

Start Guide

SANMOTION G - EtherCAT AC サーボ CiA402 CSP モード (1 軸)



86Duino Coding IDE 501

EtherCAT ライブラリ

(Version 1.0)

改訂履歴

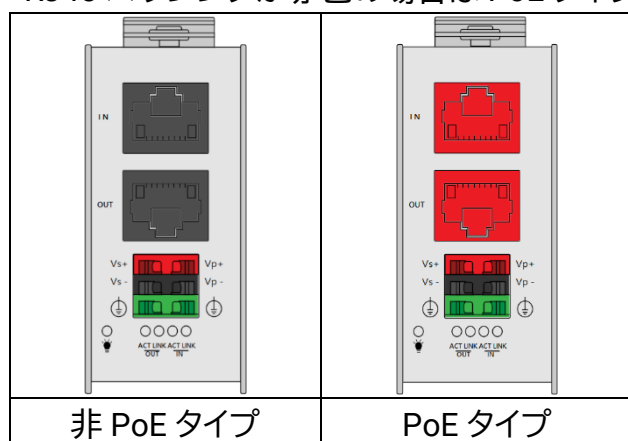
記述日	バージョン	備考
2025/9/12	Version1.0	New Release.

序文

本ガイドでは、QEC-M-01 (EtherCAT M デバイス) と山洋電機株式会社製 SANMORION G Series **GADSA01AH24**(AC サーボアンプ, EtherCAT ドライバ)の CiA402 Profile Position(CSP)モードによる動作での使用方法を説明します。

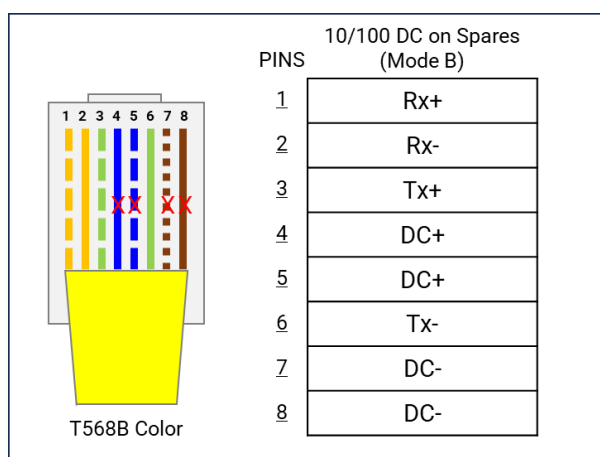
注意 QEC 機器の PoE (Power over Ethernet)について

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。



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 EtherCATM デバイスを他社の EtherCATSub 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. SANYO DENKI SANMOTION G シリーズ GADSA01AH24 (AC Servo アンプ, EtherCAT ドライバ)
3. 220 AC 電源 & 24VDC 電源 & ヨーロッパタイプ端子ケーブル & LAN ケーブル
4. GAM2A4005 (AC サーボモータ, フランジサイズ 40 mm 角)



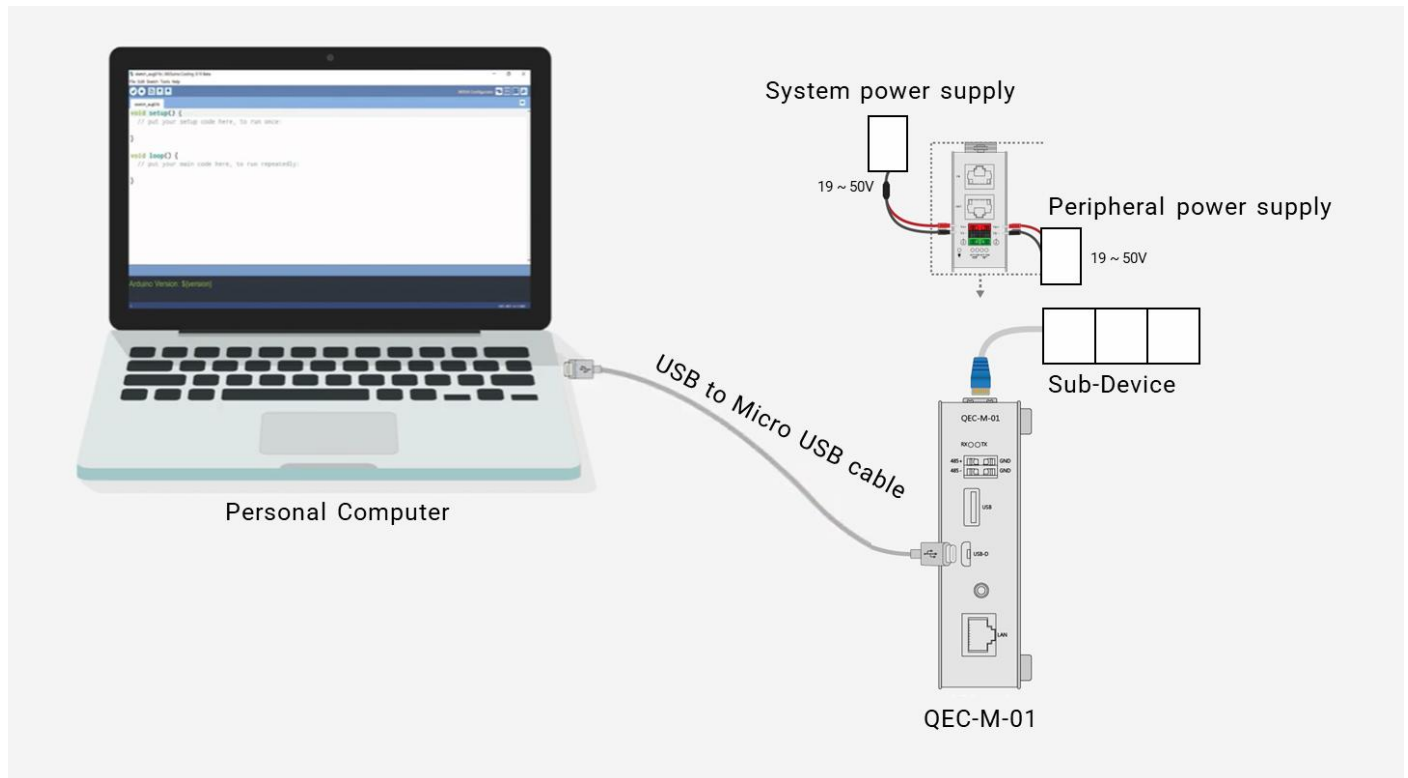
1.1 QEC-M-01

QEC EtherCAT M デバイス

1. 電源:

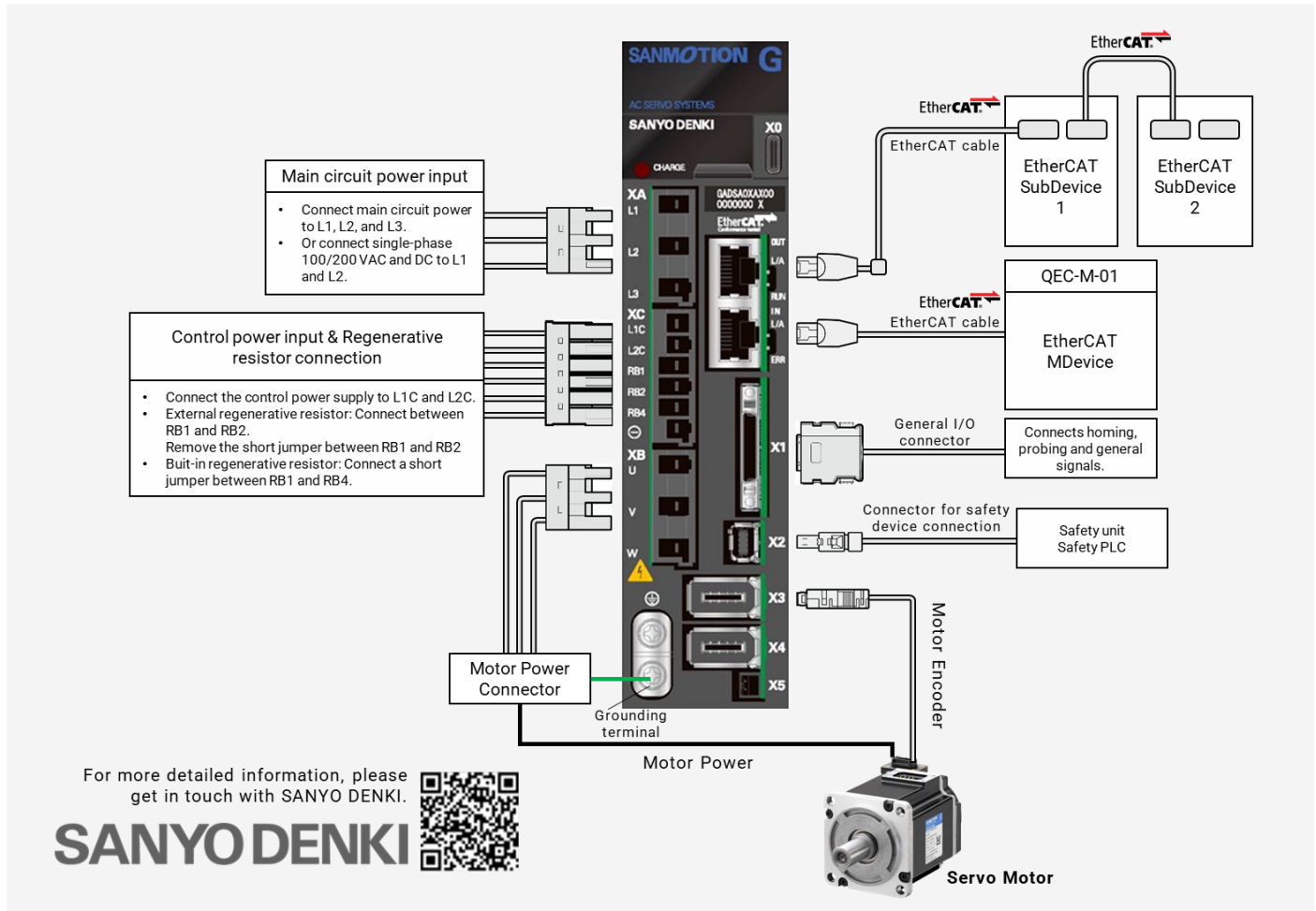
24V 電源を QEC EtherCAT M デバイスのヨーロッパタイプ端子 V_{s+}/V_{s-} および V_{p+}/V_{p-} に接続

2. EtherCAT 通信: EtherCAT 出力ポート (上側) から EtherCAT Sub デバイスの EtherCAT 入力ポートに RJ45 ケーブルで接続



1.2 GADSA01AH24

SANMOTION G シリーズ GADSA01AH24 は、CiA402 モーションプロファイル準拠の EtherCAT インターフェースを内蔵した単軸 AC サーボアンプです。この図は、GAM2A4005F0XRK0 モーターを接続した例を示しています。



接続は機能別にグループ化されています:

1. EtherCAT 接続

- QEC-M01 → EtherCAT ケーブルでドライバ EtherCAT IN に接続します。
- ドライバ EtherCAT OUT から他の EtherCAT サブデバイスへのデージーチェーン接続 (オプション)。
- CiA402 コマンドを発行する前に、ネットワークを OP にします。

2. 主回路電源(モータ出力段)

- AC 電源を L1/L2/L3 に接続します。単相の場合は L1/L2 (AC 200～240 V) を使用します。
- シャーシを PE 端子で接地します。
- 単相での注意** 位相欠相アラームを回避するには、System1 ID01 = 01 (AC 単相) を設定します (AL.63.0)

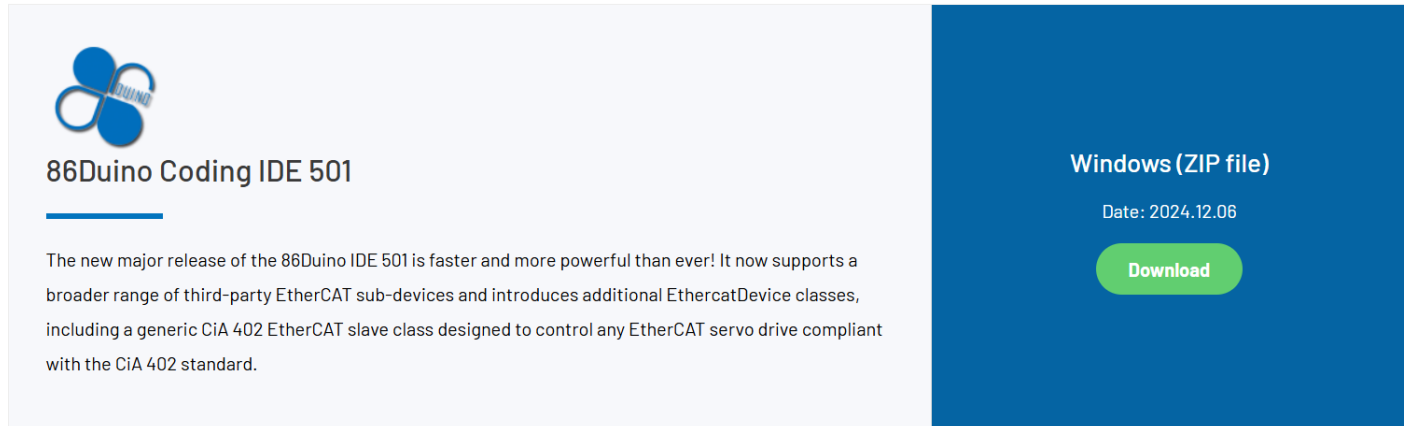
3. 制御電源 & 回生抵抗

- L1C/L2C で制御回路に電力を供給します (モデルごとに AC 200～240 V または DC 24 V)。
- 外付け回生抵抗器: RB1 と RB2 を接続します。

- 内蔵抵抗器(該当する場合):マニュアルに従って RB1 ~ RB4 を短絡します。
4. モーター & エンコーダー
- U/V/W → サーボモーター電源コネクタ。モーターPE をドライブ PE に接続したままにします。
 - X3/X4 → モーターエンコーダー(ES+/ES-、5V/SG、シールド)。
 - X5 (オプション) → アブソリュートエンコーダ用のエンコーダバッテリー。
5. 安全性(STO) & 一般的な I/O
- X2 → STO 用のセーフティ PLC/セーフティリレー
(両方のチャンネルを有効にする必要があり、STO はソフトウェアによって解除できません)。

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。



The image shows the 86Duino Coding IDE 501 download page. On the left, there is a logo and the title "86Duino Coding IDE 501". Below it, a paragraph describes the new major release, stating it is faster and more powerful, supporting a broader range of third-party EtherCAT sub-devices and introducing additional EthercatDevice classes, including a generic CiA 402 EtherCAT slave class. On the right, there is a blue sidebar with the text "Windows (ZIP file)", the date "Date: 2024.12.06", and a green "Download" button.

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。

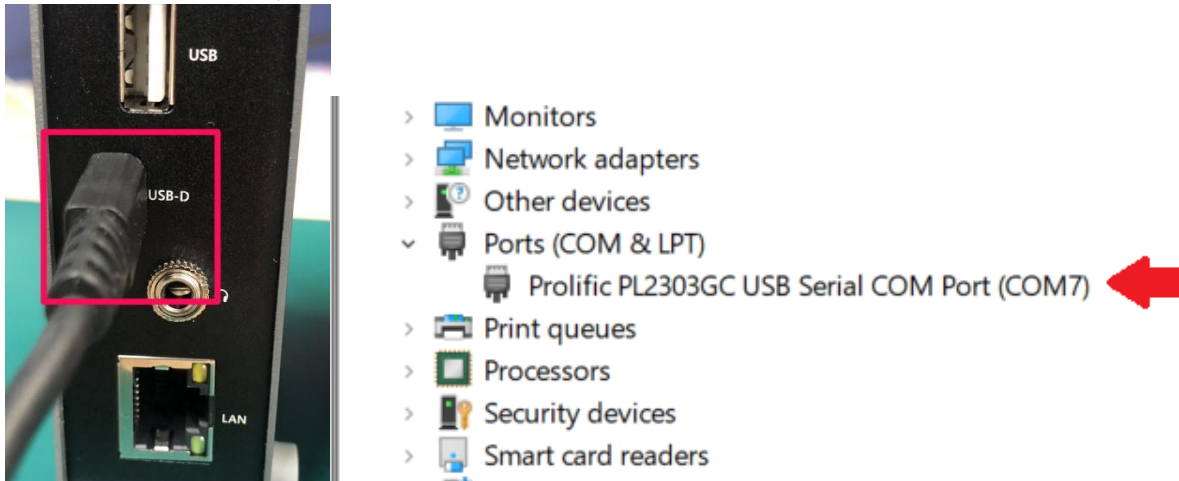
86Duino コーディング IDE 500+ は下図です。



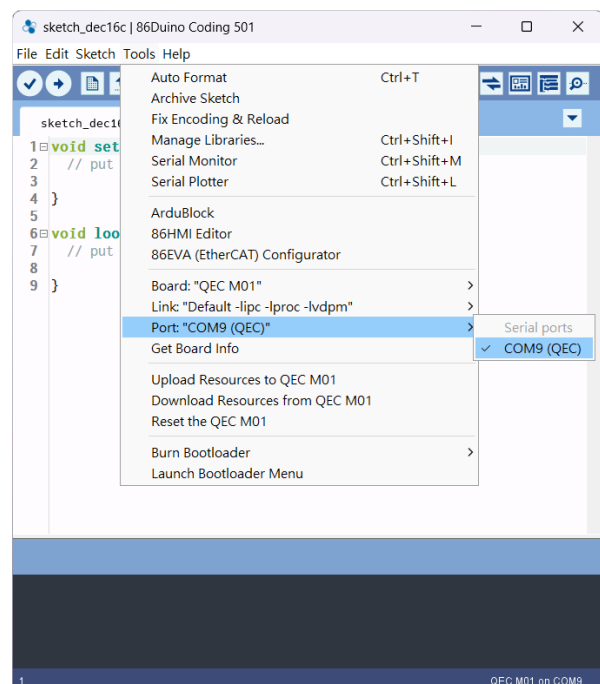
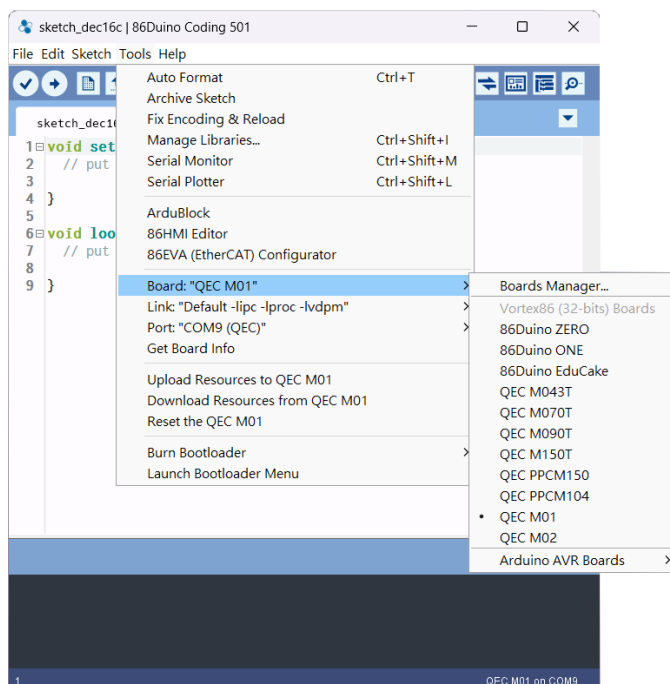
3. PC に接続して環境をセットアップする

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 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 M デバイス型名) 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-MM 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、SANMOTION G Driver (GADSA01AH24)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 GADSA01AH24 を CiA402 Cyclic-Synchronous Position モードに設定します。

- EtherCAT サイクルタイム: 1ms Cycle Time:
- EtherCAT モード: ECAT_SYNC
- モーションは、コマンド位置を固定レートで (等速) ステップ処理することで、1ms の周期的な callback 内で生成されます。

EthercatMaster のオブジェクト ("master")は QEC-M-01 を意味し、EthercatDevice_CiA402 オブジェクトの("motor")は GADSA01AH24 ドライバを意味します。

A. Set up()関数内の説明:

`setup()` 関数では、通信を初期化し、モーターを CiA402 CSP モードに設定します。下記の手順に従ってください。

1. Serial 通信の初期化
 - Serial 通信をボーレート 115200 でスタート
2. EtherCAT M デバイスのスタート
 - `begin()` 関数を使用して EtherCAT M デバイスをスタート、EtherCAT ステートマシンを PRE-OPERATIONAL 状態に切り替える
3. GADSA01AH24 EtherCAT Sub デバイスの接続
 - `attach()`関数を使用して、EtherCAT Sub デバイスを EtherCAT ネットワークに接続
ノード番号と特定の M デバイスを設定する
4. Distributed Clocks を設定する(cycle = 1 ms)
 - `setDc(1000000)` 1ms の制御周期に合わせる
5. CiA402 モードを Cyclic-Synchronous Position (CSP)モードに設定する
 - `setCiA402Mode(CIA402_CSP_MODE)`を使用してモーターを CSP モードに設定する
6. CSP モードに必要な最小限の PDO を GADSA01AH24 にマッピングする
(デフォルトでは PDO マッピングはありません。ユーザが入力する必要があります)
 - **RxPDO:**
`sdoDownload8(0x1C12, 0x00, 0)` と `sdoDownload8(0x1601, 0x00, 0)` を使用して RxPDO index を空にする。次に `sdoDownload32()`を使用して Controlword (0x6040) と Target position (0x607A) を Index 0x1601 マッピングし、そして `sdoDownload8()` を使用して Index 0x1601 を Index 0x1C12 にマッピングする。
 - **TxPDO:**
`sdoDownload8(0x1C13, 0x00, 0)` と `sdoDownload8(0x1A01, 0x00, 0)`を使用して

TxPDO index を空にする。次に `sdoDownload32()`を使用して Statusword (0x6041) と Position actual value (0x6064) を Index 0x1A01 マッピングする, 次に `sdoDownload8()`を使用して Index 0x1A01 を Index 0x1C13 へマッピングする。

7. cyclic callback function の登録(これは `start()`の前に呼び出す)

- `attachCyclicCallback(MyCyclicCallback)` function を使用して cyclic EtherCAT ハンドラに登録する。これは毎サイクル (例: 1 ミリ秒) ごとにリアルタイムで実行され、PDO のみである必要があります。SDO アクセス、ログ記録/Serial.print、遅延、呼び出しのブロック、動的割り当て、または callback 内の大量の計算を避けてください。

8. EtherCAT M デバイスのスタート

- Use the `start(1000000, ECAT_SYNC)`関数を使用して EtherCAT state machine を OPERATIONAL state に切り替えます。サイクルタイムを 1ms、ECAT_SYNC モードに設定します。

9. コマンド位置を初期化し、モーターを有効化する

- 安全のため、最初の CSP ターゲットを実際の位置に設定する
`motor.setTargetPosition(position = motor.getPositionActualValue());`
- `enable()`関数を使用してモーターを有効化し、CiA402_OPERATION_ENABLED 状態に移行させます。

B. Loop 関数内の説明:

`loop()` モニタリング用の実際の位置を出力します。すべての動作ロジックは周期的な callback で実行されます。

C. Callback 関数内の説明

The `MyCyclicCallback()` 関数は等速運動(constant-velocity motion)を実現するため、毎サイクル (1ms)ごとに新しい CSP ターゲットを発行し、毎サイクル 10000 単位ずつ増加させます。

次の手順に従って:

1. 状態の確認 CiA402 の状態が CiA402_OPERATION_ENABLED であることを確認
 - もし `getCiA402State() != CiA402_OPERATION_ENABLED` の場合はなにもしないに戻る
2. CSP 位置コマンドの生成
 - ソフトウェア位置アキュムレータをインクリメント(例: `position += 10000`)し、各サイクルで `setTargetPosition(position)`を呼び出す。

これにより一定の位置ランプ(ramp)が生成される。1ms 周期の場合、+10,000 カウント/周期 ≈ 10 M counts/s となる。－ ドライブ/モーターの限界に合わせて調整すること。

プログラム例:

```
#include "Ethercat.h"

EthercatMaster master;
EthercatDevice_CiA402 motor;
```

```

int32_t position = 0;

void MyCyclicCallback() {
    if (motor.getCiA402State() != CIA402_OPERATION_ENABLED)
        return;
    motor.setTargetPosition(position += 10000);
}

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

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

    Serial.print("Slave: ");
    Serial.println(motor.attach(0, master));
    motor.setDc(1000000);
    motor.setCiA402Mode(CIA402_CSP_MODE);

    // No PDO mapping by default, needs to be filled out by users.
    // RxPDO mapping configuration.
    motor.sdoDownload8(0x1C12, 0x00, 0);
    motor.sdoDownload8(0x1601, 0x00, 0);
    motor.sdoDownload32(0x1601, 0x01, 0x60400010);
    motor.sdoDownload32(0x1601, 0x02, 0x607A0020);
    motor.sdoDownload8(0x1601, 0x00, 2);
    motor.sdoDownload16(0x1C12, 0x01, 0x1601);
    motor.sdoDownload8(0x1C12, 0x00, 1);
    // TxPDO mapping configuration.
    motor.sdoDownload8(0x1C13, 0x00, 0);
    motor.sdoDownload8(0x1A01, 0x00, 0);
    motor.sdoDownload32(0x1A01, 0x01, 0x60410010);
    motor.sdoDownload32(0x1A01, 0x02, 0x60640020);
    motor.sdoDownload8(0x1A01, 0x00, 2);
    motor.sdoDownload16(0x1C13, 0x01, 0x1A01);
    motor.sdoDownload8(0x1C13, 0x00, 1);

    master.attachCyclicCallback(MyCyclicCallback);
    Serial.print("Start: ");
    Serial.println(master.start(1000000, ECAT_SYNC));
}

```

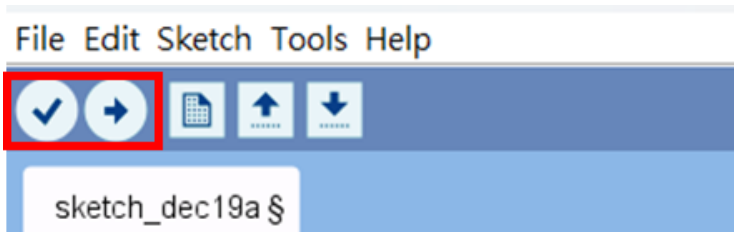
```

motor.setTargetPosition(position = motor.getPositionActualValue());
Serial.print("Enable: ");
Serial.println(motor.enable());
}

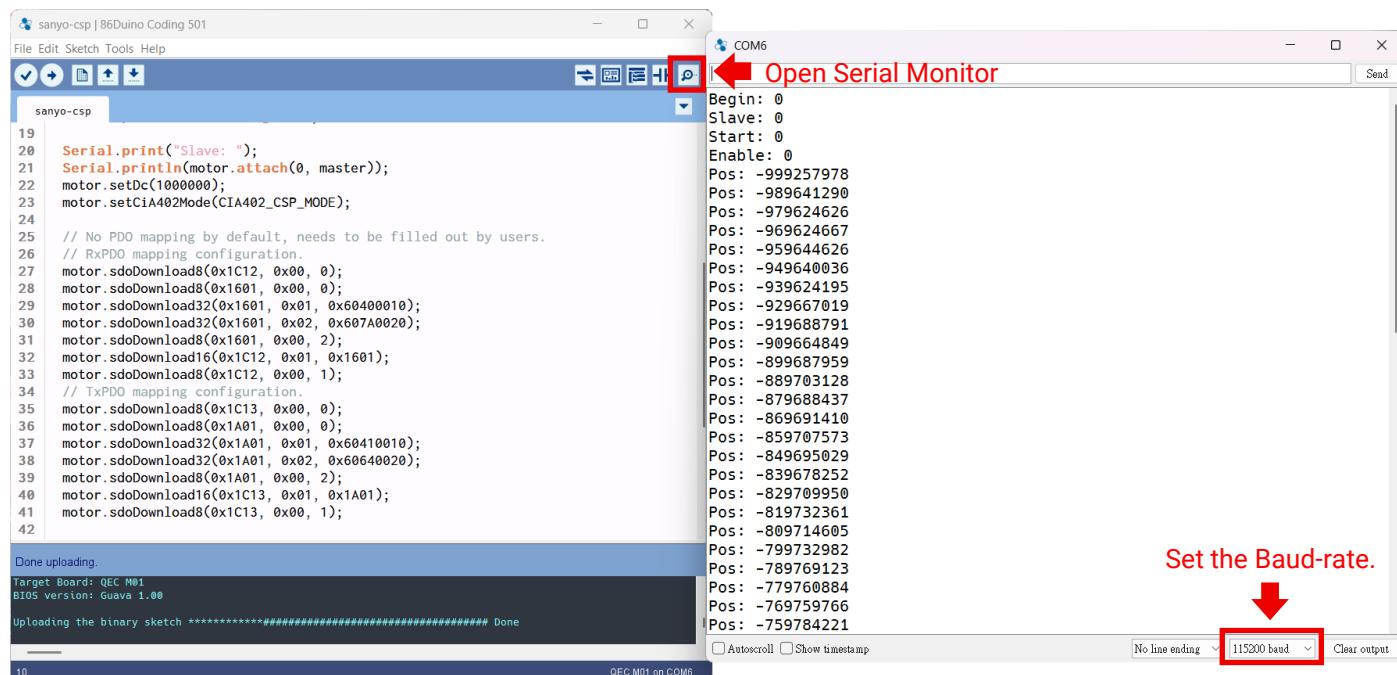
void loop() {
  Serial.print("Pos: ");
  Serial.println(motor.getPositionActualValue());
  delay(1000);
}

```

***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  をクリックしてプログラムをアップロードします。アップロードが完了するとプログラムが実行されます。



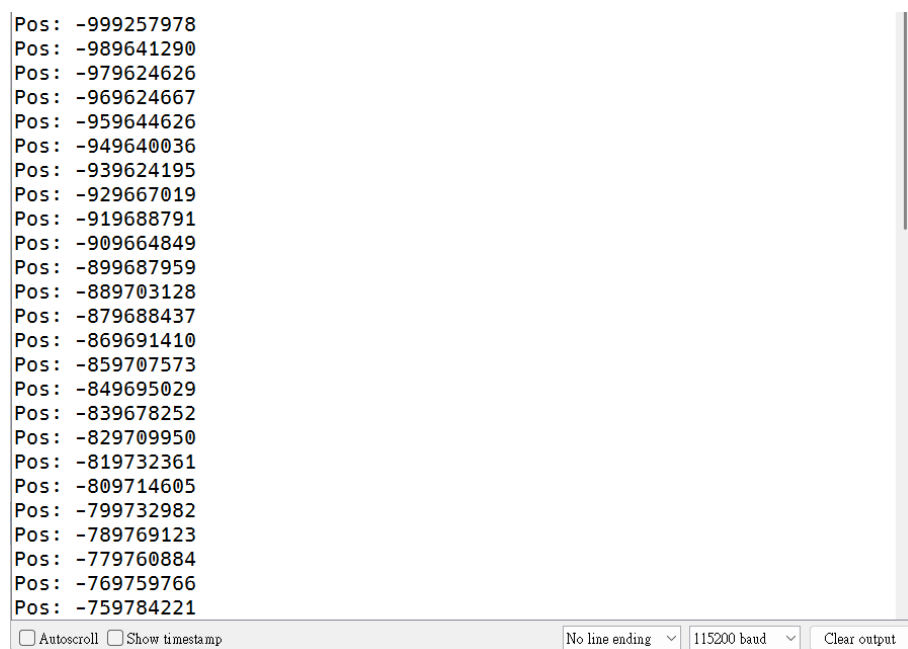
QEC-M-01 にプログラムをアップロードしたら、86Duino IDE でシリアルモニタを開きます。シリアルボーレートがあなたの設定と同じであることを確認してください。



EtherCAT 通信の設定が成功すると、シリアルモニターには EtherCAT の各ステータスに対して「0」、CiA402 に対して「Enable: 0」と表示されます。



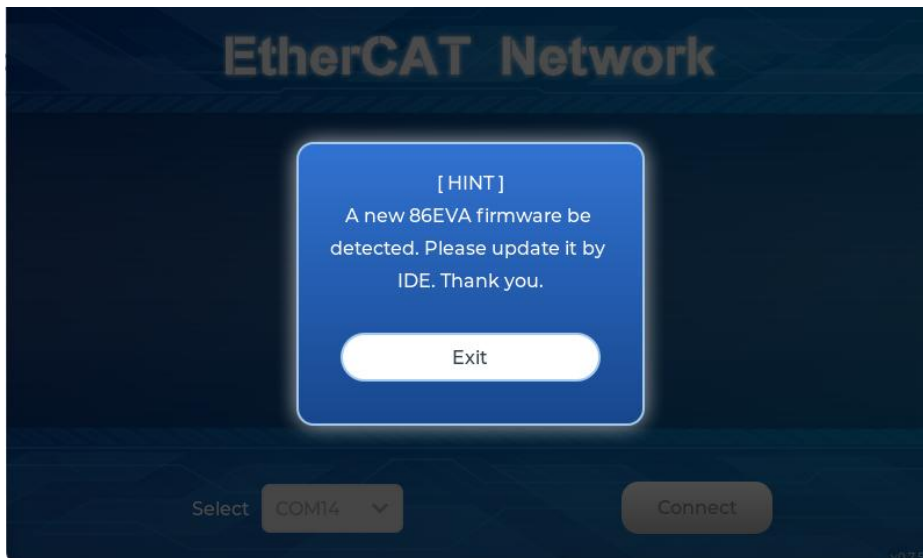
モーターの現在位置をシリアルモニターに出力しています。



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 EtherCATM 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 EtherCATM 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：

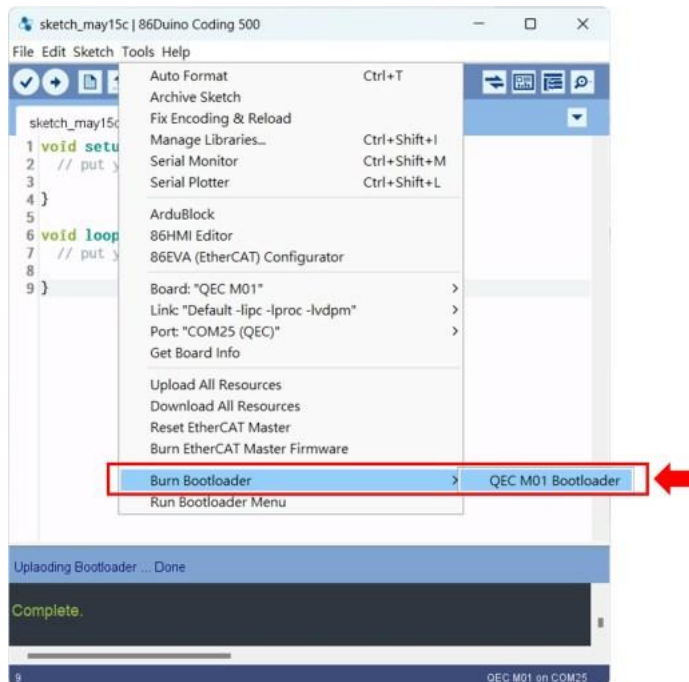
ステップ 1: QEC-M のセットアップ

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。

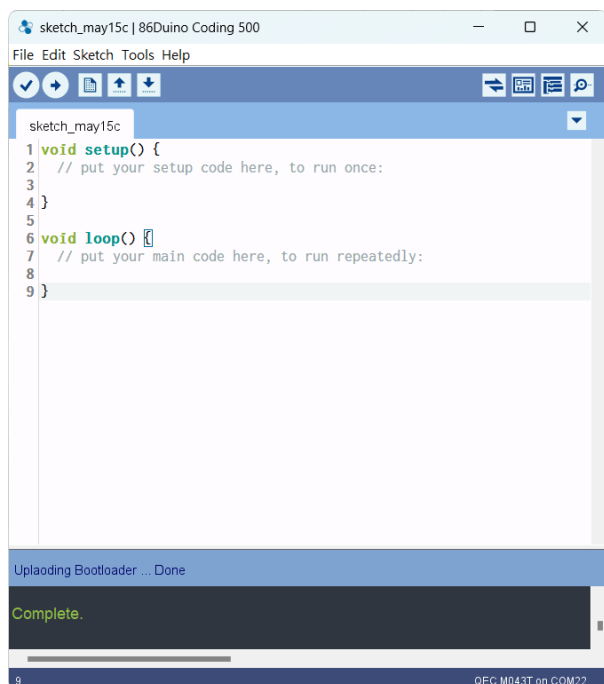
ステップ 2: 「Burn Bootloader」ボタンをクリック

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.

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. 2025

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。