

HID複合裝置的實作(滑鼠，鍵盤，多媒體鍵)與鍵盤燈號狀態控制

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	HID 複合裝置的實作(滑鼠，鍵盤，多媒體鍵)與鍵盤燈號狀態控制
BSP 版本	NUC122 Series BSP CMSIS V3.00.003
開發平台	NuTiny-EVB-122-LQFP64 v2.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 功能介紹

1.1 簡介

該範例示範2組介面的HID複合裝置實作。每個介面擁有一項功能，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2組介面，每個介面使用HID(Human Interface Devices)標準實作一個功能。一個USB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格 1。

Function	USB Composite Device	NUC122 Hardware setting
Control Transfer	Control Pipe endpoint 0: Control IN/OUT	endpoint 0: Control IN endpoint 1: Control OUT
HID Mouse	Interface 0 endpoint 1: Interrupt IN	endpoint 2: Interrupt IN
HID Keyboard	Interface 1 endpoint 2: Interrupt IN	endpoint 3: Interrupt IN

表格 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，使用表格 2的資料格式回傳資料。

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

表格 2 HID 滑鼠報告格式

介面1是一個HID鍵盤裝置 (多媒體鍵) · 使用表格 3的資料格式回傳資料 · 鍵盤代碼定義於 "HID USAGE TABLE V1.12" 第53頁 (可從[USB-IF](#)下載)。

Byte	Bits	Description	
0	0~7	Report ID :0x01	Report ID :0x02
1	0	Modifier Keys	Mute
1	1	Modifier Keys	Volume+
1	2	Modifier Keys	Volume-
1	3	Modifier Keys	Brightness+
1	4	Modifier Keys	Brightness-
1	5~7	Modifier Keys	Reserved
2	0~7	Reserved	Reserved
3	0~7	Key code 1	Reserved
4	0~7	Key code 2	Reserved
5	0~7	Key code 3	Reserved
6	0~7	Key code 4	Reserved
7	0~7	Key code 5	Reserved
8	0~7	Key code 6	Reserved

表格 3 HID 鍵盤報告格式

Set_Report請求是特定類別請求之一，其允許主機傳送報告給裝置，可能是設定輸入、輸出或功能控制的狀態。這些傳輸會通過控制管線來完成，因此，它們必需遵照USB規範中定義的控制管線請求格式。Set_Report請求格式列出如表格 4所示。

組成	大小 (位元組)	描述
bmRequestType	8	00100001
bRequest	8	SET_REPORT (0x09)
wValue	16	報告類型與報告標識
wIndex	16	介面
wLength	16	報告長度

表格 4 Set_Reort 請求格式表

其中，wValue欄位的高位元組指定為報告類型，而報告類型被指定為如表格 5所示。因此，在本範例中，當報告類型為0x2時，使用者可以讀到從控制輸出端點傳送來的輸出報告。

值	報告類型
0x1	輸入
0x2	輸出
0x3	功能
0x4-0xFF	保留

表格 5 報告類型

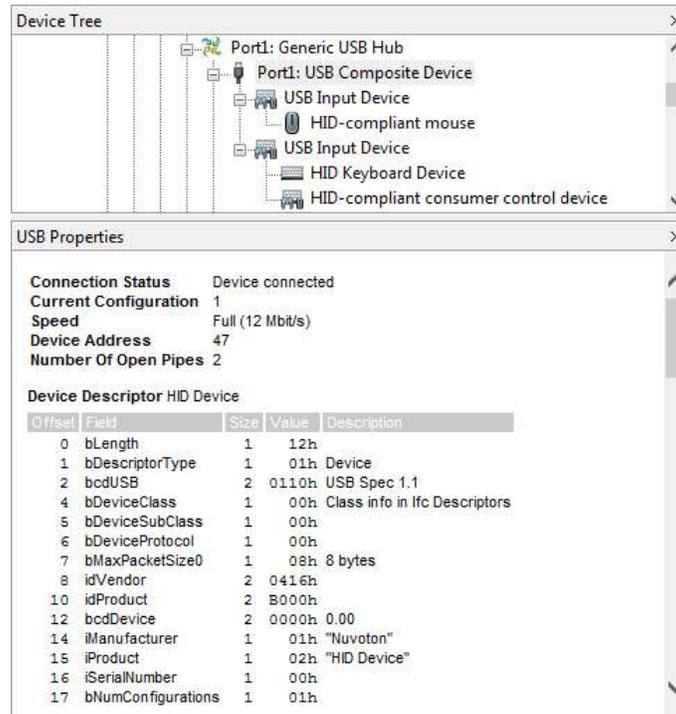
根據範例的HID報告描述符，電腦會送出鍵盤燈號狀態的輸出報告。輸出報告格式如下：

Byte	Bits	Description
0	0~7	Report ID :0x01
1	0	數字鎖定鍵
1	1	大寫鎖定
1	2	捲動鎖定鍵
1	3	死鍵
1	4	Kana
1	5~7	Reserved

表格 6 輸出報告格式

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](#)我們可以獲得圖表 1 的裝置清單。可以看到此複合裝置有2組介面。



圖表 1 裝置清單

1.3.1 HID 滑鼠

若GPIO PD1準位為低準元時(按鈕按下)，可以看見X軸的欄位(見表格 2與圖表 2) 的傳輸數值變成1。電腦滑鼠的指標會往右移動。

Transfer	F	Interrupt	ADDR	ENDP	Usage Page	Button 1	Button 2	Button 3	Button 1	Usage Page	X	Y	Bytes Transferred
215	S	IN	31	1	Button (0x09)	0	0	0	Button 0	Generic Desktop Controls (0x01)	0x00	0x00	3
Transaction													
309	S	IN	31	1	0 00 00 00	ACK	16.000 ms	Time Stamp					
217	S	IN	31	1	Button (0x09)	0	0	0	Button 0	Generic Desktop Controls (0x01)	0x00	0x00	3
Transaction													
311	S	IN	31	1	1 00 00 00	ACK	16.000 ms	Time Stamp					
219	S	IN	31	1	Button (0x09)	0	0	0	Button 0	Generic Desktop Controls (0x01)	0x01	0x00	3
Transaction													
313	S	IN	31	1	0 00 01 00	ACK	16.000 ms	Time Stamp					
221	S	IN	31	1	Button (0x09)	0	0	0	Button 0	Generic Desktop Controls (0x01)	0x01	0x00	3
Transaction													
315	S	IN	31	1	1 00 01 00	ACK	16.000 ms	Time Stamp					

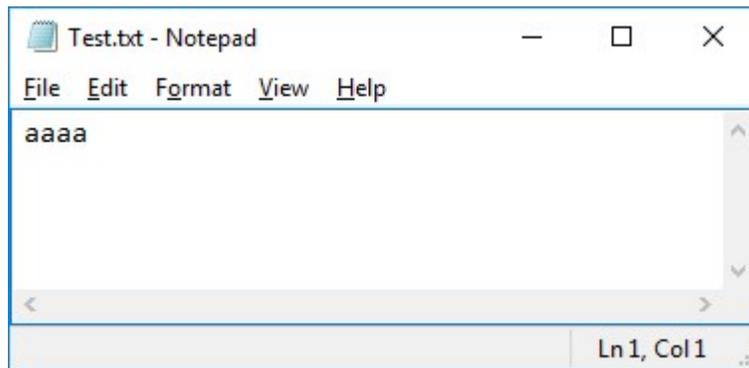
圖表 2 滑鼠的傳輸封包

1.3.2 HID 鍵盤

若GPIO PD2準位為低準位時(按鈕按下)，鍵盤碼的第一個欄位的傳輸數值由0變成0x04。此時這裝置可在記事本或電腦可輸入的地方輸入的數位 'A'，如表格 3。0x04這個數值是在“HID USAGE TABLE V1.12”中定義的字元 'A'。

Transfer	F	Interrupt	ADDR	ENDP	Report ID	Usage Page	Left Ctrl	Left Shift	Left Alt	Left GUI	Right Ctrl	Right Shift	Right Alt	Right GUI	Key 1	Usage Page	Key	Key	Key	Key	Key	Key	Feature	Bytes Transferred
60	S	IN	32	2	0x01	Key (0x07)	0	0	0	0	0	0	0	0		Key (0x07)							9	
Transaction																								
154	S	IN	0x96	32	2	01 00 00 00 00 00 00 00 00 00	ACK	0x4B	Time	16.000 ms	Time Stamp	1.123 910 050												
62	S	IN	32	2	0x01	Key (0x07)	0	0	0	0	0	0	0			A							9	
Transaction																								
156	S	IN	0x96	32	2	1 01 00 00 04 00 00 00 00 00	ACK	0x4B	Time	16.000 ms	Time Stamp	1.139 909 982												
64	S	IN	32	2	0x01	Key (0x07)	0	0	0	0	0	0	0										9	
Transaction																								
158	S	IN	0x96	32	2	0 01 00 00 00 00 00 00 00 00	ACK	0x4B	Time	16.000 ms	Time Stamp	1.155 909 950												

圖表 3 鍵盤的傳輸封包



圖表 4 輸入文字到記事本

1.3.3 多媒體鍵

圖表 5中，當GPIO PD3或PD4拉至低準元(按鈕按下)，可觀察到電腦的音量被調大或調小。

Transfer	F	Interrupt	ADDR	ENDP	Report ID	Usage Page	Mute	Volume Increment	Volume Decrement	Unassigned	Unassigned	Usage Page	System Sleep	Feature	Bytes Transferred
638	S	IN	33	2	0x02	Consumer (0x0C)	0	1	0	0	0	Generic Desktop Controls (0x01)	0	0 0	9
Transaction															
632	S	IN	0x96	33	2	1 02 02 00 00 00 00 00 00 00	ACK	0x4B	Time	16.000 ms	Time Stamp	4.488 919 332			
540	S	IN	33	2	0x02	Consumer (0x0C)	0	0	1	0	0	Generic Desktop Controls (0x01)	0	0 0	9
Transaction															
634	S	IN	0x96	33	2	0 02 04 00 00 00 00 00 00 00	ACK	0x4B	Time	16.000 ms	Time Stamp	4.502 919 250			

圖表 5 多媒體的傳輸封包

1.3.4 鍵盤燈號狀態控制

當鍵盤大寫鎖定鍵被按下時，可觀察到GPIO PD5所連接的LED有變化。傳輸封包如圖表 6。

Transfer		F	Control	ADDR	ENDP	bRequest	
21	S		SET	23	0	SET_REPORT	

Transaction	F	SETUP	ADDR	ENDP	T	D	Tr	R	bRequest	wValue	wIndex	wLength	ACK	Time
113	S	0xB4	23	0	0	H->D	C	I	0x09	0x0201	0x0001	2	0x4B	31.266 us
Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time						
114	S	0x87	23	0	1	01 03	0x4B	35.684 us						
Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time						
115	S	0x96	23	0	1		0x4B	1.222 sec						

Transfer		F	Control	ADDR	ENDP	bRequest	
174	S		SET	23	0	SET_REPORT	

Transaction	F	SETUP	ADDR	ENDP	T	D	Tr	R	bRequest	wValue	wIndex	wLength	ACK	Time
268	S	0xB4	23	0	0	H->D	C	I	0x09	0x0201	0x0001	2	0x4B	25.250 us
Transaction	F	OUT	ADDR	ENDP	T	Data	ACK	Time						
269	S	0x87	23	0	1	01 01	0x4B	25.684 us						
Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time						
270	S	0x96	23	0	1		0x4B	264.137 ms						

圖表 6 多媒體的傳輸封包

2 代碼介紹

此範例程式使用 HID 標準實現了兩個功能介面，滑鼠、鍵盤和多媒體鍵的功能。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，如圖表 7。2.1 節將介紹程式處理的流程。

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time
1113	S	0x96	25	1	0	3 bytes	0x4B	10.550 us
Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time
1114	S	0x96	25	2	0	9 bytes	0x4B	15.989 ms
Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time
1115	S	0x96	25	1	1	3 bytes	0x4B	10.550 us

圖表 7 2 個 IN 中斷端點的 transaction

2.1 代碼介紹

2.1.1 main.c

While 迴圈內檢查鍵盤燈號狀態並有 2 個副程式用於資料的傳輸。每個子程式皆可獨立運行，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_UpdateKbData();

    HID_UpdateMsData();

    /* Enter power down when USB suspend */
    if(g_u8Suspend)
        PowerDown();

    if(memcmp(SetReport, PreSetReport, 2))
    {
        if(SetReport[1] & 0x2)
            PD5 = 1; /* Turn On CapsLock LED */
        else
            PD5 = 0; /* Turn Off CapsLock LED */
        memcpy(PreSetReport, SetReport, 2);
    }
}

```

2.1.2 hid.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 的中斷處理常式會把 `g_u8EP2Ready` 設為真。當 `HID_UpdateMsData()` 檢查到 `g_u8EP2Ready` 為真時，就會設定並回傳數據。當 GPIO PD1 準位為 0 時，設定 `buf[2]` (表格 1 的 X 軸欄位) 為 `0x1`。鍵盤和搖桿的 `HID_UpdateKbData()` 副程式的處理流程，與 `HID_UpdateMsData()` 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 9 個位元組的長度。

```
void HID_ClassRequest(void)
{
    uint8_t buf[8];

    USBD_GetSetupPacket(buf);

    if (buf[0] & 0x80) /* request data transfer direction */
    {
        // Device to host
        switch (buf[1])
        {
            ...
        }
    }
    else
    {
        // Host to device
        switch (buf[1])
        {
            case SET_REPORT:
            {
                if (buf[3] == 2)
                {
                    /* Request Type = Output */
                    USBD_SET_DATA1(EP1);
                    USBD_SET_PAYLOAD_LEN(EP1, buf[6]);
                    USBD_PrepareCtrlOut((uint8_t *)&SetReport, buf[6]);
                    /* Status stage */
                    USBD_PrepareCtrlIn(0, 0);
                }
                break;
            }
        }
    }
}
```

```
...

    case SET_PROTOCOL:
    {
        USBD_SET_DATA1(EP0);
        USBD_SET_PAYLOAD_LEN(EP0, 0);
        break;
    }
    default:
    {
        // Stall
        /* Setup error, stall the device */
        USBD_SetStall(EP0);
        USBD_SetStall(EP1);
        break;
    }
}
}
```

2.1.3 hid.c

HID_MouseReportDescriptor是滑鼠的HID報告描述符元。此描述元定義了表格 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，從圖表 2可以看到對應的欄位，傳輸資料改變。

```
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) */
    0x95, 0x03, /* Report Count(0x3) */
    0x75, 0x01, /* Report Size(0x1) */
    0x81, 0x02, /* Input(3 button bit) */
    0x95, 0x01, /* Report Count(0x1) */
```

```

0x75, 0x05, /* Report Size(0x5) */
0x81, 0x01, /* Input(5 bit padding) */
0x05, 0x01, /* Usage Page(Generic Desktop Controls) */
0x09, 0x30, /* Usage(X) */
0x09, 0x31, /* Usage(Y) */
0x15, 0x81, /* Logical Minimum(0x81)(-127) */
0x25, 0x7F, /* Logical Maximum(0x7F)(127) */
0x75, 0x08, /* Report Size(0x8) */
0x95, 0x02, /* Report Count(0x2) */
0x81, 0x06, /* Input(1 byte) */
0xC0,      /* End Collection */
0xC0,      /* End Collection */
};

```

2.2 組態描述元

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2組介面的描述。“bNumInterfaces”欄位設定為2，用來讓USB主機知道此複合裝置有2組介面。2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 */
    LEN_CONFIG_AND_SUBORDINATE & 0x00FF, /* wTotalLength */
    ((LEN_CONFIG_AND_SUBORDINATE & 0xFF00) >> 8),
    0x02, /* bNumInterfaces */
    0x01, /* bConfigurationValue */
    0x00, /* iConfiguration */
    0x80 | (USBD_SELF_POWERED << 6) | (USBD_REMOTE_WAKEUP << 5), /* bmAttributes */
    USBD_MAX_POWER, /* MaxPower */

    /* HID Interface Descriptor*/
    LEN_INTERFACE, /* bLength */
    DESC_INTERFACE, /* bDescriptorType */
    0x00, /* bInterfaceNumber */
    0x00, /* bAlternateSetting */
    0x01, /* bNumEndpoints */

```

```

0x03, /* bInterfaceClass */
0x01, /* bInterfaceSubClass */
HID_MOUSE, /* bInterfaceProtocol */
0x00, /* iInterface */

/* HID Descriptor */
LEN_HID, /* bLength */
DESC_HID, /* bDescriptorType */
0x10, 0x01, /* bcdHID */
0x00, /* bCountryCode */
0x01, /* bNumDescriptors*/
DESC_HID_RPT, /* bDescriptorType. */
/* Total length of report descriptor. */
HID_MOUSE_REPORT_DESCRIPTOR_SIZE& 0x00FF,
(HID_MOUSE_REPORT_DESCRIPTOR_SIZE& 0xFF00) >> 8),

/* EP Descriptor: interrupt in. */
LEN_ENDPOINT, /* bLength */
DESC_ENDPOINT, /* bDescriptorType */
(INT_IN_EP_NUM_MS | EP_INPUT),/* bEndpointAddress */
EP_INT, /* bmAttributes */
EP2_MAX_PKT_SIZE & 0x00FF,/* wMaxPacketSize */
((EP2_MAX_PKT_SIZE & 0xFF00) >> 8),
HID_DEFAULT_INT_IN_INTERVAL, /* bInterval */

/* HID Interface Descriptor */
LEN_INTERFACE, /* bLength */
DESC_INTERFACE, /* bDescriptorType */
0x01, /* bInterfaceNumber */
0x00, /* bAlternateSetting */
0x01, /* bNumEndpoints */
0x03, /* bInterfaceClass */
0x01, /* bInterfaceSubClass */
HID_KEYBOARD, /* bInterfaceProtocol */
0x00, /* iInterface */

/* HID Descriptor */
LEN_HID, /* bLength */
DESC_HID, /* bDescriptorType */
0x10, 0x01, /* bcdHID */
0x00, /* bCountryCode*/

```

```
    0x01, /* bNumDescriptors*/
    DESC_HID_RPT, /* bDescriptorType */
/* Total length of report descriptor. */
    HID_KEYBOARD_REPORT_DESCRIPTOR_SIZE& 0x00FF,
    ((HID_KEYBOARD_REPORT_DESCRIPTOR_SIZE& 0xFF00) >> 8),

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

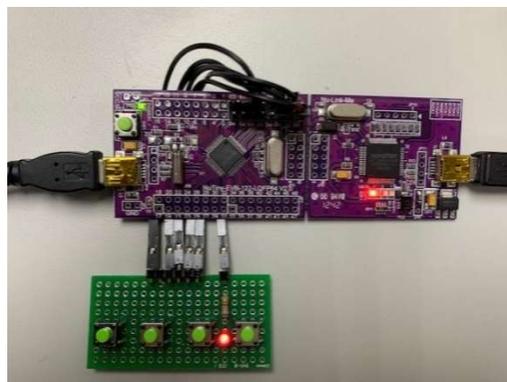
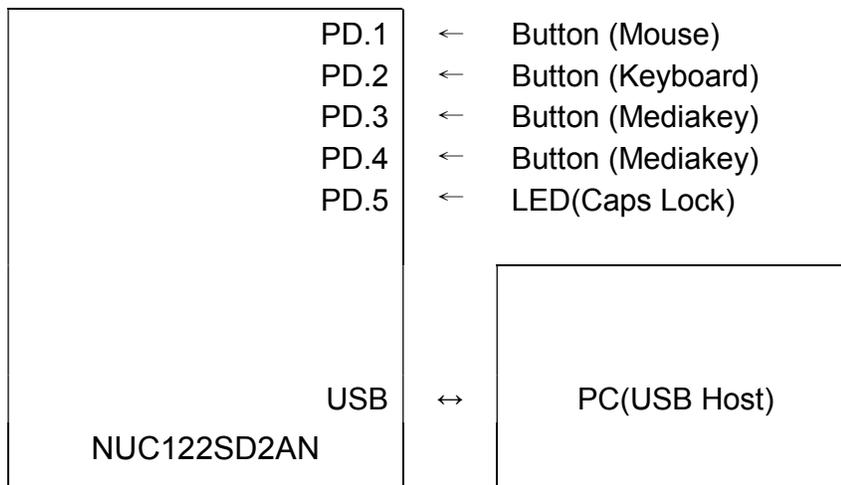
3 軟體與硬體環境

- 軟體環境

- BSP 版本
 - ◆ NUC122 Series BSP CMSIS V3.000.003
- IDE 版本
 - ◆ Keil uVersion5.26

- 硬體環境

- 電路元件
 - ◆ NuTiny-EVB-122-LQFP64 V2.0
 - ◆ USB mini USB cable
- 示意圖



4 目錄資訊

📁 EC_NUC122_USBD_HID KB_MS_MMKey_LEDCtrl_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
📁 SampleCode	
📁 ExampleCode	Source file of example code

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 USB_D_HID_KB_MS_MMKey_LED Ctrl.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
Jul.01, 2019	1.00	1. 初始發布.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*