Start Guide

EK1100 + EL1889 + EL2889 : EtherCAT Coupler, DI, and DO

86Duino Coding IDE 501+

EtherCAT Library

(Version 1.1)

Revision

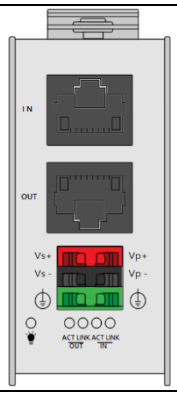
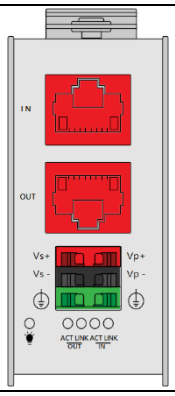
Date	Version	Description
2026/1/13	Version1.0	New release.
2026/11/3	Version1.1	Corrected ground-switching I/O wiring; added loopback test and clarified timing.

Preface

In this guide, we will show you how to use the EtherCAT MDevice **QEC-M-01**, the **EK1100** (EtherCAT Coupler), the **EL1889** (EtherCAT Digital Input terminal), and the **EL2889** (EtherCAT Digital Output terminal).

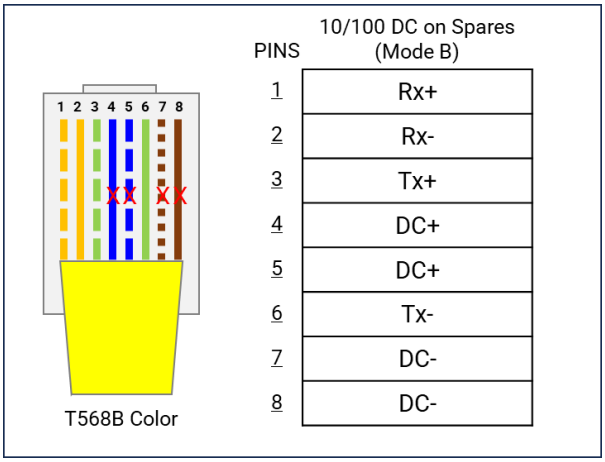
Notes on QEC's PoE (Power over Ethernet)

In QEC product installations, users can easily distinguish between PoE and non-PoE: if the RJ45 housing is red, it is PoE type, and if the RJ45 housing 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:



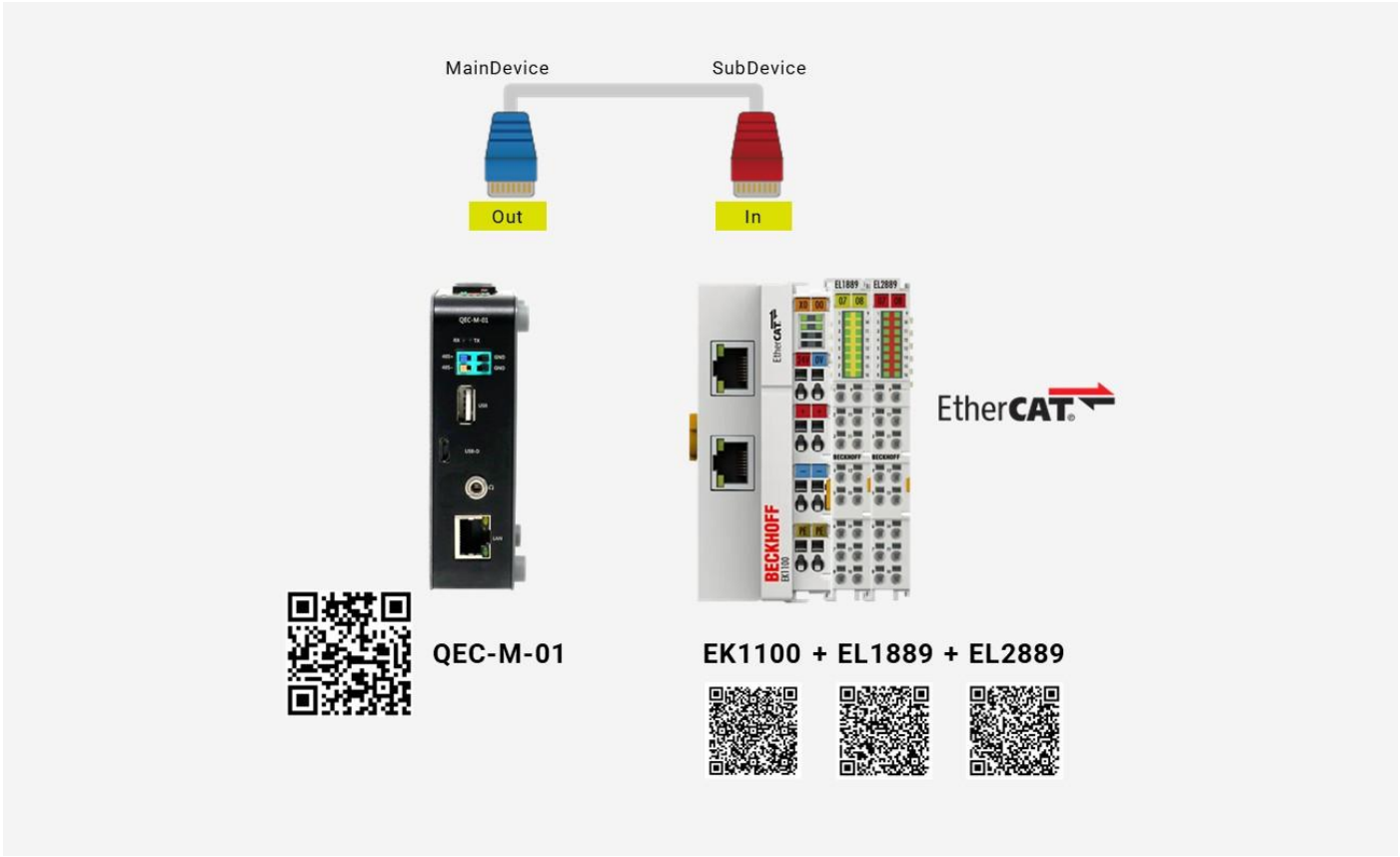
2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).

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

1. Connection and wiring hardware

The following devices are used here:

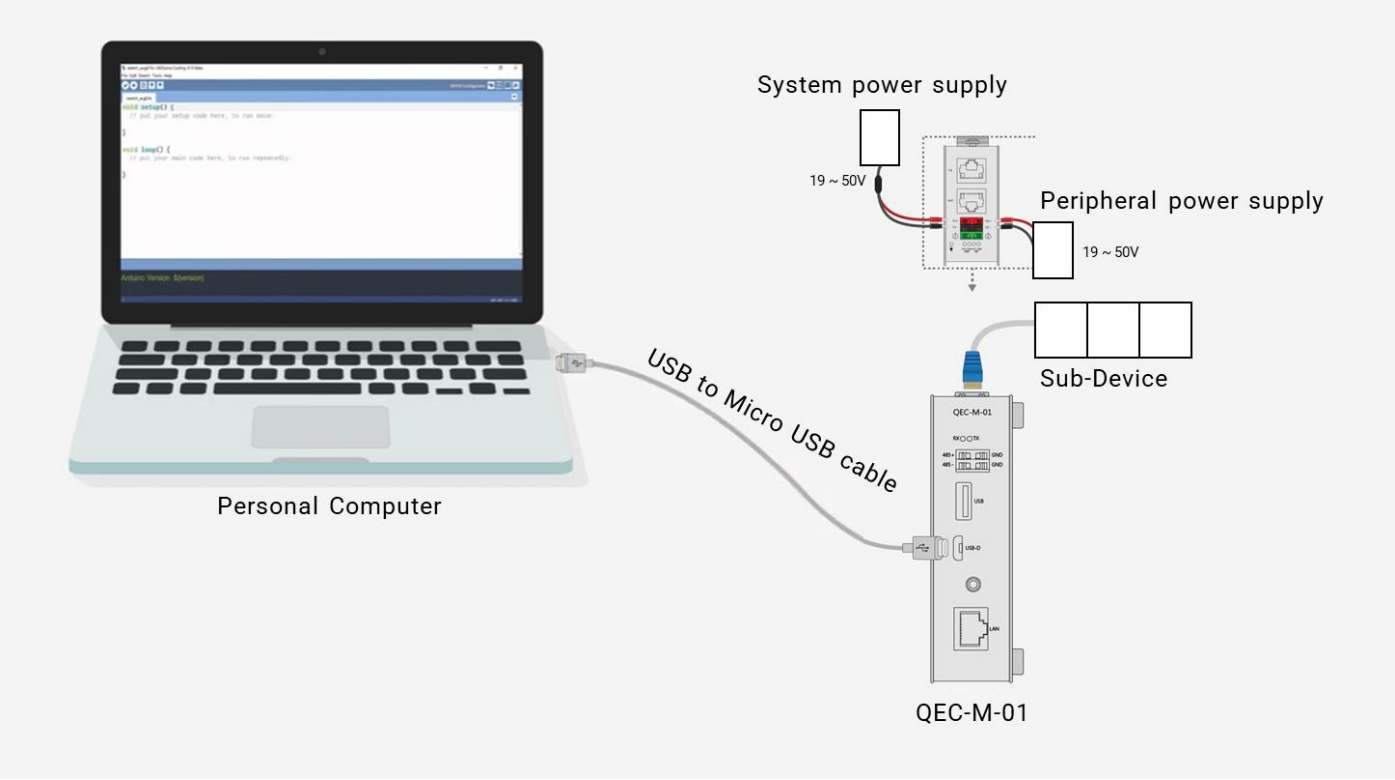
1. QEC-M-01 (EtherCAT MDevice)
2. EK1100 (EtherCAT Coupler)
3. EL1889 (EtherCAT Terminal, 16-channel digital input, 24 V DC, 3 ms, ground switching)
4. EL2889 (EtherCAT Terminal, 16-channel digital output, 24 V DC, 0.5 A, ground switching)
5. EL9011 or EL9012 end cap (after the last terminal)
6. 24VDC power supply & EU-type terminal cable & LAN cable



1.1 QEC-M-01

QEC EtherCAT MDevice.

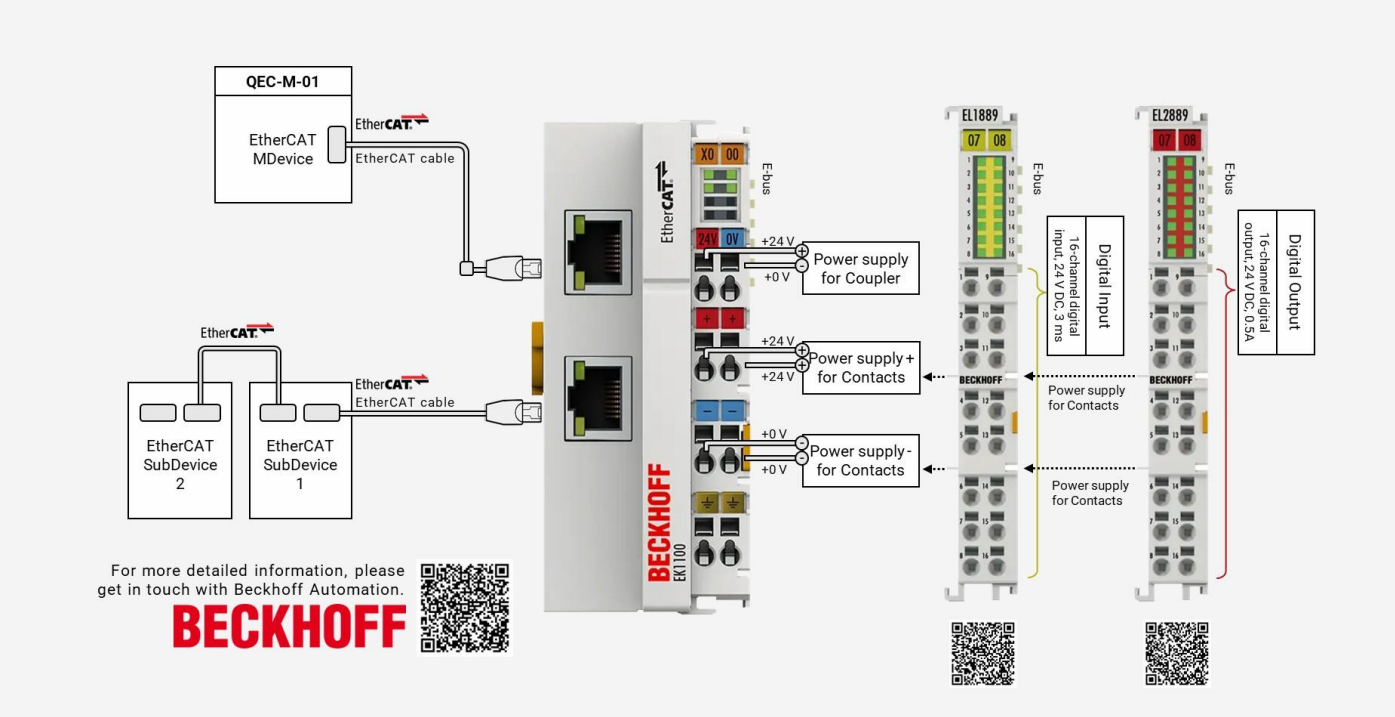
1. Power Supply: Connect to Vs+/Vs- and Vp+/Vp- power supplies via EU terminals for 24V power.
2. EtherCAT Connection: Using the EtherCAT Out port (On the top side) connected to the EtherCAT In port of EtherCAT SubDevice via RJ45 cable.



1.2 EK1100 + EL1889 + EL2889

The **Beckhoff EK1100** is an EtherCAT Coupler used to connect modular EtherCAT Terminals. In this setup, the **EL1889** provides digital inputs (DI) and the **EL2889** provides digital outputs (DO). They can be used as EtherCAT SubDevices in a QEC EtherCAT network to implement basic discrete input sensing and output driving.

The diagram shows a typical wiring example with a **QEC MDevice (e.g., QEC-M-01**) and the Beckhoff EtherCAT coupler and terminal modules.

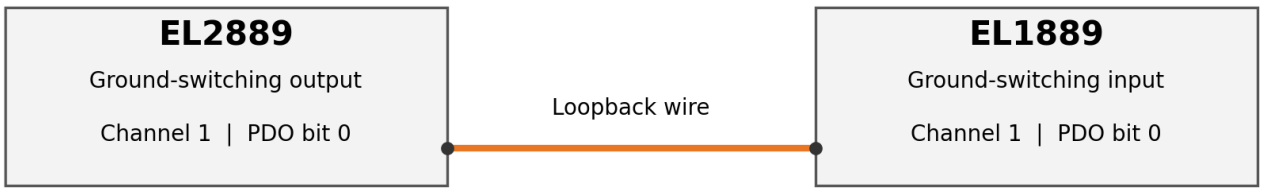


Connections are grouped by function:

- EtherCAT
  - IN (EK1100): Connect to the QEC MDevice EtherCAT port (or the previous SubDevice).
  - OUT (EK1100): Connect to the next SubDevice (if any).
- Power & Grounding
  - EK1100: Supply 24 V / 0 V to the coupler terminals and separately feed +24 V / 0 V to the field power contacts (see Section 1.3).
- Digital Inputs (EL1889, DI)
  - EL1889 is ground switching: 0-7 V = ON (1); 18-30 V = OFF (0). Its inputs use the +24 V power contact as their reference.
- Digital Outputs (EL2889, DO)
  - EL2889 outputs switch toward the 0 V power contact.
  - Load wiring: Connect the load between +24 V and the output. When ON, the output sinks current toward 0 V (ground switching).
- Indicators / LEDs
  - EK1100: Status LEDs indicate module power and EtherCAT link/activity.
  - EL1889 / EL2889: Channel LEDs indicate each DI/DO channel state.

1.3 Loopback wiring and expected results

Use this wiring for the DO0-to-DI0 test in Chapter 4. Software channel indices start at 0; the first physical channel on each terminal is marked 1.



EK1100 field power contacts: +24 V and 0 V supplied to both terminals

Connect with power OFF. Fit an end cap after the last terminal.

Wiring procedure

1. Switch off the power. Assemble EK1100, EL1889 and EL2889 in this order, and fit an EL9011 or EL9012 end cap after EL2889.
2. Supply the EK1100 upper 24 V / 0 V terminals for the coupler and E-bus. Also supply the field power contacts (+24 V / 0 V). These are separate circuits; a suitably rated 24 V supply may feed both.
3. Connect EL2889 output channel 1 (DO0) directly to EL1889 input channel 1 (DI0). Use the same field 0 V reference for both terminals. Do not connect +24 V directly to the EL2889 output.
4. Check the wiring, then power on and run the example. The EL1889 input uses the +24 V power contact as its reference; the EL2889 output switches toward the 0 V power contact.

Channel mapping

Physical channel	Program / PDO index	86EVA virtual pin
EL2889 output 1	DO0 / output bit 0	D0 (ArduBlock guide)
EL1889 input 1	DI0 / input bit 0	D1 (ArduBlock guide)

Expected result

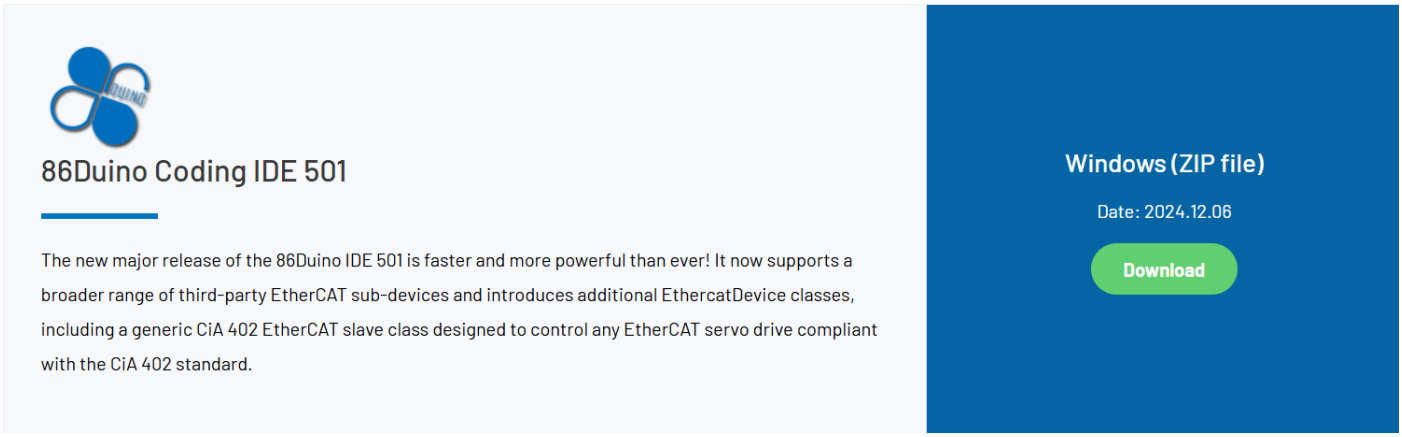
With HIGH (PDO value 1), EL2889 DO0 is enabled and sinks toward 0 V; after the 50 ms wait, EL1889 DI0 should read 1. With LOW (PDO value 0), DO0 is off and DI0 should read 0. HIGH describes the logical output command, not a +24 V terminal voltage.

EL1889 thresholds: 0-7 V = input 1; 18-30 V = input 0. Avoid the undefined range between these thresholds. For a load test instead of loopback, connect the load between +24 V and the EL2889 output.

Reference: Beckhoff product data and Information System (EL1889, EL2889 and EK1100).

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.

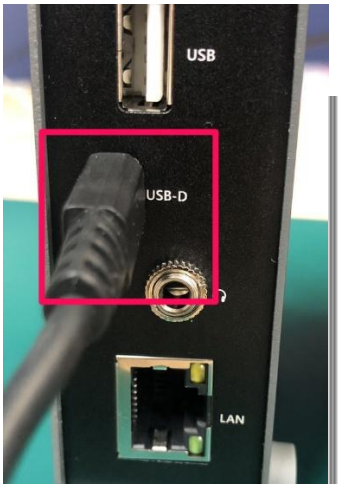
86Duino Coding IDE 501+ looks like below.

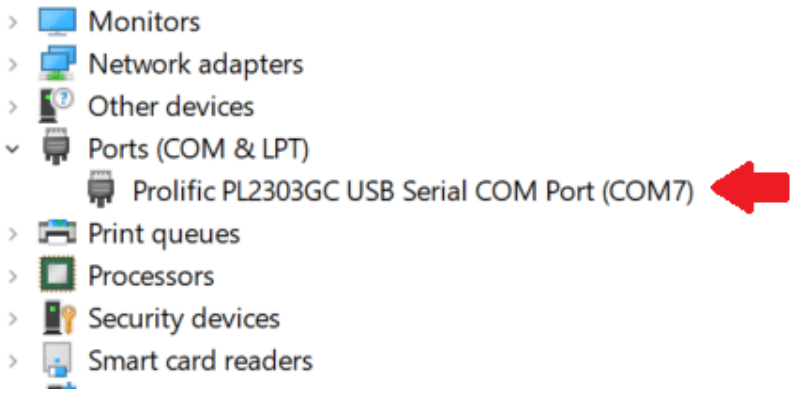


3. Connect to PC and set up the environment

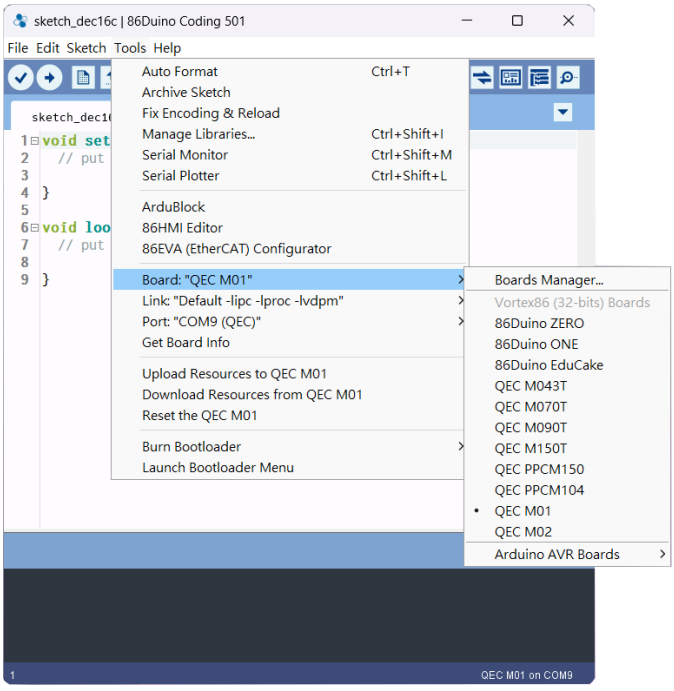
Follow the steps below to set up the environment:

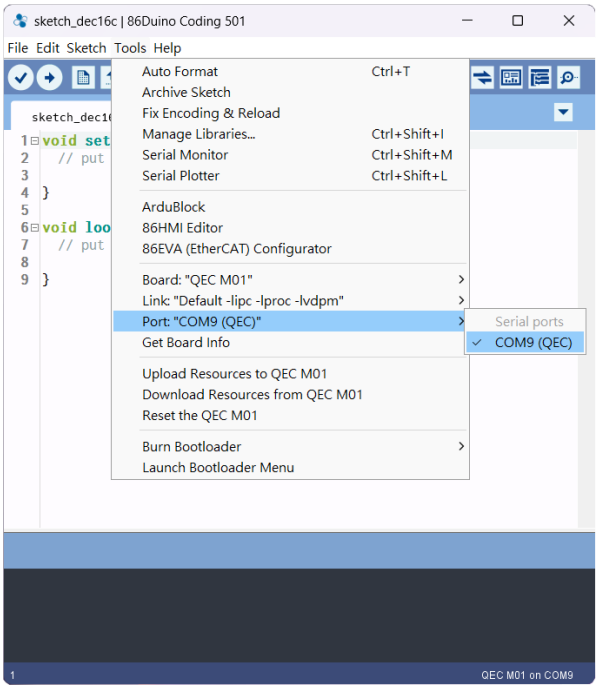
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open “**Device Manager**” (select in the menu after pressing Win+X) ->” **Ports (COM & LPT)**” in your PC and expand the ports; you should see that the “**Prolific PL2303GC USB Serial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

The EtherCAT MDevice (**QEC-M-01**) and the Beckhoff coupler and terminal modules (**EK1100** + **EL1889** + **EL2889**) 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 for this example. The EL1889 input filter is 3 ms; filter time and EtherCAT cycle time are separate settings.
- EtherCAT Mode: ECAT_SYNC.

The `EthercatMaster` object (“master”) represents QEC MDevice: **QEC-M-01**.
The `EthercatDevice_Generic` objects represent the Beckhoff terminals: **EK1100** (“ek1100”), **EL1889** (“el1889”), and **EL2889** (“el2889”).

This example toggles EL2889 DO0, waits 50 ms for input filtering and PDO updates, then prints EL1889 DI0. Connect the loopback wire as shown in Section 1.3; the expected input values are 1 when the output is ON and 0 when it is OFF.

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 (QEC-M-01)
    - Call `master.begin()` to initialize the EtherCAT MDevice and enter the EtherCAT state machine (typically to PRE-OP / initialization stage).
3. Attach the SubDevice (`EthercatDevice_Generic` Class)
    - Use `attach(node, master)` to attach the first EtherCAT SubDevice:
      `ek1100.attach(0, master)` → SubDevice 0: EK1100
      `el1889.attach(1, master)` → SubDevice 1: EL1889 (DI)
      `el2889.attach(2, master)` → SubDevice 2: EL2889 (DO)
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 `el2889.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 `el1889.pdoBitRead(0)` to read PDO bit 0 (DI 0).
   - Print the value to Serial: “EL1889 DI 0: <value>”.
3. Hold Time
   - Wait 1000 ms (`delay(1000)`) before changing the state.
4. Set DO Bit via PDO (OFF) and Repeat
   - Use `el2889.pdoBitWrite(0, LOW)` to set PDO bit 0 (DO 0) to LOW.
   - Wait 50 ms, read back `el1889.pdoBitRead(0)` again, and print the result to Serial.
   - Wait 1000 ms, and the program repeats this cycle continuously.
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 holds each output state for about 1.05 seconds and logs the corresponding input state.

***Note:** EL2889 is ground switching (sinking). HIGH enables the output toward 0 V; it does not supply +24 V. Connect a load between +24 V and the output.

Check the return values printed during setup: 0 means success. If `begin()`, `attach()` or `start()` returns a nonzero value, stop the test and resolve that error before using PDO reads or writes.

The example code is as follows:

```
#include "Ethercat.h"

EthercatMaster master;
EthercatDevice_Generic ek1100;
EthercatDevice_Generic el1889;
EthercatDevice_Generic el2889;

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

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

  Serial.print("EK1100: "); Serial.println(ek1100.attach(0, master));
```

```
  Serial.print("EL1889: "); Serial.println(el1889.attach(1, master));
  Serial.print("EL2889: "); Serial.println(el2889.attach(2, master));

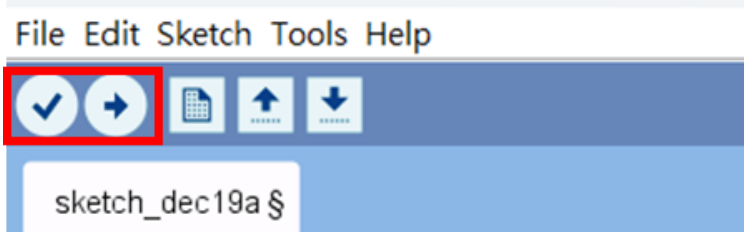
  Serial.print("Start: ");
  Serial.println(master.start(3000000, ECAT_SYNC));
}

void loop() {
  el2889.pdoBitWrite(0, HIGH);
  delay(50);
  Serial.print("EL1889 DI 0: ");
  Serial.println(el1889.pdoBitRead(0));
  delay(1000);

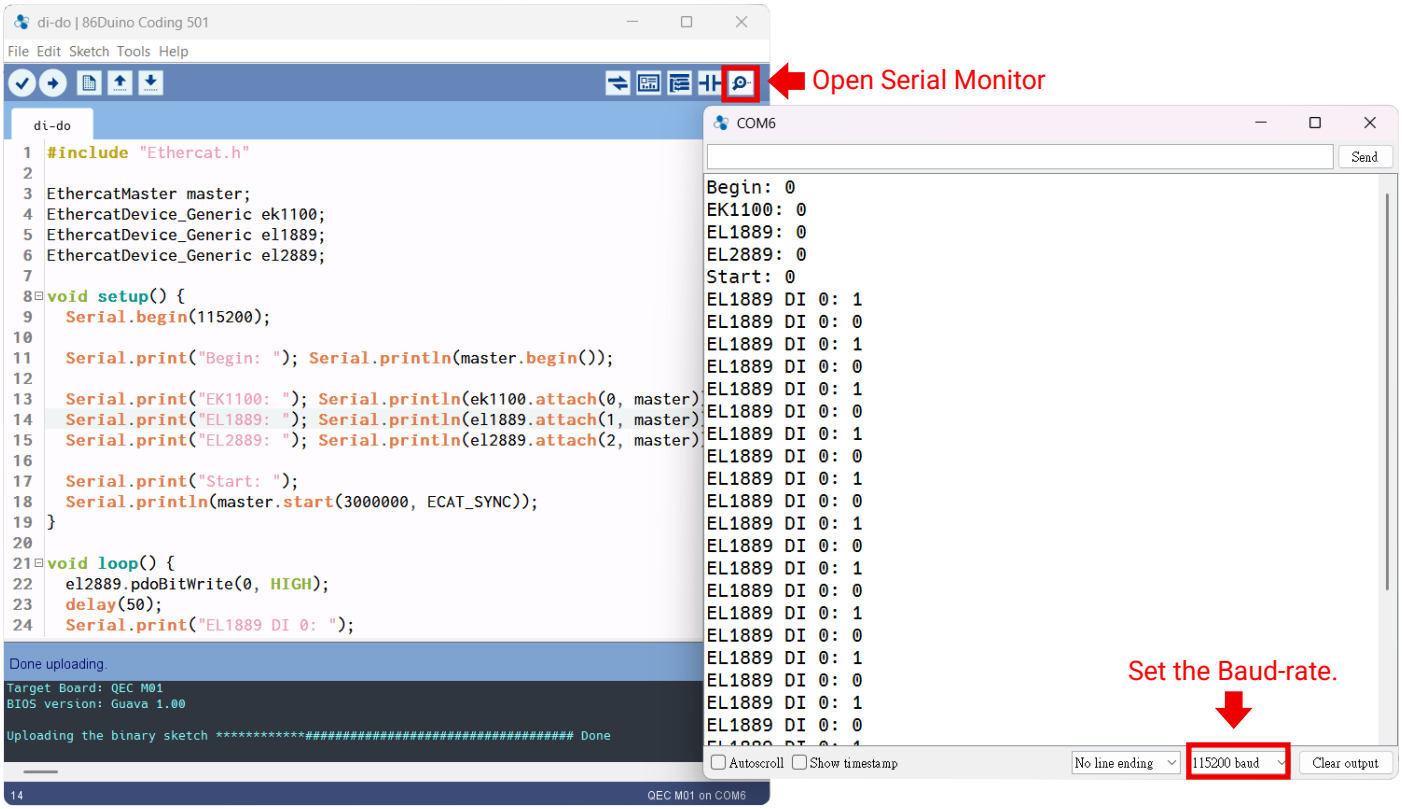
  el2889.pdoBitWrite(0, LOW);
  delay(50);
  Serial.print("EL1889 DI 0: ");
  Serial.println(el1889.pdoBitRead(0));
  delay(1000);

}
```

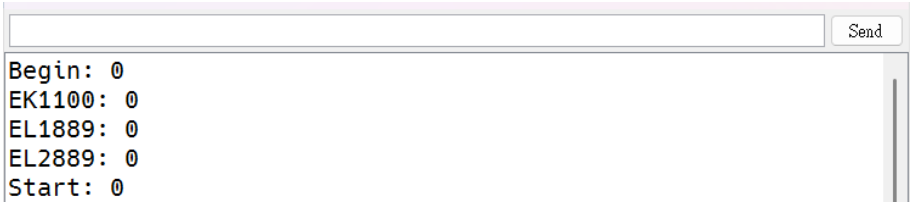
*__Note:__ Once the code is written, click on the toolbar to compile, and to confirm that the compilation is complete and error-free, you can click to upload.



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.



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



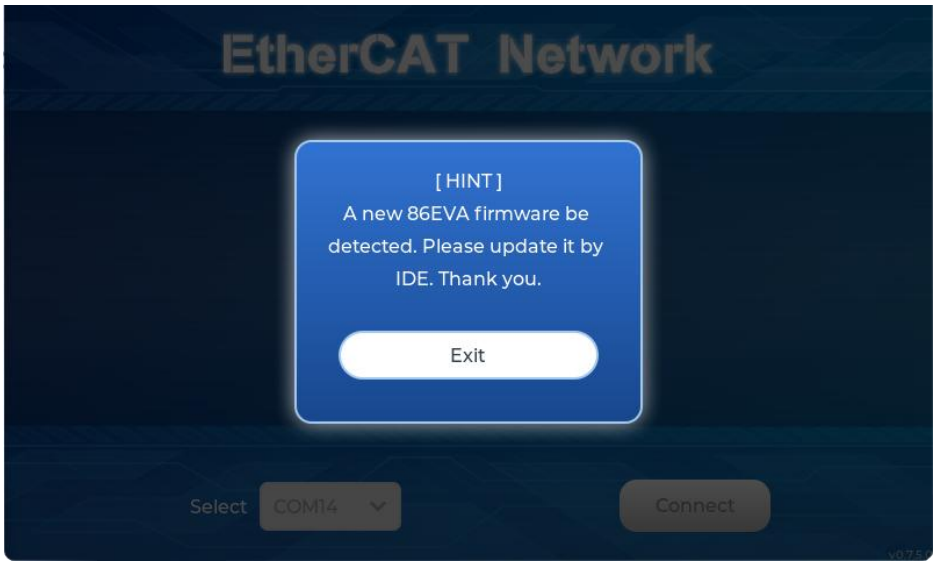
And it will print the EL1889 DI 0 value to the serial monitor.



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:

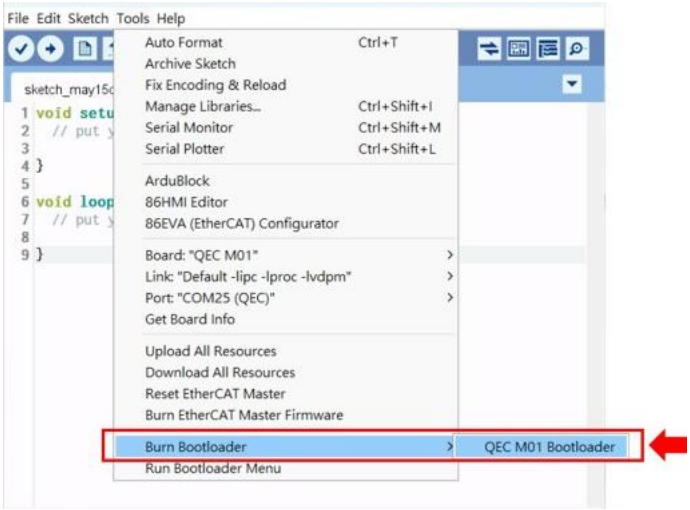
Step 1: Setting up QEC-M

1. Download and install 86Duino IDE 501+ (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.

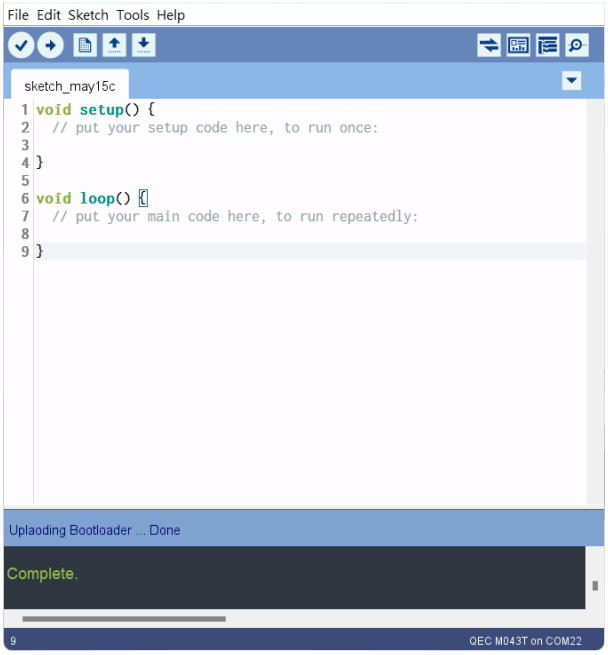
Step 2: Click “Burn Bootloader” button

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.

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

©ICOP Technology Inc. 2026