Start Guide

EK1814 : EtherCAT Coupler with integrated digital inputs/outputs with 86EVA and ArduBlock

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

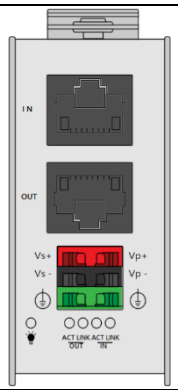
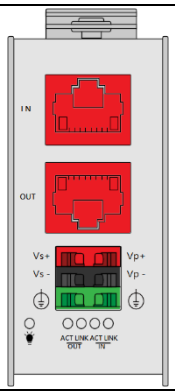
Date	Version	Description
2025/12/22	Version1.0	New release.

Preface

In this guide, we will show you how to use the EtherCAT MDevice **QEC-M-01** and the **EK-1814** series (EtherCAT Coupler with integrated digital inputs/outputs).

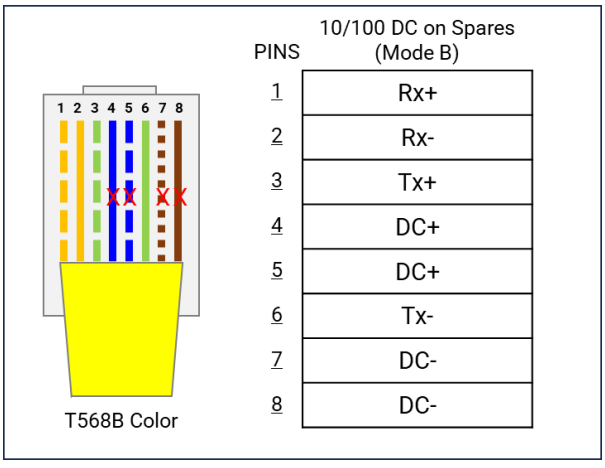
Notes QEC's PoE (Power over Ethernet)

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-

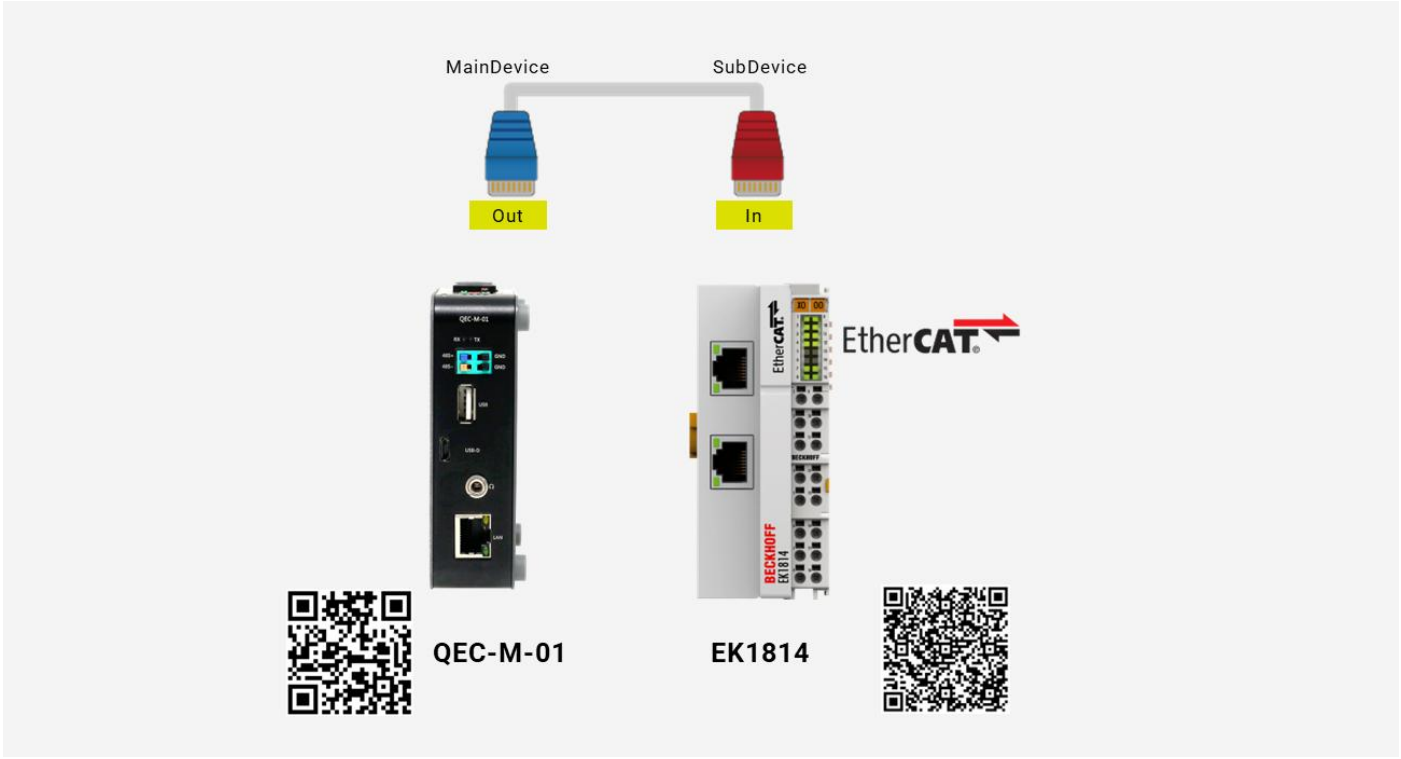
2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).

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

1. Connection and wiring hardware

The following devices are used here:

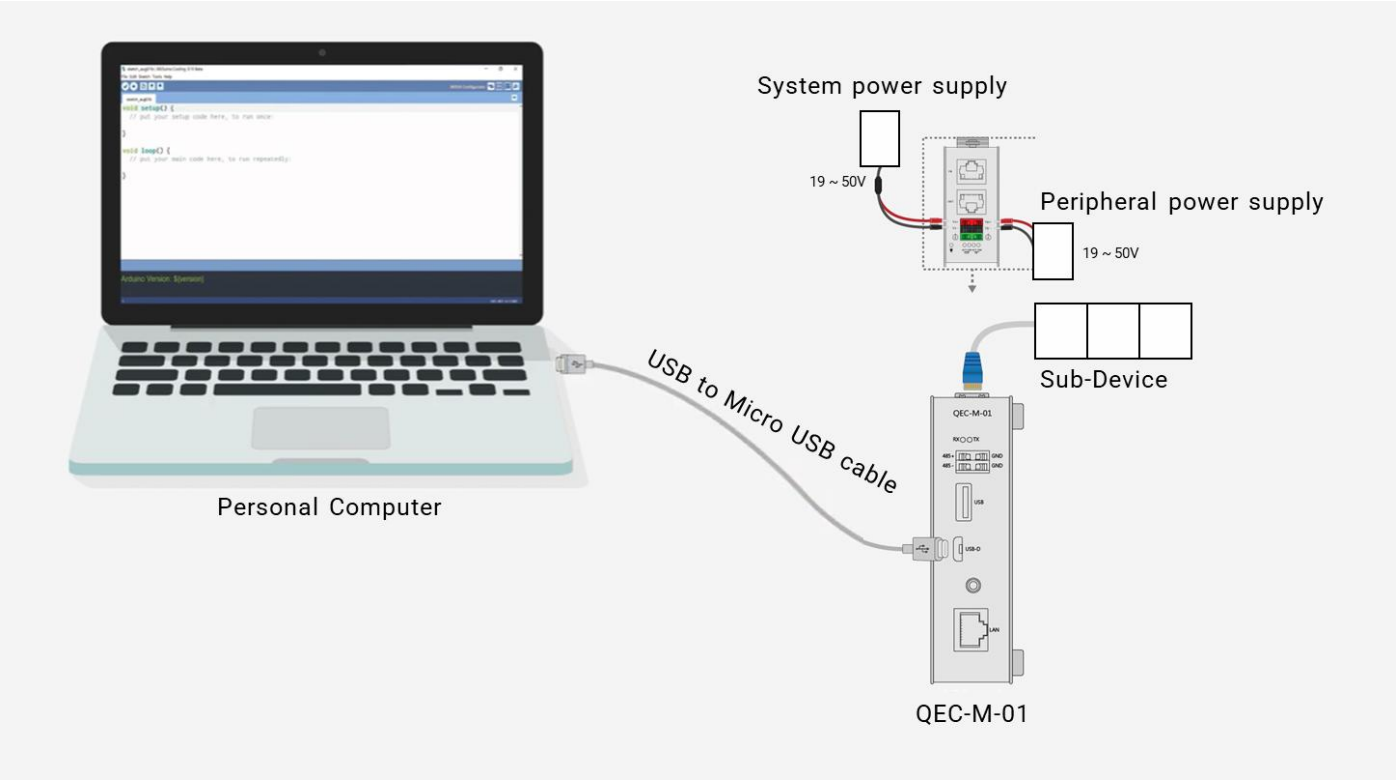
1. QEC-M-01 (EtherCAT MDevice)
2. EK-1814 series (EtherCAT Coupler with integrated digital inputs/outputs)
3. 24VDC power supply & EU-type terminal cable & LAN cable



1.1 QEC-M-01

QEC EtherCAT MDevice.

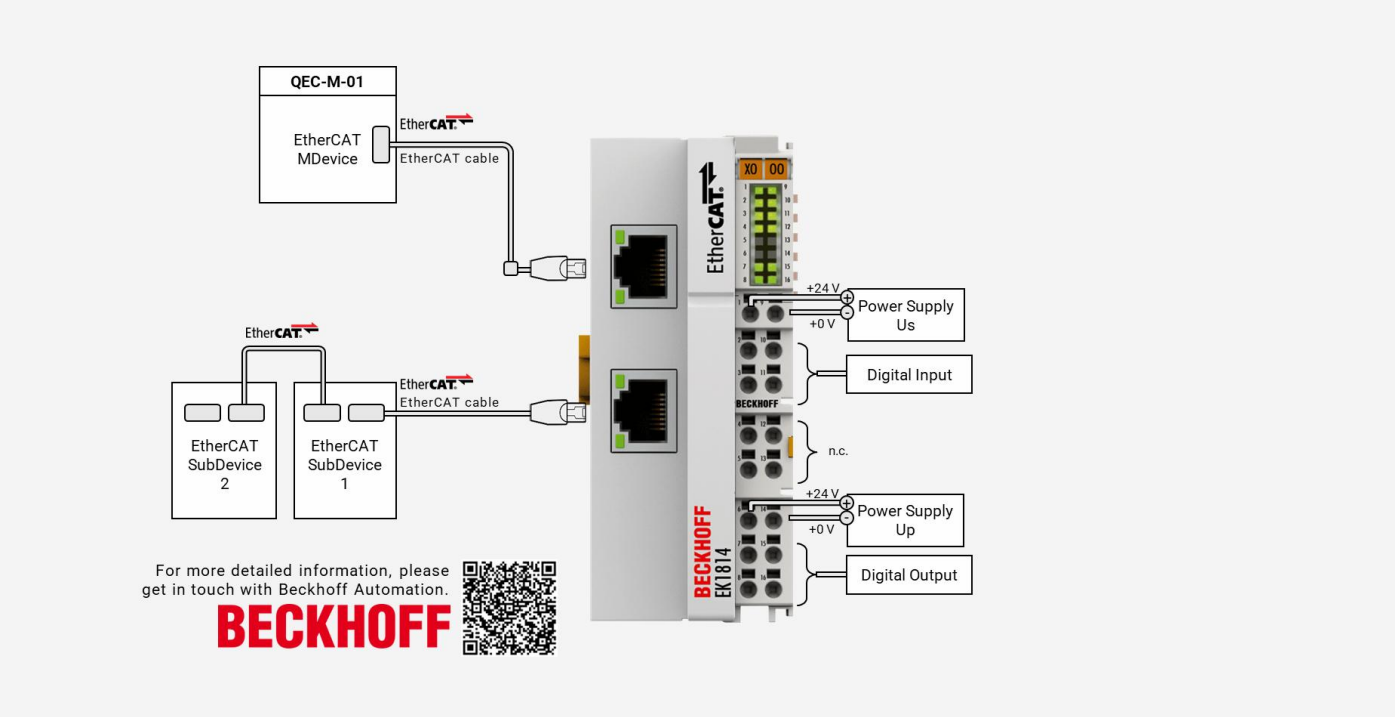
1. Power Supply: Connect to Vs+/Vs- and Vp+/Vp- power supplies via EU terminals for 24V power.
2. EtherCAT Connection: Using the EtherCAT Out port (On the top side) connected to the EtherCAT In port of EtherCAT SubDevice via RJ45 cable.



1.2 EK1814

The **Beckhoff EK1814** is an EtherCAT Coupler featuring **integrated digital I/O** (DI/DO). It can be used as an EtherCAT SubDevice in a QEC EtherCAT network to provide basic discrete input sensing and output driving.

The diagram shows a typical wiring example with a **QEC MDevice (e.g., QEC-M-01)** and an EtherCAT network.

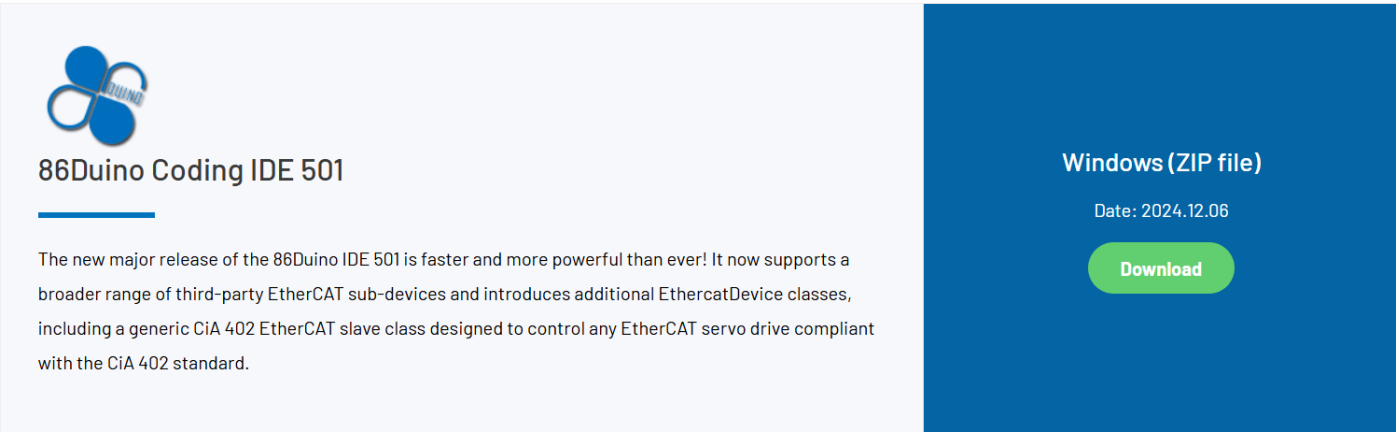


Connections are grouped by function:

- EtherCAT
  - IN: Connect to QEC MDevice EtherCAT port (or previous SubDevice).
  - OUT: Connect to next SubDevice.
- Power & Grounding
  - Us (Electronics / sensor supply): +24 V / 0 V (GND).
  - Up (Output supply): +24 V / 0 V (GND).
- Digital Inputs (DI)
  - Input 1–4: Connect external input signals referenced to the Us 0 V (GND).
- Digital Outputs (DO)
  - Output 1–4: Powered by Up +24 V.
  - Note (DO type): EK1814 DO channels are Sourcing (PNP) outputs. This means the output sources +24 V to the load when ON; wire the load return to 0 V (Up).
- Indicators / LEDs
  - PWR / RUN / Link/Act: Module and EtherCAT link status.
  - Channel LEDs: DI/DO channel status (Input 1–4, Output 1–4).

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.

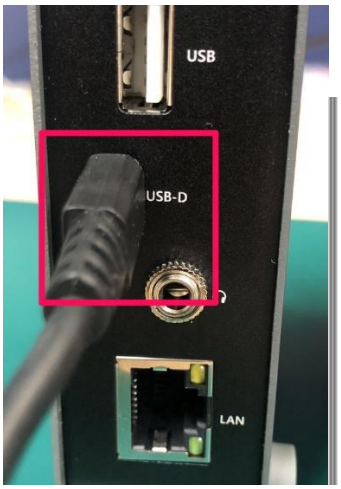
*__Note__: If Windows displays a warning, click Details once and then click the Continue Run button once.

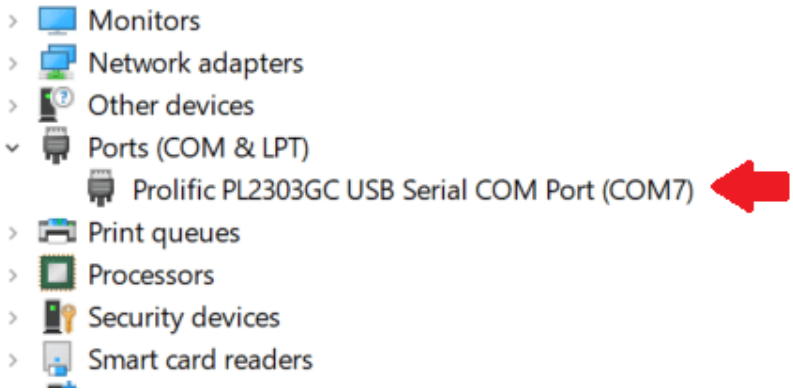
86Duino Coding IDE 501+ looks like below.

3. Connect to PC and set up the environment

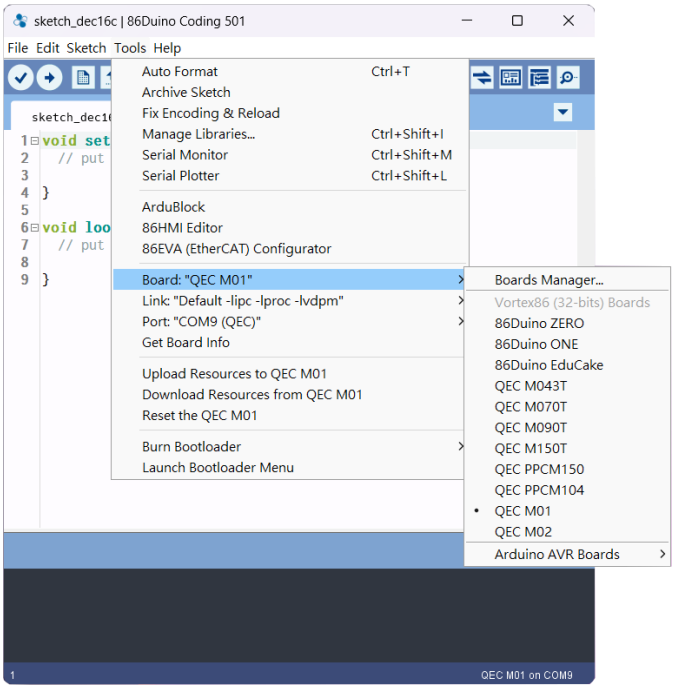
Follow the steps below to set up the environment:

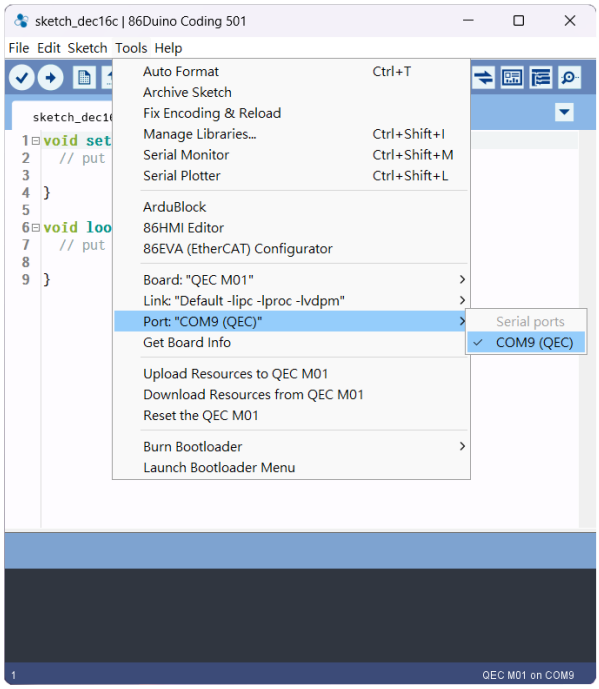
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “**Device Manager**” (select in the menu after pressing Win+X) ->” **Ports (COM & LPT)**” in your PC and expand the ports; you should see that the “**Prolific PL2303GC USB Serial 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-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 EK-1814 series (EtherCAT Coupler with integrated digital inputs/outputs) through the 86Duino IDE's graphical low-code programming tools, 86EVA and ArduBlock.

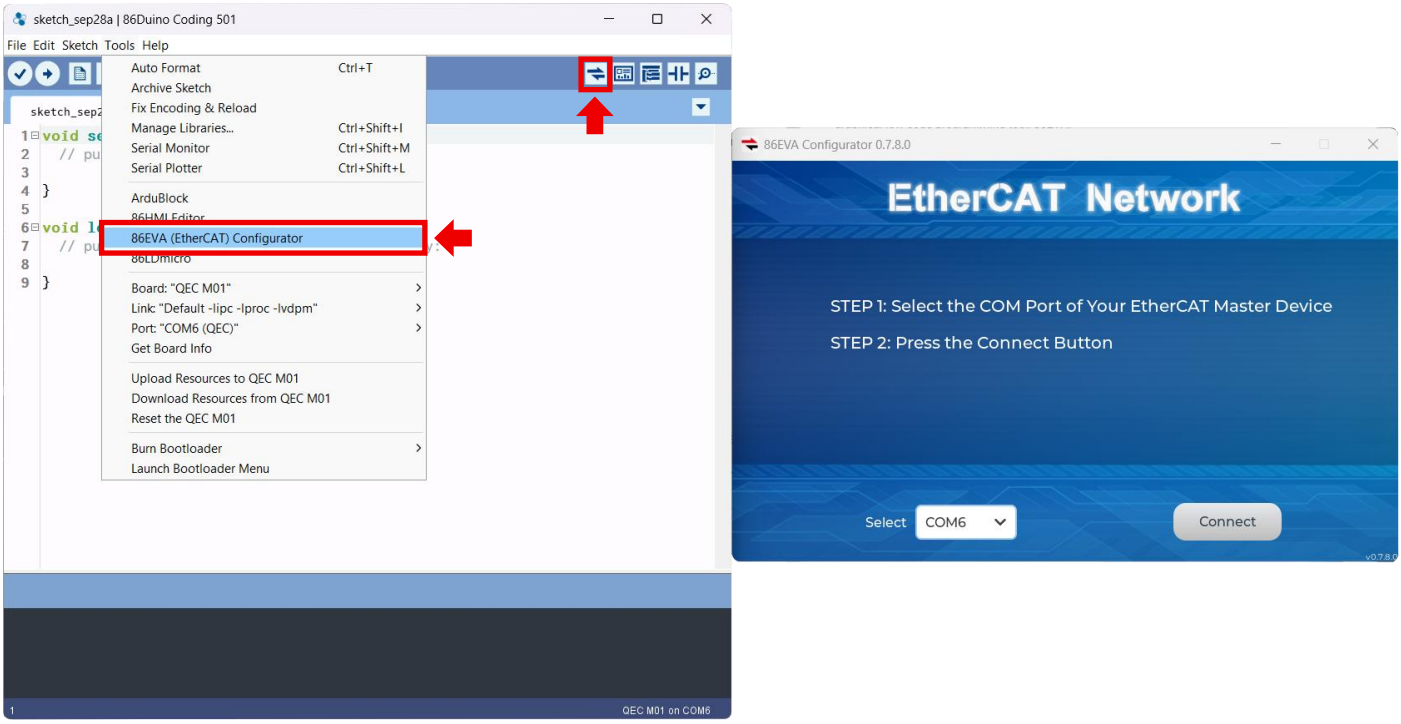
Software Tools Description:

- **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 section, we will periodically toggle EK1814 DO0 (HIGH/LOW), then read and print EK1814 DI0 after a short settling delay to verify that the input state reflects the output change.

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

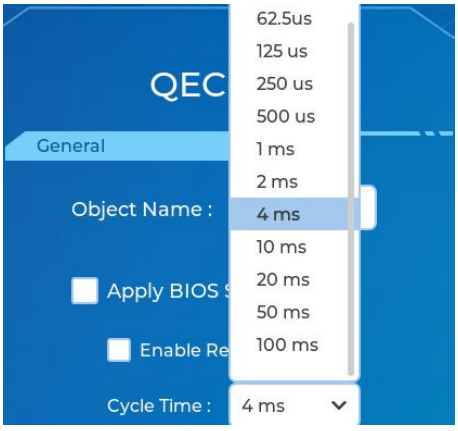
Press on the scanned device image to enter the corresponding parameter setting screen.

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.

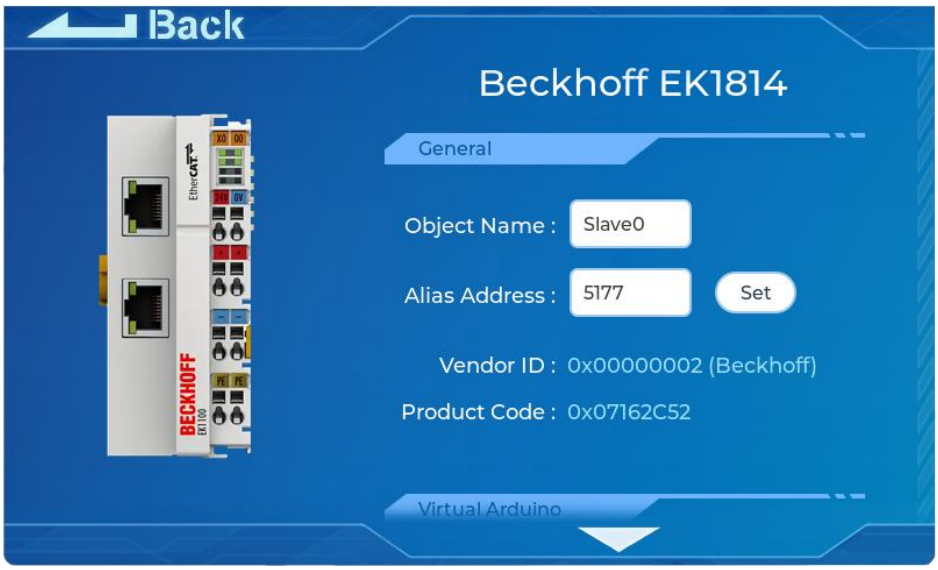
1. Turn off the "**Apply BIOS Settings**".
2. Select "**4ms**" to the Cycle Time.



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

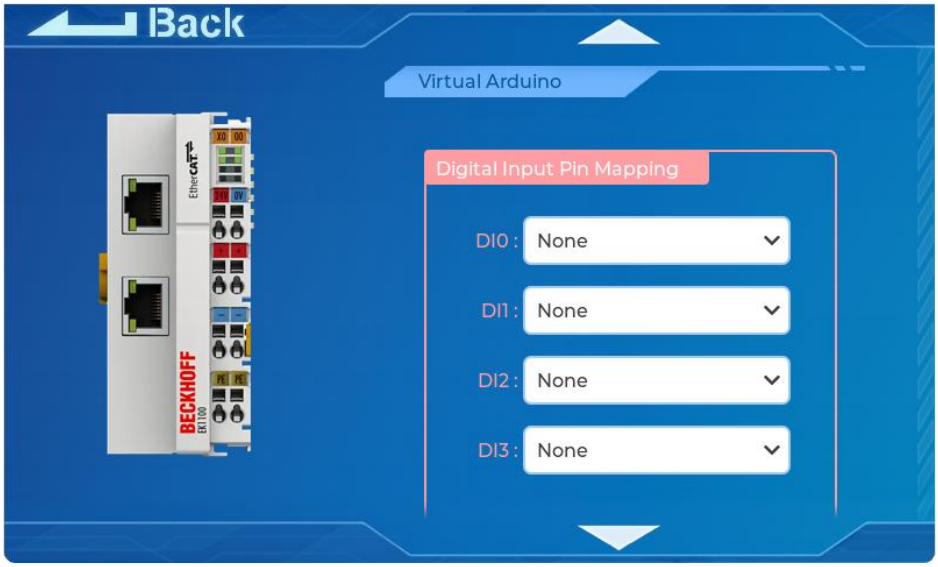
Step 2.2: EK1814

Press on the image of the EK1814 to see the parameter settings.



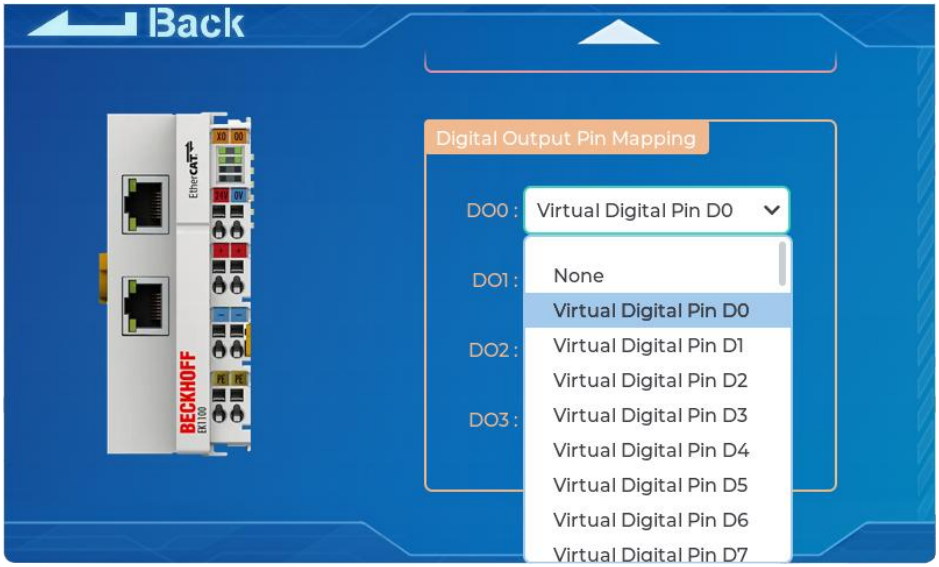
The page will show the Object Name, Alias Address, Vendor ID, Product Code, and Virtual Arduino Mapping.

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

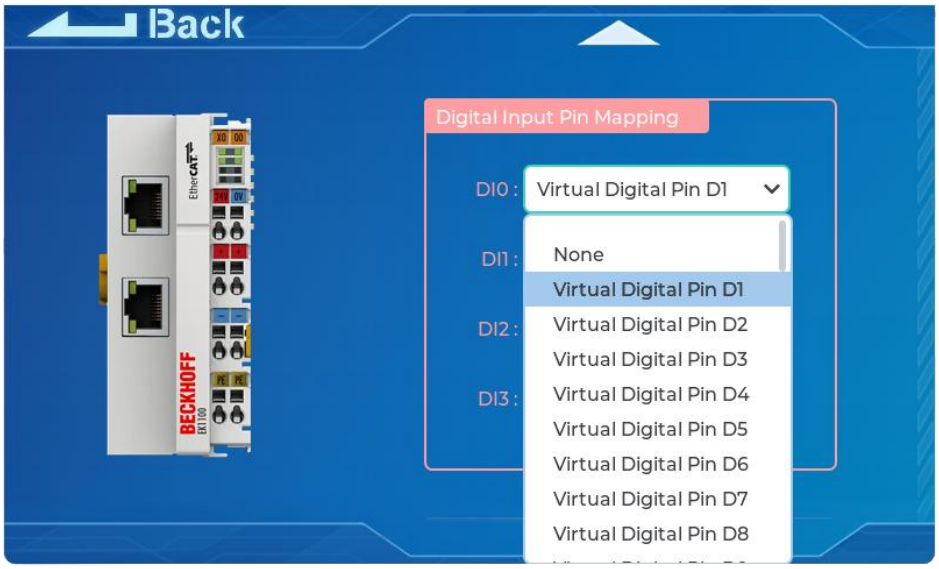


There are two areas of the Virtual Arduino in the EK1814: **Digital Input Pin Mapping** and **Digital Output Pin Mapping**.

First, we go down to the “**Digital Output Pin Mapping**” area, and we select “**Virtual Digital Pin D0**” in the drop-down box of "**DO0**".

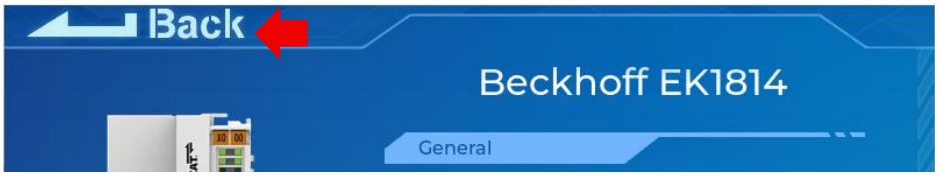


Next, we select “**Virtual Digital Pin D1**” in the drop-down box of “**DI0**” in the “**Digital Input Pin Mapping**” area.



All these settings and mappings are for **ArduBlock** tool configuration.

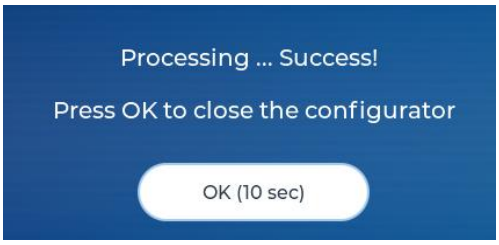
Next, click "**Back**" in the upper left corner to return.



Step 3: Generate the code

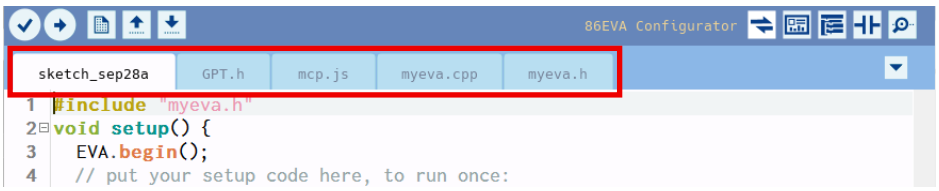
Once you've set your device's parameters, go back to the home screen and press the "**Code Generation**" button in the bottom right corner.

When you're done, 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.

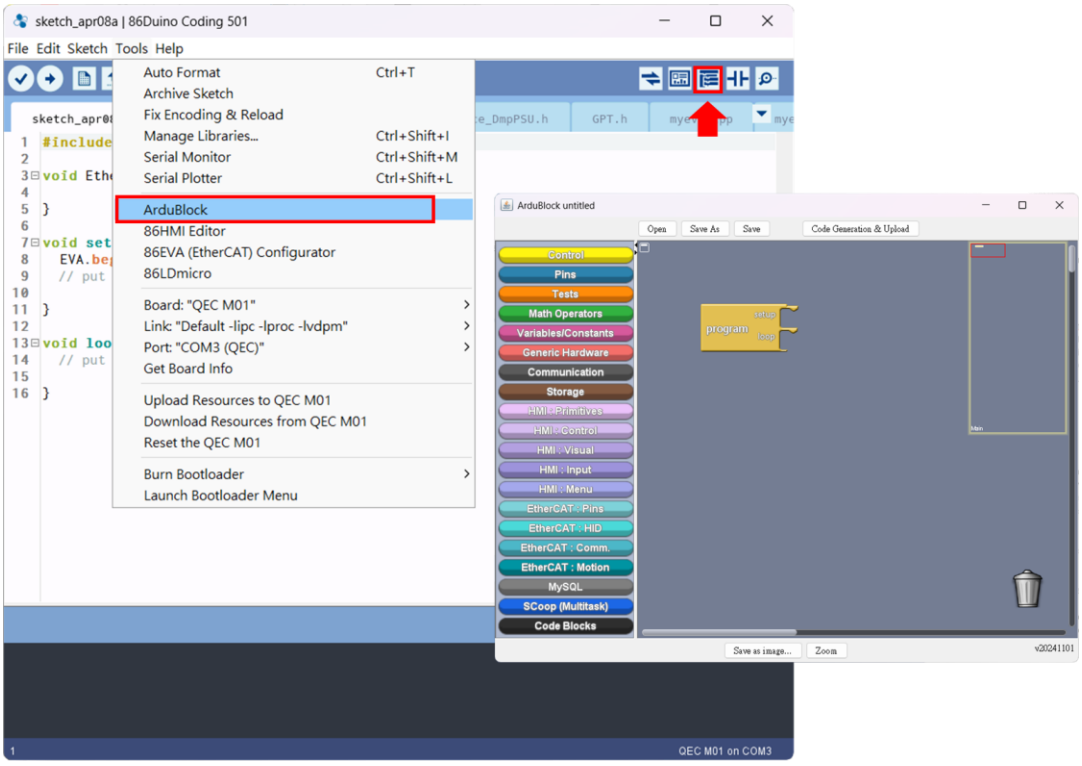
The programming code from 86EVA are set as the following by default:

- EK1814 module: `EthercatDevice_Generic` object.
- EtherCAT mode: ECAT_SYNC.

And here is the setting by users:

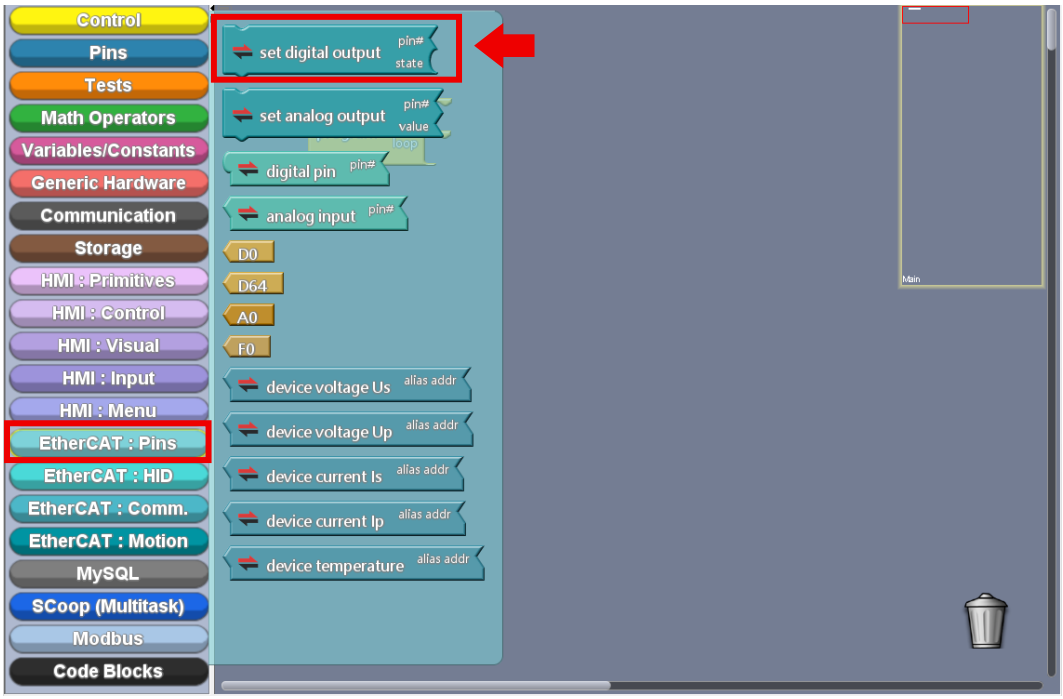
- EtherCAT Cycle time: 4 milliseconds (Depend on EK1814 3ms filter).
- The `EthercatMaster` object (“EcatMaster”) represents the **QEC-M-01**, while the `EthercatDevice_Generic` object (“slave0”) represents the **EK1814** module.

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

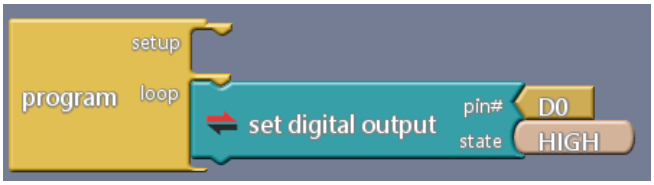


In this section, we will periodically toggle EK1814 DO0 (HIGH/LOW), then read and print EK1814 DI0 after a short settling delay to verify that the input state reflects the output change.

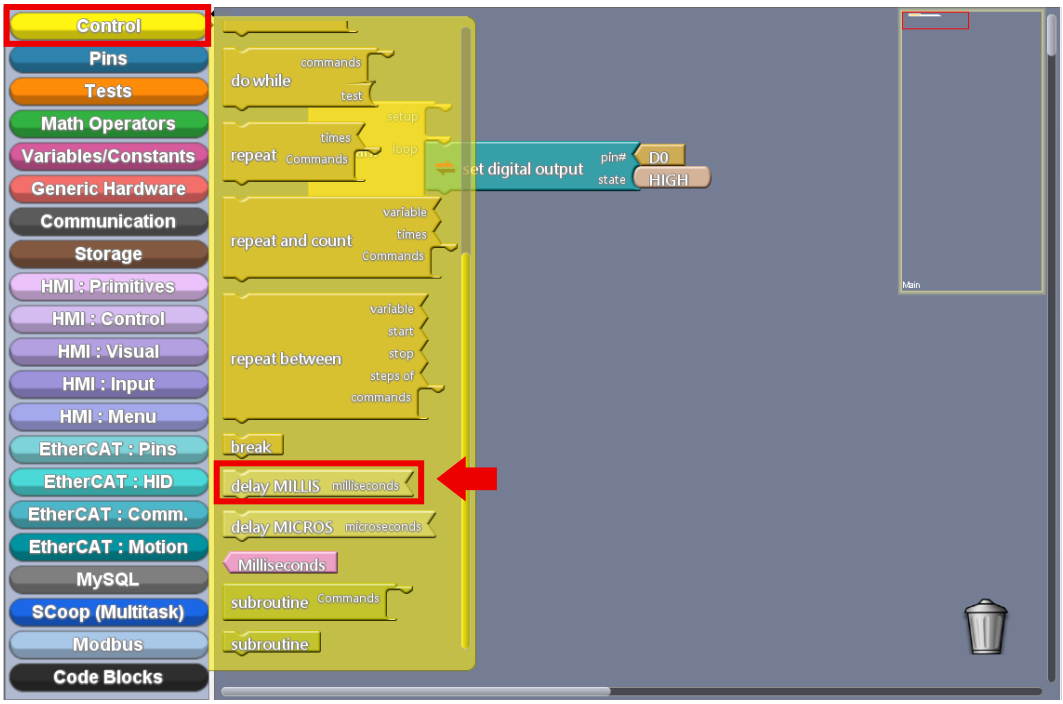
First, we use the “**set digital output**” block from the “**EtherCAT : Pins**” class to the program’s main loop to set the digital output pin #D0 value to “**HIGH**” (we select “**Virtual Digital Pin D0**” for EK1814 DO 0).



Like this. We set D0 to “**HIGH**”.

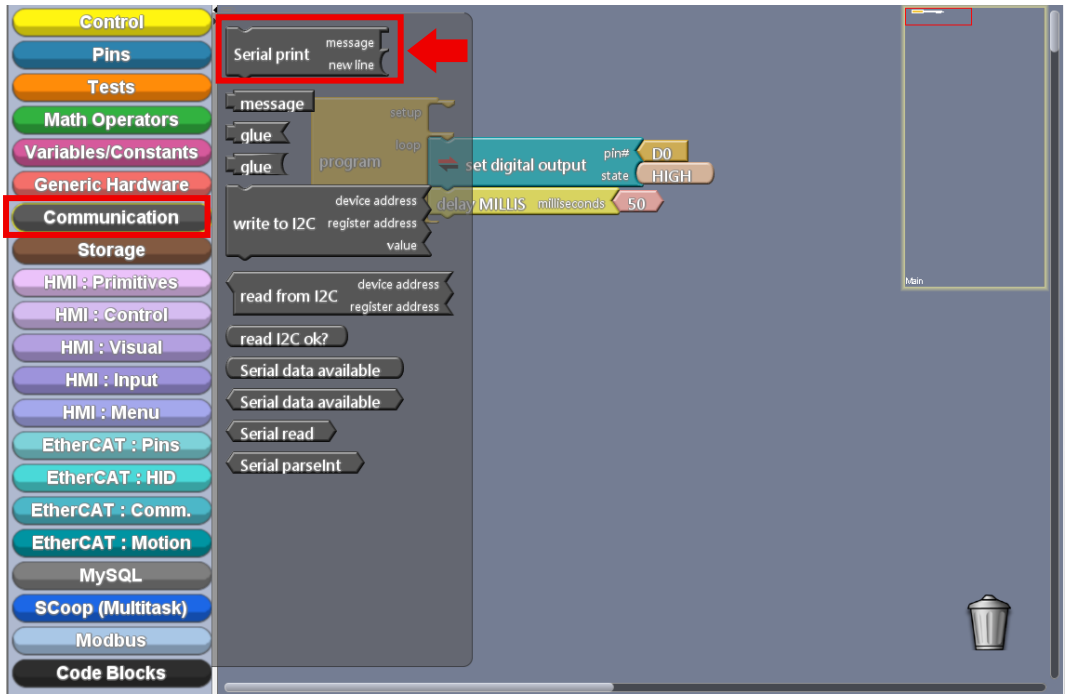


Because we need to wait for the EtherCAT communication and EK1814 filter time, we put a "**delay MILLIS**" block from the "**Control**" class.

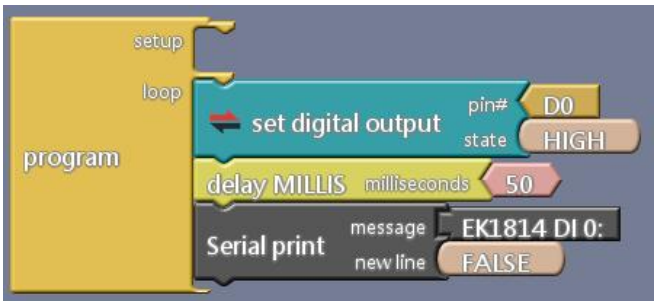


In this case, we delay for 50 milliseconds, which is used to wait for DI to reflect the DO change and for EtherCAT to refresh the input PDO data before reading.
Like this.

And we use the "**Serial print**" block in the "**Communication**" class, to print out the digital input status.

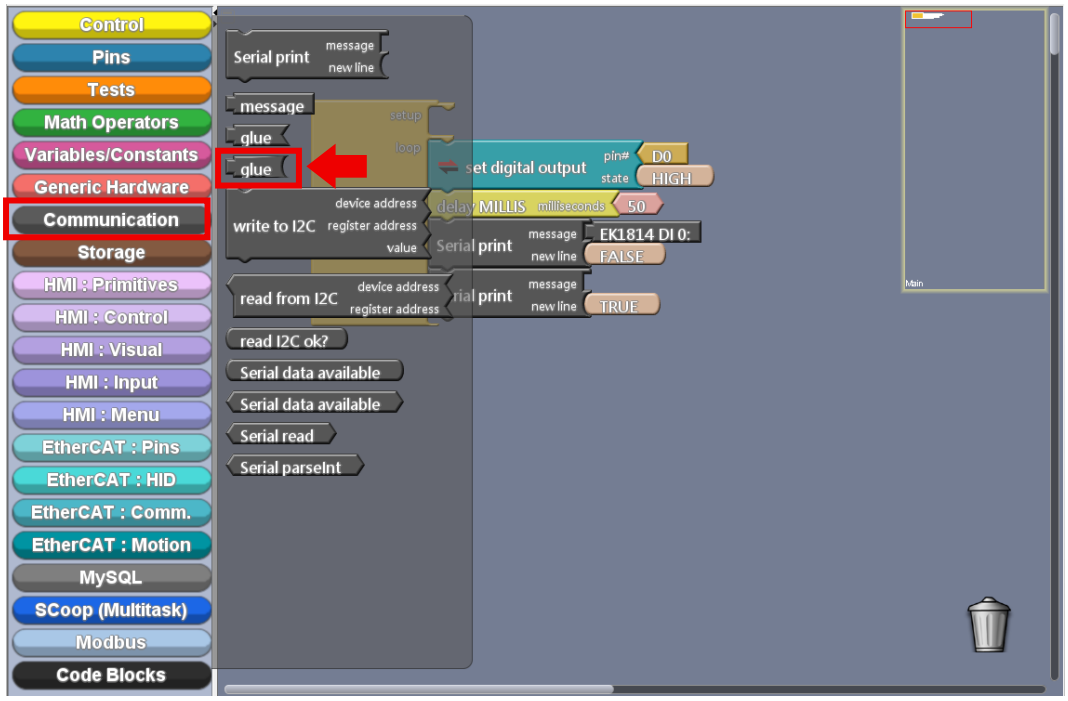


We set the "**message**" to "**EK1814 DI 0:** ", and "**new line**" to "**FALSE**" in the "**Serial print**" block.

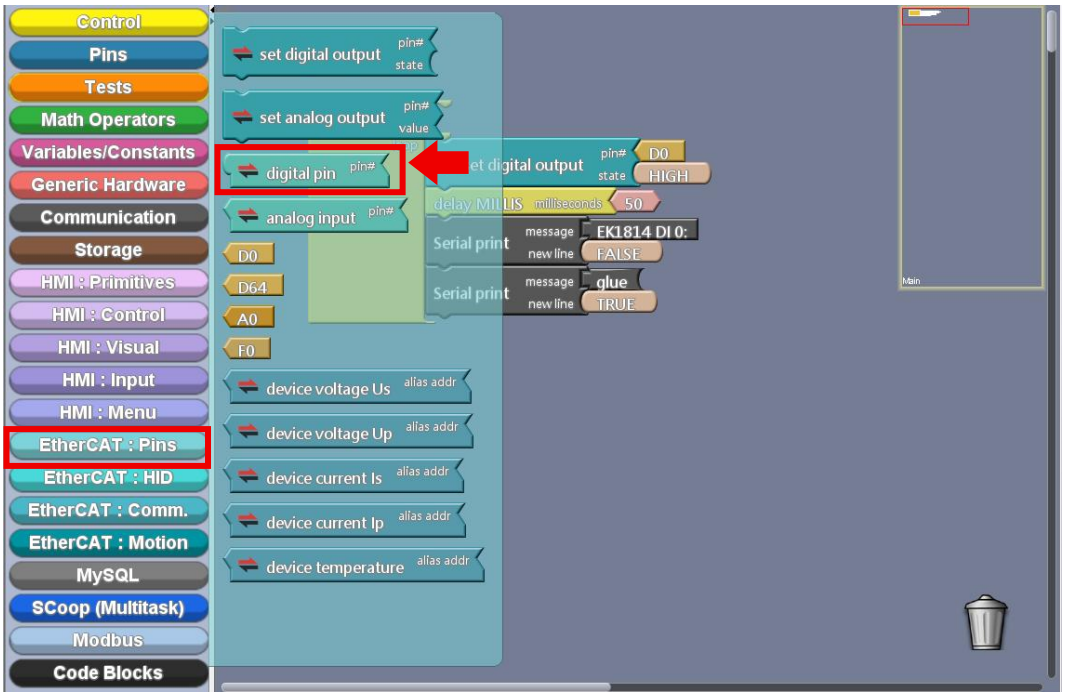


Then we add another “**Serial print**” block after it to print the digital input status.
We use the “**glue**” block from the “**Communication**” Class to glue the "**digital pin**" block from the "**EtherCAT : Pins**" class to the "**message**" area of the “**Serial Print**” block.

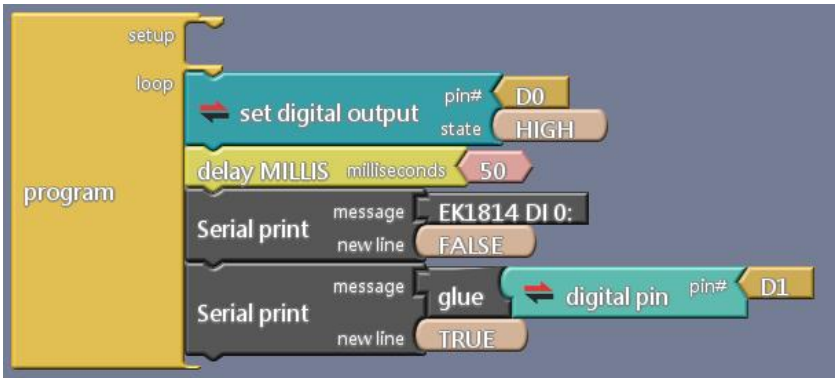
“**glue**” block:



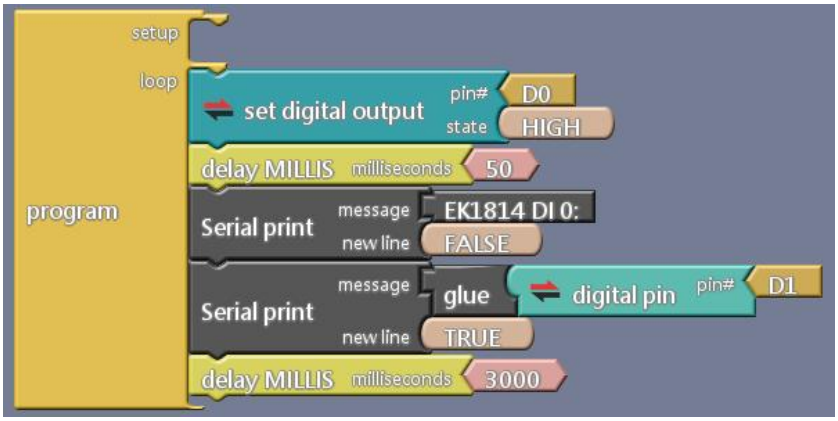
“**digital pin**” block:



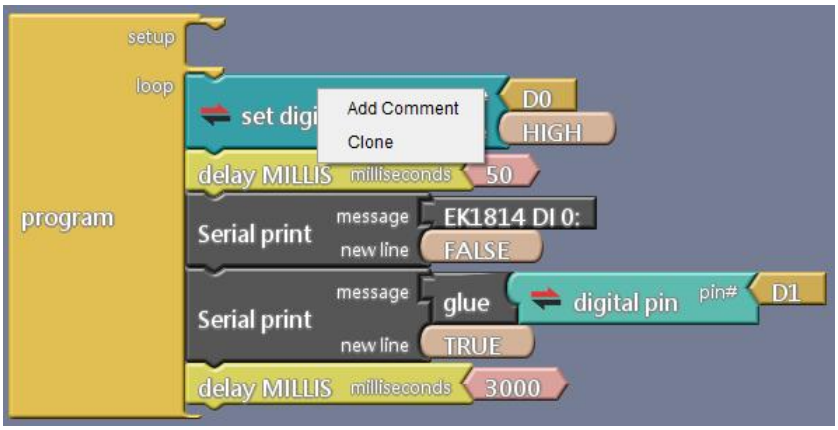
We read the digital pin **#D1** value (we select "**Virtual Digital Pin D1**" for EK1814 DI 0).
Like this.



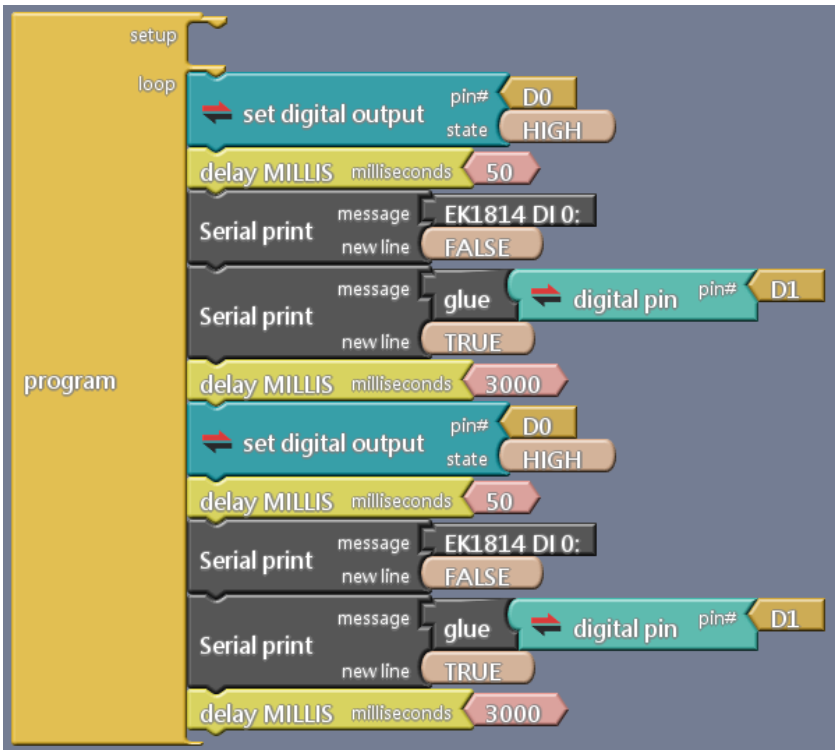
We set the 3 seconds to the "**delay MILLIS**" block to hold time before changing the state.



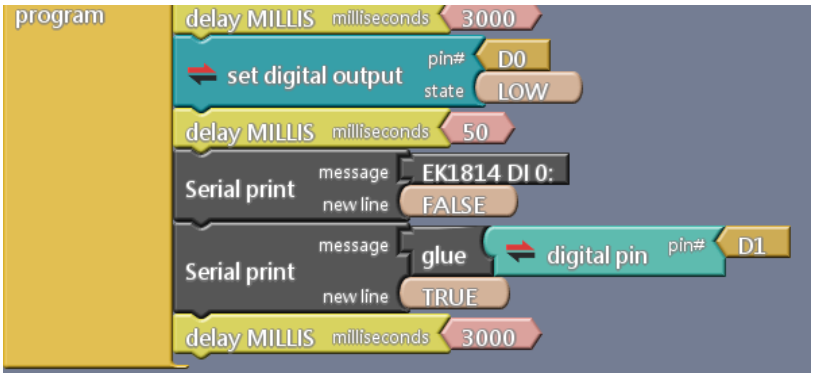
Next, we repeatedly set the digital output and print the digital input state. We can right-click the "**set digital output**" block, and it'll show "**Add Comment**" and "**Clone**".



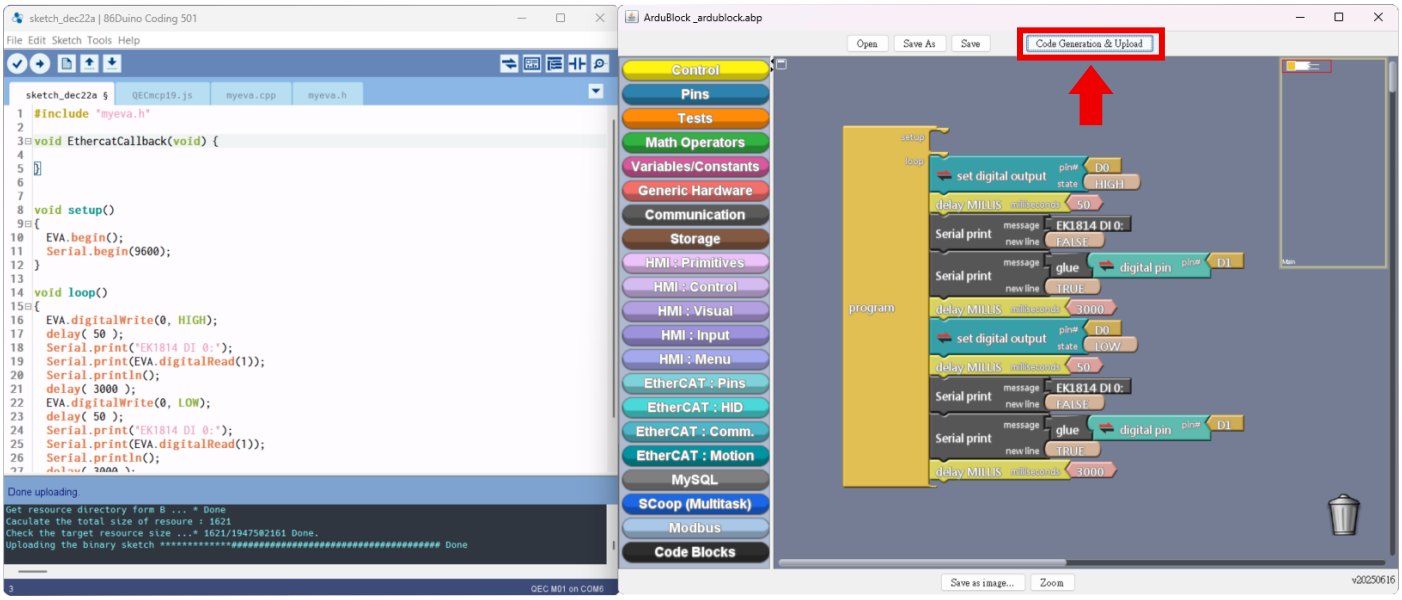
We can click "**Clone**" to copy all blocks and put them after.



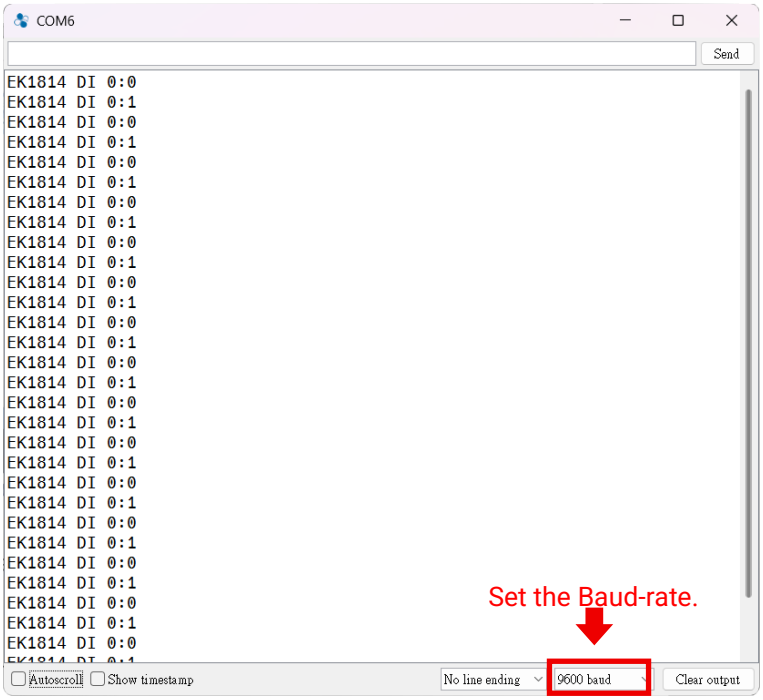
And we change the state value of the second "**set digital output**" block to "**LOW**".
Like this.



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 the EtherCAT communication configuration is successful, it will print the EK1814 DI 0 value to the serial monitor.



***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:

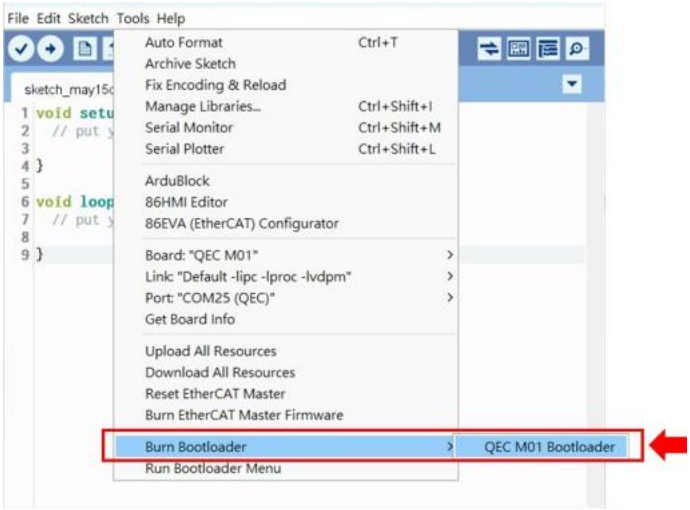
Step 1: Setting up QEC-M

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.

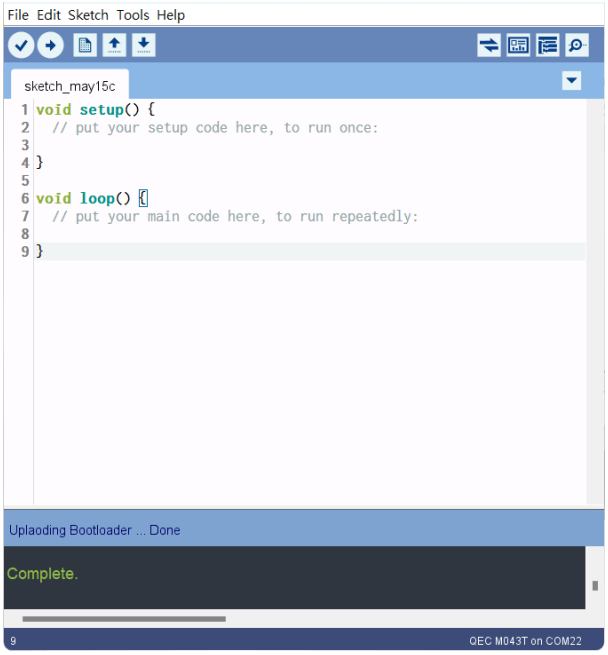
Step 2: Click “Burn Bootloader” button

After connecting to your QEC-M 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.