ICOP Technology Inc.

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 (GUI コンフィグレーション)編

86Duino Coding IDE 500 EtherCAT Library

(Version 1.0)



記述日	バージョン	備考
2024/9/11	VERSION1.0	NEW RELEASE.



本ガイドでは、QEC-M-01 (EtherCAT マスタ) とオリエンタルモーター株式会社製 AZ-mini EtherCAT スレーブの 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。

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:

	Pin #	Signal Name	Pin #	Signal Name
	1	LAN1_TX+	2	LAN1_TX-
	3	LAN1_RX+	4	VS+
8 2,1	5	VP+	6	LAN1_RX-
	7	VS-(GND)	8	VP-(GND)

* PoE LAN with the Red Housing; Regular LAN with Black Housing.

* L4, L5, L7, L8 pins are option, for RJ45 Power IN/OUT.

 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 Master)
- 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)
- 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 マスタ

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。
- 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。
- 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。

ノードアドレスは ECAT ID ×1 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)	マスタ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. ソフトウエア/開発環境

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

QEC Quicker, Easier Control	EtherCAT 🗸 Pi	roducts ~ [Download 🗸	Documentations $$	📕 English 🗸	Q
Download The open source 86Duino Software (IDE) makes it easy to write code and up <u>Getting Started page</u> for Installation instructions. (<u>Release Note</u>)	pload it to the QEC. Re	efer to the	8	86Duino Coding Date: 2024.08 Download	IDE 500 3.15	
About how to update the QEC Master (QEC-M series products) with the latest v	ersion of the 86Duinc	IDE, please se	e <u>this page</u> .			

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。

e) tools	2023/10/26 14:54
🐻 tools-builder	2023/10/26 14:54
💦 86duino.exe	2023/10/26 14:49
💦 86duino_debug.exe	2023/10/26 14:49
arduino.l4j.ini	2023/10/26 14:49
86duino.exe	2023/10/26 14:49 2023/10/26 14:49 2023/10/26 14:49

*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。
- 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、COM4 (QEC))。

💲 sketch_mar28a	86Duino Coding 500	-			👌 sketch_mar28a	86Duino Coding 500	_		×
File Edit Sketch To	ols Help				File Edit Sketch To	ools Help			
Sketch_mar28a void setu 2 // put y 3 4 } 5 6 void loop 7 // put y	Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries Serial Monitor Serial Plotter ArduBlock 86HMI Editor 86EVA (EtherCAT) Configurator	Ctrl+T Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L			<pre>> • • • • 1 sketch_mar28a 1 void setu 2 // put y 3 4 } 5 6 void loop 7 // put y</pre>	Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries Serial Monitor Serial Plotter ArduBlock 86HMI Editor 86EVA (EtherCAT) Configurator	Ctrl+T Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L		₽
9 }	Board: "QEC M01" Link: "Default -lipc -lproc -lvdpm Port: "COM3 (QEC)"	n	 Board Vortex 86Dui 	s Manager 186 (32-bits) Boards no ZERO	9 }	Board: "QEC M01" Link: "Default -lipc -lproc -lvdpm' Port: "COM3 (QEC)"		> > > Se	rial p
	Get Board Info		86Dui	no ONE		Get Board Info		~ cc) 6М
	Upload All Resources Download All Resources Reset EtherCAT Master		86Dui QEC M QEC M OEC M	no EduCake 1043T 1070T 1090T		Upload All Resources Download All Resources Reset EtherCAT Master			
	Burn Bootloader Run Bootloader Menu		> QEC N	1150T 101		Burn Bootloader Run Bootloader Menu		>	
Uplaoding Bootloader	Done		Arduir	no AVR Boards	> Uplaoding Bootloade	r Done			
Complete.					Complete.				•
9		QEC	M01 on COM3		9		QEC	M01 on CO	М3

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。

ソフトウエア・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、EtherCAT 通信を確立し、Profile Position (PP) モードで AZ-mini ドライバを制御しま す。モータの位置は周期的に更新され、目標位置は 0~10,000 units の間で交互に変化し、連続的な正と逆移動 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。

★ 86EVA Configurator 0.7.5.0	-	×
EtherCAT Network		
Scan Slave Devices wait		
100%		
Select COM11 V	t	
		v0.7.5.0

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。



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。

<table-cell-rows> 86EVA Conf</table-cell-rows>	igurator 0.7.5.0					-	□ ×
	Back						
			General				
			Object Nar	me : az_m	nin i		
			Alias Addre	ess : 2000		Set	
			Vendor	סאט: סו	0002BE (0	riental	motor)
1# q	w	e r	t y	u	io	р	≤
ABC	a s	d	f g	h j	k		4
	z	x c	v b	n	m	•	, :
cancel	<				;	•	~

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

👌 sketch_sep10b 86[Duino Coding 500	-		×
File Edit Sketch Tools	; Help			
		4	II E	P
sketch_sep10b	GPT.h myeva.cpp myeva.h			•
1 #include	"myeva.h"			
2 void set	up() {			
3 EVA.be	<pre>gin();</pre>			
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 $\pm \downarrow$: Profile Position (PP) $\pm \downarrow$
- EtherCAT $\pm \models$: 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"、AZD-KRED は"az_mini"

1. Setup()関数内の説明:

- Serial の初期化 (115200)
- EVA.begin()関数に "myCallback"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。
 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り、モータの現在位置をリアルタイムで更新

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: profile position 開始し、目標位置(10,000 単位)へ移動を開始する。移動開始後、次の ステートに遷移する
- ステート**1**:現在の profile position 移動を終了する
- ステート2:モータが目標位置に到達するのを待つ。目標位置に到達したら、次の状態に遷移する
- ステート 3:元の位置(0 units)に戻る profile position 移動を開始する。移動開始後、次の状態に遷 移する
- ステート **4**:現在の profile position 移動を終了する
- ステート 5:モータが目標位置に到達するのを待つ。目標位置に到達したら、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。

このコードは EtherCAT 通信を確立し、Profile Position (PP) モードで AZ-mini ドライバを制御します。モータの 位置は周期的に更新され、目標位置は 0~10,000 units の間で交互に変化し、連続的な正と逆移動をシミュレー トします。

```
プログラム例:
```

```
#include "myeva.h"
int pp_state = 0, pos = 0;
void myCallback() {
  pos = az_mini.driveGetPositionActualValue();
}
void setup() {
  Serial.begin(115200);
  EVA.begin(myCallback);
  // put your setup code here, to run once:
  Serial.print("Enable: ");
  Serial.println(az_mini.driveEnable());
  delay(1000);
}
void loop() {
  Serial.print("Pos: ");
  Serial.println(pos);
  switch (pp_state) {
    case 0:
       if (az_mini.profilePositionBegin(10000, 10000, 10000) == 0)
        pp_state++;
       break;
    case 1:
       az_mini.profilePositionEnd();
       pp_state++;
       break;
    case 2:
       if (az_mini.drivelsTargetReached())
        pp_state++;
       break;
    case 3:
       if (az_mini.profilePositionBegin(0, 10000, 10000) == 0)
        pp_state++;
       break;
    case 4:
```

```
az_mini.profilePositionEnd();
    pp_state++;
    break;
    case 5:
        if (az_mini.driveIsTargetReached()) {
            pp_state = 0;
        }
        break;
    }
    delay(100);
}
```

sketch_dec19a | 86Duino Coding 500
 File Edit Sketch Tools Help
 File Edit Sketch I a sketch_dec19a s

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。

🚯 pp_mode_86eva 86Duino Coding 500	- 🗆 X	
File Edit Sketch Tools Help		
	🔹 📾 🖬 💽 🛑 Open Serial Monitor.	
pp_mode_86eva GPT.h myeva.cpp myeva.h		
1 #include "myeva.h"	🍣 COM11	- 🗆 X
3 int pp state = 0 pos = 0:		Send
4	16:38:11.596 -> Enable: 0	
<pre>5 void myCallback() {</pre>	16:38:12.618 -> Pos: 0	
<pre>6 pos = az_mini.driveGetPositionActualValue();</pre>	16:38:12.711 -> Pos: 903	
7 }	16:38:12.804 -> Pos: 1903	
8	16:38:12.896 -> Pos: 2903	
<pre>9 void setup() {</pre>	16:38:13.037 -> Pos: 3894	
10 Serial.begin(115200);	16:38:13.129 -> Pos: 4894	
	16:38:13.224 -> Pos: 5894	
12 EVA.begin(myCallback);	16:38:13.318 -> Pos: 6894	
13 // put your setup code nere, to run once: 14 Serial print("Enable: "):	16:38:13.410 -> Pos: 7894	
15 Serial println(az mini driveEnable()):	16:38:13.504 -> Pos: 8894	
16 delay(1000):	16:38:13.599 -> Pos: 9893	
17 }	16:38:13.691 -> Pos: 9999	
	16:38:13.833 -> Pos: 9999	
Reset EtherCAT Master Done	16:38:13.926 -> Pos: 9096	
1.AutoUpload.ini(10bytes) *# Done. 2.cfn.eva(1374bytes) *# Done	16:38:14.017 -> Pos: 8095	
totalSize.v86(4bytes) *# Done.	16:38:14.111 -> Pos: 7095	Set the Baud-rate.
oproducing the othery sketch concerns and an and an and an and an one	16:38:14.207 -> Pos: 6096	
43	16:38:14.298 -> Pos: 5096	
		No line and inc. of 115000 band of Class entert
	- Autoscion Snow nuestamp	1100 mile enging - 110200 baud - Clear output

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

👌 сом11	-	×
		Send
16:38:11.596 -> Enable: 0		

モータの現在位置をシリアルモニタに表示します。

♣ COM11	- 🗆 X
	Send
16:38:11.596 -> Enable: 0	
16:38:12.618 -> Pos: 0	
16:38:12.711 -> Pos: 903	
16:38:12.804 -> Pos: 1903	
16:38:12.896 -> Pos: 2903	
16:38:13.037 -> Pos: 3894	
16:38:13.129 -> Pos: 4894	
16:38:13.224 -> Pos: 5894	
16:38:13.318 -> Pos: 6894	
16:38:13.410 -> Pos: 7894	
16:38:13.504 -> Pos: 8894	
$16:38:13.599 \rightarrow Pos: 9893$	
$16:38:13.691 \rightarrow Pos: 9999$	
$16:38:13,833 \rightarrow Pos: 9999$	
16:38:13 926 -> Pos: 9096	
$16:38:14 \ 0.17 \ -> Pos: 8095$	
16:38:14 111 -> Pos: 7095	
16.38.14 207 -> Post 6006	
16.38.14 208 -> Pos: 5006	
10.30.14.230 -> POS. 3090	
🗌 Autoscroll 🗹 Show timestamp No li	ne ending \vee 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 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: <u>Software</u>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。