3個界面的HID複合裝置的實作(滑鼠，鍵盤，搖桿)

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	3 個界面的 HID 複合裝置的實作(滑鼠，鍵盤，搖桿)
BSP 版本	NUC123 Series BSP CMSIS V3.01.001
開發平台	NuTiny-EVB-NUC123-LQFP64 v1.0

1 功能介紹

1.1 簡介

該範例示範3組介面的HID複合裝置實作。每個界面擁有一項功能，分別是滑鼠，鍵盤和搖桿。此三個功能可以使用GPIO觸發。

1.2 原理

此範例程式複合裝置有3組界面，每個介面使用HID(Human Interface Devices)標準實作一個功能。

一個UBS裝置一定會有一個控制管線(端點0)，用於控制傳輸。其他端點可以根據應用而有不同的設定。此範例中的控制管線由裝置端硬體的的端點0和端點1構成，從USB主機的角度看來，只會看到端點0；一個介面運行一個欲展示的功能，每個介面包含一個中斷的輸入端點，此端點用於輸入方向的中斷傳輸(Interrupt Transfer)。這些介面和端點的關係列於表 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 複合裝置的介面和端點組態

介面0是一個HID滑鼠裝置，使用表 2的資料格式回傳資料。

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 滑鼠 報告格式

介面1是一個HID鍵盤裝置，使用表 3的資料格式回傳資料。鍵盤代碼的定義，可從USB-IF下載

的 ” HID USAGE TABLE V1.12”的第53頁找到。

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 鍵盤報告格式

介面2是HID搖桿裝置，使用表 4的資料格式回傳資料。

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 搖桿報告格式

1.3 執行結果

此範例需與連接個人電腦才能運行。此範例程式執行時，使用USBLyzer我們可以獲得圖 1的裝置列表。可以看到此複合裝置有3組介面。

圖 1:裝置列表

USBLyzer 也可用於傳輸封包的紀錄。圖 2、圖 3 和圖 5 記錄了此範例程式運行時的傳輸封包。

1.3.1 HID 滑鼠

若GPIO PB7 準位為低準位時(模擬按鈕按下)，可以看見Y軸的欄位(見表 2)的傳輸數值變成1。電腦滑鼠的指標會往下移動。

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:

圖 2:滑鼠的傳輸封包

1.3.2 HID 鍵盤

若GPIO PB6 準位為低準位時(模擬按鈕按下)，鍵盤碼的第一個欄位的傳輸數值由0變成0x1E。此時這裝置可在記事本或電腦可輸入的地方輸入的數字'1'，如圖 4。0x1E這個數值是在" HID USAGE TABLE V1.12"中定義的字元'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

圖 3:鍵盤的傳輸封包

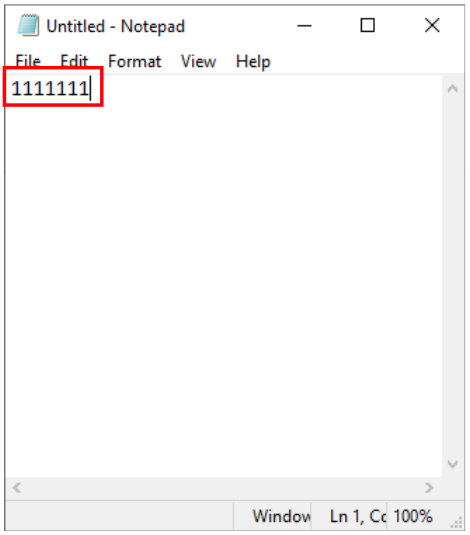


圖 4:輸入文字到記事本

1.3.3 HID 搖桿

圖 5中，當GPIO PB5拉至低準位，可觀察到表 4中的傳輸資料改變。資料的改變，也可以從圖 6觀察到對應的控制項目變更。

使用者可用Windows 內建的"遊戲控制器”，觀察搖桿的行為。開啟圖 6視窗的步驟，可以參考微軟公司的教學，進入此裝置的選項(圖 7)，找到此裝置的屬性選項，點選並切換到測試頁，就可以進入圖 6的測試頁面。

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	

圖 5: 鍵盤的傳輸封包

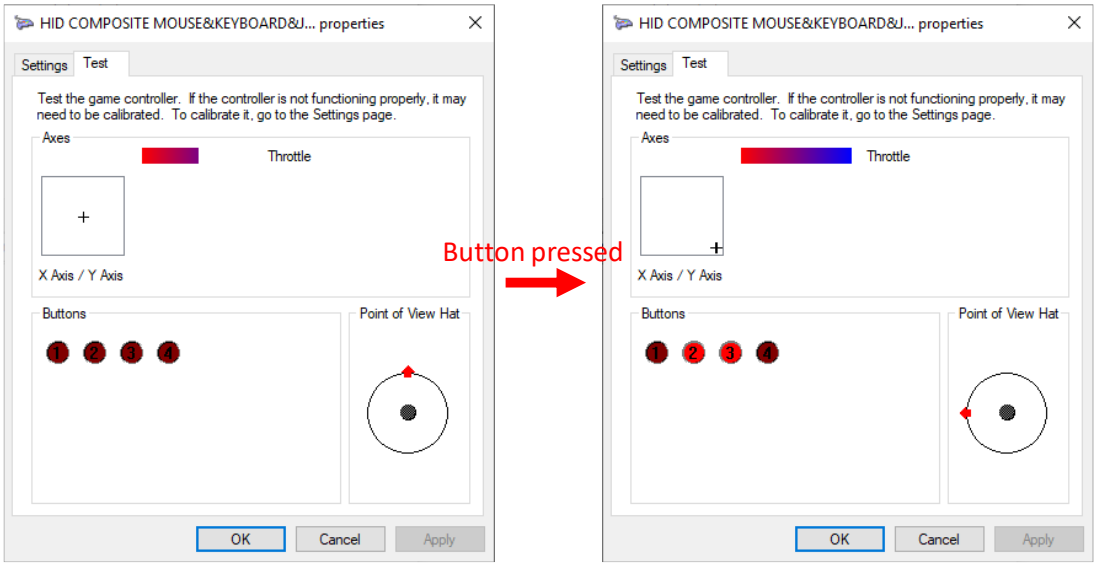


圖 6:搖桿的測試

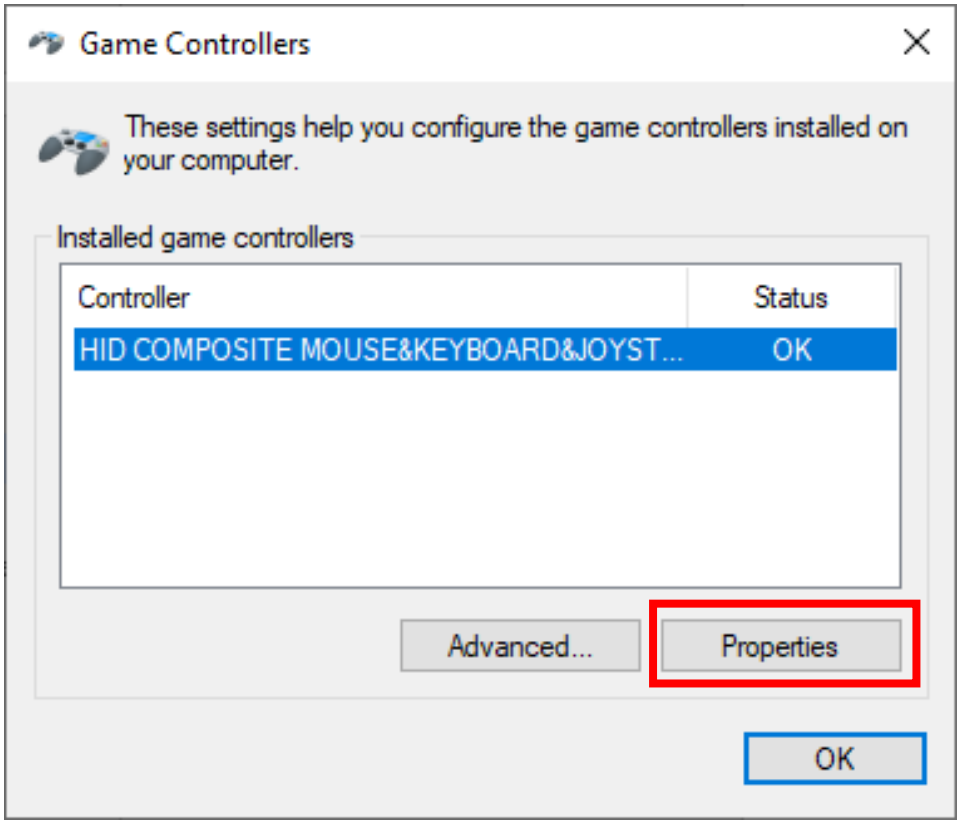
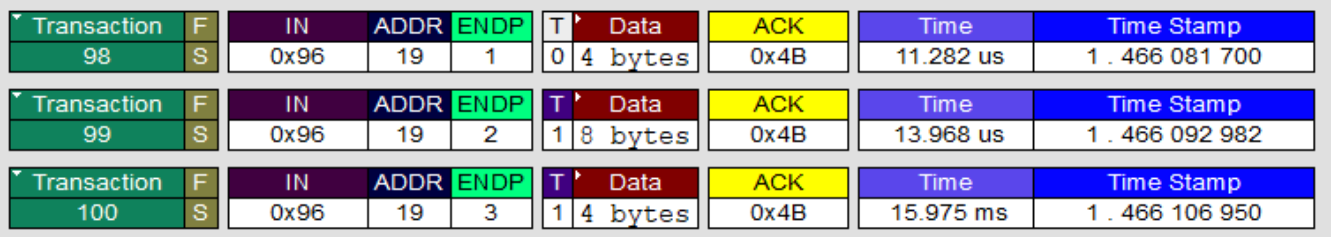


圖 7: 遊戲控制選項

2 代碼介紹

此範例程式使用 HID 標準實現了三個功能介面，滑鼠、鍵盤和搖桿的功能。這三個功能傳輸時的處理流程和介面非常類似。每個介面有一個 IN 的中斷傳輸端點。當主機發出 IN TOKEN 時，每個 IN 的中斷傳輸端點會使用定義於 HID report descriptor 的資料格式回傳資料，如圖 8。2.1 節將介紹程式處理的流程。



Transaction	F	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		0x96	19	2	1	8 bytes	0x4B	13.968 us	1 . 466 092 982
100		0x96	19	3	1	4 bytes	0x4B	15.975 ms	1 . 466 106 950

圖 8: 3 個 IN 中斷端點的 transaction

2.1 滑鼠、鍵盤和搖桿

2.1.1 Main.c

While回圈內的3個副程式用於資料的傳輸。每個副程式皆可獨立運行，所以使用者可保留其中需要的程式就可以進行測試評估。

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

2.1.2 HID_Composite_Device_Mouse_Keyboard_Joystick.c

以滑鼠為例，USB主機會發出IN Token向端點1請求資料。當IN Token收到時，端點2的中斷處理程序會把g_u8EP2Ready設為真。當HID_UpdateMouseData()檢查到g_u8EP2Ready為真時，就會設定並回傳資料。當GPIO PB5電位為0時，設定buf[2] (表 2的Y軸欄位)為0x1。

鍵盤和搖桿的HID_UpdateKeyboardData() 和HID_UpdateJoystickData()副程式的處理流程，與HID_UpdateMouseData()幾乎相同。只有端點數字和緩衝區的設定有些不同。鍵盤使用的IN中斷傳輸使用端點3，緩衝區使用8個字節的長度；而搖桿使用中斷端點4，和4字節的緩衝區。端點和緩衝區的關係，整理於表 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是滑鼠的HID報告描述符元。此描述元定義了表 2中的資料格式。此資料格式內有，3個按鈕、X軸、Y軸和滾輪。當關聯的按鈕按下後，從圖 2可以看到對應的欄位，傳輸資料改變。

```
/*!<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主機需要組態描述元(Configuration Descriptor)來辨識裝置。gu8ConfigDescriptor是此範例使用的組態描述元，裡面含有3組介面的描述。“bNumInterfaces”欄位設定為3，用來讓USB主機知道此複合裝置有3組介面。3組介面描述元(Interface Descriptor)，HID描述元(HID descriptor)和端點描述元(Endpoint Descriptor)，依序列於組態描述元之後。

```
/*!<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,
```

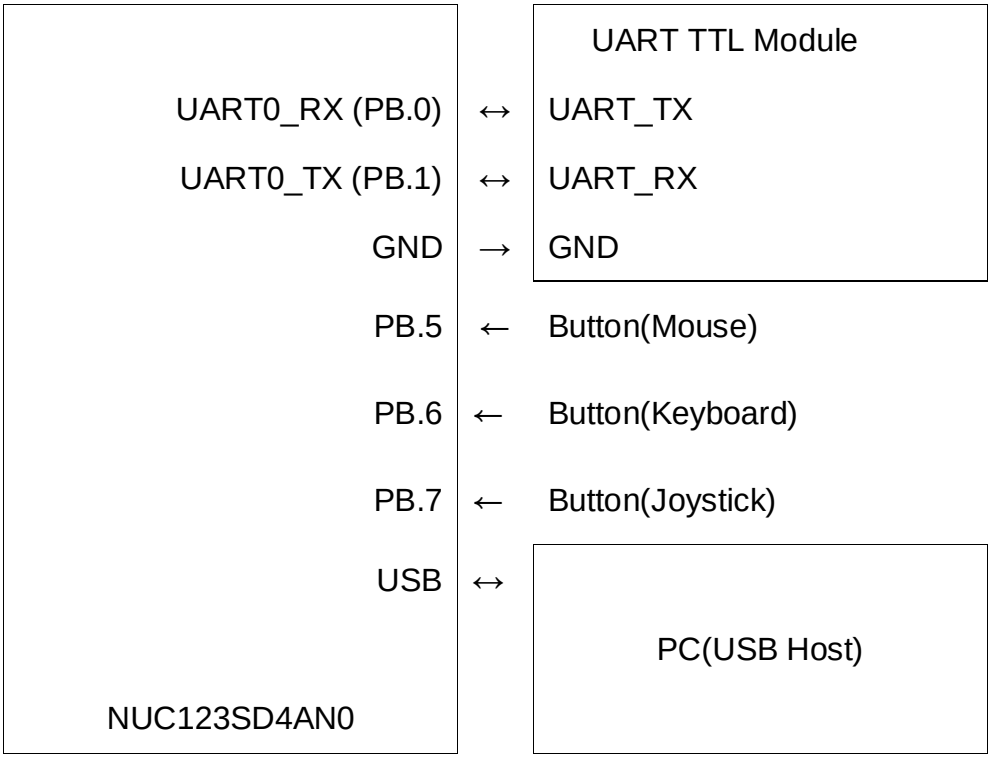
```
    /* EP Descriptor: interrupt in. */
    LEN_ENDPOINT,   /* bLength */
    DESC_ENDPOINT,  /* bDescriptorType */
    (INT_IN_EP_NUM | EP_INPUT), /* bEndpointAddress */
    EP_INT,         /* bmAttributes */
    /* wMaxPacketSize */
    EP2_MAX_PKT_SIZE & 0x00FF,
    (EP2_MAX_PKT_SIZE & 0xFF00) >> 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 軟體與硬體環境

- **軟體環境**
  - BSP 版本
    - NUC123 Series BSP CMSIS V3.01.001
  - IDE 版本
    - Keil uVersion 5.26
  - USB 封包擷取工具
    - USBLyzer (http://www.usblyzer.com/)
- **硬體環境**
  - 電路元件
    - NuTiny-EVB-NUC123-LQFP64 v1.0
    - USB mini USB cable
    - USB-UART TTL(Optional)

■ 示意圖


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 目錄資訊

- 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 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 NUC123_USBD_HID_Composite_Mouse_Keyboard_Joystick.uvproj。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 進入除錯模式介面
   a. 執行代碼

6 修訂紀錄

Date	Revision	Description
July. 01, 2019	1.00	1. 初始發佈.