

スタートガイド

オリエンタルモーター AZ-mini

EtherCAT スレーブ (PP モード)

86EVA 版



86Duino Coding IDE 501

EtherCAT Library

(Version 2.0)

改訂履歴

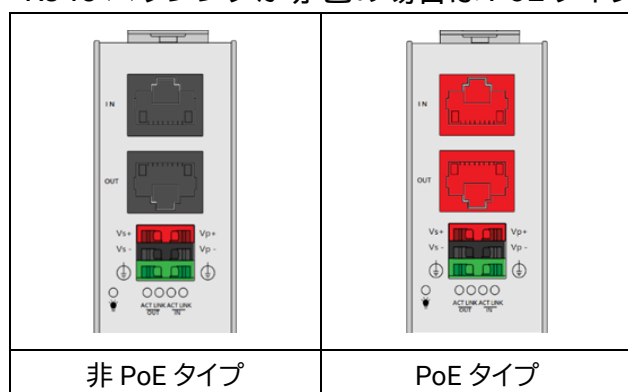
記述日	バージョン	備考
2024/9/11	Version1.0	New Release
2025/4/7	Version2.0	・86Duino IDE501 運用に訂正 ・マスタ表記を M デバイスに、スレーブ表記を Sub デバイスに変更

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 AZ-mini EtherCAT Sub デバイスの CiA402 Profile Position(PP)モードによる動作での使用方法を説明します。

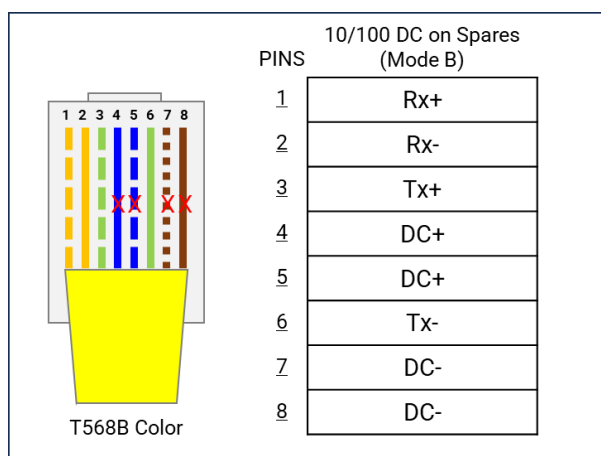
注意 QEC 機器の PoE (Power over Ethernet)について

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。



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. AZD-KRED, AZ シリーズ mini EtherCAT ドライバ
(オリエンタルモーター ステップ・サーボ・ドライバ)
3. AZM48AK, 1.65 in. (42 mm) AZ シリーズ ステッピング・モータ,
アブソリュート形 エンコーダ(DC 入力)
4. 24V 電源入力



1.1 QEC-M-01

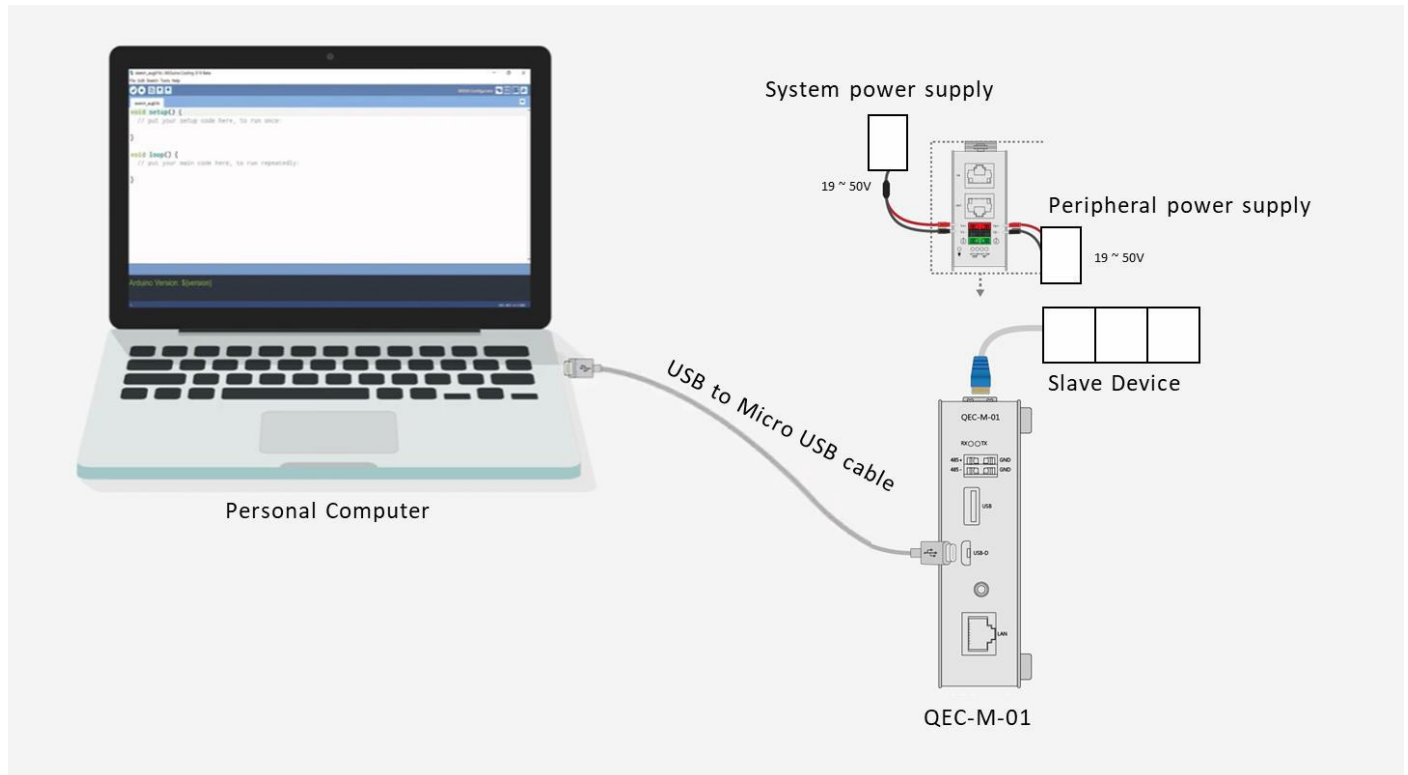
QEC EtherCAT M デバイス

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

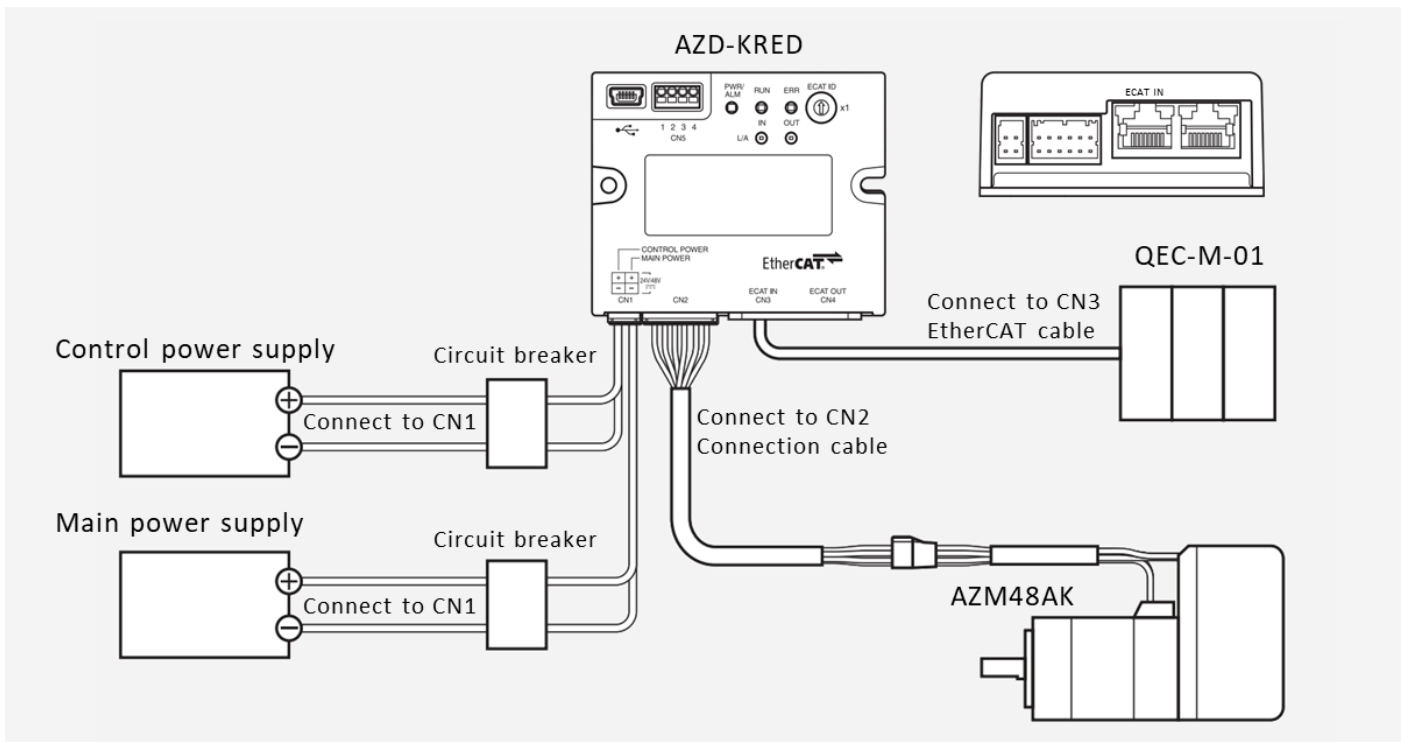
2. EtherCAT 通信:

EtherCAT 出力ポート (上側) から AZD-KRED の EtherCAT 入力ポートに RJ45 ケーブルで接続



1.2 AZD-KRED

AZD-KRED, AZ シリーズ mini EtherCAT ドライバ(オリエンタルモーター ステップ・サーボ・ドライバ)
下図は AZM48AK モータ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可能性があります。
- 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:

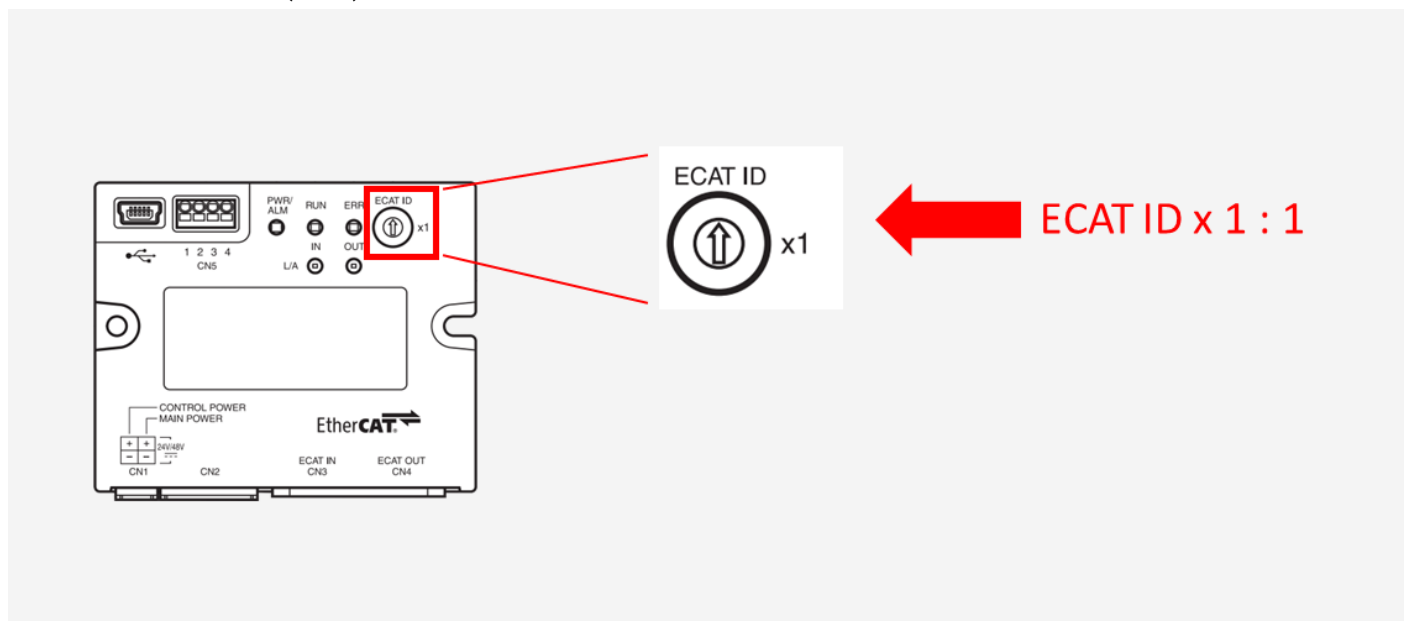
- コネクタの着脱を行う前に、主電源と制御電源を切り、PWR/ALM LED が消灯していることを確認してください。
- コネクタを外すときは、コネクタのラッチを指で押しながら引き抜いてください。

AZ-mini ドライバのノードアドレス設定:

ノードアドレスは ECAT ID x1 スイッチで設定します。このスイッチでは、16 進数で 0 から 15 までのノードアドレスを設定できます。

- ノードアドレス・スイッチは 16 段階に設定可能で、アドレス 0 から 15 は、16 進数の 0x0 から 0xF に対応します。
- 複数のデバイスを EtherCAT ネットワークに接続する場合は、競合を避けるために 2 つのデバイスが同じノードアドレスを持っていないことを確認してください。

出荷時初期設定: 0(x1:0)

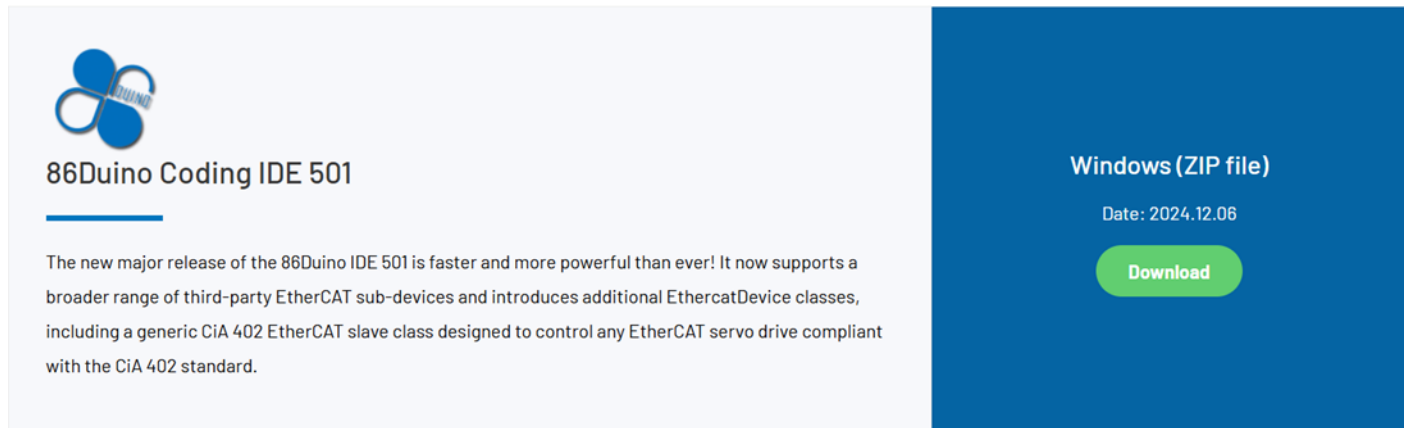


設定範囲	定義
0 (00h)	M デバイスの設定が適用される
1-15 (01h-0Fh)	ドライバの設定が適用される

注: スイッチの設定を調整する際は、主電源と制御電源の両方がオフになっていることを確認してください。電源が入ったままスイッチを調整しても設定は反映されません。

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。



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。

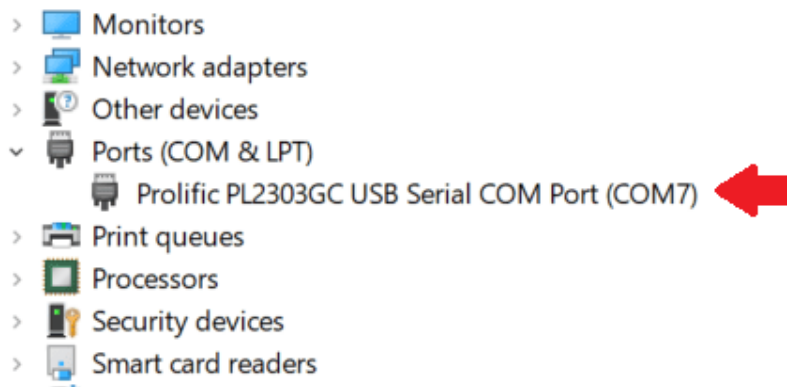
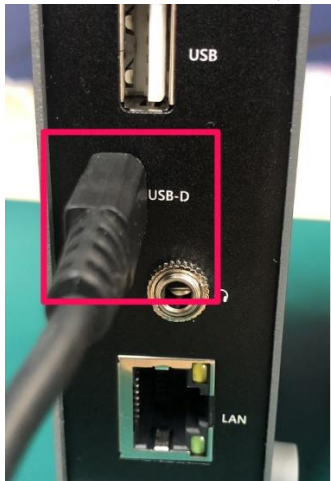
86Duino コーディング IDE 501+ は下図です。



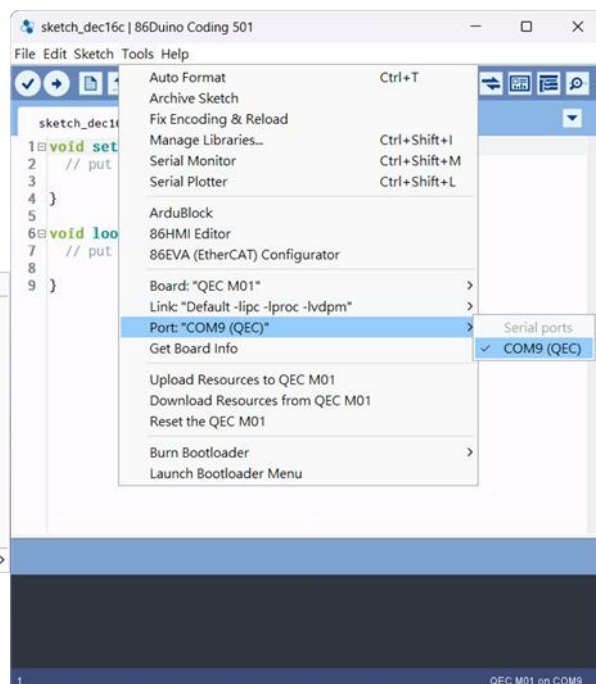
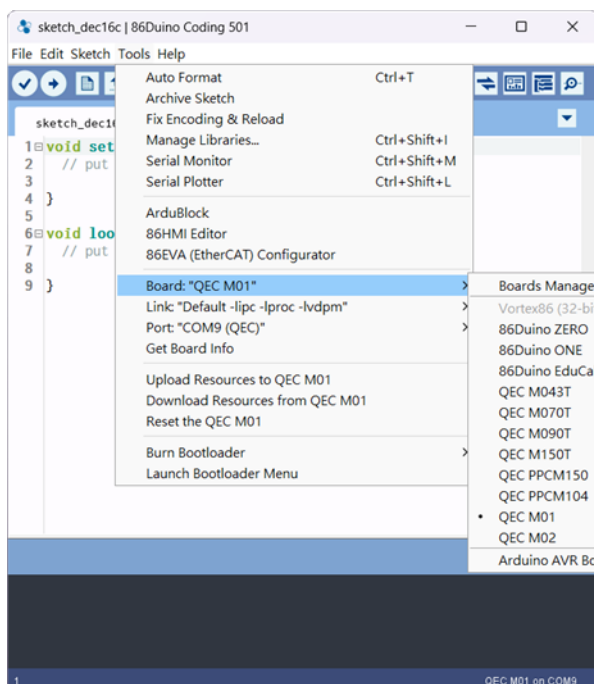
3. PC に接続して環境をセットアップする

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. 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 マスタ(QEC-M-01)とオリエンタルモーター AZ-mini EtherCAT ドライバを操作する方法を示します。

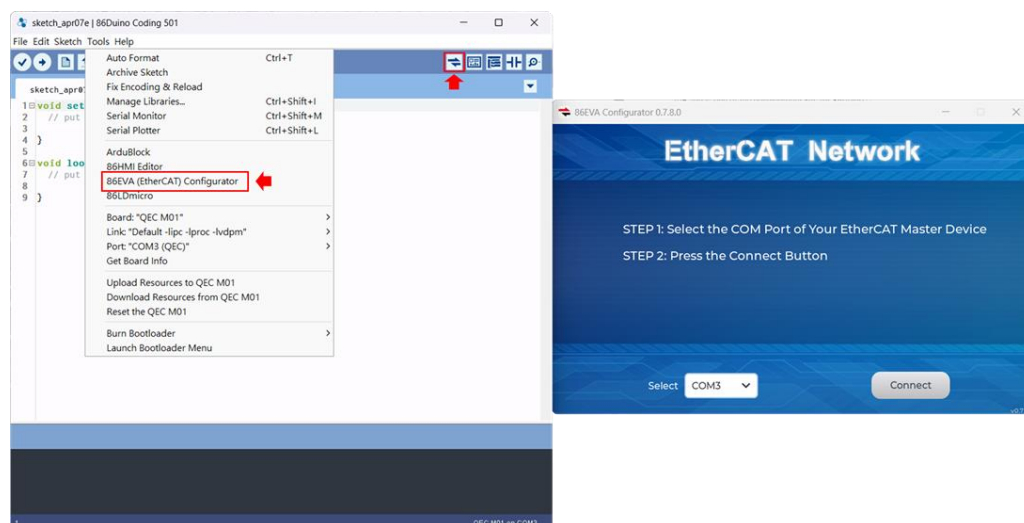
ソフトウェア・ツールの説明:

- 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。

このプログラムは、AZ-mini ドライバーを CiA402 プロファイルポジション (PP) モードに設定します。AZ-mini モーターの位置を読み取り、動作状態に基づいて目標位置を更新してモーターを制御します。

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

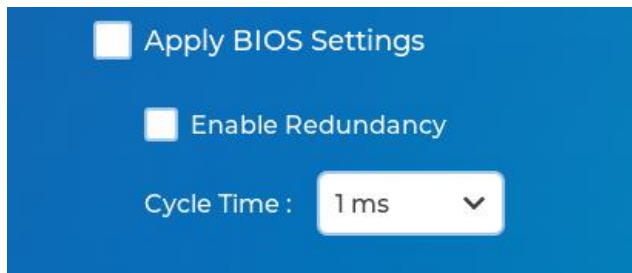
QEC-M-01 の画像をダブルクリックすると、パラメータ設定が表示されます。



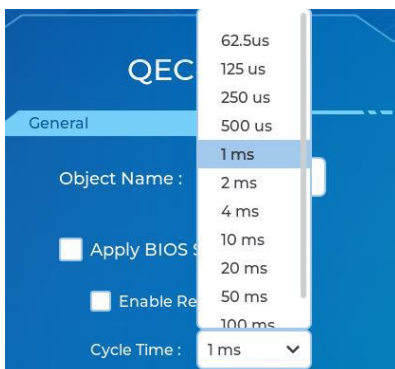
ここでは、Object Name, BIOS Settings, Cycle Time, DC mode, ENI file および仮想 Arduino 構成が表示されます。

以下の設定を確認してください。

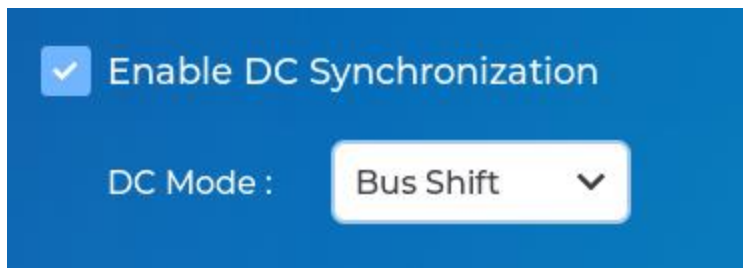
1. 「Apply BIOS Settings」をオフ



2. 「Cycle Time」で “1ms” を選択



3. 「Enable DC Synchronization」をオン

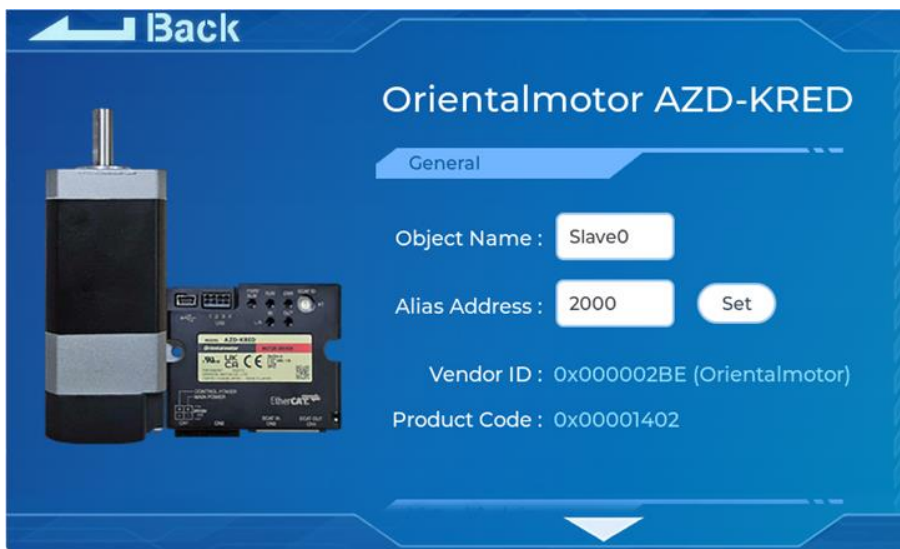


左上の「Back」をクリックして戻ります。



AZD-KRED

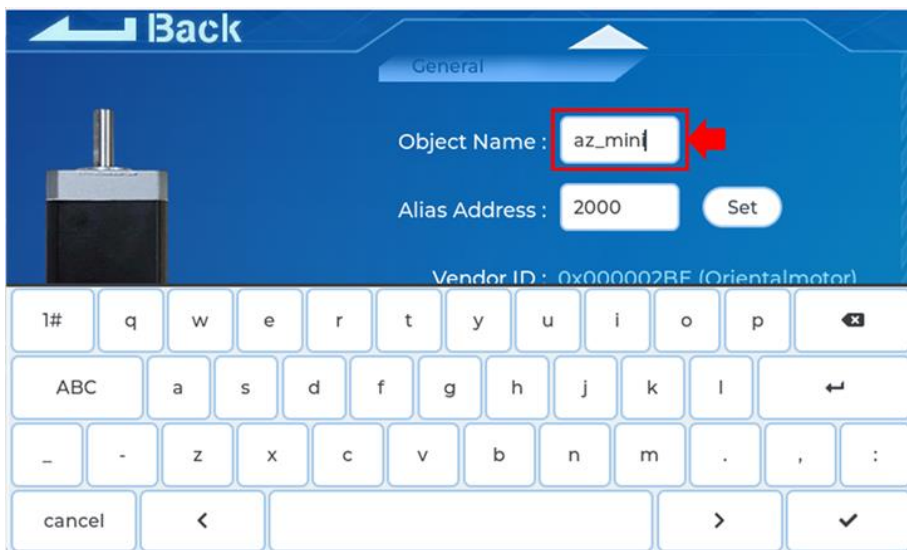
AZD-KRED の画像をダブルクリックして、AZD-KRED のパラメータ設定を確認します。



このページでは Object Name, Alias Address, Vendor ID, Product Code が表示されます。

「Object Name」を "az-mini"に変更します。

Object Name をクリックすると、キーボードが表示されます。

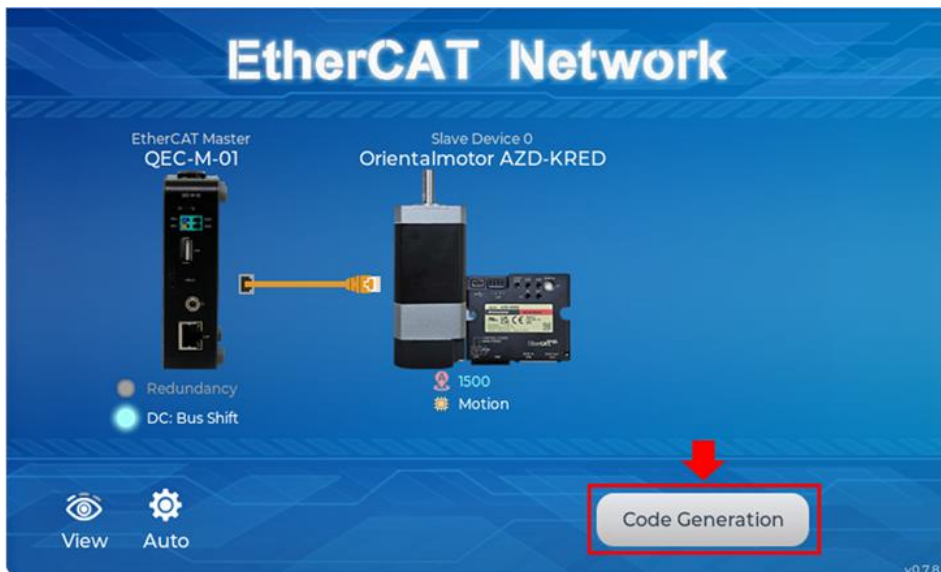


左上の「Back」をクリックして戻ります。



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。

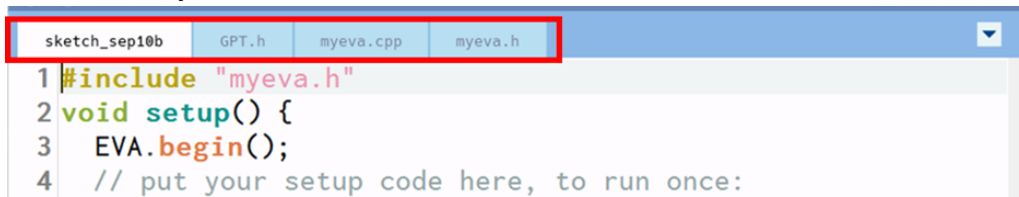


完了後「OK」ボタンをダブルクリックして 86EVA を閉じます。そのままにしておいても自動的に 10 秒後に終了します。



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)
- GPT.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



The screenshot shows an IDE interface. At the top, there is a file explorer bar with four tabs: 'sketch_sep10b', 'GPT.h', 'myeva.cpp', and 'myeva.h'. The 'sketch_sep10b' tab is selected and highlighted with a red rectangle. Below the tabs, the code editor displays the contents of the selected file, which is the main sketch (.ino) file. The code is as follows:

```
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。

- オリエンタルモーター AZ-KRED: EthercatDevice_CiA402 object
- CiA402 モード: Profile Position (PP) モード
- EtherCAT モード: ECAT_SYNC

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"、AZD-KRED は"az_mini"
- Distributed Clock(高精度時刻同期機能):Open, サイクルタイムに従う

1. Setup()関数内の説明:

- Serial の初期化 (115200)
- EVA.begin()関数に "EthercatCallback"という名前の Cyclic Callback 関数を登録する
- AZ-mini 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を設定する。モーションプロファイルタイプ:Linear Ramp, プロファイル速度:100,000 加速度:100,000 減速:100,000
delay(1000)を使用して、正常に変更されるまで待つ

2. Callback()関数内の説明:

モータの実際の位置の値を周期的に読み取る。

3. Loop()関数の内説明:

main loop は、モータの現在位置をシリアルモニタに表示します。そして Profile Position(PP)モードのステートマシンを管理します。モータは特定の位置に移動し、目標に到達するのを待ち、それから開始位置に反転して戻ります:

- ステート 0:目標位置(100,000 単位)へ移動する。
- ステート 1:モータが目標位置に到達するのを待ち、到達したら次の状態に遷移する。
- ステート 2:別の目標位置(-100,000 単位)へ移動する。
- ステート 3:モータが目標位置に到達するのを待ち、到達したらステートマシンをリセットしてステート 0 に戻し、このサイクルを繰り返す
各ステートは、スムーズな遷移を確実にするために 100ms の遅延をいれます。

要約すると、このコードは EtherCAT 通信を確立し、AZ-mini モータの位置を読み取り、動作状態に基づいて目標位置を更新することによりモータを制御します。

プログラム例:

```

#include "myeva.h"

int pp_state = 0, pos = 0;

void EthercatCallback(void) {
    pos = az_mini.driveGetPositionActualValue();
}

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

    Serial.print("Enable: "); Serial.println(az_mini.enable());
    az_mini.pp_SetMotionProfileType(0);
    az_mini.pp_SetVelocity(100000);
    az_mini.pp_SetAcceleration(100000);
    az_mini.pp_SetDeceleration(100000);
    delay(1000);
}

void loop() {
    Serial.print("Pos: "); Serial.println(pos);

    switch (pp_state) {
        case 0:
            if (az_mini.pp_Run(100000) == 0)
                pp_state++;
            break;
        case 1:
            if (az_mini.driveIsTargetReached())
                pp_state++;
            break;
        case 2:
            if (az_mini.pp_Run(-100000) == 0)
                pp_state++;
            break;
        case 3:
            if (az_mini.driveIsTargetReached())
                pp_state = 0;
            break;
    }
}

```

```

delay(100);
}

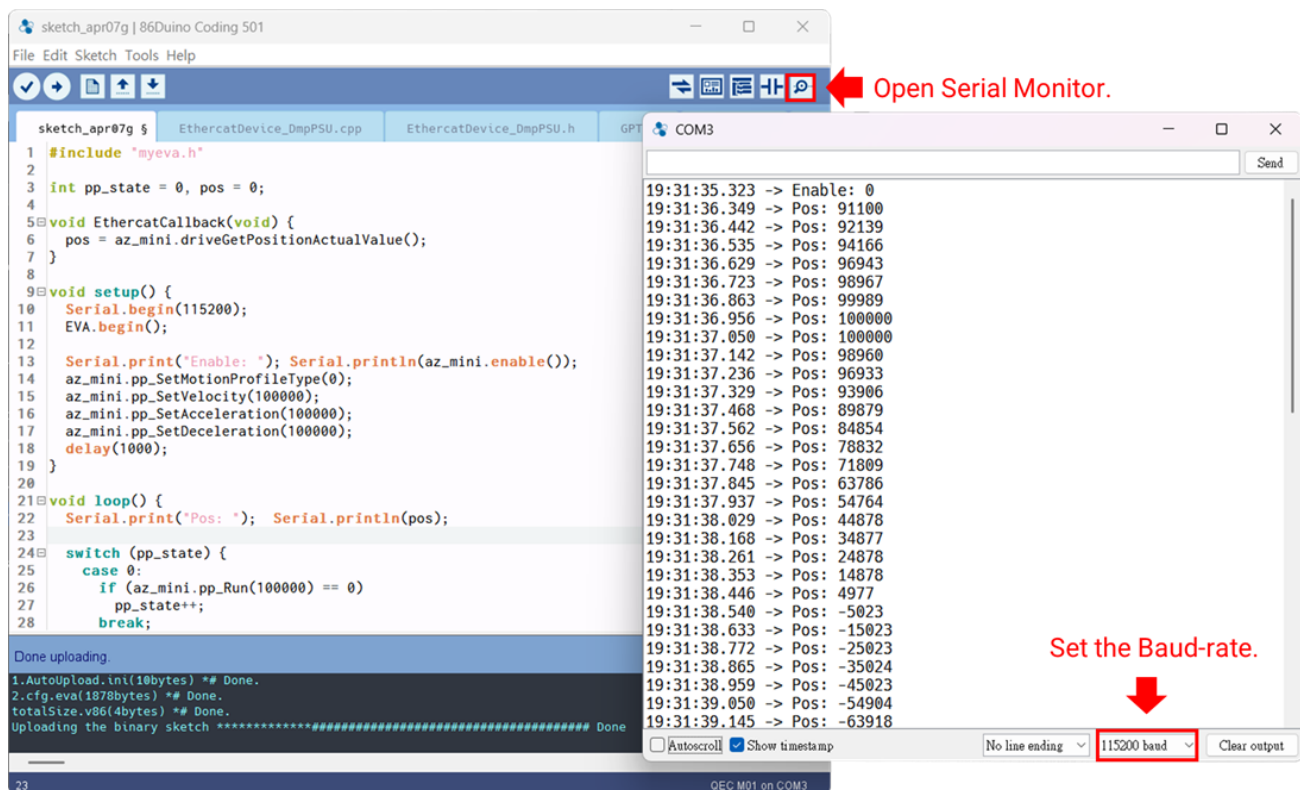
```

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。

File Edit Sketch Tools Help



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 “Enable: 0”を表示します。



モータの現在位置をシリアルモニタに表示します。

```
19:31:36.349 -> Pos: 91100
19:31:36.442 -> Pos: 92139
19:31:36.535 -> Pos: 94166
19:31:36.629 -> Pos: 96943
19:31:36.723 -> Pos: 98967
19:31:36.863 -> Pos: 99989
19:31:36.956 -> Pos: 100000
19:31:37.050 -> Pos: 100000
19:31:37.142 -> Pos: 98960
19:31:37.236 -> Pos: 96933
19:31:37.329 -> Pos: 93906
19:31:37.468 -> Pos: 89879
19:31:37.562 -> Pos: 84854
19:31:37.656 -> Pos: 78832
19:31:37.748 -> Pos: 71809
19:31:37.845 -> Pos: 63786
19:31:37.937 -> Pos: 54764
19:31:38.029 -> Pos: 44878
19:31:38.168 -> Pos: 34877
19:31:38.261 -> Pos: 24878
19:31:38.353 -> Pos: 14878
19:31:38.446 -> Pos: 4977
19:31:38.540 -> Pos: -5023
19:31:38.633 -> Pos: -15023
19:31:38.772 -> Pos: -25023
19:31:38.865 -> Pos: -35024
19:31:38.959 -> Pos: -45023
19:31:39.050 -> Pos: -54904
19:31:39.145 -> Pos: -63918
```

☐ Autoscroll ☒ Show timestamp No line ending 115200 baud Clear output

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 マスタ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 マスタ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：

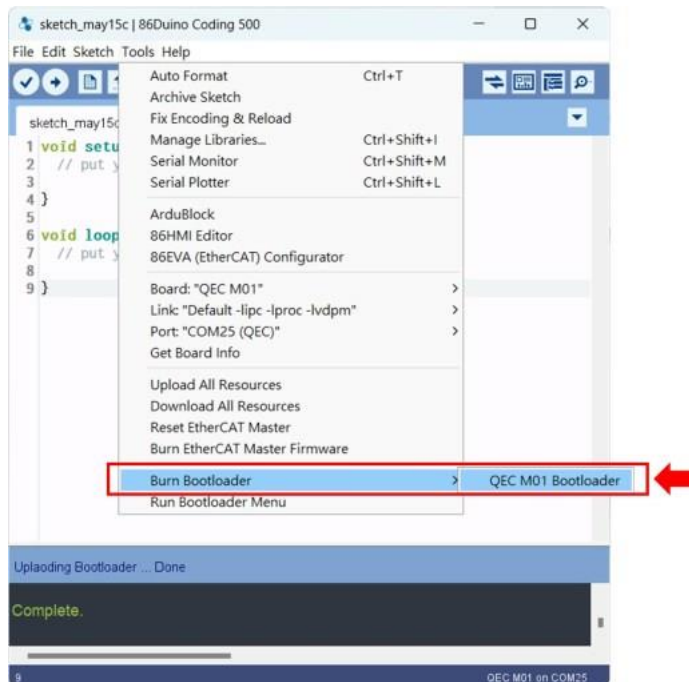
ステップ 1: QEC-M のセットアップ

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。

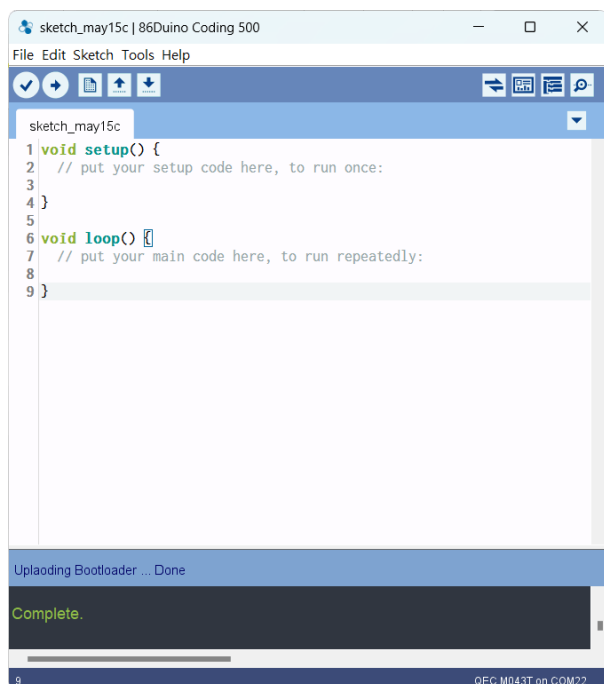
ステップ 2: 「Burn Bootloader」ボタンをクリック

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.

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。