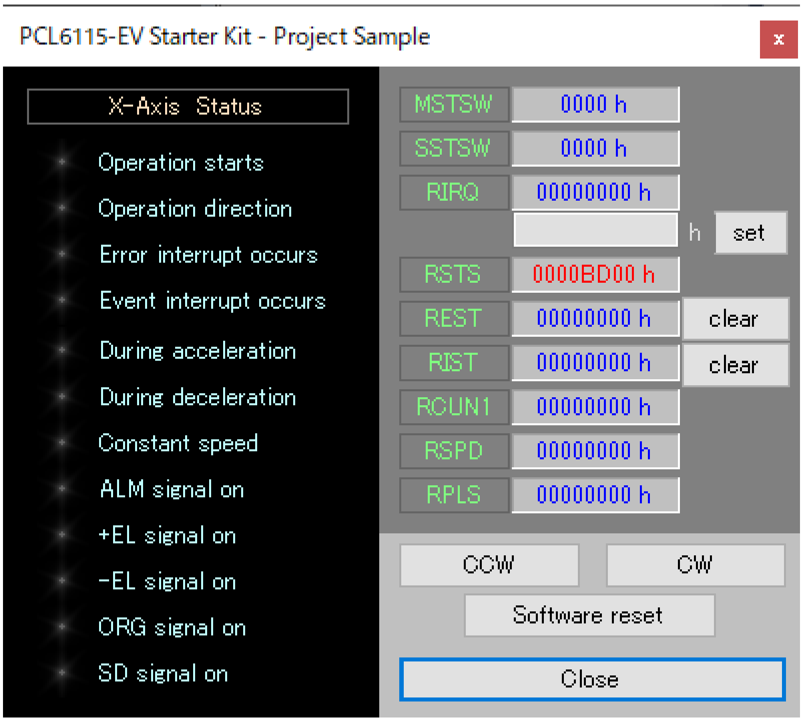
PCL6115 Starter Kit

PCL6115-EV

User’s Manual

Sample Program



PCL6115-EV Starter Kit - Project Sample

X-Axis Status

- Operation starts
- Operation direction
- Error interrupt occurs
- Event interrupt occurs
- During acceleration
- During deceleration
- Constant speed
- ALM signal on
- +EL signal on
- -EL signal on
- ORG signal on
- SD signal on

MSTSW 0000 h
SSTSW 0000 h
RIRQ 00000000 h
RSTS 0000BD00 h
REST 00000000 h
RIST 00000000 h
RCUN1 00000000 h
RSPD 00000000 h
RPLS 00000000 h

Impress, not just satisfy
NPM Nippon Pulse Motor Co., Ltd.

INDEX

1 Introduction

This manual describes the specifications, functions, and usages of the sample program for our PCL6115-EV Starter Kit (PCL6115-EV_ApplicationSample_V320.zip).

Be sure to read this manual thoroughly and keep it handy in order to use the product appropriately.

1.1 How to use this manual

1. Reproduction of this manual in whole or in part without permission is prohibited by the Copyright Act.
2. The contents of this manual are subject to change without the prior notice along with the improvement of performance and quality.
3. Although this manual is produced with the utmost care, please contact our sales representative if there are any questions, errors, or omissions.

1.2 Notice

This document aims to describe the details of functions of the product. It does not warrant fitness for a particular purpose of the customer. Also, the examples of applications and circuit diagrams in this manual are included only for your reference. Please confirm the features and the safety of device or equipment before use.

1.3 Confirmation

Please do not use this product in the following conditions. If you need to use in the following conditions, please contact our sales representatives:

1. Any equipment that may require a high reliability or a safety, such as nuclear facilities, electricity or gas supply systems, transportation facilities, vehicles, various safety systems, medical equipment, etc.
2. Any equipment that may directly affect human survival or property.
3. Usages under conditions or circumstances that are not specified in the catalog, manual, etc.

For applications that may cause serious damages to a human life or property due to failure of this product, ensure high reliability and safety by redundant design.

2 Information

Thank you for considering PCL6115-EV Starter Kit.

By using PCL6115-EV Starter Kit, this manual can learn the motor control function using pulse control LSI PCL6115.

Please make use of the source code of this software as a reference for software creation while modifying etc. to your own control contents.

Please refer to t the following manuals along with this manual.

(x: revision)

	Manual Name [Outline]	Document File name	Software File name	Document No.
Hardware Manual	PCL6115 Starter Kit User's Manual (Hardware)	PCL6115-EV _HardwareManual_VerxE.pdf	―	TA600021-ENx/x (This document)
	PCL6115 Starter Kit User's Manual (Simple Manual)	PCL6115-EV_ SimpleManual_VerxJE.pdf	―	TA600020-ENx/x
Application Software Manual	PCL6115 Starter Kit User's Manual (Application Software) [Setting accel/decel pattern and register display]	PCL6115-EV _ApplicationManual_VerxE.pdf	PCL6115-EV_Application_VxxxJEzip	TA600018-ENx/x
	PCL6115 Starter Kit User's Manual (Language File Creation Rule) [Multi-language]	PCL6115-EV _ApplicationLanguageFile Manual_VerxE.pdf	PCL6115-EV_ApplicationLanguageFile_VxxxE.zip	TA600007-ENx/x
	PCL6115 Starter Kit User's Manual (Sample program) [Check and add motion pattern on development environment]	PCL6115-EV _ApplicationSampleManual_VerxE.pdf	PCL6115-EV_ApplicationSample_VxxxE.zip	TA600022-ENx/x (This document)
Motion Pattern Builder Manual	PCL6115 Starter Kit User's Manual (Motion Pattern Builder Application Software) [To describe function to perform axis control visually with a flowchart]	PCL6115-EV _MotionBuilderManual_VerxE.pdf	PCL6115-EV_MotionBuilder_VxxxJE.zip	TA600023-ENx/x
	PCL6115 Starter Kit User's Manual (Motion Pattern Builder Language File Creation Rule) [Motion Pattern Builder in Multi-language]	PCL6115-EV _MotionBuilder LanguageFileManual_VerxE.pdf	PCL6115-EV_MotionBuilderLanguageFile_VxxxE.zip	TA600008-ENx/x
	PCL6115 Starter Kit User's Manual (Motion Pattern Builder Sample Project) [Check and add motion pattern created by Motion Pattern Builder on development environment]	PCL6115-EV _MotionBuilderSample Manual_VerxE.pdf	PCL6115-EV_MotionBuilderSample_VxxxE.zip	TA600024-ENx/x
Reference	PCL6115/6125/6145 User's Manual		-	DA70152-0/xE

Please download application software and related materials from our NPM website.

2.1 Operating Environment

We have checked the operation of this software with Windows 7 and Windows 10 (both 32-bit and 64-bit).

(We have not checked it with OS other than the above.)

Please change the power saving setting so as not to operate “sleep mode” during operation.

2.2 Operation mode

This software controls PCL6115 in serial bus I/F mode from USB.

2.3 Programming language used

This software uses the following products of Microsoft Corporation.

Microsoft Visual Studio Express 2013 for Windows Desktop (Free)

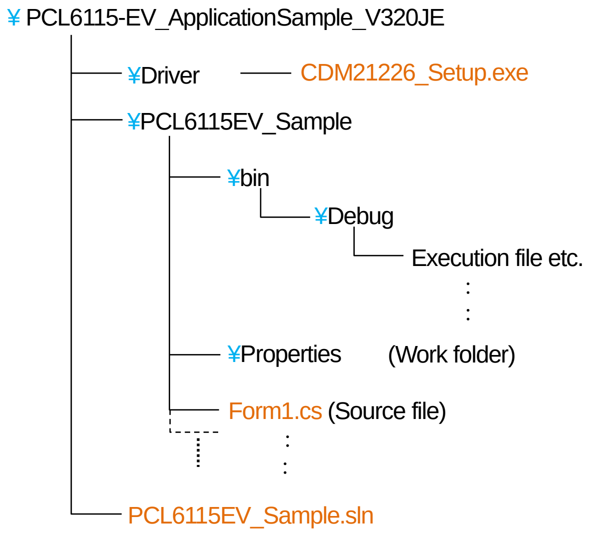
2.4 Notes

- We can not answer about usage of “Microsoft Visual C#” etc.
- We can not answer about usage of FTDI made products etc.
- Please understand that we will not be responsible at all even if damage occurs as a result of operating the application based on this sample program.

3 Outline of sample program

3.1 Folder structure

When you unzip the compressed file (PCL6115-EV_ApplicationSample_V320JE.zip), the following folders are created.



3.2 File structure

<¥PCL6115-EV_ApplicationSample_V320JE>

PCL6115EV_Sample.sln	·····	Solution file

<¥PCL6115-EV_ApplicationSample_V320JE¥Driver >

CDM21226_Setup.exe	·····	Device driver installer (FTDI)

<¥PCL6115-EV_ApplicationSample_V320JE¥PCL6115EV_Sample>

Form1.cs	·····	Source code
clsFTDI_Serial.cs	·····	FTDI Access function
accessPCL_Serial.cs	·····	PCL6115 Access function
FTD2XX_NET.dll	·····	FTDI Library
FTD2XX_NET.xml	·····	FTDI XML document
*.bmp	·····	Image data
Others		

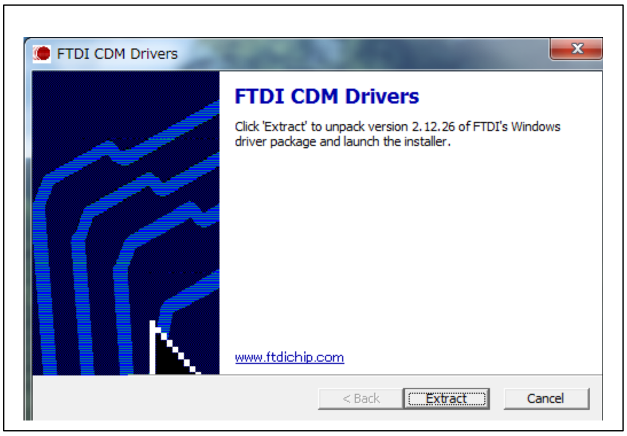
<¥PCL6115-EV_ApplicationSample_V320JE¥PCL6115EV_Sample¥bin¥Debug>

PCL6115EV_Sample.exe	·····	Execution file
FTD2XX_NET.dll	·····	FTDI Library (Required at execution)
FTD2XX_NET.xml	·····	FTDI XML document (No need for execution)
Others	·····	Work file etc. (No need for execution)

4 Installation of device driver

Double-click “CDM21226_Setup.exe” to launch the installer. Then, follow the instructions on screen to complete the installstion.

If you have already installed it, you do not need to re-install.



FTDI CDM Drivers

FTDI CDM Drivers

Click 'Extract' to unpack version 2.12.26 of FTDI's Windows driver package and launch the installer.

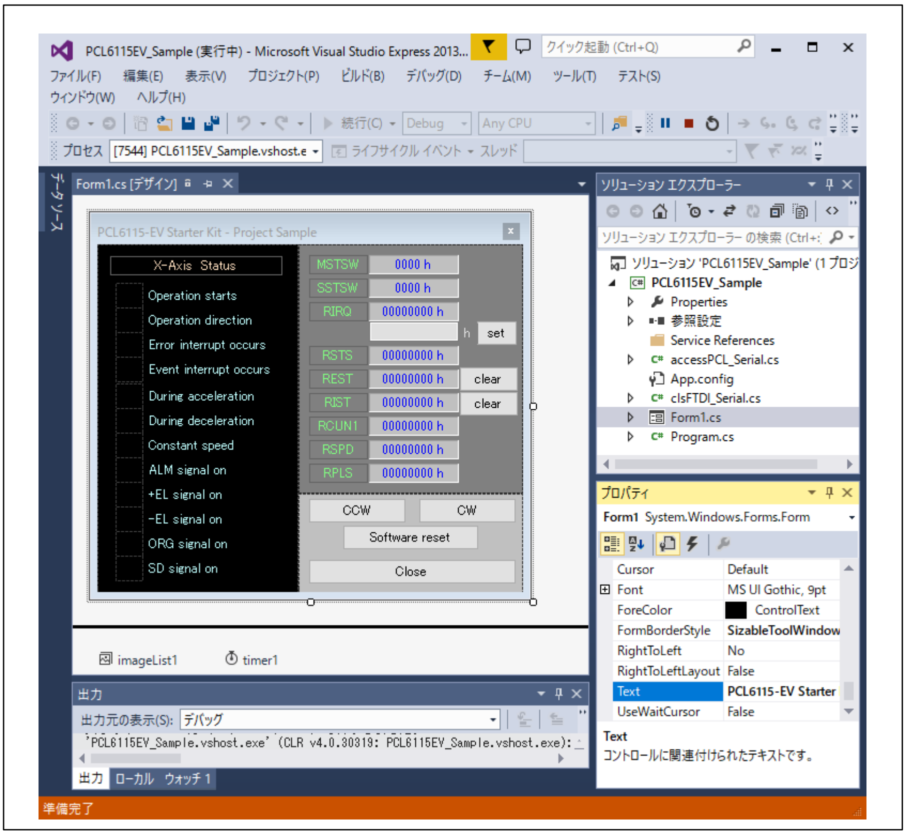
www.ftdichip.com

Note: Please download and use the latest version of the device driver on FTDI's website (http://www.ftdichip.com/Drivers/D2XX.htm),

5 Start-up project in C#

Please make sure that PCL6115-EV is connected to your PC properly.

Confirm that "Microsoft Visual C#" is installed, and double-click PCL6115EV_Sample.sln “Solution file”.



PCL6115EV_Sample (実行中) - Microsoft Visual Studio Express 2013...

PCL6115-EV Starter Kit - Project Sample

X-Axis Status

Operation starts
Operation direction
Error interrupt occurs
Event interrupt occurs
During acceleration
During deceleration
Constant speed
ALM signal on
+EL signal on
-EL signal on
ORG signal on
SD signal on

MSTSW	0000 h	
SSTSW	0000 h	
RIRQ	00000000 h	
	h	set
RSTS	00000000 h	
REST	00000000 h	clear
RIST	00000000 h	clear
RCUN1	00000000 h	
RSPD	00000000 h	
RPLS	00000000 h	

ソリューション 'PCL6115EV_Sample' (1 プロジ
PCL6115EV_Sample
Properties
参照設定
Service References
accessPCL_Serial.cs
App.config
clsFTDI_Serial.cs
Form1.cs
Program.cs

Form1 System.Windows.Forms.Form

Cursor	Default
Font	MS UI Gothic, 9pt
ForeColor	ControlText
FormBorderStyle	SizableToolWindow
RightToLeft	No
RightToLeftLayout	False
Text	PCL6115-EV Starter
UseWaitCursor	False

Text
コントロールに関連付けられたテキストです。

出力元の表示(S): デバッグ

```
'PCL6115EV_Sample.vshost.exe' (CLR v4.0.30319: PCL6115EV_Sample.vshost.exe):
```

準備完了

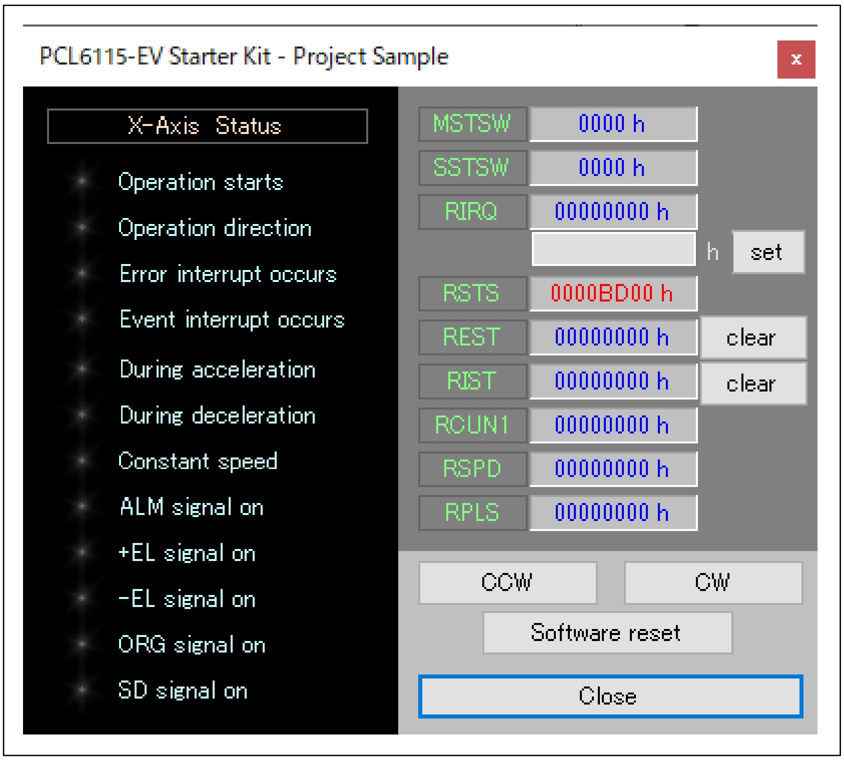
For installation of Microsoft’s product, please refer to Microsoft's website.

For details on how to build and debug projects, please refer to Microsoft website.

6 Operations

6.1 Start-up program

When you start debugging, the software on the following screen will start up.



PCL6115-EV Starter Kit - Project Sample

X-Axis Status

Operation starts
Operation direction
Error interrupt occurs
Event interrupt occurs
During acceleration
During deceleration
Constant speed
ALM signal on
+EL signal on
-EL signal on
ORG signal on
SD signal on

MSTSW 0000 h
SSTSW 0000 h
RIRQ 00000000 h
h
RSTS 0000BD00 h
REST 00000000 h
RIST 00000000 h
RCUN1 00000000 h
RSPD 00000000 h
RPLS 00000000 h

6.2 Display the status information

In the column “X - Axis Status”, you can check the X-axis status of PCL6115. Details are as follows.

Items	Descriptions
Operation starts	Becomes “1” by writing a start command. (MSTSW.SSCM)
Operation direction	Operation direction (CW: “0”, CCW: “1”) (RSTS.SDIR)
Error interrupt occurs	Becomes “1” when an error interrupt occurs. (MSTSW.SERR)
Event interrupt occurs	Becomes “1” when an event interrupt occurs. (MSTSW.SINT)
During acceleration	Becomes “1” while accelerating. (SSTSW.SFU)
During deceleration	Becomes “1” while decelerating. (SSTSW.SFD)
Constant speed	Becomes “1” while feeding at constant speed. (SSTSW.SFC)
ALM signal on	Becomes “1” when ALM input is ON. (SSTSW.SALM)
+EL signal on	Becomes “1” when +EL input is ON. (SSTSW.SPEL)
-EL signal on	Becomes “1” when -EL input is ON. (SSTSW.SMEL)
ORG signal on	Becomes “1” when ORG input is ON. (SSTSW.SORG)
SD signal on	Becomes “1” when SD input is ON. (SSTSW.SSD)

When the status becomes "1", the display color next to each item changes.

6.3 Display the resister information

The status of the X axis and the contents of several registers are displayed. Details are as follows.

Items	Descriptions
MSTSW	Main status
SSTSW	Sub status
RIRQ	Event interrupt factor setting register
RSTS	Extension status register
REST	Error interrupt factor register
RIST	Event interrupt factor register
RCUN1	COUNTER 1
RSPD	Current speed monitor
RPLS	Positioning counter

All are displayed in hexadecimal notation. A value 0 is desplaied in blue and other than 0 is displayed in red.

6.4 Operation button

6.4.1 CCW

By clicking, the operation in CCW direction starts, 2,304 pulses are output and the motor stops.

After the operation starts, the motoro accelerates from 1 pps to 1024 pps in about 1 second. After the motor runs at 1,024 pps for a while, it decelerates and stops in about 1 second.

6.4.2 CW

By clicking, the operation in CW direction starts.

The operation is the same for CCW, except the direction.

6.4.3 Software reset

Reset PCL6115.

6.4.4 set

You can set a value to RIRQ (Event interrupt factor setting register).

Enter a hexadecimal value in the column next to the “set” button. Then, click the "set" button.

6.4.5 clear

You can clear the status of REST (Error interrupt factor register) and RIST(Event interrupt factor register).

By clicking the button next to each register, the value of corresponding register is cleared to zero.

6.4.6 Close

Quit this software.

7 Source code

The source file is "Form1.cs".

Please check the operation procedure by revising the codes as you want.

7.1 Default setting

The default setting is described in the function "InitSet". It is called when you start the software and you reset the software.

Default setting operates as follows.

Operation	Contents
PRMG = 0x000004AF	Set the magnification value to 1.
PRMD = 0x00000041	Set operation mode as follows. Positioning operation. Linear acceleration/deceleration. Slow-down point automatic setting.
RENV1 = 0x00000002	Set environment setting 1 as follows. Set the output pulse specification to "010". Outputs pulses in the negative logic from OUT pin. Outputs LOW from DIR pin in the plus direction.
RENV2 = 0x80001D40	Set environment setting 2 as follows. Set general-purpose ports P3 to P7 as output ports. Automatic read and reset in REST and RIST registers are canceled. This software periodically reads all registers and displays them. Because the REST and RIST registers are also read, the flag automatically disappears because of reading automatic reset even if an error occurs. Therefore, you cannot check an error visually. To prevent this, set RENV2.MRST =1.

7.2 CCW operation

CCW operation is performed by the function "Exec_CCW" in the source file.

The following operations are performed in the function.

Operation	Contents
PRFL = 0x00000001	Set "1h" as start speed.
PRFH = 0x00000400	Set "400h"(1024) as operation speed.
PRUR = 0x00002588	Set "2588h" as acceleration rate.
PRMV = 0xFFFFF830	Set "FFFFF700h"(-2304) as positioning operations
STAUD	Run at high-speed start 2

By executing the start command, the motor accelerates from 1 pps to 1,024 pps in about 1 second, after running at 1,024 pps for a while it decelerates and stops in about 1 second.

The number of pulses output during operation is 2,304 pulses in the minus direction.

7.3 CW operation

CW operation is performed by the “Exec_CW” function in the source file.

The following operations are performed in the function.

Operation	Contents
PRFL = 0x00000001	Set “1h” as start speed.
PRFH = 0x00000400	Set “400h” (1,024) as operation speed.
PRUR = 0x00002588	Set “2588h” as acceleration rate.
PRMV = 0x000007D0	Set “0000900h” (+2,304) as positioning operations.
STAUD	Execute high speed start 2.

By executing the start command, a motor accelerates from 1 pps to 1,024 pps in about 1 second, after running at 1,024 pps for a while it decelerates and stops in about 1 second.

The number of pulses output during operation is 2,304 pulses in the plus direction.

7.4 Software reset operation

The software reset operation is performed by the function “Exec_SoftReset” in the source file.

The following operations are performed in the function.

Operation	Contents
SRST	Execute software reset command.

After executing the command, a default value is set by executing the function "InitSet".

7.5 “set” operation and “clear” operation

Setting the value to RIRQ register and clearing REST and RIST registers are conducted by the function "Exec_WriteReg" in the source file.

In this software, bit 31 of RENV 2 register is set to "1", therefore, the REST and RIST registers are cleared by writing "1" to a bit to be cleared.

REST and RIST registers are cleared to zero by writing values read from REST or RIST registers as they are.

7.6 Access function to PCL6115

You can access to PCL6115-EV board via USB.

Therefore, the commands to PCL6115 are stored in the buffer of the program on the PC side first before sending.

When reading data from PCL6115, the results of multiple reading commands are received via USB at once.

In this sample source, there are functions to store comments in the buffer and functions to send and receive commands.

Also, the buffer to store the commands is defined as follows in this sample source:

Buffer name	: FtBuff
Buffer size	: 1024byte

7.6.1 Status read function (Read_STATUS)

A command to read the main status of PCL6115 or to read the sub status and general port status is stored in the buffer.

By this command, 2 bytes of data are read on PCL6115-EV.

Receive this data with the function "GetUSB".

Read_STATUS (ref byte[] FtBuff, ref int FtIndex, int subS)	
FtBuff	Please specify a buffer for storing reading commands of main status (or sub status) for PCL6115-EV board. Please note that the function does not judge whether a command exceeds the buffer size (1024 byte).
FtIndex	Please specify a variable that manages the number of array variables.
subS	Please specify a target to be read. 0 : Read main status Other than 0 : Read sub status

Number of buffers used when setting a command	6 byte (Add to FtIndex)
Number of data to be received after command execution	2 byte
Order of received data	1 : main status bit 7 to bit 0 (or ports) 2 : main status bit 15 to bit 8 (or sub status)

7.6.2 Resister read function (Read_REG)

The command to read data from register of PCL6115 is stored in the buffer.

With this command, 4 bytes of data are read on PCL6115-EV side.

Receive this data with the function "GetUSB".

Even if a register whose register length is less than 32 bits is read out, the value of 4 bytes is always read.

The upper bits at this time are filled with zero.

Read_REG (ref byte[] FtBuff, ref int FtIndex, byte comm)	
FtBuff	Please specify the buffer for storing data read command from registers for PCL6115-EV board. Please note that the function does not judge whether a command exceeds the buffer size (1,024 byte).
FtIndex	Please specify a variable that manages the number of array variable usage.
comm	Please specify register read command of PCL6115.
Number of buffers used when setting command	20 byte (Add to FtIndex)
Number of data to be received after command execution	4 byte
Order of received data	1 : Register value bit 7 to bit 0 2 : Register value bit 15 to bit 8 3 : Register value bit 23 to bit 16 4 : Register value bit 31 to bit 24

7.6.3 Resister write function (Write_REG)

The command to write data to the register of PCL6115 and the write data are stored in the buffer.

There is no data read from PCL6115-EV by this command.

Write_REG (ref byte[] FtBuff, ref int FtIndex, uint RegD, byte Jsc, byte comm)	
FtBuff	Please specify the buffer for storing the data writing command to the register for PCL6115-EV board. Please note that the function does not judge whether a command exceeds the buffer size (1024 byte).
FtIndex	Please specify a variable that manages the number of array variable.
RegD	Please specify data you want to write to the register.
Jsc	Always set to zero.
comm	Please specify register write command of PCL6115.
Number of buffers used when setting command	24byte (Add to FtIndex)
Number of data to be received after command execution	0 byte
Order of received data	

7.6.4 Operation command write function (Write_COM)

The operation command of PCL6115 is stored in the buffer.

There is no data read from PCL6115-EV by this command.

Write_COM (ref byte[] FtBuff, ref int FtIndex, byte Jsc, byte comm)	
FtBuff	Please specify the buffer for storing the operation command writing command for PCL6115-EV board. Please note that the function does not judge whether a command exceeds the buffer size (1024 byte).
FtIndex	Please specify a variable that manages the number of array variable.
Jsc	Always set to zero.
comm	Please specify operation command of PCL6115.

Number of buffer used when setting a command	8 byte (Add to FtIndex)
Number of data to be received after command execution	0 byte
Order of received data	

7.6.5 Sending function to PCL6115-EV (SendUsb)

Send the commands stored in the buffer to PCL6115-EV.

After sending, commands are executed on PCL6115-EV board.

SendUsb (ref byte[] FtBuff, ref int FtIndex)	
FtBuff	Please specify a buffer that stores command groups with functions such as "Read_STATUS", "Read_REG", "Write_REG", "Write_COM" etc.
FtIndex	Please specify the number of buffers used. After transmission, this variable is cleared to zero.

7.6.6 Receiving function from PCL6115-EV (GetUsb)

When executing a reading command on PCL6115-EB board, data to be read out is stored in the sending buffer of PCL6115-EV board.

This function sends the contents of this buffer to the PC.

GetUsb (ref byte[] FtBuff)	
FtBuff	The read data is stored in the execution result of the executed read command.

[Example]

Suppose that commands are stores and executed in the following order.

```
FtIndex=0;
Read_STATUS(ref FtBuff,   ref FtIndex,   0);   // Read main status
Read_REG(ref FtBuff, ref FtIndex, 0xE3);      // Read RCUN1 register
Read_STATUS(ref FtBuff,   ref FtIndex,   1);   // Read sub status
//
SendUsb(ref FtBuff,   ref FtIndex);
GetUsb(ref FtBuff);
```

The buffer after receiving is as follows.

	FtBuff	
0	Main status bit 7 ~ bit 0	// Read main status
1	Main status bit 15 ~ bit 8	
2	RCUN1 register bit 7 ~ bit 0	// Read RCUN1 register
3	RCUN1 register bit 15 ~ bit 8	
4	RCUN1 register bit 23 ~ bit 16	
5	RCUN1 register bit 31 ~ bit 24	
6	State of general-purpose port	// Read sub status
7	Sub status	

Please note the above order to read a software on PC.

NPM reserves all intellectual property rights, including copyrights, for the software. NPM does not transfer any rights regarding the software to customer(s). Customer(s) may use the current software only for the purpose of using the products. NPM does not provide any warranty with our software, whether expressed or implied, including the completeness, accuracy, applicability, usefulness, or non-infringement of third-party intellectual property. In addition, NPM is not responsible for any damages (including lost revenue or profits) caused by using the software. Customer(s) must comply with the export control laws and regulations of the countries in case the customer(s) use(s) the software outside the country

Revision

Revision	Date	Contents
1st	March 19. 2018	Initial Release.
2nd	July 16, 2019	Change Document No. Add the manual list.
3rd	June 16, 2021	Software version Corrected errors on display Operation is start → Operation starts Ivent interrupt occurs → Event interrupt occurs Added software disclaimer

NPM Impress, not just satisfy
Nippon Pulse Motor Co., Ltd.
www.pulsemotor.com/group/

Information
www.pulsemotor.com/group/support

Third edition issued in June 2021