USB HID报告描述符教程

李大虾

望天一眼，云烟消散如云烟。

原文地址：

Tutorial about USB HID Report Descriptors

USB HID报告描述符是USB主机请求于USB设备的一种描述符。HID设备用报告的形式发送数据到主机，描述符告诉主机如何解释数据。下面将展示如何写一个描述符。
首先，到USB.org - HID Tools页面找到“Device Class Definition for HID”文档，下面叙述的内容本质上是该文档中的重要部分。
其次，在上述页面获得HID描述符工具，然后读完本教程之后想着如何使用它。手动写HID报告描述符是一件令人头痛不已的事情，本工具可以替代你转换二进制和十六进制，并查找数字代表的意义。

什么是USB HID报告描述符?
HID协议使得设备的实现变得简单，设备会定义数据包为HID描述符发送给主机。HID描述符是描述设备数据包的固定代码字节数组，包括设备支持多少个包，包有多大，以及包中每个字节和比特的含义。比如，带有计算程序按键的键盘告诉主机按键是按下还是松开状态，该信息放在数据包4的第6个字节的第2个比特，注意这个位置是设备指定说明的。设备通常将HID描述符存放在ROM里，不必深入理解或分析HID描述符。今天市场上的一些鼠标和键盘硬件实现仅仅使用一个8比特的CPU。--以上来自维基百科

简单一点开始，做一个标准的鼠标，三个按键，在X轴和Y轴上移动。因此要发送关于按键和移动的数据。每个按键用1个比特表示，每个字节表示移动一个轴上的有符号整型值，数据结构表示如下：

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Useless	Useless	Useless	Useless	Useless	Left Button	Middle Button	Right Button
Byte 1	X Axis Relative Movement as Signed Integer							
Byte 2	Y Axis Relative Movement as Signed Integer							

C语言格式如下：

```
struct mouse_report_t
{
    uint8_t buttons;
    int8_t x;
    int8_t y;
}
```

因此在描述符中，第一个项目必须描述按键，包含三个字段：

```
USAGE_PAGE (Button)
USAGE_MINIMUM (Button 1)
USAGE_MAXIMUM (Button 3)
```

每个按键的状态用1个比特表示，0或1：

```
LOGICAL_MINIMUM (0)
LOGICAL_MAXIMUM (1)
```

有3个比特来表示：

```
REPORT_COUNT (3)
REPORT_SIZE (1)
```

发送这些变量数据到电脑：

```
INPUT (Data,Var,Abs)
```

最终表述按键的表述如下：

```
USAGE_PAGE (Button)
USAGE_MINIMUM (Button 1)
USAGE_MAXIMUM (Button 3)
LOGICAL_MINIMUM (0)
LOGICAL_MAXIMUM (1)
REPORT_COUNT (3)
REPORT_SIZE (1)
INPUT (Data,Var,Abs)
```

5个没有用的填充比特：

```
REPORT_COUNT (1)
REPORT_SIZE (5)
INPUT (Cnst,Var,Abs)
```

X轴的移动：

```
USAGE_PAGE (Generic Desktop)
USAGE (X)
```

用1个字节的有符号整型，范围设成-127~127（实际上是-127~128，这样做是为了对称），来表示移动距离：

```
LOGICAL_MINIMUM (-127)
LOGICAL_MAXIMUM (127)
```

使用整个字节来发送：

```
REPORT_SIZE (8)
REPORT_COUNT (1)
```

当作坐标变量发送给电脑：

```
INPUT (Data,Var,Rel)
```

因此最终X轴移动的表述如下：

```
USAGE_PAGE (Generic Desktop)
USAGE (X)
LOGICAL_MINIMUM (-127)
LOGICAL_MAXIMUM (127)
REPORT_SIZE (8)
REPORT_COUNT (1)
INPUT (Data,Var,Rel)
```

那么Y轴移动的呢？可以写成这样：

```
USAGE_PAGE (Generic Desktop)
USAGE (X)
LOGICAL_MINIMUM (-127)
LOGICAL_MAXIMUM (127)
REPORT_SIZE (8)
REPORT_COUNT (1)
INPUT (Data,Var,Rel)
USAGE_PAGE (Generic Desktop)
USAGE (Y)
LOGICAL_MINIMUM (-127)
LOGICAL_MAXIMUM (127)
REPORT_SIZE (8)
REPORT_COUNT (1)
INPUT (Data,Var,Rel)
```

上述表述虽然没有问题，为了节省内存，可以表述如下：

```
USAGE_PAGE (Generic Desktop)
USAGE (X)
USAGE (Y)
LOGICAL_MINIMUM (-127)
LOGICAL_MAXIMUM (127)
REPORT_SIZE (8)
REPORT_COUNT (2)
INPUT (Data,Var,Rel)
```

综上，整个的表述是：

```
USAGE_PAGE (Button)
USAGE_MINIMUM (Button 1)
USAGE_MAXIMUM (Button 3)
LOGICAL_MINIMUM (0)
LOGICAL_MAXIMUM (1)
REPORT_COUNT (3)
REPORT_SIZE (1)
INPUT (Data,Var,Abs)
REPORT_COUNT (1)
REPORT_SIZE (5)
INPUT (Cnst,Var,Abs)
USAGE_PAGE (Generic Desktop)
USAGE (X)
USAGE (Y)
LOGICAL_MINIMUM (-127)
LOGICAL_MAXIMUM (127)
REPORT_SIZE (8)
REPORT_COUNT (2)
INPUT (Data,Var,Rel)
```

但是，这并未结束，为了让电脑知道这个是鼠标设备，必须这样：

```
USAGE_PAGE (Generic Desktop)
USAGE (Mouse)
COLLECTION (Application)
    USAGE (Pointer)
    COLLECTION (Physical)

    ... What we wrote already goes here

    END COLLECTION
END COLLECTION
```

因此在最后，下面是标准的鼠标USB HID报告描述符：

```
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)
0x15, 0x81,                    //     LOGICAL_MINIMUM (-127)
0x25, 0x7f,                    //     LOGICAL_MAXIMUM (127)
0x75, 0x08,                    //     REPORT_SIZE (8)
0x95, 0x02,                    //     REPORT_COUNT (2)
0x81, 0x06,                    //     INPUT (Data,Var,Rel)
0xc0,                          //   END_COLLECTION
0xc0                           // END_COLLECTION
```

这实际上是USB HID文档上的例子，同样也是HID工具提供的例子。

现在有了一些概念，继续研究：
Usage Pages（用例页）：文档中好像解释的不够好，概念包含有USAGE、USAGE_PAGE、USAGE_MINIMUM、USAGE_MAXIMUM。在描述符中，首先设置一个USAGE_PAGE，某些可用的USAGEs。在鼠标的例子中，在USAGE_PAGE (Generic Desktop)允许你使用USAGE (Mouse)，当用例页改为USAGE_PAGE (Button)，允许在USAGE(X) 和USAGE(Y)之前给按键指定USAGE_MINIMUM和USAGE_MAXIMUM，再然后用例页变回到USAGE_PAGE (Generic Desktop)。用例页就像是命名空间一般，改变用例页作用于“usages”是否可用。请阅读文档“HID Usage Tables”获得更多细节。

Collections（集合）：通过阅读文档关于集合的官方使用方法，用自己的话讲，集合用来组织数据，比如键盘可能內建触摸板，因此键盘数据需要保存在一个应用集合里，触摸板数据则保存在另一个应用集合里。可以给每个集合指定“Report ID”，具体将在后面叙述。

可以将USAGE (Mouse)变为USAGE (Gamepad)，让电脑知道这是一个带1个操作杆和3个按键的游戏手柄。将PS2控制器变为USB游戏手柄怎么样？控制器有16个按键和两个拇指棒，数据结构如下：

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Button	Button	Button	Button	Button	Button	Button	Button
Byte 1	Button	Button	Button	Button	Button	Button	Button	Button
Byte 2	Left X Axis as Signed Integer							
Byte 3	Left Y Axis as Signed Integer							
Byte 4	Right X Axis as Signed Integer							
Byte 5	Right Y Axis as Signed Integer							

C语言的数据结构如下：

```
struct gamepad_report_t
{
    uint16_t buttons;
    int8_t left_x;
    int8_t left_y;
    int8_t right_x;
    int8_t right_y;
}
```

让电脑知道这是一个游戏手柄：

```
USAGE_PAGE (Generic Desktop)
USAGE (Game Pad)
COLLECTION (Application)
    COLLECTION (Physical)

    ...

    END COLLECTION
END COLLECTION
```

按键的描述如下：

```
USAGE_PAGE (Button)
USAGE_MINIMUM (Button 1)
USAGE_MAXIMUM (Button 16)
LOGICAL_MINIMUM (0)
LOGICAL_MAXIMUM (1)
REPORT_COUNT (16)
REPORT_SIZE (1)
INPUT (Data,Var,Abs)
```

拇指棒的四个轴：

```
USAGE_PAGE (Generic Desktop)
USAGE (X)
USAGE (Y)
USAGE (Z)
USAGE (Rx)
LOGICAL_MINIMUM (-127)
LOGICAL_MAXIMUM (127)
REPORT_SIZE (8)
REPORT_COUNT (4)
INPUT (Data,Var,Abs)
```

注：Z表示右边拇指棒的X轴，Rx表示右边拇指棒的Y轴。这看起来不合理，但是是目前大多数游戏手柄的工作方式。在《战地：叛逆连队2》上测试是可以工作的。
注：用“absolute”表示类似操作杆的东西，“relative”表示类似鼠标的东西。
结果如下：

```
USAGE_PAGE (Generic Desktop)
USAGE (Game Pad)
COLLECTION (Application)
    COLLECTION (Physical)
        USAGE_PAGE (Button)
        USAGE_MINIMUM (Button 1)
        USAGE_MAXIMUM (Button 16)
        LOGICAL_MINIMUM (0)
        LOGICAL_MAXIMUM (1)
        REPORT_COUNT (16)
        REPORT_SIZE (1)
        INPUT (Data,Var,Abs)
        USAGE_PAGE (Generic Desktop)
        USAGE (X)
        USAGE (Y)
        USAGE (Z)
        USAGE (Rx)
        LOGICAL_MINIMUM (-127)
        LOGICAL_MAXIMUM (127)
        REPORT_SIZE (8)
        REPORT_COUNT (4)
        INPUT (Data,Var,Abs)
    END COLLECTION
END COLLECTION
```

两个玩家呢？这就显示出集合的方便之处：

```
USAGE_PAGE (Generic Desktop)
USAGE (Game Pad)
COLLECTION (Application)
    COLLECTION (Physical)
        REPORT_ID (1)
        ...
    END COLLECTION
END COLLECTION
USAGE_PAGE (Generic Desktop)
USAGE (Game Pad)
COLLECTION (Application)
    COLLECTION (Physical)
        REPORT_ID (2)
        ...
    END COLLECTION
END COLLECTION
```

填充上数据区域，得到：

```
USAGE_PAGE (Generic Desktop)
USAGE (Game Pad)
COLLECTION (Application)
    COLLECTION (Physical)
        REPORT_ID (1)
        USAGE_PAGE (Button)
        USAGE_MINIMUM (Button 1)
        USAGE_MAXIMUM (Button 16)
        LOGICAL_MINIMUM (0)
        LOGICAL_MAXIMUM (1)
        REPORT_COUNT (16)
        REPORT_SIZE (1)
        INPUT (Data,Var,Abs)
        USAGE_PAGE (Generic Desktop)
        USAGE (X)
        USAGE (Y)
        USAGE (Z)
        USAGE (Rx)
        LOGICAL_MINIMUM (-127)
        LOGICAL_MAXIMUM (127)
        REPORT_SIZE (8)
        REPORT_COUNT (4)
        INPUT (Data,Var,Abs)
    END COLLECTION
END COLLECTION
USAGE_PAGE (Generic Desktop)
USAGE (Game Pad)
COLLECTION (Application)
    COLLECTION (Physical)
        REPORT_ID (2)
        USAGE_PAGE (Button)
        USAGE_MINIMUM (Button 1)
        USAGE_MAXIMUM (Button 16)
        LOGICAL_MINIMUM (0)
        LOGICAL_MAXIMUM (1)
        REPORT_COUNT (16)
        REPORT_SIZE (1)
        INPUT (Data,Var,Abs)
        USAGE_PAGE (Generic Desktop)
        USAGE (X)
        USAGE (Y)
        USAGE (Z)
        USAGE (Rx)
        LOGICAL_MINIMUM (-127)
        LOGICAL_MAXIMUM (127)
        REPORT_SIZE (8)
        REPORT_COUNT (4)
        INPUT (Data,Var,Abs)
    END COLLECTION
END COLLECTION
```

这样看起来，在数据结构上加上Report ID真的很重要：

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Report ID							
Byte 1	Button	Button	Button	Button	Button	Button	Button	Button
Byte 2	Button	Button	Button	Button	Button	Button	Button	Button
Byte 3	Left X Axis as Signed Integer							
Byte 4	Left Y Axis as Signed Integer							
Byte 5	Right X Axis as Signed Integer							
Byte 6	Right Y Axis as Signed Integer							

C语言数据结构：

```
struct multiplayer_gamepad_report_t
{
    uint8_t report_id;
    uint16_t buttons;
    int8_t left_x;
    int8_t left_y;
    int8_t right_x;
    int8_t right_y;
}
```

在发送数据到电脑上之前必须手动的设置Report ID，目的是为了让电脑知道数据来自哪个游戏手柄。

```
multiplayer_gamepad_report_t player1_report;
multiplayer_gamepad_report_t player2_report;
player1_report.report_id = 1;
player2_report.report_id = 2;
```

也可以用Collections和Report ID设成综合性设备。因此包含键盘、鼠标和游戏手柄（2）的描述符如下：

```
USAGE_PAGE (Generic Desktop)
USAGE (Keyboard)
COLLECTION (Application)
    REPORT_ID (1)
    ...
END COLLECTION
USAGE_PAGE (Generic Desktop)
USAGE (Mouse)
COLLECTION (Application)
    USAGE (Pointer)
    COLLECTION (Physical)
        REPORT_ID (2)
        ...
    END COLLECTION
END COLLECTION
USAGE_PAGE (Generic Desktop)
USAGE (Game Pad)
COLLECTION (Application)
    COLLECTION (Physical)
        REPORT_ID (3)
        ...
    END COLLECTION
END COLLECTION
USAGE_PAGE (Generic Desktop)
USAGE (Game Pad)
COLLECTION (Application)
    COLLECTION (Physical)
        REPORT_ID (4)
        ...
    END COLLECTION
END COLLECTION
```

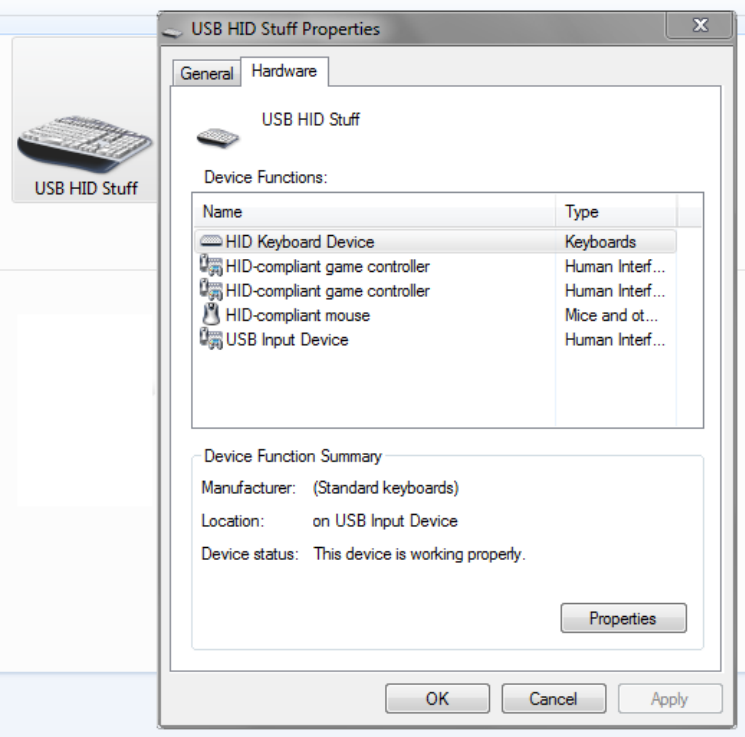
当然数据结构也要增加Report ID：

```
struct keyboard_report_t
{
    uint8_t report_id;
    uint8_t modifier;
    uint8_t reserved;
    uint8_t keycode[6];
}

struct mouse_report_t
{
    uint8_t report_id;
    uint8_t buttons;
    int8_t x;
    int8_t y;
}

struct gamepad_report_t
{
    uint8_t report_id;
    uint16_t buttons;
    int8_t left_x;
    int8_t left_y;
    int8_t right_x;
    int8_t right_y;
}
```

由于改变了数据结构，设备不再支持启动协议，因此不用定义协议。相应的改变usbconfig.h。想要看到这个效果，USnooBie导入工程示例，Windows的“设备和打印”上显示：



编辑于 2017-07-27 17:35