Start Guide

CM-TE66NA3G-N + CM-TE66PA3G-P EtherCAT DI/DO with 86EVA and ArduBlock

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

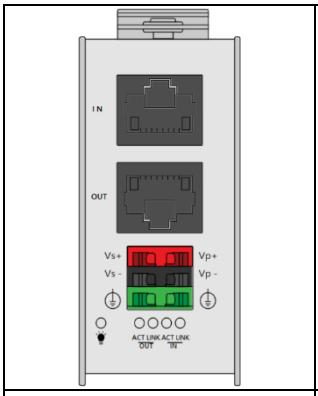
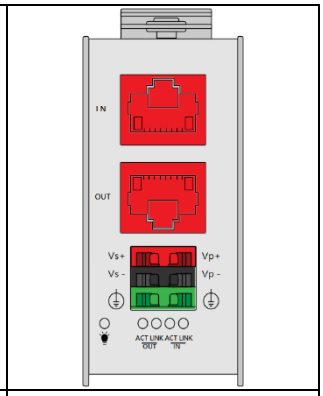
Date	Version	Description
2026/3/27	Version1.0	New release.

Preface

In this guide, we will show you how to use the EtherCAT MDevice **QEC-M-01**, the **CM-TE66NA3G-N** (EtherCAT 32 DO) and the **CM-TE66PA3G-P** (EtherCAT 32 DI).

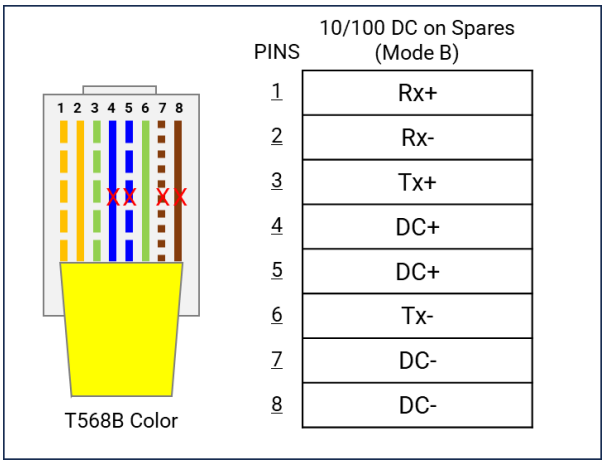
Notes QEC's PoE (Power over Ethernet)

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:



PINS	10/100 DC on Spares (Mode B)
1	Rx+
2	Rx-
3	Tx+
4	DC+
5	DC+
6	Tx-
7	DC-
8	DC-

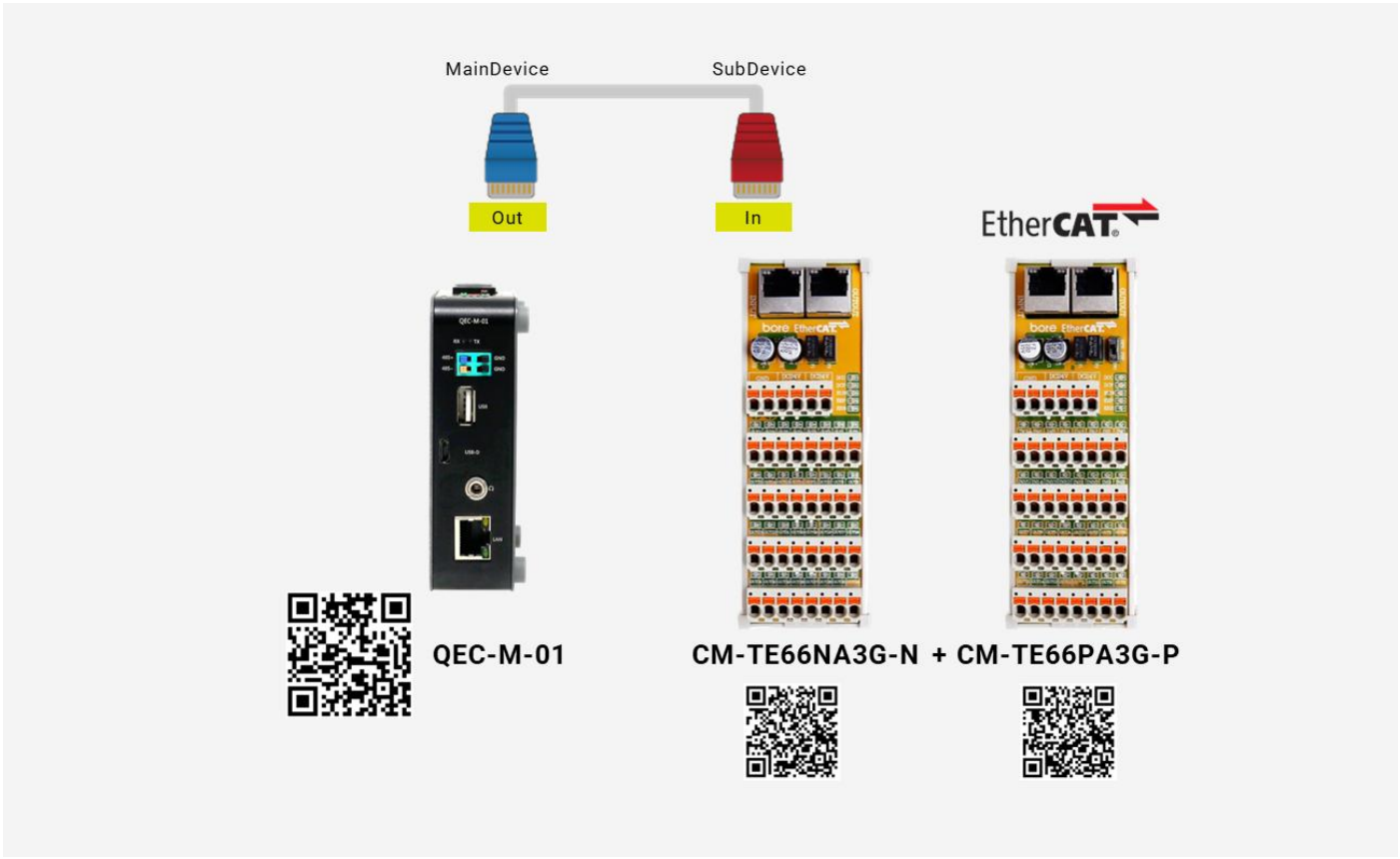
2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).

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

1. Connection and wiring hardware

The following devices are used here:

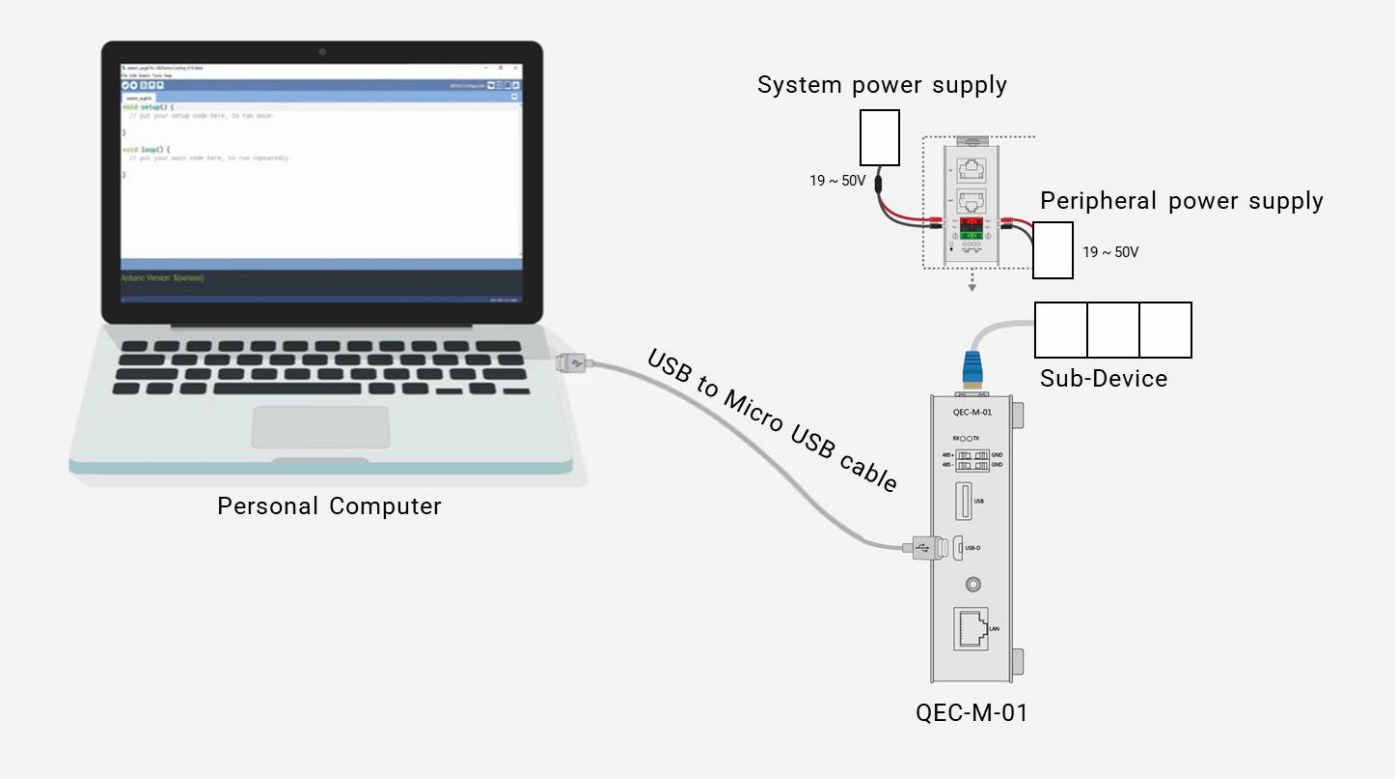
1. QEC-M-01 (EtherCAT MDevice)
2. Bore CM-TE66NA3G-N (EtherCAT 32 DO)
3. Bore CM-TE66PA3G-P (EtherCAT 32 DI)
4. 24VDC power supply & EU-type terminal cable & LAN cable



1.1 QEC-M-01

QEC EtherCAT MDevice, **QEC-M-01**.

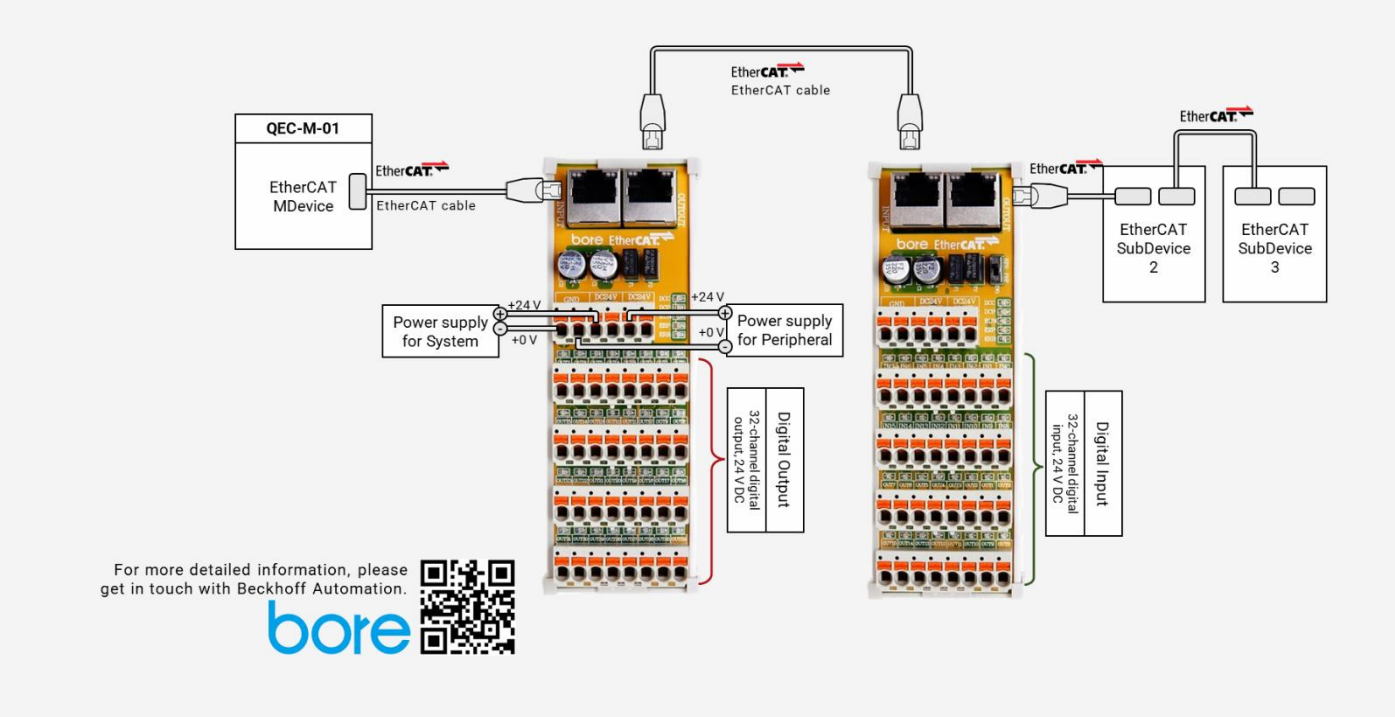
1. Power Supply: Connect to Vs+/- and Vp+/- power supplies via EU terminals for 24V power.
2. EtherCAT Connection: Using the EtherCAT Out port (On the top side) connected to the EtherCAT In port of EtherCAT SubDevice via RJ45 cable.



1.2 Bore 32 DI/DO

The **Bore CM-TE66NA3G-N** (EtherCAT 32 DO) and the **Bore CM-TE66PA3G-P** (EtherCAT 32 DI) are 32-point EtherCAT digital I/O modules. They can be used as EtherCAT SubDevices in a QEC EtherCAT network for basic digital input sensing and output control.

The diagram below shows a typical wiring example using a **QEC MDevice** (for example, **QEC-M-01**) together with **Bore EtherCAT DI/DO modules**.

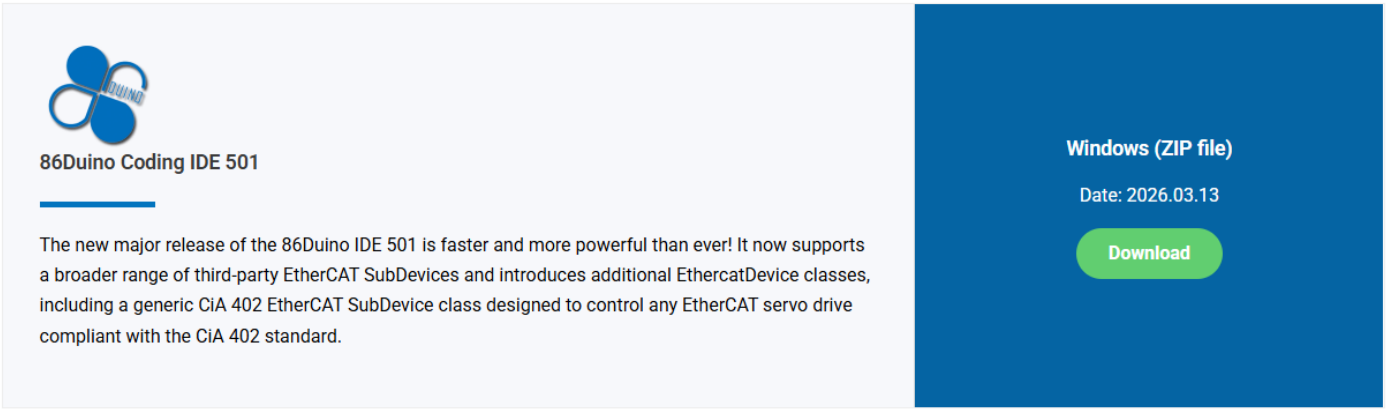


Connections are grouped by function:

- EtherCAT
  - IN: Connect to the EtherCAT port of the QEC MDevice or the previous SubDevice.
  - OUT: Connect to the next EtherCAT SubDevice, if any.
- Power & Grounding
  - Provide 24 VDC power for module operation and field I/O.
  - Connect 0 V (GND) properly to ensure stable communication.
- Digital Inputs (CM-TE66PA3G-P, DI)
  - Input channels receive external digital signals from field devices.
  - Input signals should be referenced to the module input common or 0 V (GND).
- Digital Outputs (CM-TE66NA3G-N, DO)
  - Output channels are used to drive external loads.
  - A suitable external 24 VDC power supply is required for output operation.
  - Ensure the load wiring matches the module's output type and electrical rating.
- Indicators / LEDs
  - EtherCAT LEDs indicate power status and EtherCAT communication activity.
  - Channel LEDs indicate the ON/OFF status of each DI or DO point.

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.

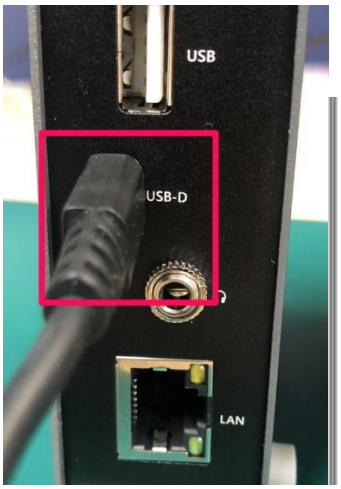
*__Note__: If Windows displays a warning, click Details once and then click the Continue Run button.

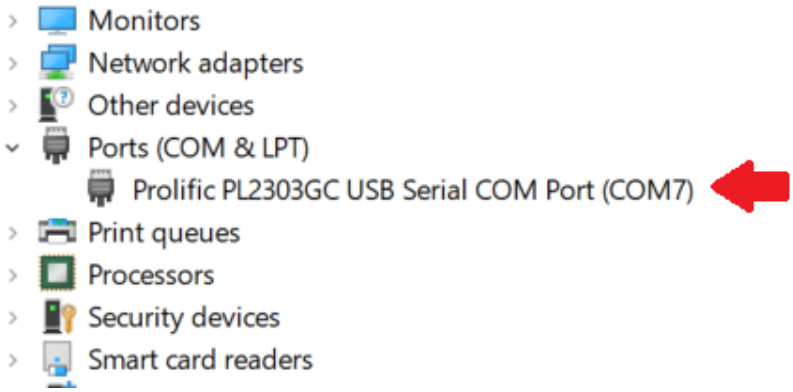
86Duino Coding IDE 501+ looks like below.

3. Connect to PC and set up the environment

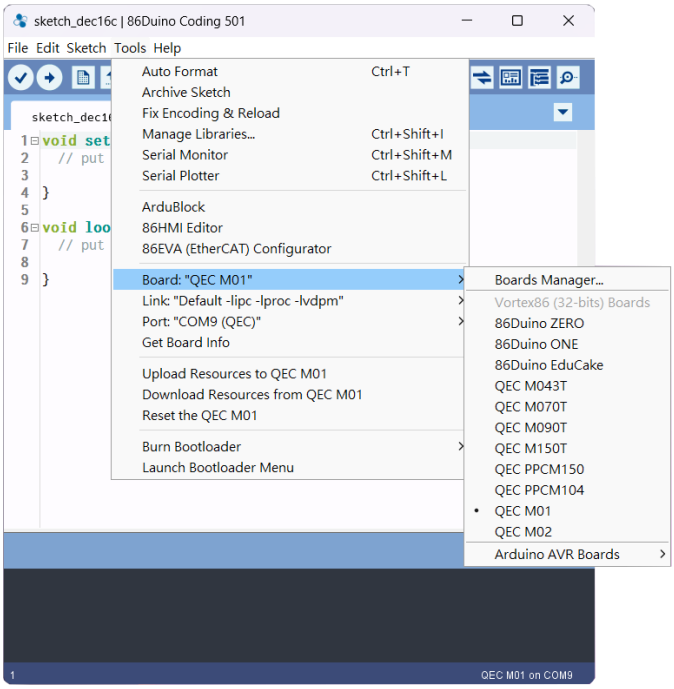
Follow the steps below to set up the environment:

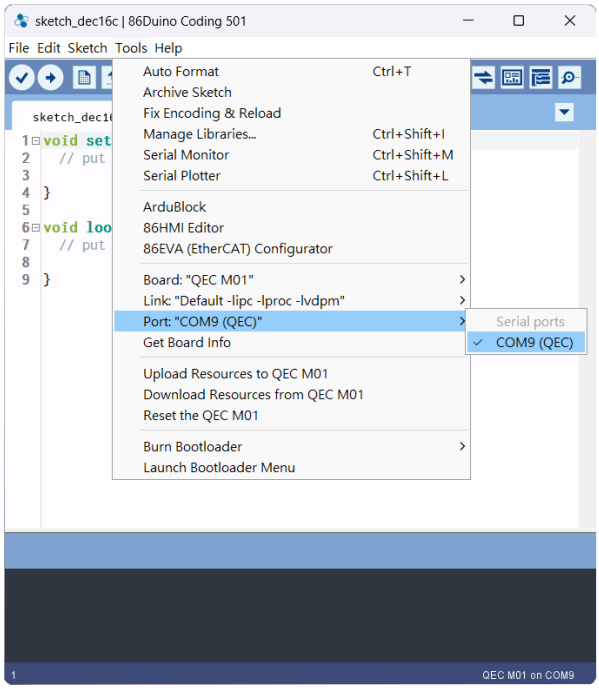
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “**Device Manager**” (select in the menu after pressing Win+X) ->” **Ports (COM & LPT)**” in your PC and expand the ports; you should see that the “**Prolific PL2303GC USB Serial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

The **Bore CM-TE66NA3G-N** (EtherCAT 32 DO) and **CM-TE66PA3G-P** (EtherCAT 32 DI) can be configured and programmed using the EtherCAT library and a graphical low-code programming tool, **86EVA**, in the **86Duino IDE**, together with a **QEC MDevice** such as **QEC-M-01**.

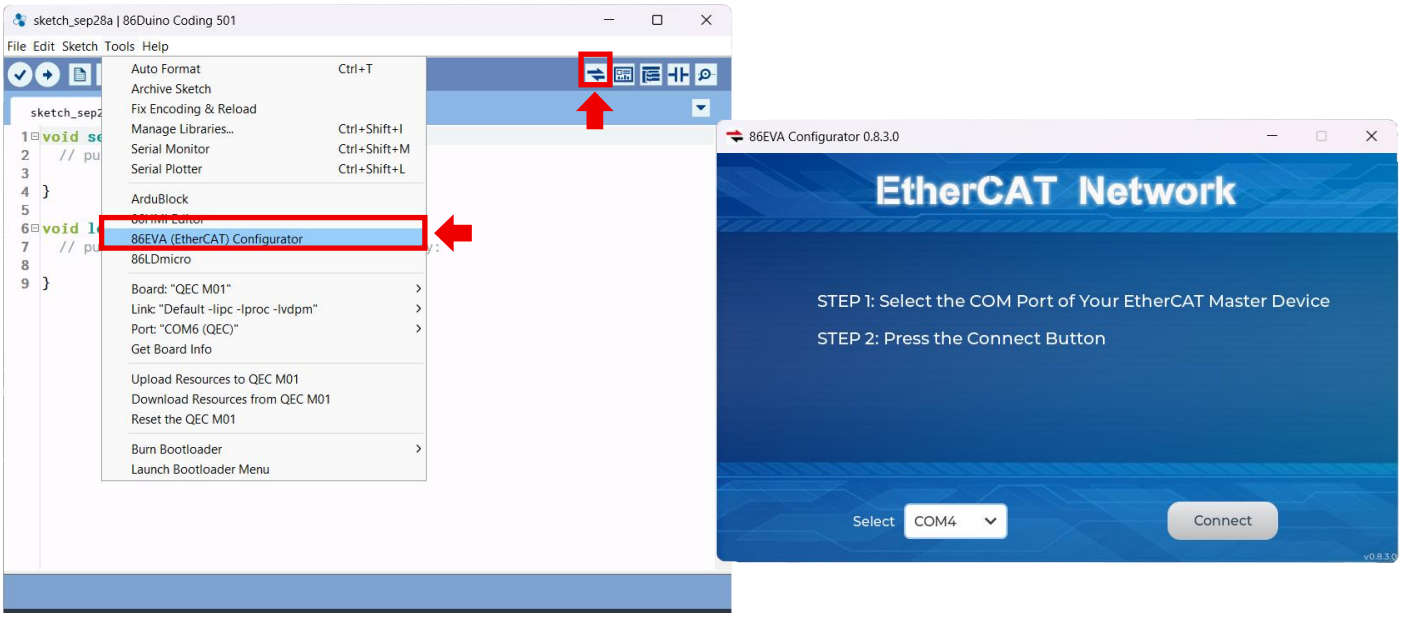
Software Tools Description:

- **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, the program periodically toggles one output bit of the CM-TE66NA3G-N, then reads back one input bit of the CM-TE66PA3G-P after a short delay. This is used to verify that the input state changes correctly in response to the output signal.
For testing, the selected DO channel and DI channel can be connected by external wiring, or the DI channel can be driven by an external input signal.

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

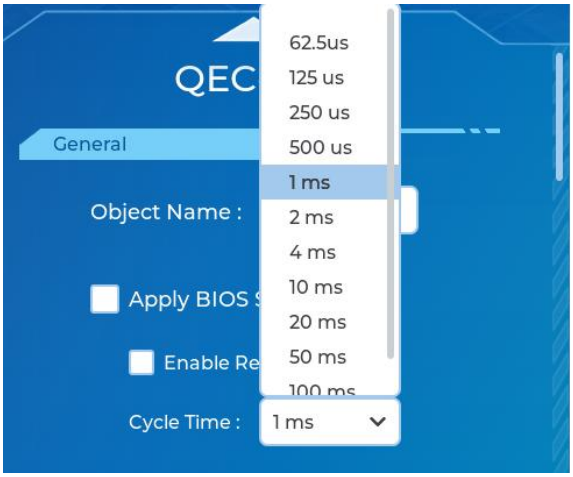
Press on the scanned device image to enter the corresponding parameter setting screen.

Step 2.1: QEC-M-01

Press 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.

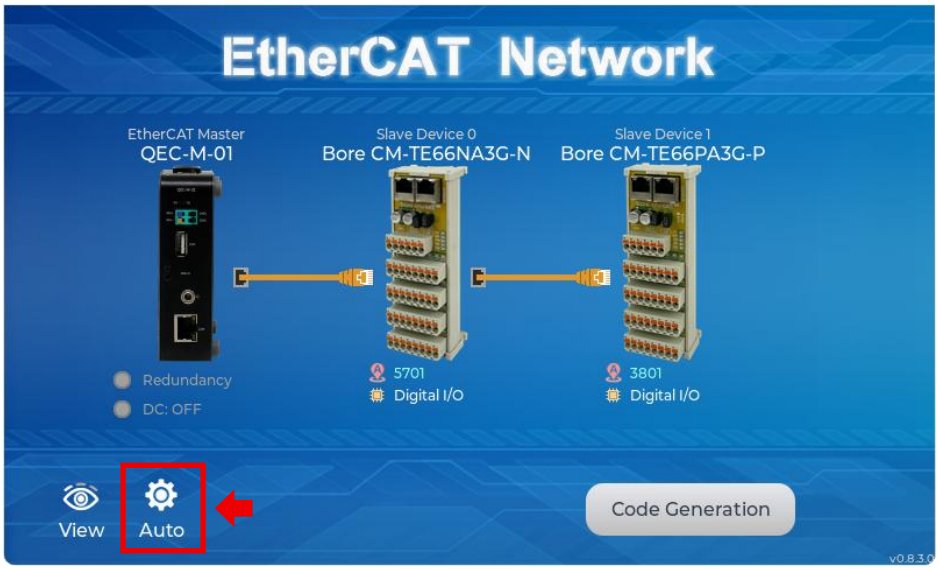


Click "**Back**" in the upper left corner to return.

Step 2.2: Bore CM-TE66NA3G-N & CM-TE66PA3G-P

Before we go to the setting pages, we can use the auto mapping virtual digital pins function, which is located on the bottom left side of the scanned page.

Click the “**Auto**” button.



And click “**Auto Set Digital Pin Mapping of the Virtual Arduino**” button.

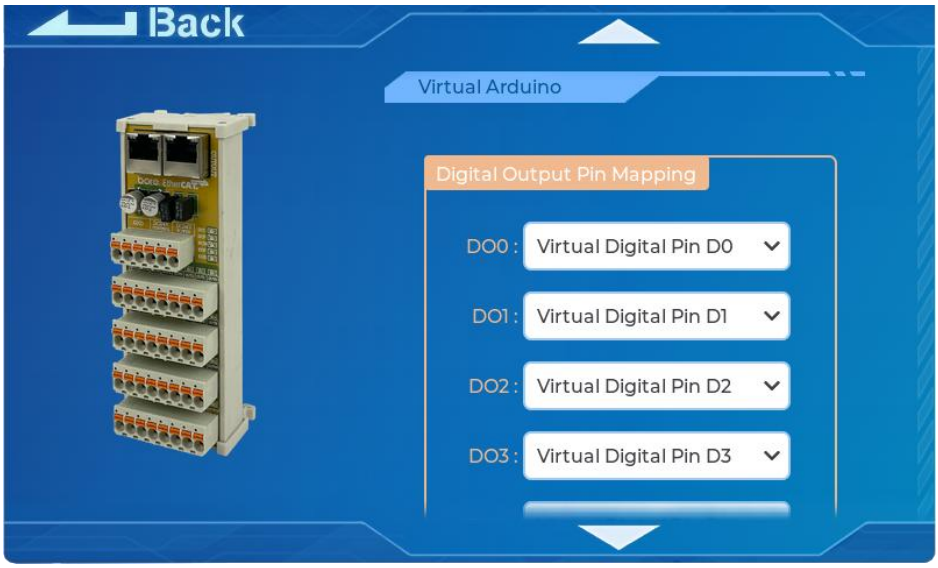
It’ll ask you to confirm, click “**Yes**”.

And it'll show "**[Success]**" to set all virtual Arduino pins for all slave devices.

***Note**: this auto mapping will map the digital output pins first, then the digital input pins.

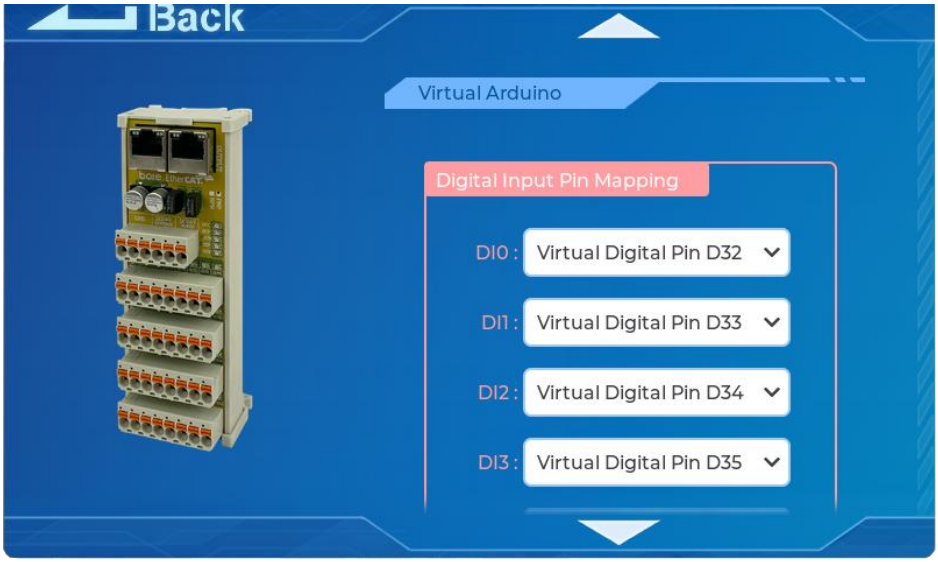
Then, you can press on the image of the Bore CM-TE66NA3G-N to see the parameter settings.

Continue down to the "**Virtual Arduino**" area. It'll complete the virtual digital pins, that DO0 start at "**Virtual Digital Pin D0**".



Next, go to the Bore CM-TE66PA3G-P setting page.

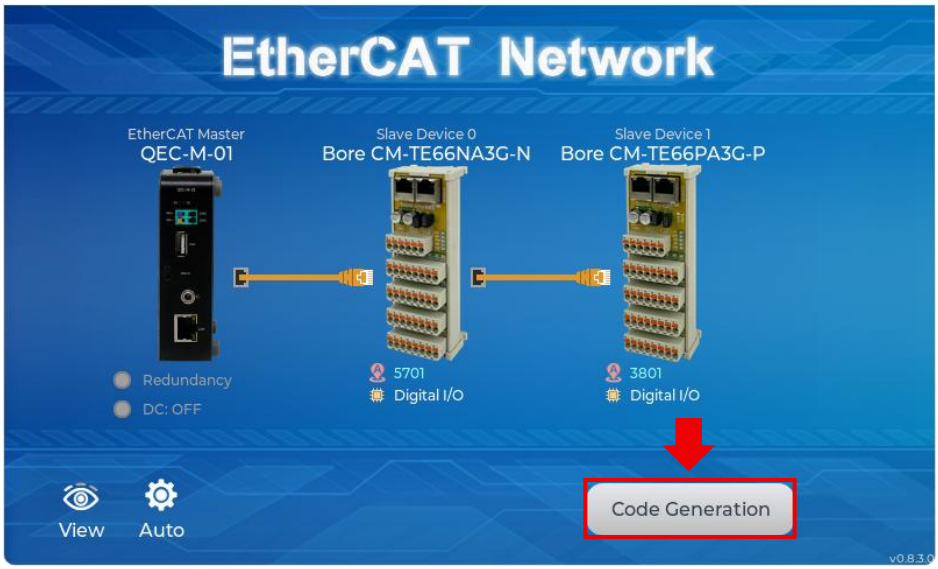
Continue down to the “**Virtual Arduino**” area. It’ll complete the virtual digital pins, that DI0 start at “**Virtual Digital Pin D32**”.



This step is mapping the Virtual Arduino pins for ArduBlock.

Step 3: Generate the code

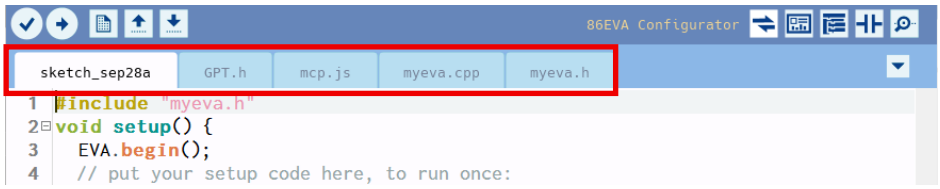
Once you've set your device's parameters, go back to the home screen and press the "**Code Generation**" button in the bottom right corner.



When you're done, 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

Step 4: Turn on ArduBlock and setup

Before operating the EtherCAT network, you must configure it once. The process should be from Pre-OP to OP mode in EtherCAT devices. 86EVA will automatically handle the EtherCAT State Machine in the background.

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

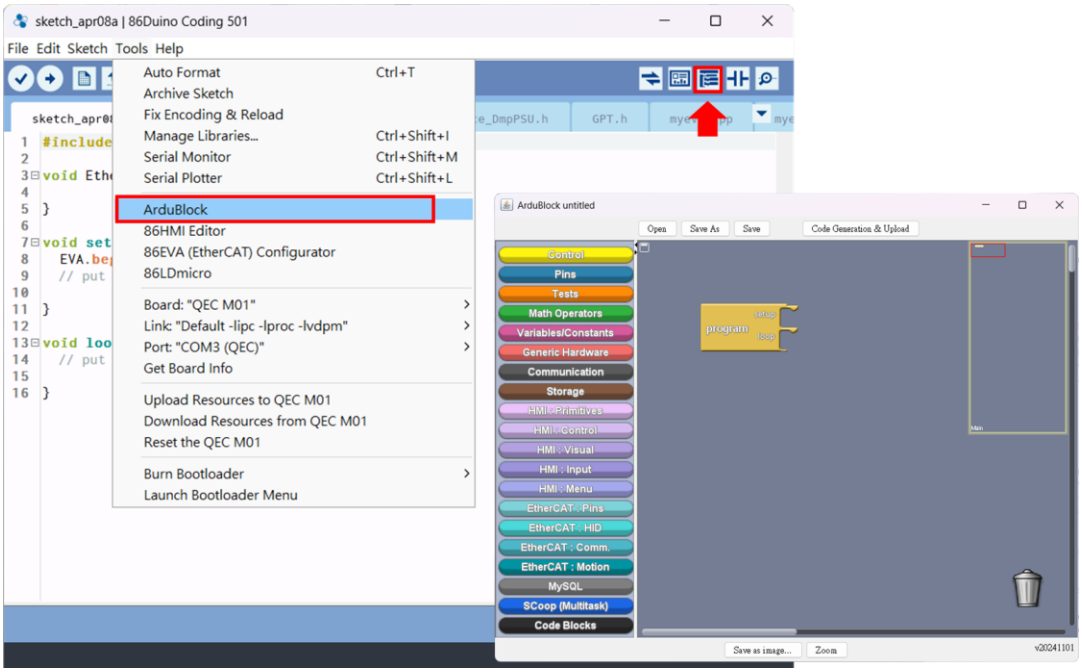
- The Bore CM-TE66NA3G-N (EtherCAT 32 DO) and the CM-TE66PA3G-P (EtherCAT 32 DI) module: `EthercatDevice_Generic` object.
- EtherCAT mode: ECAT_SYNC.

And here is the setting by users:

- EtherCAT Cycle time: 1 milliseconds (Depend on DI/DO filter).
- The `EthercatMaster` object ("`EcatMaster`") represents QEC MDevice: **QEC-M-01**.
- The `EthercatDevice_Generic` objects represent the Bore modules: CM-TE66NA3G-N ("`Slave0`") and CM-TE66PA3G-P ("`Slave1`").

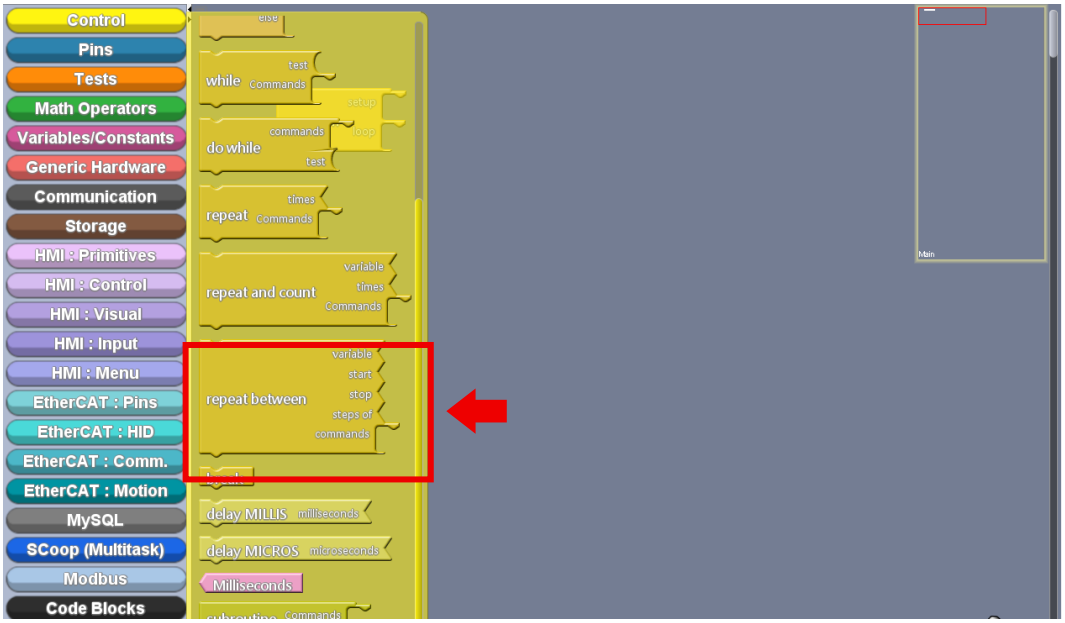
In this example, the program periodically toggles one output bit of the CM-TE66NA3G-N, then reads back one input bit of the CM-TE66PA3G-P after a short delay.

First, we open ArduBlock.



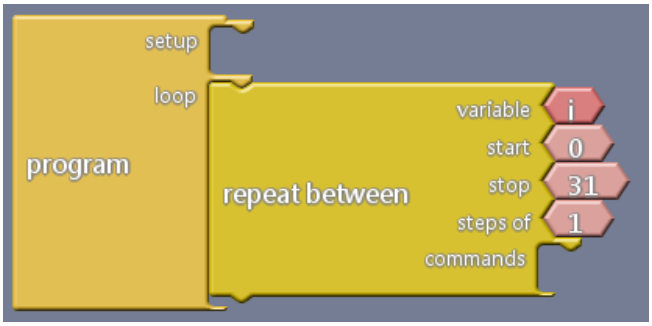
We'll use **for loop block** in this example.

We use the "**repeat between**" block from the "**Control**" class to the "**loop**" area of "**program**" block.

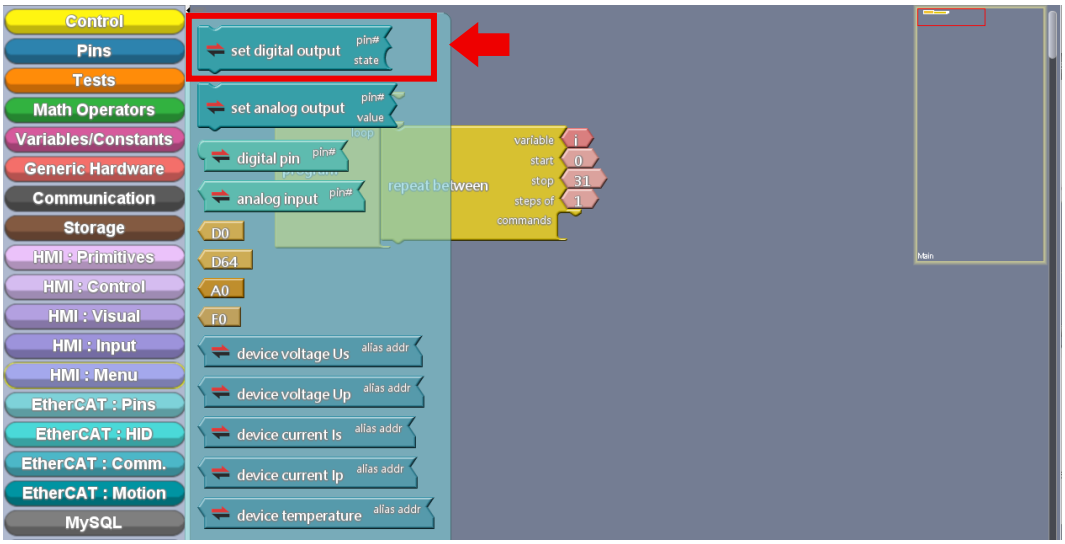


We set variable to "**i**", start from "**0**", stop to "**31**", and steps of "**1**".

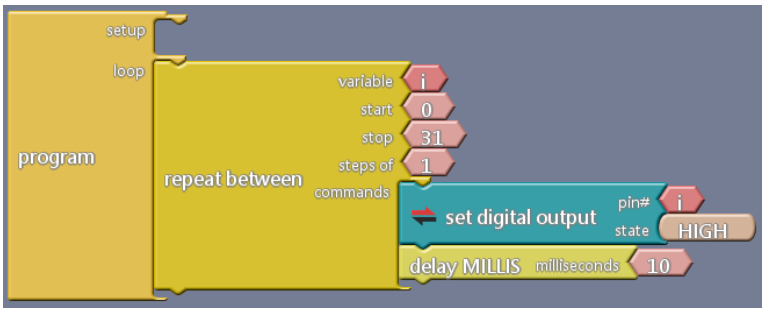
Like this.



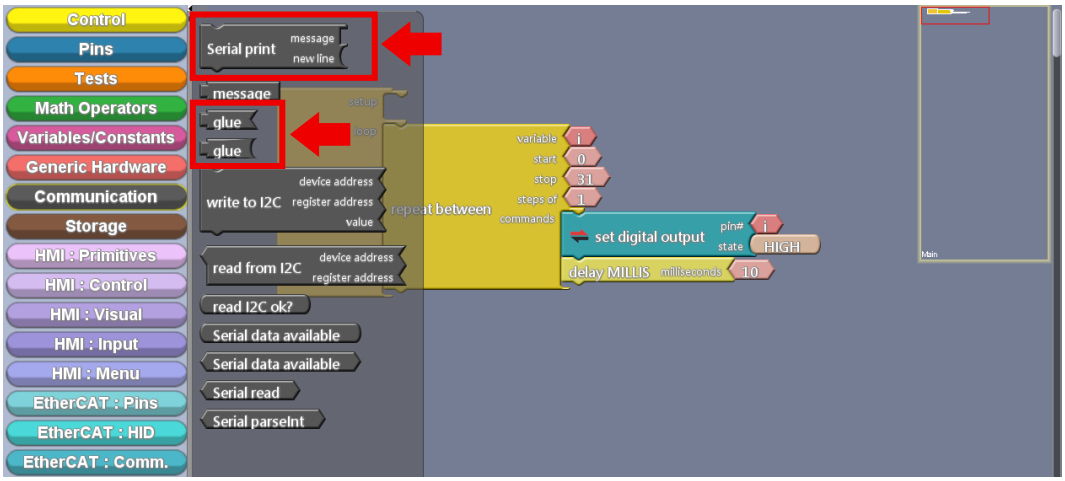
Next, we use the “**set digital output**” block from the “**EtherCAT : Pins**” class to the “**commands**” area of “**repeat between**” block.



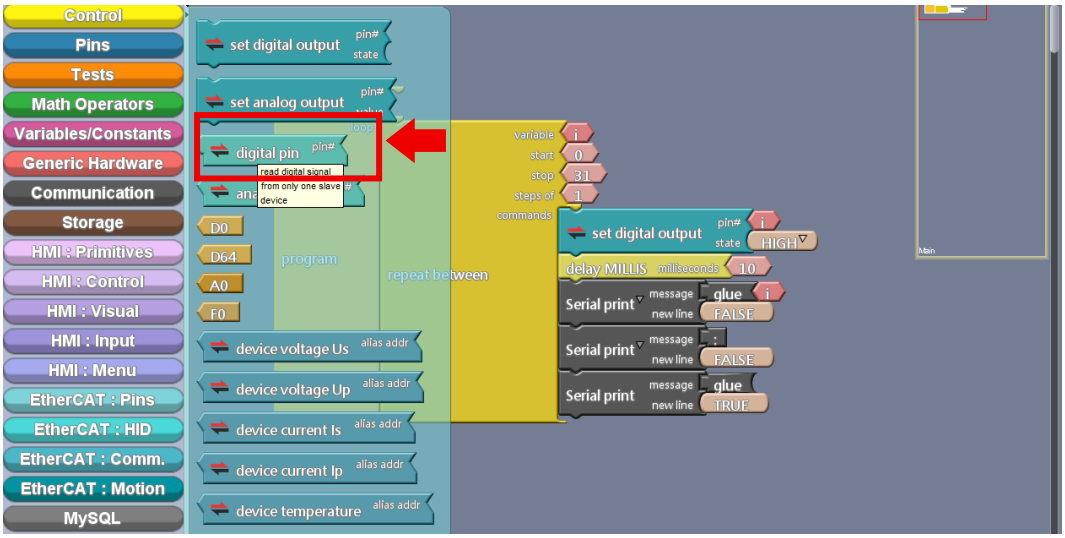
We set pin# to “**i**”, and state to “**HIGH**”; and because we need to wait for the EtherCAT cycle, we put a "**delay MILLIS**" block and set milliseconds to “**10**”.
Like this.



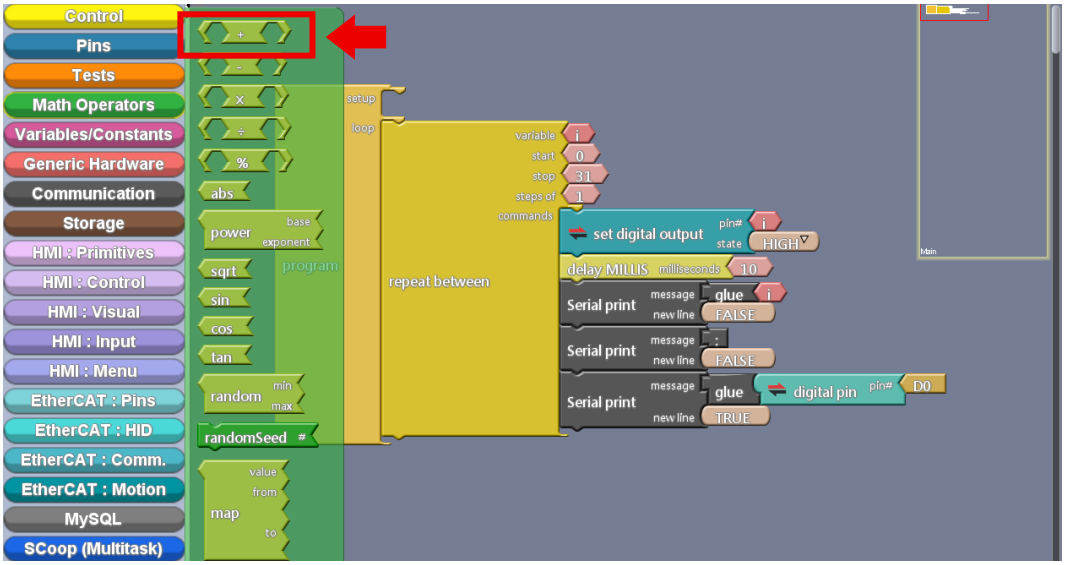
And we use the “**Serial print**” and “**glue**” blocks from the “**Communication**” class, to print out the digital input status.



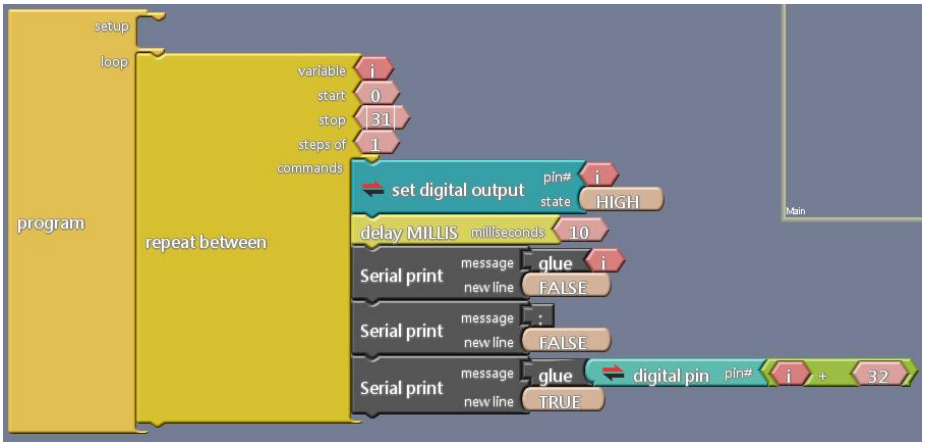
We also use the "**digital pin**" block from the "**EtherCAT : Pins**" class.



Also use the "**+**" block from the "**Math Operators**" class.

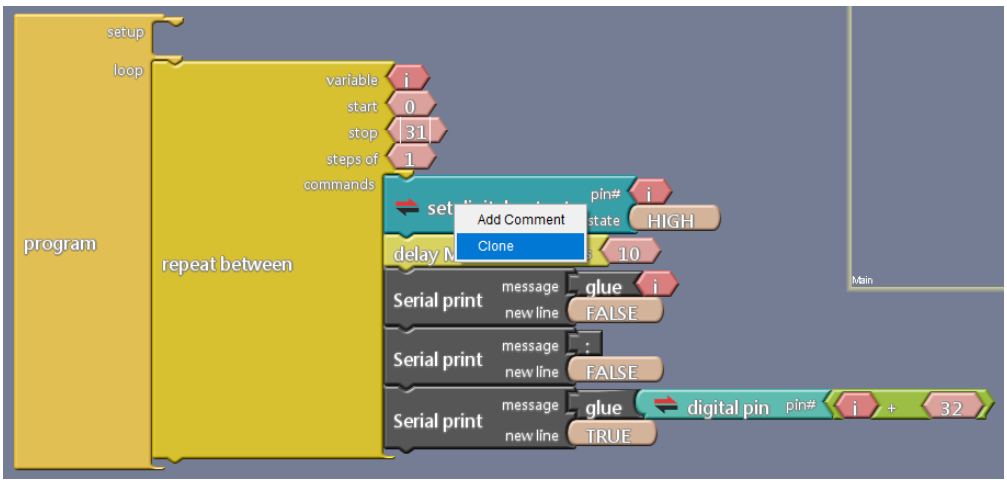


We want to print out "**i: <value>**", so we use the "**glue**" block to glue the variable "**i**" and the "**digital pin**" block to the "**message**" area of the "**Serial Print**" block.
Because the digital input pins map to the **Virtual Digital Pins #32~63**, we need to increment the value of variable i by 32.
Like this.



Next, we repeat the flow.

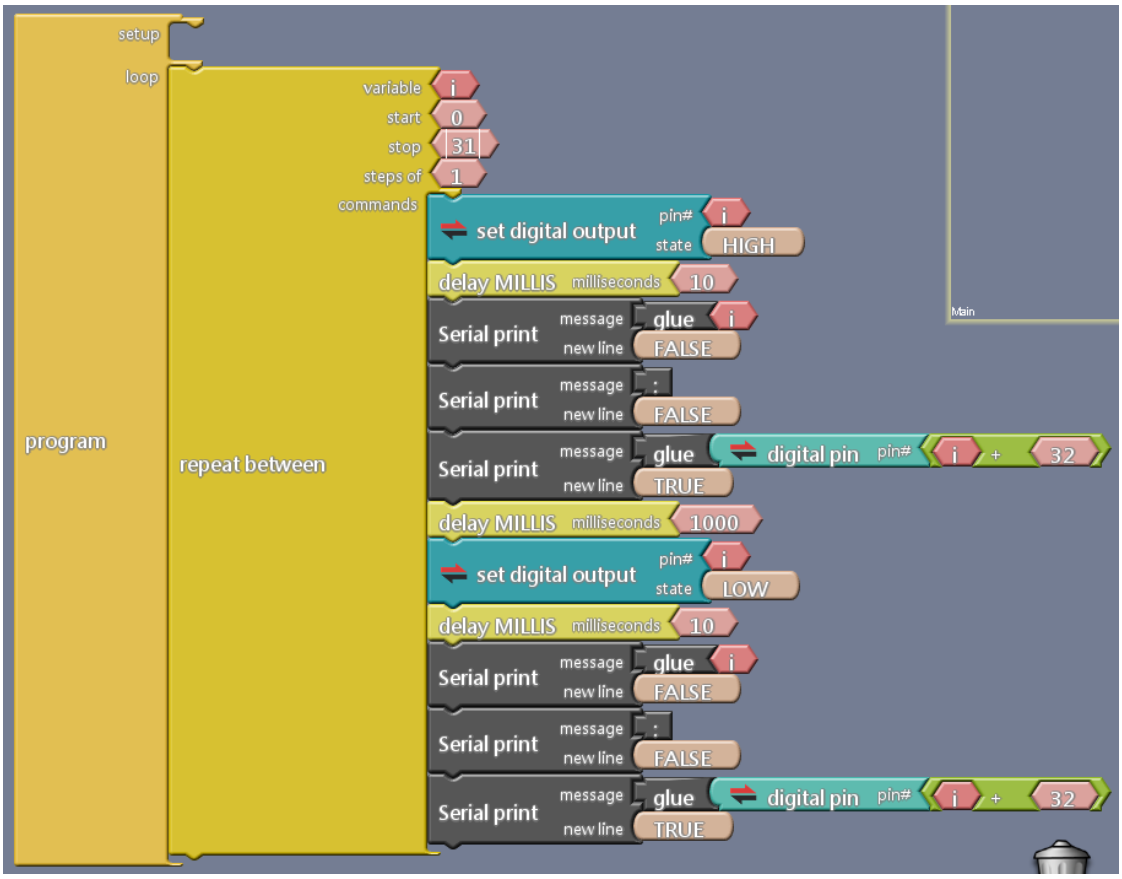
We can right-click the "**set digital output**" block, and it'll show "**Add Comment**" and "**Clone**".



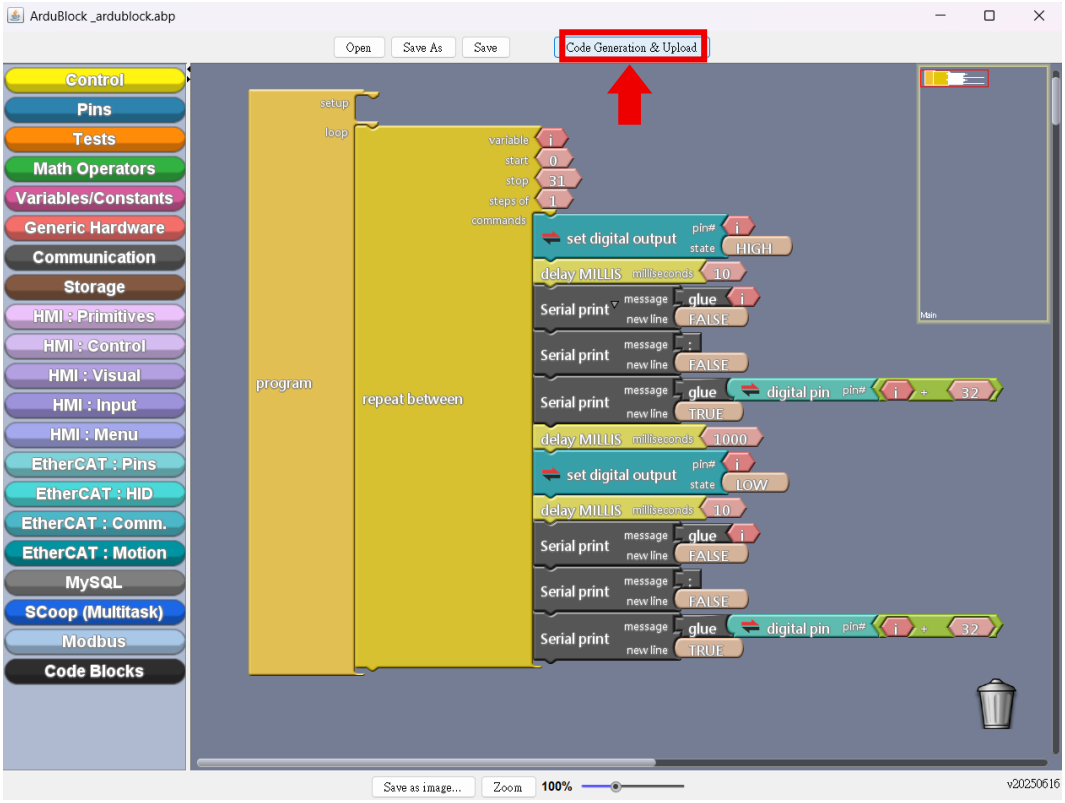
We can click "**Clone**" to copy all blocks and put them after.

And we change the state value of the second "**set digital output**" block to "**LOW**", and add a "**delay**" block (with a 1000 value) before it.

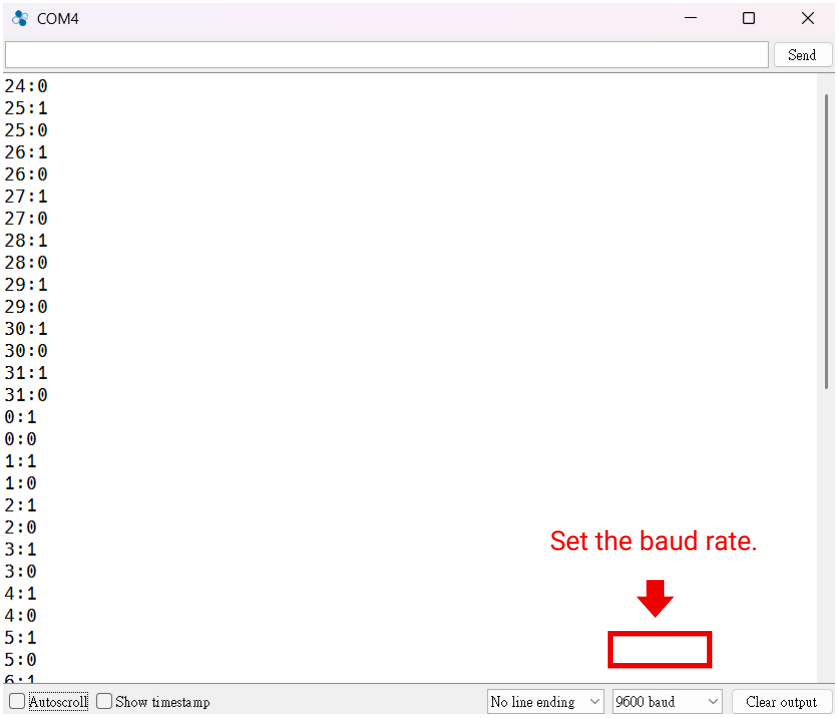
Like this.



After you finish, click the “**Code Generation & Upload**” button, and ArduBlock will automatically generate the program code and upload it to the QEC-M-01.



If EtherCAT initialization is successful, the Serial Monitor will print the Bore DI values.



***Note:** The default baud-rate of Serial monitor is 9600.

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:

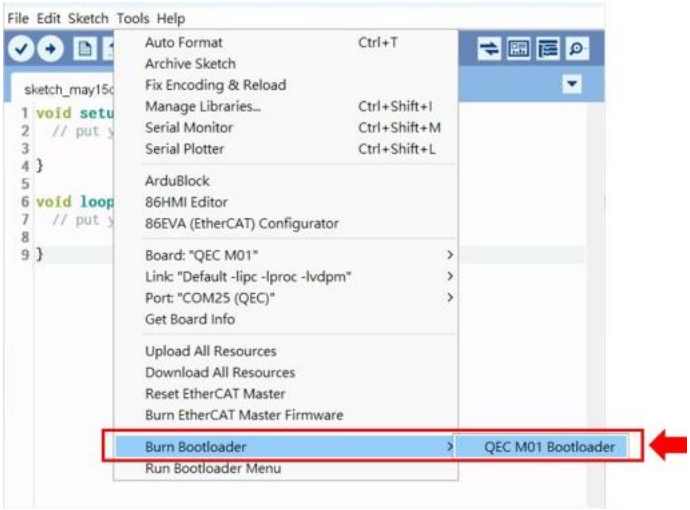
Step 1: Setting up QEC-M

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.

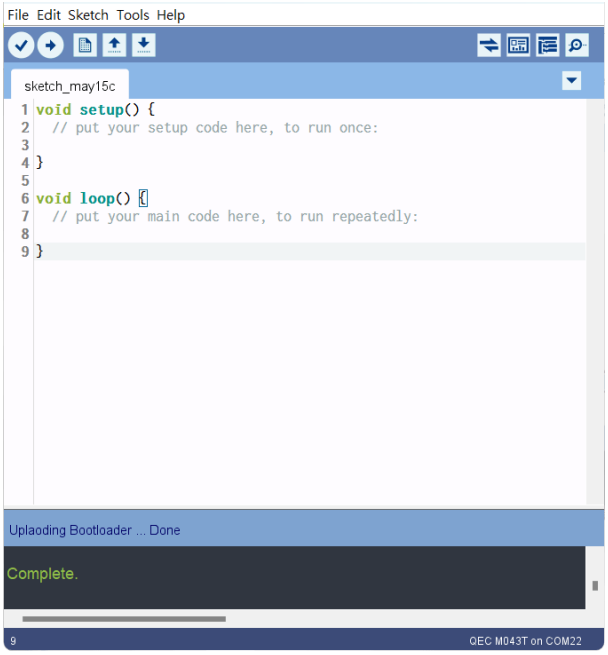
Step 2: Click “Burn Bootloader” button

After connecting to your QEC-M device, go to “**Tools**”> “**Burn Bootloader**”.
The name of the currently selected QEC-M device will be displayed.
Clicking on it will start the update process, which will take approximately 5 to 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.