3个界面的HID复合装置的实作(鼠标，键盘，游戏杆)

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	3 个界面的 HID 复合装置的实作(鼠标，键盘，游戏杆)
BSP 版本	NUC123 Series BSP CMSIS V3.01.001
开发平台	NuTiny-EVB-NUC123-LQFP64 v1.0

1 功能介绍

1.1 简介

该范例示范3组接口的HID复合装置实作。每个界面拥有一项功能，分别是鼠标，键盘和游戏杆。此三个功能可以使用GPIO触发。

1.2 原理

此范例程序复合装置有3组界面，每个接口使用HID(Human Interface Devices)标准实作一个功能。

一个UBS装置一定会有一个控制管线(端点0)，用于控制传输。其他端点可以根据应用而有不同的设定。此范例中的控制管线由装置端硬件的的端点0和端点1构成，从USB主机的角度看来，只会看到端点0；一个接口运行一个欲展示的功能，每个接口包含一个中断的输入端点，此端点用于输入方向的中断传输(Interrupt Transfer)。这些接口和端点的关系列于表 1。

Endpoints in NUC123	**Composite Device**	**Function**
Endpoint 0 Control IN **Endpoint 1 Control OUT**	Control Pipe (Endpoint 0)	Control Transfer
Endpoint 2	Interface 0: Endpoint 1: Interrupt IN	HID Mouse
Endpoint 3	Interface 1 Endpoint 2: Interrupt IN	HID Keyboard
Endpoint 4	Interface 2 Endpoint 3: Interrupt IN	HID Joystick

表 1: HID 复合装置的接口和端点组态

接口0是一个HID鼠标装置，使用表 2的数据格式回传数据。

Byte	**Bits**	**Description**
0	0	Button 1
0	1	Button 2
0	2	Button 3
0	3 ~ 7	Padding
1	0 ~ 7	X-axis
2	0 ~ 7	Y-axis
3	0 ~ 7	Wheel

表 2: HID 鼠标 报告格式

接口1是一个HID键盘装置，使用表 3的数据格式回传数据。键盘代码的定义，可从USB-IF下载的 " HID USAGE TABLE V1.12"的第53页找到。

Byte	**Description**
0	Modifier Keys
1	Reserved

2	Key code 1
3	Key code 2
4	Key code 3
5	Key code 4
6	Key code 5
7	Key code 6

表 3: HID 键盘报告格式

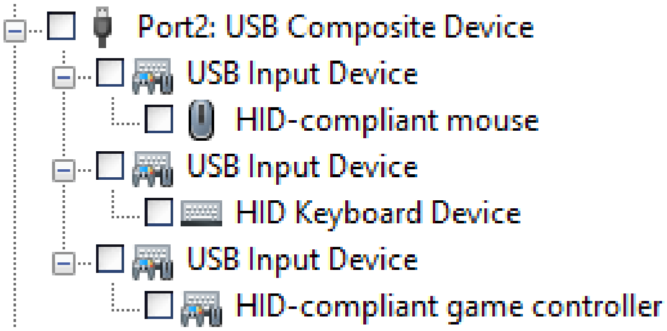
接口2是HID游戏杆装置，使用表 4的数据格式回传数据。

Byte	Bits	Description
0	0 ~ 7	Throttle
1	0 ~ 7	X axis
2	0 ~ 7	Y axis
3	0 ~ 3	Hat Switch
3	4 ~ 7	Button 1 ~ 4

表 4:HID 游戏杆报告格式

1.3 执行结果

此范例需与连接个人计算机才能运行。此范例程序执行时，使用USBLyzer我们可以获得图 1 的装置列表。可以看到此复合装置有3组接口。



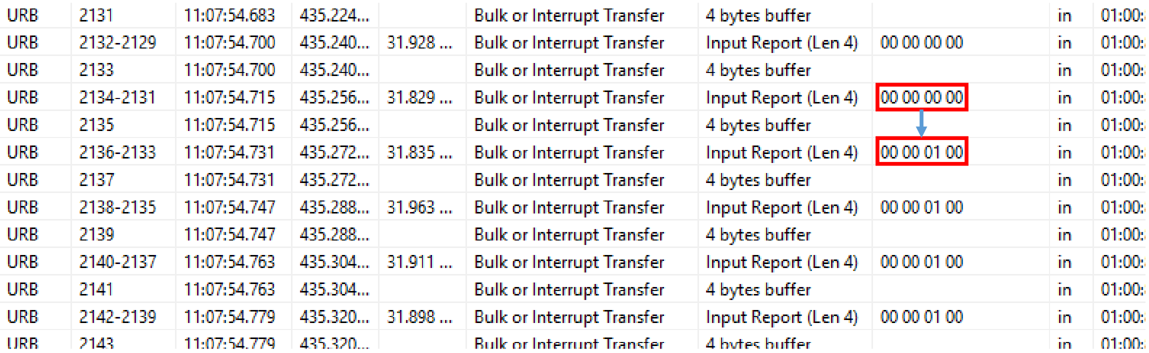
□ Port2: USB Composite Device
  □ USB Input Device
    □ HID-compliant mouse
  □ USB Input Device
    □ HID Keyboard Device
  □ USB Input Device
    □ HID-compliant game controller

图 1:装置列表

USBLyzer 也可用于传输封包的纪录。图 2、图 3 和图 5 记录了此范例程序运行时的传输封包。

1.3.1 HID 鼠标

若GPIO PB7 准位为低准位时(仿真按钮按下)，可以看见Y轴的字段(见表 2)的传输数值变成1。计算机鼠标的指针会往下移动。

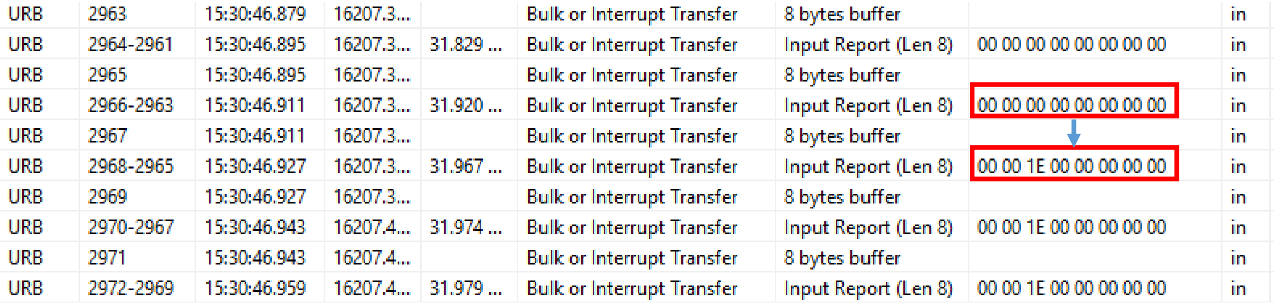


URB	2131	11:07:54.683	435.224...		Bulk or Interrupt Transfer	4 bytes buffer		in	01:00:
URB	2132-2129	11:07:54.700	435.240...	31.928 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	00 00 00 00	in	01:00:
URB	2133	11:07:54.700	435.240...		Bulk or Interrupt Transfer	4 bytes buffer		in	01:00:
URB	2134-2131	11:07:54.715	435.256...	31.829 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	00 00 00 00	in	01:00:
URB	2135	11:07:54.715	435.256...		Bulk or Interrupt Transfer	4 bytes buffer		in	01:00:
URB	2136-2133	11:07:54.731	435.272...	31.835 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	00 00 01 00	in	01:00:
URB	2137	11:07:54.731	435.272...		Bulk or Interrupt Transfer	4 bytes buffer		in	01:00:
URB	2138-2135	11:07:54.747	435.288...	31.963 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	00 00 01 00	in	01:00:
URB	2139	11:07:54.747	435.288...		Bulk or Interrupt Transfer	4 bytes buffer		in	01:00:
URB	2140-2137	11:07:54.763	435.304...	31.911 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	00 00 01 00	in	01:00:
URB	2141	11:07:54.763	435.304...		Bulk or Interrupt Transfer	4 bytes buffer		in	01:00:
URB	2142-2139	11:07:54.779	435.320...	31.898 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	00 00 01 00	in	01:00:
URB	2143	11:07:54.779	435.320...		Bulk or Interrupt Transfer	4 bytes buffer		in	01:00:

图 2:鼠标的传输封包

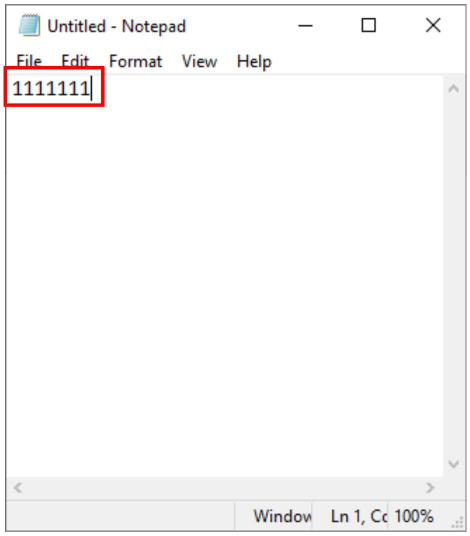
1.3.2 HID 键盘

若GPIO PB6 准位为低准位时(仿真按钮按下)，键盘码的第一个字段的传输数值由0变成0x1E。此时这装置可在记事本或计算机可输入的地方输入的数字’1’，如图 4。0x1E这个数值是在” HID USAGE TABLE V1.12”中定义的字符’1’。



URB	2963	15:30:46.879	16207.3...		Bulk or Interrupt Transfer	8 bytes buffer		in
URB	2964-2961	15:30:46.895	16207.3...	31.829 ...	Bulk or Interrupt Transfer	Input Report (Len 8)	00 00 00 00 00 00 00 00	in
URB	2965	15:30:46.895	16207.3...		Bulk or Interrupt Transfer	8 bytes buffer		in
URB	2966-2963	15:30:46.911	16207.3...	31.920 ...	Bulk or Interrupt Transfer	Input Report (Len 8)	00 00 00 00 00 00 00 00	in
URB	2967	15:30:46.911	16207.3...		Bulk or Interrupt Transfer	8 bytes buffer		in
URB	2968-2965	15:30:46.927	16207.3...	31.967 ...	Bulk or Interrupt Transfer	Input Report (Len 8)	00 00 1E 00 00 00 00 00	in
URB	2969	15:30:46.927	16207.3...		Bulk or Interrupt Transfer	8 bytes buffer		in
URB	2970-2967	15:30:46.943	16207.4...	31.974 ...	Bulk or Interrupt Transfer	Input Report (Len 8)	00 00 1E 00 00 00 00 00	in
URB	2971	15:30:46.943	16207.4...		Bulk or Interrupt Transfer	8 bytes buffer		in
URB	2972-2969	15:30:46.959	16207.4...	31.979 ...	Bulk or Interrupt Transfer	Input Report (Len 8)	00 00 1E 00 00 00 00 00	in

图 3:键盘的传输封包



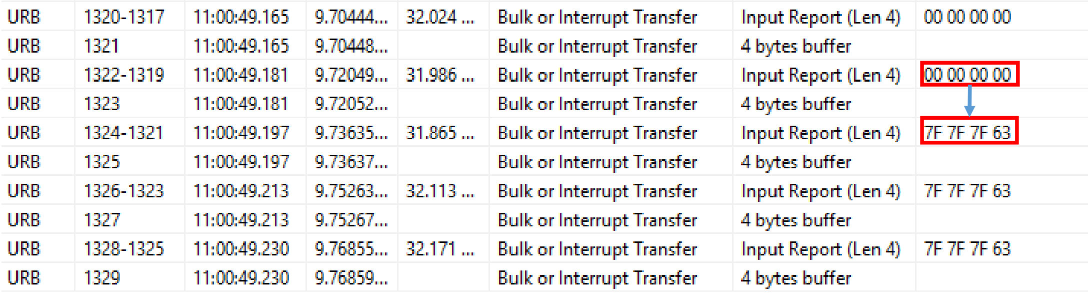
1111111

图 4:输入文字到记事本

1.3.3 HID 游戏杆

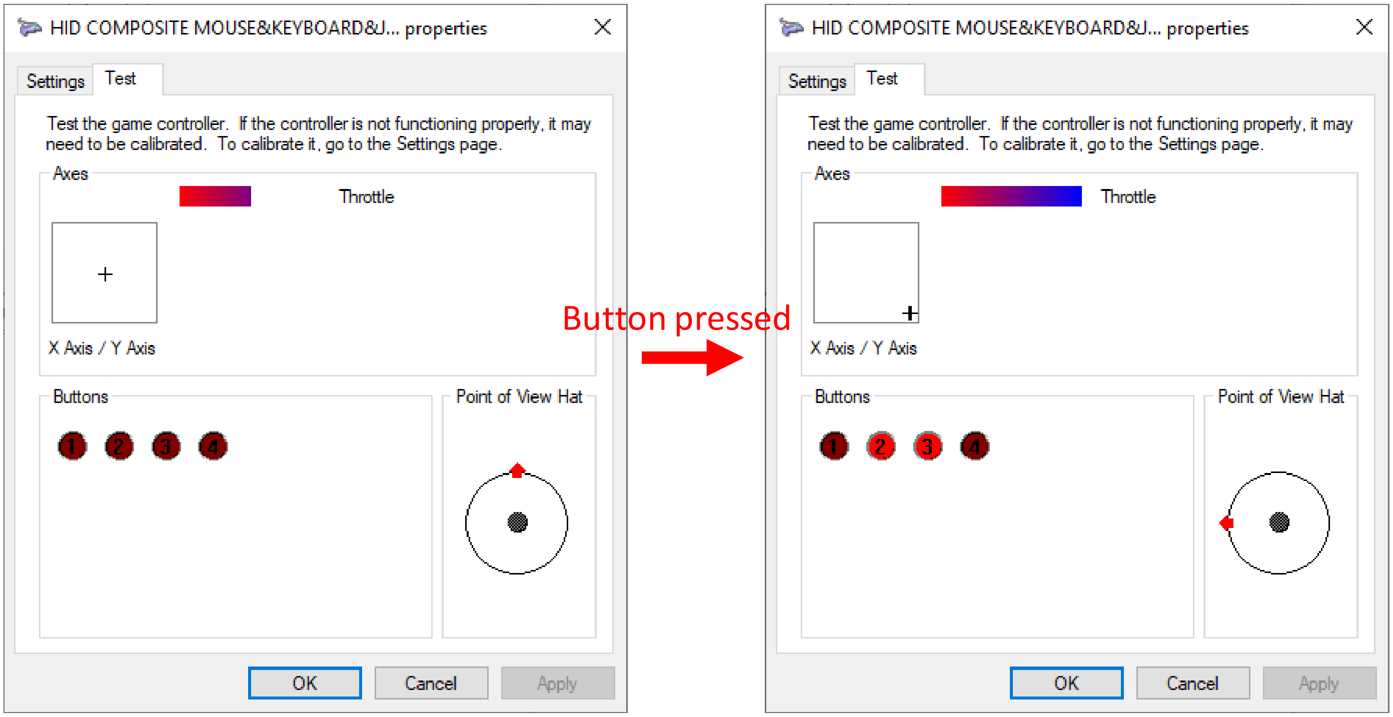
图 5中，当GPIO PB5拉至低准位，可观察到表 4中的传输数据改变。数据的改变，也可以从图 6观察到对应的控件目变更。

使用者可用Windows 内建的"游戏控制器”，观察游戏杆的行为。开启图 6窗口的步骤，可以参考微软公司的教学，进入此装置的选项(图 7)，找到此装置的属性选项，点选并切换到测试页，就可以进入图 6的测试页面。



URB	1320-1317	11:00:49.165	9.70444...	32.024 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	00 00 00 00
URB	1321	11:00:49.165	9.70448...		Bulk or Interrupt Transfer	4 bytes buffer	
URB	1322-1319	11:00:49.181	9.72049...	31.986 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	00 00 00 00
URB	1323	11:00:49.181	9.72052...		Bulk or Interrupt Transfer	4 bytes buffer	
URB	1324-1321	11:00:49.197	9.73635...	31.865 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	7F 7F 7F 63
URB	1325	11:00:49.197	9.73637...		Bulk or Interrupt Transfer	4 bytes buffer	
URB	1326-1323	11:00:49.213	9.75263...	32.113 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	7F 7F 7F 63
URB	1327	11:00:49.213	9.75267...		Bulk or Interrupt Transfer	4 bytes buffer	
URB	1328-1325	11:00:49.230	9.76855...	32.171 ...	Bulk or Interrupt Transfer	Input Report (Len 4)	7F 7F 7F 63
URB	1329	11:00:49.230	9.76859...		Bulk or Interrupt Transfer	4 bytes buffer	

图 5: 键盘的传输封包



Test the game controller. If the controller is not functioning properly, it may need to be calibrated. To calibrate it, go to the Settings page.

Test the game controller. If the controller is not functioning properly, it may need to be calibrated. To calibrate it, go to the Settings page.

图 6:游戏杆的测试

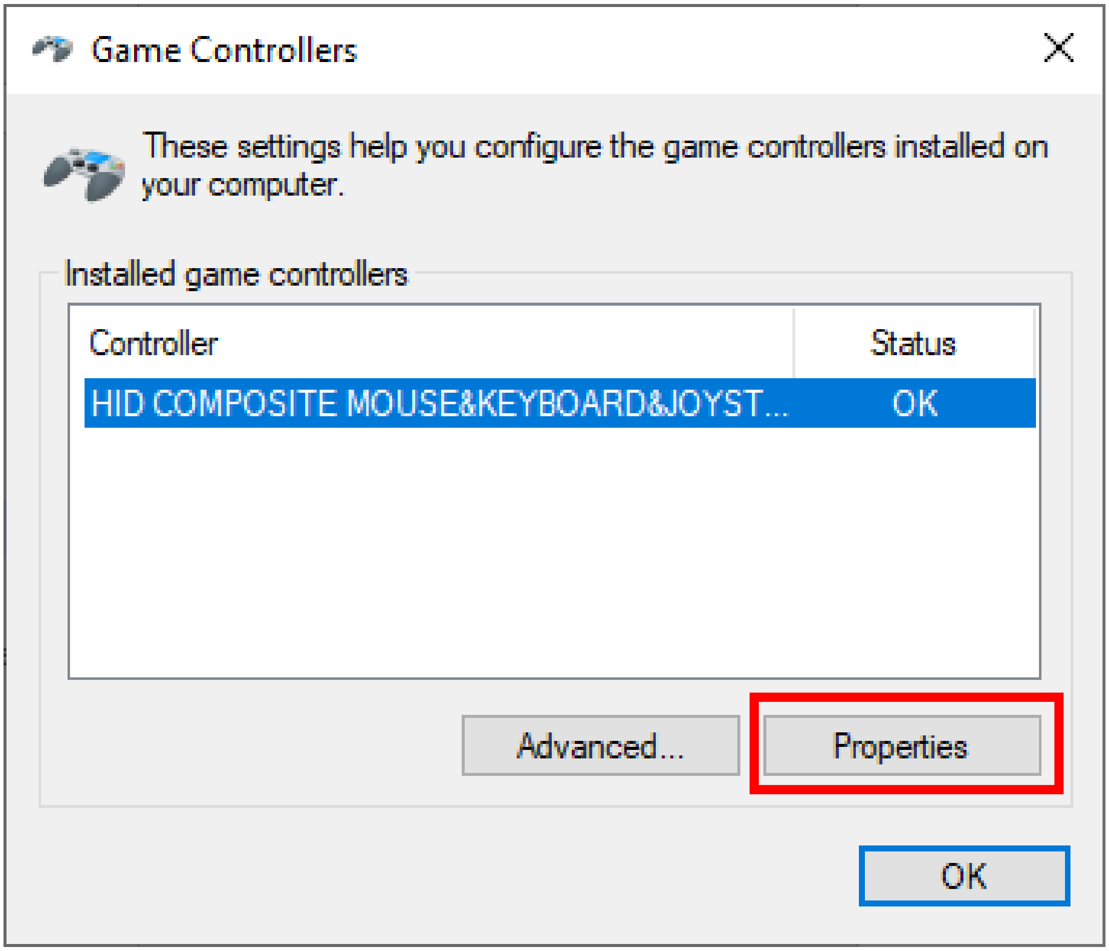
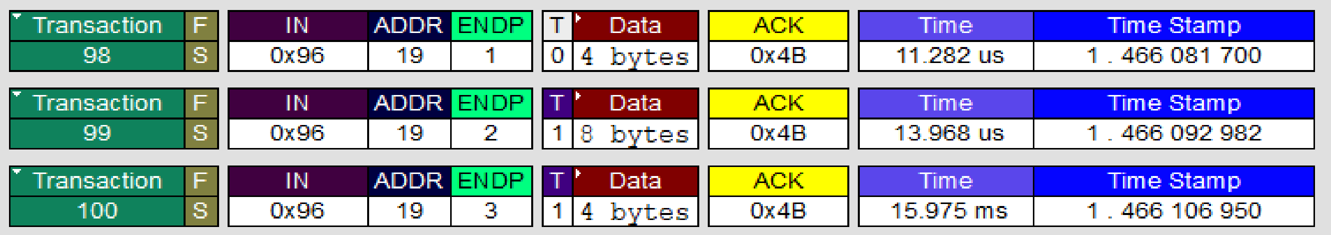

Game Controllers
These settings help you configure the game controllers installed on your computer.
Installed game controllers
Controller
Status
HID COMPOSITE MOUSE&KEYBOARD&JOYST...
OK
Advanced...
Properties
OK

图 7: 游戏控制选项

2 代码介绍

此范例程序使用 HID 标准实现了三个功能接口，鼠标、键盘和游戏杆的功能。这三个功能传输时的处理流程和接口非常类似。每个接口有一个 IN 的中断传输端点。当主机发出 IN TOKEN 时，每个 IN 的中断传输端点会使用定义于 HID report descriptor 的数据格式回传数据，如图 8。2.1 节将介绍程序处理的流程。



Transaction	F S	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
98	S	0x96	19	1	0	4 bytes	0x4B	11.282 us	1 . 466 081 700
99	S	0x96	19	2	1	8 bytes	0x4B	13.968 us	1 . 466 092 982
100	S	0x96	19	3	1	4 bytes	0x4B	15.975 ms	1 . 466 106 950

图 8: 3 个 IN 中断端点的 transaction

2.1 鼠标、键盘和游戏杆

2.1.1 Main.c

While回圈内的3个子程序用于数据的传输。每个子程序皆可独立运行，所以使用者可保留其中需要的程序就可以进行测试评估。

```
    while (1)
    {
        HID_UpdateMouseData();
        HID_UpdateKeyboardData();
        HID_UpdateJoystickData();
    }
```

2.1.2 HID_Composite_Device_Mouse_Keyboard_Joystick.c

以鼠标为例，USB主机会发出IN Token向端点1请求数据。当IN Token收到时，端点2的中断处理程序会把g_u8EP2Ready设为真。当HID_UpdateMouseData()检查到g_u8EP2Ready为真时，就会设定并回传数据。当GPIO PB5电位为0时，设定buf[2] (表 2的Y轴字段)为0x1。

键盘和游戏杆的HID_UpdateKeyboardData() 和HID_UpdateJoystickData()子程序的处理流程，与HID_UpdateMouseData()几乎相同。只有端点数字和缓冲区的设定有些不同。键盘使用的IN中断传输使用端点3，缓冲区使用8个字节的长度；而游戏杆使用中断端点4，和4字节的缓冲区。端点和缓冲区的关系，整理于表 1。

```
void HID_UpdateMouseData(void)
{
    uint8_t *buf;

    if (g_u8EP2Ready)
    {
        buf = (uint8_t *)(USBD_BUF_BASE + USBD_GET_EP_BUF_ADDR(EP2));
        g_u8EP2Ready = 0; /* Clear flag */

        buf[0] = buf[1] = buf[2] = buf[3] = 0;
        if (PB5 == 0)
        {
            buf[0] = 0x00; /* Button*/
            buf[1] = 0x00; /* X      */
            buf[2] = 0x01; /* Y      */
            buf[3] = 0x00; /* wheel */

        }


        /* Set transfer length and trigger IN transfer */
        USBD_SET_PAYLOAD_LEN(EP2, 4);
    }
}
```

2.1.3 Descriptor.c

HID_MouseReportDescriptor是鼠标的HID报告描述符元。此描述元定义了表 2中的数据格式。此数据格式内有，3个按钮、X轴、Y轴和滚轮。当关联的按钮按下后，从图 2可以看到对应的字段，传输资料改变。

```
/*!<USB HID Report Descriptor */
const uint8_t HID_MouseReportDescriptor[] =
{
    0x05, 0x01,     /* Usage Page(Generic Desktop Controls) */
    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(0x1) */
    0x29, 0x03,             /* Usage Maximum(0x3) */
    0x15, 0x00,             /* Logical Minimum(0x0) */
    0x25, 0x01,             /* Logical Maximum(0x1) */
    0x75, 0x01,             /* Report Size(0x1) */
    0x95, 0x03,             /* Report Count(0x3) */
    0x81, 0x02,             /* Input(3 button bit) */
    0x75, 0x05,             /* Report Size(0x5) */
    0x95, 0x01,             /* Report Count(0x1) */
    0x81, 0x01,             /* Input(5 bit padding) */
    0x05, 0x01,             /* Usage Page(Generic Desktop Controls) */
    0x09, 0x30,             /* Usage(X) */
    0x09, 0x31,             /* Usage(Y) */
    0x09, 0x38,             /* Usage(Wheel) */
    0x15, 0x81,             /* Logical Minimum(0x81)(-127) */
    0x25, 0x7F,             /* Logical Maximum(0x7F)(127) */
    0x75, 0x08,             /* Report Size(0x8) */
    0x95, 0x03,             /* Report Count(0x3) */
    0x81, 0x06,             /* Input(1 byte wheel) */
    0xC0,               /* End Collection */
    0xC0,           /* End Collection */
};
```

2.2 Configuration Descriptor for 3 Interfaces Composite Device

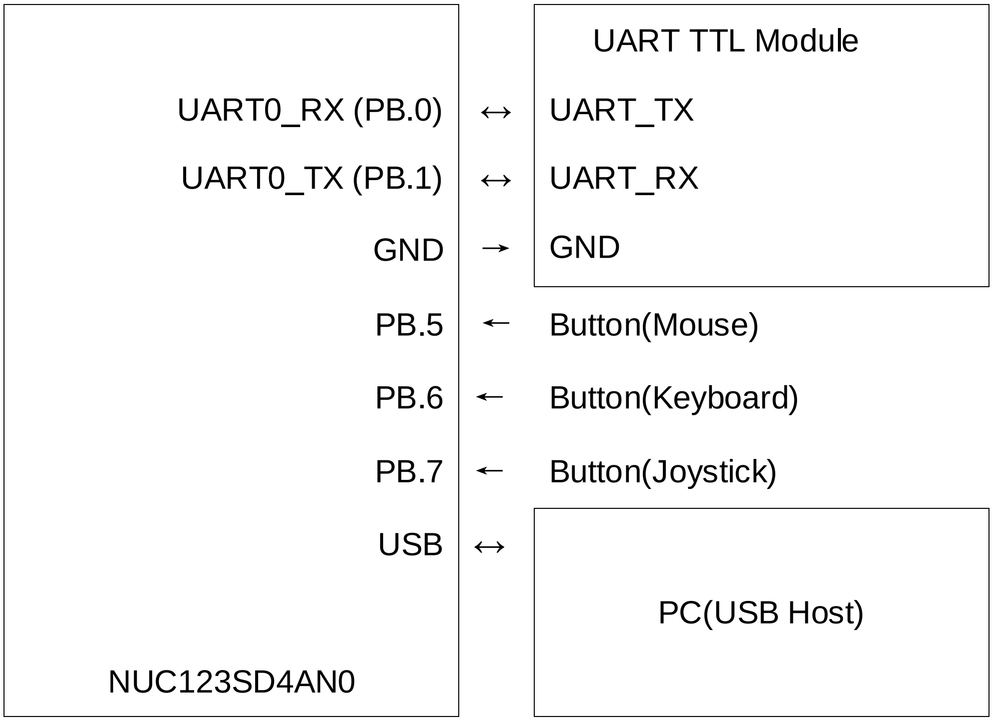
USB主机需要组态描述元(Configuration Descriptor)来辨识装置。gu8ConfigDescriptor是此范例使用的组态描述元，里面含有3组界面的描述。“bNumInterfaces”字段设定为3，用来让USB主机知道此复合装置有3组接口。3组界面描述元(Interface Descriptor)，HID描述元(HID descriptor)和端点描述元(Endpoint Descriptor)，依序列于组态描述元之后。

```
/*!<USB Configure Descriptor */
const uint8_t gu8ConfigDescriptor[] =
{
    LEN_CONFIG,     /* bLength */
    DESC_CONFIG,    /* bDescriptorType */
    /* wTotalLength */
    LEN_CONFIG_AND_SUBORDINATE & 0x00FF,
    (LEN_CONFIG_AND_SUBORDINATE & 0xFF00) >> 8,
    0x03,           /* bNumInterfaces */
    0x01,           /* bConfigurationValue */
    0x00,           /* iConfiguration */
    0x80 | (USBD_SELF_POWERED << 6) | (USBD_REMOTE_WAKEUP << 5),/* bmAttributes */
    USBD_MAX_POWER,         /* MaxPower */

    /* I/F descr: HID */
    LEN_INTERFACE,  /* bLength */
    DESC_INTERFACE, /* bDescriptorType */
    0x00,           /* bInterfaceNumber */
    0x00,           /* bAlternateSetting */
    0x01,           /* bNumEndpoints */
    0x03,           /* bInterfaceClass */
    0x00,           /* bInterfaceSubClass */
    HID_MOUSE,      /* bInterfaceProtocol */
    0x00,           /* iInterface */

    /* HID Descriptor */
    LEN_HID,        /* Size of this descriptor in UINT8s. */
    DESC_HID,       /* HID descriptor type. */
    0x10, 0x01,     /* HID Class Spec. release number. */
    0x00,           /* H/W target country. */
    0x01,           /* Number of HID class descriptors to follow. */
    DESC_HID_RPT,   /* Dscriptor type. */
    /* Total length of report descriptor. */
    sizeof(HID_MouseReportDescriptor) & 0x00FF,
    (sizeof(HID_MouseReportDescriptor) & 0xFF00) >> 8,

    /* EP Descriptor: interrupt in. */
    LEN_ENDPOINT,   /* bLength */
    DESC_ENDPOINT,  /* bDescriptorType */
    (INT_IN_EP_NUM | EP_INPUT), /* bEndpointAddress */
    EP_INT,         /* bmAttributes */
    /* wMaxPacketSize */
    EP2_MAX_PKT_SIZE & 0x00FF,
    (EP2_MAX_PKT_SIZE & 0xFF00) >> 8,
    HID_DEFAULT_INT_IN_INTERVAL,     /* bInterval */
    /* I/F descr: HID */
    LEN_INTERFACE,  /* bLength */
    DESC_INTERFACE, /* bDescriptorType */
    0x01,           /* bInterfaceNumber */
    0x00,           /* bAlternateSetting */
    0x01,           /* bNumEndpoints */
    0x03,           /* bInterfaceClass */
    0x00,           /* bInterfaceSubClass */
    HID_KEYBOARD,   /* bInterfaceProtocol */
    0x00,           /* iInterface */

    /* HID Descriptor */
    LEN_HID,        /* Size of this descriptor in UINT8s. */
    DESC_HID,       /* HID descriptor type. */
    0x10, 0x01,     /* HID Class Spec. release number. */
    0x00,           /* H/W target country. */
    0x01,           /* Number of HID class descriptors to follow. */
    DESC_HID_RPT,   /* Dscriptor type. */
    /* Total length of report descriptor. */
    sizeof(HID_KeyboardReportDescriptor) & 0x00FF,
    (sizeof(HID_KeyboardReportDescriptor) & 0xFF00) >> 8,

    /* EP Descriptor: interrupt in. */
    LEN_ENDPOINT,   /* bLength */
    DESC_ENDPOINT,  /* bDescriptorType */
    (INT_IN1_EP_NUM | EP_INPUT), /* bEndpointAddress */
    EP_INT,         /* bmAttributes */
    /* wMaxPacketSize */
    EP3_MAX_PKT_SIZE & 0x00FF,
    (EP3_MAX_PKT_SIZE & 0xFF00) >> 8,
    HID_DEFAULT_INT_IN_INTERVAL,     /* bInterval */

    /* I/F descr: HID */
    LEN_INTERFACE,  /* bLength */
    DESC_INTERFACE, /* bDescriptorType */
    0x02,           /* bInterfaceNumber */
    0x00,           /* bAlternateSetting */
    0x01,           /* bNumEndpoints */
    0x03,           /* bInterfaceClass */
    0x00,           /* bInterfaceSubClass */
    HID_NONE,       /* bInterfaceProtocol */
    0x00,           /* iInterface */

    /* HID Descriptor */
    LEN_HID,        /* Size of this descriptor in UINT8s. */
    DESC_HID,       /* HID descriptor type. */
    0x10, 0x01,     /* HID Class Spec. release number. */
    0x00,           /* H/W target country. */
    0x01,           /* Number of HID class descriptors to follow. */
    DESC_HID_RPT,   /* Dscriptor type. */
    /* Total length of report descriptor. */
    sizeof(HID_JoytickReportDescriptor) & 0x00FF,
    (sizeof(HID_JoytickReportDescriptor) & 0xFF00) >> 8,

    /* EP Descriptor: interrupt in. */
    LEN_ENDPOINT,   /* bLength */
    DESC_ENDPOINT,  /* bDescriptorType */
    (INT_IN2_EP_NUM | EP_INPUT), /* bEndpointAddress */
    EP_INT,         /* bmAttributes */
    /* wMaxPacketSize */
    EP4_MAX_PKT_SIZE & 0x00FF,
    (EP4_MAX_PKT_SIZE & 0xFF00) >> 8,
    HID_DEFAULT_INT_IN_INTERVAL     /* bInterval */
};
```

3 软件与硬件环境

- **软件环境**
  - BSP 版本
    - NUC123 Series BSP CMSIS V3.01.001
  - IDE 版本
    - Keil uVersion 5.26
  - USB 封包撷取工具
    - USBLyzer (http://www.usblyzer.com/)
- **硬件环境**
  - 电路组件
    - NuTiny-EVB-NUC123-LQFP64 v1.0
    - USB mini USB cable
    - USB-UART TTL(Optional)

- 示意图


UART TTL Module
UART0_RX (PB.0) ↔ UART_TX
UART0_TX (PB.1) ↔ UART_RX
GND → GND
PB.5 ← Button(Mouse)
PB.6 ← Button(Keyboard)
PB.7 ← Button(Joystick)
USB ↔
PC(USB Host)
NUC123SD4AN0

4 目录信息

🗁 EC_NUC123_USBD_HID_Composite_Mouse_Keyboard_Joystick_V1.00	
🗁 Library	Sample code header and source files
🗁 CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
🗁 Device	CMSIS compliant device header file
🗁 StdDriver	All peripheral driver header and source files
🗁 SampleCode	
🗁 ExampleCode	Source file of example code
🗁 KEIL	KEIL project file

5 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 NUC123_USBD_HID_Composite_Mouse_Keyboard_Joystick.uvproj。
2. 进入编译模式接口
    a. 编译
    b. 下载代码至内存
    c. 进入 / 离开除错模式
3. 进入除错模式接口
    a. 执行代码

6 修订纪录

Date	Revision	Description
July. 01, 2019	1.00	1. 初始发布.