

プログラマブル多軸モーションコントローラ

スタートアップガイド DirectPWM インタフェース編

形CK3W-AX1313□

形CK3W-AX2323□

Startup
Guide

おことわり

- (1) 本マニュアルの内容の一部または全部を無断で複写、複製、転載することを禁じます。
- (2) 本マニュアルの内容に関しては、改良のため予告なしに仕様などを変更することがあります。あらかじめご了承ください。
- (3) 本マニュアルの内容に関しては、万全を期しておりますが、万一ご不審な点や誤りなど、お気づきのことがありましたら、お手数ですが、弊社支店または営業所までご連絡ください。その際、巻末記載の**Man.No.**（マニュアルNo.）も併せてお知らせください。

商標

- Microsoft、Windows、Excel、Visual Basic、Microsoft Edgeは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- EtherCAT®は、ドイツBeckhoff Automation GmbHによりライセンスされた特許取得済み技術であり登録商標です。

その他、本文中に掲載している会社名および製品名は、それぞれ各社の商標または登録商標です。

著作権

- スクリーンショットはマイクロソフトの許諾を得て使用しています。
- 本製品にはサードパーティ製のソフトウェアが組み込まれています。このソフトウェアに関連するライセンスと著作権については、http://www.fa.omron.co.jp/nj_info_j/ をご覧ください。

目次構成

		1
1	本資料の概要	2
2	対象機器とデバイス構成	3
3	DirectPWM インタフェースの接続手順	

目次

目次構成.....	3
注意事項.....	5
関連マニュアル	6
用語と定義.....	7
改訂履歴.....	8

第 1 章 本資料の概要

1-1 概要	1-2
1-1-1 対象となる読者の方々	1-2

第 2 章 対象機器とデバイス構成

2-1 デバイス構成.....	2-2
-----------------	-----

第 3 章 DirectPWM インタフェースの接続手順

3-1 作業の流れ.....	3-2
3-2 コントローラの設定準備	3-3
3-2-1 新規プロジェクトの作成.....	3-3
3-2-2 コントローラの初期設定.....	3-4
3-3 各種機器の接続.....	3-6
3-4 コントローラの各種設定	3-8
3-4-1 注記一覧.....	3-13
3-5 設定の確認.....	3-17
3-6 モータのチューニング	3-19
3-6-1 オープンループテスト	3-19
3-6-2 電流ループのゲイン設定.....	3-20
3-6-3 バンド幅のオート設定	3-21
3-6-4 バンド幅の手動設定	3-23
3-6-5 フィードフォワード値の設定	3-25
3-6-6 動作確認とチューニングパラメータのプロジェクト化	3-27

注意事項

- 実際のシステム構築に際しては、システムを構成する各機器・装置の仕様をご確認のうえ、定格・性能に対し余裕を持った使い方をし、万一故障があっても危険を最小にする安全回路などの安全対策を講じてください。
- システムを安全にご使用いただくため、システムを構成する各機器・装置のマニュアルや取扱説明書などを入手し、「安全上のご注意」「安全上の要点」など安全に関する注意事項を含め、内容を確認し理解したうえで使用してください。
- システムが適合すべき規格・法規または規制に関しては、お客様自身でご確認ください。
- 本資料の一部または全部を、オムロン株式会社の許可なしに複写、複製、再配布することを禁じます。
- 本資料の記載内容は、改良のため予告なく変更されることがあります。

本資料で使われているマークには、次のような意味があります。



使用上の注意

製品の動作不能、誤動作、または性能・機能への悪影響を予防するために実施または回避すべきことを示します。



参考

必要に応じて読んでいただきたい項目です。
知っておくと便利な情報や、使用するうえで参考となる内容について説明しています。

関連マニュアル

システムを安全にご使用いただくため、機器・装置のマニュアルや取扱説明書などを必ず入手し、「安全上のご注意」「安全上の要点」など安全に関する注意事項を含め、内容を確認のうえ使用してください。

オムロン株式会社（以下、オムロン）、米国デルタ タウ データ システムズ社（以下、DT 社）のマニュアルは以下のとおりです。

メーカー	Man. No.	形式	マニュアル名称
オムロン	SBCE-435	形 CK5M-□ 形 CK3M-□ 形 CK3W-□	プログラマブル多軸モーションコントローラ ユーザーズマニュアル ハードウェア編
DT 社	SBCE-404	—	Power PMAC ユーザーズマニュアル
DT 社	SBCE-405	—	Power PMAC ソフトウェアリファレンスマニュアル
DT 社	SBCE-439	—	Power PMAC IDE ユーザーズマニュアル

用語と定義

用語	説明・定義
PMAC	Programmable Multi Axis Controller の略です。
Power PMAC IDE	コントローラの設定、ユーザプログラムの作成、モニタを行うためのパソコン用ソフトウェアです。
DirectPWM	米国 Delta Tau Data System 社独自のサーボドライバとのインタフェース方式です。
パルスエンコーダ	パルス信号を出力するタイプのエンコーダです。

改訂履歴

改訂記号は、表表紙・裏表紙に記載されている Man. No.の後尾に付記されます。

Man. No. **SBCE-467B**

↑
改訂記号

改訂記号	改訂年月	改訂理由
A	2019 年 7 月	初版
B	2023 年 1 月	- 対象サーボドライバの変更による改訂 - 誤記訂正

本資料の概要

本章では、本資料の概要を記載します。

1-1	概要	1-2
1-1-1	対象となる読者の方々	1-2

1-1 概要

本資料はオムロン製プログラマブル多軸モーションコントローラ 形 CK□M-□□□□（以下、コントローラ）と、DirectPWM インタフェース方式に対応したサーボドライバを接続する手順とその確認方法をまとめたものです。

「第 3 章 DirectPWM インタフェースの接続手順 (P.3-1)」で記載している設定内容および設定手順のポイントを理解することにより、DirectPWM インタフェース方式に対応したサーボドライバに指令を送り、サーボモータを制御できるようになります。

本資料では CK3W-AX1313□を用い、パルスエンコーダによって位置フィードバック、速度フィードバックを行う場合を例に挙げて接続手順を説明します。^{*1}

^{*1}. CK3W-AX2323□を使用する場合、DirectPWM インタフェースは CK3W-AX1313□と同じですが、接続できるエンコーダの種類が異なるためエンコーダの設定を変更する必要があります。エンコーダの設定に関しては下記の資料を参照してください。

- 『スタートアップガイド 正弦波エンコーダ編』
- 『スタートアップガイド SSI/Mitutoyo/EnDat 2.1/2.2 シリアルエンコーダ編』

1-1-1 対象となる読者の方々

本ガイドは、電気の知識（電気工事士あるいは同等の知識）を有する、次の方を対象に記述しています。

- FA 機器の導入を担当される方
- FA システムを設計される方
- FA 機器を設置、接続される方
- FA 現場を管理される方

また、DT 社のマニュアルに記載されている内容をご理解いただいている方を対象に記述しています。

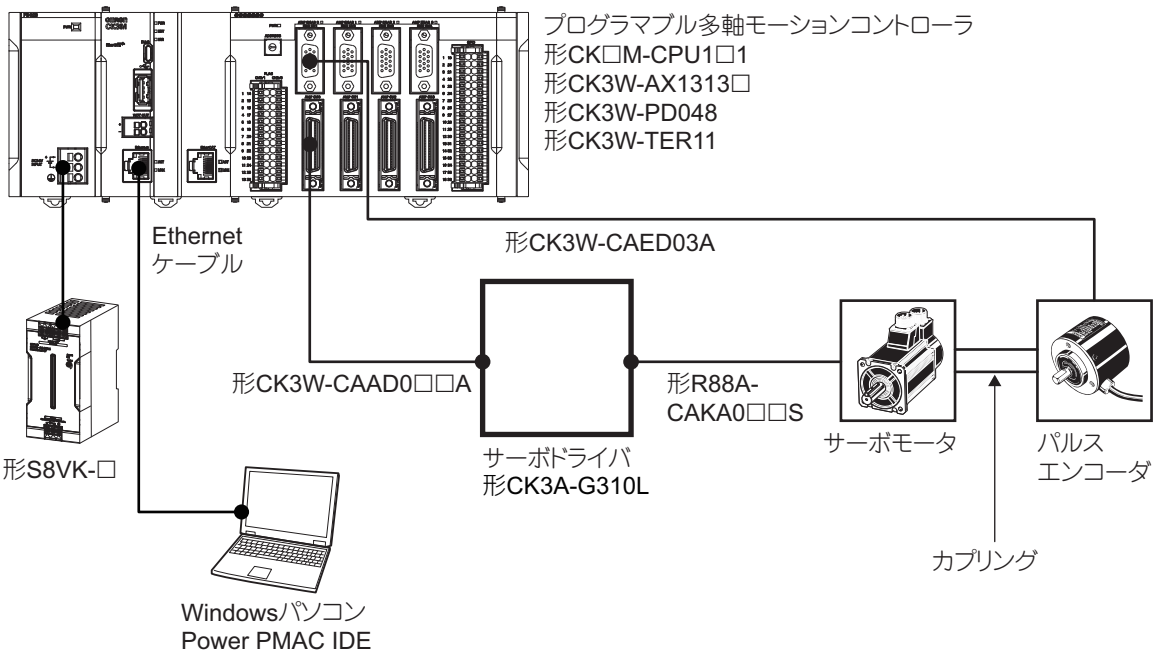
対象機器とデバイス構成

本章では、本資料における接続の対象機器およびシステム構成を記載します。

2-1	デバイス構成	2-2
-----	--------------	-----

2-1 デバイス構成

本資料の接続手順を再現するための構成機器は以下のとおりです。
本資料では DirectPWM インタフェースの設定例を示すために、モータの出力軸にパルスエンコーダを接続した構成を用いて説明しています。あくまで設定例を示すための構成であり、一般的な構成とは異なります。



使用上の注意

サーボモータおよびエンコーダは必ず固定して使用してください。固定せずにモータを動作させると機器の故障の原因となります。

メーカー	名称	形式	バージョン
オムロン	プログラマブル多軸モーションコントローラ CPU ユニット	形 CK□M-CPU1□1	Ver.2.5.2 以降
オムロン	プログラマブル多軸モーションコントローラ 軸インタフェースユニット	形 CK3W-AX1313□	—
オムロン	プログラマブル多軸モーションコントローラ 電源供給ユニット	形 CK3W-PD048	—
オムロン	プログラマブル多軸モーションコントローラ エンドカバー	形 CK3W-TER11	—
オムロン	DirectPWM ケーブル	形 CK3W-CAAD0□□A	—
オムロン	モータケーブル	形 R88A-CAKA0□□S	—
オムロン	エンコーダケーブル	形 CK3W-CAES03A	—
オムロン	サーボドライバ	形 CK3A-G310L	—
オムロン	サーボモータ	形 R88M-K05030T	—
オムロン	パルスエンコーダ	形 E6B2-CWZ1X	—
オムロン	カプリング	形 E69-C68B	—

メーカー	名称	形式	バージョン
オムロン	スイッチング パワーサプライ	形 S8VK-□	—
—	Windows パソコン	—	—
DT 社	Power PMAC 設定ツール	Power PMAC IDE	Ver.4.3 以降

3

DirectPWM インタフェースの接続手順

3

本章では、コントローラとサーボドライバを接続し、DirectPWM インタフェースを用いてモーション制御機器を動作させるための手順について記載します。説明は、コントローラが工場出荷時の初期設定状態であることを前提とします。

3-1	作業の流れ	3-2
3-2	コントローラの設定準備	3-3
3-2-1	新規プロジェクトの作成	3-3
3-2-2	コントローラの初期設定	3-4
3-3	各種機器の接続.....	3-6
3-4	コントローラの各種設定	3-8
3-4-1	注記一覧	3-13
3-5	設定の確認	3-17
3-6	モータのチューニング	3-19
3-6-1	オープンループテスト	3-19
3-6-2	電流ループのゲイン設定	3-20
3-6-3	バンド幅のオート設定	3-21
3-6-4	バンド幅の手動設定	3-23
3-6-5	フィードフォワード値の設定	3-25
3-6-6	動作確認とチューニングパラメータのプロジェクト化	3-27

3-1 作業の流れ

コントローラとサーボドライバを接続し、DirectPWM インタフェースを用いてモーション制御機器を動作させるための手順は以下のとおりです。

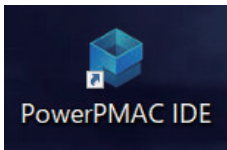
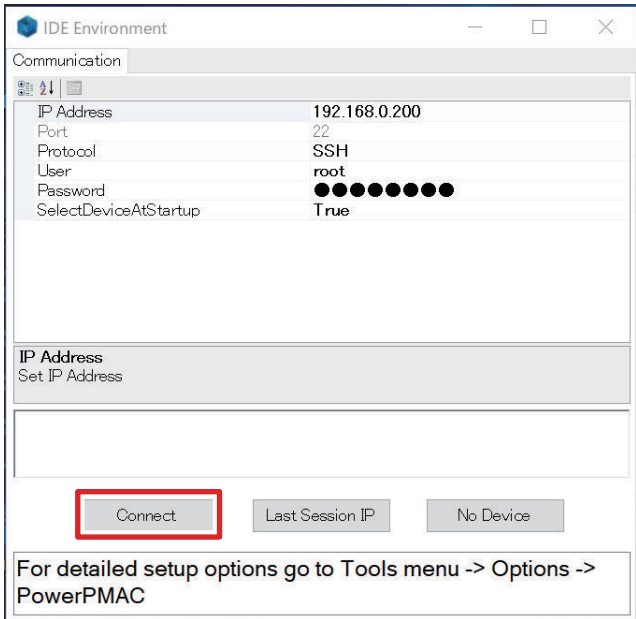
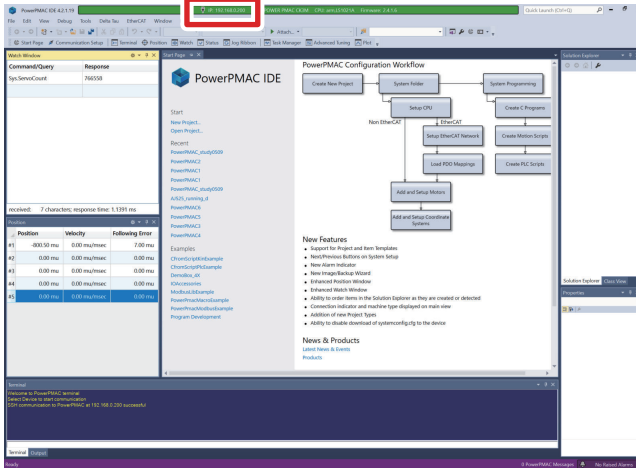
「3-2 コントローラの設定準備 (P.3-3)」	コントローラの設定準備を行います。
▼	
「3-2-1 新規プロジェクトの作成 (P.3-3)」	
▼	
「3-2-2 コントローラの初期設定 (P.3-4)」	
▽	
「3-3 各種機器の接続 (P.3-6)」	各種機器の接続と配線を行います。
▽	
「3-4 コントローラの各種設定 (P.3-8)」	コントローラの設定を行います。
▽	
「3-5 設定の確認 (P.3-17)」	これまでの設定が正しいことを確認します。
▽	
「3-6 モータのチューニング (P.3-19)」	Power PMAC IDE のチューニングツールを使用してモータのチューニングを行います。
▼	
「3-6-1 オープンループテスト (P.3-19)」	
▼	
「3-6-2 電流ループのゲイン設定 (P.3-20)」	
▼	
「3-6-3 バンド幅のオート設定 (P.3-21)」	
▼	
「3-6-4 バンド幅の手動設定 (P.3-23)」	
▼	
「3-6-5 フィードフォワード値の設定 (P.3-25)」	
▼	
「3-6-6 動作確認とチューニングパラメータのプロジェクト化 (P.3-27)」	

3-2 コントローラの設定準備

コントローラの設定準備を行います。
Power PMAC IDE を、あらかじめパソコンにインストールしてください。

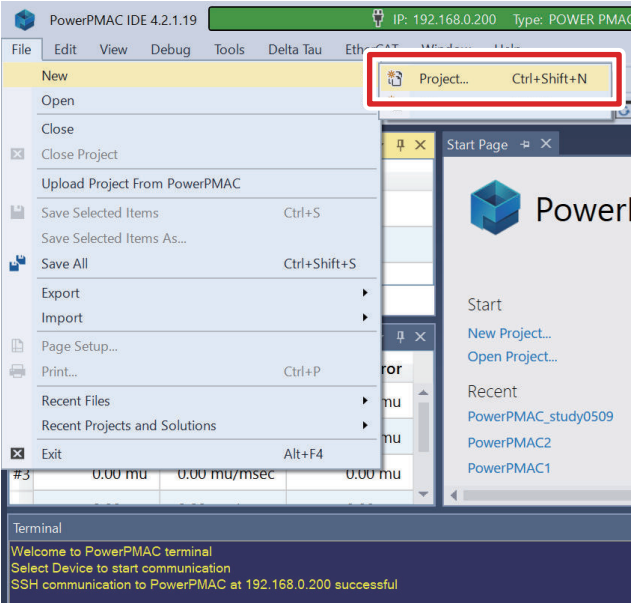
3-2-1 新規プロジェクトの作成

以下の手順に従い、新規プロジェクトの作成を行います。

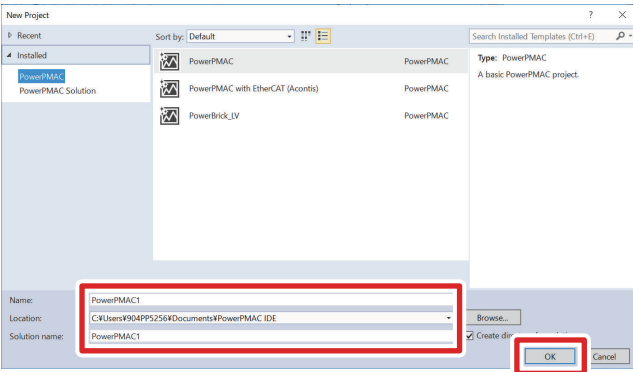
1	コントローラとパソコンを Ethernet ケーブルで接続します。	
2	コントローラの電源を投入します。	
3	Power PMAC IDE を起動します。 ・ 起動時に、アクセス権確認用のダイアログが表示される場合、起動する選択を行ってください。	
4	Communication 画面が表示されますので、接続先のコントローラの IP アドレスを指定し、[Connect] ボタンをクリックします。 ・ コントローラのデフォルトの IP アドレスは“192.168.0.200”です。 ・ 必要であれば、Windows の IP アドレスを“192.168.0.X”へ変更してください。	
5	Power PMAC IDE が起動し、コントローラとオンラインの状態になります。	

- 6**

[File] メニューの [New] - [Project] を選択します。


- 7**

任意のプロジェクト名、保存先を入力し、[OK] ボタンをクリックします。



3-2-2 コントローラの初期設定

以下の手順に従い、コントローラの初期設定を行います。

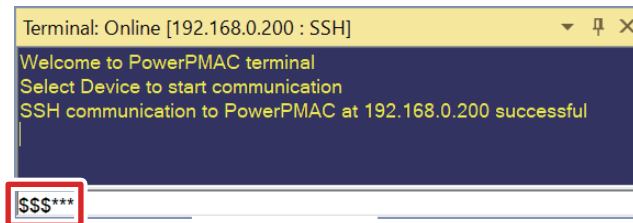


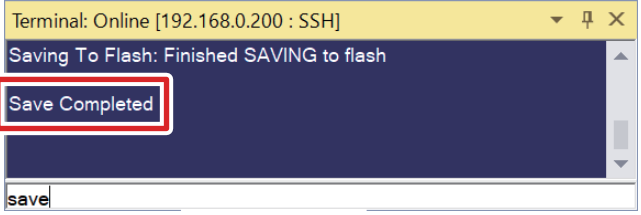
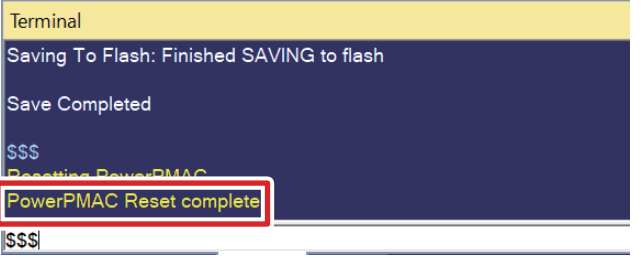
使用上の注意

初期設定により、メモリがすべてクリアされますので、必要なデータがコントローラ内に残っている場合は、データを保管してください。

- 1**

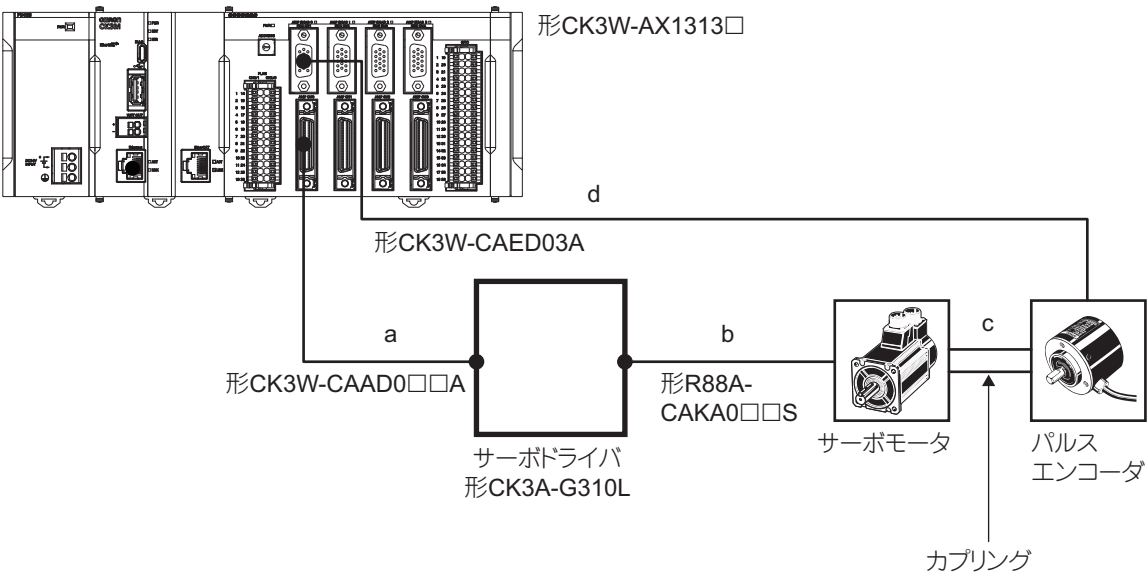
Terminal から [\$\$\$***] コマンドをタイプし、コントローラを工場出荷状態にします。



2	Power PMAC IDE の Terminal で [save] コマンドをタイプします。 終了すると、Terminal に “Save Completed” と表示されます。	 Terminal: Online [192.168.0.200 : SSH] Saving To Flash: Finished SAVING to flash Save Completed save
3	Power PMAC IDE の Terminal で [\$\$\$] コマンドをタイプします。 終了すると、Terminal に “PowerPMAC Reset complete” と表示されます。	 Terminal Saving To Flash: Finished SAVING to flash Save Completed \$\$\$ Resetting PowerPMAC PowerPMAC Reset complete \$\$\$

3-3 各種機器の接続

軸インタフェースユニットと各種機器の接続について示します。



上図 a、b、c、d の箇所を以下の要領に従い接続します。

a. コントローラとサーボドライバ間の接続

形 CK3W-AX1313□のアンプ接続コネクタとサーボドライバの C2 コネクタを、以下の専用ケーブルで接続します。

メーカー	名称	形式	長さ
オムロン	DirectPWM ケーブル	形 CK3W-CAAD009A	0.9m
		形 CK3W-CAAD018A	1.8m
		形 CK3W-CAAD036A	3.6m

b. サーボドライバとサーボモータ間の接続

サーボドライバの P2 コネクタとサーボモータのモータ用コネクタを、以下の専用ケーブルで接続します。

メーカー	名称	形式	長さ
オムロン	モータケーブル	形 R88A-CAKA003S	3m
		形 R88A-CAKA005S	5m
		形 R88A-CAKA010S	10m
		形 R88A-CAKA015S	15m
		形 R88A-CAKA020S	20m
		形 R88A-CAKA030S	30m
		形 R88A-CAKA040S	40m
		形 R88A-CAKA050S	50m

c. サーボモータとエンコーダ間の接続

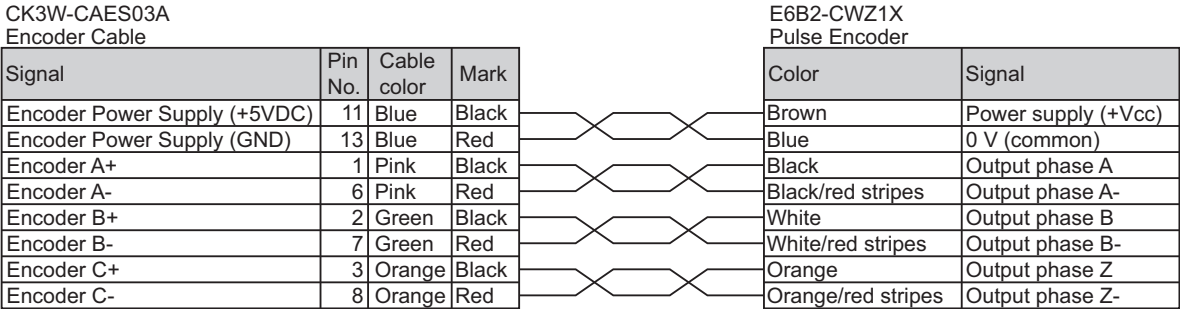
サーボモータの回転軸とパルスエンコーダの回転軸を、以下のカプリングで接続します。
カプリング: 形 E69-C68B

d. コントローラとエンコーダ間の配線

形 CK3W-AX1313□とパルスエンコーダは、以下の専用ケーブルで接続します。

メーカー	名称	形式	長さ
オムロン	エンコーダケーブル	形 CK3W-CAES03A	3m

専用ケーブル（形 CK3W-CAES03A）とパルスエンコーダは、以下の配線図に従って配線します。



3-4 コントローラの各種設定

以下の手順に従い、DirectPWM インタフェースとパルスエンコーダでサーボモータを制御する場合のコントローラの各種設定を行います。

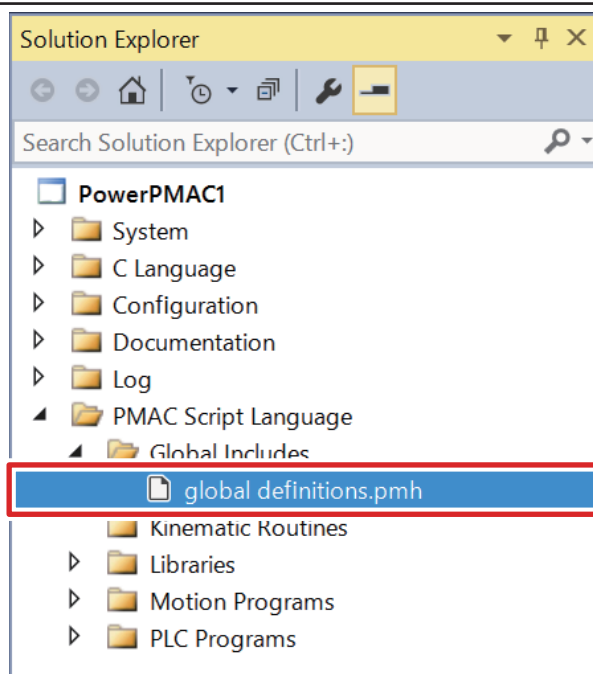


使用上の注意

- モータを動作させる前に、以下に示す手順 2 で global definitions.pmh に書き込む設定項目は、使用するモータとサーボドライバに応じて適切な値を設定してください。設定値が適切でない場合、過剰な電流が流れ、機器が故障する可能性があります。
設定に関しては、「3-4-1 注記一覧 (P.3-13)」の注記*24～*26 を参照してください。
- 以下に示す手順 9、11 で [Motor[1].laBias]、[Motor[1].lbBias] を 0 以外に設定する場合、モータが回転する可能性があります。回転しても問題のないように機器の安全性を確認してから設定を行ってください。

1

Solution Explorer の [PMAC Script Language] - [Global Includes] の下にある global definitions.pmh を開きます。



2

右のテキストを global definitions.pmh に書き込みます。

- 右記のテキスト中の注記*1～*30 の各設定項目の詳細は、「3-4-1 注記一覧 (P.3-13)」を参照してください。

```
Sys. WpKey = $AAAAAAA

//global setting
Gate3[0]. PhaseServoDir = 0;
Gate3[0]. PhaseFreq = 10000; //10kHz
Gate3[0]. ServoClockDiv = 9; //1kHz
Sys. PhaseOverServoPeriod = 0.1;
Sys. ServoPeriod = 1;

//Encoder Setting
EncTable[1]. Type = 1; /*1
EncTable[1]. pEnc = Gate3[0]. Chan[0]. ServoCapt. a; /*2
EncTable[1]. ScaleFactor = 1/exp2(8); /*3
Gate3[0]. EncClockDiv = 5; //3.125MHz
Gate3[0]. Chan[0]. EncCtrl = 7; /*4

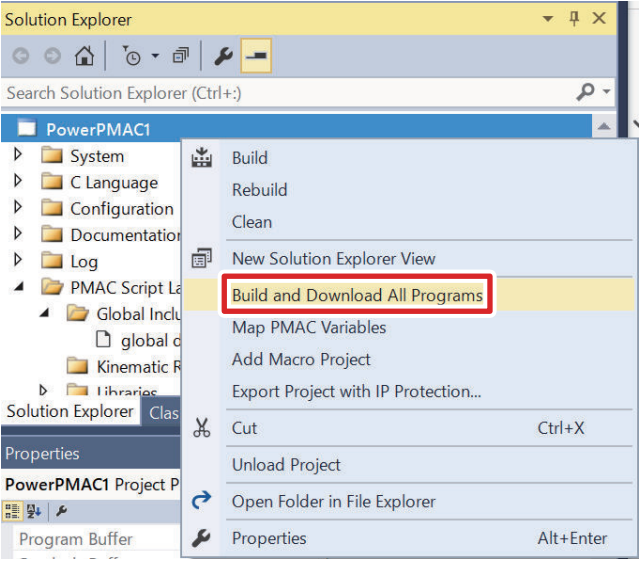
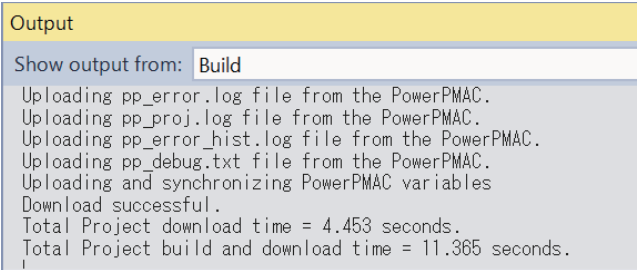
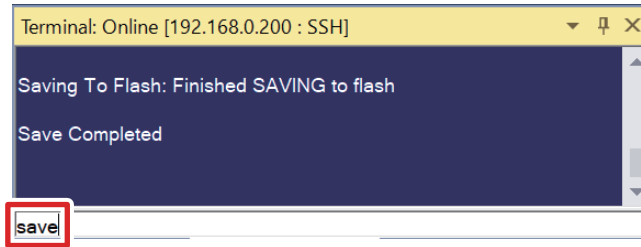
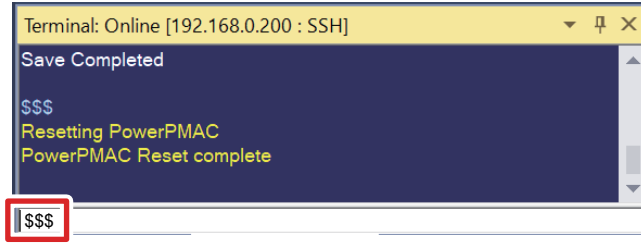
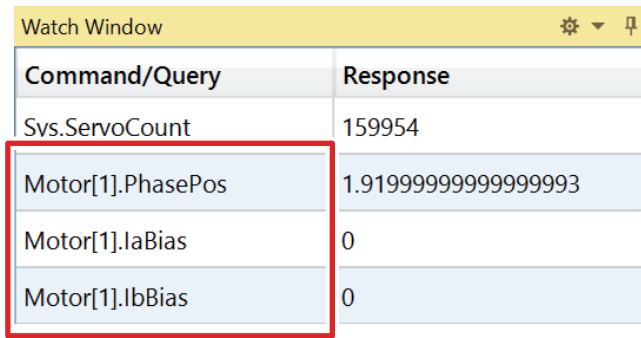
//DirectPWM AD Convertor setting
Gate3[0]. AdcAmpStrobe = $901001; /*5
Gate3[0]. AdcAmpHeaderBits = 4; /*6
Gate3[0]. AdcAmpClockDiv = 5; //3.125MHz

//DirectPWM PWM output setting
Gate3[0]. Chan[0]. PwmFreqMult = 2; /*7
Gate3[0]. Chan[0]. PwmDeadTime = 3.1 / 0.0533; /*8
Gate3[0]. Chan[0]. PackInData = 0; /*9
Gate3[0]. Chan[0]. PackOutData = 0; /*10
```

```
Sys. WpKey=$0

//Motor setting
Motor[1]. ServoCtrl = 1; //Enable the Motor[1].
Motor[1]. PhaseCtrl = 1; //Enable the commutation task.
Motor[1]. pPhaseEnc = Gate3[0]. Chan[0]. PhaseCapt. a; /*
11
Motor[1]. PhasePosSf = 2048/(256*2000*4/5); /*12
Motor[1]. PwmSf = 13458; /*13
Motor[1]. PhaseOffset = 683; /*14
Motor[1]. AmpFaultLevel = 1; /*15
Motor[1]. pLimits = 0; //Disable the Overtravel limit.
Motor[1]. WarnFeLimit = 4000; /*16
Motor[1]. FatalFeLimit = 8000; /*17
Motor[1]. pAmpEnable = Gate3[0]. Chan[0]. OutCtrl. a;
Motor[1]. pAmpFault = Gate3[0]. Chan[0]. Status. a;
Motor[1]. pCaptFlag = Gate3[0]. Chan[0]. Status. a;
Motor[1]. pCaptPos = Gate3[0]. Chan[0]. HomeCapt. a;
Motor[1]. pEncCtrl = Gate3[0]. Chan[0]. OutCtrl. a;
Motor[1]. pEncStatus = Gate3[0]. Chan[0]. Status. a;
Motor[1]. pMasterEnc = EncTable[1]. a;
Motor[1]. CurrentNullPeriod = 0; /*18
Motor[1]. pEnc = EncTable[1]. a /*19
Motor[1]. pEnc2 = EncTable[1]. a /*20
Motor[1]. pDac = Gate3[0]. Chan[0]. Pwm[0]. a; /*21
Motor[1]. pAdc = Gate3[0]. Chan[0]. AdcAmp[0]. a; /*22
Motor[1]. AdcMask = $FFFF0000; /*23
Motor[1]. MaxDac = 28377 * 3.33 / 16.25; /*24
Motor[1]. I2tSet = 28377 * 1.1 / 16.25; /*25
Motor[1]. I2tTrip = (Motor[1]. MaxDac * Motor[1]. MaxDac
-Motor[1]. I2tSet * Motor[1]. I2tSet) * 3; /*26
Motor[1]. AbsPhasePosOffset = 400; /*27
Motor[1]. PhaseFindingDac = 400; /*28
Motor[1]. PhaseFindingTime = 1000; /*29
Motor[1]. PowerOnMode=0; /*30
Motor[1]. InPosBand = 100;

// Setting Coordinate System
&1
#1->x
&1%100;
```

3 Power PMAC IDE の画面右上の [Solution Explorer] のプロジェクト名を右クリックし、[Build and Download All Programs] を選択し、ビルド&ダウンロードを実行します。											
4 Output Window で異常がないことを確認します。 転送が失敗した場合は、Output Window でエラーの内容を確認してください。プログラムのエラーであればプログラムを修正してください。											
5 Power PMAC IDE の Terminal で [save] コマンドをタイプします。 終了すると、Terminal に “Save Completed” と表示されます。											
6 Terminal で [\$\$\$] コマンドをタイプします。											
7 [Motor[1].PhaseOffset]の符号を決めるために、Watch Window に [Motor[1].PhasePos]、[Motor[1].IaBias]、[Motor[1].IbBias] を貼り付けます。	 <table border="1"> <thead> <tr> <th>Command/Query</th> <th>Response</th> </tr> </thead> <tbody> <tr> <td>Svs.ServoCount</td> <td>159954</td> </tr> <tr> <td>Motor[1].PhasePos</td> <td>1.9199999999999993</td> </tr> <tr> <td>Motor[1].IaBias</td> <td>0</td> </tr> <tr> <td>Motor[1].IbBias</td> <td>0</td> </tr> </tbody> </table>	Command/Query	Response	Svs.ServoCount	159954	Motor[1].PhasePos	1.9199999999999993	Motor[1].IaBias	0	Motor[1].IbBias	0
Command/Query	Response										
Svs.ServoCount	159954										
Motor[1].PhasePos	1.9199999999999993										
Motor[1].IaBias	0										
Motor[1].IbBias	0										

8	Terminal で [#1out0] コマンドをタイプします。	Terminal: Online [192.168.0.200 : SSH] Welcome to PowerPMAC terminal Select Device to start communication SSH communication to PowerPMAC at 192.168.0.200 successful \$\$\$ Resetting PowerPMAC PowerPMAC Reset complete #1out0 #1out0										
9	Terminal で [Motor[1].laBias=200]、 [Motor[1].lbBias=0] に設定します。	Terminal: Online [192.168.0.200 : SSH] SSH communication to PowerPMAC at 192.168.0.200 successful \$\$\$ Resetting PowerPMAC PowerPMAC Reset complete #1out0 Motor[1].laBias=200 Motor[1].lbBias=0 Motor[1].lbBias=0										
10	Watch Window で [Motor[1].PhasePos] の値を確認しま す。	Watch Window <table><thead><tr><th>Command/Query</th><th>Response</th></tr></thead><tbody><tr><td>Sys.ServoCount</td><td>230995</td></tr><tr><td>Motor[1].PhasePos</td><td>1505.92000000008966</td></tr><tr><td>Motor[1].laBias</td><td>200</td></tr><tr><td>Motor[1].lbBias</td><td>0</td></tr></tbody></table>	Command/Query	Response	Sys.ServoCount	230995	Motor[1].PhasePos	1505.92000000008966	Motor[1].laBias	200	Motor[1].lbBias	0
Command/Query	Response											
Sys.ServoCount	230995											
Motor[1].PhasePos	1505.92000000008966											
Motor[1].laBias	200											
Motor[1].lbBias	0											
11	Terminal で [Motor[1].lbBias=200] に設定します。 • [Motor[1].laBias] は 200 のままで す。	Terminal: Online [192.168.0.200 : SSH] \$\$\$ Resetting PowerPMAC PowerPMAC Reset complete #1out0 Motor[1].laBias=200 Motor[1].lbBias=0 Motor[1].lbBias=200 Motor[1].lbBias=200										
12	Watch Window で [Motor[1].PhasePos] の値を確認しま す。	Watch Window <table><thead><tr><th>Command/Query</th><th>Response</th></tr></thead><tbody><tr><td>Sys.ServoCount</td><td>320449</td></tr><tr><td>Motor[1].PhasePos</td><td>1503.36000000008971</td></tr><tr><td>Motor[1].laBias</td><td>200</td></tr><tr><td>Motor[1].lbBias</td><td>200</td></tr></tbody></table>	Command/Query	Response	Sys.ServoCount	320449	Motor[1].PhasePos	1503.36000000008971	Motor[1].laBias	200	Motor[1].lbBias	200
Command/Query	Response											
Sys.ServoCount	320449											
Motor[1].PhasePos	1503.36000000008971											
Motor[1].laBias	200											
Motor[1].lbBias	200											

13

Terminal で [Motor[1].IaBias=0]、
[Motor[1].IbBias=0] をタイプし、A
相と B 相のバイアス電流を 0 に戻しま
す。

Terminal: Online [192.168.0.200 : SSH]

PowerPMAC Reset complete
#1 out0
Motor[1].IaBias=200
Motor[1].IbBias=0
Motor[1].IbBias=200
Motor[1].IaBias=0
Motor[1].IbBias=0

Motor[1].IbBias=0

14

手順 10 と手順 12 の値を比較して、
[Motor[1].PhasePos] の値が減少して
いる場合は [Motor[1].PhaseOffset]
の符号を+に、増加している場合は - に
設定します。
• 今回の例の場合は減少しているため、
global definitions.pmh で
[Motor[1].PhaseOffset=683] に設
定します。
[Motor[1].PhaseOffset] の符号の
変更が必要な場合は、global
definitions.pmh で符号を変更し、手
順 3～6 に従って再びダウンロード
を行います。

15

モータとエンコーダをつなぐカプリ
ングを手動で回転させ、Position ウィ
ンドウで現在位置が目的のスケールで反
映されていることを確認します。
• 今回は 1 回転で 8000 カウントとな
るように
[EncTable[1].ScaleFactor] の値を
設定しているため、1 回転で現在位置
が 8000mu 増加します。
• 磁極のペア数が 5 のため、モータ 1
回転で [Motor[1].PhasePos] が 0
～2047.9999 を 5 回カウントアッ
プするように設定しています。
モータを 1 回転させながら、ウォッ
チウィンドウで
[Motor[1].PhasePos] を確認しま
す。

Position

Position

#1	632.00 mu
#2	0.00 mu
#3	0.00 mu
#4	0.00 mu



使用上の注意

[save] コマンドが正常実行されないと、転送したプロジェクトはコントローラに保存されませ
ん。この状態でコントローラの電源を OFF すると、転送したプロジェクトは消失します。



参考

パルスエンコーダのカウント方向(CW/CCW)を変更する場合は、手順 2 で global
definitions.pmh に書き込む下記の設定値の符号を、-に変更してください。
• EncTable[1].ScaleFactor
• Motor[1].PhasePosSf

3-4-1 注記一覧

手順 2 の各注記（設定項目の説明）の詳細を下表に示します。

No.	設定項目	設定値	説明
*1	EncTable[1].Type	1	EncTable[1]を Single-word (32 ビット) 読み出しとして有効にします。
*2	EncTable[1].pEnc	Gate3[0].Chan[0].ServoCapt.a	EncTable[1]にパルスエンコーダデータを割り付けます。
*3	EncTable[1].ScaleFactor	1/exp2(8)	EncTable[1]は 32 ビット、Gate3[0].Chan[0].ServoCapt (パルスエンコーダデータ) は 24 ビットのため、スケールファクタの設定値は以下の式で導出します。 設定値: $\frac{1}{2^{(32 \text{ ビット} - 24 \text{ ビット})}}$
*4	Gate3[0].Chan[0].EncCtrl	7	パルスエンコーダ変換方式を 4 通倍 CCW に設定します。
*5	Gate3[0].AdcAmpStrobe	\$ffffc	AMP Strobe Word を指定します。「\$ffffc」に設定すると、すべての AD コンバータに適合します。
*6	Gate3[0].AdcAmpHeaderBits	2	AD 変換データのヘッダ長を 2 ビットに設定します。サーボドライバの仕様に応じて設定してください。
*7	Gate3[0].Chan[0].PwmFreqMult	2	PWM 周波数を 15kHz に設定します。PWM 周波数は以下の式で計算します。 $f_{\text{PWM}} = \frac{\text{Gate3[0].Chan[0].PwmFreqMult} + 1}{2} \times f_{\text{IntPhase}}$ f_{IntPhase} : 内部フェーズクロック周波数 PWM 周波数は、40kHz 以下かつサーボドライバの最高入力周波数以下の値に設定してください。
*8	Gate3[0].Chan[0].PwmDeadTime	3.1 / 0.0533	PWM 信号のデッドタイムを 3.1 μs に設定します。デッドタイムは、以下の式で計算します。 Dead Time = 0.0533 μs $\times$ Gate3[0].Chan[0].PwmDeadTime サーボドライバの仕様に応じて設定してください。
*9	Gate3[0].Chan[0].PackInData	0	AdcAmp 圧縮: 無効 デジタル電流ループを実行している場合、データの圧縮はアルゴリズムの効率を向上させるため、有効に設定します。 今回は無効に設定します。
*10	Gate3[0].Chan[0].PackOutData	0	PWM/DAC 圧縮を無効に設定します。 転流とデジタル電流ループを計算している場合、データの圧縮はアルゴリズムの効率を向上させるため、有効に設定します。 今回は無効に設定します。
*11	Motor[1].pPhaseEnc	Gate3[0].Chan[0].PhaseCapt.a	転流位置フィードバックにパルスエンコーダデータを使用します。
*12	Motor[1].PhasePosSf	2048/ (256*2000*4/5)	転流位置 (角度) のスケール係数(Sf)を設定します。今回のように 32 ビットの EncTable[1]に 24 ビットのパルスエンコーダデータ (Gate3[0].Chan[0].ServoCapt) を割り付けて使用する場合、スケール係数は以下の式で計算します。 $\text{Sf} = \frac{2048 \times \text{モータ極ペア数}}{256 \times \text{Encoder 分解能} \times \text{Encoder 通倍設定}}$ 使用する機器の仕様に応じて設定してください。 今回使用している機器のパラメータは以下です。 256: $2^{(32 \text{ ビット} - 24 \text{ ビット})} = 256$ Encoder 分解能: 2000pulse/rotate Encoder 通倍設定: 4 通倍 モータ極ペア数: 5 ペア (10 極)

No.	設定項目	設定値	説明
*13	Motor[1].PwmSf	13458	PWM 出力のスケール係数を設定します。フルレンジは 16384 です。PWM 波形が 0%または 100%のデューティサイクルに達しないように、通常、フルレンジの 95%未満に設定します。今回は 82%程度に設定しています。 サーボドライバの仕様に応じて設定してください。
*14	Motor[1].PhaseOffset	683	三相モータでは+683 または-683 に設定します。
*15	Motor[1].AmpFaultLevel	1	AMP Fault 検出の論理を指定します。サーボドライバの仕様に応じて設定してください。 0: 負論理で AMP Fault を検出します。 1: 正論理で AMP Fault を検出します。
*16	Motor[1].WarnFeLimit	4000	位置偏差がこの値を超えるとステータスビット Motor[1].AmpWarn がセットされます。 今回はモータ半回転の値に設定しています。使用するアプリケーションに応じて設定してください。
*17	Motor[1].FatalFeLimit	8000	位置偏差がこの値を超えるとモータが kill され、ステータスビット Motor[1].FeFatal がセットされます。 今回はモータ半回転の値に設定しています。使用するアプリケーションに応じて設定してください。
*18	Motor[1].CurrentNullPeriod	0	設定値 1 の場合、フェーズサーチ時 [Motor[1].PhaseFindingStep=1] に [Motor[1].IaBias] と [Motor[1].IbBias] が自動で設定されます。
*19	Motor[1].pEnc	EncTable[1].a	モータの位置ループフィードバックに使用するアドレスに、パルスエンコーダを指定します。 今回は Note.*2 で EncTable[1]にパルスエンコーダデータを割り付けています。
*20	Motor[1].pEnc2	EncTable[1].a	モータの速度ループフィードバックに使用するアドレスに、パルスエンコーダを指定します。 今回は Note.*2 で EncTable[1]にパルスエンコーダデータを割り付けています。
*21	Motor[1].pDac	Gate3[0].Chan[0].Pwm[0].a	モータ指令出力レジスタに DirectPWM を割り付けます。
*22	Motor[1].pAdc	Gate3[0].Chan[0].AdcAmp[0].a	デジタル電流フィードバックに使用する AD コンバータに、DirectPWM インタフェースの AD コンバータを指定します。
*23	Motor[1].AdcMask	\$FFFF0000	実電流値として使用されるのが、32 ビットの電流フィードバックワードのどのビットなのかを指定します。 今回は 16 ビットの AD コンバータに設定しています。サーボドライバの仕様に応じて設定してください。
*24	Motor[1].MaxDac	28377*3.33/16.25	瞬時電流の制限値(RMS)を設定します。サーボドライバとモータを比較して小さい方を使用してください。今回はモータが小さい方になります。 以下の式で計算します。 $\text{MaxDac} = \frac{\cos(30^\circ) \times 32767 \times \text{最大瞬時電流}}{\text{Servo driver ADC フルレンジ電流}}$ 各パラメータは使用する機器に応じて決定してください。 今回使用している機器のパラメータは以下です。 形 R88M-K05030T の最大瞬時電流: $4.7\text{A(p-p)} / \sqrt{2} = 3.33\text{A(RMS)}$ 形 CK3A-G310L の ADC フルレンジ電流: 16.25A(RMS)

No.	設定項目	設定値	説明
*25	Motor[1].I2tSet	28377*1.1/16.25	定格電流の制限値(RMS)を設定します。サーボドライバとモータを比較して小さい方を使用してください。今回はモータが小さい方になります。 以下の式で計算します。 $I2tSet = \frac{\cos(30^\circ) \times 32767 \times \text{定格電流}}{\text{Servo driver ADC フルレンジ電流}}$ 各パラメータは使用する機器に応じて決定してください。 今回使用している機器のパラメータは以下です。 形 R88M-K05030T の定格電流: 1.1A(RMS) 形 CK3A-G310L の ADC フルレンジ電流: 16.25A(RMS)
*26	Motor[1].I2tTrip	$(\text{Motor}[1].\text{MaxDac} \times \text{Motor}[1].\text{MaxDac} - \text{Motor}[1].\text{I2tSet} \times \text{Motor}[1].\text{I2tSet}) \times 3$	モータの積分電流制限を設定します。以下の式で計算します。 $I2tTrip = (\text{MaxDAC}^2 - I2tSet^2) \times \text{許容時間 (秒)}$ 形 R88M-K05030T の許容時間: 3 秒
*27	Motor[1].AbsPhasePosOffset	400	有効なフェーズサーチであるとみなす最小の動作を指定します。理想は転流サイクル(2048)の 1/4=512(90°)ですが、摩擦などにより動作できないことを考慮して、今回は約 80%の値に設定しています。 フェーズサーチ時 [Motor[1].PhaseFindingStep=1] の移動がこれより小さな動作となる場合、Power PMAC はフェーズサーチが失敗したとみなします。
*28	Motor[1].PhaseFindingDac	400	フェーズサーチでモータの各相へ出力される相電流の大きさを設定します。使用する機器に応じて調整してください。
*29	Motor[1].PhaseFindingTime	1000	フェーズサーチでの各ステップの継続時間を設定します。使用する機器に応じて調整してください。 今回の設定では以下の継続時間となります。 $\text{継続時間} = \text{Servo 周期} \times \text{Motor}[1].\text{PhaseFindingTime} = 1\text{ms} \times 1000 = 1000\text{ms}$
*30	Motor[1].PowerOnMode	0	1: フェーズサーチ後、モータをイネーブルにします。 0: フェーズサーチ後、モータは Kill されます。

3-5 設定の確認

以下の手順に従い、これまでの設定が正しいことを確認します。

1

Terminal から
[Motor[1].PhaseFindingStep=1] コマンドをタイプし、フェーズサーチを実行します。

- [Motor[1].PhaseFindingStep] の値が 1、6、7、0 と遷移します。フェーズサーチが成功すると [Motor[1].ClosedLoop] と [Motor[1].PhaseFound] の値が 0 から 1 になります。また、[Motor[1].New[0].Pos] の値が [Motor[1].AbsPhasePosOffset] の設定値よりも大きくなります。このとき、AMP ENAB 0 の LED が点灯します。*1

Terminal: Online [192.168.0.200 : SSH]

Motor[1].PhaseFindingStep=1

Motor[1].PhaseFindingStep=1

Watch Window

Command/Query	Response
Sys.ServoCount	644992
Motor[1].PhaseFindingStep	0
Motor[1].ClosedLoop	1
Motor[1].PhaseFound	1
Motor[1].New[0].Pos	467.199999999999989

2

Terminal から [#1 out1] コマンドをタイプします。

Terminal: Online [192.168.0.200 : SSH]

\$\$\$

Resetting PowerPMAC

PowerPMAC Reset complete

#1 out1

#1 out1

3

モータが回転していることを確認します。また、Position ウィンドウの [Position] の値が正方向に増加していることを確認します。

- [#1 out1] コマンドをタイプしてもモータが回転しない場合は、[#1 out2]、[#1 out3] と数値を徐々に大きくしてください。

Position

Position

#1	2,176,411.25 mu
#2	0.00 mu
#3	0.00 mu
#4	0.00 mu

4

Terminal から [kill] コマンドをタイプし、モータを停止させます。

Terminal: Online [192.168.0.200 : SSH]

#1 out3

#1 out4

#1 out5

#1 out6

kill

kill

*1. [Motor[1].PhaseFound] が 1 にならない場合はフェーズサーチが失敗しています。設定値が適切かを確認してください。
フェーズサーチが失敗した場合の設定値の調整例を、以下に示します。

- フェーズサーチ後に [Motor[1].New[0].Pos] の値が [Motor[1].AbsPhasePosOffset] の設定値より小さい場合は、[Motor[1].PhaseFindingDac] の値を大きくしてください。また、[Motor[1].PhasePosSf] の設定値が適切であることを確認してください。
- フェーズサーチ中に I2tFault のステータス異常が発生する場合は、[Motor[1].PhaseFindingDac] や [Motor[1].PhaseFindingTime] の値を小さくしてください。
- フェーズサーチ後に [Motor[1].New[0].Pos] の値が-になる場合は、[Motor[1].PhasePosSf] と [EncTable[1].ScaleFactor] の符号を変更してください。

3-6 モータのチューニング

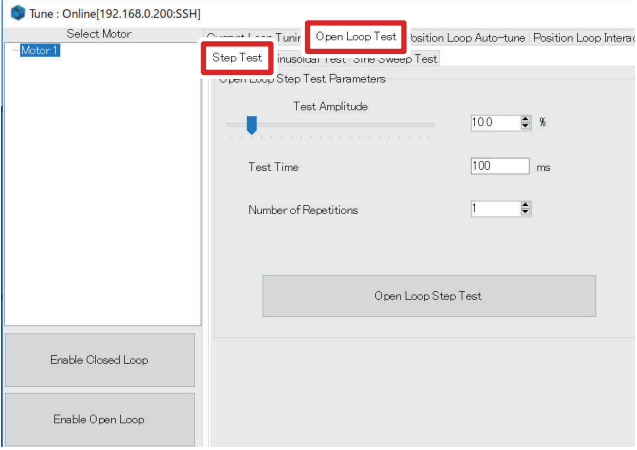
以下の手順に従い、Power PMAC IDE のチューニングツールを使用してモータのチューニングを行います。

3-6-1 オープンループテスト

以下の手順に従い、オープンループでモータを動作させ、各種設定が正しいことを確認します。

1

[Delta Tau] の [Tools] メニューから [Advanced Tuning] を選択し、Tune 画面を開き、[Open Loop Test] の [Step Test] を選択します。



2

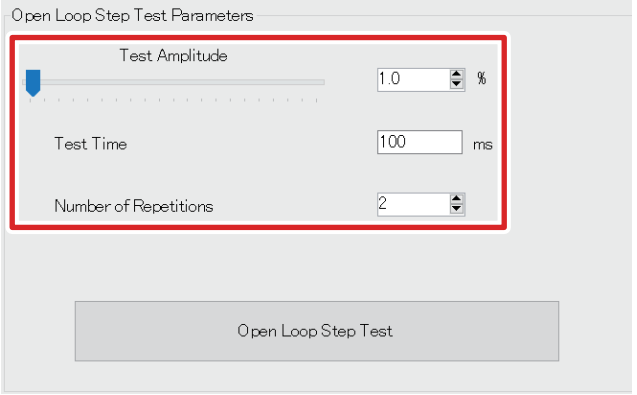
次のチューニングパラメータを設定します。

[Test Amplitude] : 1.0%*1

[Test Time] : 100ms

[Number of Repetitions] : 2

*1. モータが回転しない場合は大きな値を設定してください。



3

[Open Loop Step Test] ボタンをクリックします。

Step Test Sinusoidal Test Sine Sweep Test

Open Loop Step Test Parameters

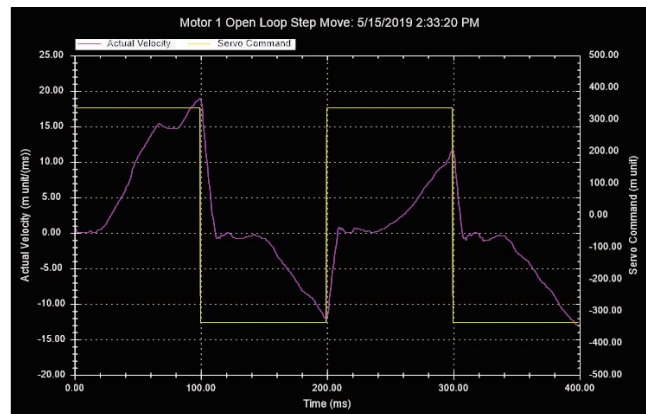
Test Amplitude %

Test Time ms

Number of Repetitions

Open Loop Step Test

- モータが往復動作し、右のようなテスト結果が表示されます。
- モータが回転しない場合は、[Test Amplitude] を大きな値に変更してください。
- 今回のテスト結果は、[Test Amplitude] が 8.0% の場合になります。



3-6-2 電流ループのゲイン設定

以下の手順に従い、電流ループのゲイン設定を行い、目的の応答特性となるように調整します。

1

Tune 画面で [Current Loop Tuning] の [Interactive Tune] を選択します。

Tune: Online 192.168.0.207

Select Motor

Current Loop Tuning Interactive Tune

Auto-tune Parameters

Integral Gain (IiGain)

Forward Path Proportional Gain (IpfGain)

Back Path Proportional Gain (IpbGain)

Phase Current Bias Offsets

Phase A (IaBias)

Phase B (IbBias)

Current Loop Step Parameter Setup

Magnitude bits

Rough Phasing Magnitude bits

Dwell Time ms

Do A Current Loop Step

Full Motor

2

次のパラメータを設定します。
 [IliGain] : 0.0099999998 (デフォルト)
 [IpfGain] : 0
 [IpbGain] : 0
 [Magnitude] : 3000bits
 [Dwell Time] : 50ms

Simple Auto-tune Auto-tune Interactive Tune

Auto-tune Parameters

Integral Gain (IiGain)

Forward Path Proportional Gain (IpfGain)

Back Path Proportional Gain (IpbGain)

Phase Current Bias Offsets

Phase A (IaBias)

Phase B (IbBias)

Current Loop Step Parameter Setup

Magnitude bits

Rough Phasing Magnitude bits

Dwell Time ms

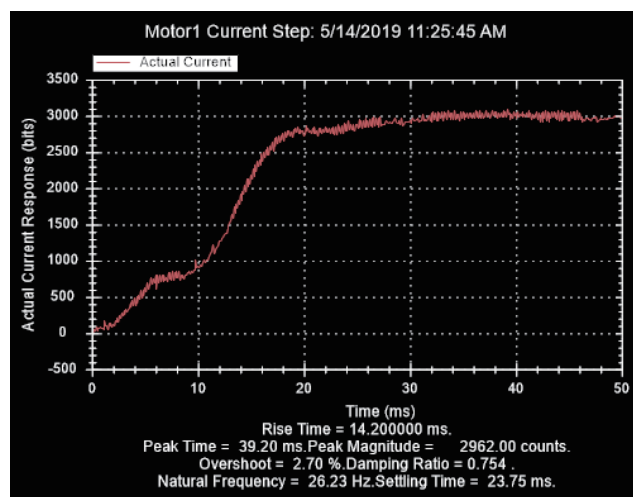
Do A Current Loop Step

Full Motor

3 [Do A Current Loop Step] ボタンをクリックします。

The screenshot shows the 'Auto-tune Parameters' and 'Current Loop Step Parameter Setup' sections. The 'Do A Current Loop Step' button is highlighted with a red rectangle.

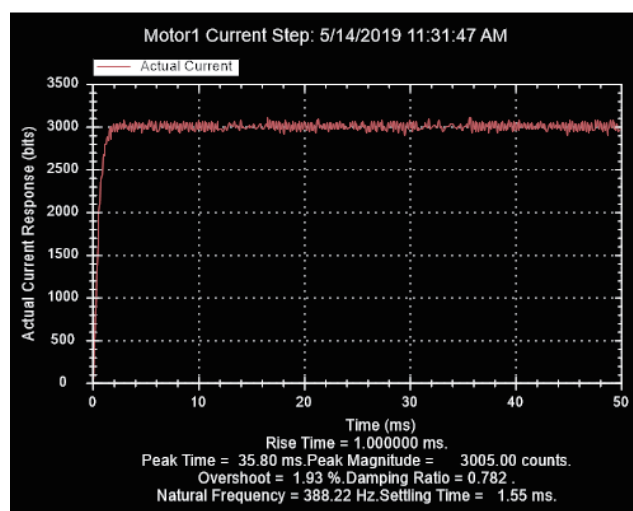
- 電流のステップ応答が表示されます。



4 目的の応答特性が得られるよう、[IliGain]、[lpfGain]、[lpbGain] を調整します。

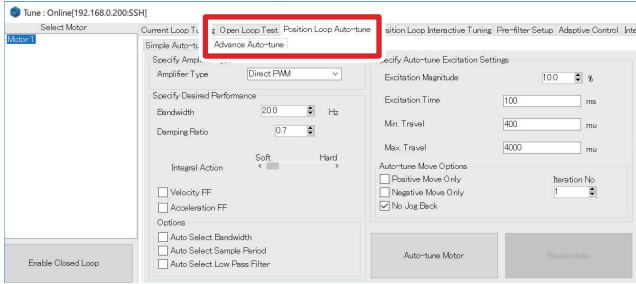
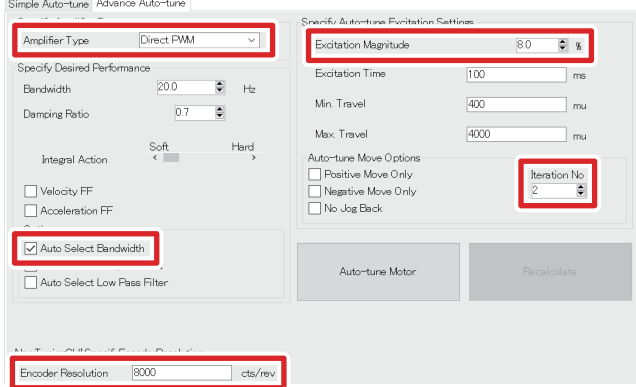
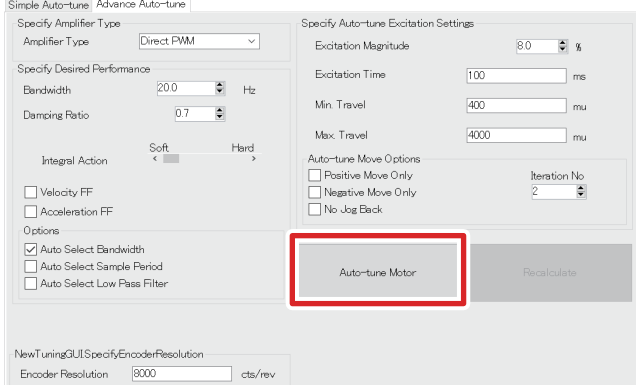
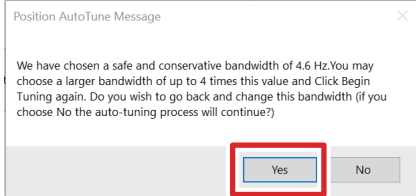
The screenshot shows the 'Auto-tune Parameters' section. The 'IliGain' field is highlighted with a red rectangle.

- 立ち上がりの応答が遅い場合は [IliGain] を大きくします。
- オーバーシュートや振動が大きい場合は [lpfGain] や [lpbGain] を大きくします。
- 各ゲインは小さい値から徐々に大きくしてください。



3-6-3 バンド幅のオート設定

以下の手順に従い、オートチューニング機能を使用して、サーボループのバンド幅を自動的に設定します。

1	Tune 画面で [Position Loop Auto-tune] の [Advance Auto-tune] を選択します。	
2	次のパラメータを設定します。 [Amplifier Type] : Direct PWM [Auto Select Bandwidth] : チェックを入れる [Encoder Resolution] : 8000cts/rev [Excitation Magnitude] : 8.0%^{*1} [Iteration No] : 2 ^{*1}「3-6-1 オープンループテスト (P.3-19)」の手順3のオープンループで回転した値を選択してください。 • [Encoder Resolution] はモータ 1 回転あたりのパルスカウント数を設定してください。 今回は 2000pulse/rotate のパルスエンコーダを 4 通倍の設定でカウントしているので、[Encoder Resolution] を 8000 としています。	
3	[Auto-tune Motor] ボタンをクリックします。	
4	右のメッセージが表示されたら、[Yes] ボタンをクリックします。	

- 5 右の画面が表示されたら、[Implement] ボタンをクリックします。

- 6 [Current Gains] に [Recommended Gains] の値が反映されていることを確認し、[OK] ボタンをクリックします。

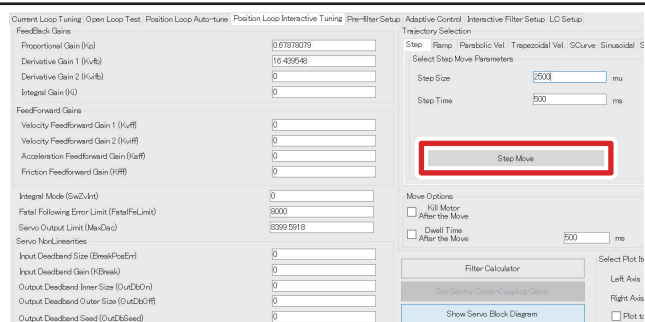
3-6-4 バンド幅の手動設定

以下の手順に従い、ステップ応答特性を観測しながら、より適切なバンド幅に設定します。

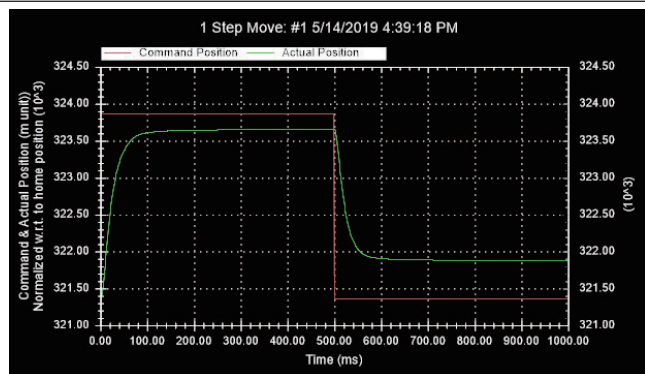
- 1 Tune 画面で [Position Loop Interactive Tuning] の [Step] を選択します。

- 2 次のパラメータを設定します。
[Step Size] : 2500mu

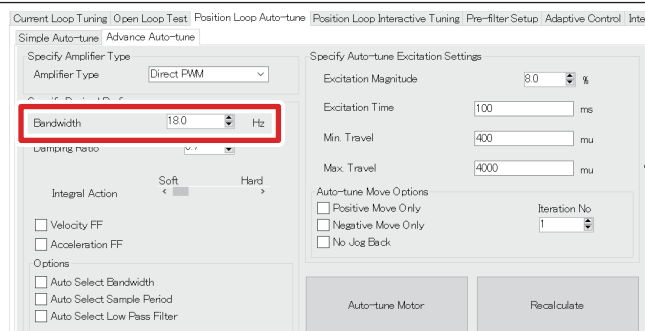
3 [Step Move] ボタンをクリックします。



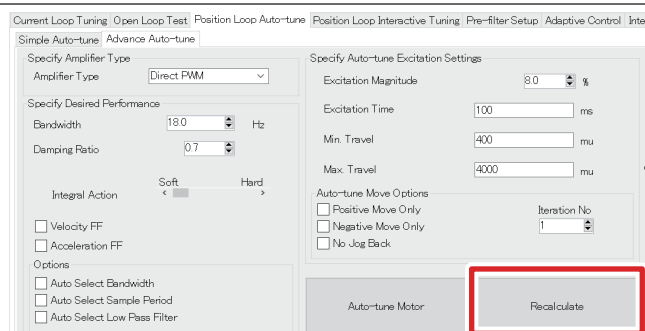
4 ステップ応答特性を確認します。



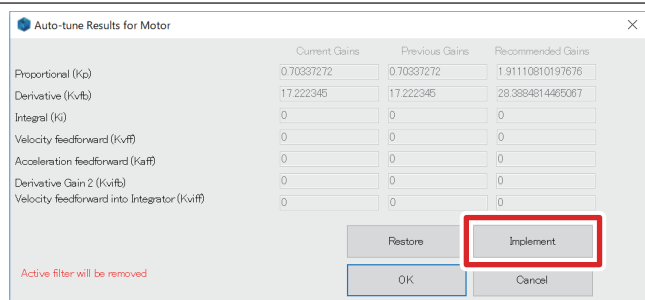
5 目標位置に達していない場合は、[Advance Auto-tune] 画面に戻り、[Bandwidth] をさらに大きな値に設定します。



6 [Recalculate] ボタンをクリックします。



7 右の画面が表示されたら、[Implement] ボタンをクリックします。



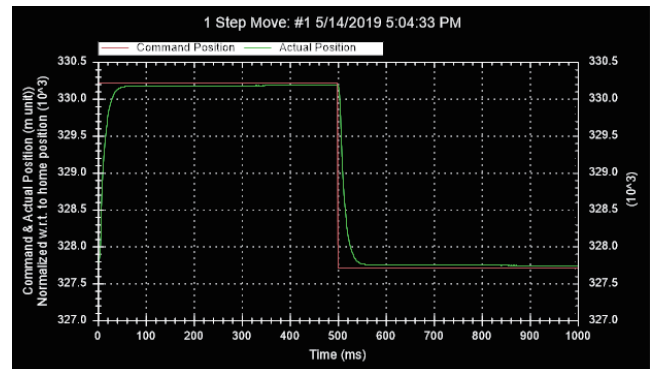
- 8 [Current Gains] に [Recommended Gains] の値が反映されていることを確認し、[OK] ボタンをクリックします。

	Current Gains	Previous Gains	Recommended Gains
Proportional (Kp)	1.91110810197676	0.70337272	1.91110810197676
Derivative (Kv/b)	28.3884814465067	17.222345	28.3884814465067
Integral (Ki)	0	0	0
Velocity feedforward (Kvff)	0	0	0
Acceleration feedforward (Kaff)	0	0	0
Derivative Gain 2 (Kv/b)	0	0	0
Velocity feedforward into Integrator (Kvff)	0	0	0

Active filter will be removed

Restore Implement OK Cancel

- 9 手順 1 に戻り、目的の応答特性が得られるまで操作を繰り返します。



3-6-5 フィードフォワード値の設定

以下の手順に従い、ステップ応答特性を観測しながら、より適切なバンド幅に設定します。

- 1 Tune 画面で [Position Loop Auto-tune] の [Advance Auto-tune] を選択し、[Velocity FF] と [Acceleration FF] にチェックを入れます。

- 2 [Recalculate] ボタンをクリックします。

- 3 右の画面が表示されたら、[Implement] ボタンをクリックします。

Auto-tune Results for Motor

	Current Gains	Previous Gains	Recommended Gains
Proportional (Kp)	1.9111081	1.9111081	1.91110810197676
Derivative (Kvfb)	28.388481	28.388481	28.3884814465067
Integral (Ki)	0	0	0
Velocity feedforward (Kvff)	0	0	28.3884814465067
Acceleration feedforward (Kaff)	0	0	215.150816623052
Derivative Gain 2 (Kvfb2)	0	0	0
Velocity feedforward into integrator (Kviff)	0	0	0

Restore Implement OK Cancel

Active filter will be removed

- 4 [Current Gains] に [Recommended Gains] の値が反映されていることを確認し、[OK] ボタンをクリックします。

Auto-tune Results for Motor

	Current Gains	Previous Gains	Recommended Gains
Proportional (Kp)	1.91110810197676	1.9111081	1.91110810197676
Derivative (Kvfb)	28.3884814465067	28.388481	28.3884814465067
Integral (Ki)	0	0	0
Velocity feedforward (Kvff)	28.3884814465067		28.3884814465067
Acceleration feedforward (Kaff)	215.150816623052		215.150816623052
Derivative Gain 2 (Kvfb2)	0	0	0
Velocity feedforward into integrator (Kviff)	0	0	0

Restore Implement OK Cancel

Active filter will be removed

- 5 [Position Loop Interactive Tuning] の [Parabolic Vel.] を選択し、次のパラメータを設定します。
[Move Size] : 2500mu
[Move Time] : 500ms
[Left Axis] : Velocity
[Right Axis] : Following Error

Position Loop Interactive Tuning

Step: Parabolic Vel. Move Size: 2500 Move Time: 500

Left Axis: Velocity Right Axis: Following Error

Parabolic Velocity Move

Filter Calculator

Get Servo Open-Loop Gains

Show Servo Block Diagram

- 6 [Parabolic Velocity Move] ボタンをクリックします。

Position Loop Interactive Tuning

Step: Parabolic Vel. Move Size: 2500 Move Time: 500

Left Axis: Velocity Right Axis: Following Error

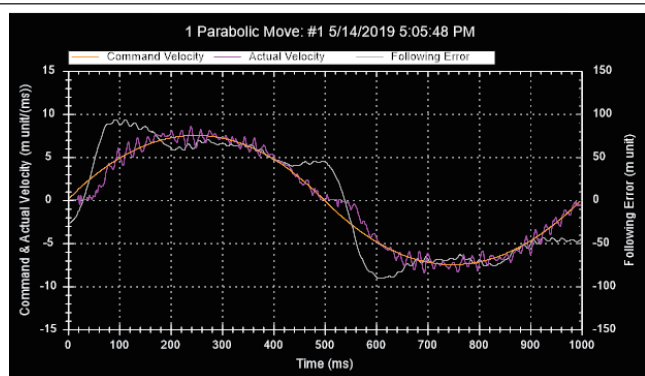
Parabolic Velocity Move

Filter Calculator

Get Servo Open-Loop Gains

Show Servo Block Diagram

- 7 速度の放物線応答特性を確認します。



8

9

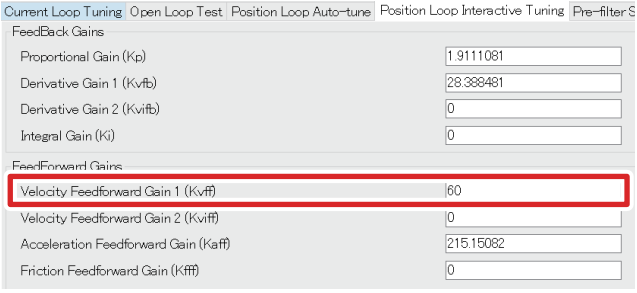
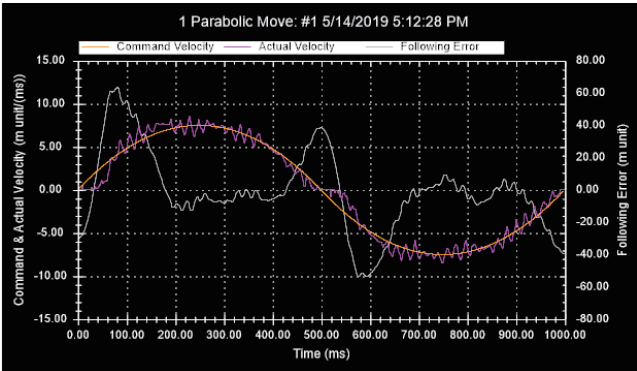
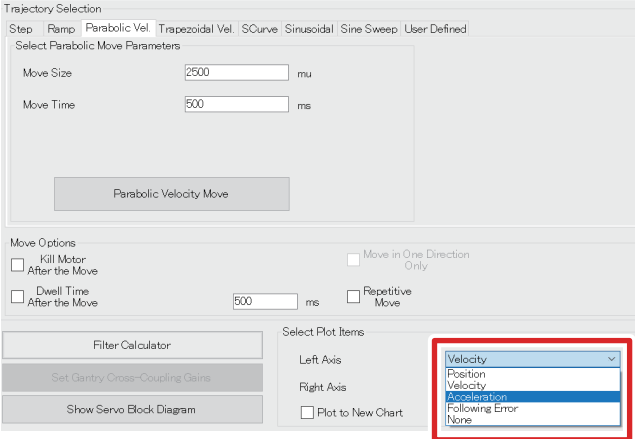
10

[Following Error] が速度に対して正相関を有する場合は [Kvff] を大きくします。逆相関を有する場合は [Kvff] を小さくします。

再び [Parabolic Velocity Move] ボタンをクリックします。

- これを [Following Error] が速度に対して相関がなくなるまで繰り返します。

同様に、[Following Error] が [Acceleration] (加速度)、[Position] (位置) などに相関を持つ場合は、[Kaff]、[Kfff] の値を増減させます。

3-6-6 動作確認とチューニングパラメータのプロジェクト化

以下の手順に従って動作確認を行い、チューニングしたパラメータのプロジェクト化を行います。

1

Terminal から
[Motor[1].PhaseFindingStep=1] コマンドをタイプし、フェーズサーチを実行します。

- [Motor[1].PhaseFindingStep] の値が 1、6、7、0 と遷移します。
- フェーズサーチが成功すると [Motor[1].ClosedLoop] と [Motor[1].PhaseFound] の値が 0 から 1 になります。また、[Motor[1].New[0].Pos] の値が [Motor[1].AbsPhasePosOffset] の設定値よりも大きくなります。このとき、AMP ENAB 0 の LED が点灯します。

Terminal: Online [192.168.0.200 : SSH]

Motor[1].PhaseFindingStep=1

Watch Window

Command/Query	Response
Sys.ServoCount	4088603
Motor[1].PhaseFindingStep	0
Motor[1].ClosedLoop	1
Motor[1].PhaseFound	1
Motor[1].New[0].Pos	469.759999999999991

2

Terminal から [#1 j+] コマンドをタイプします。

Terminal: Online [192.168.0.200 : SSH]

Motor[1].PhaseFindingStep=1

#1 j+

3

モータが回転していることを確認します。また、Position ウィンドウで #1 の [Velocity] の値が +32 付近であることを確認します。

- [Velocity] は [Motor[1].JogSpeed] (デフォルト: 32) の値で決まります。

Position

	Position	Velocity
#1	10,608,759.57 mu	31.32 mu/msec
#2	0.00 mu	0.00 mu/msec

4

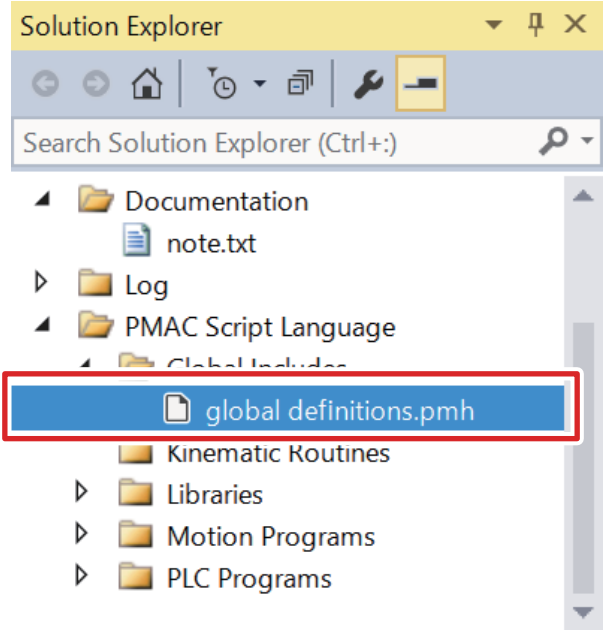
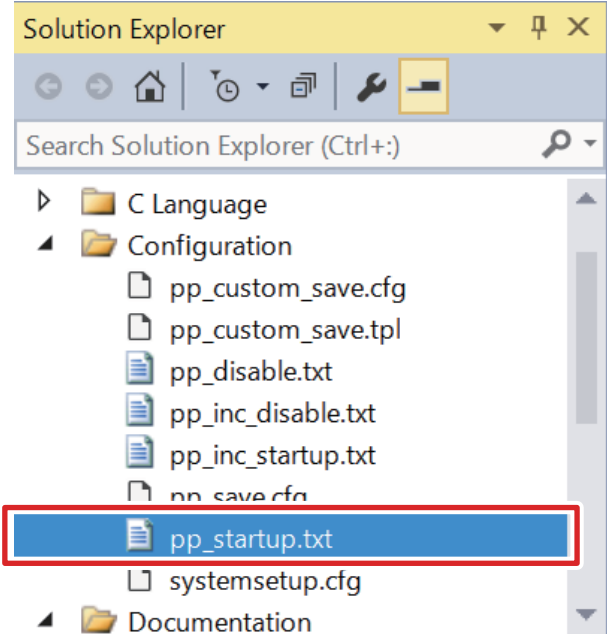
Terminal から [kill] コマンドをタイプし、モータを停止させます。

Terminal: Online [192.168.0.200 : SSH]

Motor[1].PhaseFindingStep=1

#1 j+

kill

5	Solution Explorer の [PMAC Script Language] - [Global Includes] の下にある global definitions.pmh を開きます。	
6	チューニングにより得たゲインを global definitions.pmh に追記します。	Motor[1]. liGain = *** Motor[1]. lpfGain = *** Motor[1]. lpbGain = *** Motor[1]. Servo. Kp = *** Motor[1]. Servo. Kvfb = *** Motor[1]. Servo. Kaff = *** Motor[1]. Servo. Kvff = ***
7	Solution Explorer の [Configuration] の下にある pp_startup.txt を開きます。	
8	右に示すフェーズサーチ実行コマンドを記載します。	Motor[1]. PhaseFindingStep = 1

9

プロジェクトをビルド&ダウンロードします。

- ビルド&ダウンロード方法については、「3-4 コントローラの各種設定 (P.3-8)」の手順 3~6 を参照してください。
- 手順 5、6 に示すように global definitions.pmh にゲインを記述することで、ゲインをプログラムとして PMAC にダウンロードできます。
- 手順 7、8 に示すように pp_startup.txt にフェーズサーチ実行コマンドを記述することで、電源投入後またはリセット後に自動的にフェーズサーチを実行し、Motor[1]を有効にします。

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー

製品に関するお問い合わせ先



0120-919-066

携帯電話・IP電話などではご利用いただけませんので、右記の電話番号へおかけください。

055-982-5015
(通話料がかかります)

受付時間：9:00～19:00 (12/31～1/3を除く)

クイック オムロン



オムロンFAクイックチャット

www.fa.omron.co.jp/contact/tech/chat/

技術相談員にチャットでお問い合わせいただけます。(I-Webメンバーズ限定)

受付時間：平日9:00～12:00 / 13:00～17:00 (土日祝日・年末年始・当社休業日を除く)

※受付時間、営業日は変更の可能性がございます。最新情報はリンク先をご確認ください。



その他のお問い合わせ：納期・価格・サンプル・仕様書は貴社のお取引先、または貴社担当オムロン販売員にご相談ください。オムロン制御機器販売店やオムロン販売拠点は、Webページでご案内しています。



オムロン制御機器の最新情報をご覧ください。緊急時のご購入にもご利用ください。 www.fa.omron.co.jp

本誌には主に機種のご選定に必要な内容を掲載しており、ご使用上の注意事項等を掲載していない製品も含まれています。

本誌に注意事項等の掲載のない製品につきましては、ユーザーズマニュアル掲載のご使用上の注意事項等、ご使用の際に必要な内容を必ずお読みください。

- 本誌に記載の標準価格はあくまで参考であり、確定されたユーザ購入価格を表示したものではありません。本誌に記載の標準価格には消費税が含まれておりません。
- 本誌にオープン価格の記載がある商品については、標準価格を決めていません。
- 本誌に記載されているアプリケーション事例は参考用ですので、ご採用に際しては機器・装置の機能や安全性をご確認の上、ご使用ください。
- 本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娯楽機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途に使用される際には、当社の意図した特別な商品用途の場合や特別の合意がある場合を除き、当社は当社商品に対して一切保証をいたしません。
- 本製品の内、外国為替及び外国貿易法に定める輸出許可、承認対象貨物（又は技術）に該当するものを輸出（又は非居住者に提供）する場合は同法に基づく輸出許可、承認（又は役務取引許可）が必要です。
- 規格認証/適合対象機種などの最新情報につきましては、当社Webサイト (www.fa.omron.co.jp) の「規格認証/適合」をご覧ください。

オムロン商品のご用命は

カタログ番号 SBCE-467B

2023年1月現在

©OMRON Corporation 2019 - 2023 All Rights Reserved.

お断りなく仕様などを変更することがありますのでご了承ください