PenPaper Sample Code for Windows Application

Date: 2018/3/27

The sample code and program demonstrate the procedure to communicate with the PenPaper device and get the data of the pen position and button status of the PenPaper. To test this sample program, you have to pair the PenPaper with Windows by the system’s adding device manner first. The PenPaper is a Bluetooth LE device, a Bluetooth LE enabled Windows system is necessary to pair with the PenPaper.

There are two methods used in the sample code to communicate with the PenPaper:

Method 1: PenPaper HID Mini-driver

Method 2: BluetoothGATTxxx functions

These two methods can not work at the same time. If the Windows system has installed the PenPaper HID mini-driver, then the method 1 will work with the driver to get the PenPaper position data and button status. In this situation, the method 2 cannot work because the BluetoothGATTxxx functions need an input of the BluetoothLE device HANDLE, but the HANDLE of the PenPaper device is already owned the PenPaper HID mini-driver.

Conversely, if the Windows system does not install the PenPaper HID mini-driver, the method 1 cannot work because the driver did not exist. The application can get the HANDLE of the PenPaper device and then call to the BluetoothGATTxxx functions to communicate with the PenPaper device directly.

The sample program checks if the PenPaper HID mini-driver exists or not when it starts. If the driver exists, the program creates a handle to the driver and enable the function button “Connect to Driver”, which indicates that the method 1 will be used to get data from PenPaper. Otherwise, the “Connect to Device” button is enabled to indicate that the method 2 will be used.

Click the function button “Connect to Driver” or “Connect to Device”, which enabled, to start the test of the PenPaper. Use the pen of the PenPaper to draw on the PenPaper and the program will immediately show the position data and button status, plus the pressure-sensitive stroke in the client area of the program.

**Method 1: PenPaper Hid Mini-driver (Driver Interface Command)**

Since the PenPaper HID mini-driver emulates the PenPaper as a HID device, your application may communicate with the driver by the HID interface (HidD\_xxx() functions). The mostly used HID interface in the sample program are HidD\_SetFeature() and HidD\_GetFeature() functions. For example, by calling the HidD\_SetFeature() function, the program can switch the working mode of the driver between the Pen Function mode and Writing mode. Currently, the PenPaper HID mini-driver support the following interface command for applications:

**For HidD\_SetFeature():**

|  |  |
| --- | --- |
| Code | Command and Parameter |
| 0x01 | COMMAND\_ENABLE\_PEN\_FUNCTION |
| 0x02 | COMMAND\_DISABLE\_PEN\_FUNCTION |
| 0x03 | COMMAND\_ENABLE\_WRITING\_MODE |
| 0x04 | COMMAND\_DISABLE\_WRITING\_MODE |
| 0x05 | Reserved |
| 0x06 | COMMAND\_SET\_TRACKING\_AREA  Parameter: UCHAR\_DATA\_1: Orientation  USHORT\_DATA\_1: Tracking Area Left Margin  USHORT\_DATA\_2: Tracking Area Top Margin  USHORT\_DATA\_3: Tracking Area Width  USHORT\_DATA\_4: Tracking Area Height |
| 0x07 | Reserved |
| 0x08 | COMMAND\_SET\_ORIENTATION |
| 0x09 | COMMAND\_SET\_PRESSURE\_LEVEL  Parameter: UCHAR\_DATA\_1: Pressure Adjust Level  UCHAR\_DATA\_2: Position of the Pressure Value 1  in the pressure table. There are  1024 values for each pressure  table.  USHORT\_DATA\_1: Pressure Value 1  USHORT\_DATA\_2: Pressure Value 2  USHORT\_DATA\_3: Pressure Value 3  USHORT\_DATA\_4: Pressure Value 4 |
| 0x0A | COMMAND\_SET\_CLICK\_THRESHOLD  Parameter: UCHAR\_DATA\_1: Click Threshold Level  USHORT\_DATA\_1: Tip On Threshold  USHORT\_DATA\_2: Tip Off Threshold |
| 0x0B | COMMAND\_SET\_FEATURE\_NUMBER\_RETURNED  Parameter: UCHAR\_DATA\_1: The set number of the Feature  content that returned by the  IOCTL\_UMDF\_HID\_GET\_FEATURE |

**For HidD\_GetFeature():**

Based on the Feature Number set by the above command 0x0B (COMMAND\_SET\_FEATURE\_NUMBER\_RETURNED), the HidD\_GetFeature() function has different return value.

|  |
| --- |
| Set number = 1 |
| Return Value:  UCHAR\_DATA\_1: Major version of the driver  UCHAR\_DATA\_2: Minor version of the driver |

|  |
| --- |
| Set number = 2 |
| Return Value:  UCHAR\_DATA\_1: Current Orientation  UCHAR\_DATA\_2: Connection state of the PenPaper  USHORT\_DATA\_1: Working area left margin  USHORT\_DATA\_2: Working area top margin  USHORT\_DATA\_3: Width of the working area  USHORT\_DATA\_4: Height of the working area |

Note: The PenPaper HID mini-driver use the struct

PENPAPER\_DRIVER\_CONTROL\_SET for HidD\_SetFeature() and

HidD\_GetFeature() functions. Please refer to the file stdafx.h for more information.

**Report Data Format:**

The sample program creates a thread and while loop to continuously read the Report data from the driver. The Report data format is as follows:

{

BYTE ReportID;

BYTE DeviceID;

struct

{

BYTE TipSwitch : 1;

BYTE Eraser : 1;

BYTE BarrelSwitch : 1;

BYTE InRange : 1;

BYTE Invert : 1;

BYTE Padding : 3;

} Status;

USHORT X\_coor;

USHORT Y\_coor;

USHORT PressureValue;

}

**Method 2: BluetoothGATTxxx functions**

To use the BluetoothGATTxxx functions to communicate with the PenPaper, you need the PenPaper Bluetooth related information. Please refer to the document: [PenPaper Bluetooth LE Specification (V1.0).doc](PenPaper%20Bluetooth%20LE%20Specification%20(V1.0).doc).