ICOP Technology Inc.

Start Guide

HID: RS232 with 86EVA and ArduBlock

86Duino Coding IDE 500

EtherCAT Library

(Version 1.0)

Revision

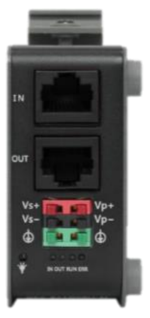
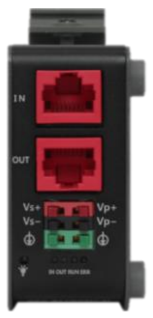
Date	Version	Description
2024/10/17	VERSION1.0	NEW RELEASE.

Preface

In this guide, we will show you how to use the EtherCAT Master QEC-M-01P and the QEC-RXXHUX Series (EtherCAT HID Slave).

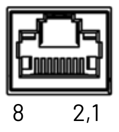
Notes QEC's PoE (Power over Ethernet)

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:

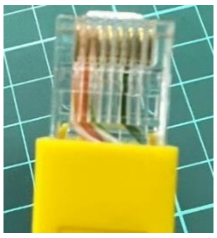
1. The PoE function of QEC is different and incompatible with EtherCAT P, and the PoE function of QEC is based on PoE Type B, and the pin functions are as follows:

	Pin #	Signal Name	Pin #	Signal Name
	1	LAN1_TX+	2	LAN1_TX-
	3	LAN1_RX+	4	VS+
	5	VP+	6	LAN1_RX-
	7	VS- (GND)	8	VP- (GND)

* PoE LAN with the Red Housing; Regular LAN with Black Housing.

* L4, L5, L7, L8 pins are option, for RJ45 Power IN/OUT.

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 master connects with a third-party EtherCAT slave).

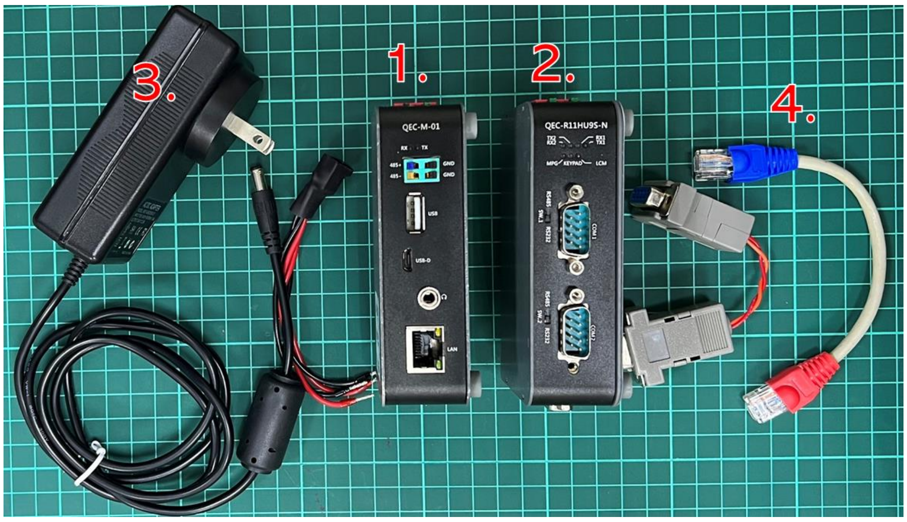


3. QEC's PoE power supply is up to 24V/3A.

1. Connection and wiring hardware

The following devices are used here:

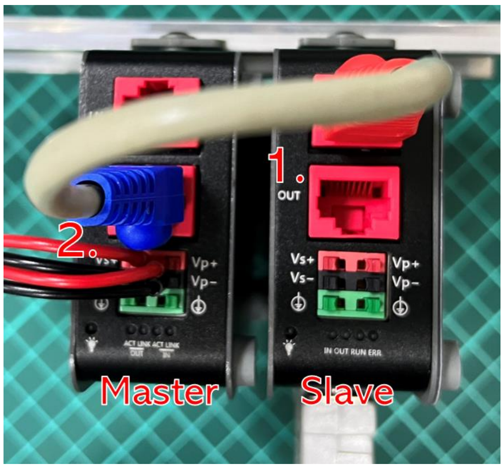
1. QEC-M-01P（EtherCAT Master/PoE）
2. QEC-R11HU9S-N (EtherCAT HID Slave, supports 2 UART, 1 MPG, 1 Keypad, 1 LCM)
3. 24V power supply & EU-type terminal cable
4. RS232 cable & LAN cable



1.1 QEC-M-01P

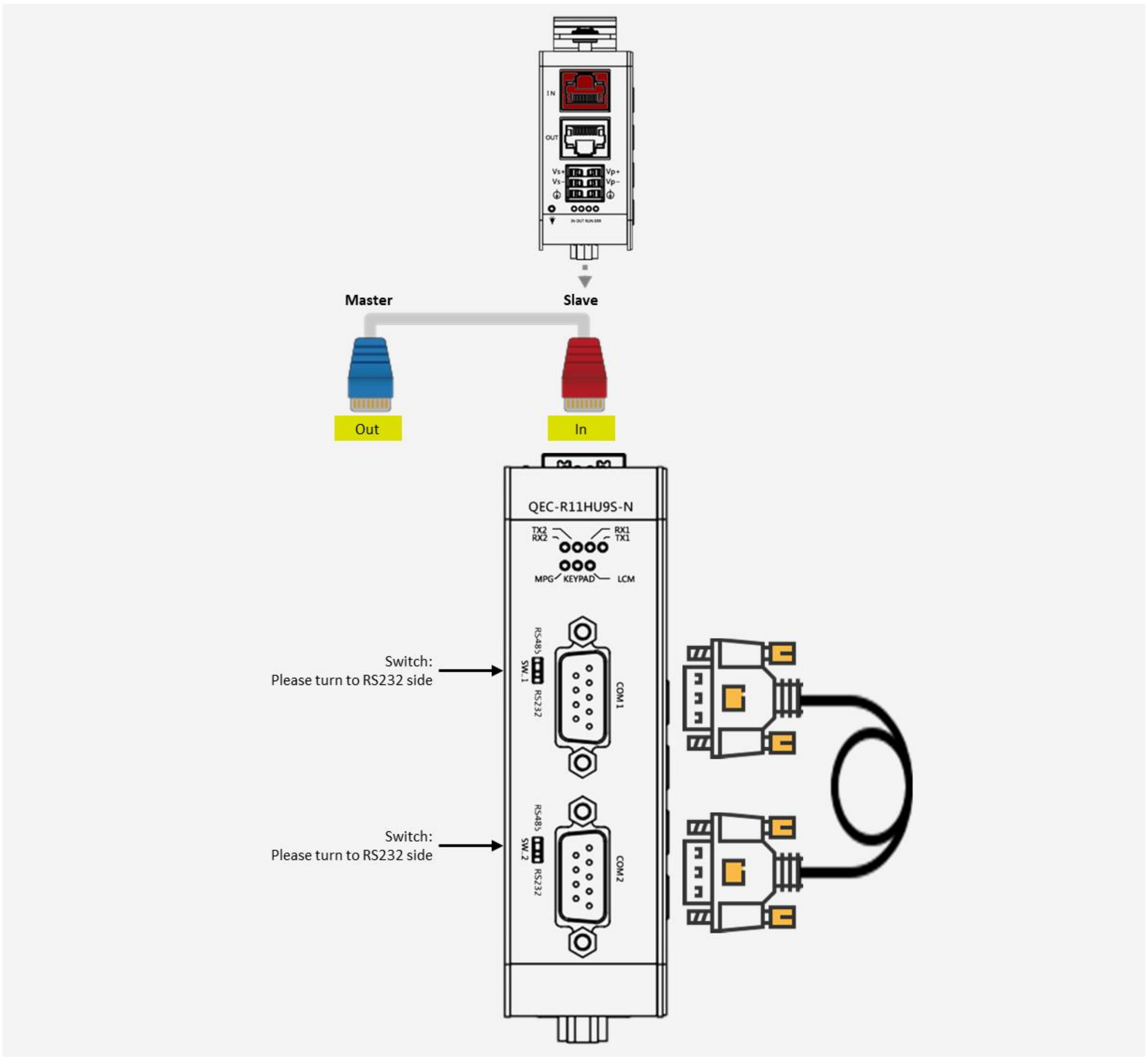
QEC EtherCAT master with PoE function.

1. Using the EtherCAT Out port (top side) connected to the EtherCAT In port of QEC-R11HU9S via RJ45 cable (powered by PoE).
2. Connect to Vs+/Vs- and Vp+/Vp- power supplies via EU terminals for 24V power.



1.2 QEC-R11HU9S-N

Connect COM1 and COM2 on QEC-R11HU9S-N via RS232 cable.

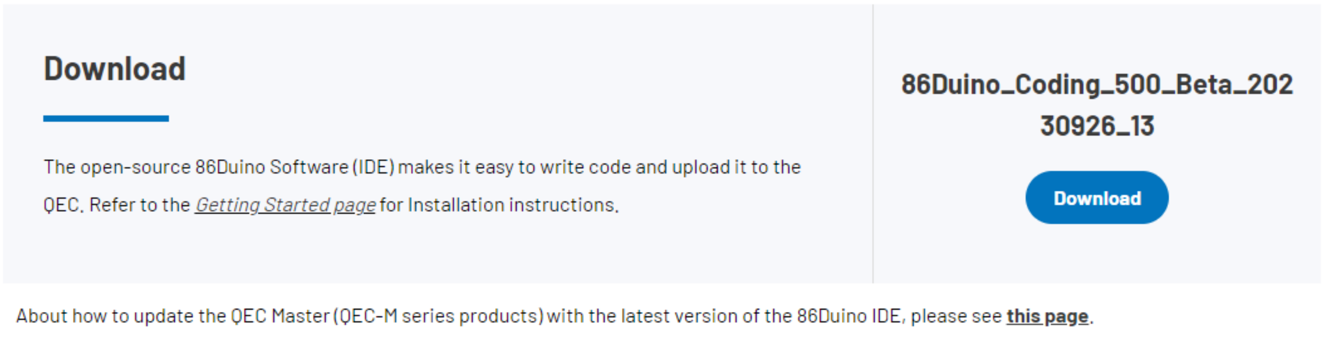


Note:

There are switches near both UART ports; users can change the switch's direction to adjust to the RS232 function or RS485 function. We use RS232 in this example.

2. Software/Development Environment

Download 86duino IDE from https://www.qec.tw/software/.



Download

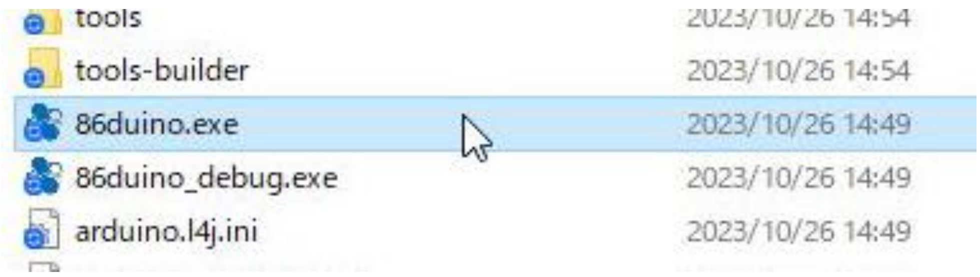
The open-source 86Duino Software (IDE) makes it easy to write code and upload it to the QEC. Refer to the *Getting Started page* for Installation instructions.

86Duino_Coding_500_Beta_20230926_13

Download

About how to update the QEC Master (QEC-M series products) with the latest version of the 86Duino IDE, please see **this page**.

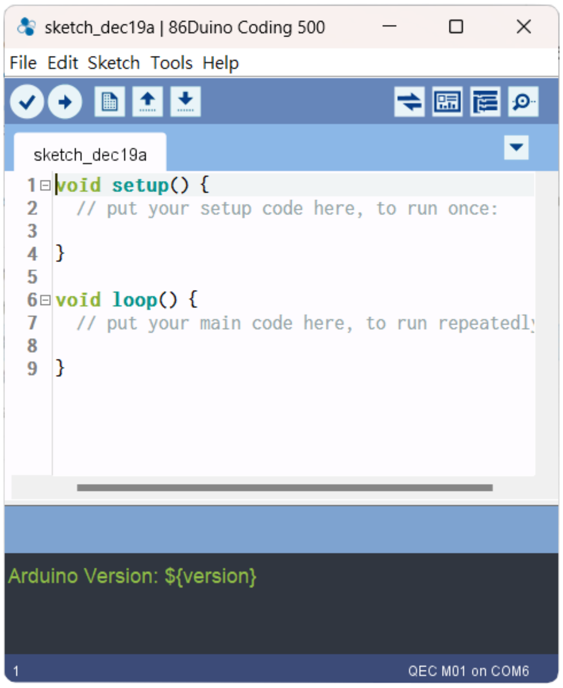
After downloading, please unzip the downloaded zip file, no additional software installation is required, just double-click 86duino.exe to start the IDE.



```
tools                 2023/10/26 14:54
tools-builder         2023/10/26 14:54
86duino.exe           2023/10/26 14:49
86duino_debug.exe     2023/10/26 14:49
arduino.l4j.ini       2023/10/26 14:49
```

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

86Duino Coding IDE 500+ looks like below.



sketch_dec19a | 86Duino Coding 500
File Edit Sketch Tools Help
sketch_dec19a

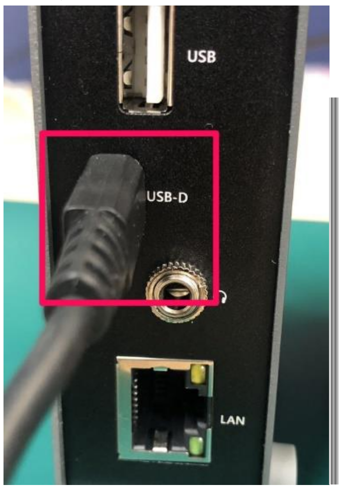
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedl
  
}
```

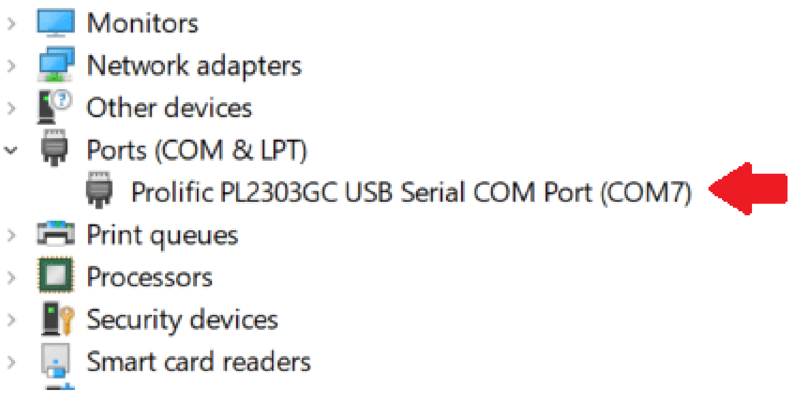
```
Arduino Version: ${version}
```

3. Connect to PC and set up the environment

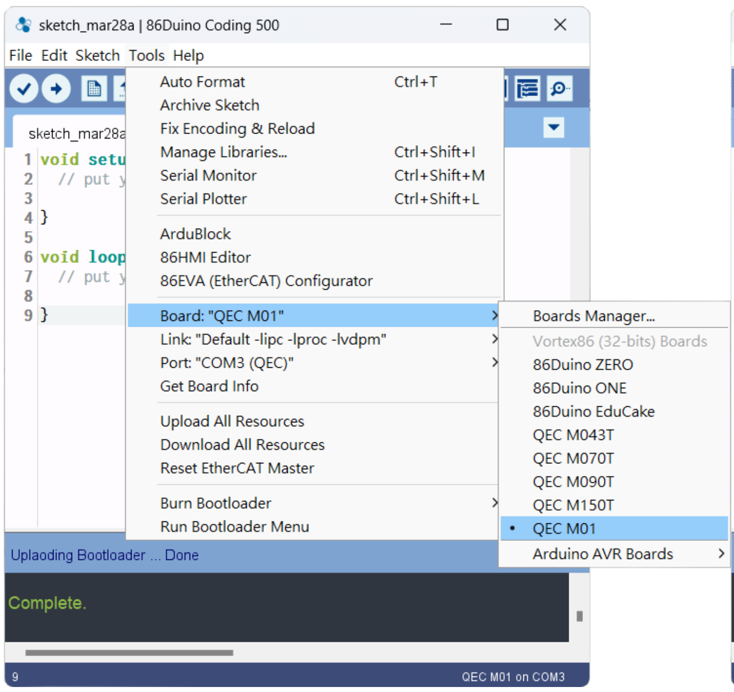
Follow the steps below to set up the environment:

1. Connect the QEC-M-01P 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)




Monitors
Network adapters
Other devices
Ports (COM & LPT)
Prolific PL2303GC USB Serial COM Port (COM7)
Print queues
Processors
Security devices
Smart card readers

4. Open the 86Duino IDE.
5. Select the correct board: In the IDE's menu, select Tools> Board > QEC-M-01 (or the QEC-M master model you use).
6. Select Port: In the IDE's menu, select Tools > Port and select the USB port to connect to the QEC-M master (in this case, COM3 (QEC)).

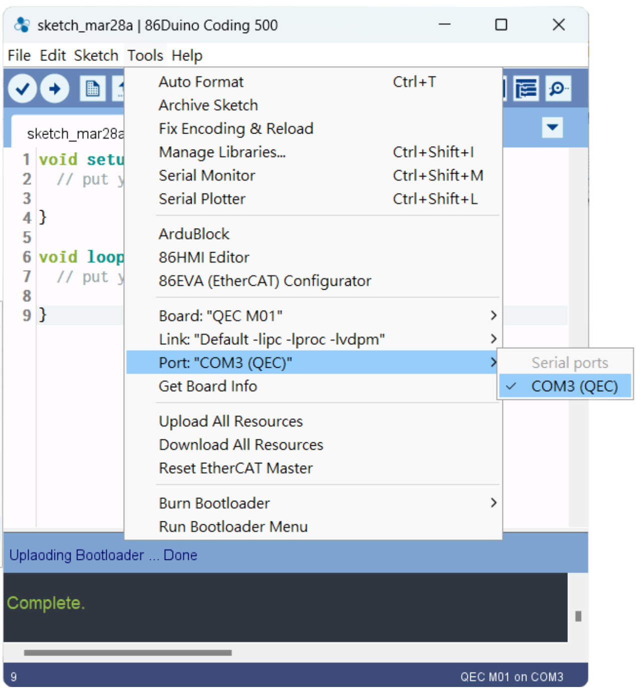

sketch_mar28a | 86Duino Coding 500

```
void setu
  // put y
}

void loop
  // put y
}
```

Uploading Bootloader ... Done

Complete.

QEC M01 on COM3


sketch_mar28a | 86Duino Coding 500

```
void setu
  // put y
}

void loop
  // put y
}
```

Uploading Bootloader ... Done

Complete.

QEC M01 on COM3

4. Use 86EVA and ArduBlock

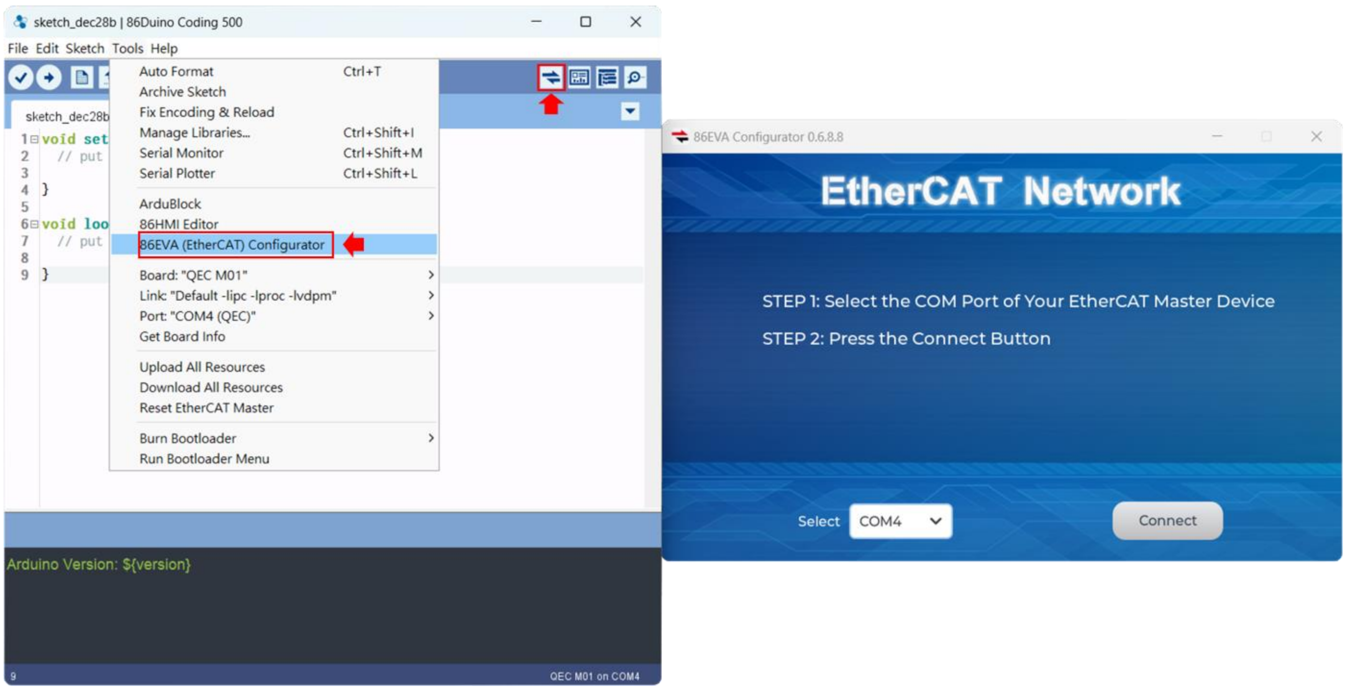
The following example is reading the data from the Serial Monitor in 86Duino IDE and transferring it from COM1 to COM2. After COM2 receives the data, we print it on the Serial Monitor.

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.

Step 1: Turn on 86EVA and scan

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

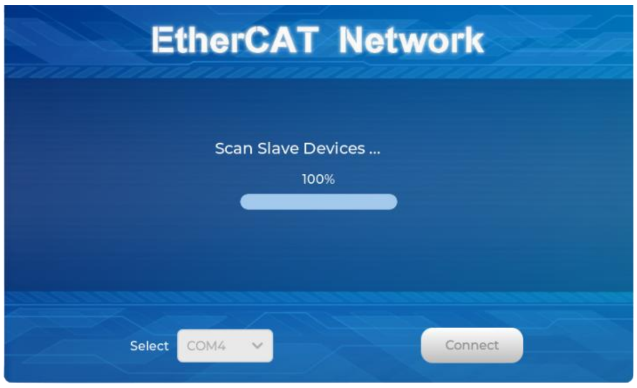


```
void set
  // put
}

void loo
  // put
}
```

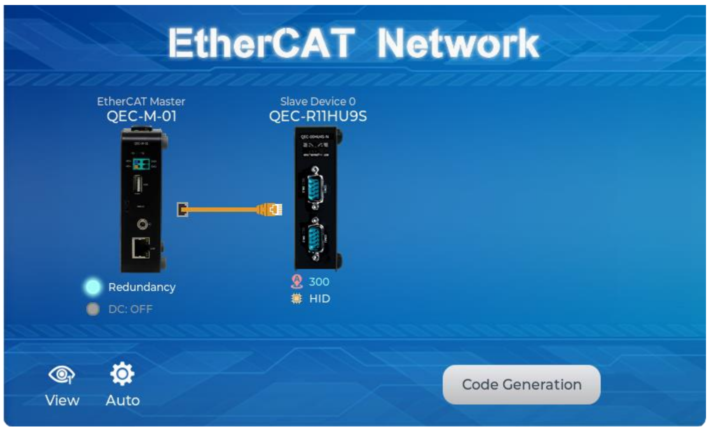
STEP 1: Select the COM Port of Your EtherCAT Master Device

STEP 2: Press the Connect Button

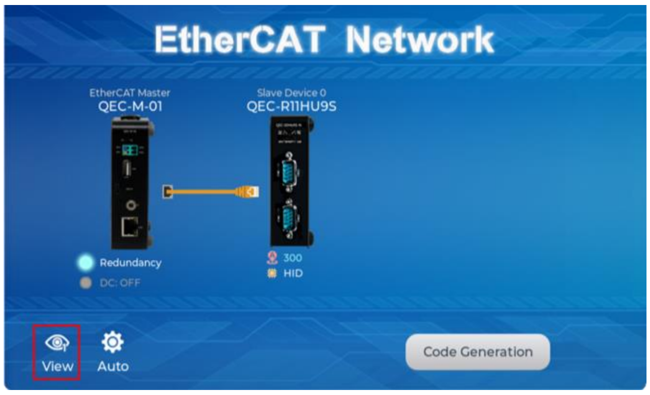
Once you have confirmed that the correct COM port has been selected of QEC-M-01P, press the Connect button to start scanning the EtherCAT network.

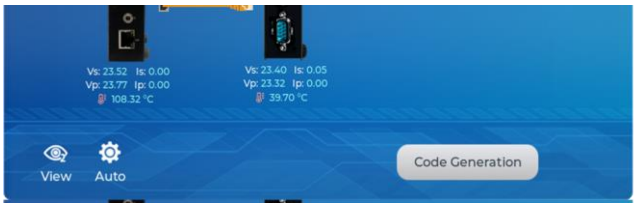


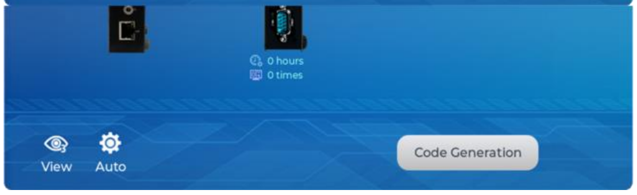
The connected devices will be displayed after the EtherCAT network has been scanned.



Press the "View" button in the lower left corner to check the device's status (Voltage, Current, and Temperature; View2) and operating time (Hours; View3).

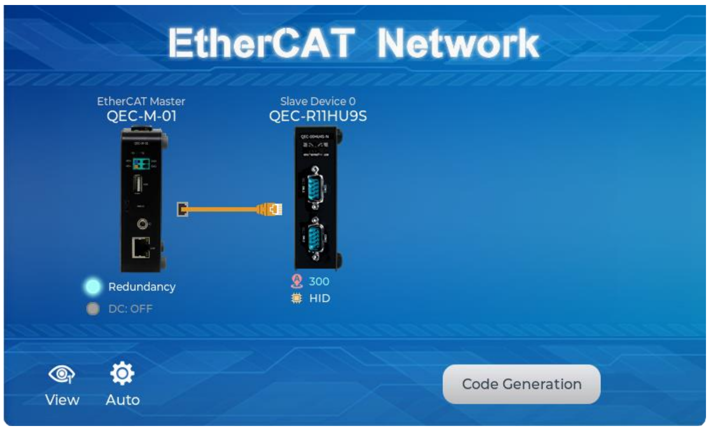






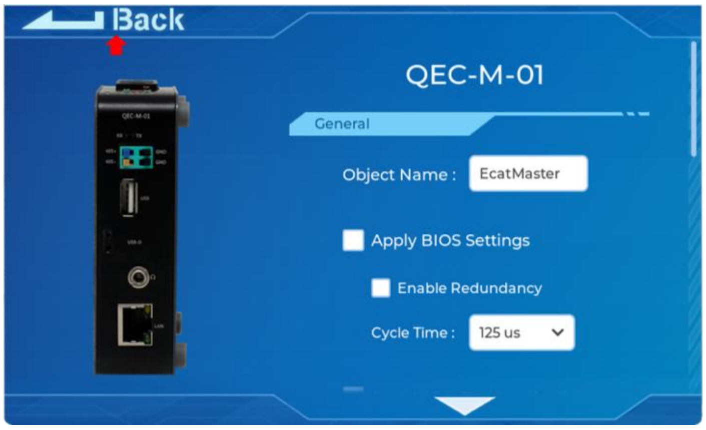
Step 2: Set the parameters

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.
This example will use the default settings and not change any settings; please click "Back" in the upper left corner to return.



Object Name : EcatMaster

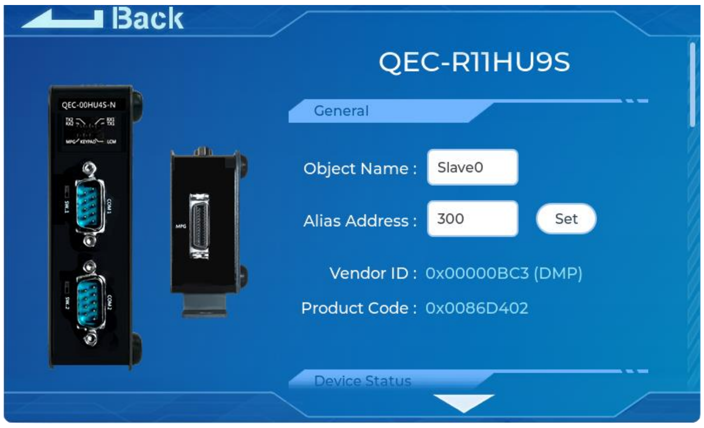
Apply BIOS Settings

Enable Redundancy

Cycle Time : 125 us

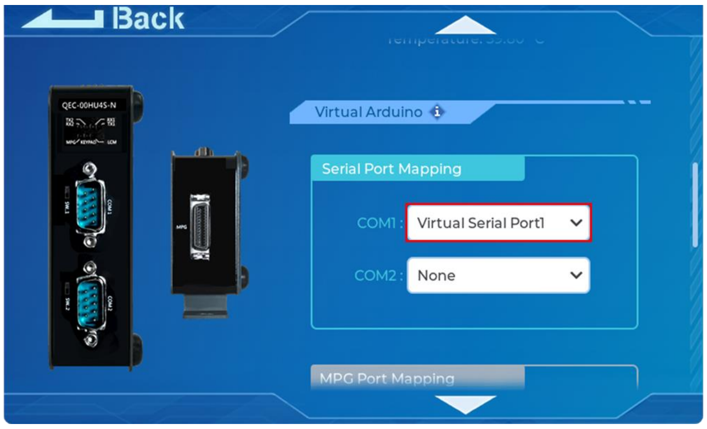
QEC-R11HU9S-N

Press twice on the image of the QEC-R11HU9S to see the parameter settings.



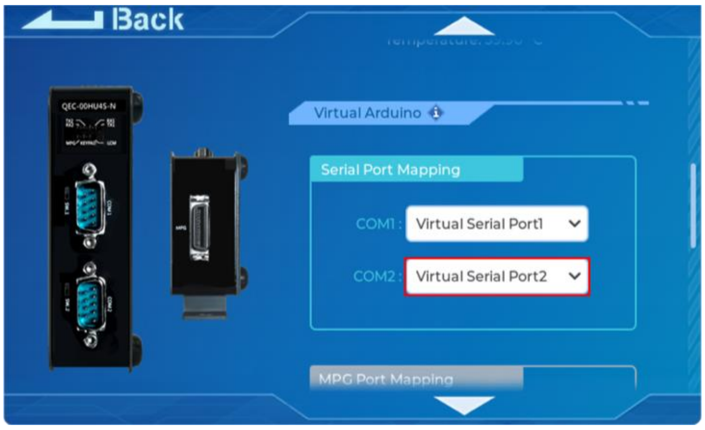
Object Name : Slave0
Alias Address : 300
Vendor ID : 0x00000BC3 (DMP)
Product Code : 0x0086D402

Go to the “Serial Port Mapping” area. Among them, we select “Virtual Serial Port1” in the drop-down box of COM1 in “Serial Port Mapping”.



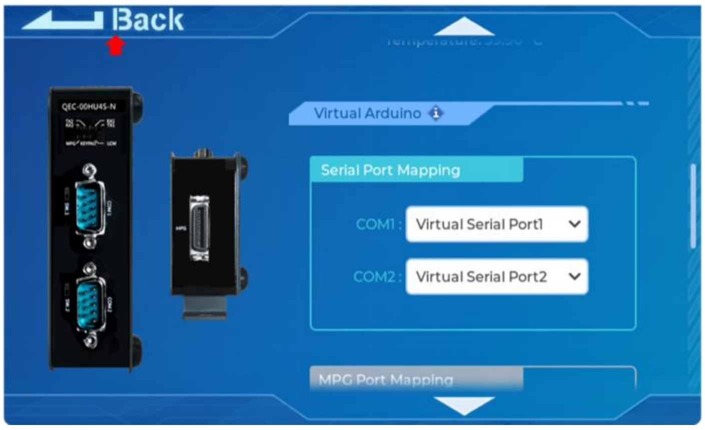
COM1 : Virtual Serial Port1
COM2 : None

And then, select “Virtual Serial Port2” in the drop-down box of COM2 in “Serial Port Mapping”.



COM1 : Virtual Serial Port1
COM2 : Virtual Serial Port2

After finishing, click "Back" in the upper left corner to return.



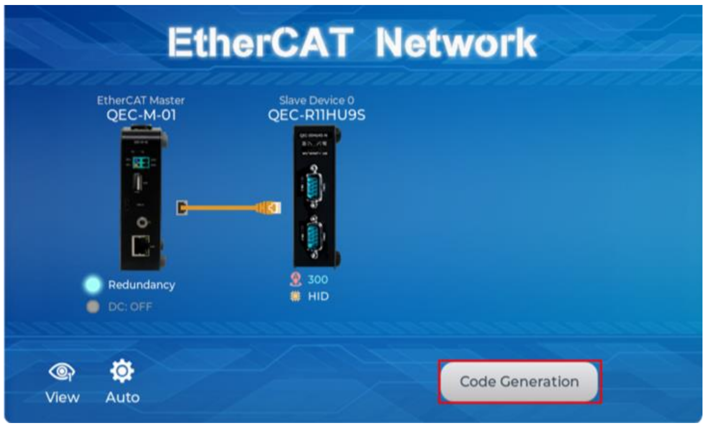
COM1 : Virtual Serial Port1

COM2 : Virtual Serial Port2

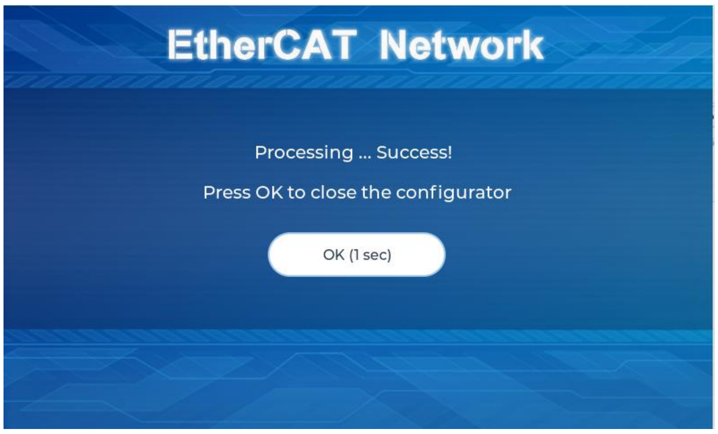
These actions are to set the QEC-R11HU9S’s COM1 to virtual serial port1 of EVA, and COM2 to virtual serial port2 of EVA.

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.

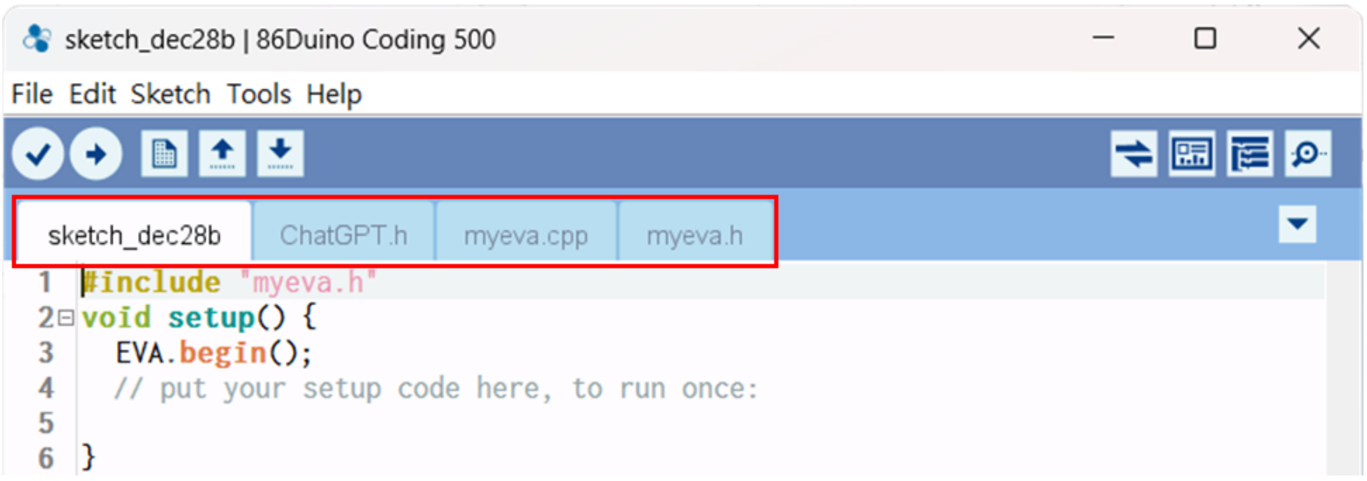


Processing ... Success!

Press OK to close the configurator

The generated code and files are as follows:

- sketch_dec28b: Main Project (.ino, depending on your project name)
- ChatGPT.h: Parameters to provide to ChatGPT referred
- myeva.cpp: C++ program code of 86EVA
- myeva.h: Header file of 86EVA


sketch_dec28b | 86Duino Coding 500
File Edit Sketch Tools Help
sketch_dec28b
ChatGPT.h
myeva.cpp
myeva.h

```
#include "myeva.h"
void setup() {
  EVA.begin();
  // put your setup code here, to run once:

}
```

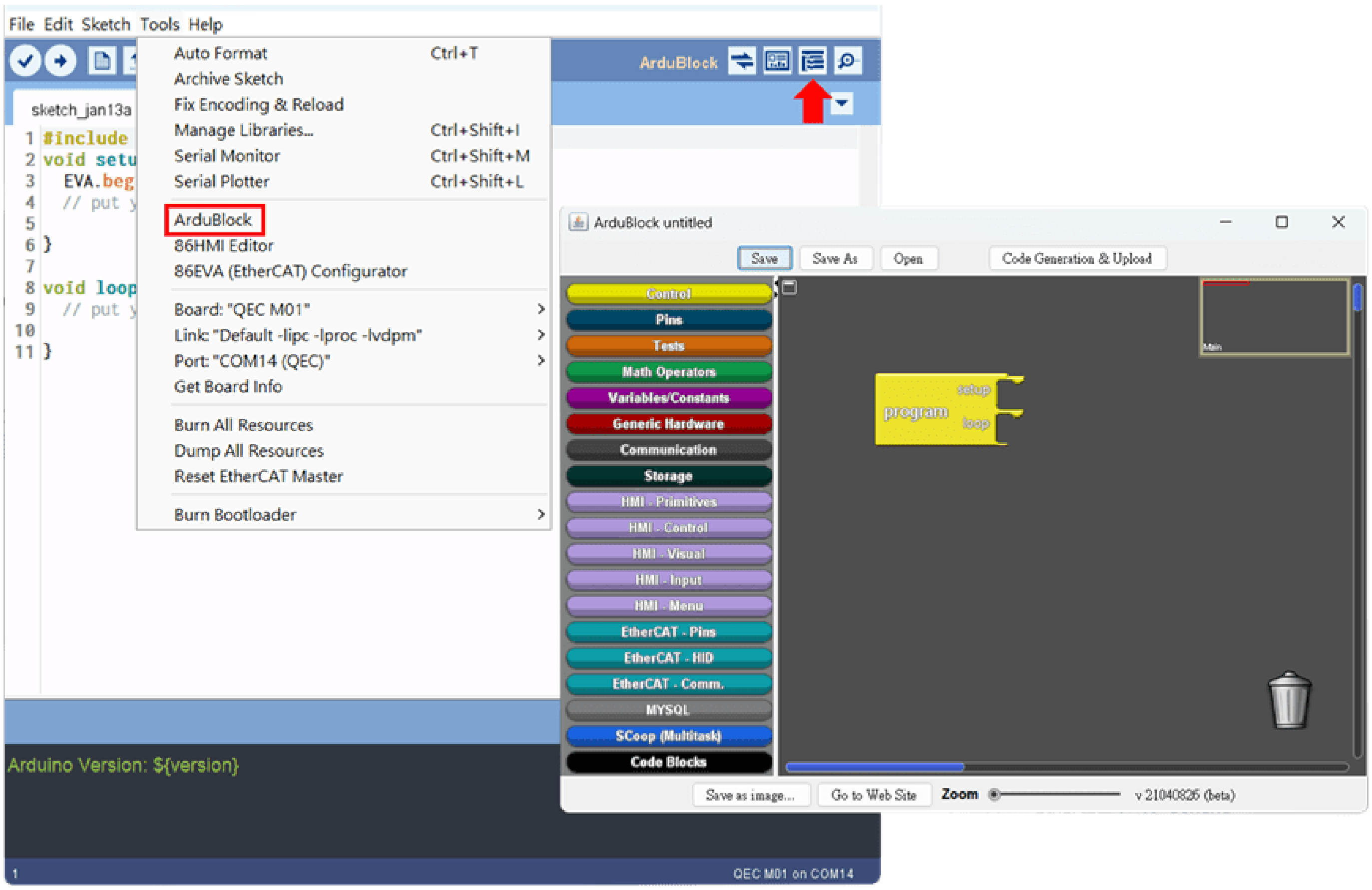
Additional note:

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

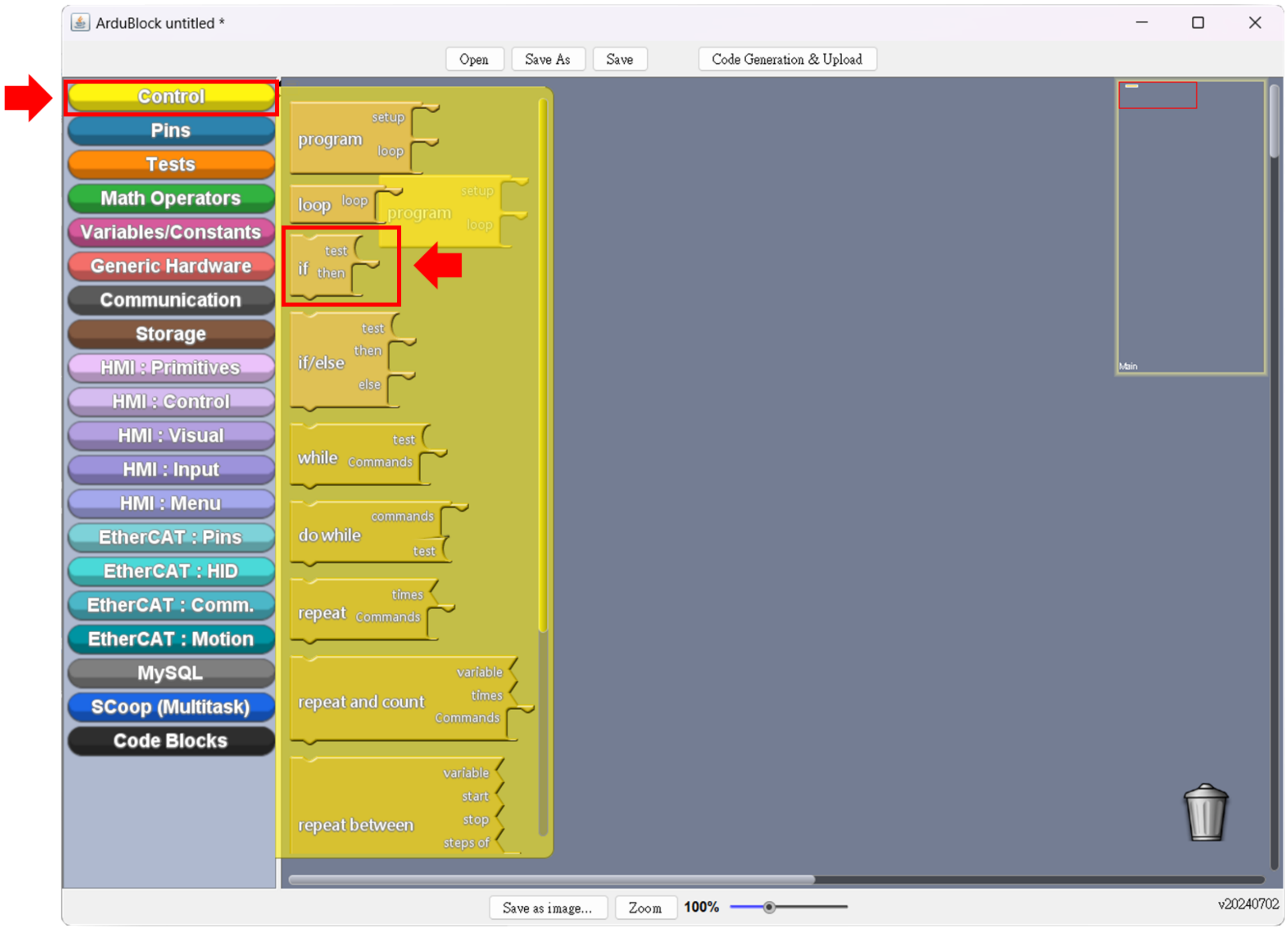
Step 4: Turn on ArduBlock and setup

Open ArduBlock.

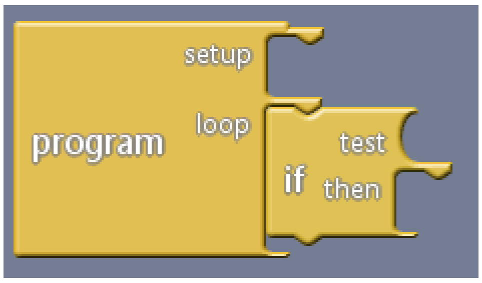


```
#include
void setu
  EVA.beg
  // put y

}

void loop
  // put y
}
```

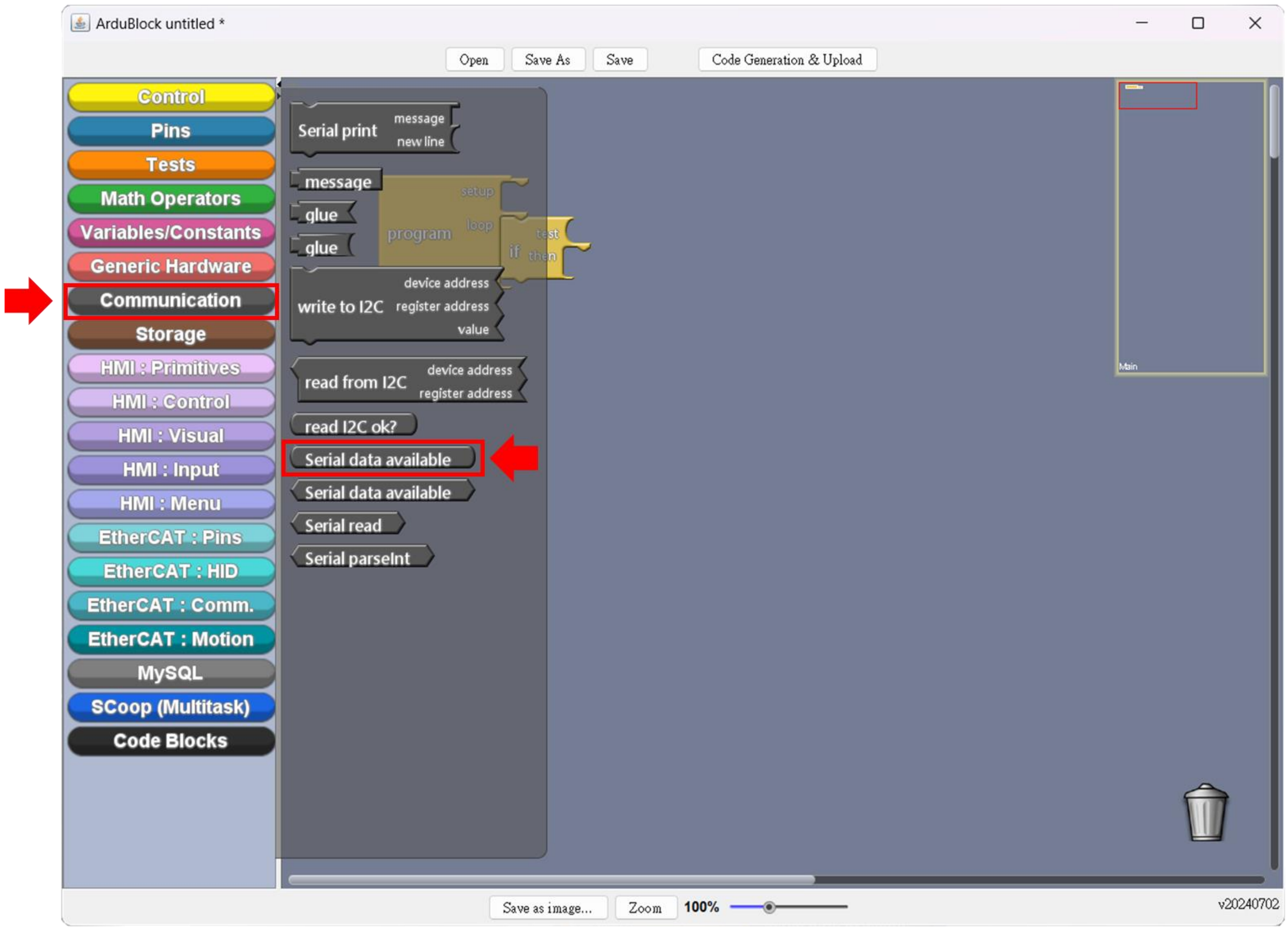
We use the "if " block from the "Control" class to the program's main loop to judge whether the Serial Monitor is available.



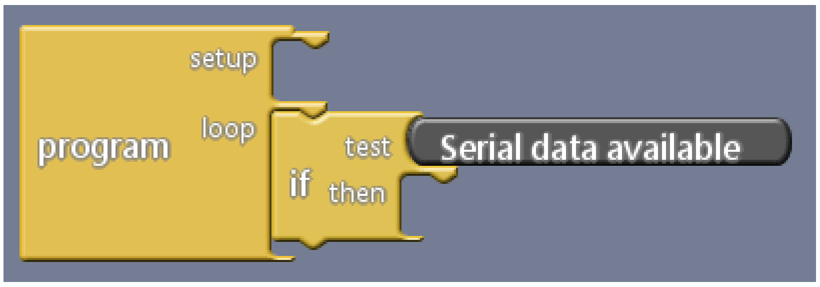
Put under the “loop” area in the “program” block.



Then, we use the "Serial data available" block from the "Communication" class.



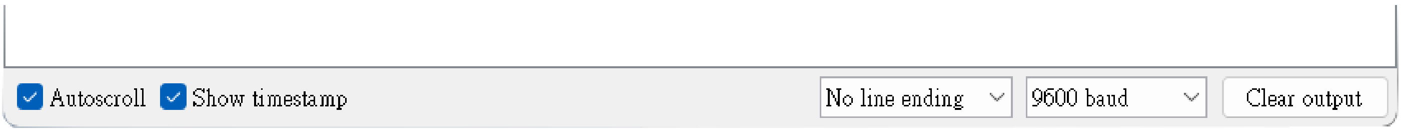
And put in the "if " block test area.



The "if " will judge the "Serial data available". If it returns true, it will execute the program.

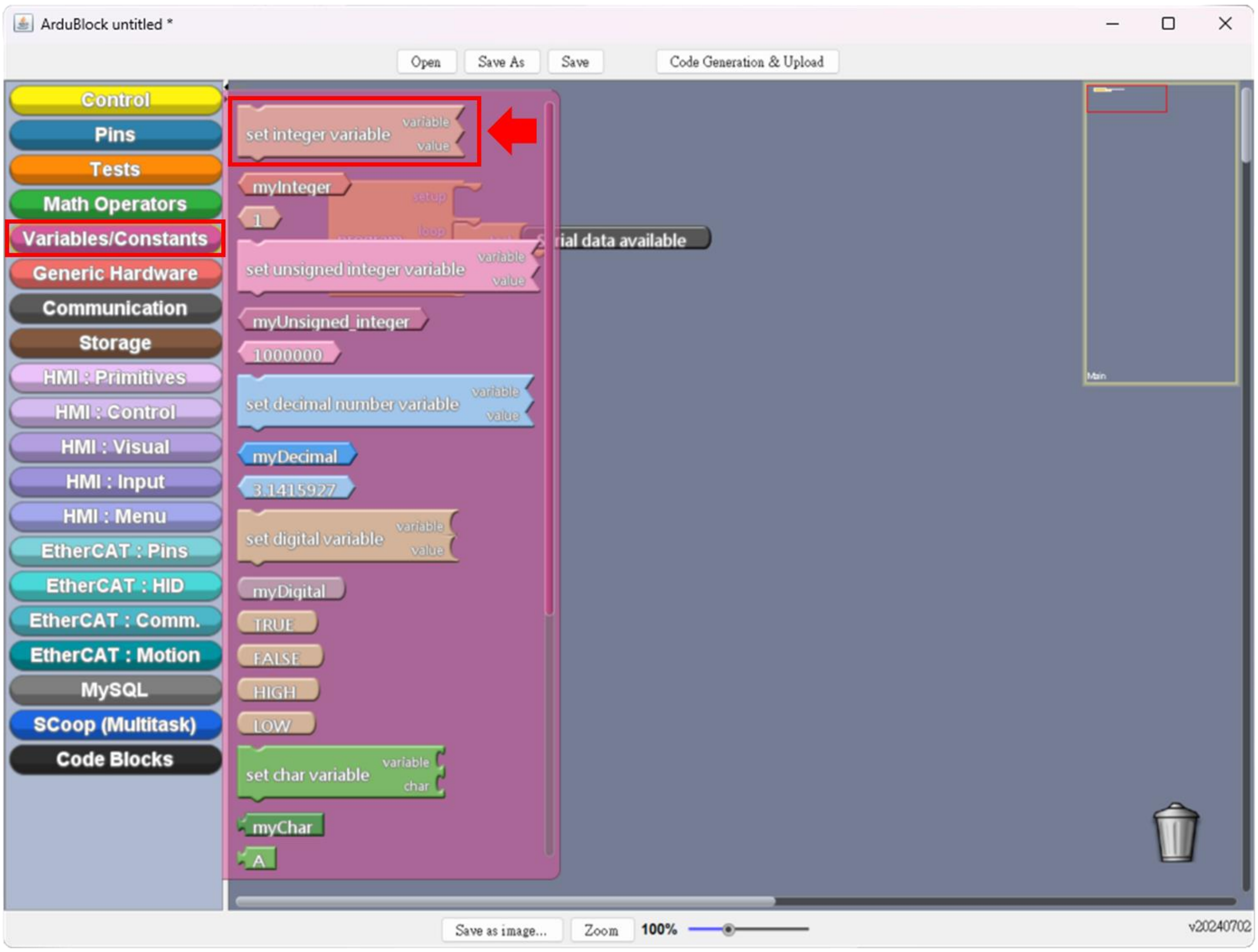
Note:

The default baud-rate of Serial monitor is 9600.

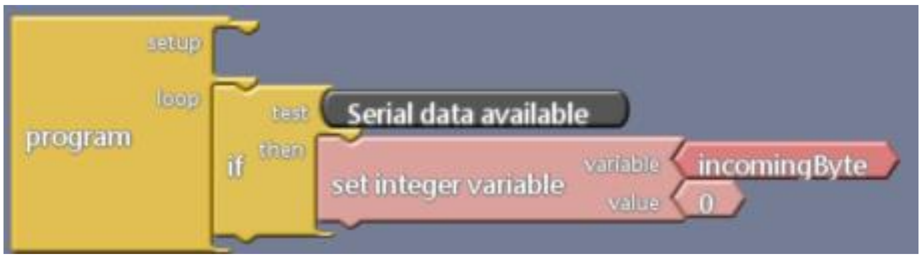


In the then area of “if” block, we will start processing the HID RS232 communication.

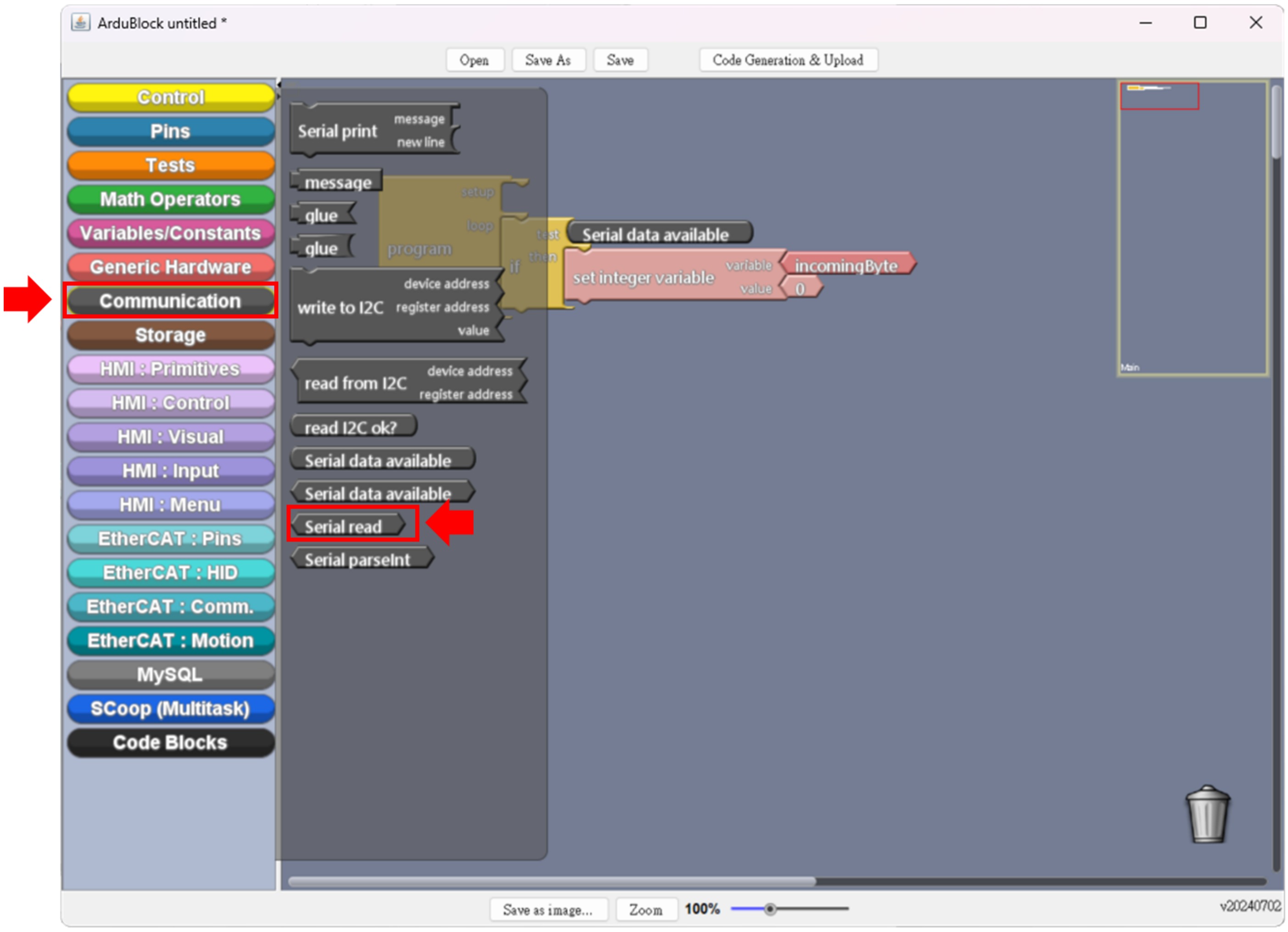
First, we use the “set integer variable” block from the “Variables/Constants” class to create an input value received from Serial Monitor.



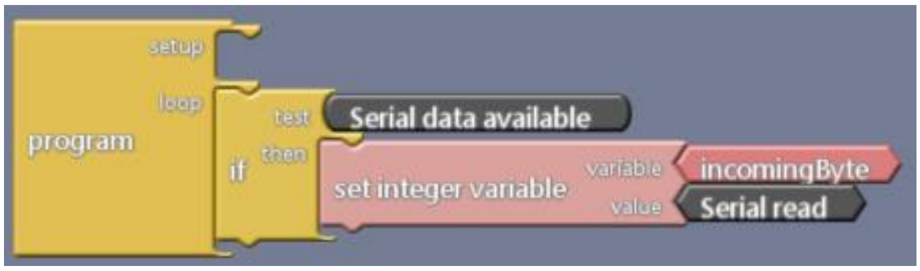
Put in the "if " block then area and change the variable name to “incomingByte”.



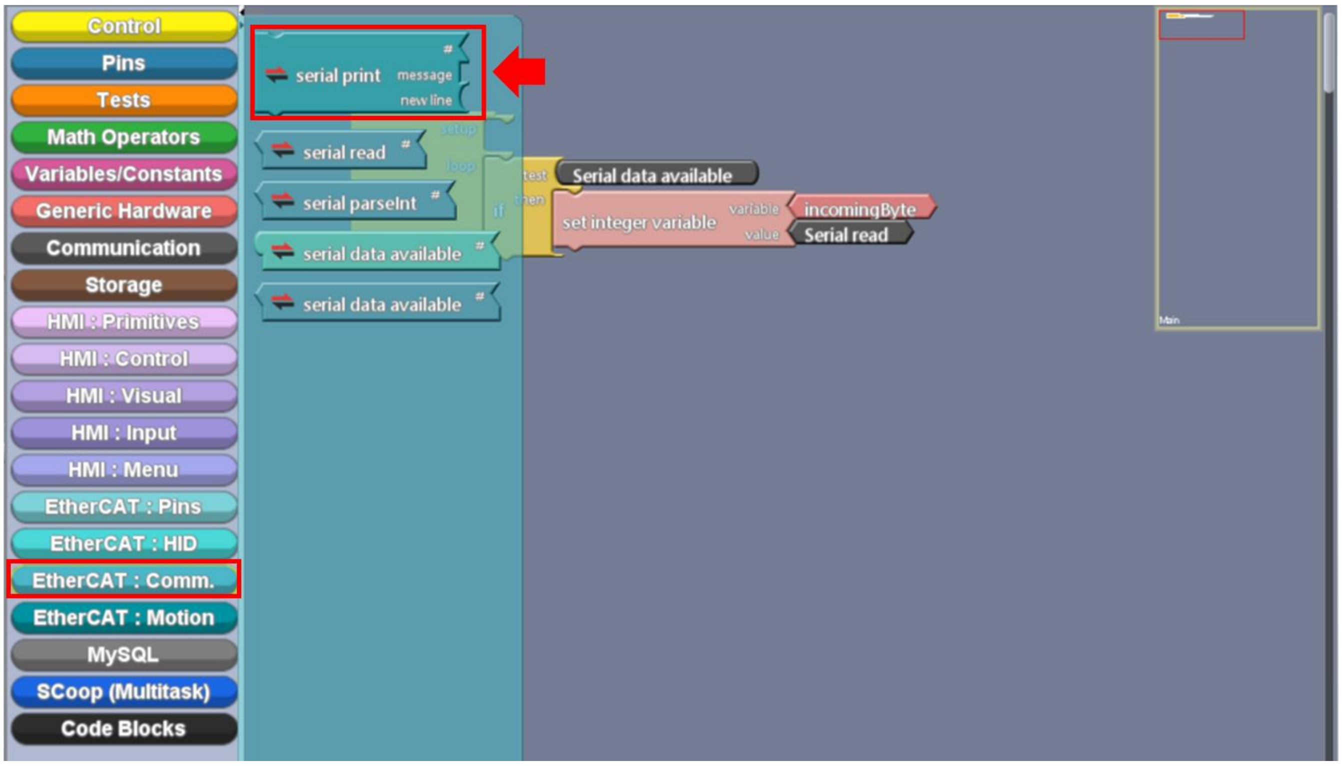
We use the “Serial read” block from the “Communication” class, and put it in the value area of “set integer variable” block.



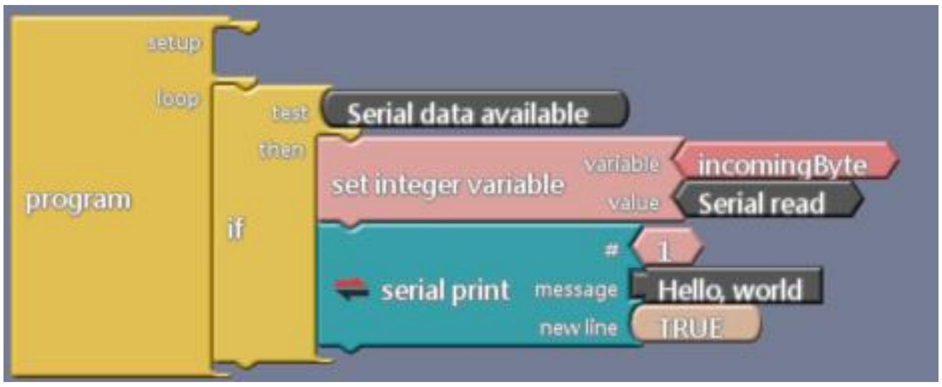
Like this.



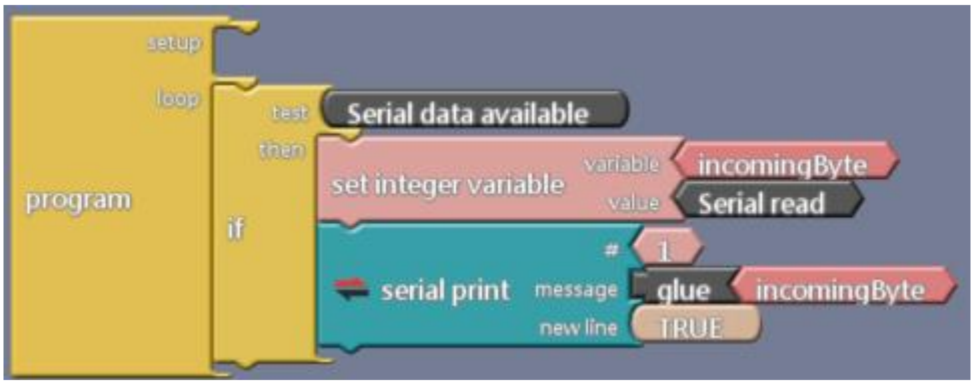
Next, we use “serial print” block from “EtherCAT : Comm.” Class, to send data from Serial Monitor input through “Virtual Serial Port1” (we select “Virtual Serial Port1” for HID COM1).



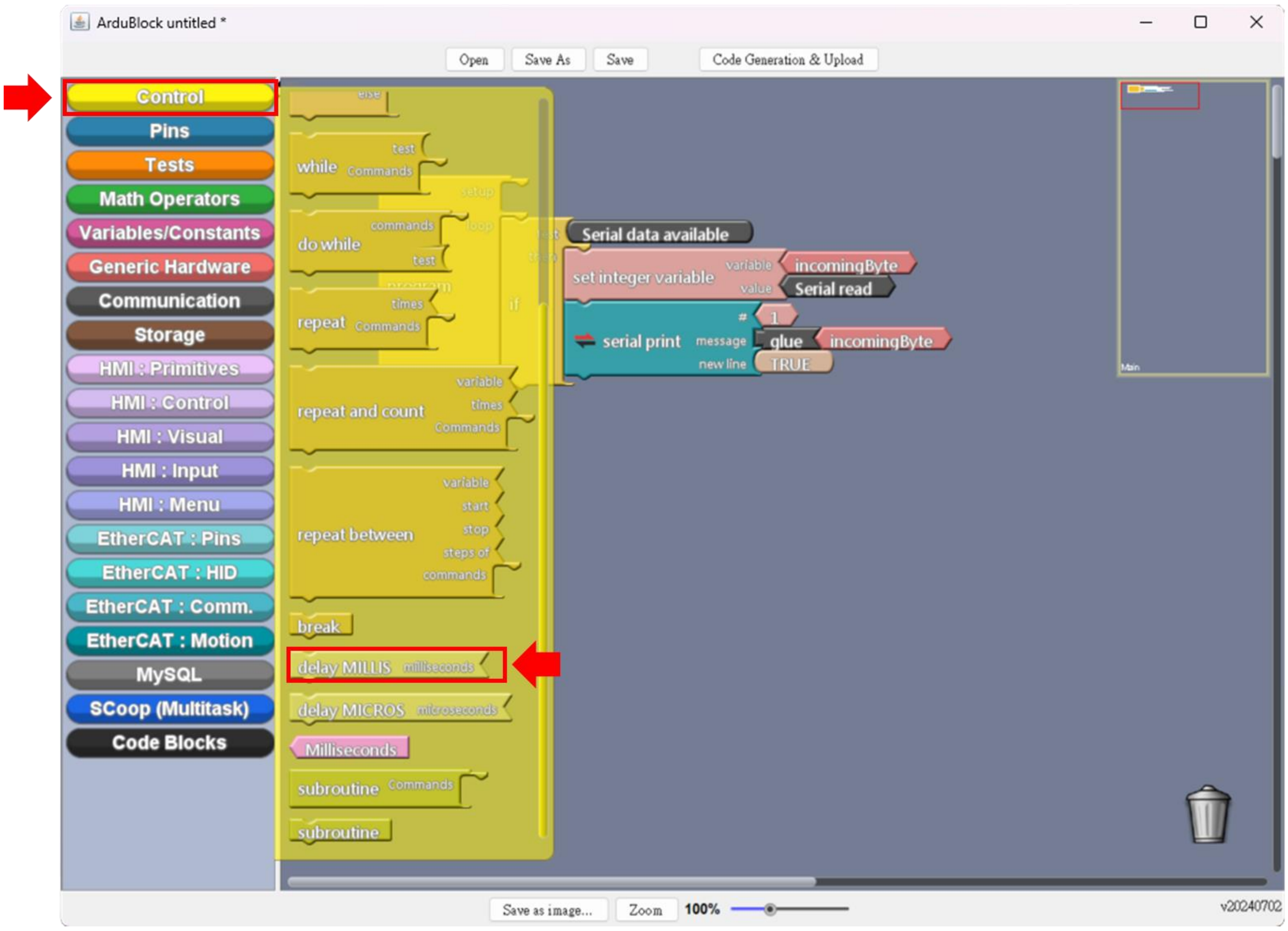
Like this.


program
setup
loop
if
test Serial data available
then
set integer variable variable incomingByte value Serial read
serial print # 1 message Hello, world new line TRUE

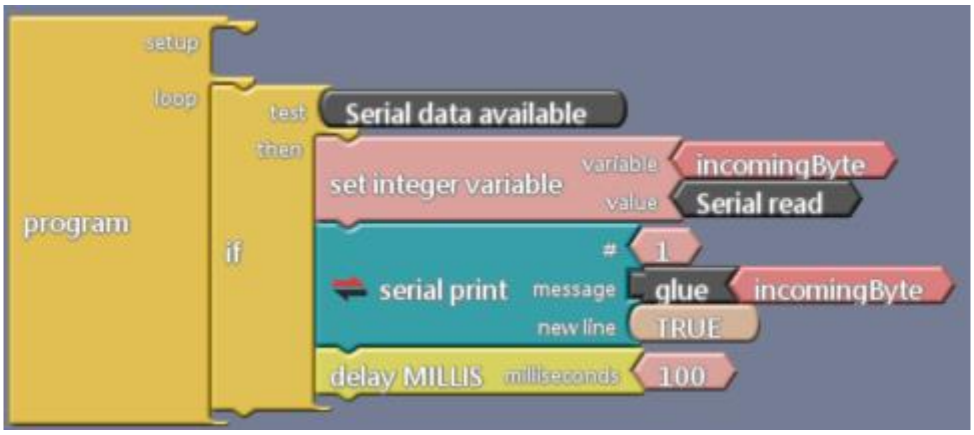
And we need to change the message content from “Hello, world” to the variable incomingByte. We use “glue” block from “Communication” Class to glue the incomingByte to the message area of “serial print”.


program
setup
loop
if
test Serial data available
then
set integer variable variable incomingByte value Serial read
serial print # 1 message glue incomingByte new line TRUE

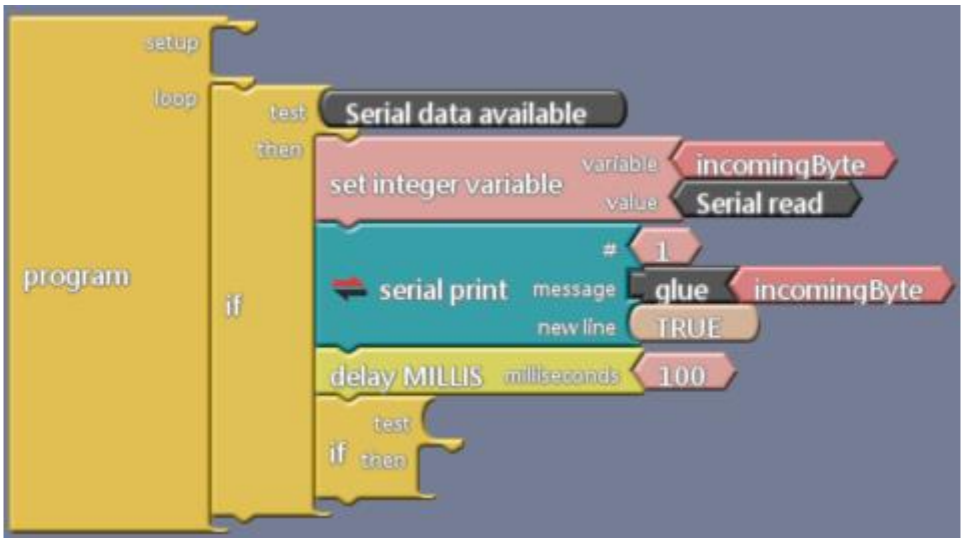
Because we need to wait for the serial communication time, we put a "delay MILLIS" block in the "Control" class.



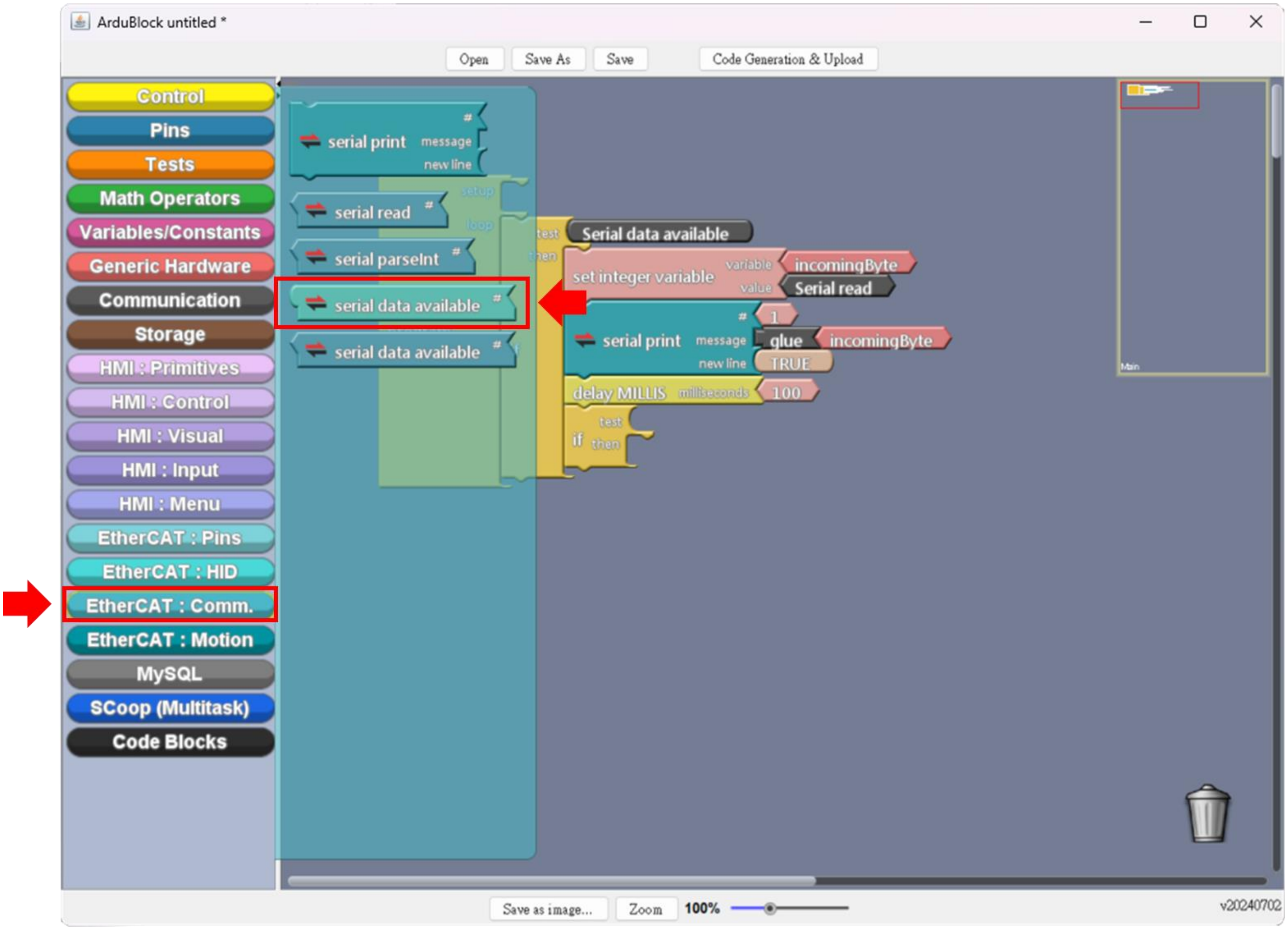
In this case, we delay for 100 milliseconds.



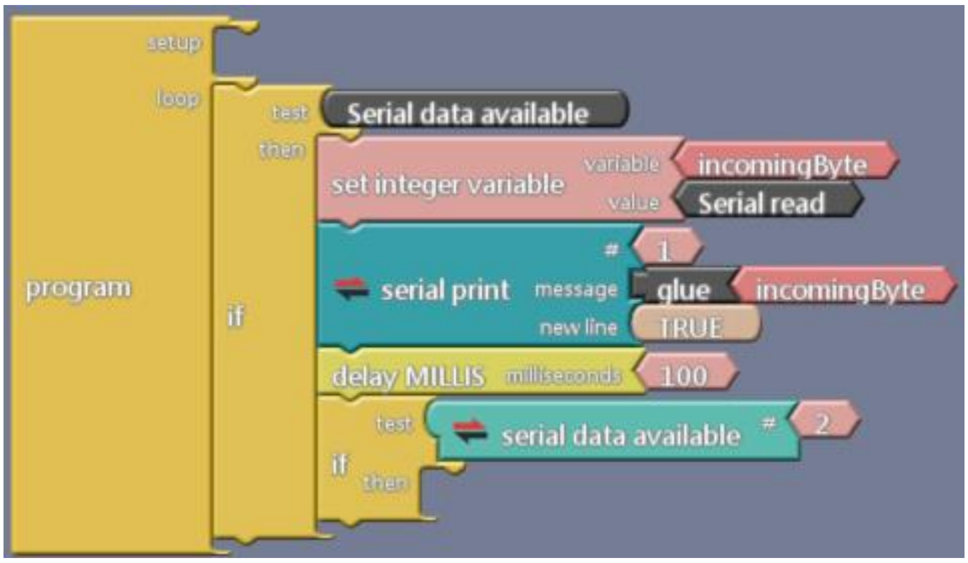
Then, we must judge whether the "Virtual Serial Port2" received the data from "Virtual Serial Port1". So, we create an "if" block after the "delay MILLIS" block.



In the test area in “if” block, we use “serial data available” block from “EtherCAT : Comm.” Class.

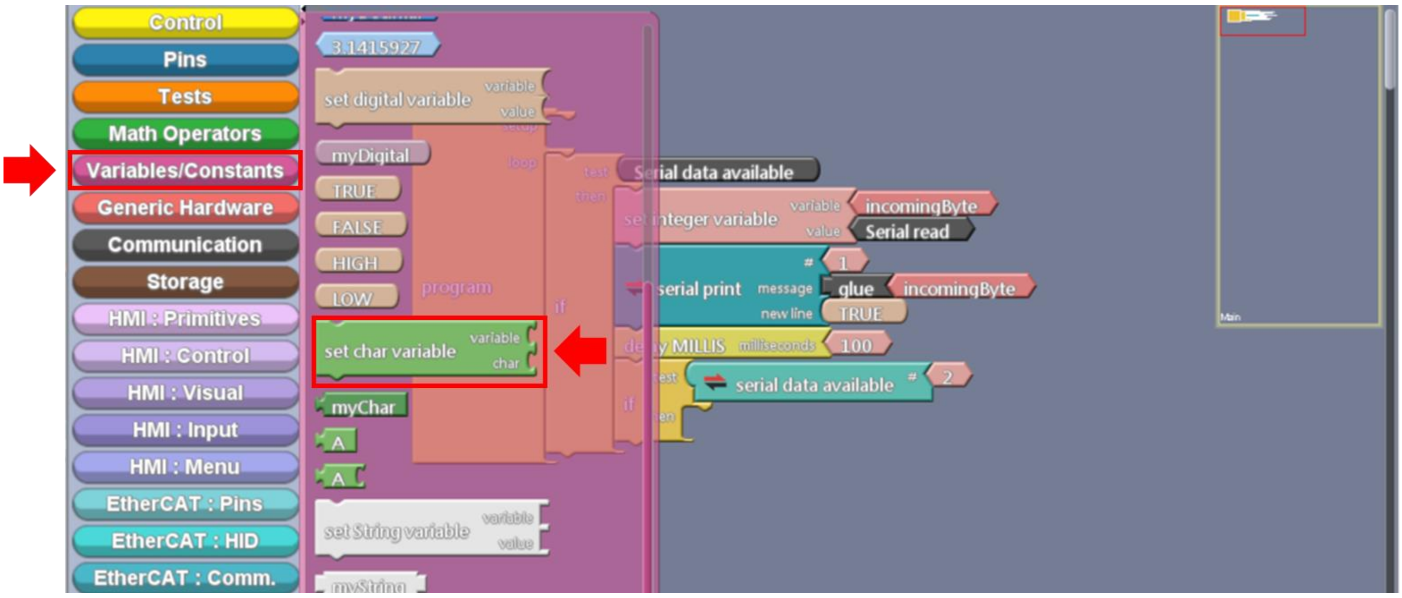


And set the "#" number to 2, to let "serial data available" to read the "Virtual Serial Port 2" (we select "Virtual Serial Port2" for HID COM2).

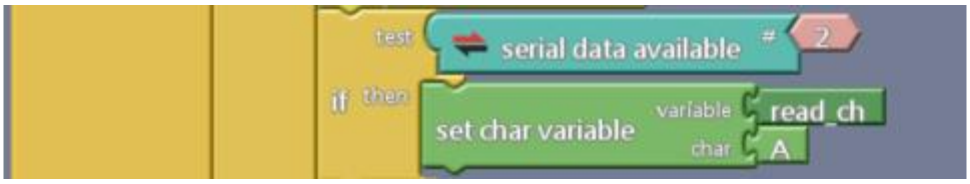


The "if" will judge whether the "Virtual Serial Port2" is available. If it returns true, it will execute the program.

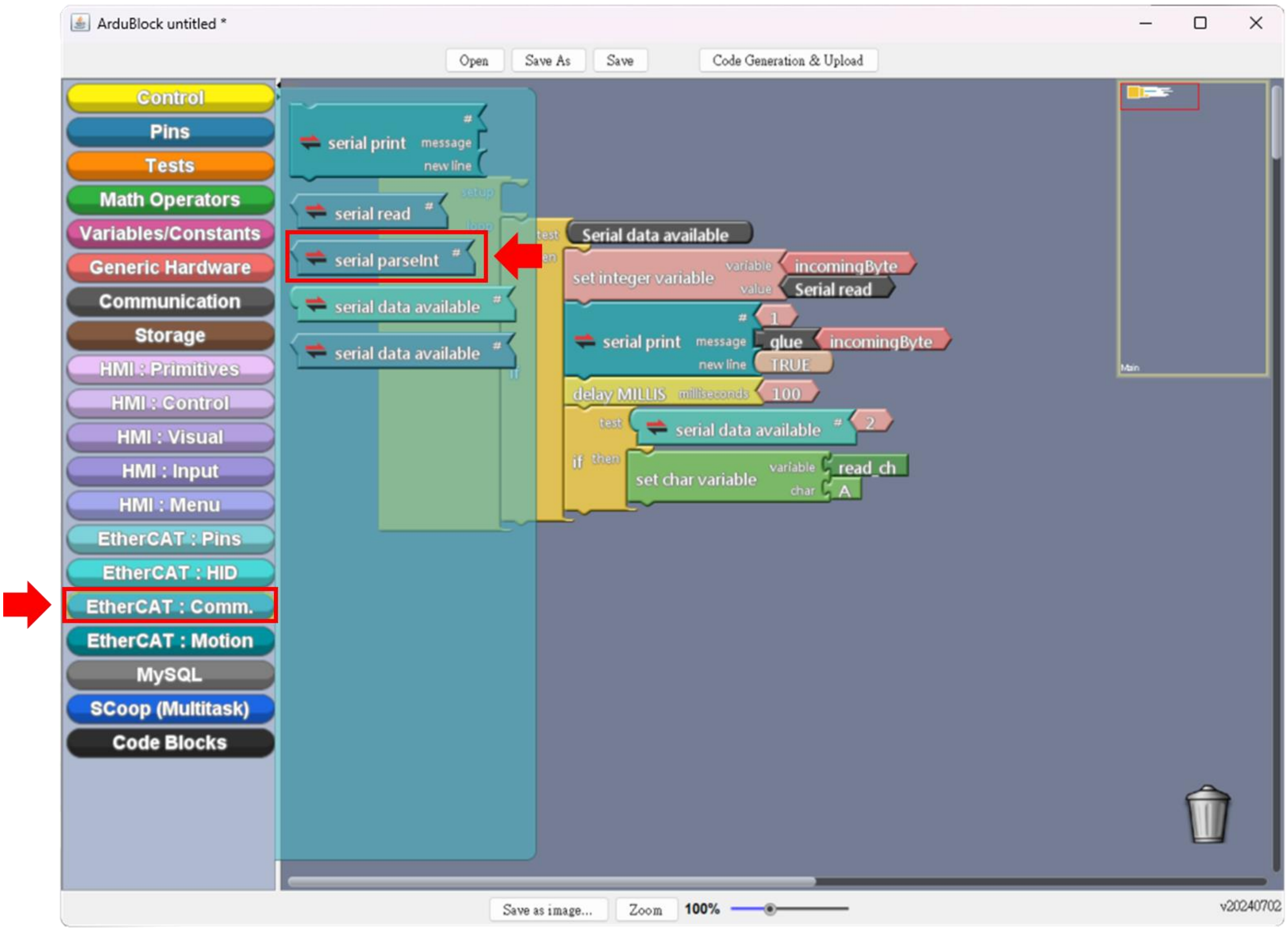
In then area, we will read the "Virtual Serial Port2" received data, and set it in a char variable, which will call "read_ch".
First, we use "set char variable" block from "Variables/Constants" Class, and set the variable name to "read_ch".



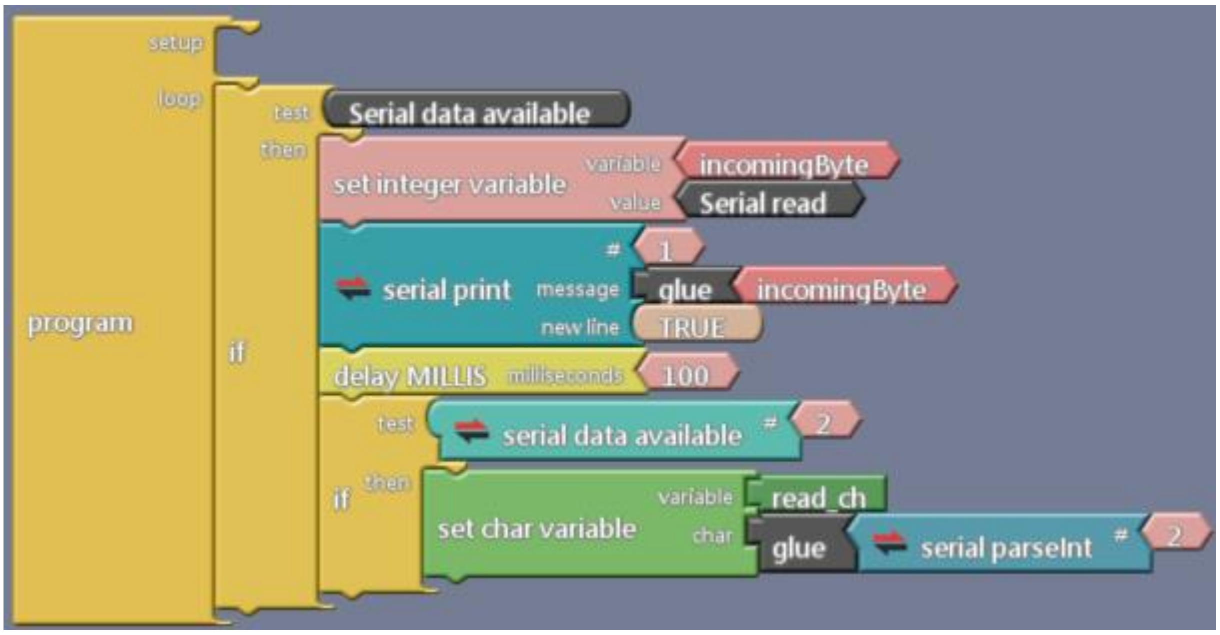
Like this.



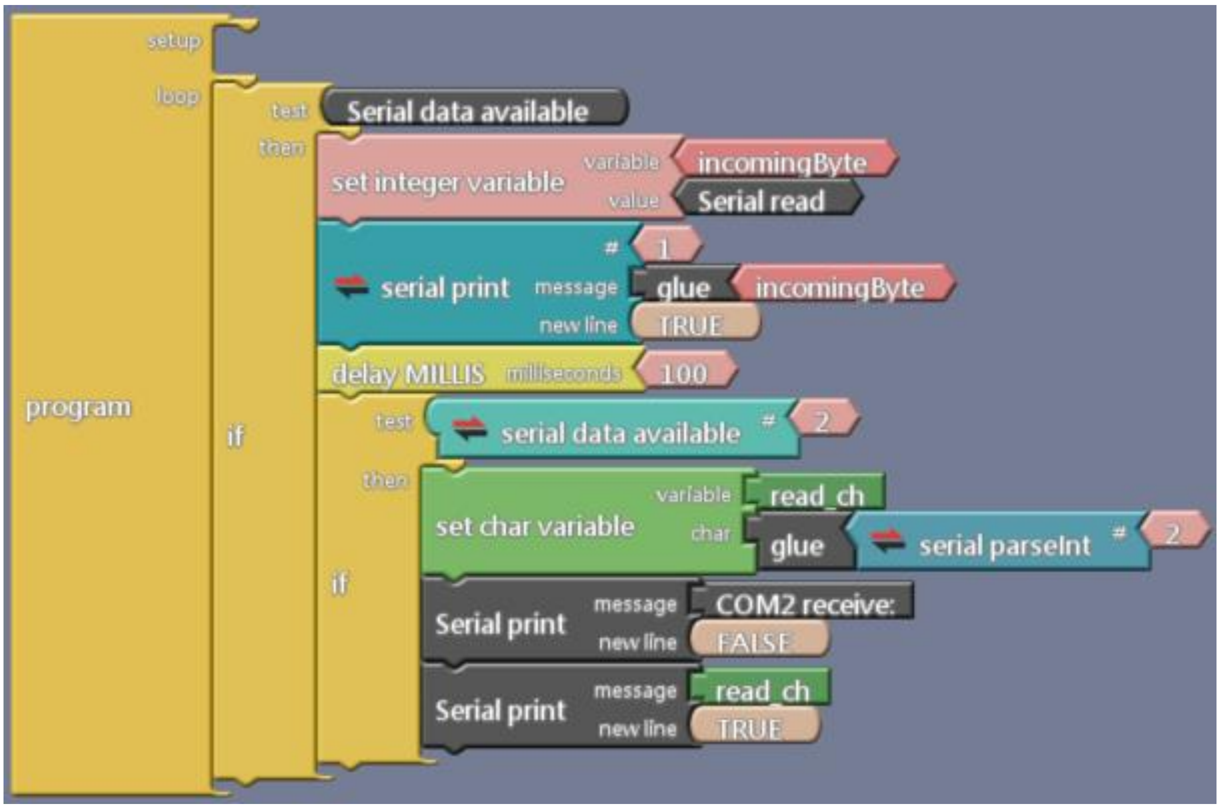
And we use “glue” block from “Communication” Class to glue the received data from “Virtual Serial Port2” by using “serial parseInt" from "EtherCAT : Comm" to the char area of “set char variable”.



Like this.

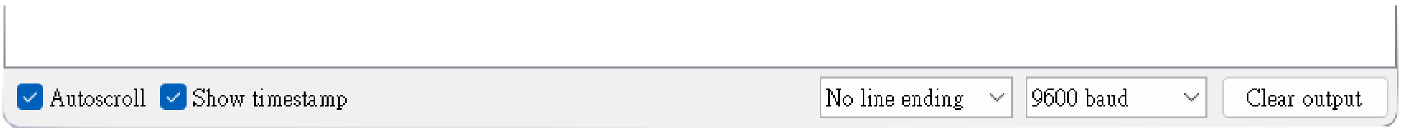


We use the “Serial print” block to print out the char variable read_ch (the data received from “Virtual Serial Port2”) in Serial Monitor.

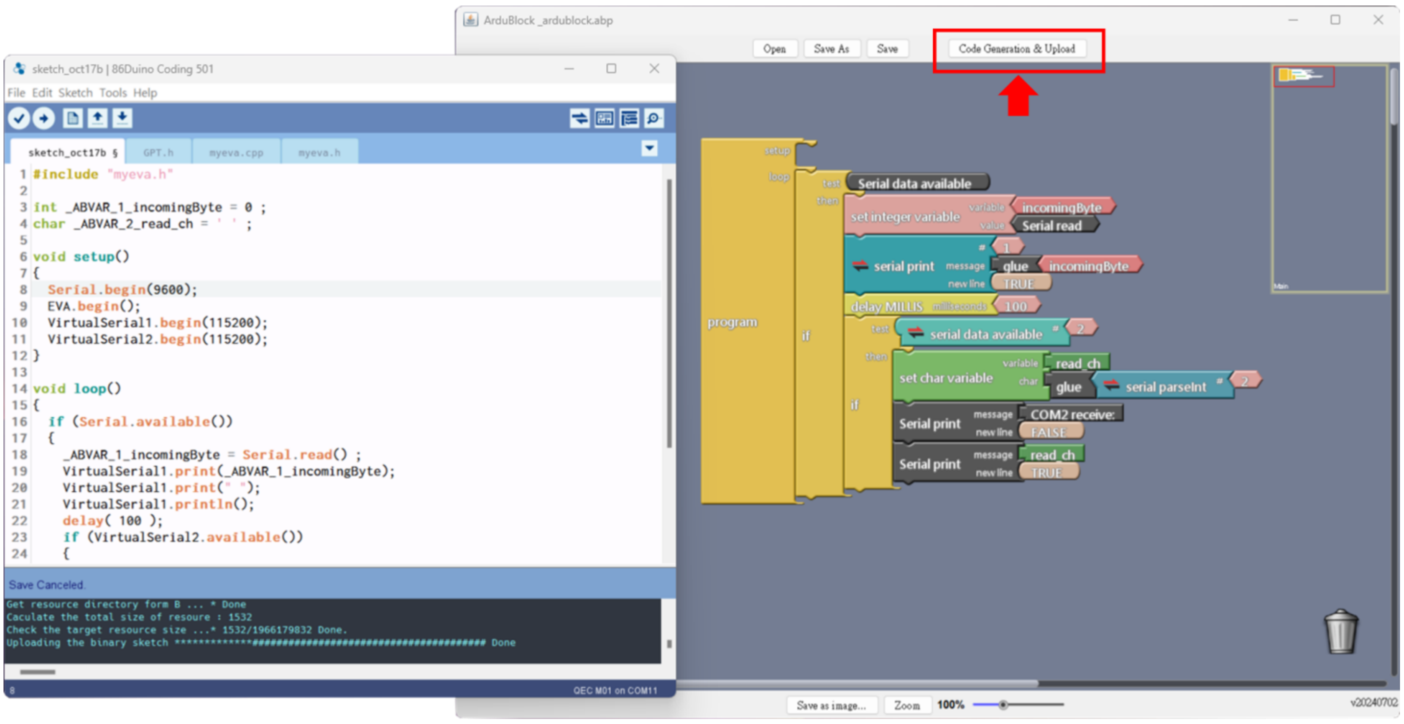


Note:

The default baud-rate of Serial monitor is 9600.



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

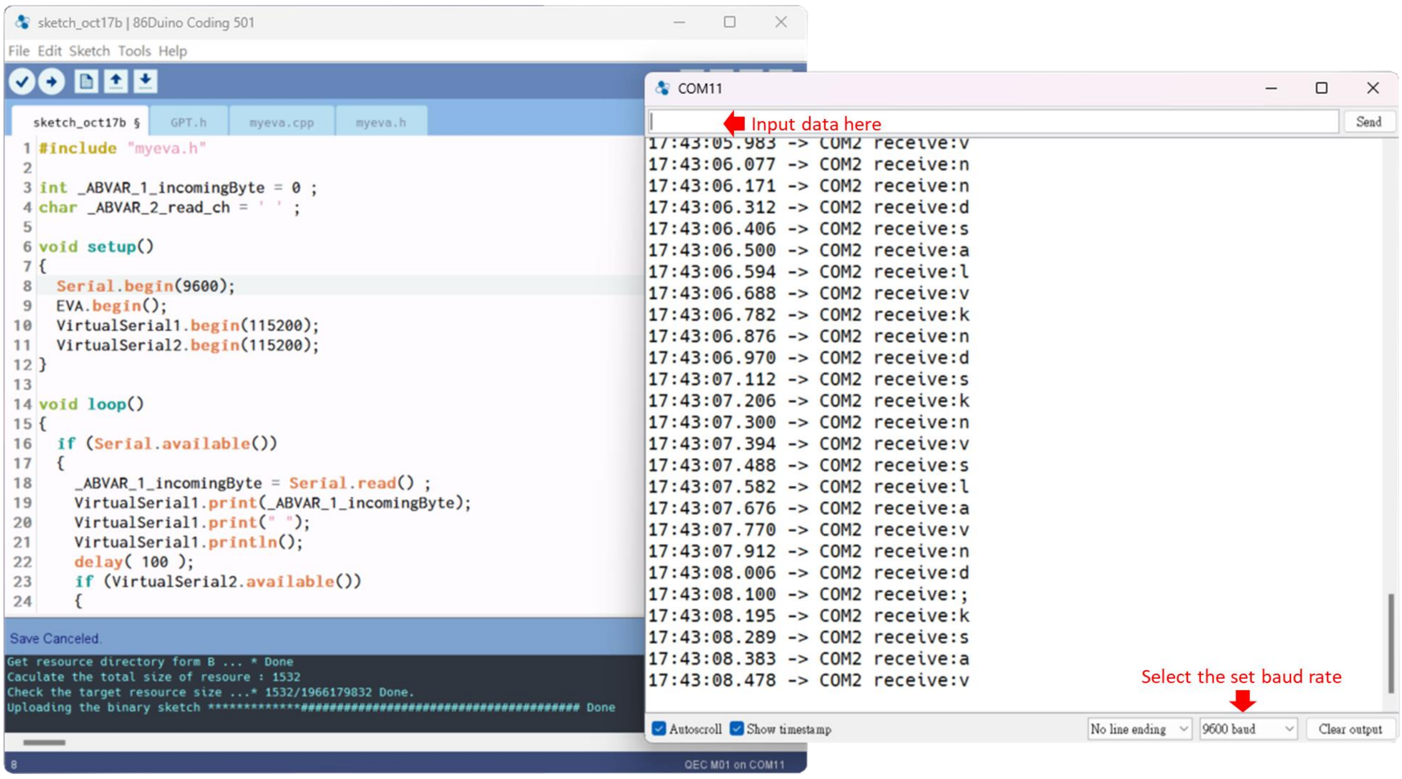


```
#include "myeva.h"

int _ABVAR_1_incomingByte = 0 ;
char _ABVAR_2_read_ch = ' ' ;

void setup()
{
  Serial.begin(9600);
  EVA.begin();
  VirtualSerial1.begin(115200);
  VirtualSerial2.begin(115200);
}

void loop()
{
  if (Serial.available())
  {
    _ABVAR_1_incomingByte = Serial.read() ;
    VirtualSerial1.print(_ABVAR_1_incomingByte);
    VirtualSerial1.print(" ");
    VirtualSerial1.println();
    delay( 100 );
    if (VirtualSerial2.available())
    {
```

```
Save Canceled.
Get resource directory form B ... * Done
Caculate the total size of resoure : 1532
Check the target resource size ...* 1532/1966179832 Done.
Uploading the binary sketch *************########################################### Done
```

After uploading, you can input a number or letter to the Serial Monitor in 86Duino IDE.

All data will transfer from COM1 to COM2. After COM2 receives the data, we print it on the Serial Monitor, as in the image below.



```
#include "myeva.h"

int _ABVAR_1_incomingByte = 0 ;
char _ABVAR_2_read_ch = ' ' ;

void setup()
{
  Serial.begin(9600);
  EVA.begin();
  VirtualSerial1.begin(115200);
  VirtualSerial2.begin(115200);
}

void loop()
{
  if (Serial.available())
  {
    _ABVAR_1_incomingByte = Serial.read() ;
    VirtualSerial1.print(_ABVAR_1_incomingByte);
    VirtualSerial1.print(" ");
    VirtualSerial1.println();
    delay( 100 );
    if (VirtualSerial2.available())
    {
```

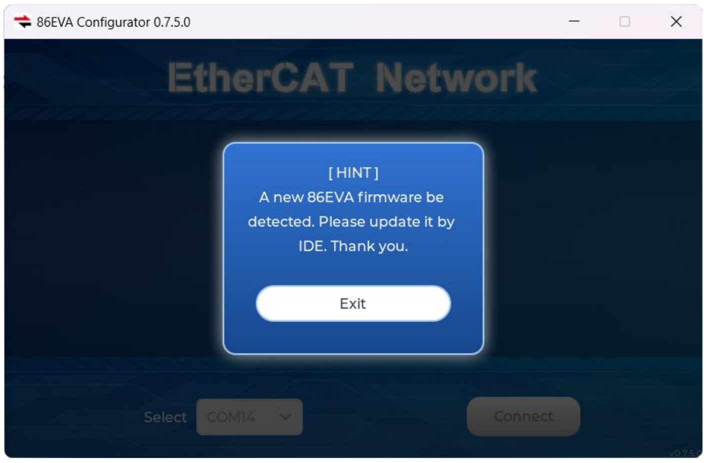
```
Save Canceled.
Get resource directory form B ... * Done
Caculate the total size of resoure : 1532
Check the target resource size ...* 1532/1966179832 Done.
Uploading the binary sketch *************######################################## Done
```

```
17:43:05.983 -> COM2 receive:v
17:43:06.077 -> COM2 receive:n
17:43:06.171 -> COM2 receive:n
17:43:06.312 -> COM2 receive:d
17:43:06.406 -> COM2 receive:s
17:43:06.500 -> COM2 receive:a
17:43:06.594 -> COM2 receive:l
17:43:06.688 -> COM2 receive:v
17:43:06.782 -> COM2 receive:k
17:43:06.876 -> COM2 receive:n
17:43:06.970 -> COM2 receive:d
17:43:07.112 -> COM2 receive:s
17:43:07.206 -> COM2 receive:k
17:43:07.300 -> COM2 receive:n
17:43:07.394 -> COM2 receive:v
17:43:07.488 -> COM2 receive:s
17:43:07.582 -> COM2 receive:l
17:43:07.676 -> COM2 receive:a
17:43:07.770 -> COM2 receive:v
17:43:07.912 -> COM2 receive:n
17:43:08.006 -> COM2 receive:d
17:43:08.100 -> COM2 receive:;
17:43:08.195 -> COM2 receive:k
17:43:08.289 -> COM2 receive:s
17:43:08.383 -> COM2 receive:a
17:43:08.478 -> COM2 receive:v
```

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 Master's environment is abnormal. As shown in the figure below, please try updating your QEC EtherCAT Master's environment, which will include the following three items: Bootloader, EtherCAT firmware, and EtherCAT tool.


86EVA Configurator 0.7.5.0

[HINT]
A new 86EVA firmware be detected. Please update it by IDE. Thank you.

Now, we will further explain how to proceed with the update:

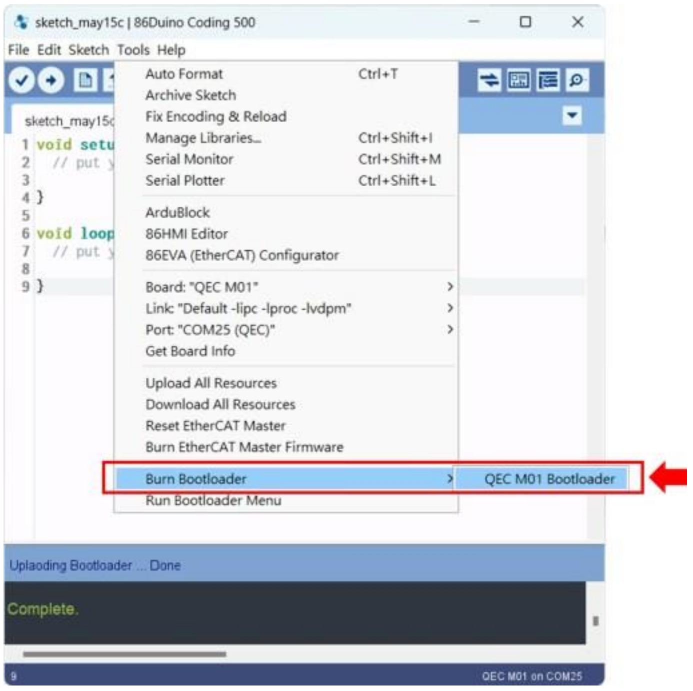
Step 1: Setting up QEC-M

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

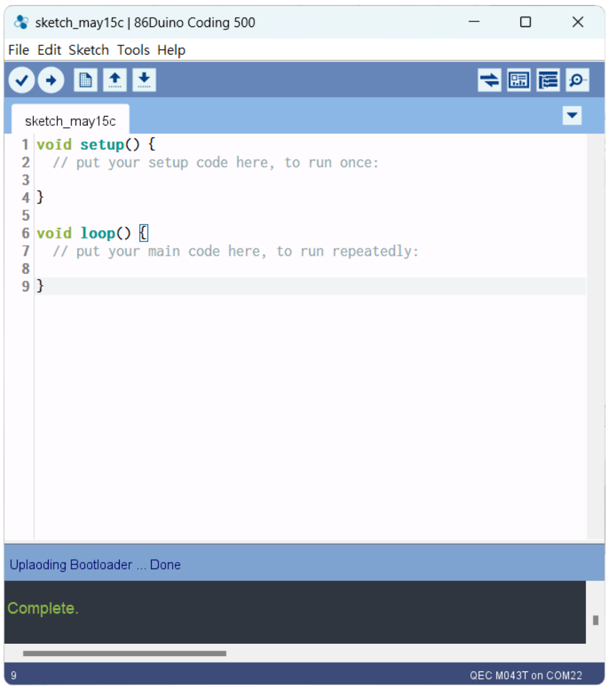
After connecting to your QEC-M product, go to “Tools”> “Burn Bootloader”. The currently selected QEC-M name will appear. Clicking on it will start the update process, which will take approximately 5-20 minutes.

QEC-M-01:



```
Uplaoding Bootloader ... Done
Complete.
```

Step 3: Complete the Update


sketch_may15c | 86Duino Coding 500
File Edit Sketch Tools Help
sketch_may15c
QEC M043T on COM22

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
Uplaoding Bootloader ... Done
Complete.
```

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.