

PCL6125 評価ボード

PCL6125-EB

取扱説明書

サンプルプログラム

PCL6125-EB_KIT Project Sample x

Status	X axis	Y axis
<input type="checkbox"/> X <input type="checkbox"/> Y	MSTSW 0008 h	MSTSW 0408 h
<input type="checkbox"/> Operation starts	SSTSW 0010 h	SSTSW 0010 h
<input type="checkbox"/> Operation direction	RIRQ 00000000 h <input type="button" value="set"/>	RIRQ 00000000 h <input type="button" value="set"/>
<input type="checkbox"/> Error interrupt occurs	RSTS 00001800 h	RSTS 00001800 h
<input type="checkbox"/> Event interrupt occurs	REST 00000000 h <input type="button" value="clear"/>	REST 00000000 h <input type="button" value="clear"/>
<input type="checkbox"/> During acceleration	RIST 00000000 h <input type="button" value="clear"/>	RIST 00000000 h <input type="button" value="clear"/>
<input type="checkbox"/> During deceleration	RCUN1 00000000 h	RCUN1 00001B00 h
<input type="checkbox"/> Constant speed	RSPD 00000000 h	RSPD 00000000 h
<input type="checkbox"/> ALM signal on	RPLS 00000000 h	RPLS 00000000 h
<input type="checkbox"/> +EL signal on	<input type="button" value="CCW"/> <input type="button" value="CW"/>	<input type="button" value="CCW"/> <input type="button" value="CW"/>
<input type="checkbox"/> -EL signal on	<input type="button" value="XY CCW"/> <input type="button" value="XY CW"/>	<input type="button" value="Software reset"/> <input type="button" value="Close"/>
<input type="checkbox"/> ORG signal on		
<input type="checkbox"/> SD signal on		

目次

1. はじめに.....	1
1.1 本書の取扱い.....	1
1.2 注意事項.....	1
1.3 お願い.....	1
2. 紹介.....	2
2.1 動作環境.....	2
3. サンプルプログラムの構成.....	3
3.1 フォルダ構成.....	3
3.2 ファイル構成.....	3
3.3 デバイスドライバーのインストール.....	4
4. C#でのプロジェクト起動.....	4
5. 動作説明.....	5
5.1 プログラムの起動.....	5
5.2 ステータス情報の表示.....	5
5.3 レジスタ情報の表示.....	6
5.4 動作ボタン.....	6
5.4.1 CCW.....	6
5.4.2 CW.....	6
5.4.3 set.....	6
5.4.4 clear.....	6
5.4.5 XY CCW.....	6
5.4.6 XY CW.....	7
5.4.7 Software reset.....	7
5.4.8 Close.....	7
6. ソースコード説明.....	7
6.1 初期設定.....	7
6.2 CCW 動作.....	8
6.3 CW 動作.....	8
6.4 set 動作 および clear 動作.....	8
6.5 Software reset 動作.....	8
6.6 PCL6125 へのアクセス関数.....	9
6.6.1 ステータスの読出し関数 (Read_STATUS).....	9
6.6.2 レジスタの読出し関数 (Read_REG).....	10
6.6.3 レジスタへの書き込み関数 (Write_REG).....	10
6.6.4 動作コマンドの書き込み関数 (Write_COM).....	11

6.6.5 PCL6125-EB ボードへの送信関数 (SendUsb).....	11
6.6.6 PCL6125-EB ボードからのデータ受信関数(GetUsb).....	12

1. はじめに

本取扱説明書は PCL6125 評価ボード(PCL6125-EB)用サンプルプログラム(PCL6125-EB_ApplicationSample_V110.zip)の仕様、機能、及び使用方法等を記載しています。

本製品を使用していただくため、必ず本書をお読みいただき、保管してください。

1.1 本書の取扱い

- ① 本書の全部又は一部を無断で転載することは、著作権法によって禁止されています。
- ② 本書の内容については、性能や品質の向上に伴い、将来予告なく変更することがあります。
- ③ 本書の内容については、万全を期しておりますが、万一不可解な点や誤り、並びに記載もれ等お気づきの点がありましたら、弊社営業担当へ連絡をお願いいたします。

1.2 注意事項

本書は、製品に含まれる機能詳細を説明するものであり、お客様の特定目的に適合することを保証するものではありません。

また、本書に記載されている応用例、回路図等は参考用です。

機器・装置の機能や安全性を確認の上、使用してください。

1.3 お願い

本製品は、原則として、次のいずれかの用途には、使用しないでください。使用する場合は、必ず弊社営業担当へ連絡をお願いいたします。

- ① 原子力設備、電力やガス等の供給システム、交通機関、車両設備、各種安全装置、医療機器等の高い信頼性と安全性が必要とされる設備
- ② 人命や財産に直接、危険を及ぼす可能性がある設備
- ③ カタログ、取扱説明書等に記載のない条件や環境での使用

本製品の故障により、人命や財産に重大な損害を及ぼす可能性のある用途では、冗長設計等により、高い信頼性と安全性を確保して、使用してください。

2. 紹介

本書は制御基板を操作するアプリケーションソフトウェアの取扱説明書です。

本ソフトウェアは、PCL6125-EB 評価ボードを利用することでパルスコントロール LSI PCL6125 を使用したモーター制御機能を学習することができます。

別途弊社の取扱説明書(下記に記載)と併せてご覧ください。

(x は版数)

	取扱説明書名【概要】	文書ファイル名	対象ソフトウェア名	文書番号
ハードウェア 取扱説明書	PCL6125評価ボード(PCL6125-EB) 取扱説明書(ハードウェア)	PCL6125-EB_ HardwareManual_VerxJ.pdf		TA600038-JPx/x
	PCL6125-EB評価キット (PCL6125-EB_KIT)取扱説明書	PCL6125-EB_KIT Manual_VerxJ.pdf		TA600071-JPx/x
アプリケーション ソフトウェア 取扱説明書	PCL6125評価ボード(PCL6125-EB) 取扱説明書 (アプリケーションソフトウェア) 【加減速パターンの設定と全レジスタの 表示】	PCL6125-EB_ ApplicationManual_VerxJ.pdf	PCL6125-EB_ Application_Vxxx JE.zip	TA600039-JPx/x
	PCL6125評価ボード(PCL6125-EB) 取扱説明書 (サンプルプログラム)	PCL6125-EB_ ApplicationSampleManual_ VerxJ.pdf	PCL6125-EB_ ApplicationSampl e_VxxxJE.zip	TA600075-JPx/x (本書)
モーション パターンビルダー 取扱説明書	PCL6125評価ボード(PCL6125-EB) 取扱説明書 (モーションパターンビルダー アプリケーションソフトウェア) 【フローチャートにて視覚的に軸制御を 行う機能説明】	PCL6125-EB_ MotionBuilderManual_VerxJ. pdf	PCL6125-EB_ MotionBuilder_Vx xxJE.zip	TA600040-JPx/x
参考資料	PCL6115/6125/6145 ユーザズマニュアル			DA70152-0/x

アプリケーションソフト及び関係資料は、NPM ウェブサイトよりダウンロードしてください。

2.1 動作環境

本ソフトウェアは、Windows7、およびWindows10(共に 32 bit と 64 bit)での動作確認を行っています。

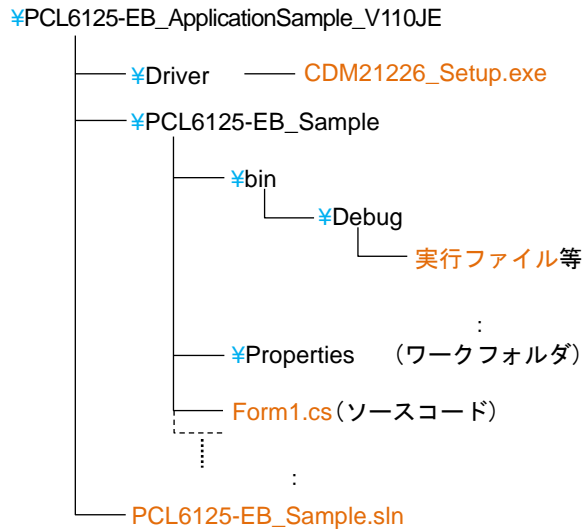
(上記以外の OS については動作確認を行っていません。)

また動作中に OS がスリープモードへ移行しないように省電力設定を変更してください。

3. サンプルプログラムの構成

3.1 フォルダ構成

圧縮ファイル(PCL6125-EB_ApplicationSample_V110JE.zip)を解凍するとサンプルプログラムは下記のようなフォルダ構成になっています。



3.2 ファイル構成

<※PCL6125-EB_ApplicationSample_V110JE フォルダ内>

PCL6125-EB_Sample.sln	ソリューションファイル
-----------------------	-------	-------------

<※PCL6125-EB_ApplicationSample_V110JE※Driver フォルダ内>

CDM21226_Setup.exe	デバイスドライバーのインストーラー(FTDI 社製)
--------------------	-------	----------------------------

<※PCL6125-EB_ApplicationSample_V110JE※PCL6125-EB_Sample フォルダ内>

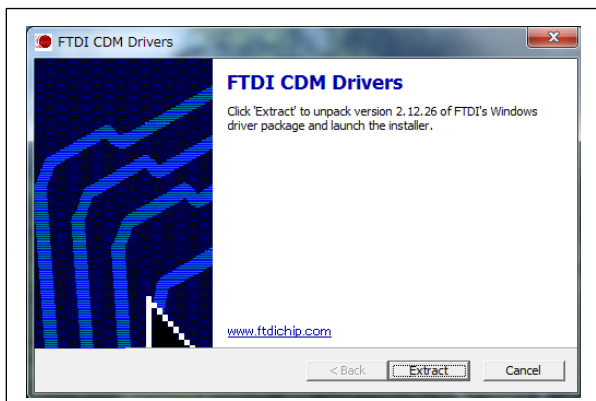
Form1.cs	ソースコード
clsFTDI.cs	FTDI アクセス関数
accessPCL.cs	PCL6125 アクセス関数
FTD2XX_NET.dll	FTDI ライブラリ
FTD2XX_NET.xml	FTDI XML ドキュメント
*.bmp	画像データ
その他	ワークファイル類(実行時には不要)

<※PCL6125-EB_ApplicationSample_V110JE※PCL6125-EB_Sample※bin※Debug フォルダ内>

PCL6125-EB_Sample.exe	実行ファイル
FTD2XX_NET.dll	FTDI ライブラリ(実行時に必須)
FTD2XX_NET.xml	FTDI XML ドキュメント(実行時には不要)
その他	ワークファイル類(実行時には不要)

3.3 デバイスドライバーのインストール

「CDM21226_Setup.exe」をダブルクリックしてインストーラーを起動し、画面の指示に従ってインストールを完了させてください。ただし、既にインストール済の場合、再度インストールする必要はありません。

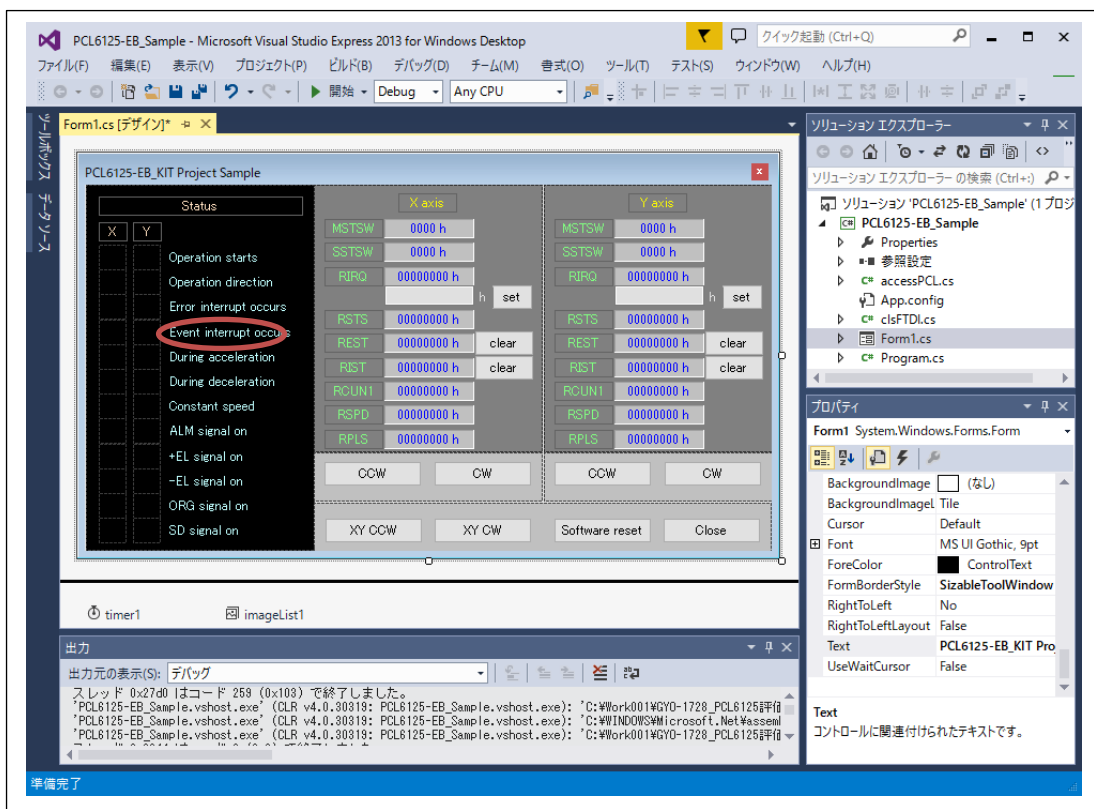


注：FTDI 社の Web サイト(<http://www.ftdichip.com/Drivers/D2XX.htm>)に最新版のデバイスドライバーがある場合、そちらをダウンロードしてご利用ください。

4. C#でのプロジェクト起動

PCL6125-EB 評価キット(PCL6125-EB_KIT)がパソコンに接続されていることを確認してください。

"Microsoft Visual C#" がインストールされていることを確認し、PCL6125-EB_Sample.sln 「ソリューションファイル」をダブルクリックしてください。



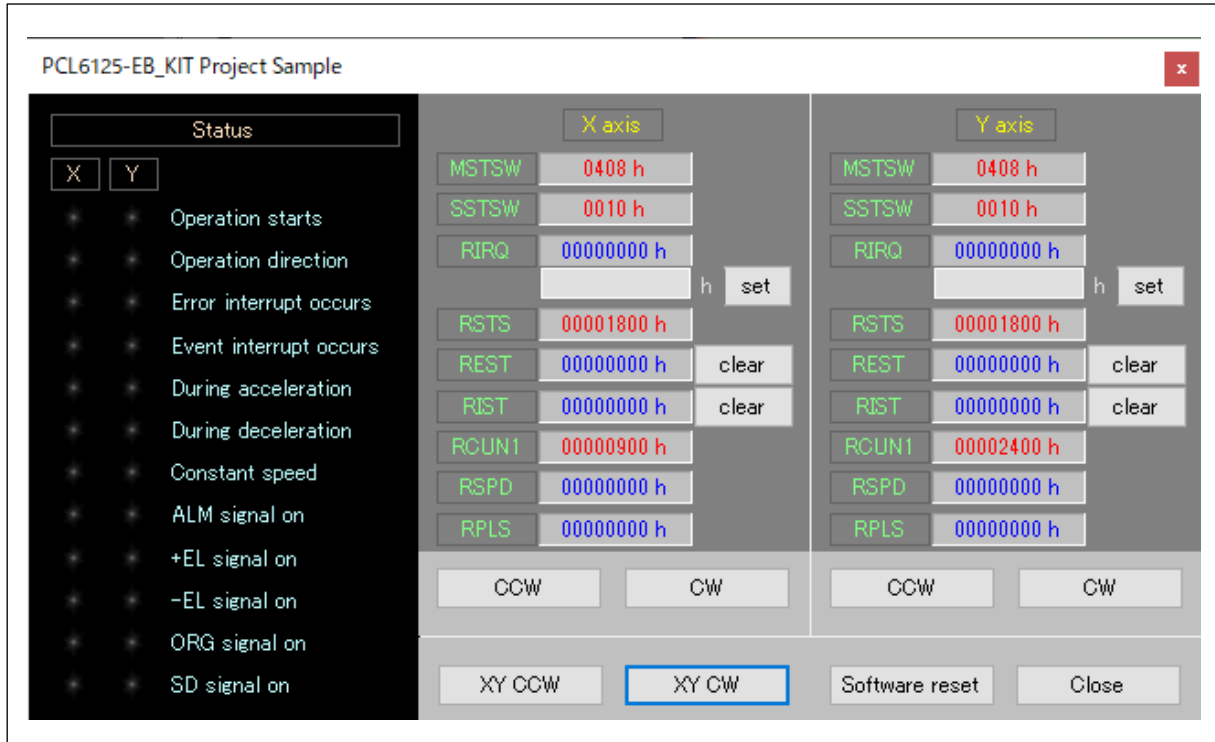
マイクロソフト製品のインストールに関しては、マイクロソフトの Web サイトを参照ください。

プロジェクトのビルドやデバッグについても、その操作方法はマイクロソフトの Web サイトを参照ください。

5. 動作説明

5.1 プログラムの起動

デバッグを開始すると、以下の画面のソフトウェアが起動します。



5.2 ステータス情報の表示

「Status」の欄で、PCL6125のX軸、Y軸の状態を確認できます。詳細は次のとおりです。

表記	内容	名称
Operation starts	動作中に "1"	(MSTSW.SSCM)
Operation direction	CW 動作で "0"、CCW 動作で "1"	(RSTS.SDIR)
Error interrupt occurs	エラー割込みが発生した時に "1"	(MSTSW.SERR)
Event interrupt occurs	イベント割込みが発生した時に "1"	(MSTSW.SINT)
During acceleration	加速中に "1"	(SSTS.SFU)
During deceleration	減速中に "1"	(SSTS.SFD)
Constant speed	定速中に "1"	(SSTS.SFC)
ALM signal on	ALM 入力が入力 ON の時に "1"	(SSTS.SALM)
+EL signal on	+EL 入力が入力 ON の時に "1"	(SSTS.SPEL)
-EL signal on	-EL 入力が入力 ON の時に "1"	(SSTS.SMEL)
ORG signal on	ORG 入力が入力 ON の時に "1"	(SSTS.SORG)
SD signal on	SD 入力が入力 ON の時に "1"	(SSTS.SSD)

状態が "1" になると、各項目横の表示色が変わります。

5.3 レジスタ情報の表示

X軸、Y軸のステータスと、いくつかのレジスタの内容が表示されています。詳細は次のとおりです。

表記	内容
MSTSW	メインステータス
SSTSW	サブステータス
RIRQ	イベント割込み要因設定レジスタ
RSTS	拡張ステータス
REST	エラー割込み要因レジスタ
RIST	イベント割込み要因レジスタ
RCUN1	カウンター1
RSPD	現在速度モニター
RPLS	位置決めカウンター取得レジスタ

すべて 16 進数で表示されており、値がゼロのときは青、ゼロ以外のときは赤で表示されます。

各レジスタの値(ビット)については、PCL6115/6125/6145 ユーザーズマニュアルを参照してください。

5.4 動作ボタン

5.4.1 CCW

X軸の「CCW」をクリックすると、X軸がCCW方向への動作が開始され、パルスを2304回出力して停止します。

動作開始後、約1秒で1ppsから1024ppsまで加速し、しばらく1024ppsで動作した後、約1秒かけて減速停止します。

動作中にクリックしても、無視されます。

Y軸の「CCW」をクリックすると、Y軸がCCW方向へ、X軸動作内容と同様です。

5.4.2 CW

X軸の「CW」をクリックすると、X軸がCW方向への動作が開始されます。

Y軸の「CW」をクリックすると、Y軸がCW方向への動作が開始されます。

動作内容はCCWと同様です。

5.4.3 set

RIRQ(イベント割込み要因設定レジスタ)への値を設定できます。

設定したい値は、"set"ボタンの横に16進数で入力し、"set"ボタンをクリックしてください。

5.4.4 clear

REST(エラー割込み要因レジスタ)、RIST(イベント割込み要因レジスタ)の状態をクリアできます。

各レジスタ横のボタンで、対応するレジスタの値をゼロクリアします。

5.4.5 XY CCW

「XY CCW」をクリックすると、X軸、Y軸が同時にCCW方向への動作が開始され、パルスを2304回出力して停止します。

動作開始後、約1秒で1ppsから1024ppsまで加速し、しばらく1024ppsで動作した後、約1秒かけて減速停止します。

5.4.6 XY CW

「XY CW」をクリックすると、X 軸、Y 軸が同時に CW 方向への動作が開始されます。

動作内容はXY CCW と同様です。

5.4.7 Software reset

PCL6125 をリセットします。動作(クロック出力)中に「Software reset」をクリックすると、すぐに動作が停止します。

5.4.8 Close

本ソフトウェアを終了します。位置決めモードで動作(クロック出力)中に「Close」をクリックすると、サンプルプログラムは終了しますが、移動量分動作して停止します。

6. ソースコード説明

ソースファイルは「Form1.cs」です。

お客様が試したい動作に修正することで、操作手順の確認を行ってみてください。

PCL6115/6125/6145 ユーザーズマニュアルを参照してください。

6.1 初期設定

初期設定は「InitSet」関数に記述されており、ソフトウェアの起動時と、ソフトウェアリセット実行後に呼び出されます。

初期設定として以下の操作が行われています。

操作	内容
PRMG = 0x000004AF	倍率を 1 倍に設定
PRMD = 0x00000041	動作モードを次のように設定 位置決めモード 直線加減速 スローダウンポイント自動設定
RENV1 = 0x00000002	環境設定 1 を次のように設定 出力パルス仕様を "010" に設定 OUT 端子から負論理でパルスを出力 DIR 端子からプラス方向時に Low を出力
RENV2 = 0x80000055	環境設定 2 を次のように設定 汎用ポート 0~3 を出力ポートに設定 REST および RIST レジスタの読出し自動リセットを解除 <div style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px; margin-top: 10px;"> <p>本ソフトウェアは定期的に全てのレジスタをリードし、表示しています。REST および RIST もリードしているため、読出し自動リセット機能が働く場合、エラー発生時であっても定期的なリードでフラグが消えてしまい、目視確認ができない場合があります。これを防止するため、RENV2.MRST = 1 としています。</p> </div>

6.2 CCW 動作

CCW 動作を行っているのは、ソースファイル内の「Exec_XCCW」, 「Exec_YCCW」, 「Exec_XYCCW」関数です。

関数内では以下の操作が行われています。

操作	内容
PRFL = 0x00000001	スタート速度として "1h" を設定
PRFH = 0x00000400	動作速度として "400h"(1024) を設定
PRUR = 0x00002588	加速レートとして "2588h" を設定
PRMV = 0xFFFFF700	移動量として "FFFFF700h"(-2304) を設定
STAUD	高速スタート2 を実行

スタートコマンドの実行により、約1秒で1ppsから1024ppsまで加速し、しばらく1024ppsで動作した後、約1秒かけて減速停止します。動作中に出力されるパルス数は、マイナス方向に2304パルスです。

6.3 CW 動作

CW 動作を行っているのは、ソースファイル内の「Exec_XCW」, 「Exec_YCW」, 「Exec_XYCW」関数です。

関数内では以下の操作が行われています。

操作	内容
PRFL = 0x00000001	スタート速度として "1h" を設定
PRFH = 0x00000400	動作速度として "400h"(1024) を設定
PRUR = 0x00002588	加速レートとして "2588h" を設定
PRMV = 0x00000900	移動量として "900h"(+2304) を設定
STAUD	高速スタート2 を実行

スタートコマンドの実行により、約1秒で1ppsから1024ppsまで加速し、しばらく1024ppsで動作した後、約1秒かけて減速停止します。動作中に出力されるパルス数は、プラス方向に2304パルスです。

6.4 set 動作 および clear 動作

RIRQ への値設定および REST、RIST のクリア動作を行っているのは、ソースファイル内の「Exec_WriteReg」関数です。

本ソフトウェアは、RENV2 レジスタの bit31 を "1" に設定してあります。この場合の REST や RIST のクリアは、クリアしたいビット部分に "1" を書き込むことで行われます。

REST や RIST からリードした値をそのまま書込めばゼロクリアが行われます。

6.5 Software reset 動作

ソフトウェアリセット動作を行っているのは、ソースファイル内の「Exec_SoftReset」関数です。

関数内では以下の操作が行われています。

操作	内容
SRST	ソフトウェアリセットコマンド実行

コマンド実行後、「InitSet」関数を実行して初期設定を行っています。

6.6 PCL6125 へのアクセス関数

PCL6125-EB ボードへのアクセスは USB 経由で行われます。

このため PCL6125 へ与えるコマンドは、一度 PC 側のプログラム内のバッファにためてから送信します。

また PCL6125 からデータを読み出す場合、複数の読出しコマンドの結果を一度に USB 経由で受信します。

本サンプルソースでは、コマンドをバッファに溜め込む関数と、送信、受信するための関数を用意してあります。

またコマンドをため込むバッファは、本サンプルソース内では以下のように定義しています。

バッファ名 : FtBuff

サイズ : 1024 バイト

6.6.1 ステータスの読出し関数 (Read_STATUS)

PCL6125 のメインステータスを読み出すコマンド、またはサブステータスと汎用ポートの状態を読み出すコマンドを、バッファへ格納します。

このコマンドにより、PCL6125-EB 側で 4 バイトのデータが読出されます。このデータは「GetUSB」関数で受信してください。

Read_STATUS (ref byte[] FtBuff, ref int FtIndex, int subS)	
FtBuff	PCL6125-EB ボードに対するメインステータス(またはサブステータス)の読出しコマンドを格納するためのバッファを指定してください。 バッファのサイズは 1024 バイトですが、サイズを超えたか否かの判断は行っていませんので、注意が必要です。
FtIndex	配列変数の使用数を管理している変数を指定してください。
subS	リードする対象を指定します。 0 : メインステータスをリード 0 以外 : サブステータスをリード
コマンド設定時のバッファ使用数	14 バイト(FtIndex に加算されます)
コマンド実行後に受信すべきデータ数	4 バイト
受信データの順番	1 : X 軸メインステータス bit7~bit0 (またはポート) 2 : X 軸メインステータス bit15~bit8 (またはサブステータス) 3 : Y 軸メインステータス bit7~bit0 (またはポート) 4 : Y 軸メインステータス bit15~bit8 (またはサブステータス)

6.6.2 レジスタの読出し関数 (Read_REG)

PCL6125 のレジスタからデータを読み出すコマンドを、バッファへ格納します。

このコマンドにより、PCL6125-EB 側で 8 バイトのデータが読出されます。このデータは「GetUSB」関数で受信してください。

レジスタ長が 32bit に満たないレジスタを読み出した場合でも、必ず 4 バイトの値が読み出されます。このときの上位ビットはゼロで満たされています。

Read_REG (ref byte[] FtBuff, ref int FtIndex, byte comm)	
FtBuff	PCL6125-EB ボードに対するレジスタからデータの読出しコマンドを格納するためのバッファを指定してください。 バッファのサイズは 1024 バイトですが、サイズを超えたか否かの判断は行っていないので、注意が必要です。
FtIndex	配列変数の使用数を管理している変数を指定してください。
comm	PCL6125 のレジスタ読出しコマンドを指定してください。
コマンド設定時のバッファ使用数	19 バイト(FtIndex に加算されます)
コマンド実行後に受信すべきデータ数	8 バイト
受信データの順番	1 : X 軸レジスタ値 bit7~bit0 2 : X 軸レジスタ値 bit15~bit8 3 : X 軸レジスタ値 bit23~bit16 4 : X 軸レジスタ値 bit31~bit24 5 : Y 軸レジスタ値 bit7~bit0 6 : Y 軸レジスタ値 bit15~bit8 7 : Y 軸レジスタ値 bit23~bit16 8 : Y 軸レジスタ値 bit31~bit24

6.6.3 レジスタへの書き込み関数 (Write_REG)

PCL6125 のレジスタへデータを書込むコマンドと書き込みデータを、バッファへ格納します。

このコマンドによる PCL6125-EB 側の読出しデータはありません。

Write_REG (ref byte[] FtBuff, ref int FtIndex, uint RegD, byte Jsc, byte comm)	
FtBuff	PCL6125-EB ボードに対するレジスタへデータの書き込みコマンドを格納するためのバッファを指定してください。 バッファのサイズは 1024 バイトですが、サイズを超えたか否かの判断は行っていないので、注意が必要です。
FtIndex	配列変数の使用数を管理している変数を指定してください。
RegD	レジスタへ書込みたいデータを指定してください。
Jsc	常にゼロを設定してください。
comm	PCL6125 のレジスタ書き込みコマンドを指定してください。
コマンド設定時のバッファ使用数	15 バイト(FtIndex に加算されます)
コマンド実行後に受信すべきデータ数	0 バイト
受信データの順番	—

6.6.4 動作コマンドの書き込み関数 (Write_COM)

PCL6125 の動作コマンドを、バッファへ格納します。

このコマンドによる PCL6125-EB 側の読出しデータはありません。

Write_COM (ref byte[] FtBuff, ref int FtIndex, byte Jsc, byte comm)	
FtBuff	PCL6125-EB ボードに対する動作コマンドの書き込みコマンドを格納するためのバッファを指定してください。 バッファのサイズは 1024 バイトですが、サイズを超えたか否かの判断は行っていないので、注意が必要です。
FtIndex	配列変数の使用数を管理している変数を指定してください。
Jsc	X 軸「0x01」、Y 軸「0x02」、X-Y 軸「0x03」を設定してください。
comm	PCL6125 の動作コマンドを指定してください。
コマンド設定時のバッファ使用数	11 バイト(FtIndex に加算されます)
コマンド実行後に受信すべきデータ数	0 バイト
受信データの順番	—

【例】

次の順番でレジスタへのデータ、動作コマンドを格納し、実行します。

```

FtIndex=0;
hAPCL.Write_REG(ref FtBuff, ref FtIndex, 0x00000001, 0x01, 0x81); // write PRFL (X-Axis)
hAPCL.Write_REG(ref FtBuff, ref FtIndex, 0x00000400, 0x01, 0x82); // write PRFH (X-Axis)
hAPCL.Write_REG(ref FtBuff, ref FtIndex, 0x00002588, 0x01, 0x83); // write PRUR (X-Axis)
hAPCL.Write_REG(ref FtBuff, ref FtIndex, 0xFFFFF700, 0x01, 0x80); // write PRMV (X-Axis)
//
hAPCL.Write_COM(ref FtBuff, ref FtIndex, 0x01, 0x53); // write High speed start 2 (X-Axis)
//
ftStatus = hAPCL.SendUsb(ref FtBuff, ref FtIndex);

```

6.6.5 PCL6125-EB ボードへの送信関数 (SendUsb)

バッファに格納されているコマンド類を、PCL6125-EB に送信します。

送信後、コマンド類は PCL6125-EB ボード側で実行されます。

SendUsb (ref byte[] FtBuff, ref int FtIndex)	
FtBuff	「Read_STATUS」、「Read_REG」、「Write_REG」、「Write_COM」等の関数でコマンド群を格納したバッファを指定してください。
FtIndex	バッファの使用数を指定してください。 送信後、この変数はゼロにクリアされます。

6.6.6 PCL6125-EB ボードからのデータ受信関数(GetUsb)

PCL6125-EB ボード側で読み出しコマンドを実行すると、読みだされたデータが PCL6125-EB ボード側の送信バッファにたまり、このバッファの内容を PC 側へ送信させます。

GetUsb (ref byte[] FtBuff)	
FtBuff	実行した読み出しコマンドの実行結果の順番で、読みだされたデータが格納されます。

【例】

次の順番でコマンドを格納し、実行したと想定します。

```
FtIndex=0;
Read_STATUS(ref FtBuff, ref FtIndex, 0); // メインステータスリード
Read_REG(ref FtBuff, ref FtIndex, 0xE3); // RCUN1 レジスタリード
Read_STATUS(ref FtBuff, ref FtIndex, 1); // サブステータスリード
//
SendUsb(ref FtBuff, ref FtIndex);
GetUsb(ref FtBuff);
```

受信後のバッファは次のようになります。

FtBuff		
0	ステータス bit7~bit0	// X 軸メインステータスリード
1	ステータス bit15~bit8	
2	ステータス bit7~bit0	// Y 軸メインステータスリード
3	ステータス bit15~bit8	
4	RCUN1 レジスタ bit7~bit0	// X 軸 RCUN1 レジスタリード
5	RCUN1 レジスタ bit15~bit8	
6	RCUN1 レジスタ bit23~bit16	
7	RCUN1 レジスタ bit31~bit24	
8	RCUN1 レジスタ bit7~bit0	// Y 軸 RCUN1 レジスタリード
9	RCUN1 レジスタ bit15~bit8	
10	RCUN1 レジスタ bit23~bit16	
11	RCUN1 レジスタ bit31~bit24	
12	汎用ポートの状態	// X 軸サブステータスリード
13	サブステータス	
14	汎用ポートの状態	// Y 軸サブステータスリード
15	サブステータス	

PC 側のソフトは、上記順番を意識して読みだしてください。

弊社は、弊社ソフトウェアについて著作権を含む一切の知的所有権を保持します。弊社は、弊社ソフトウェアに関するいかなる権利もお客様に譲渡しません。お客様は、弊社の製品を使用する目的でのみ、現状有姿の弊社ソフトウェアを使用することができます。弊社は、弊社ソフトウェアの完全性、正確性、適用性、有用性、第三者知財の非侵害性を含め、明示たると黙示たるとを問わず何らの保証をいたしません。また、弊社ソフトウェアを使用したことで生じる損害（収入または利益の逸失を含む）について、一切の責任を負いません。お客様が、購入国以外で弊社ソフトウェアを使用する場合は、購入国と使用国の輸出管理法や規制を遵守する必要があります。

改訂履歴

版数	日付	内容
初版	2020年12月17日	新規作成
2版	2021年6月15日	ソフトウェアバージョンV1.10 画面修正 Operation is start → Operation starts Event interrupt occurs → Event interrupt occurs ソフトウェア免責事項追加

NPM 顧客「満足」から「感動」へ。
日本パルスモーター株式会社

www.pulsemotor.com

お問い合わせ

www.pulsemotor.com/support

東京 電話 03(3813)8841 FAX 03(3813)8550

大阪 電話 06(6576)8330 FAX 06(6576)8335

お電話受付時間 平日 9:00~17:00