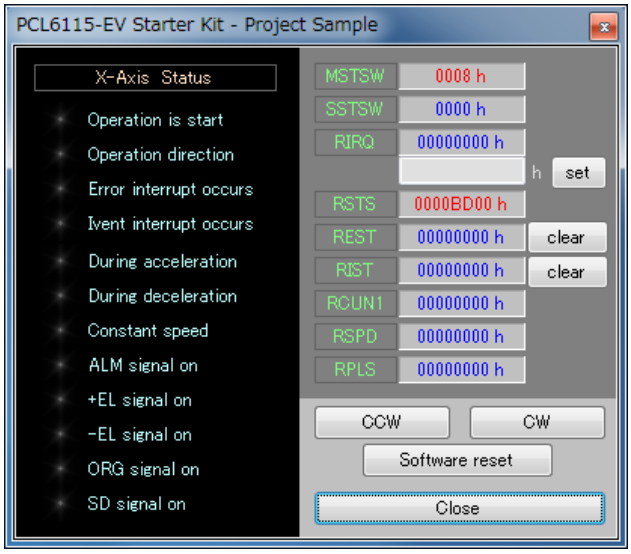
PCL6115 スターターキット

PCL6115-EV

取扱説明書

サンプルプログラム



NPM 顧客「満足」から「感動」へ。
日本パルスモーター株式会社

目 次

NPM

1. はじめに

PCL6115-EV スターターキットをご検討いただき、ありがとうございます。
本書は PCL6115-EV スターターキットを利用することでパルスコントロールＬＳＩ PCL6115 を使用したモータ制御機能を学習することができます。
本ソフトウェアのソースコードを、お客様独自の制御内容に修正などを行ないながら、ソフトウェア作成の参考としてご活用ください。

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	対象ソフトファイル名	文書番号
ハードウェア取扱説明書	PCL6115スターターキット 取扱説明書 （ハードウェア）	PCL6115-EV _HardwareManual_VerxJ.pdf	ー	TA600021-JPx/x
	PCL6115スターターキット 取扱説明書 （簡易版）	PCL6115-EV_ SimpleManual_VerxJE.pdf	ー	TA600020-JPx/x
アプリケーションソフトウェア取扱説明書	PCL6115スターターキット 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】	PCL6115-EV _ApplicationManual_VerxJ.pdf	PCL6115-EV_Application_VxxxJEzip	TA600018-JPx/x
	PCL6115スターターキット 取扱説明書 （言語ファイル作成ルール） 【多言語化】	PCL6115-EV _ApplicationLanguageFile Manual_VerxJ.pdf	PCL6115-EV_ApplicationLanguageFile_VxxxE.zip	TA600007-JPx/x
	PCL6115スターターキット 取扱説明書 （サンプルプログラム） 【開発環境上での確認と追加】	PCL6115-EV _ApplicationSampleManual_VerxJ.pdf	PCL6115-EV_ApplicationSample_VxxxJ.zip	TA600022-JPx/x （本書）

(x は版数)

	取扱説明書名【概要】	[文書ファイル名]	対象ソフトファイル名	文書番号
モーション パターン ビルダー 取扱 説明書	PCL6115 スターターキット 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】	PCL6115-EV _MotionBuilderManual_VerxJ .pdf	PCL6115-EV_Motion Builder_VxxxJE.zip	TA600023-JPx/x
	PCL6115 スターターキット 取扱説明書 （モーションパターンビルダー言語ファイル作成ルール） 【モーションパターンビルダーでの多言語化】	PCL6115-EV _MotionBuilder LanguageFileManual_VerxJ. pdf	PCL6115-EV_Motion BuilderLanguageFile _VxxxJ.zip	TA600008-JPx/x
	PCL6115 スターターキット 取扱説明書 （モーションパターンビルダーサンプルプロジェクト） 【モーションパターンビルダーで作成した動作パターンを開発環境上で確認追加】	PCL6115-EV _MotionBuilderSample Manual_VerxJ.pdf	PCL6115-EV_Motion BuilderSample_Vxxx J.zip	TA600024-JPx/x
参考資料	PCL6115/6125/6145 ユーザーズマニュアル		-	DA70152-0/x

サンプルプログラム及び関係資料は、ＮＰＭウエブサイトよりダウンロードしてください。

1-1. 動作環境

本ソフトウェアは、Windows7、および Windows10(共に 32bit と 64bit)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なっておりません。）
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。

1-2. 動作モード

PCL6115 を USB からシリアルバス I/F モードで制御しています。

1-3. 使用したプログラミング言語

マイクロソフト社の以下の製品を使用しています。

Microsoft Visual Studio Express 2013 for Windows Desktop （無償版）

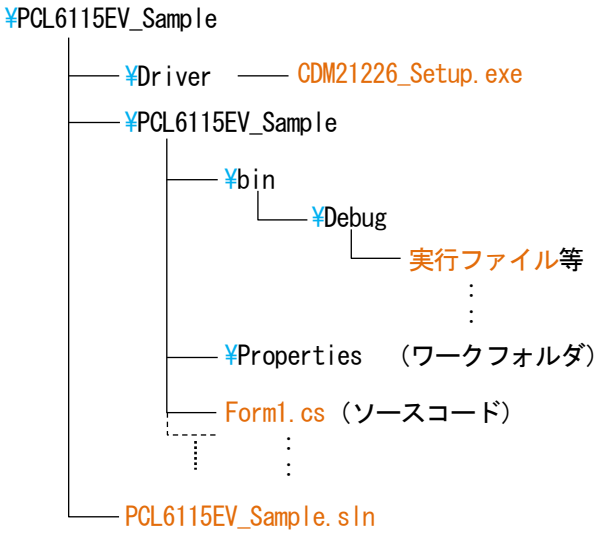
1-4. 注意

・"Microsoft Visual C#" の使用法などに関しては、お答えすることはできません。
・FTDI 社製の製品の使用法などに関しては、お答えすることはできません。
・本サンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、
弊社では一切責任を負いませんのでご了承ください。

2. サンプルプログラムの構成

2-1. フォルダ構成

圧縮ファイル(PCL6115-EV_ApplicationSample_V300JE.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。



2-2. ファイル構成

＜¥PCL6115EV_Sample フォルダ内＞

PCL6115EV_Sample.sln	·····	ソリューションファイル

＜¥PCL6115EV_Sample¥Driver フォルダ内＞

CDM21226_Setup.exe	·····	デバイスドライバのインストーラ（FTDI 社製）

＜¥PCL6115EV_Sample¥PCL6115EV_Sample フォルダ内＞

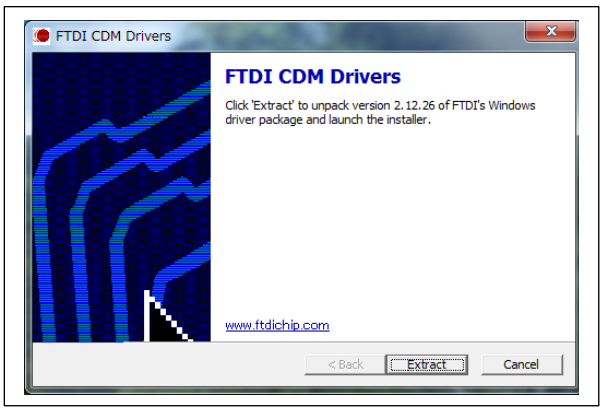
Form1.cs	·····	ソースコード
clsFTDI_Serial.cs	·····	FTDI アクセス関数
accessPCL_Serial.cs	·····	PCL6115 アクセス関数
FTD2XX_NET.dll	·····	FTDI ライブラリ
FTD2XX_NET.xml	·····	FTDI XML ドキュメント
＊.bmp	·····	画像データ
その他		

＜¥PCL6115EV_Sample¥PCL6115EV_Sample¥bin¥Debug フォルダ内＞

PCL6115EV_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. デバイスドライバのインストール

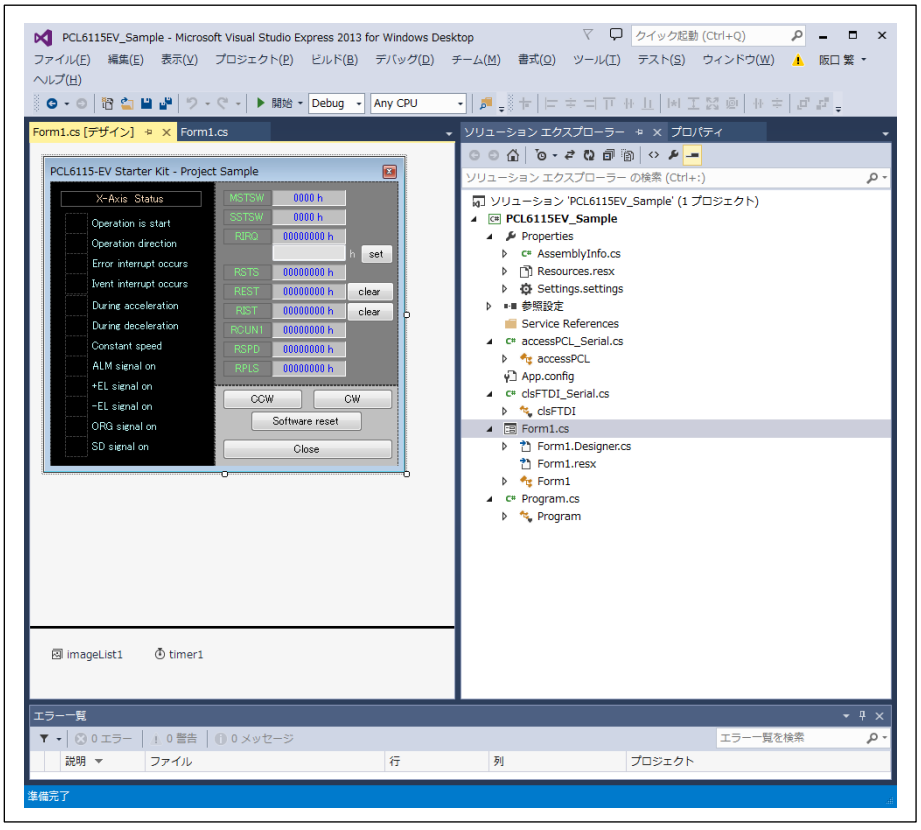
「CDM21226_Setup.exe」をダブルクリックしてインストーラを起動し、画面の指示に従ってインストールを完了させてください。
ただし、既にインストール済の場合、再度インストールする必要はありません。



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

PCL6115-EV 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、PCL6115EV_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。

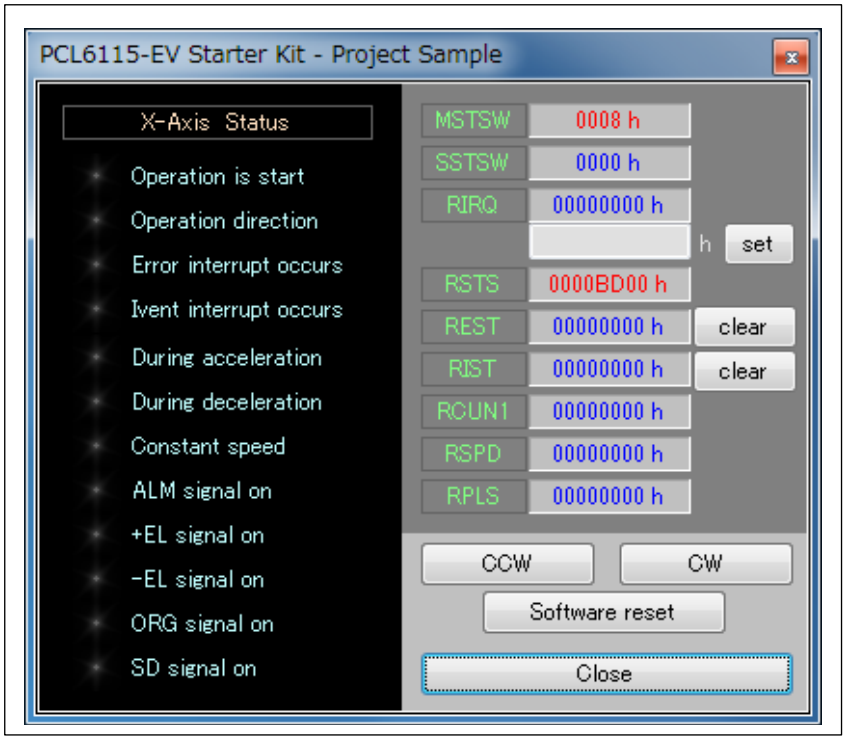


マイクロソフト製品のインストールに関しては、マイクロソフトの Web サイトを参照ください。
プロジェクトのビルドやデバッグに関しても、その操作方法はマイクロソフトの Web サイトを参照ください。

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

「X-Axis Status」の欄で、PCL6115 の X 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
Operation is start	動作中に "1" になります。(MSTSW.SSCM)
Operation direction	CW 動作で "0" 、CCW 動作で "1" になります。(RSTS.SDIR)
Error interrupt occurs	エラー割込みが発生したときに "1" になります。(MSTSW.SERR)
Ivent interrupt occurs	イベント割込みが発生したときに "1" になります。(MSTSW.SINT)
During acceleration	加速中に "1" になります。(SSTSW.SFU)
During deceleration	減速中に "1" になります。(SSTSW.SFD)
Constant speed	定速中に "1" になります。(SSTSW.SFC)
ALM signal on	ALM 入力が ON の時に "1" になります。(SSTSW.SALM)
+EL signal on	+EL 入力が ON の時に "1" になります。(SSTSW.SPEL)
-EL signal on	-EL 入力が ON の時に "1" になります。(SSTSW.SMEL)
ORG signal on	ORG 入力が ON の時に "1" になります。(SSTSW.SORG)
SD signal on	SD 入力が ON の時に "1" になります。(SSTSW.SSD)

状態が "1" になると、各項目横の表示色が変化します。

5-3. レジスタ情報の表示

X 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１。
RSPD	現在速度モニタ。
RPLS	位置決めカウンタ。

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。

5-4. 動作ボタン

5-4-1. CCW

クリックすると、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。

5-4-2. CW

クリックすると、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. Software reset

PCL6115 をリセットします。

5-4-4. 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。

5-4-5. 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。

5-4-6. Close

本ソフトウェアを終了します。

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。

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 = 0x80001D40	環境設定 2 を次のように設定。 汎用ポート 3~7 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。 （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。）

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_CCW」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 = 0xFFFFF830	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_CW」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 = 0x000007D0	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. 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。

6-5. 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。

6-6. PCL6115 へのアクセス関数

PCL6115-EV ボードへのアクセスは USB 経由で行われます。
このため PCL6115 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 PCL6115 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)

PCL6115 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、PCL6115-EV 側で 2 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	PCL6115-EV 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。
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	6 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	2 バイト
受信データの順番	1：メインステータス bit7～bit0 （またはポート） 2：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)

PCL6115 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、PCL6115-EV 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。
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	PCL6115-EV 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	PCL6115 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	20 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 : レジスタ値 bit7～bit0 2 : レジスタ値 bit15～bit8 3 : レジスタ値 bit23～bit16 4 : レジスタ値 bit31～bit24

6-6-3. レジスタへの書き込み関数(Write_REG)

PCL6115 のレジスタへデータを書込むコマンドと書込みデータを、バッファへ格納します。
このコマンドによる PCL6115-EV 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	PCL6115-EV ボードに対するレジスタへデータの書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	PCL6115 のレジスタ書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	24 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)

PCL6115 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 PCL6115-EV 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	PCL6115-EV ボードに対する動作コマンドの書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	常にゼロを設定してください。
comm	PCL6115 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	8 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	ー

6-6-5. PCL6115-EV 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、PCL6115-EV に送信します。
送信後、コマンド類は PCL6115-EV 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. PCL6115-EV ボードからのデータ受信関数(GetUsb)

PCL6115-EV ボード側で読出しコマンドを実行すると、読みだされたデータが PCL6115-EV ボード側の送信バッファにたまります。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	// メインステータスリード
1	ステータス bit15～bit8	
2	RCUN1 レジスタ bit7～bit0	// RCUN1 レジスタリード
3	RCUN1 レジスタ bit15～bit8	
4	RCUN1 レジスタ bit23～bit16	
5	RCUN1 レジスタ bit31～bit24	
6	汎用ポートの状態	// サブステータスリード
7	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。

改訂履歴

版数	日付	内容
初版	2018 年 3 月 19 日	新規作成
第 2 版	2019 年 7 月 16 日	文書番号変更 1. はじめに 取扱説明書リスト追加

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

2019 年 7 月発行