スタートガイド

NPM EC-AD1441A4EtherCAT ドライバ

CiA402 PP モード(4 軸)

86Duino Coding IDE 501

EtherCAT Library

(Version 1.0)

改訂履歴

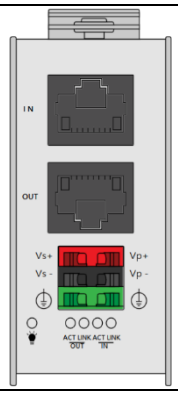
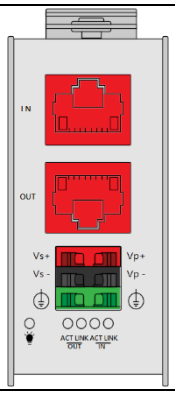
記述日	バージョン	備考
2025/1/23	Version1.0	New Release.

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 EC-AD1441A4 (2 相バイポーラ定電流、4 軸ステッピングモータードライバー)の使用方法を説明します。

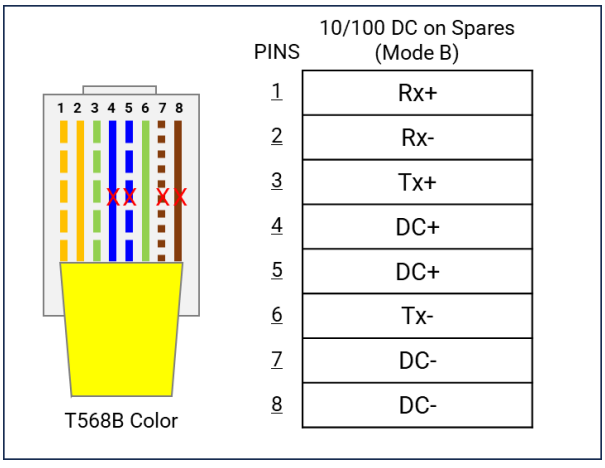
注意 QEC 機器の PoE (Power over Ethernet)について

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:

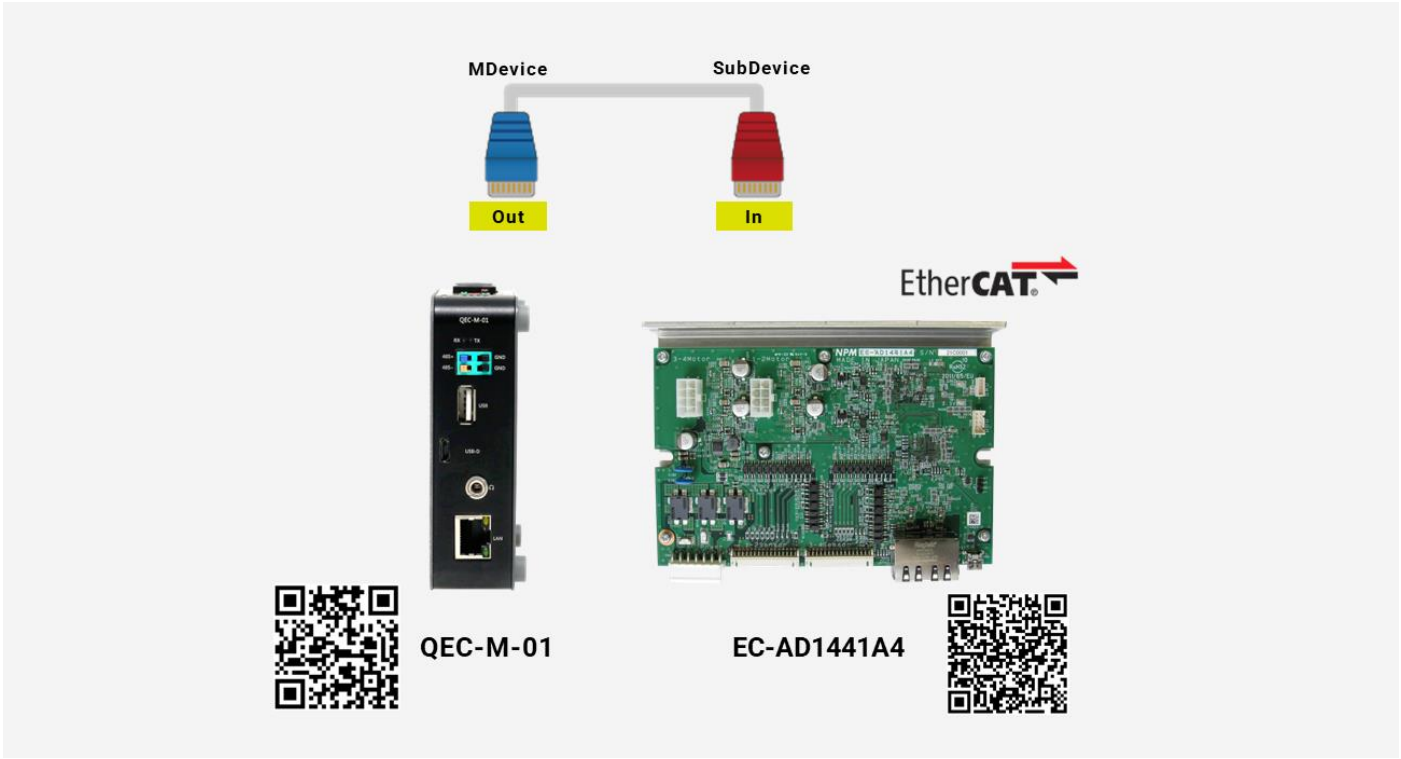


2. PoE デバイスと非 PoE デバイスを接続するときは、必ず EtherNet ケーブルのピン 4、5、7、8 を切断してください (例えば PoE 対応の QEC EtherCAT M デバイスを他社の EtherCAT サブデバイスに接続する場合)。
3. QEC PoE 電源は最大 24V/3A です。

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：

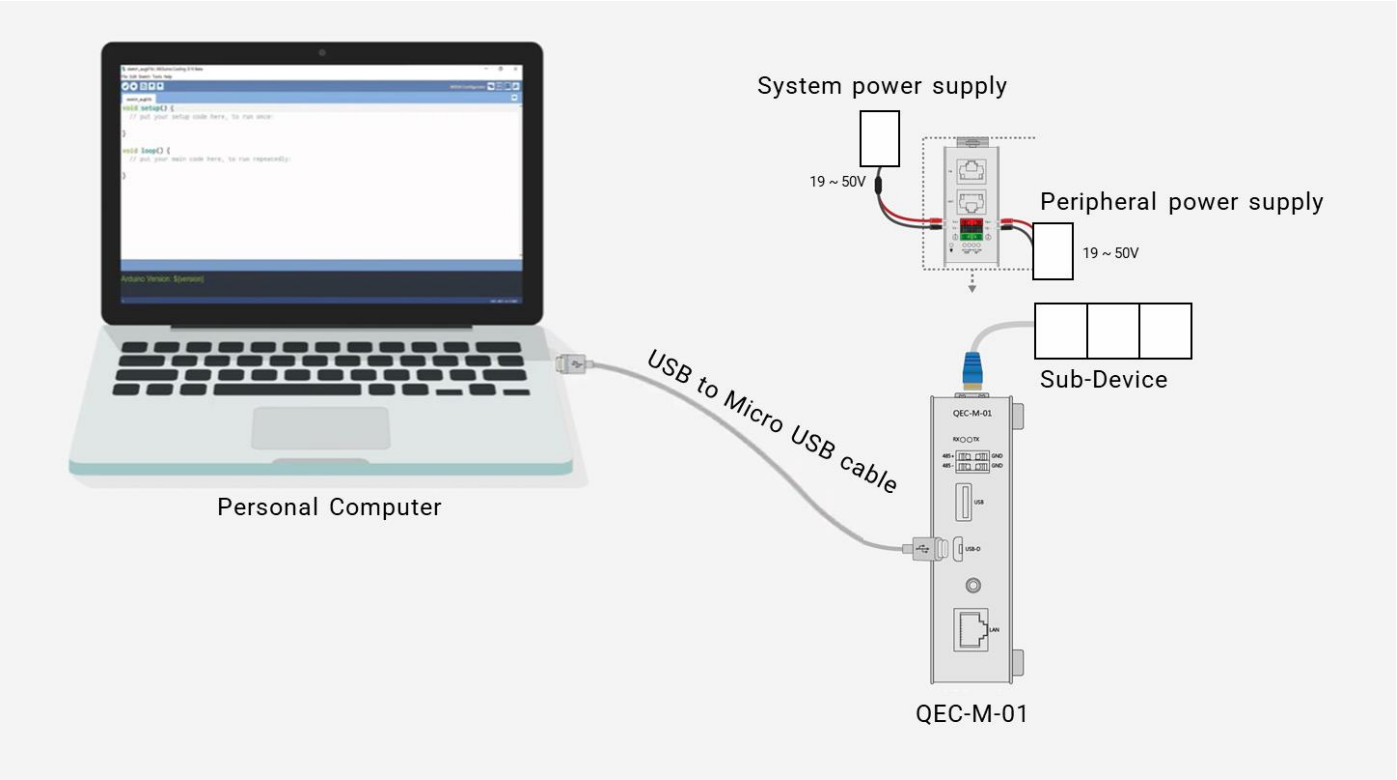
1. QEC-M-01 (EtherCAT M デバイス)
2. NPM EC-AD1441A4 (2 相バイポーラ定電流、4 軸ステッピングモータードライバー)
3. 24V 電源, EU タイプ端子接続ケーブル, LAN ケーブル
4. 86STEP-42*4 (エンコーダ内蔵ステッピングモータ, 42 mm square)



1.1 QEC-M-01

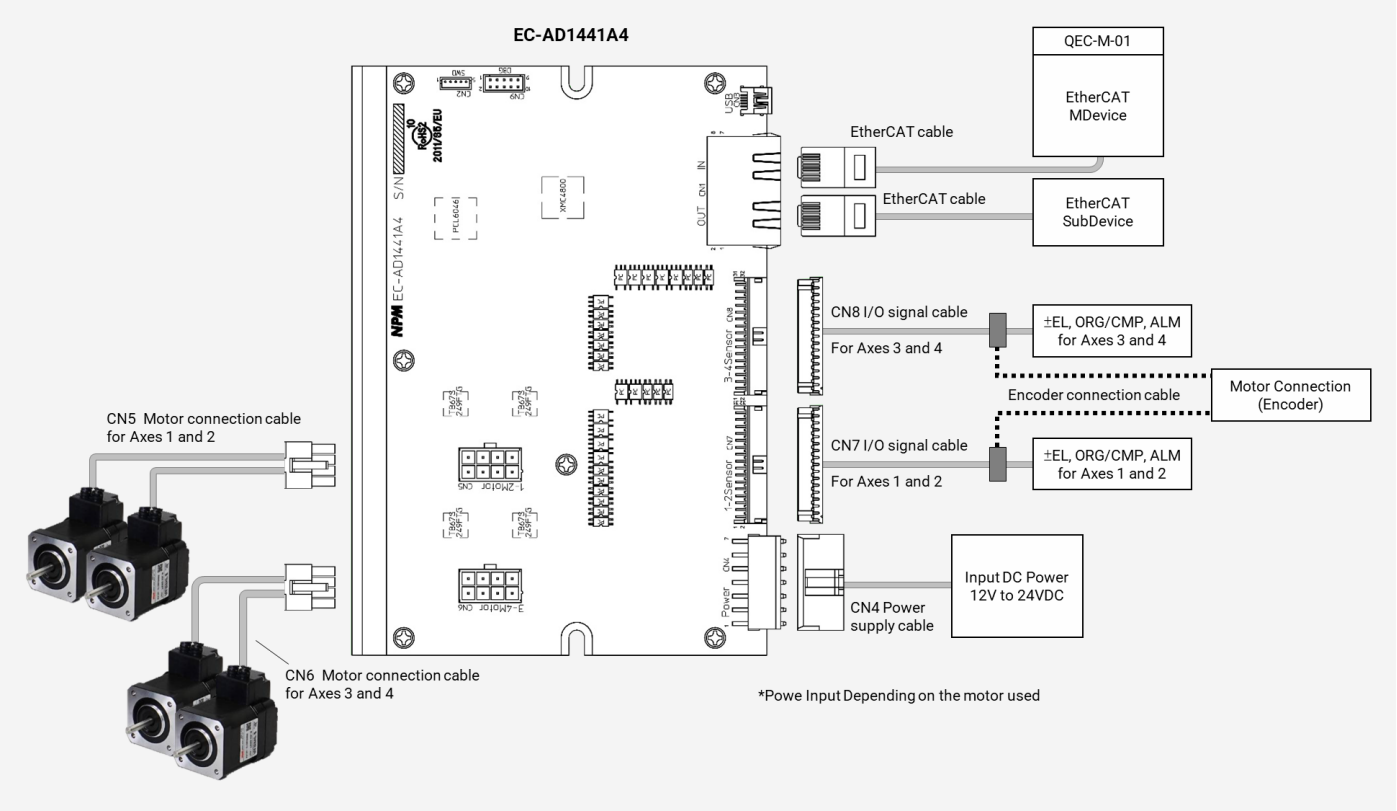
QEC EtherCAT M デバイス

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 EC-AD1441A4

EC-AD1441A4 は 2 相バイポーラ定電流、4 軸ステッピングモータードライバーです。
下図は **86STEP-42** モータを接続した場合の例です。



1. EtherCAT 接続:
   - M デバイス(QEC-M01)とサブデバイス間 2 つの EtherCAT 用通信ポート(IN、OUT) を接続。
2. 信号接続:
   - CN7: 軸 1,2 用 IO 信号 (EL, ORG, CMP, ALM).
   - CN8: 軸 3,4 用 IO 信号 (EL, ORG, CMP, ALM).
   - エンコーダ接続: 各軸用モータ位置追跡
3. モータ接続:
   - CN5: 軸 1,2 用モータ接続.
   - CN6: 軸 3,4 用モータ接続.
4. 電源供給:
   - CN4: DC 電源入力(12V~24V) デバイス及びモータ用
5. 安全性とステータス:
   - 安全性の為、緊急信号(EMG, ALM)を内蔵しています。
   - LED によりデバイスと EtherCAT の状態を示します。

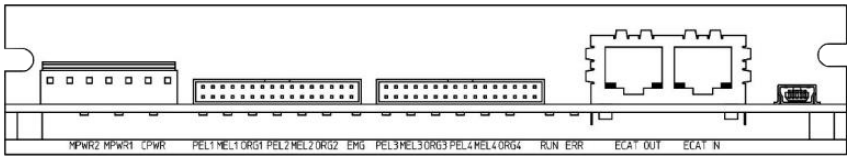
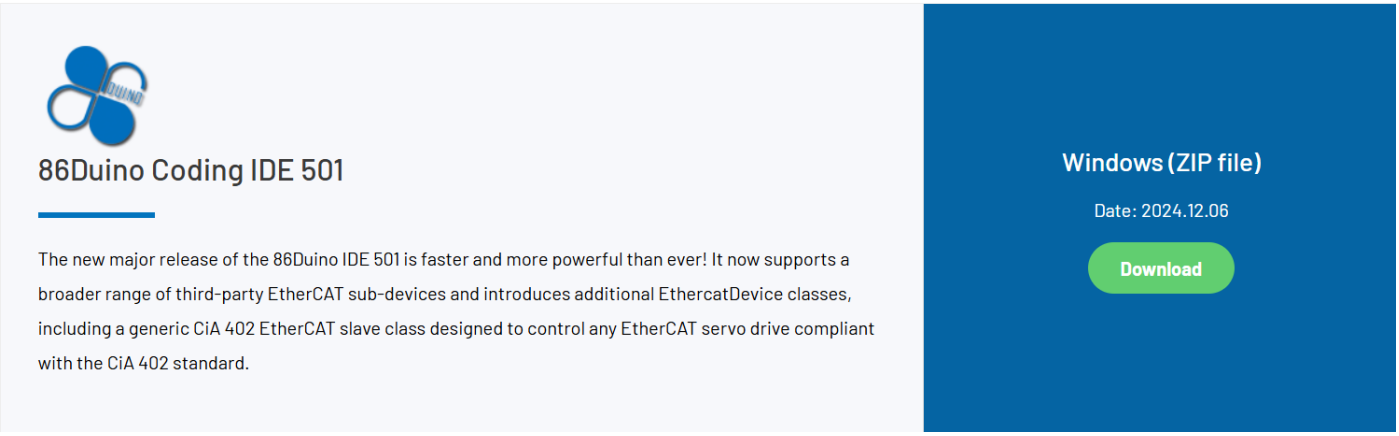


Figure: EC-AD1441A4 側面

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。

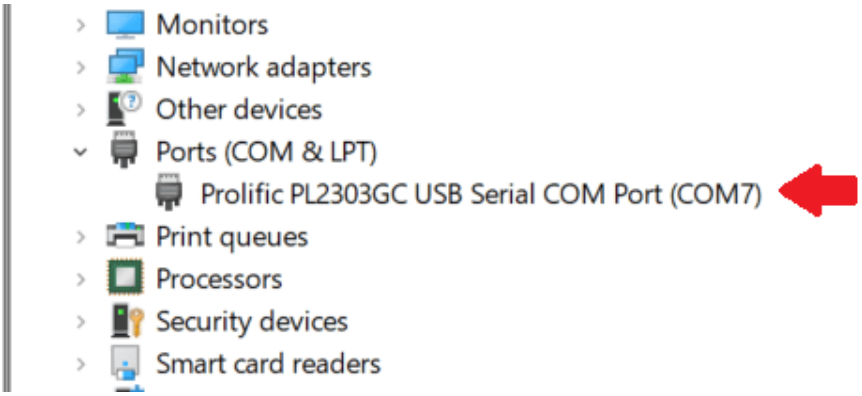
注: Windows が警告を表示させた場合は、[詳細]を 1 回クリックし、[実行を続行]ボタンを 1 回クリックします。

86Duino コーディング IDE 501+のイメージ

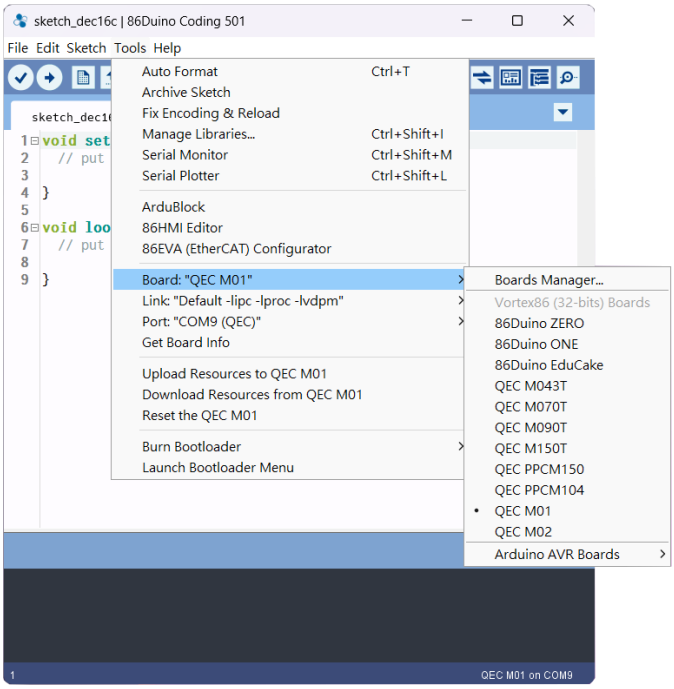
3. PC に接続して環境をセットアップする

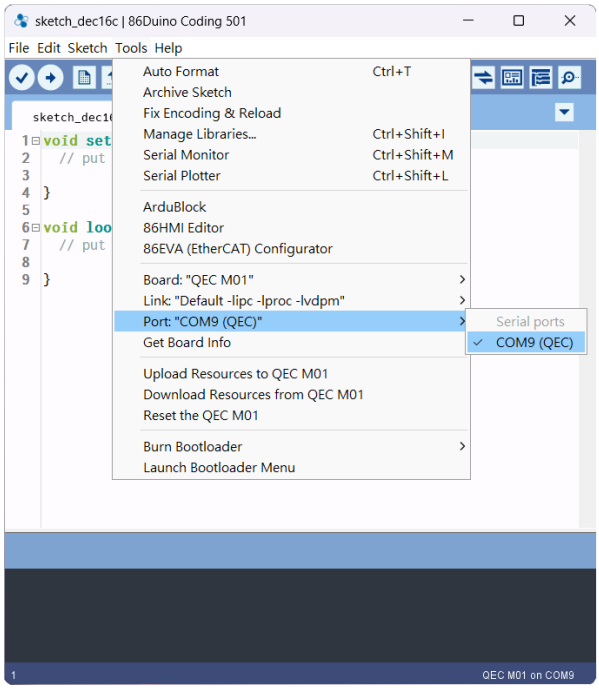
以下の手順に従って開発環境をセットアップします:

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. コードの記述

86Duino IDE の EtherCAT ライブラリを介して、EtherCATM デバイス(QEC-M-01)と NPM EC-AD1441A4 (2 相バイポーラ定電流、4 軸ステッピングモータードライバー)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 EC-AD1441A4 を CiA402 Profile Position (PP) モードに設定します。

- EtherCAT サイクルタイム: 1 ms
- EtherCAT モード : ECAT_SYNC

EthercatMaster の object (“master”) は QEC-M-01 を意味しますが、EthercatDevice_CiA402 の object (“motor[i]”) は NPM EC-AD1441A4 ステッピングモータードライバーの 4 軸を意味します。

A. Setup 関数:
setup() 関数では通信を初期化し、モーターを CiA402 プロファイル位置 (PP) モードに設定します。以下の手順に従ってください。:

1. シリアル通信初期化
    - ボーレート 115200 でシリアル通信を開始
2. EtherCAT M デバイスの開始
    - begin() 関数を使用して EtherCAT M デバイスを初期化し、EtherCAT ステートマシンを PRE-OPERATIONAL 状態に設定します。.
3. EtherCAT CiA402 モーターを EtherCAT M デバイスに接続する:
    - attach(0, i, master) 関数を使用して EtherCAT ネットワーク上のステッピングモータ軸に接続します。
4. DC(Distributed Clock)モードに設定
    - setDc() 関数を使用して EC-AD1441A4 Driver を DC モードに設定し EtherCAT 通信と同じサイクルタイムで同期。
5. EtherCAT M デバイスの実行
    - start() 関数を使用して EtherCAT ステートマシンを OPERATIONAL に切り替え、サイクルタイムを 1ms とし ECAT_SYNC モードにする。
6. PP(Set Profile Position)モードに設定
    - setCiA402Mode(CIA402_PP_MODE)関数を使用してモータを PP モードに設定
7. モータを有効化
    - enable() 関数を使用してモータを有効化し CIA402_OPERATION_ENABLED に移行
8. プロファイルパラメータの設定
    - 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：50,000、加速：3,000 減速：3,000

B. Loop 関数:

loop() 関数によりモータの現在置がシリアルモニタに表示され、モータを正転／反転させます。

1. 機器状態
    - 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ると、次の状態に移行します。
    - ケース 1: モータが目標位置に到達するまで待機します。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ると、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、機器状態をケース 0 にリセットして、移動サイクルを繰り返します。
2. pp_done を使用して完了したサイクルの数を追跡します。すべてのモータがサイクルを完了すると、すべての機器状態 (pp_state[]) と pp_done カウンターがリセットされます。

コードロジックの概要

- pp_Run()関数を使用してポジション移動を初期化する。
- pp_IsTargetReached()関数を使用して、目標位置に到達したかを確認します。
- 機器状態はケース 0 から始まり、ケース 3 を完了するとリセットされます。

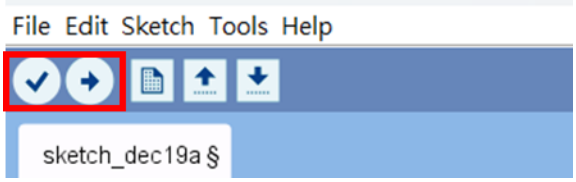
以下がコードの一例となります:

```
#include "Ethercat.h"

#define NUM 4
#define CYCLE 1000000

EthercatMaster master;
EthercatDevice_CiA402 motor[NUM];

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

  Serial.println(master.begin());

  for (int i = 0; i < NUM; i++) {
    Serial.println(motor[i].attach(0, i, master));
    Serial.println(motor[i].setDc(CYCLE));
  }

  Serial.println(master.start(CYCLE, ECAT_SYNC));
  delay(100);

  for (int i = 0; i < NUM; i++)
```

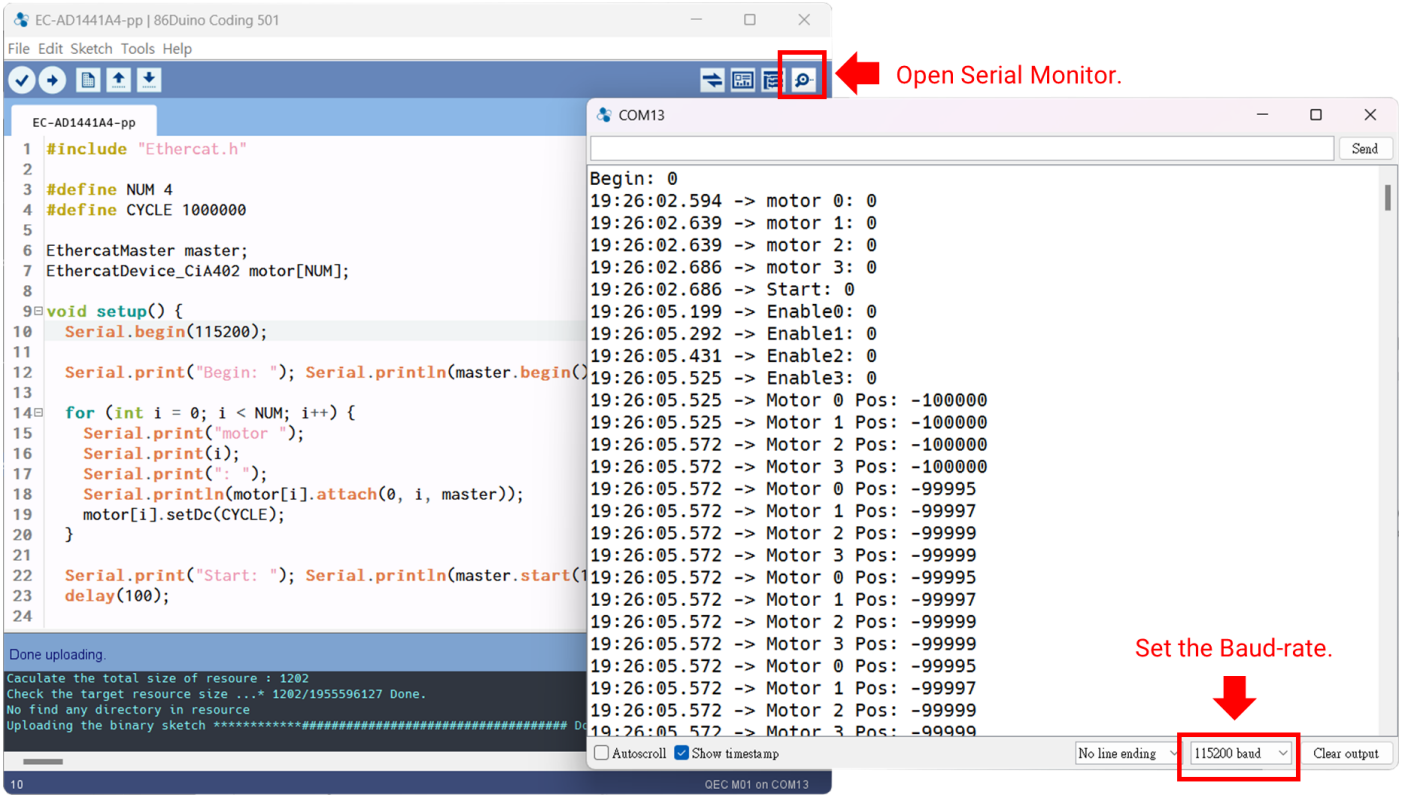
```
    Serial.println(motor[i].setCiA402Mode(CIA402_PP_MODE));


  for (int i = 0; i < NUM; i++) {
    delay(100);
    Serial.print("Enable");
    Serial.print(i);
    Serial.print(": ");
    Serial.println(motor[i].enable());
    motor[i].pp_SetMotionProfileType(0);
    motor[i].pp_SetVelocity(50000);
    motor[i].pp_SetAcceleration(3000);
    motor[i].pp_SetDeceleration(3000);
  }
}

int pp_state[NUM]; // State machine for PP
int pp_done = 0;
void loop() {

  for (int i = 0; i < NUM; i++) {
    Serial.print("Motor ");
    Serial.print(i);
    Serial.print(" Pos: ");
    Serial.println(motor[i].getPositionActualValue());

    switch (pp_state[i]) {
      case 0:
        if (motor[i].pp_Run(100000) == 0)
          pp_state[i]++;
        break;
      case 1:
        if (motor[i].pp_IsTargetReached())
          pp_state[i]++;
        break;
      case 2:
        if (motor[i].pp_Run(-100000) == 0)
          pp_state[i]++;
        break;
      case 3:
        if (motor[i].pp_IsTargetReached())
          pp_state[i] = 0;
```

```
            pp_done++;
            break;
        }
    }

    // Once all motors are done, reset states
    if (pp_done == NUM) {
        pp_done = 0;
        for (int i = 0; i < NUM; i++)
            pp_state[i] = 0;
    }
}
```

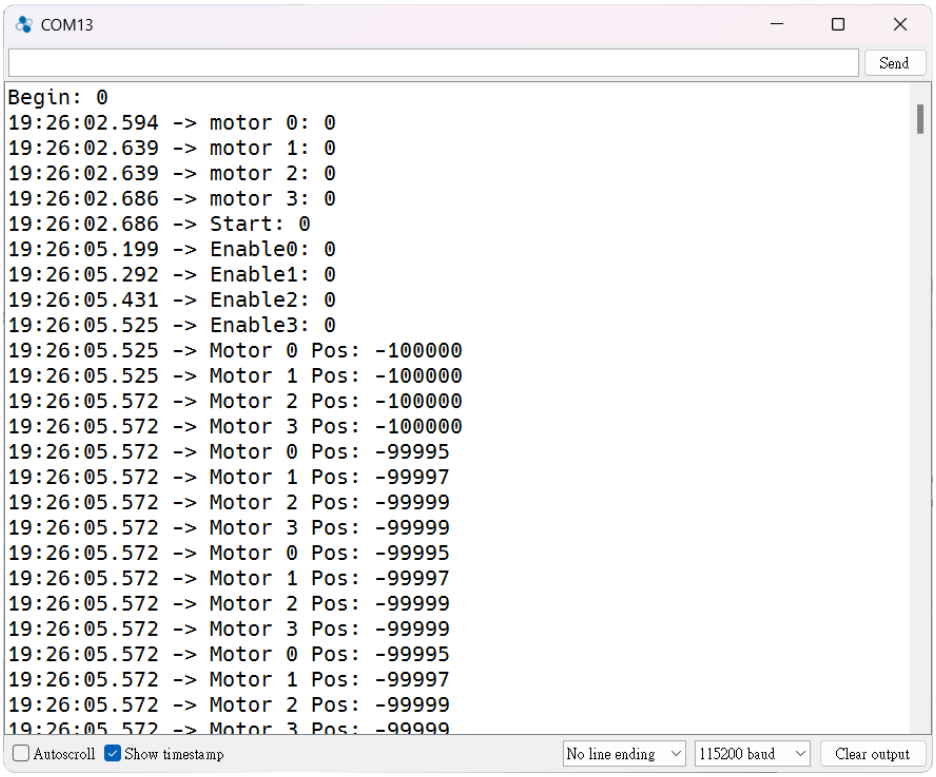
注: コードを記述したならば、ツールバーの をクリックしてコンパイルします。コンパイルが完了しエラーがないことを確認したら クリックしてプログラムをアップロードします。



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 “motor i: 0”, “Start: 0”, そして “Enable i: 0”を表示します。そしてモータの現在位置をシリアルモニタに表示します。



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：

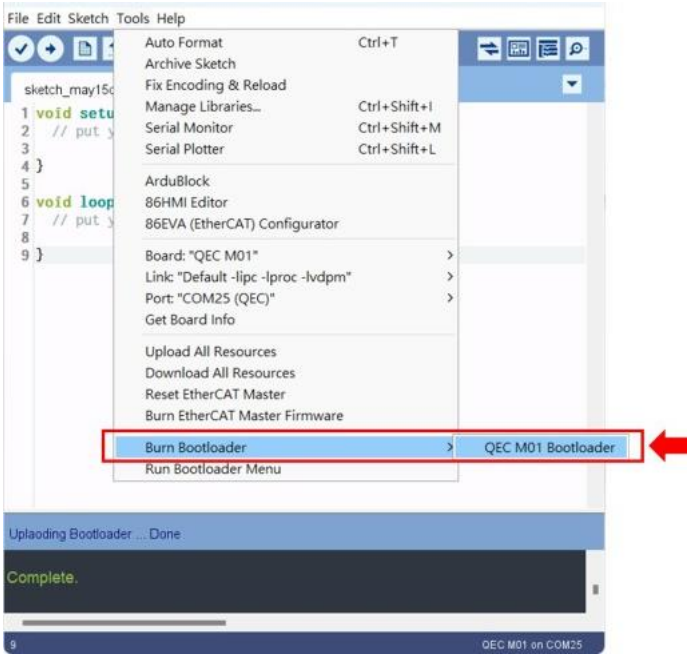
ステップ 1: QEC-M のセットアップ

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。

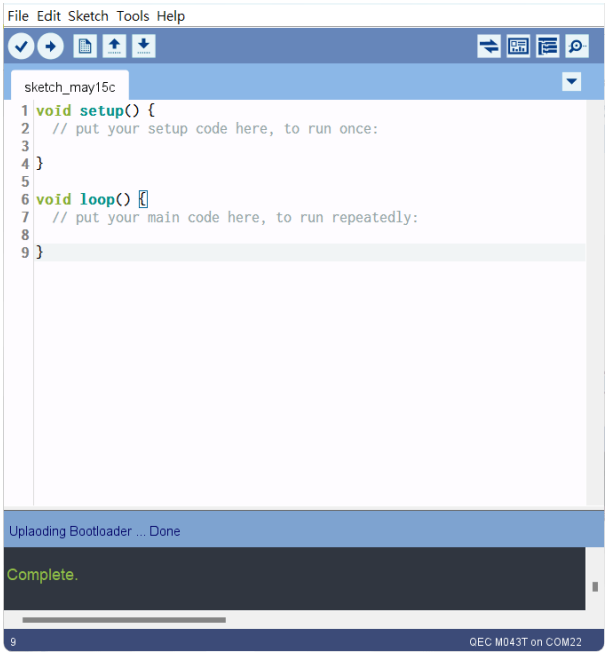
ステップ 2: 「Burn Bootloader」ボタンをクリック

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。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.