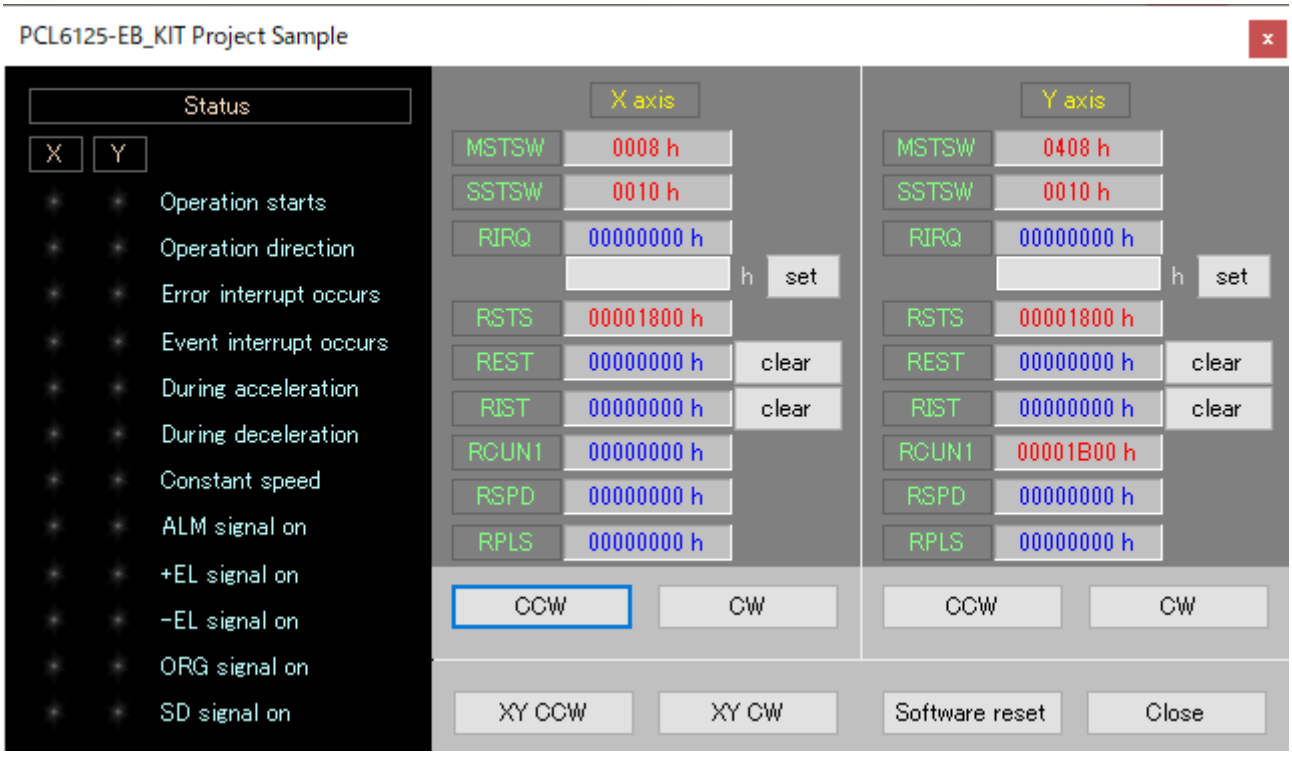
PCL6125 Evaluation Board

PCL6125-EB

User's Manual

Sample Program



INDEX

NPM

1. Introduction

This manual describes the specifications, functions, and usages of the sample program for our PCL6125-EB Evaluation Board (PCL6125-EB_ApplicationSample_V110.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

This is a user's manual of the application software to operate a control board.

By using this software and PCL6125 Evaluation Board (PCL6125-EB), you can learn motor control functions using pulse control LSI, PCL6125.

Please refer to the following manuals along with this manual.

(x: revision)

	Manual name [Outline]	Document file name	Software file name	Document no.
Hardware Manual	PCL6125 Evaluation Board (PCL6125-EB) User's Manual (Hardware)	PCL6125-EB_ HardwareManual_VerxE.pdf		TA600038-ENx/x
	PCL6125-EB Evaluation Kit (PCL6125-EB_KIT) User's Manual	PCL6125-EB_KIT Manual_VerxE.pdf		TA600071-ENx/x
Application Software Manual	PCL6125 Evaluation Board (PCL6125-EB) User's Manual (Application software) [Setting acceleration/deceleration pattern and register indication]	PCL6125-EB_ ApplicationManual_VerxE.pdf	PCL6125-EB_ Application_VxxxJE.zip	TA600039-ENx/x
	PCL6125 Evaluation Board (PCL6125-EB) User's Manual (Sample Program)	PCL6125-EB_ Application SampleManual_ VerxJ.pdf	PCL6125-EB_ ApplicationSample _VxxxJE.zip	TA600075-ENx/x (This document)
Motion Pattern Builder Manual	PCL6125 Evaluation Board (PCL6125-EB) User's Manual (Motion Pattern Builder Application Software) [Visually describes the functions to control an axis with a flowchart]	PCL6125-EB_ MotionBuilderManual_VerxE.pdf	PCL6125-EB_ MotionBuilder_ VxxxJE.zip	TA600040-ENx/x
Reference material	PCL6115/6125/6145 User's Manual			DA70152-0/xE

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

2.1 Operation environment

We have checked the operation of this software with Windows 7 and Windows10 (both 32-bit and 64-bit).

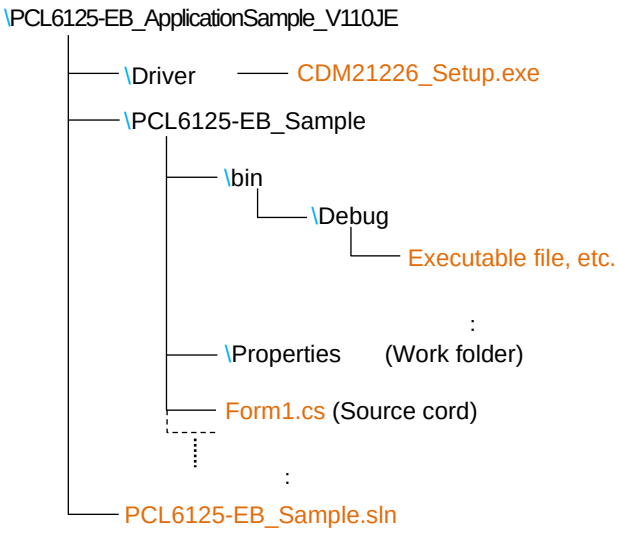
(We have not checked it with OS other than the above.)

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

3. Outline of sample program

3.1 Folder structure

When you unzip the compressed file (PCL6125-EB_ApplicationSample_V110JE.zip), the following folders are created.



3.2 File structure

<\PCL6125-EB_ ApplicationSample_V110JE folder>

PCL6125-EB_Sample.sln	·····	Solution file

<In the following folder : \PCL6125-EB_ ApplicationSample_V110JE\Driver>

CDM21226_Setup.exe	·····	Installer of device driver (made by FTDI)

<In the following folder : \PCL6125-EB_ ApplicationSample_V110JE\PCL6125-EB_Sample>

Form1.cs	·····	Source cord
clsFTDI.cs	·····	FTDI access function
accessPCL.cs	·····	PCL6125 access function
FTD2XX_NET.dll	·····	FTDI library
FTD2XX_NET.xml	·····	FTDI XML document
*.bmp	·····	Image data
Others	·····	Work file (No need during execution)

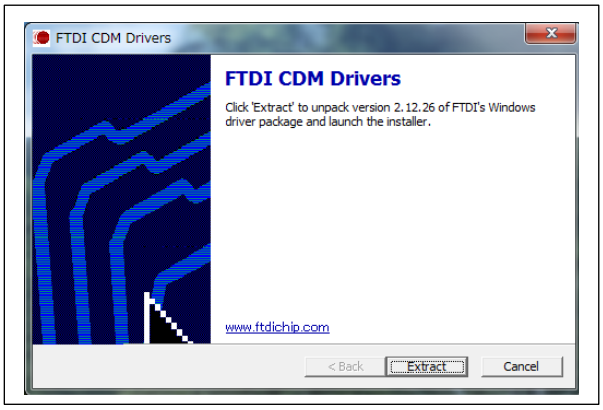
<In the following folder : \PCL6125-EB_ ApplicationSample_V110JE\PCL6125-EB _Sample\bin\Debug>

PCL6125-EB_Sample.exe	·····	Exe file
FTD2XX_NET.dll	·····	FTDI library (Necessary during execution)
FTD2XX_NET.xml	·····	FTDI XML document (No need during execution)
Others	·····	Work file (No need during execution)

3.3 Installation of device driver

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

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

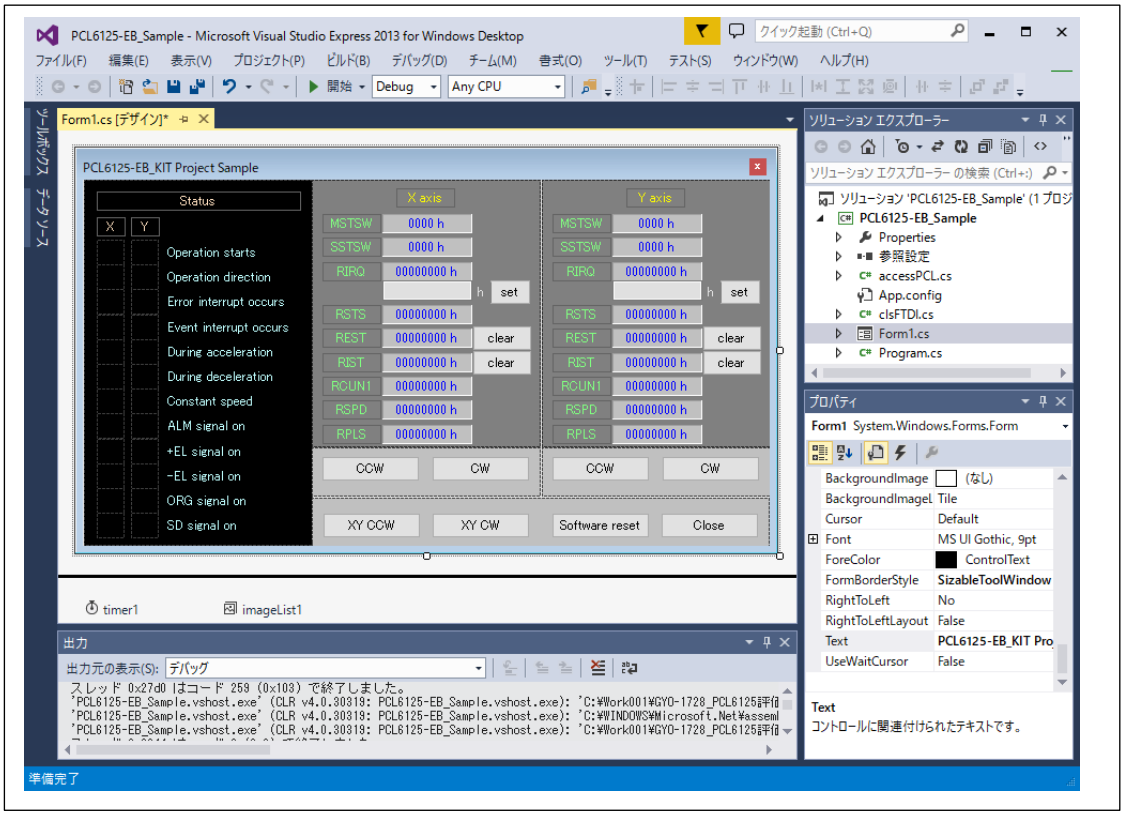


Note: Please download and use the latest version of the device driver on FTDI's website.
(http://www.ftdichip.com/Drivers/D2XX.htm).

4. Start-up a project in C#

Please make sure that PCL6125-EB Evaluation Kit (PCL6125-EB_KIT) is connected to your PC properly.

Confirm that "Microsoft Visual C#" is installed, and double-click PCL6125-EB_Sample.sln “Solution file”.



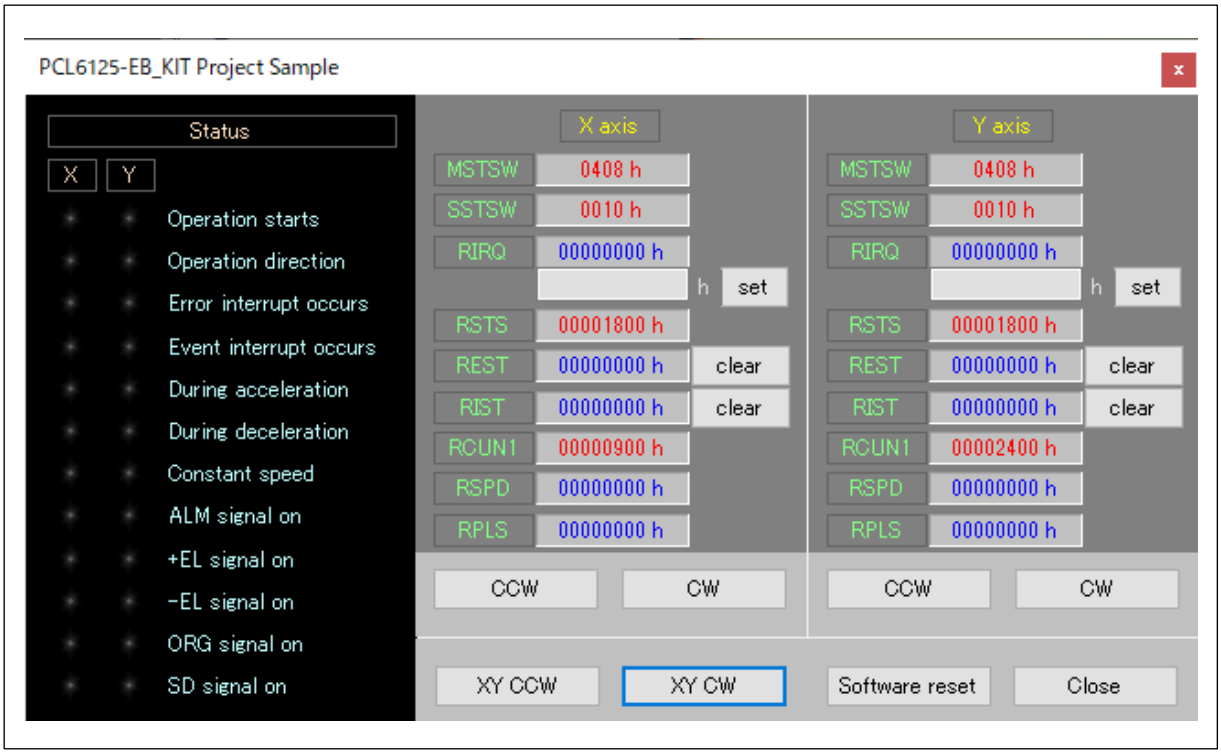
For installation of Microsoft’s product, please refer to Microsoft's website.

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

5. Operations

5.1 Start-up a program

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



5.2 Display the status information

In the column “Status”, you can check the status of X and Y axes of PCL6125. Details are as follows.

Items	Description	Name
Operation starts	Becomes "1" during running.	(MSTSW.SSCM)
Operation direction	Becomes "0" in CW and “1” in CCW.	(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" during accelerating.	(SSTSW.SFU)
During deceleration	Becomes "1" during 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.

5.3 Display the register information

The status of X axis, Y 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
REST	Error interrupt factor status register
RIST	Event interrupt factor status register
RCUN1	COUNTER 1
RSPD	Current speed monitor
RPLS	Positioning counter acquisition register

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

For values (bits) of each register, refer to PCL6115/6125/6145 User's manual.

5.4 Operation button

5.4.1 CCW

By clicking "CCW" of X axis, the operation of X axis in CCW direction starts. 2,304 pulses are output and the motor stops. After the operation starts, the motor accelerates from 1 pps to 1,024 pps in about 1 second. After the motor runs at 1,024 pps for a while, it decelerates and stops in about 1 second.

By clicking "CCW" of Y axis, the operation of Y axis in CCW direction starts and the operation is the same as X axis.

5.4.2 CW

By clicking "CW" of X axis, the operation of X axis in CW direction starts.

By clicking "CW" of Y axis, the operation of Y axis in CW direction starts.

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

5.4.3 set

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

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

5.4.4 clear

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

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

5.4.5 XY CCW

By clicking "XY CCW", a simultaneous operation of X and Y axes in CCW direction starts, and the operation stops after 2,304 pulse are output. After the operation starts, a motor accelerates from 1 pps to 1,024 pps in about 1 second, operates at 1,024 pps for a while, and then decelerates and stops in about 1 second.

5.4.6 XY CW

By clicking “XY CW”, a simultaneous operation of X and Y axes in CW direction starts. The operation is the same as XY CCW.

5.4.7 Software reset

By clicking “Software reset” to reset PCL6125. If you click it during operation (clock is being output), the operation will stop immediately.

5.4.8 Close

By clicking “Close” to exit this software. If you click it while operating in positioning mode, the sample program will end, and a motor will stop after operating for the feeding amount.

6. Source code

The source file is "Form1.cs".

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

Please refer to PCL6115/6125/6145 User’s Manual.

6.1 Default setting

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

Default setting operates as follows:

Operation	Contents
PRMG = 0x000004AF	Set the magnification value to 1.
PRMD = 0x00000041	Set the operation mode as follows. - Positioning operation - Linear acceleration/deceleration - Slow-down point automatic setting
RENV1 = 0x00000002	Set the 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 = 0x80000055	Set the environment setting 2 as follows. Set general-purpose ports P0 to P3 as output ports. Automatic read and reset in REST and RIST registers are canceled. (This software periodically reads all registers and displays the values including REST and RIST registers. If automatic read and reset function is effective, flags in REST and RIST registers may be erased automatically, so you cannot check errors visually. To prevent this, set RENV2.MRST = 1.)

6.2 CCW operation

CCW operation is performed by the functions, “Exec_XCCW”, “Exec_YCCW”, “Exec_XYCCW” in the source file. The following operations are performed in the functions.

Operation	Contents
PRFL = 0x00000001	Set "1h" as the starting speed.
PRFH = 0x00000400	Set "400h" (1024) as the operation speed.
PRUR = 0x00002588	Set "2588h" as the acceleration rate.
PRMV = 0xFFFFF700	Set "FFFFF700h"(-2304) as the feeding amount.
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 1024 pps for a while, it decelerates and stops in about 1 second. During the operation, 2,304 pulses are output in the minus direction.

6.3 CW operation

CW operation is performed by the functions, “Exec_XCW”, “Exec_YCW”, and “Exec_XYCW” in the source file. The following operations are performed in the functions.

Operation	Contents
PRFL = 0x00000001	Set “1h” as a start speed.
PRFH = 0x00000400	Set “400h” (1,024) as an operation speed.
PRUR = 0x00002588	Set “2588h” as acceleration rate.
PRMV = 0x00000900	Set “0000900h” (+2,304) as feeding amount.
STAUD	Run at 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 in about 1 second and stops. The number of pulses output during the operation is 2,304 pulses in the plus direction.

6.4 “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, REST and RIST register values can be cleared by writing "1" to the bit to be cleared.

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

6.5 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".

6.6 Access function to PCL6125

You can access to PCL6115-EV board via USB.

Therefore, the commands to PCL6125 are stored in the buffer of the program on the PC side first before sending.

When reading data from PCL6125, 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

Size : 1,024 bytes

6.6.1 Status reading function (Read_STATUS)

A command to read the main status of PCL6125 or to read the sub-status and general purpose port status are stored in the buffer.

By this command, 4 bytes of data are read on PCL6125-EB. Receive this data by the function "GetUSB".

Read_STATUS (ref byte[] FtBuff, ref int FtIndex, int subS)	
FtBuff	Please specify a buffer for storing reading command for PCL6125-EB board. Please note that the function does not judge whether a command exceeds the buffer size (1024 byte).
FtIndex	Please specify a variable that specifies the number of array variables.
subS	Specify a target to be read. 0 : Read main status. the main status (or sub status) of Other than 0 : Read sub status

Number of buffers used when setting a command	14 bytes (Add to FtIndex)
Number of data to be received after command execution	4 bytes
Order of received data	1 : X axis main status bit 7 to bit 0 (or port) 2 : X axis main status bit 15 to bit 8 (or sub status) 3 : Y axis main status bit 7 to bit 0 (or port) 4 : Y axis main status bit 15 to bit 8 (or sub status)

6.6.2 Register reading function (Read_REG)

The command to read data from register of PCL6125 is stored in the buffer. With this command, 8 bytes of data are read on PCL6125-EB side.

Receive this data by 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 a buffer for storing reading commands of main status (or sub status) for PCL6125-EB board. Please note that the function does not judge whether a command exceeds the buffer size (1024 byte).
FtIndex	Please specify a variable that specifies the number of array variables.
comm	Please specify a register reading command of PCL6125.
Number of buffers used when setting command	19 bytes (Add to FtIndex)
Number of data to be received after command execution	8 bytes
Order of received data	1 : X axis register value bit 7 to bit 0 2 : X axis register value bit 15 to bit 8 3 : X axis register value bit 23 to bit 16 4 : X axis register value bit 31 to bit 24 5 : Y axis register value bit 7 to bit 0 6 : Y axis register value bit 15 to bit 8 7 : Y axis register value bit 23 to bit 16 8 : Y axis register value bit 31 to bit 24

6.6.3 Register writing function (Write_REG)

The command to write data to the register of PCL6125 and the write data are stored in the buffer. There is no data read from PCL6125-EB by this command.

Write_REG (ref byte[] FtBuff, ref int FtIndex, uint RegD, byte Jsc, byte comm)	
FtBuff	Please specify a buffer for storing a data writing command to the register for PCL6125-EB board. Please note that the function does not judge whether a command exceeds the buffer size (1024 byte).
FtIndex	Please specify a variable that specifies 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 writing command of PCL6125.
Number of buffers used when setting command	15 byes (Add to FtIndex)
Number of data to be received after command execution	0 byte
Order of received data	—

6.6.4 Operation command write function (Write_COM)

The operation command of PCL6125 is stored in the buffer. There is no data read from PCL6125-EB by this command.

Write_COM (ref byte[] FtBuff, ref int FtIndex, byte Jsc, byte comm)	
FtBuff	Please specify a buffer for storing the operation command writing command for PCL6125-EB 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	Set “0x01” for X axis, “0x02” for Y axis, and ” 0x03” for X-Y axes.
comm	Please specify an operation command of PCL6125.

Number of buffer used when setting a command	11 bytes (Add to FtIndex)
Number of data to be received after command execution	0 byte
Order of received data	—

[Example]

The data and operation commands to the registers are stored and executed in the following order.

```
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 Sending function to PCL6125-EB board (SendUsb)

Send the commands stored in the buffer to PCL6125-EB.

After sending, commands are executed on PCL6125-EB board.

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

6.6.6 Receiving function from PCL6125-EB board (GetUsb)

When executing a reading command on PCL6215-EB board, data to be read out is stored in the sending buffer of PCL6125-EB board.

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

GetUsb (ref byte[] FtBuff)	
FtBuff	The data read is stored in the execution result order of the executed reading 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	Status bit 7 to bit 0	// Read X axis main status
1	Status bit 15 to bit 8	
2	Status bit 7 to bit 0	// Read Y axis mains status
3	Status bit 15 to bit 8	
4	RCUN1 register bit 7 to bit 0	// Read X axis RCUN1 register
5	RCUN1 register bit 15 to bit 8	
6	RCUN1 register bit 23 to bit 16	
7	RCUN1 register bit 31 to bit 24	
8	RCUN1 register bit7 to bit 0	// Read Y axis RCUN1 register
9	RCUN1 register bit 15 to bit 8	
10	RCUN1 register bit 23 to bit 16	
11	RCUN1 register bit 31 to bit 24	
12	General-purpose port status	// Read X axis sub status
13	Sub status	
14	General-purpose port status	// Read Y axis sub status
15	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 purchased.

Revision

Revision	Date	Contents
1st	May 31, 2021	New document.
2nd	June 15, 2021	Software version V1.10 Corrected errors on display Operation is start → Operation starts Ivent interrupt occurs → Event interrupt occurs Added software disclaimer

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

www.pulsemotor.com/global

Information
www.pulsemotor.com/global/contact

Second edition issued in June, 2021