GeneralTouch Touchscreen USB Interface 2038 Rev 1

Technical specification

Version 2

General Touch Co. Ltd.

.Introduction

Touchscreen is used widely all over the world today. It is so easy to use that everyone even who knows nothing about computers can use it.Meanwhile,It’s well known that Universal Serial Bus (USB) is a huge success,and USB-based peripherals are everywhere. Because of the two advantages,The combination of touchscreen and USB is a very good choice. So,We try to improve our designs to provide the customers with more easy_to_ease touchscreen connections interface.

.Touchscreen Controller Protocol

The USB devices can be divided into several classes.The HID class consists primarily of devices that are used by humans to control the operations of the computer systems. Our GeneralTouch touchscreen USB interface is a typical application of HID class. Its main function are as follows:

*. The HID firmware specification(version 1.1) is supported;

*. Supports x,y and z touchscreen coordinate system, and they are absolute touchscreen values;Its range is 0~4095 for x ,y coordinate system and 0~255 for z coordinate system.

*. Supports Left,Right,and Middle button modes, and you can change touchscreen button modes with SET_REPORT HID command;

*. Most HID commands are supported, GET_PROTOCOL and SET_PROTOCOL commands are not required, so they are not supported;

.Touch Packets Report Format

The touch packet is composed of ButtonState and X,Y, Z coordinates.Its total length in bytes in 7.

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
Buttons	**XLow**	**XHigh**	**YLow**	**YHigh**	**ZLow**	**ZHigh**

Buttons = button state,see description below;
XLow = X touch coordinate low byte;
XHigh = X touch coordinate high byte;
YLow = Y touch coordinate low byte;
YHigh = Y touch coordinate high byte;
ZLow = Z touch coordinate low byte;
ZHigh = Z touch coordinate high byte;

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	Reserved	Reserved	Reserved	Reserved	**Middle**	**Right**	**Left**

Left = Left button down flag,If 1,the left button is down;
Right = Right button down flag,If 1,the right button is down;
Middle = Middle button down flag,If 1,the middle button is down;

NOTES:The Reserved bits are defined as 0.

.Feature Report Format

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	Reserved	Reserved	Reserved	Reserved	**Middle**	**Right**	**Left**

Left = Left button mode flag,If 1,current button mode is Left button mode;
Right = Right button mode flag,If 1,current button mode is right button mode;
Middle = Middle button mode flag,If 1,current button mode is middle button mode;
Reserved = Reserved,defined as 0.

NOTES:Default button mode is Left button mode,Device driver writers can choose to use this feature to provide the user with button mode support,Also,They can just choose to use their drivers provide button mode emulation support.

.Appendix

1.)USB Touchscreen Device Descriptor

```
/* HID USB Touchscreen device descriptor */
const char device_desc_table[]=
{
        0x12,                   /* size of descriptor (18 bytes) */
        0x01,                   /* descriptor type (device descriptor) */
        0x10, 0x01,             /* USB spec release (ver 1.1) */
        0x00,                   /* class code (each interface specifies class    information) */
        0x00,                   /* device sub-class (must be set to 0 because class code is 0) */
        0x00,                   /* device protocol (no class specific protocol) */
        0x08,                   /* maximum packet size (8 bytes) */
        0xFC, 0x0D,             /* vendor ID (vendor ID) */NOTES
        0x01, 0x00,             /* product ID (USB touchscreen product ID) */NOTES
        0x01, 0x00,             /* device release number */
        0x01,                   /* index of manufacturer string (1) */
        0x02,                   /* index of product string (2) */
        0x00,                   /* index of serial number string (not supported = 0x00) */
        0x01                    /* number of configurations (1) */
};
```

NOTES:We have received our vendor ID from USB IF recently,It is decimal 3580 and hexadecimal 0DFC.So we modify this descriptor here.

2.) USB Touchscreen HID Report Descriptor

```
/* HID Touchscreen report descriptor */
const char hid_report_desc_table[]=
{
        0x05, 0x01,             //   USAGE_PAGE (Generic Desktop)
        0x09, 0x02,                         //    USAGE (Mouse)
        0xa1, 0x01,                         //    COLLECTION (Application)
        0x09, 0x01,                         //      USAGE (Pointer)
        0xa1, 0x00,                         //      COLLECTION (Physical)
        0x05, 0x09,                         //          USAGE_PAGE (Button)
        0x19, 0x01,                         //          USAGE_MINIMUM (Button 1)
        0x29, 0x03,                         //          USAGE_MAXIMUM (Button 3)
        0x15, 0x00,                         //          LOGICAL_MINIMUM (0)
        0x25, 0x01,                         //          LOGICAL_MAXIMUM (1)
        0x95, 0x03,                         //          REPORT_COUNT (3)
        0x75, 0x01,                         //          REPORT_SIZE (1)
        0x81, 0x02,                         //          INPUT (Data,Var,Abs)
        0x95, 0x01,                         //          REPORT_COUNT (1)
        0x75, 0x05,                         //          REPORT_SIZE (5)
        0x81, 0x03,                         //          INPUT (Cnst,Var,Abs)
        0x05, 0x01,                         //          USAGE_PAGE (Generic Desktop)
        0x09, 0x30,                         //          USAGE (X)
        0x09, 0x31,                         //          USAGE (Y)
        0x09, 0x00,                         //          USAGE (Undefined)
        0x15, 0x00,                         //          LOGICAL_MINIMUM (0)
        0x26, 0xFF,0x7F,                     //          LOGICAL_MAXIMUM (32767)
        0x35, 0x00,                         //          PHYSICAL_MINIMUM(0)
        0x47, 0xFF,0xFF,0x00,0x00,          //          PHYSICAL_MAXIMUM(65535)
        0x75, 0x10,                         //          REPORT_SIZE (16)
        0x95, 0x03,                         //          REPORT_COUNT (3)
        0x81, 0x02,                         //          INPUT (Data,Var,Abs)
        0xc0,                               //      END_COLLECTION
        0xa1, 0x01,                         //      COLLECTION (Application)
        0x05, 0x09,                         //          USAGE_PAGE (Button)
        0x19, 0x01,                         //          USAGE_MINIMUM (Button 1)
        0x29, 0x03,                         //          USAGE_MAXIMUM (Button 3)
        0x15, 0x00,                         //          LOGICAL_MINIMUM (0)
        0x25, 0x01,                         //          LOGICAL_MAXIMUM (1)
        0x95, 0x03,                         //          REPORT_COUNT (3)
        0x75, 0x01,                         //          REPORT_SIZE (1)
        0xb1, 0x02,                         //          FEATURE (Data,Var,Abs)
        0x95, 0x01,                         //          REPORT_COUNT (1)
        0x75, 0x05,                         //          REPORT_SIZE (5)
        0xb1, 0x03,                         //          FEATURE (Cnst,Var,Abs)
        0xc0,                               //      END_COLLECTION
        0xc0                                 // END_COLLECTION
};
```

NOTES:We are sorry to say that we’ve changed our device’s HID descriptor.We list the revision as follows:

1.Changed USAGE(Z) to USAGE(Undefined),because some platforms try to interpret this value in the way which is not expected;

2.Changed LOGICAL_MINIMUM and LOGICAL_MAXIMUM,PHYSICAL_MINIMUM and PHYSICAL_MAXIMUM range value for USAGE(X),USAGE(Y),USAGE(Undefined);

Any suggestions and comments are appreciated.

3.)USB Touchscreen EndPoint Descriptor

```
/* HID Touchscreen Endpoint descriptor */
const char Endpoint_Descriptor[]=
{
    0x07,                       /* descriptor length (7 bytes) */
    0x05,                       /* descriptor type (ENDPOINT) */
    0x81,                       /* endpoint address (IN endpoint, endpoint 1) */
    0x03,                       /* endpoint attributes (interrupt) */
    0x07, 0x00,                 /* maximum packet size (7 bytes) */
    0x0A                        /* polling interval (10ms) */
};
```

4.)USB Touchscreen Class Descriptor

```
/* HID Touchscreen HID descriptor    */
const char Class_Descriptor[]=
{
    0x09,                       /* descriptor size (9 bytes) */
    0x21,                       /* descriptor type (HID) */
    0x10, 0x01,                 /* class specification (1.10) */
    0x00,                       /* hardware target country */
    0x01,                       /* number of hid class desriptors to follow (1) */
    0x22,                       /* report descriptor type (2) */
    sizeof(hid_report_desc_table),
    0x00
};
```

5.)USB Touchscreen Interface Descriptor

```
/* HID Touchscreen Interface descriptor */
const char Interface_Descriptor[]=
{
    0x09,                       /* length of descriptor (9 bytes) */
    0x04,                       /* descriptor type (INTERFACE) */
    0x00,                       /* interface number (0) */
    0x00,                       /* alternate setting (0) */
    0x01,                       /* number of endpoints (1) */
    0x03,                       /* interface class (3..defined by USB spec) */
    0x01,                       /* interface sub-class (1..defined by USB spec) */
```

```
        0x02,                                   /* interface protocol (2..defined by USB spec) */
        0x00                                    /* interface string index (not supported) */
};
```

6.)USB Touchscreen Configuration Descriptor

```
/* HID Touchscreen Configuration descriptor table */
const char config_desc_table[]=
{
        0x09,                                   /* length of descriptor (9 bytes) */
        0x02,                                   /* descriptor type (CONFIGURATION) */
        0x22, 0x00,                             /* total length of descriptor (33 bytes) */
        0x01,                                   /* number of interfaces to configure (1) */
        0x01,                                   /* configuration value (1) */
        0x04,                                   /* configuration string index (4) */
        0xE0,                                   /* configuration attributes (self powered, remote wakeup) */
        0x00                                    /* maximum power (set at 0,because it is self_powered.)*/
};
```

NOTES:We changed the configuration attributes and maximum power used by our device here, We found that the USB bus can not provide enough power to our device,or if it can,the other device on the device may be affected.

We changed our device to Self_Powered,that is,it does not consume any power from the bus now.