ICOP Technology Inc.

Start Guide

EK1814 : EtherCAT Coupler with integrated digital inputs/outputs

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

Revision

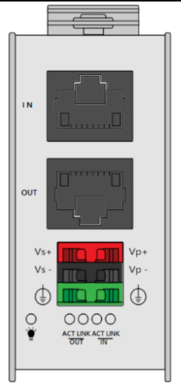
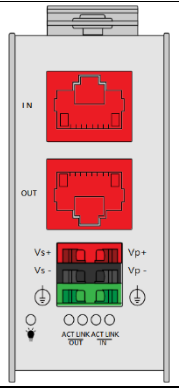
Date	Version	Description
2025/12/19	Version1.0	New release.

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).

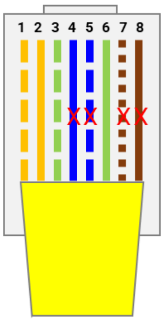
Notes QEC's PoE (Power over Ethernet)

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.

 IN OUT Vs+ Vs - Vp+ Vp - ACT LINK ACT LINK OUT IN	 IN OUT Vs+ Vs - Vp+ Vp - ACT LINK ACT LINK OUT IN
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:



T568B Color

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

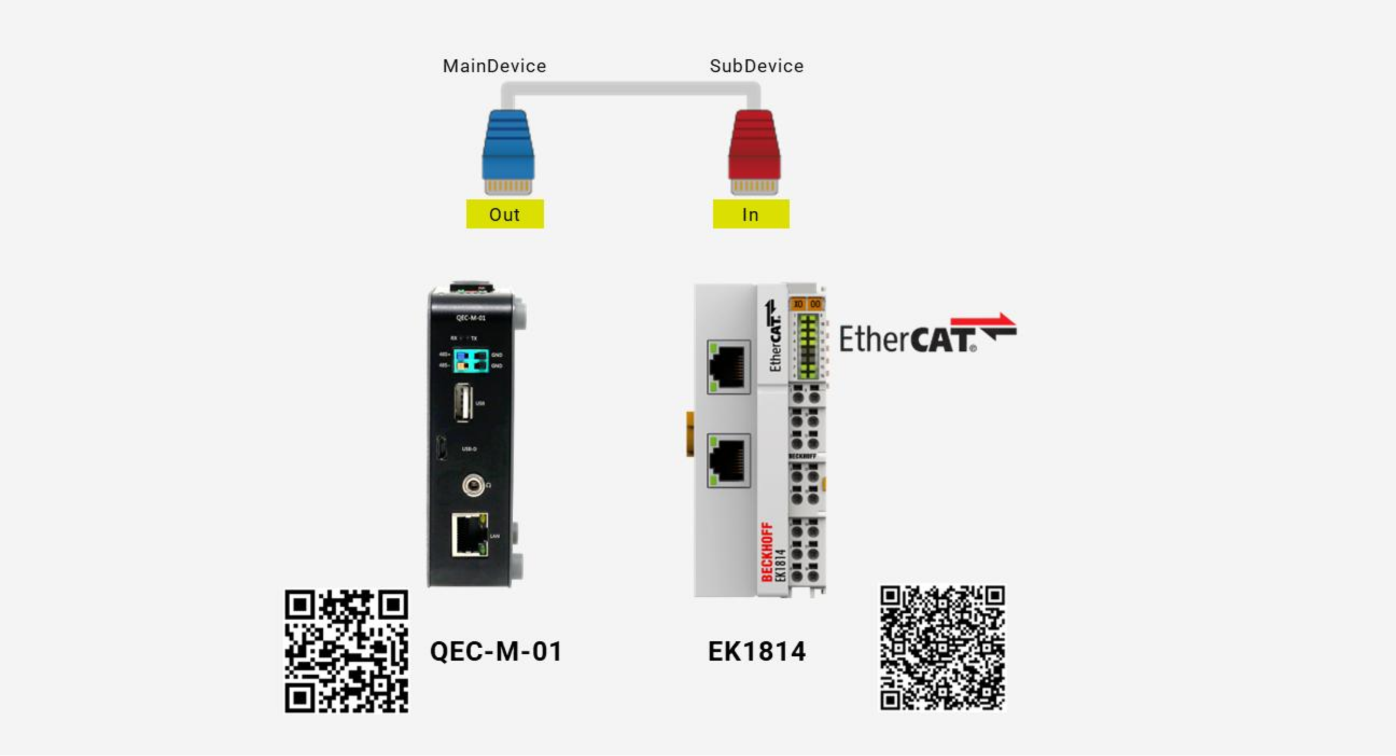
2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).

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

1. Connection and wiring hardware

The following devices are used here:

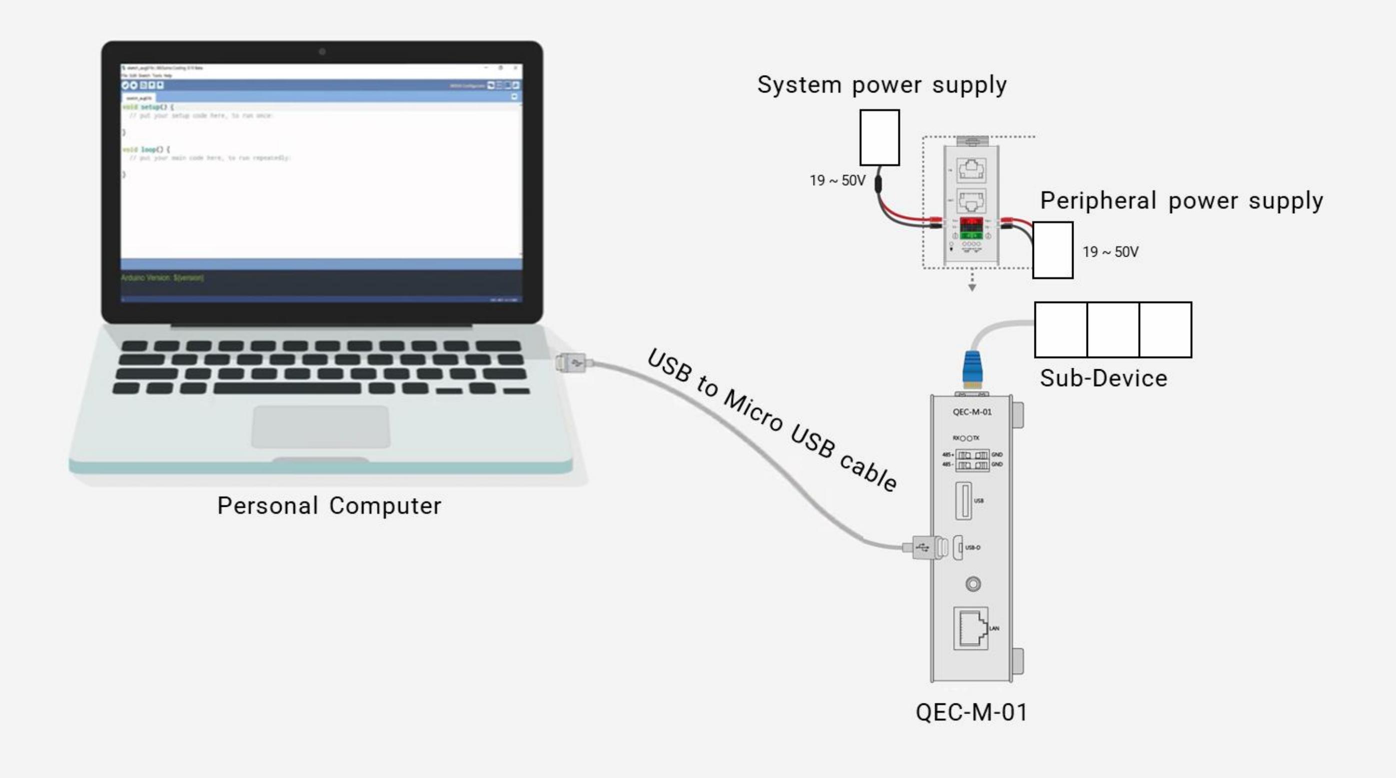
1. QEC-M-01 (EtherCAT MDevice)
2. EK-1814 series (EtherCAT Coupler with integrated digital inputs/outputs)
3. 24VDC power supply & EU-type terminal cable & LAN cable



1.1 QEC-M-01

QEC EtherCAT MDevice.

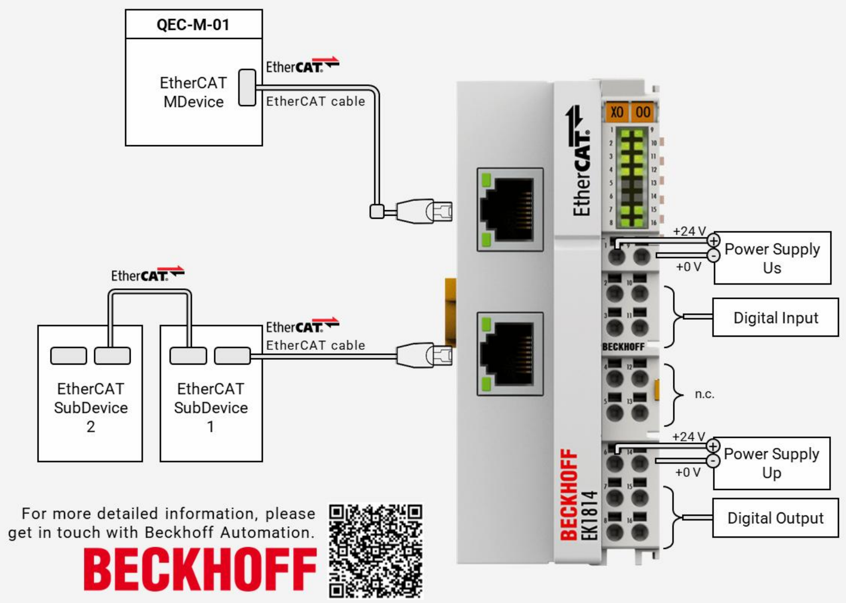
1. Power Supply: Connect to Vs+/Vs- and Vp+/Vp- power supplies via EU terminals for 24V power.
2. EtherCAT Connection: Using the EtherCAT Out port (On the top side) connected to the EtherCAT In port of EtherCAT SubDevice via RJ45 cable.



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.



For more detailed information, please get in touch with Beckhoff Automation.

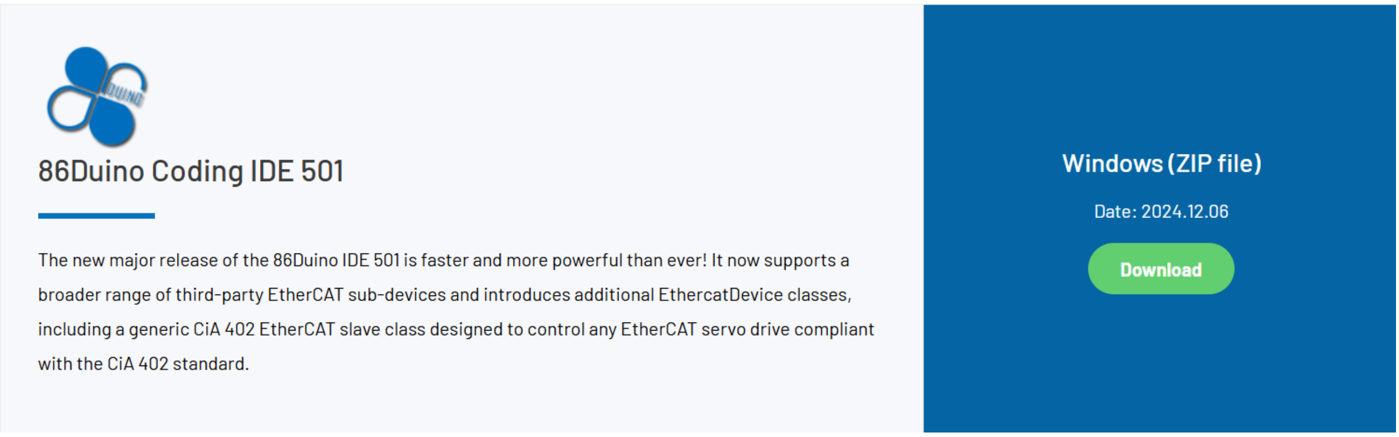
BECKHOFF

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/.



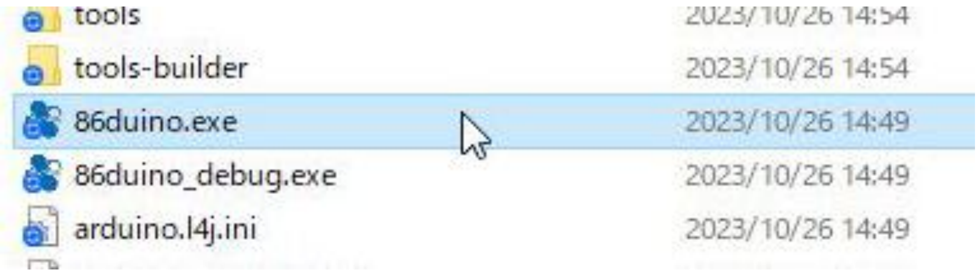
86Duino Coding IDE 501

The new major release of the 86Duino IDE 501 is faster and more powerful than ever! It now supports a broader range of third-party EtherCAT sub-devices and introduces additional EthercatDevice classes, including a generic CiA 402 EtherCAT slave class designed to control any EtherCAT servo drive compliant with the CiA 402 standard.

Windows (ZIP file)

Date: 2024.12.06

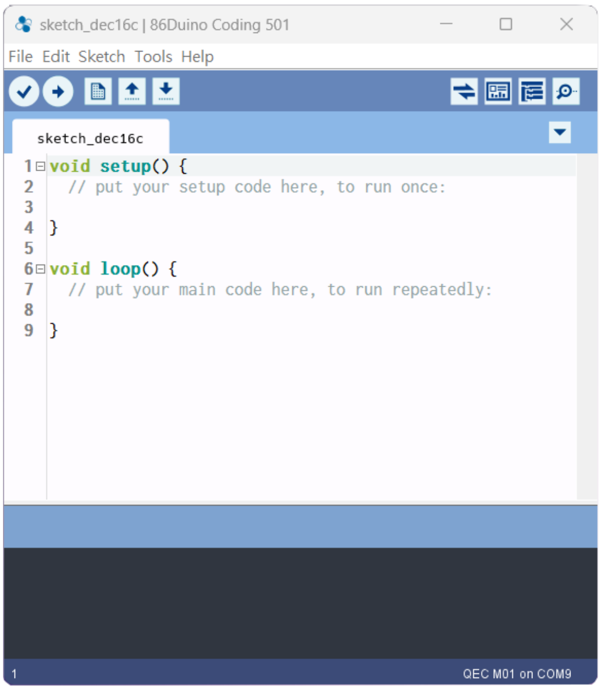
After downloading, please unzip the downloaded zip file, no additional software installation is required, just double-click 86duino.exe to start the IDE.



```
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 501+ looks like below.



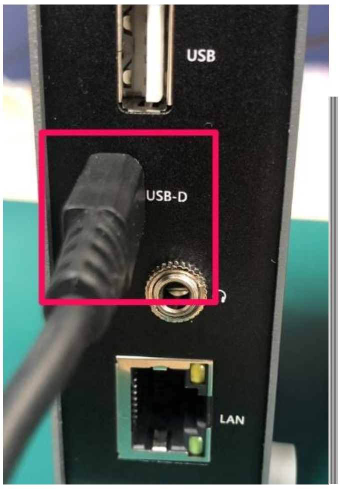
sketch_dec16c | 86Duino Coding 501

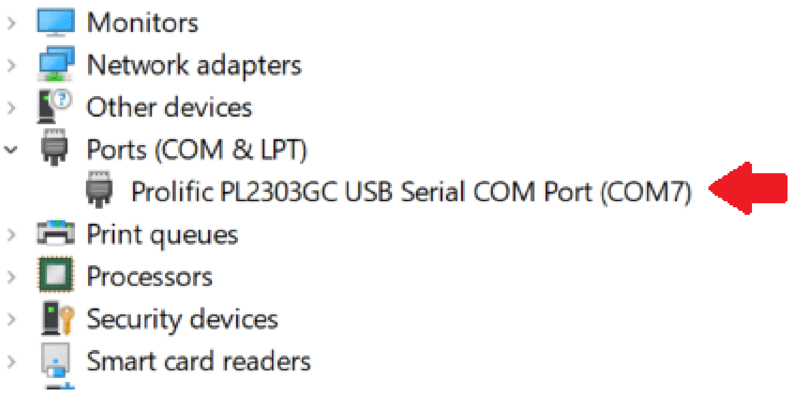
```
void setup() {
  // put your setup code here, to run once:

}

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

}
```

3. Connect to PC and set up the environment

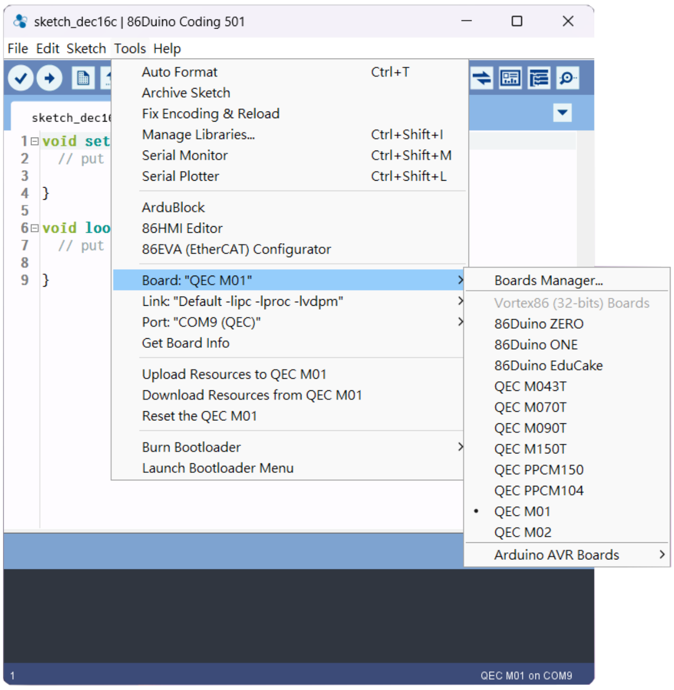
Follow the steps below to set up the environment:

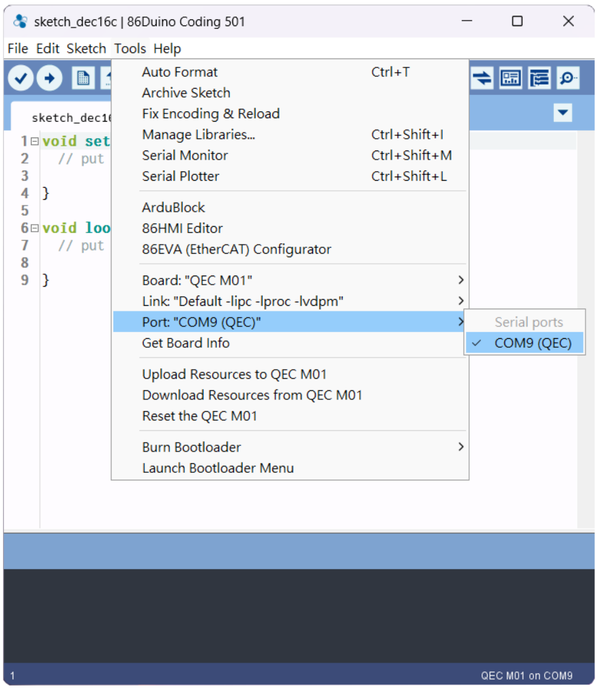
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “**Device Manager**” (select in the menu after pressing Win+X) ->” **Ports (COM & LPT)**” in your PC and expand the ports; you should see that the “**Prolific PL2303GC USB Serial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




sketch_dec16c | 86Duino Coding 501
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
ArduBlock
86HMI Editor
86EVA (EtherCAT) Configurator
Board: "QEC M01"
Link: "Default -lipc -lproc -lvdpm"
Port: "COM9 (QEC)"
Get Board Info
Upload Resources to QEC M01
Download Resources from QEC M01
Reset the QEC M01
Burn Bootloader
Launch Bootloader Menu
Serial ports
✓ COM9 (QEC)
QEC M01 on COM9

4. Write code

The EtherCAT MDevice (QEC-M-01) and the EK-1814 series (EtherCAT Coupler with integrated digital inputs/outputs) can be configured and programmed via the EtherCAT library in the 86Duino IDE.
The Arduino development environment has two main parts: `setup()` and `loop()`, which correspond to initialization and main programs. Before operating the EtherCAT network, you must configure it once. The process should be from Pre-OP to OP mode in EtherCAT devices.

The following program sets:

- EtherCAT Cycle Time: 3 milliseconds (Depend on EK1814 filter).
- EtherCAT Mode: ECAT_SYNC.

The EthercatMaster object ("master") represents the **QEC-M-01**, while the EthercatDevice_Generic object ("slave0") represents the **EK1814** module.

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.

A. In Setup Function:
In the `setup()` function, initialize Serial communication and bring up the EtherCAT network to OP mode. Follow the steps below:

1. Initialize Serial Communication
   - Start serial communication at a baud rate of 115200 for debug messages.
2. Start the EtherCAT MDevice (Master)
   - Call `master.begin()` to initialize the EtherCAT master and enter the EtherCAT state machine (typically to PRE-OP / initialization stage).
3. Attach the EK1814 SubDevice (Generic Device Class)
   - Use `slave0.attach(0, master)` to attach the first EtherCAT SubDevice (node index = 0) to the EtherCAT network using the EthercatDevice_Generic class.
4. Start EtherCAT and Switch to OPERATIONAL
   - Call `master.start(3000000, ECAT_SYNC)` to start cyclic communication and switch to OPERATIONAL state.
   - This sets the cycle time to 3 ms (3,000,000 ns) and enables ECAT_SYNC mode.

B. In Loop Function:
In the `loop()` function, toggle one PDO output bit periodically and read back one PDO input bit for verification.

1. Set DO Bit via PDO (ON)
   - Use `slave0.pdoBitWrite(0, HIGH)` to set PDO bit 0 (DO 0) to HIGH.
   - Wait 50 ms (`delay(50)`) used to wait for DI to reflect the DO change and for EtherCAT to refresh the input PDO data before reading.

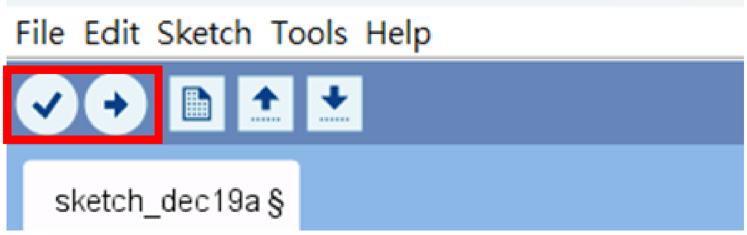
2. Read DI Bit via PDO and Print Result
    - Use `slave0.pdoBitRead(0)` to read PDO bit 0 (DI 0).
    - Print the value to "Serial: “EK1814 DI 0: <value>”.
3. Hold Time
    - Wait 3000 ms (`delay(3000)`) before changing the state.
4. Set DO Bit via PDO (OFF) and Repeat
    - Use `slave0.pdoBitWrite(0, LOW)` to set PDO bit 0 (DO 0) to LOW.
    - Wait 50 ms, read back `pdoBitRead(0)` again, print the result, and wait 3000 ms.
5. Code Logic Summary
    - `pdoBitWrite(n, HIGH/LOW)`: drives PDO bit n (commonly used for DO mapping).
    - `pdoBitRead(n)`: reads PDO bit n (commonly used for DI mapping).
    - The example toggles output every ~3 seconds and logs the corresponding input bit state.

* **Note** (EK1814 DO Type): EK1814 DO channels are Sourcing (PNP) outputs. When the DO is ON, it sources +24 V to the load, and the load return should be wired to 0 V (Up).

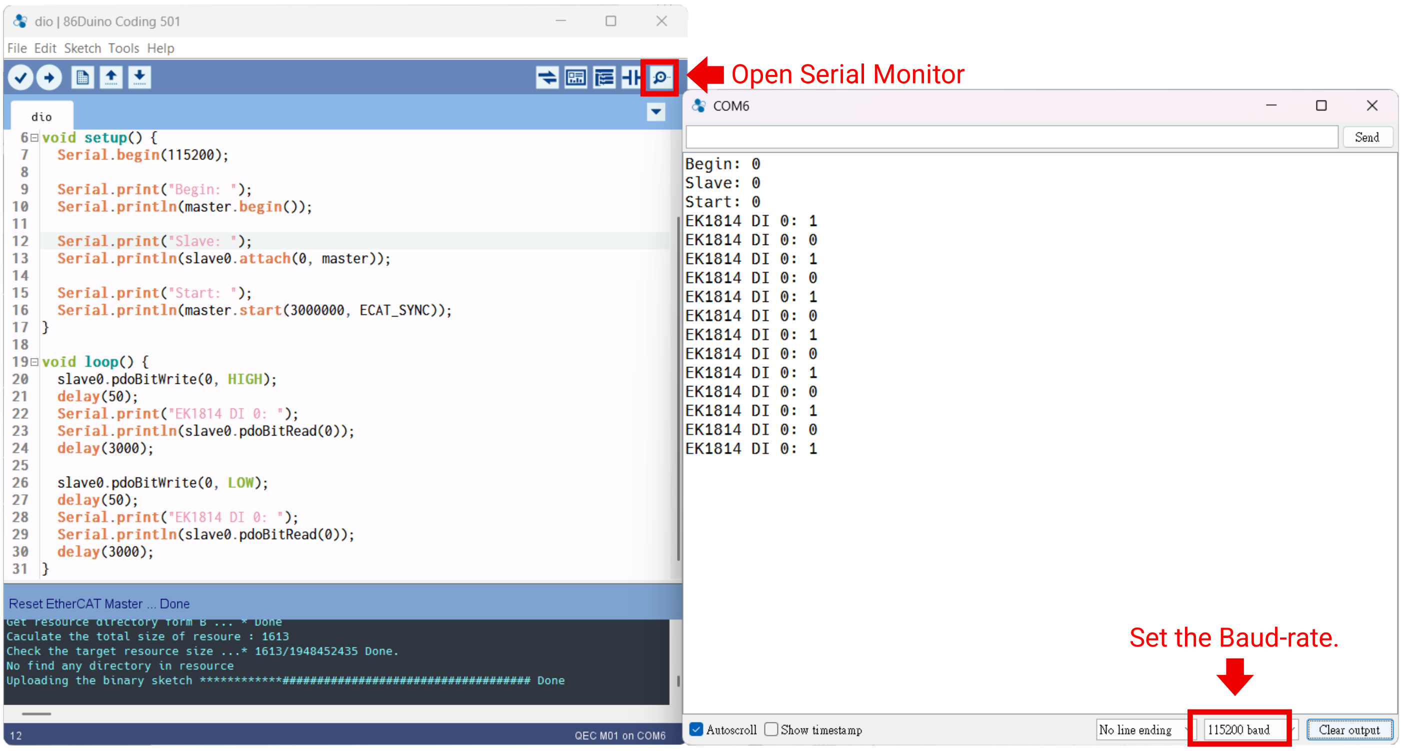
The example code is as follows:

```
#include "Ethercat.h"

EthercatMaster master;
EthercatDevice_Generic slave0;

void setup() {
  Serial.begin(115200);

  Serial.print("Begin: ");
  Serial.println(master.begin());

  Serial.print("Slave: ");
  Serial.println(slave0.attach(0, master));

  Serial.print("Start: ");
  Serial.println(master.start(3000000, ECAT_SYNC));
}

void loop() {
  slave0.pdoBitWrite(0, HIGH);
  delay(50);
  Serial.print("EK1814 DI 0: ");
  Serial.println(slave0.pdoBitRead(0));
  delay(3000);
  slave0.pdoBitWrite(0, LOW);
  delay(50);
  Serial.print("EK1814 DI 0: ");
  Serial.println(slave0.pdoBitRead(0));
  delay(3000);
}
```

***Note:** Once the code is written, click on the toolbar to ✓ compile, and to confirm that the compilation is complete and error-free, you can click → to upload.



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



```
void setup() {
  Serial.begin(115200);

  Serial.print("Begin: ");
  Serial.println(master.begin());

  Serial.print("Slave: ");
  Serial.println(slave0.attach(0, master));

  Serial.print("Start: ");
  Serial.println(master.start(3000000, ECAT_SYNC));
}

void loop() {
  slave0.pdoBitWrite(0, HIGH);
  delay(50);
  Serial.print("EK1814 DI 0: ");
  Serial.println(slave0.pdoBitRead(0));
  delay(3000);

  slave0.pdoBitWrite(0, LOW);
  delay(50);
  Serial.print("EK1814 DI 0: ");
  Serial.println(slave0.pdoBitRead(0));
  delay(3000);
}
```

Reset EtherCAT Master ... Done

```
Get resource directory form B ... * Done
Caculate the total size of resoure : 1613
Check the target resource size ...* 1613/1948452435 Done.
No find any directory in resource
Uploading the binary sketch ************#################################### Done
```

12 QEC M01 on COM6

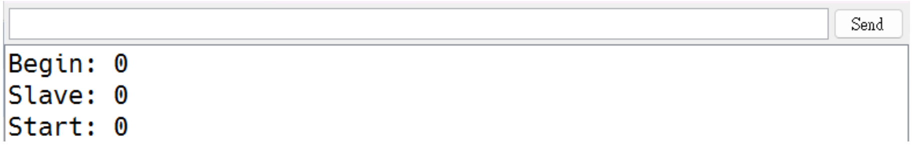
```
Begin: 0
Slave: 0
Start: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
```

115200 baud

Open Serial Monitor

Set the Baud-rate.

If the EtherCAT communication configuration is successful, the Serial Monitor will print “0” for each status for EtherCAT.



```
Begin: 0
Slave: 0
Start: 0
```

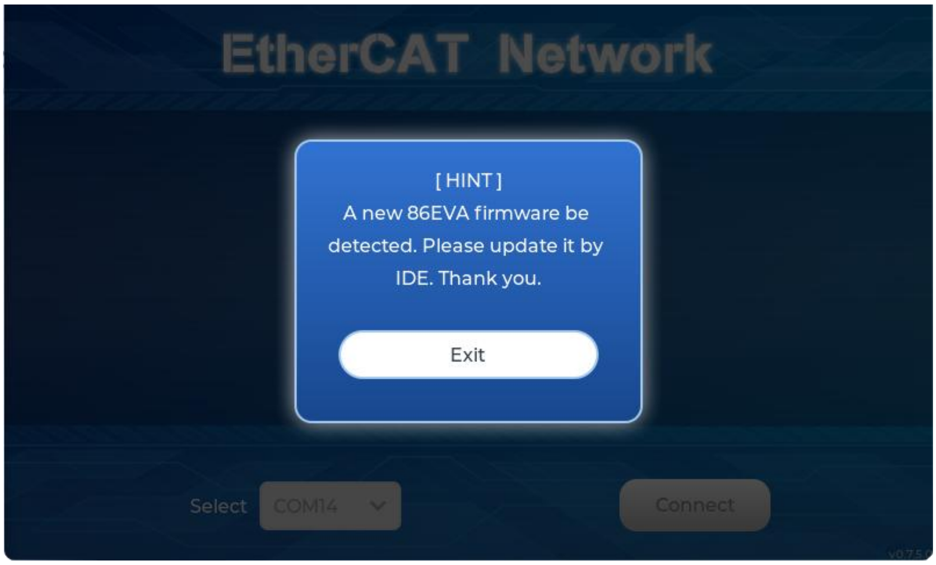
And it will print the EK1814 DI 0 value to the serial monitor.

```
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
EK1814 DI 0: 1
EK1814 DI 0: 0
```

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.



[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:

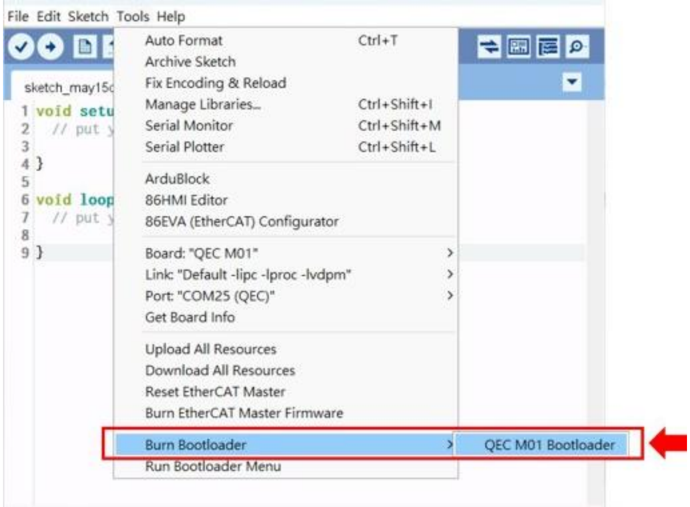
Step 1: Setting up QEC-M

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.

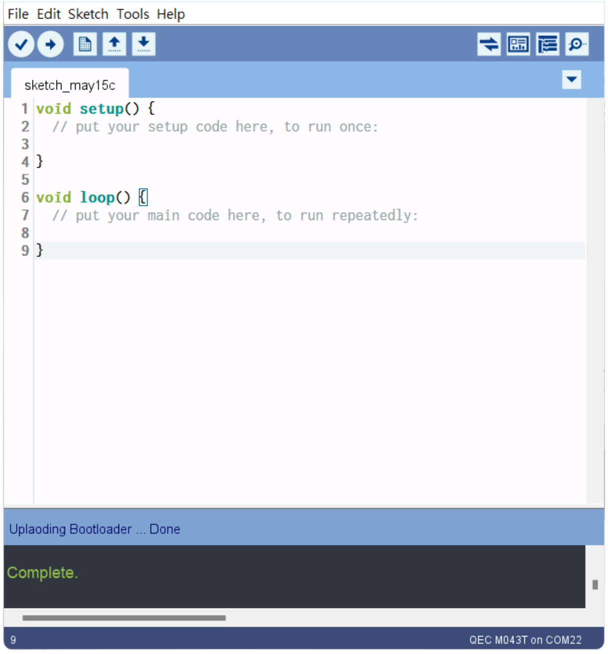
Step 2: Click “Burn Bootloader” button

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:



Step 3: Complete the Update



```
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.

All Trademarks appearing in this manuscript are registered trademark of their respective owners. All Specifications are subject to change without notice.

©ICOP Technology Inc. 2025