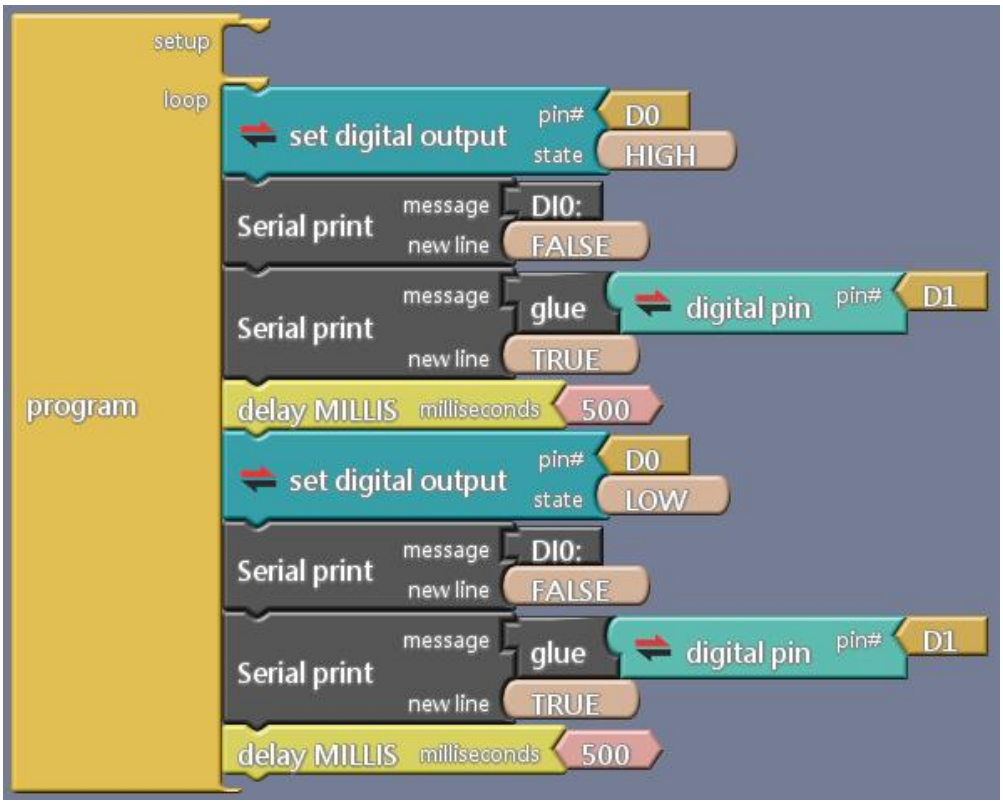


We set pin# to "**D1**", which we configure it into the DI7.

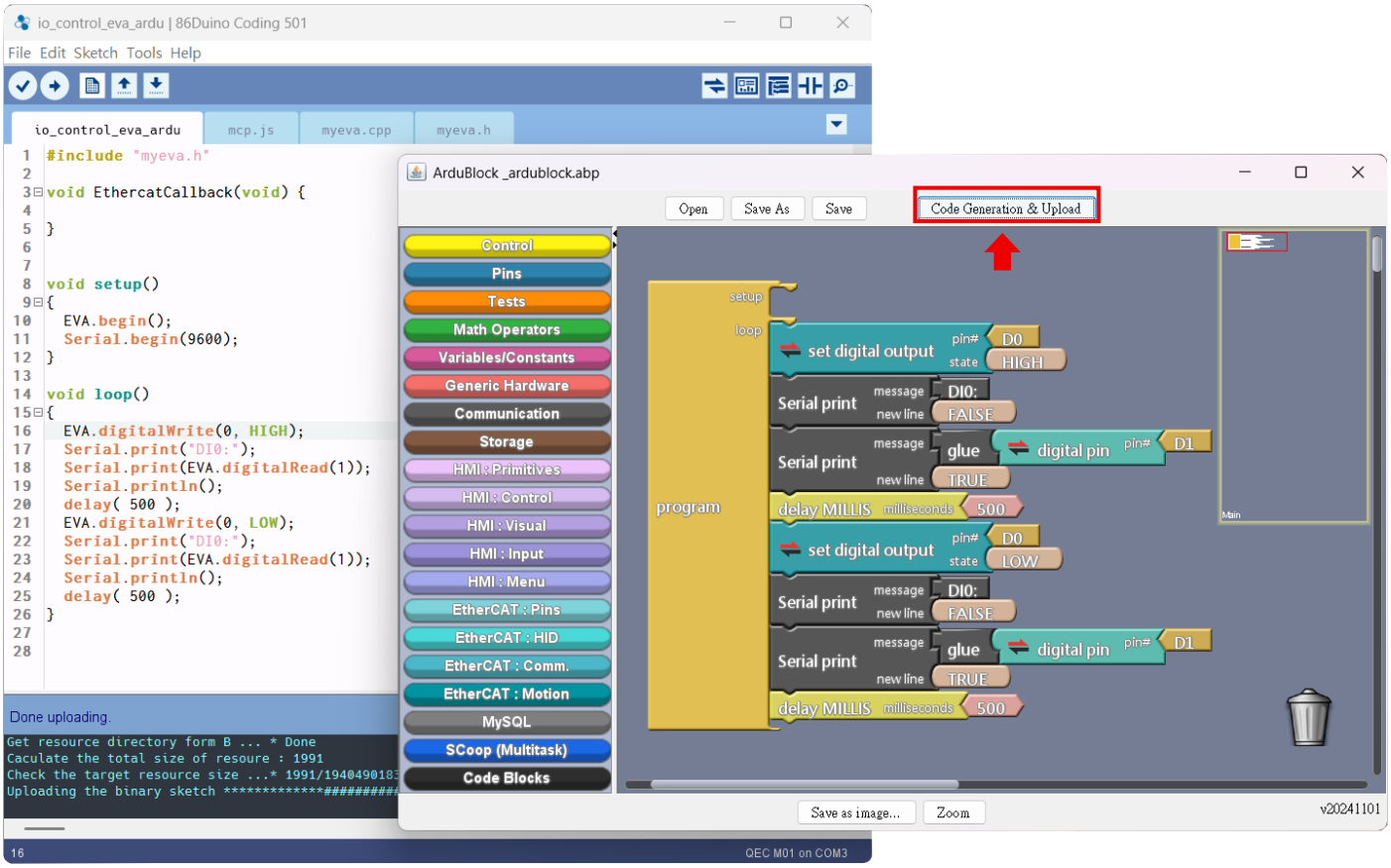
Next, we put a "**delay MILLIS**" block from "**Control**" class to wait 500 ms.



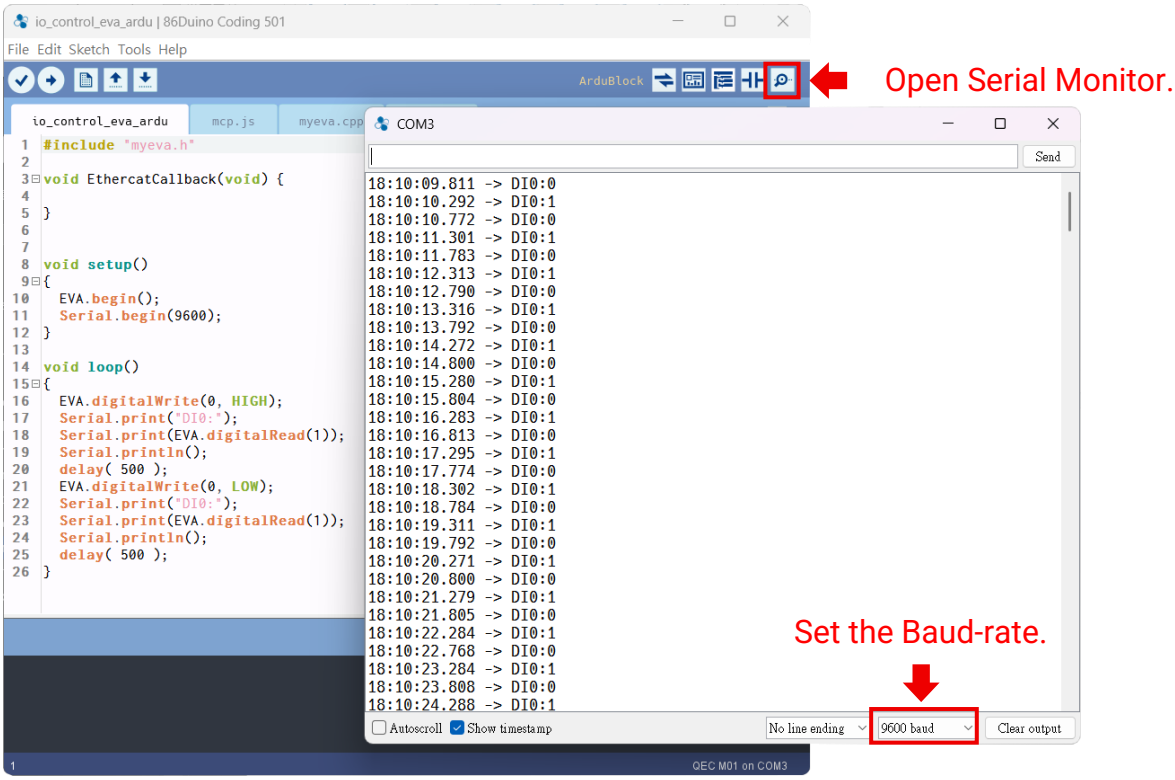
After the delay, we do the same tasks, only changing the "**set digital output**" state to "**LOW**".



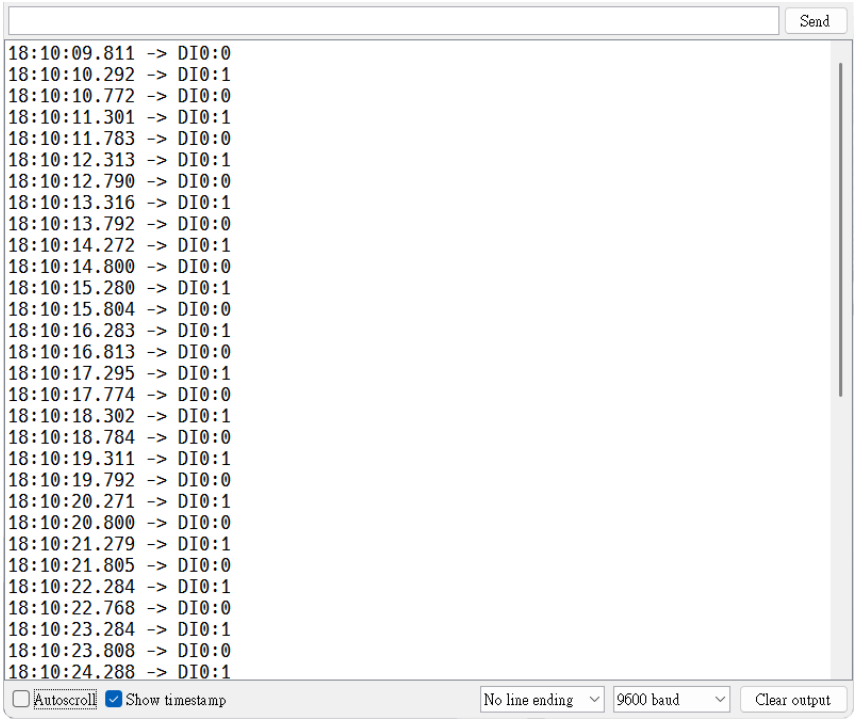
After finishing, you can click the "**Code Generate & Upload**" button to generate and upload the code to the QEC-M-01.



After you successfully upload the program to the QEC-M-01, you can open the Serial Monitor (Baud-rate: 9600) on 86Duino IDE. Please check the Serial baud rate is same as your setting.



If the EtherCAT communication config successful, Serial Monitor will print the DI ch7 state 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:

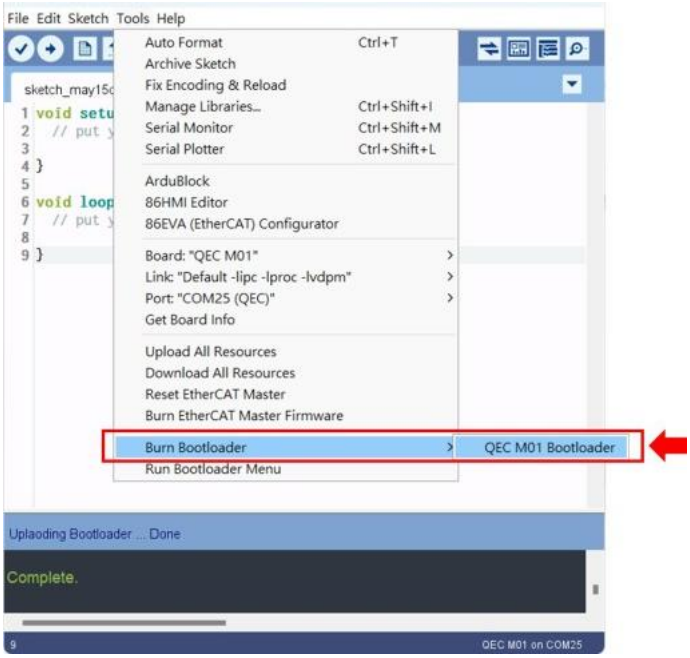
Step 1: Setting up QEC-M

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.

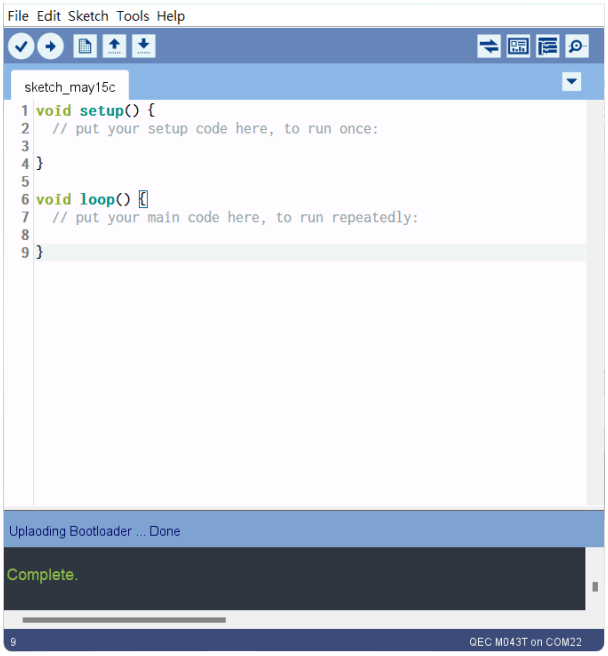
Step 2: Click “Burn Bootloader” button

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.