スタートガイド

オリエンタルモーター AZD4A-KED

EtherCAT Sub デバイス (PP モード)

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

改訂履歴

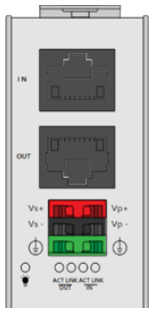
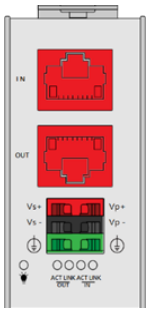
記述日	バージョン	備考
2025/8/8	Version1.0	New Release.

序文

本ガイドでは、QEC-M-01 (EtherCAT M デバイス) とオリエンタルモーター株式会社製 4軸 AZ シリーズ EtherCAT コントローラ AZD4A-KED の CiA402 Profile Position(PP)モードによる動作での使用方法を説明します。

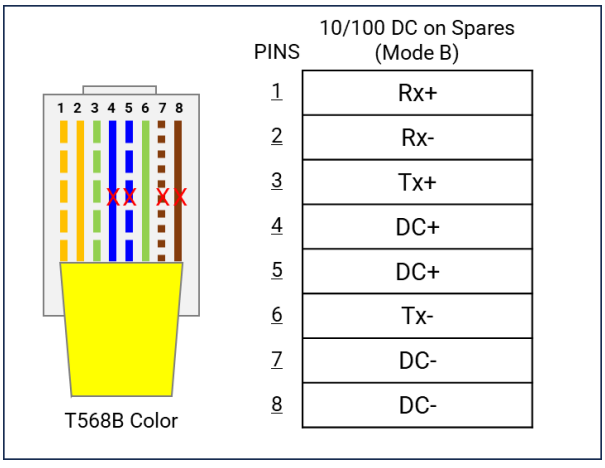
注意 QEC 機器の PoE (Power over Ethernet)について

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。

	
非 PoE タイプ	PoE タイプ

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 マスタを他社の EtherCAT スレーブに接続する場合)。

3. QEC PoE 電源は最大 24V/3A です。

1. ハードウエアの接続と配線

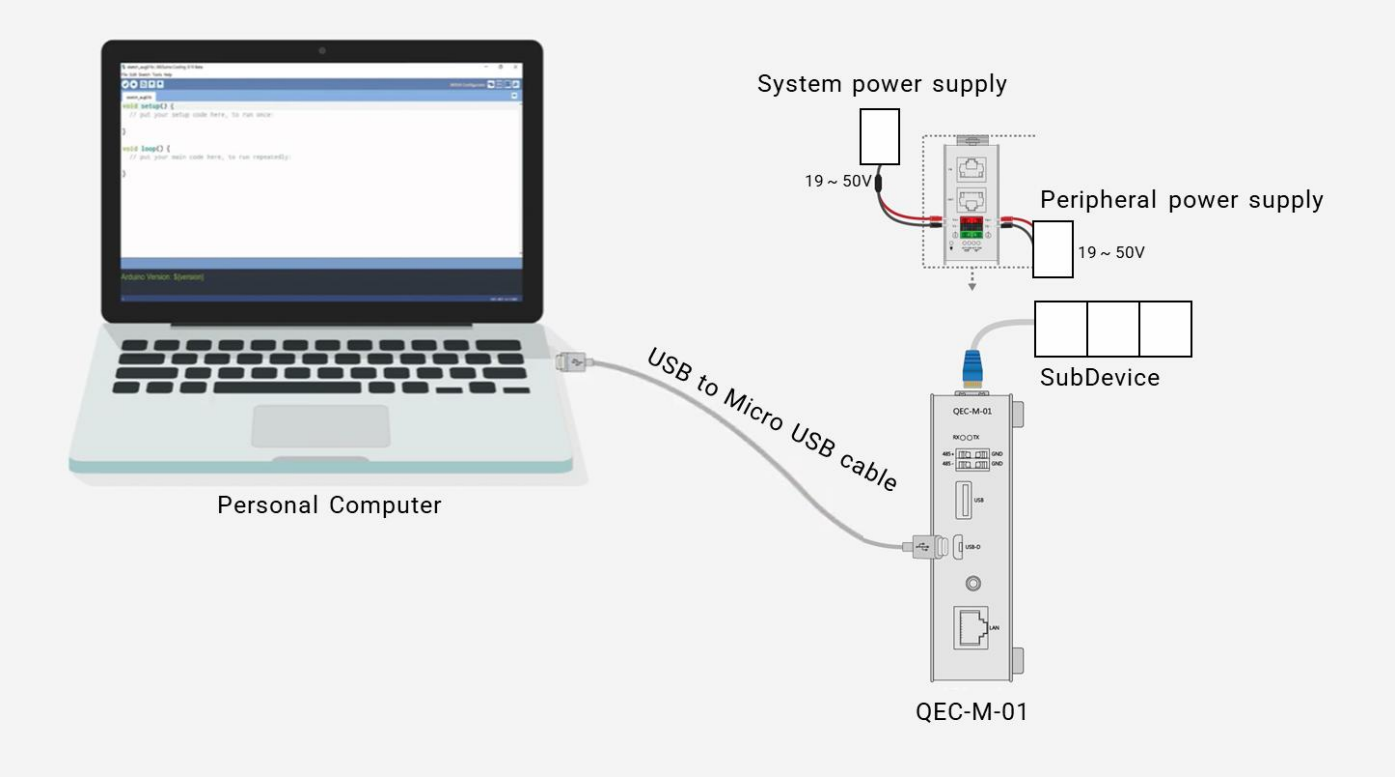
ここでは次のデバイスを使用します：

1. QEC-M-01 (EtherCAT M デバイス)
2. AZD4A-KED, 4 軸 AZ シリーズ EtherCAT コントローラ
3. AZM66MK, 2.36 in. (60 mm) アブソリュートセンサ搭載 AZ シリーズ ステッピングモータ (ブレーキ) (DC 入力)
4. AZM46A0K, 1.65 in. (42 mm) アブソリュートセンサ搭載 AZ シリーズ ステッピングモータ (DC 入力)

1.1 QEC-M-01

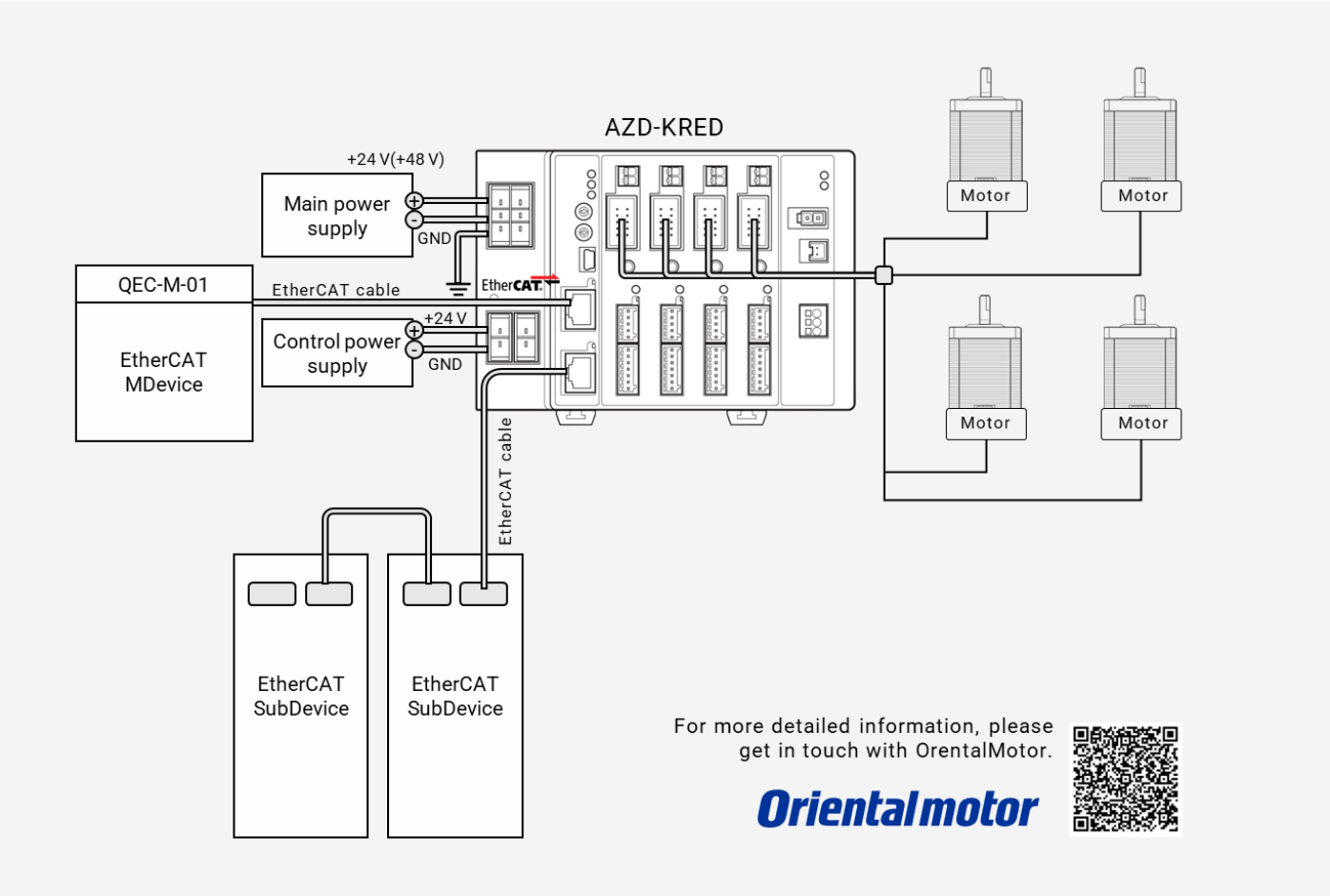
QEC EtherCAT M デバイス

1. 電源：
   24V 電源を QEC EtherCAT マスタのヨーロッパ・タイプ端子 Vs+/Vs- および Vp+/Vp- に接続
2. EtherCAT 通信：
3. EtherCAT 出力ポート (上側) から AZD4A-KED の EtherCAT 入力ポートに RJ45 ケーブルで接続



1.2 AZD4A-KED

AZD4A-KED, 4 軸 AZ シリーズ EtherCAT コントローラ/ドライバ (24/48 VDC).
下図はモーターを接続した場合の例です。



1. ケーブルは、オリエンタルモーター製のケーブルです。別途ご購入が必要です。
2. 制御電源を接続することで、主電源が遮断されてもモニタリングを継続することができます。必要に応じて接続してください。
3. 誤った配線は内部入力回路をショートさせる恐れがあるため、ブレーカーまたは回路保護装置を接続することが推奨されます。

注意:

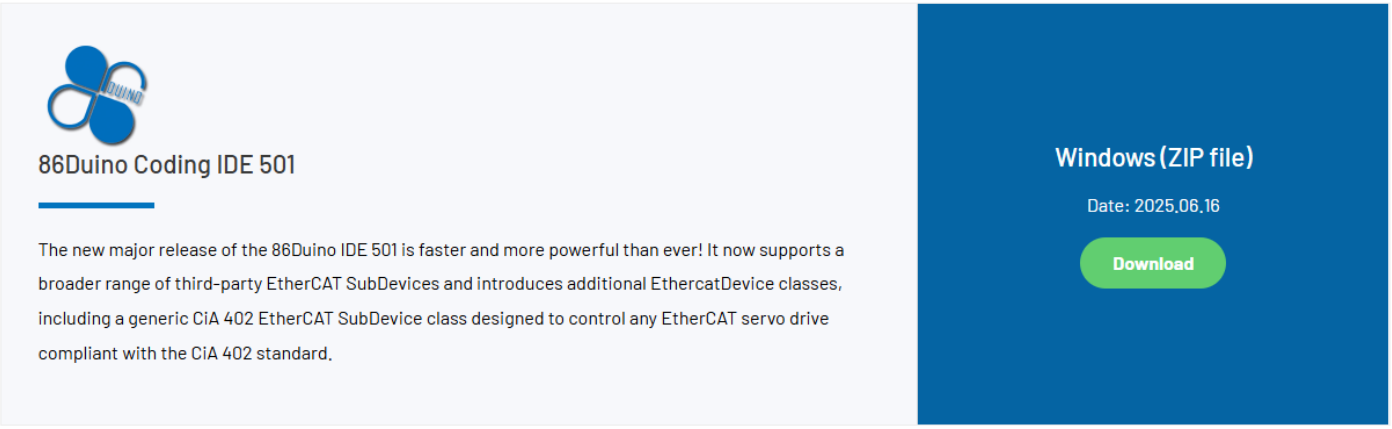
- コネクタは確実に接続してください。接続が確実でないと、モーターやドライバの誤動作や破損の原因となります。
- ケーブルを接続する際は、コネクタに負荷がかからないように固定してください。コネクタに負荷がかかると接続不良となり、ドライバの誤動作の原因となります。
- モーターとドライバの配線距離は 10m 以下としてください。配線距離が 10m を超えると、ドライバから発生する電気ノイズが大きくなることがあります。
- 主電源と制御電源のケーブル長は 2m 以下にしてください。

メモ：

- コネクタの着脱を行う前に、主電源と制御電源を切り、PWR/ALM LED が消灯していることを確認してください。
- コネクタを外すときは、コネクタのラッチを指で押しながら引き抜いてください。

2. ソフトウエア/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。



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。

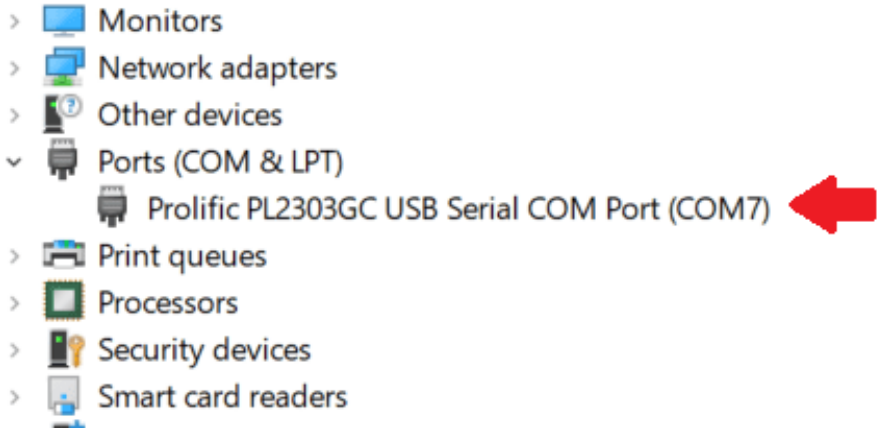
＊注: Windows が警告を表示させた場合は、[詳細]を 1 回クリックし、[実行を続行]ボタンを 1 回クリックします。

86Duino コーディング IDE 501+ は下図です。

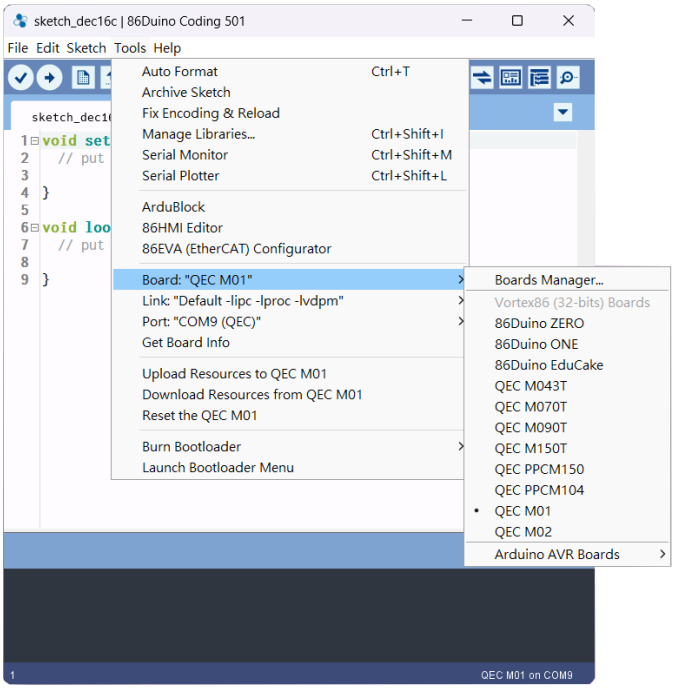
3. PC に接続して環境をセットアップする

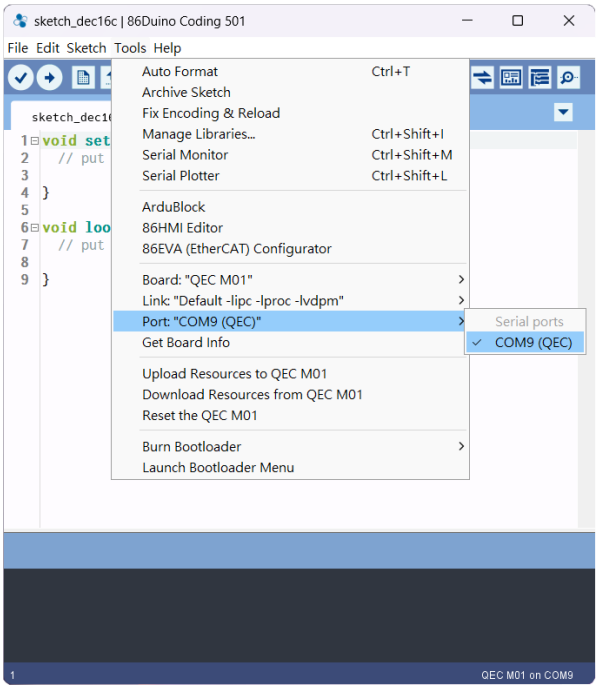
以下の手順に従って開発環境をセットアップします:

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 マスタ)の選択: IDE のメニューで、[Tools] > [Board] > [QEC-M-01] (または使用する QEC-M マスタ型名) 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 マスタに接続する USB ポートを選択します (この場合、COM9 (QEC))。





4. Write code

86Duino IDE の EtherCAT ライブラリを介して、EtherCAT M デバイス(QEC-M-01)とオリエンタルモーター 4 軸 AZ シリーズ EtherCAT コントローラ(AZD4A-KED)をコンフィギュレーションできます。Arduino 開発環境(コーディング)には 2 つの主要な部分があります：初期化のための setup() とメイン・プログラムの loop()です。EtherCAT ネットワークを操作する前に、一度コンフィグレーションする必要があります。そのプロセスにより EtherCAT デバイスは Pre-OP から OP モードになります。

以下のプログラムは、AZD4A-KED ドライバをプロファイル位置（PP）モードで 4 つの CiA402 オブジェクトに設定し、連続的な往復運動を実現します。

- EtherCAT サイクルタイム: 1ms
- 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

EthercatMaster の object, master は QEC-M-01 を意味し、EthercatDevice_CiA402 object 配列 motors[0]から motors[3]は AZD4A-KED ドライバを意味します。

1. **Setup()関数内の説明:**
   - Serial 通信の初期化 (115200)
   - EtherCAT M デバイスを開始し、EtherCAT ステートマシンを PRE-OPERATIONAL ステートに切り替える
   - AZD4A-KED の各軸を EtherCAT ネットワークに割り当てを Profile Position (PP)モードに設定
   - 各ドライバを DC(Distributed Clock)モードで設定、EtherCAT 通信と同じサイクルタイムに設定
   - AZD4A-KED 構成では、sdoDownload8(0x4148, I + 1, 0x01)によりホーミング関連のロックを解放する必要がある（ベンダーオブジェクトによれば、これは便宜上の措置とのこと）。
   - EtherCAT M デバイスをスタート。start()関数で EtherCAT サイクルタイムとモードを設定、EtherCAT ステートマシンを OPERATIONAL ステートに切り替え
   - AZD4A-KED を enable へ、 CiA402 ステートを CIA402_OPERATION_ENABLED に変更
   - プロファイルパラメータ等を設定する:
     Velocity = 10,000, Acceleration = 5,000, Deceleration = 5,000
   - delay(1000)を使用して、正常に変更されるまで待つ

2. **Loop()関数内の説明：**
   ループは各モーターの位置を継続的に監視し、その PP モードステートマシンを管理します。
   - モーターがアイドル状態（isMoving == false）の場合、その方向フラグに基づいて次の絶対目標位置を計算し、pp_Run()コマンドを送信する
   - 目標位置に到達した場合（pp_IsTargetReached()）、方向を反転させ、次の動作に向けてモーターをアイドル状態とする
   - 全てのモーターの位置値は、各サイクルごとにシリアルモニターに出力される

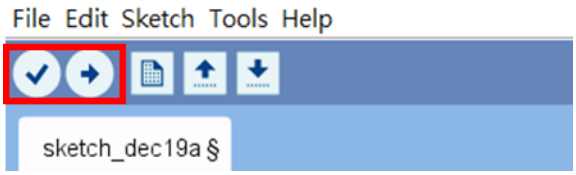
各軸は+100,000 から-100,000 の位置単位の間で絶対移動します。
4 軸すべてが独立して動作しますが、同時に動作します。
動作はノンブロッキングであり、あるモーターの動作が別のモーターを待つことはありません。

プログラム例:

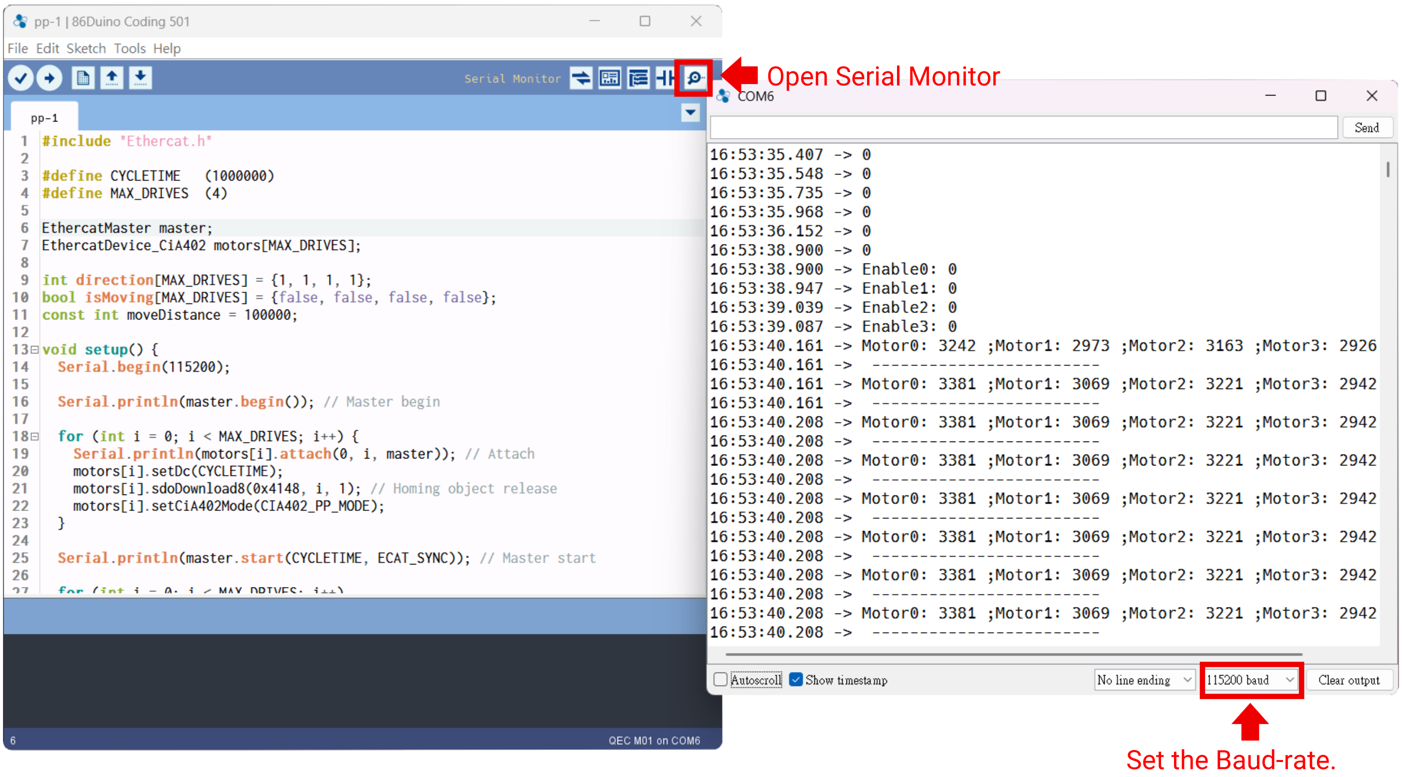
```
#include "Ethercat.h"

#define CYCLETIME      (1000000)
#define MAX_DRIVES    (4)

EthercatMaster master;
EthercatDevice_CiA402 motors[MAX_DRIVES];

int direction[MAX_DRIVES] = {1, 1, 1, 1};
bool isMoving[MAX_DRIVES] = {false, false, false, false};
const int moveDistance = 100000;

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

   Serial.println(master.begin()); // Master begin

   for (int i = 0; i < MAX_DRIVES; i++) {
      Serial.println(motors[i].attach(0, i, master)); // Attach
      motors[i].setDc(CYCLETIME);
      motors[i].sdoDownload8(0x4148, i + 1, 0x01); // Homing object release
      motors[i].setCiA402Mode(CIA402_PP_MODE);
   }

   Serial.println(master.start(CYCLETIME, ECAT_SYNC)); // Master start

   for (int i = 0; i < MAX_DRIVES; i++)
   {
      Serial.print("Enable");
      Serial.print(i);
      Serial.print(": ");
      Serial.println(motors[i].enable());

      motors[i].pp_SetVelocity(10000);
      motors[i].pp_SetAcceleration(5000);
      motors[i].pp_SetDeceleration(5000);
   }
```

```
    delay(1000);
}

void loop() {

    for (int i = 0; i < MAX_DRIVES; i++) {
        // Print out the postion
        Serial.print("Motor");
        Serial.print(i);
        Serial.print(": ");
        Serial.print(motors[i].getPositionActualValue());
        Serial.print(" ;");

        // Moving action by PP
        if (!isMoving[i]) {
            int target = direction[i] * moveDistance;
            motors[i].pp_Run(target, CIA402_PP_ABSOLUTE, true);
            isMoving[i] = true;
        }

        if (isMoving[i] && motors[i].pp_IsTargetReached()) {
            direction[i] *= -1;         // reverse
            isMoving[i] = false;        // reset
        }
    }
    Serial.println(" ");
    Serial.println(" ---------------------- ");

}
```

注:
コードを記述したならば、ツールバーの  をクリックしてコンパイルします。コンパイルが完了しエラーがないことを確認したら  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したら、86Duino IDE でシリアルモニタを開きます。シリアルボーレートがあなたの設定と同じであることを確認してください。



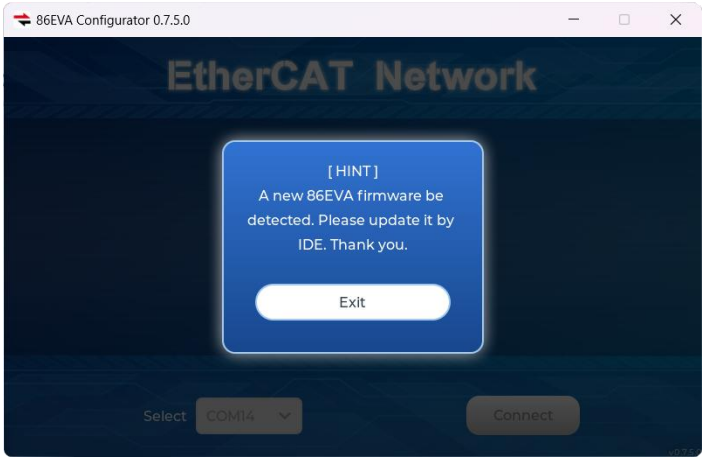
EtherCAT 通信が成功すると、シリアルモニタは各モーターに “0” と “Enable：0”を表示します。各モーターの現在の位置をシリアルモニターに出力します。

```
16:53:35.407 -> 0
16:53:35.548 -> 0
16:53:35.735 -> 0
16:53:35.968 -> 0
16:53:36.152 -> 0
16:53:38.900 -> 0
16:53:38.900 -> Enable0: 0
16:53:38.947 -> Enable1: 0
16:53:39.039 -> Enable2: 0
16:53:39.087 -> Enable3: 0
16:53:40.161 -> Motor0: 3242 ;Motor1: 2973 ;Motor2: 3163 ;Motor3: 2926
16:53:40.161 ->  ------------------------
16:53:40.161 -> Motor0: 3381 ;Motor1: 3069 ;Motor2: 3221 ;Motor3: 2942
16:53:40.161 ->  ------------------------
16:53:40.208 -> Motor0: 3381 ;Motor1: 3069 ;Motor2: 3221 ;Motor3: 2942
16:53:40.208 ->  ------------------------
16:53:40.208 -> Motor0: 3381 ;Motor1: 3069 ;Motor2: 3221 ;Motor3: 2942
16:53:40.208 ->  ------------------------
16:53:40.208 -> Motor0: 3381 ;Motor1: 3069 ;Motor2: 3221 ;Motor3: 2942
16:53:40.208 ->  ------------------------
16:53:40.208 -> Motor0: 3381 ;Motor1: 3069 ;Motor2: 3221 ;Motor3: 2942
16:53:40.208 ->  ------------------------
16:53:40.208 -> Motor0: 3381 ;Motor1: 3069 ;Motor2: 3221 ;Motor3: 2942
16:53:40.208 ->  ------------------------
16:53:40.208 -> Motor0: 3381 ;Motor1: 3069 ;Motor2: 3221 ;Motor3: 2942
16:53:40.208 ->  ------------------------
```

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：

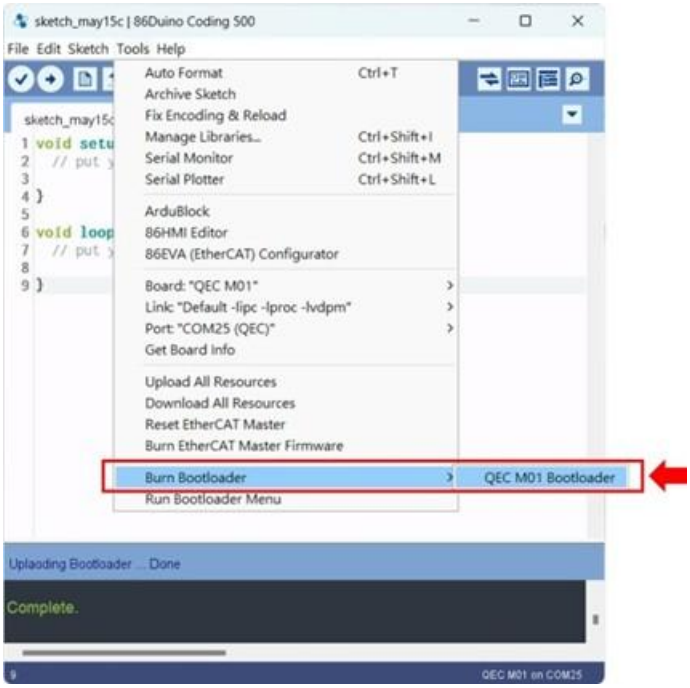
ステップ 1: QEC-M のセットアップ

1. 86Duino IDE 501 (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。

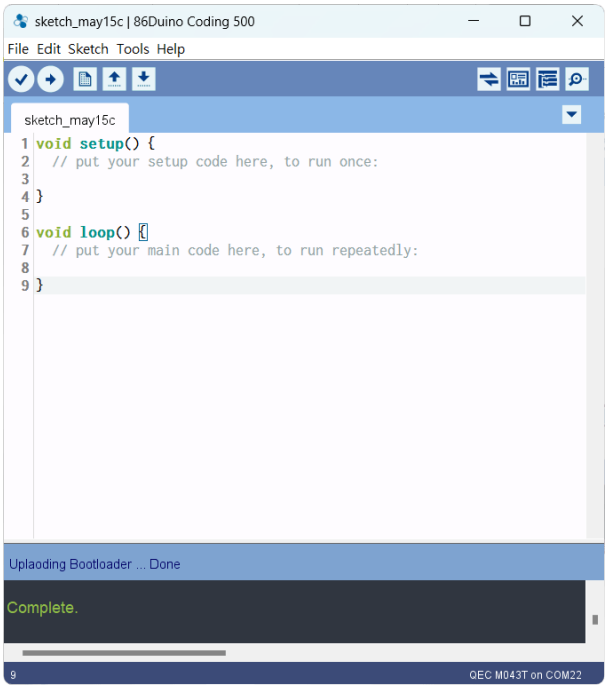
ステップ 2:「Burn Bootloader」ボタンをクリック

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。これには約 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.