

# スタートガイド

NPM PULSERVO II EtherCAT ドライバ

CiA402 PP モード (1 軸)

86EVA (GUI コンフィグレーション)編



86Duino Coding IDE 501

EtherCAT Library

(Version 1.1)

# 改訂履歴

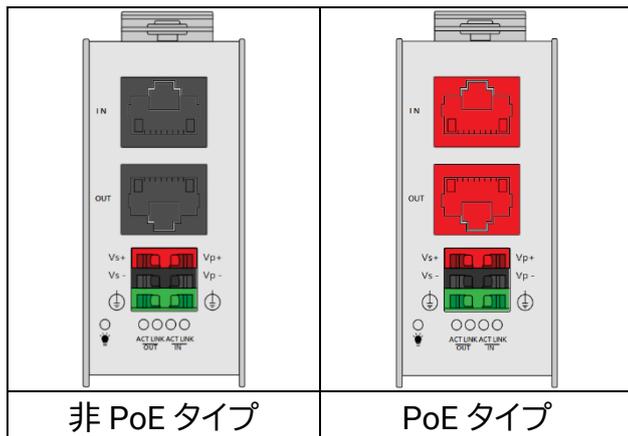
記述日	バージョン	備考
2024/12/31	Version1.0	New Release.
2025/1/23	Version1.1	Main デバイスを M デバイスへ、Sub デバイス及びスレーブを Sub デバイスへ

# 序文

本ガイドでは、QEC-M-01 (EtherCAT M デバイス) と日本パルスモーター株式会社製 PULSERVO II(クローズドループステッピングモータドライバ)の使用方を説明します。EC-AD1441A4 (2相バイポーラ定電流、4軸ステッピングモータドライバ)の使用方を説明します。

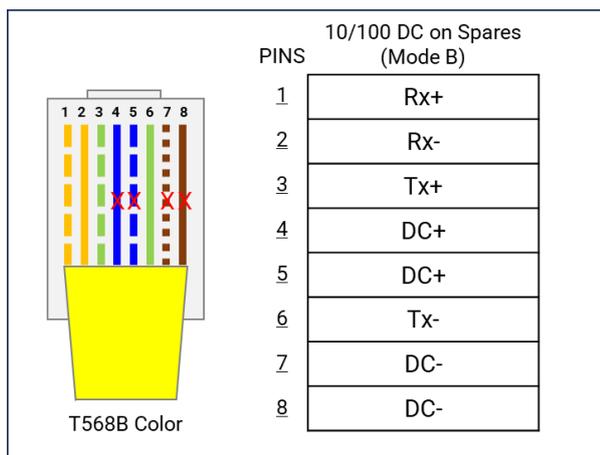
## 注意 QEC 機器の PoE (Power over Ethernet)について

QEC 製品のインストレーションでは、ユーザーは PoE と非 PoE を簡単に区別できます：RJ45 ハウジングが赤色の場合は PoE タイプ、RJ45 ハウジングが黒色は非 PoE タイプです。



PoE(Power over Ethernet)は、ネットワーク経由で電力を供給する機能です。QEC には配線を減らすためオプションとして PoE 機能を用意しています。実際には PoE はシステム機器に基づいて選択されるため次の点に注意してください。

1. QEC の機能は EtherCAT P とは異なり互換性がありません。QEC の PoE 機能は PoE タイプ B に準拠しており、下記のようなピン配列になっています：



2. PoE デバイスと非 PoE デバイスを接続するときは、必ず EtherNet ケーブルのピン 4、5、7、8 を切断してください (例えば PoE 対応の QEC EtherCAT M デバイスを他社の EtherCAT サブデバイスに接続する場合)。
3. QEC PoE 電源は最大 24V/3A です。

# 1. ハードウェアの接続と配線

ここでは次のデバイスを使用します：

1. QEC-M-01(EtherCAT M デバイス)
2. NPM PULSERVO II (クローズドループステッピングモータドライブ)
3. 24V 電源, EU タイプ端子接続ケーブル, LAN ケーブル
4. PSM2-28 (フレーム 28mm タイプ標準モータ)



## 1.1 QEC-M-01

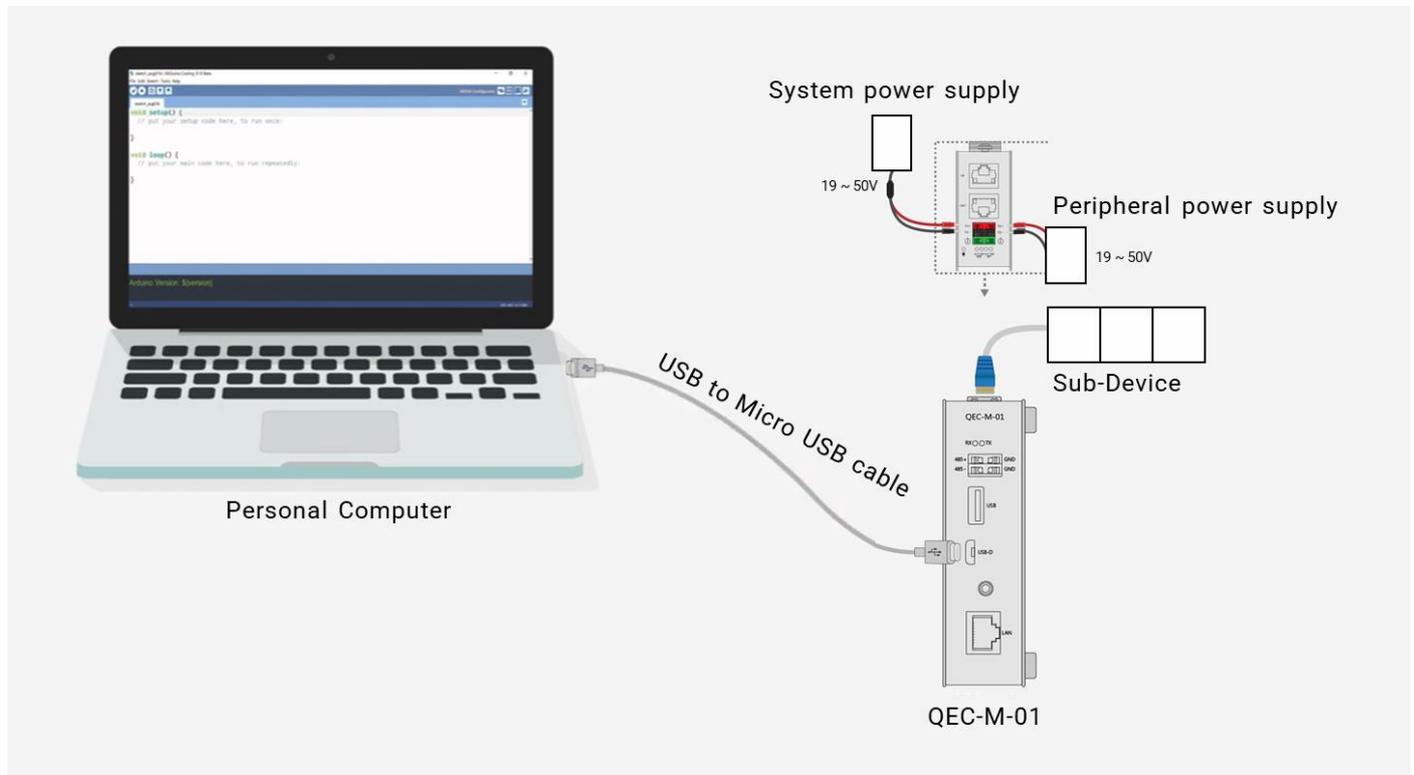
QEC EtherCAT M デバイス

1. 電源:

24V 電源を QEC EtherCAT M デバイスの EU タイプ端子 Vs+/Vs- および Vp+/Vp- に接続

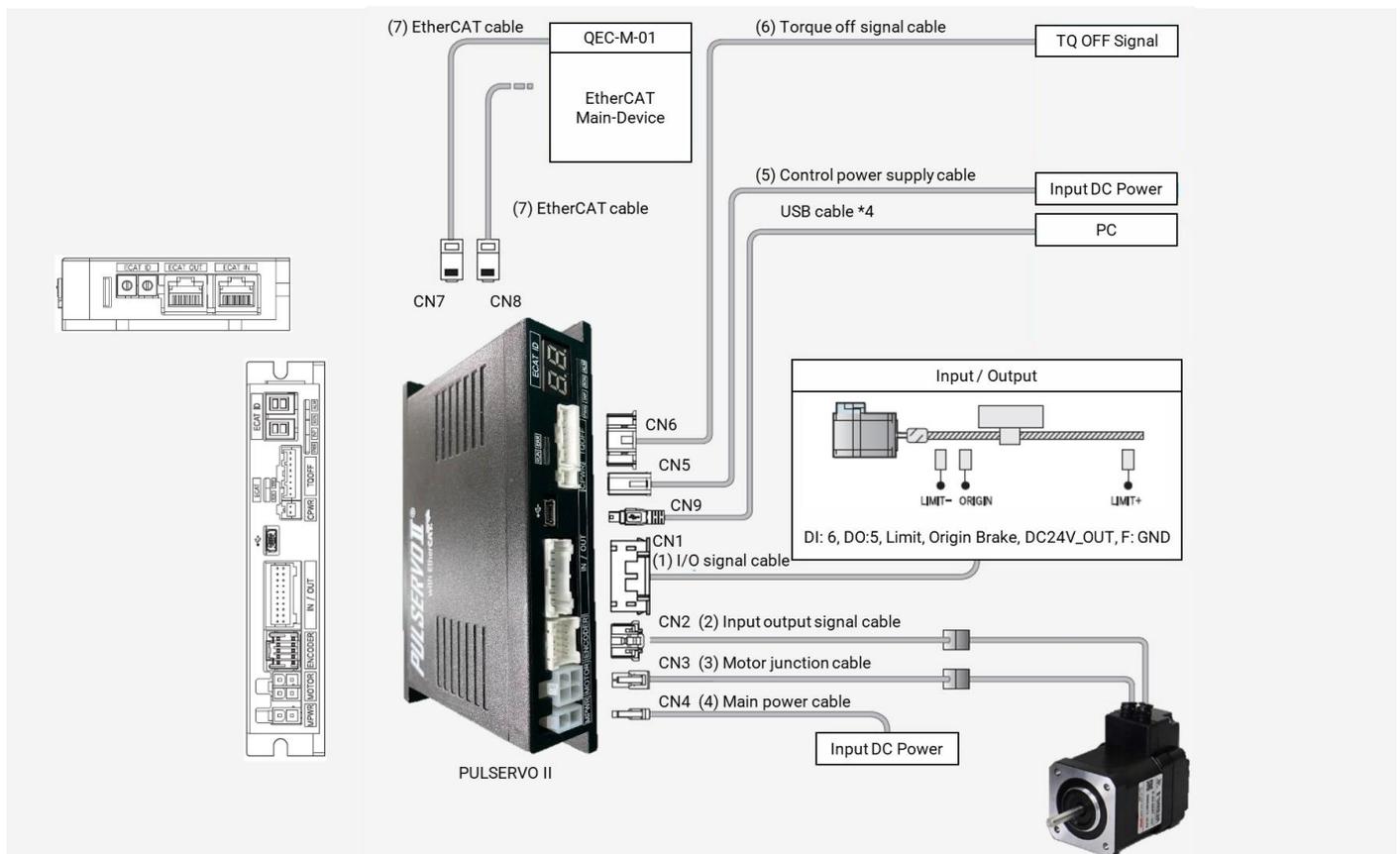
2. EtherCAT 接続:

EtherCAT 出力ポート (上側) からサブデバイスの EtherCAT 入力ポートに RJ45 ケーブルで接続



## 1.2 NPM PULSERVO II

NPM PULSERVO II シリーズの PLS2-EC セットは EtherCAT 仕様のクローズループステッピングモータです。以下の図は PSM2-28 モータを接続した時の一例です。



### 1. EtherCAT 接続

- CN7, CN8: EtherCAT 通信用コネクタ
- SW1, SW2: EtherCAT ID 設定用スイッチ

### 2. 信号接続

- CN1: デジタル入出力用コネクタ
- CN2: モータ位置追跡用エンコーダ接続
- CN3: モータ用電源と制御

### 3. 電源供給

- CN4: モータ制御用主電源入力
- CN5: ドライバ制御回路用電源入力

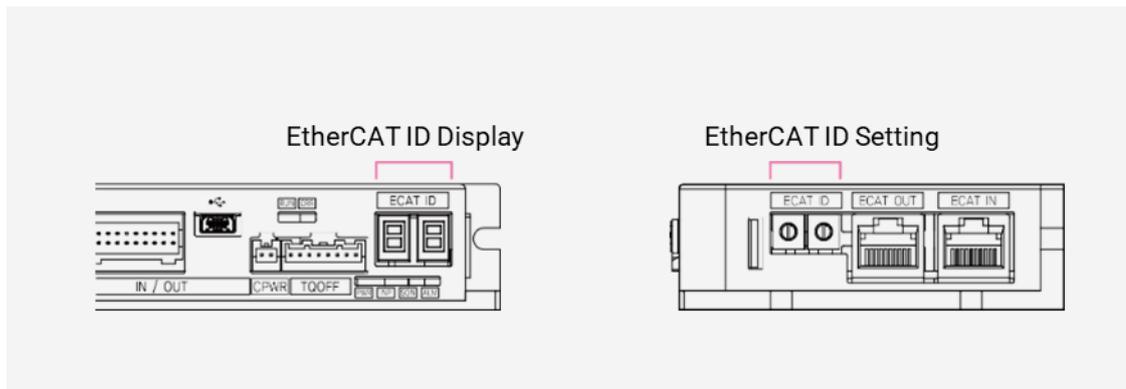
### 4. 安全性とステータス

- CN6: 緊急停止機能用トルク Off 信号入力
- LED によりドライバ、EtherCAT 通信状況及び EtherCAT ID を表示します。

### 5. USB ポート:

- CN9: 設定及びモニタ用 USB

## PULSERV II ドライバの ID 設定



ロータリースイッチにて EtherCAT ID を設定します。左側のスイッチが 10 桁目で右側のスイッチが 1 桁目を表します。0~99 まで設定可能です。

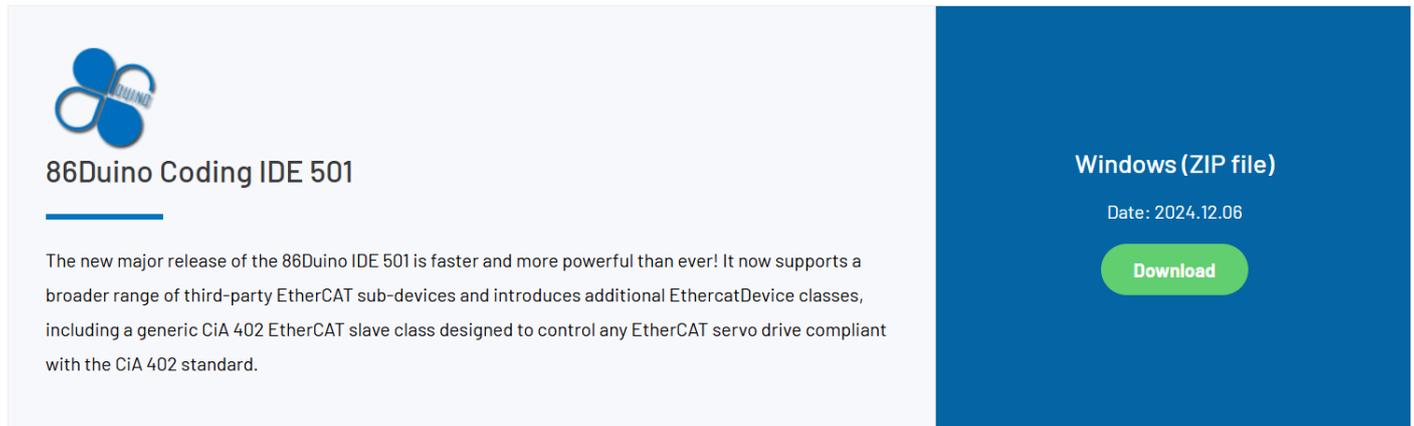
**注:** ロータリースイッチに設定された ID は機器の立ち上げ時に認識されます。

### ID 設定:

- 7セグメント LED にて実際の ID または設定された EtherCAT ID を次の様に表示します。
- すべてのロータリースイッチが「0」に設定されている場合、7セグメント LED は EtherCAT 物理アドレスを示します。サブデバイス(本製品)と M デバイスは接続されていないため、物理アドレスが割り当てられるまでは 0(ゼロ)を示します。M デバイスが各サブデバイス(本製品)に物理アドレスを割り当てると、その値が表示されます。
- ロータリースイッチが「0」以外の値に設定されている場合、7セグメント LED は対応する設定値 (EtherCAT 構成エイリアス)を示します。
- ID の 7セグメント LED が点滅している場合は、ID 値が設定されていないことを示しています。電源投入時に設定されます。

## 2. ソフトウェア/開発環境

<https://www.qec.tw/software/>から 86duino IDE をダウンロードしてください。



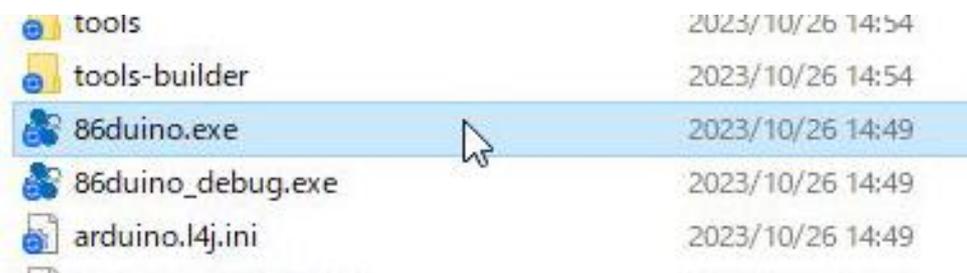
**86Duino Coding IDE 501**

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  
[Download](#)

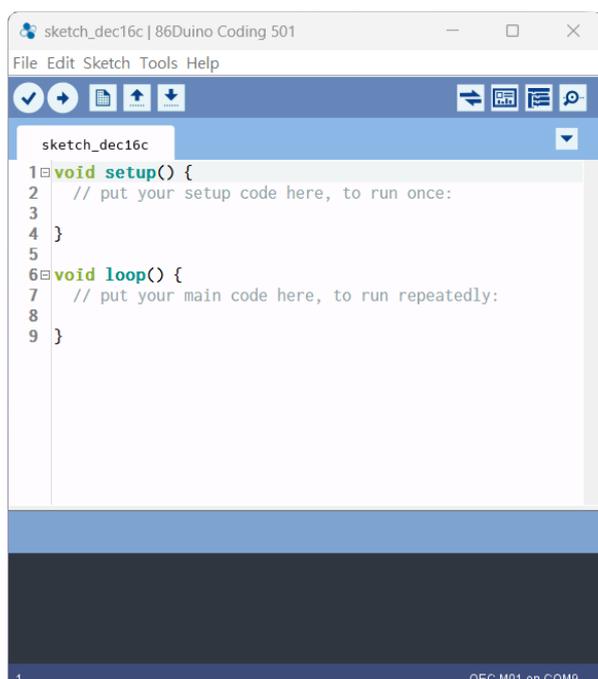
ダウンロード後、ダウンロードした zip ファイルを解凍してください。

追加のソフトウェアのインストールは必要ありません。86duino.exe をダブル・クリックして IDE を起動します。



**注:** Windows が警告を表示させた場合は、[詳細]を 1 回クリックし、[実行を続行]ボタンを 1 回クリックします。

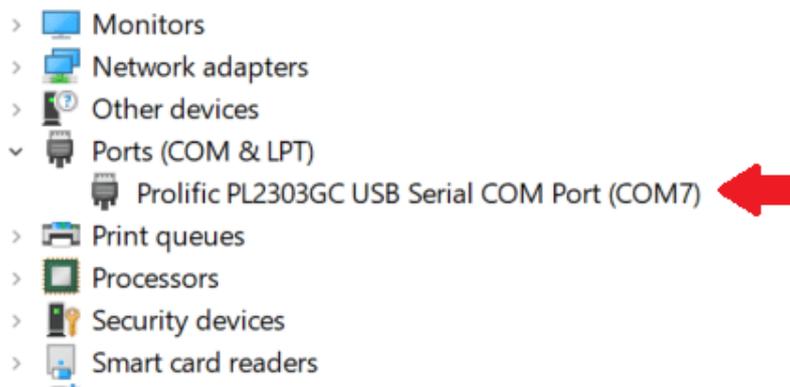
### 86Duino コーディング IDE 501+のイメージ



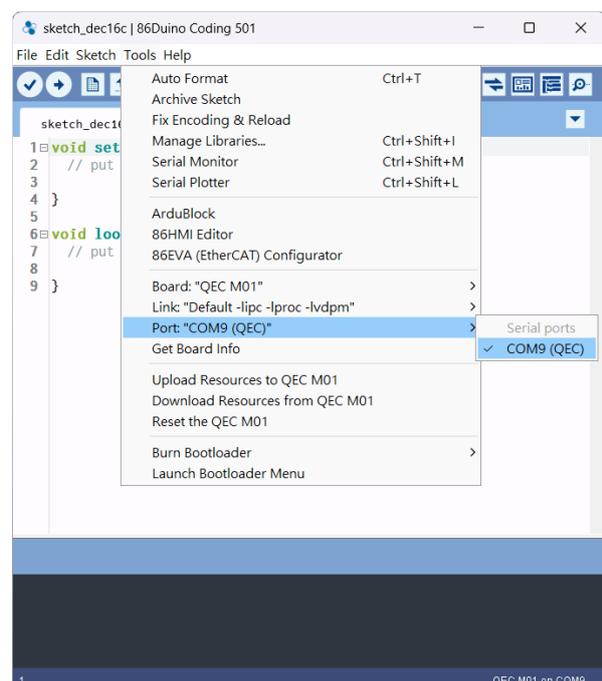
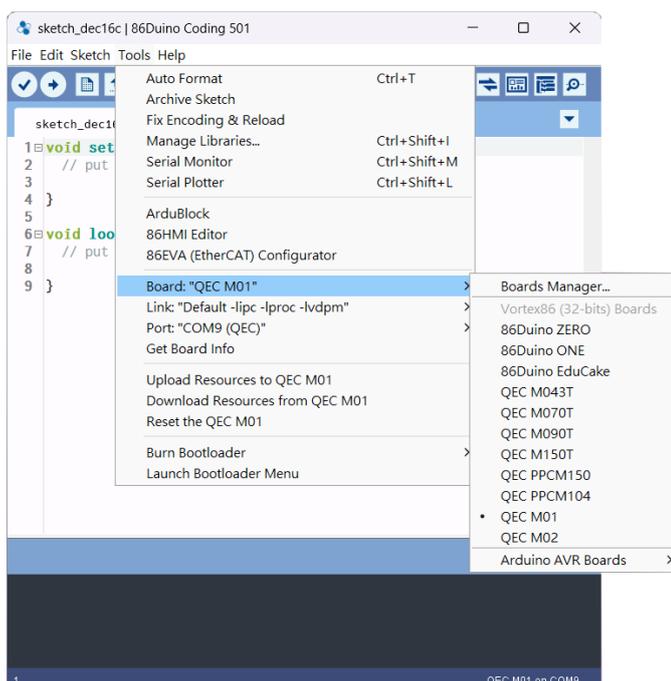
### 3. PC に接続して環境をセットアップする

以下の手順に従って開発環境をセットアップします:

1. Micro USB - USB ケーブルで QEC-M-01 を 86Duino IDE がインストールされた PC に接続します
2. QEC の電源を投入します。
3. PC で「デバイス・マネージャー」(Win+X キーを押した後のメニューで選択) -> 「ポート(COM および LPT)」を開き、ポートの内容を確認します。「Prolific PL2303GC USB Serial COM Port (COMx)」が検出されていることがわかります。検出されていない場合は、必要なドライバをインストールする必要があります(Windows PL2303 ドライバの場合は、[ここ](#)からダウンロードできます)



4. 86Duino IDE を開きます。
5. ボード(QEC M デバイス)の選択: IDE のメニューで、[Tools] > [Board] > [QEC-M-01] (または使用する QEC-M デバイス名) を選択します。
6. ポートの選択: IDE のメニューで、[Tools] > [Port] を選択し、デバイス・マネージャーで確認した QEC-M デバイスに接続する USB ポートを選択します (この場合、COM9 (QEC))



## 4. 86EVA によるコンフィグレーションとコードによる記述

この例では、86Duino IDE のグラフィカルなローコード・プログラミングツール 86EVA を使用して EtherCAT M デバイス(QEC-M-01)と NPM PULSERV II (クローズドループステッピングモータドライブ) を操作する方法を示します。

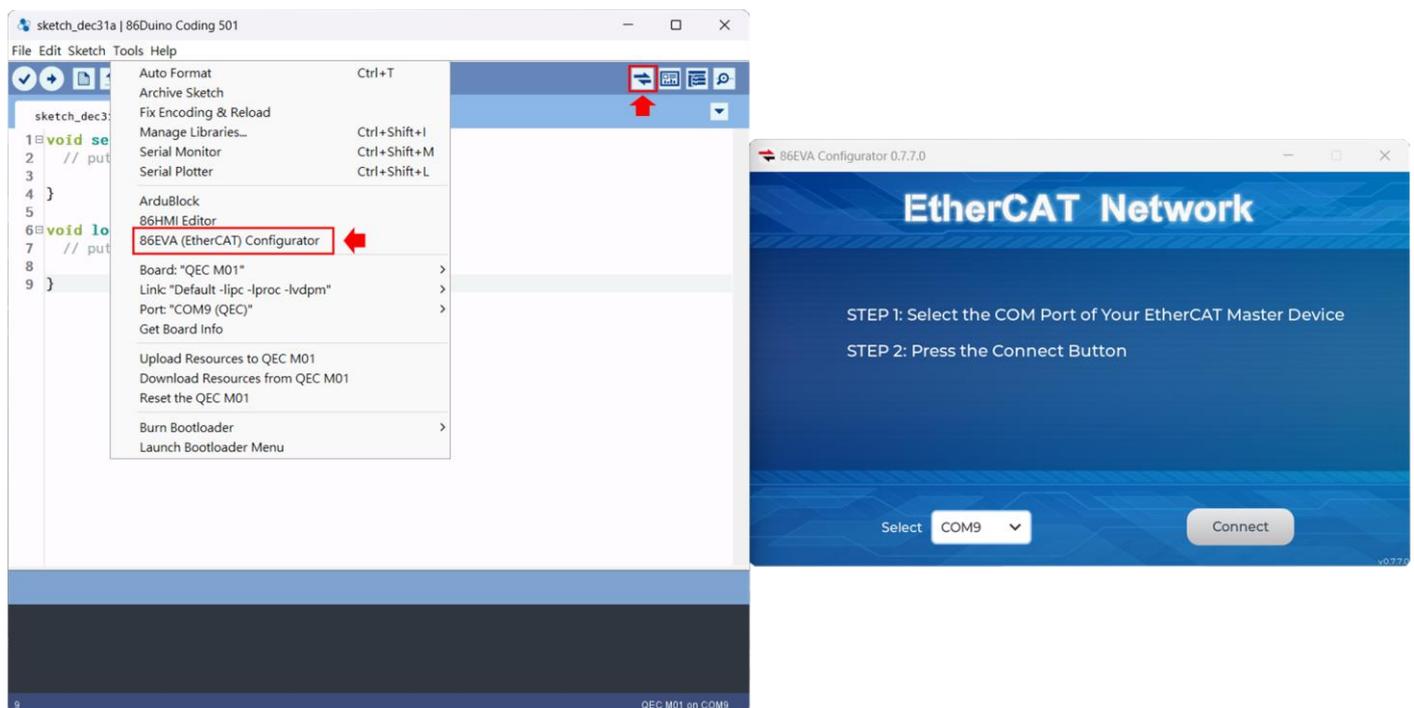
ソフトウェア・ツールの説明:

- 86EVA (EVA, EtherCAT-Based Virtual Arduino):  
86EVA は、86Duino IDE の EtherCAT ライブラリのベースのグラフィカル EtherCAT コンフィギュレータであり、86Duino の開発キットの 1 つです。

今回のプログラム例では、NPM PULSERV II を Profile Position (PP)モードに設定します。モータの位置は周期的に更新され、目標位置は 100,000 単位と 100,000 単位の間で交互に切り替わり、連続的な前進と後進の動きをシミュレートします。

### ステップ 1: 86EVA を起動してスキャンする

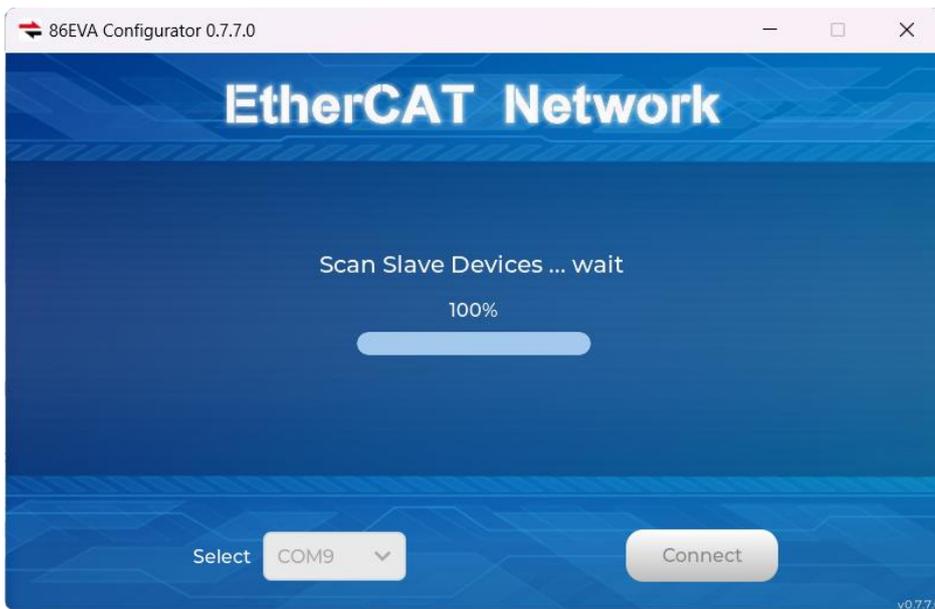
86EVA ツールは次のボタン、またはメニューから開くことができます。



QEC-M-01 の COM ポートが正しく選択されていることを確認後、「Connect」ボタンを押します。



EtherCAT ネットワークのスキャンが開始されます。



EtherCAT ネットワークのスキャンが完了すると、接続されているデバイスが表示されます。



## ステップ 2: パラメータを設定する

スキャンしたデバイス画像をダブルクリックすると対応するパラメータ設定画面に入ります。

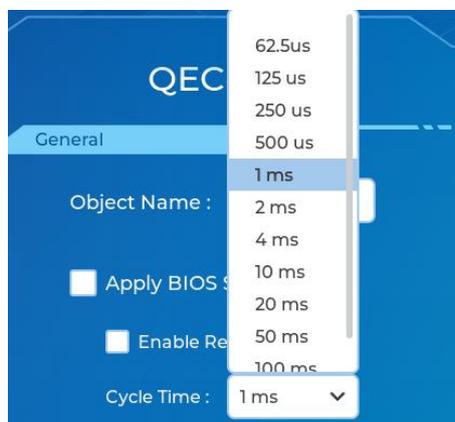
### QEC-M-01

QEC-M-01 の画像をダブルクリックすると、パラメータ設定が表示されます。



以下の設定を確認してください。

1. 「Apply BIOS Settings」をオフ
2. 「Cycle Time」で「1ms」を選択



左上の「Back」をクリックして戻ります。



## NPM PULSERVO II

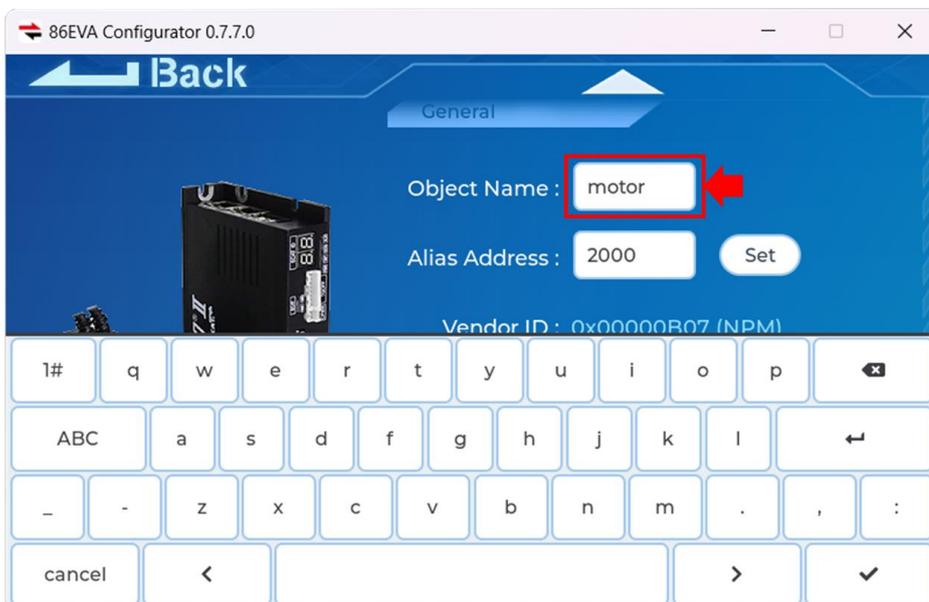
NPM Pulservo II の画像をダブルクリックしてパラメータ設定を確認します。



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

「Object Name」を “motor” に変更します。

「Object Name」入力枠をクリックするとキーボードが表示されます。



左上の「Back」をクリックして戻ります。

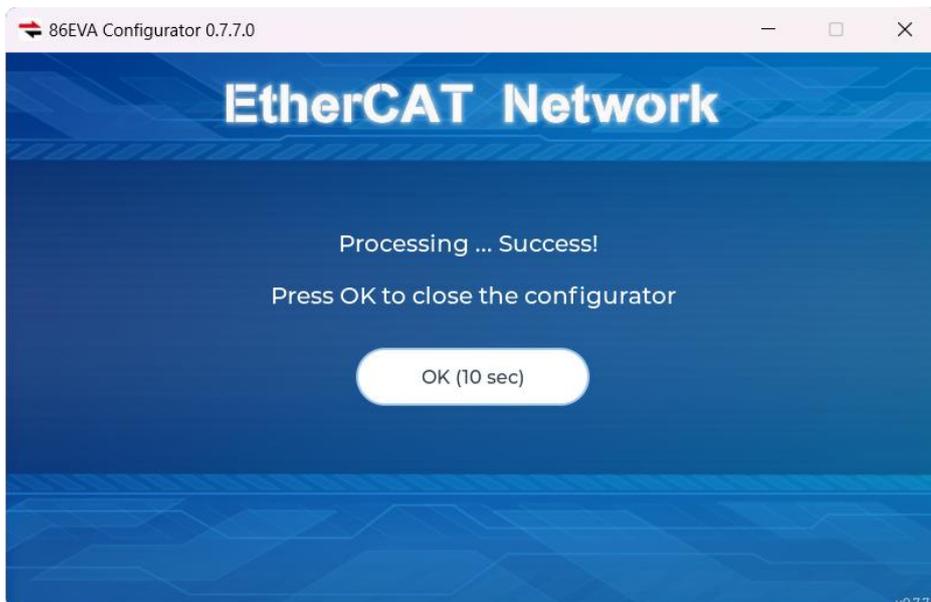


## ステップ 3: コードの生成

デバイスのパラメータを設定後、ホーム画面に戻り、右下にある「Code Generation」ボタンを押します。

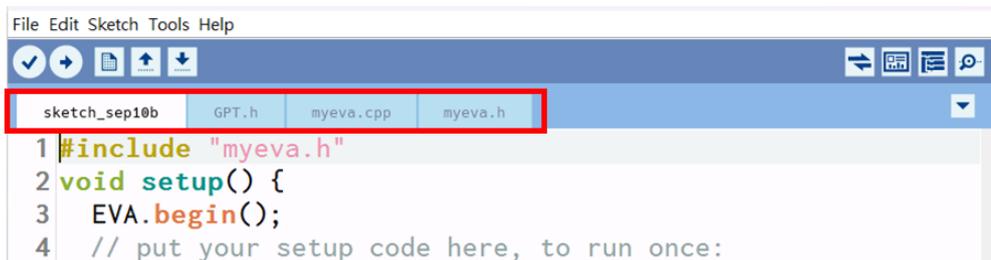


完了後「OK」ボタンをダブルクリックして 86EVA を閉じます。そのままにしておいても自動的に 10 秒後に終了します。



次のコードとファイルが生成されます:

- sketch\_sep10b: メイン・プロジェクト (.ino、プロジェクト名による)
- ChatGPT.h: ChatGPT 参照用パラメータ
- myeva.cpp: 86EVA の C++ プログラム・コード
- myeva.h: 86EVA のヘッダ・ファイル



```
File Edit Sketch Tools Help
sketch_sep10b GPT.h myeva.cpp myeva.h
1 #include "myeva.h"
2 void setup() {
3   EVA.begin();
4   // put your setup code here, to run once:
```

追記: 86EVA がコード生成後、メイン・プログラム(.ino)内に下記のコードが自動生成され、いずれかでも欠けると 86EVA が動作しなくなります。

1. #include "myeva.h": EVA ヘッダ・ファイルのインクルード
2. EVA.begin(); in setup(): EVA 機能の初期化

## ステップ 4: コードの記述

86EVA から生成されたプログラミングコードは、デフォルトで以下のように設定されています。

- NPM PULSERVO II: EthercatDevice\_CiA402 オブジェクト.
- CiA402 モード: Profile Position (PP) モード
- EtherCAT モード: ECAT\_SYNC.
- Distributed Clock(高精度時刻同期機能): Open, サイクルタイムに従う

以下の箇所はユーザにより設定されます。

- EtherCAT サイクルタイム: 1ms
- Device Object Name: QEC-M-01 は“EcatMaster”、PULSERVO II は“motor”.

このコードは、EtherCAT 通信を確立し、NPM PULSERVO II ドライバをプロファイル位置 (PP) モードで制御します。モーターの位置は周期的に更新され、ターゲット位置は 100,000 と 100,000 単位の間で交互に切り替わり、連続的な前進と後進の動きをシミュレートします。

This code establishes EtherCAT communication and controls the NPM Pulservo II driver in Profile Position (PP) mode. The motor's position is updated cyclically, and the target position alternates between 100,000 and 100,000 units, simulating continuous forward and reverse movements.

### A. Setup 関数

`setup()` 関数内で通信の初期化やモータを以下の手順で CiA402 Profile Position (PP)モードに設定します。

1. シリアル通信の初期化
  - 115,200 ボーにてシリアル通信を開始
2. 86EVA スタート
  - `EVA.begin()` 関数にて EtherCAT ネットワークを初期化して開始
3. Profile Position (PP) モードに設定
  - `setCiA402Mode(CIA402_PP_MODE)`関数にて PP モードに設定
4. モータを有効化
  - `enable()` 関数でモータを有効にし CIA402\_OPERATION\_ENABLED に移行
5. プロファイル変数の設定
  - モーションプロファイルタイプ: リニアランプ、プロファイル速度: 100,000、加速: 5,000、減速: 5,000.

### B. Loop 関数

`loop()` 関数では、モータの現在位置がシリアルモニターに表示され、モーターは繰り返しサイクルで前後に動きます。

1. モータの状態
  - ケース 0: モータを始動し、目標位置(100,000 単位)に移動します。コマンドが正常に実行されたら、次の状態に遷移します
  - ケース 1: モータが目標位置に到達するまで待ちます。目標位置に到達したら、次の状態に進みます

- ケース 2: モータを始動し、元の位置(-100,000 単位)に戻ります。コマンドが正常に実行されたら、次の状態に移行します。
- ケース 3: モータが元の位置に戻るまで待ちます。目標に到達したら、ステートマシンをケース 0 にリセットして、動作サイクルを繰り返します。

## 2. コードロジックの概要

- `pp_Run()` 関数にて位置移動を開始します
- `pp_IsTargetReached()` 関数にて目標位置に到達したかを確認します
- ケース 3 完了後、モータをリセットしケース 0 から開始します

### コード例:

```
#include "myeva.h"

int pp_state = 0;

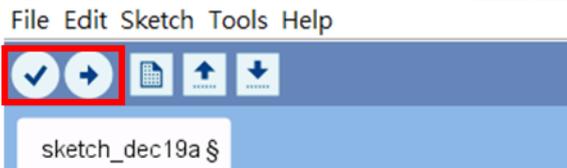
void setup() {
  Serial.begin(115200);
  while (!Serial);

  EVA.begin();
  motor.setCiA402Mode(CIA402_PP_MODE);
  Serial.print("Enable: "); Serial.println(motor.enable());
  motor.pp_SetMotionProfileType(0); // Linear ramp (trapezoidal profile)
  motor.pp_SetVelocity(100000);
  motor.pp_SetAcceleration(5000);
  motor.pp_SetDeceleration(5000);
}

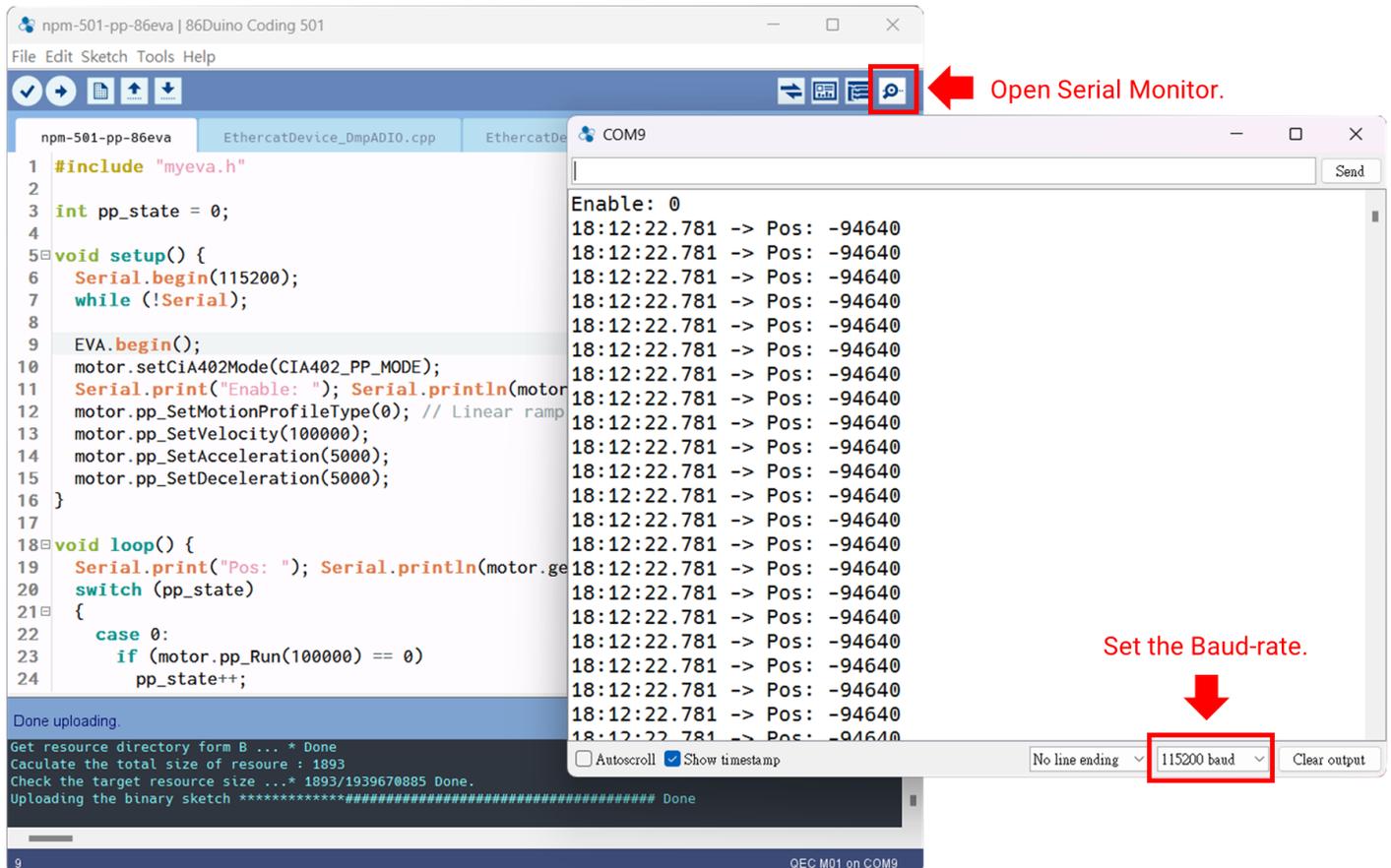
void loop() {
  Serial.print("Pos: "); Serial.println(motor.getPositionActualValue());
  switch (pp_state)
  {
    case 0:
      if (motor.pp_Run(100000) == 0)
        pp_state++;
      break;
    case 1:
      if (motor.pp_IsTargetReached())
        pp_state++;
      break;
    case 2:
      if (motor.pp_Run(-100000) == 0)
        pp_state++;
  }
}
```

```
break;
case 3:
  if (motor.pp_IsTargetReached())
    pp_state = 0;
  break;
}
}
```

**注:** コードを記述後、ツールバーの  をクリックしてコンパイルします。  
コンパイル完了後エラーがないことを確認し  をクリックしてプログラムをアップロードします。



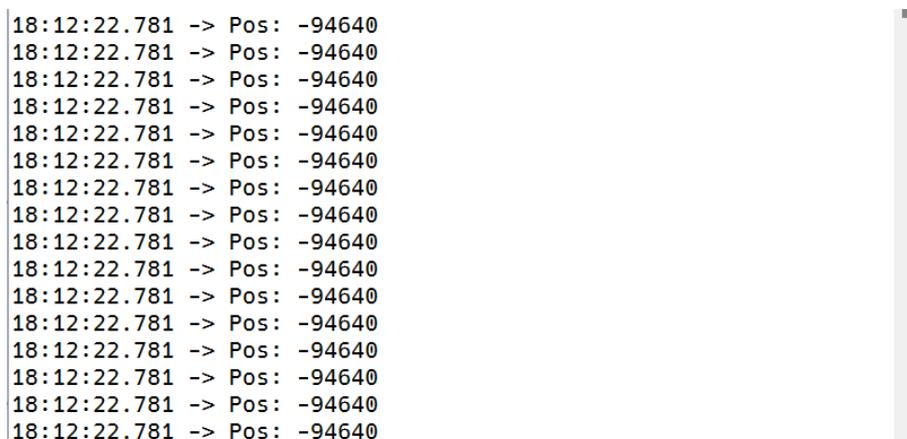
QEC-M-01 にプログラムをアップロード後、86Duino IDE でシリアルモニタを開きます。  
ボーレートが設定した値と同じであることを確認してください。



EtherCAT 通信が成功すると、シリアルモニタは “Enable: 0” を表示します。



そしてモータの現在位置を表示します。



## トラブルシューティング

### QEC-M-01 へのコードのアップロードが成功しない

コードのアップロードに成功しない場合は、86EVA を開いて QEC EtherCAT M デバイスの環境に異常がないか確認してください。下図のようにブートローダ、EtherCAT ファームウェア、EtherCAT ツールを含む QEC EtherCAT M デバイスの環境を更新してください。



アップデートの進め方を説明します：

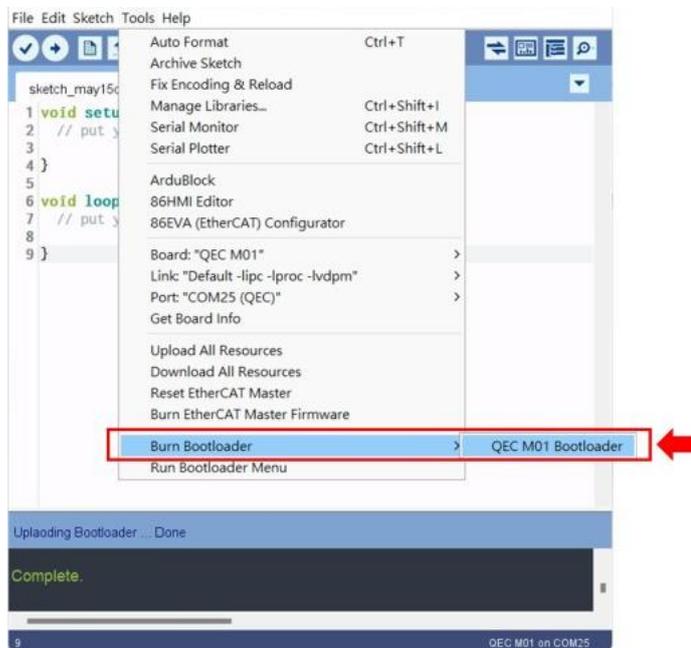
#### ステップ 1: QEC-M のセットアップ

1. 86Duino IDE 500+ (または最新バージョン) をダウンロードしてインストールします: [Software](#) からダウンロードできます。
2. QEC-M を PC に接続: USB ケーブルを使用して QEC-M を PC に接続します。
3. 86Duino IDE を開く: インストールが完了したら、86Duino IDE ソフトウェアを開きます。
4. ボードの選択: IDE メニューから、[Tools] > [Board] > [QEC-M-01] (または使用中の QEC-M の型名) を選択します。
5. ポートの選択: IDE メニューから [Tools] > [Port] を選択し、QEC-M が接続されている USB ポートを選択します。

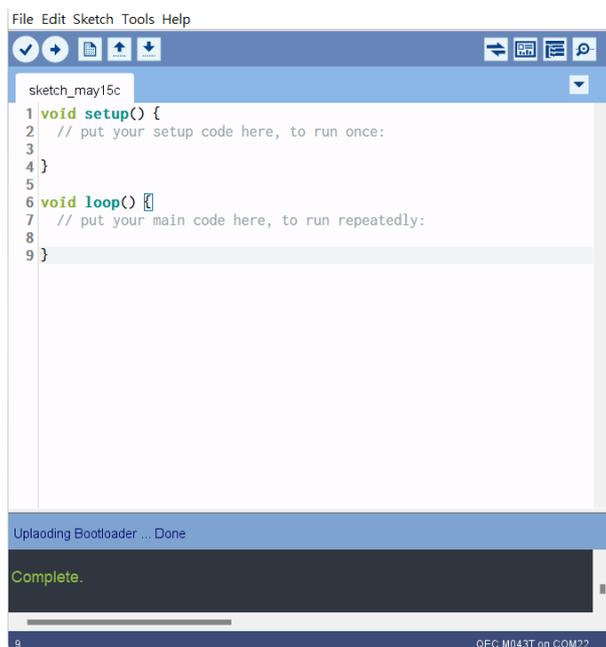
## ステップ 2: 「Burn Bootloader」ボタンをクリック

QEC-M 製品に接続後、「Tools」>「Burn Bootloader」に移動します。現在選択されている QEC-M 名が表示されます。その上をクリックすると更新プロセスが開始されますが約 5~20 分かかります。

QEC-M-01:



## ステップ 3: アップデートを完了



上記の手順を完了すると、QEC-M は最新の開発環境バージョンに正常に更新されます。

# 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.

本書に記載されているブランド名および製品名は、各社の所有物および登録商標です。  
本書に記載されている名称はすべて、識別目的のみに使用されます。

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

©ICOP Technology Inc. 2024

日本語版資料は、英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。