HID 觸控裝置

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	HID 觸控感應裝置
BSP 版本	NUC123 Series BSP CMSIS V3.01.001
開發平台	NuTiny-EVB-NUC123-LQFP64 v1.0

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

本範例實作一個HID(Human Interface Device)觸控裝置。模擬兩個輸入軌跡。

1.2 原理

本裝置內含控制管線(端點0)，用於控制傳輸。此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；HID觸控資料傳輸則是由唯一的介面完成，此介面有個中斷的輸入端點，此端點用於輸入方向的中斷傳輸(Interrupt Transfer)。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 Touch

表 1: HID 觸控裝置的介面和端點組態

介面0的觸控資料，使用表 2的格式回傳。此格式即是HID的報告描述元所定義的資料格式，定義了兩組相同的資料結構，讓此裝置可同時回傳兩個不同的軌跡。關於觸控裝置報告描述元更詳細的欄位解說，可參考微軟公司網站。

Byte	Bits	Description
0	0	Tip Switch
0	1	In Range
0	2	Touch Valid
0	3 ~ 7	Padding
1	0 ~ 7	Contact Identifier
2 ~ 3	0 ~ 7	X-axis
4 ~ 5	0 ~ 7	Y-axis
6	0	Tip Switch
6	1	In Range
6	2	Touch Valid
6	3 ~ 7	Padding
7	0 ~ 7	Contact Identifier
8 ~ 9	0 ~ 7	X-axis
10 ~ 11	0 ~ 7	Y-axis

表 2: HID 觸控裝置的 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windows的裝置管理員，可以獲得圖 1的裝置列表，HID-compliant touch screen就是此次展示的裝置。

圖 1:裝置管理員的裝置列表

圖 2是使用USB封包分析儀紀錄的傳輸封包，可看見每次交易(transaction)，有14 位元組的資料以輸入的方向傳送。

Transaction	F	IN	ADDR	ENDP	T	Data	ACK	Time	Time Stamp
196	S	0x96	60	1	1	01 00 00 00 00 00 00 00 01 00 00 00 00 00	0x4B	1.000 ms	1 . 722 543 132
197	S	0x96	60	1	0	01 00 00 00 00 00 00 00 01 00 00 00 00 00	0x4B	999.982 us	1 . 723 543 150
198	S	0x96	60	1	1	01 00 00 00 00 00 00 00 01 00 00 00 00 00	0x4B	1.000 ms	1 . 724 543 132
199	S	0x96	60	1	0	01 00 00 00 00 00 00 00 01 00 00 00 00 00	0x4B	1.000 ms	1 . 725 543 132
200	S	0x96	60	1	1	01 00 00 00 00 00 00 00 01 00 00 00 00 00	0x4B	999.982 us	1 . 726 543 150
201	S	0x96	60	1	0	01 07 00 02 02 00 01 07 01 02 02 20 01 02	0x4B	1.000 ms	1 . 727 543 132
202	S	0x96	60	1	1	01 07 00 02 02 00 01 07 01 02 02 20 01 02	0x4B	1.000 ms	1 . 728 543 132
203	S	0x96	60	1	0	01 07 00 02 02 00 01 07 01 02 02 20 01 02	0x4B	1.000 ms	1 . 729 543 132
204	S	0x96	60	1	1	01 07 00 02 02 00 01 07 01 02 02 20 01 02	0x4B	1.000 ms	1 . 730 543 132
205	S	0x96	60	1	0	01 07 00 02 02 00 01 07 01 02 02 20 01 02	0x4B	1.000 ms	1 . 731 543 132

圖 2:HID 觸控裝置的傳輸資料

在裝置連加上電腦後，打開”小畫家”軟體，此裝置會畫出兩條平行的軌跡，如圖 3。

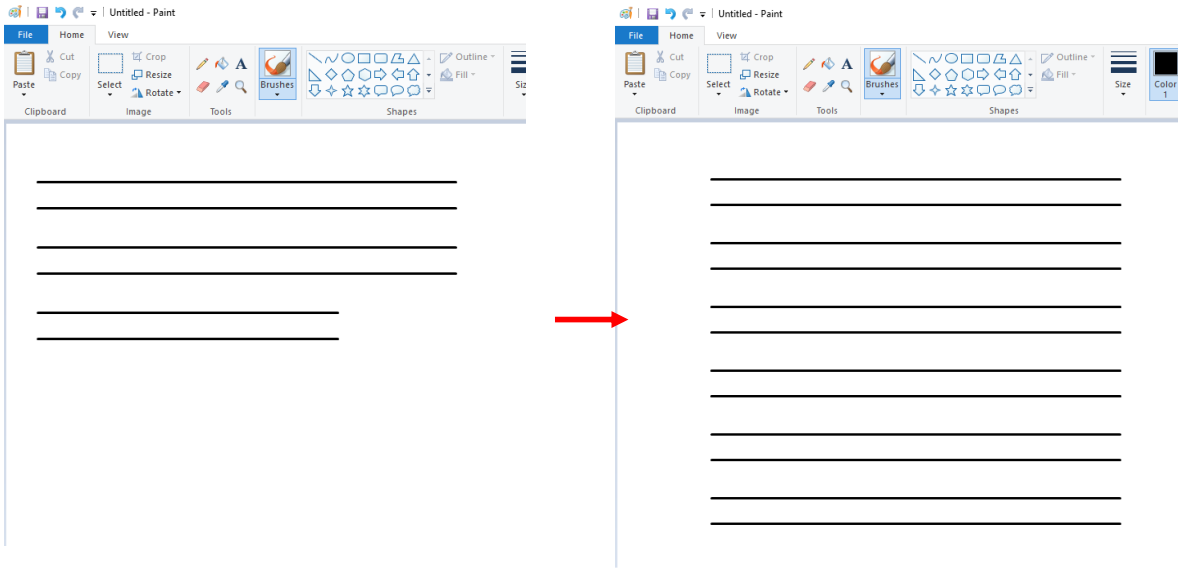


圖 3: HID 觸碰裝置在小畫家畫平行線

2 代碼介紹

此範例程式使用 HID 標準實現觸控裝置的模擬。當主機發出 IN TOKEN 時，IN 的中斷傳輸端點使用定義於 HID 報告描述元的資料格式回傳資料，如圖 2。2.1~2.2 節將介紹程式處理的流程。

2.1 Main.c

While回圈內的HID_UpdateTouchData()用於資料的傳輸。裡面實作的IN token中斷產生時的處理邏輯。

```
    while(1)
    {
       HID_UpdateTouchData();
    }
```

2.2 hid_touch.c

USB主機會發出IN Token向端點2請求資料。當IN Token收到時，端點2的USBD中斷處理程序會把g_u8EP2Ready設為真。當HID_UpdateTouchData ()檢查到g_u8EP2Ready為真時，就會設定封包回傳資料。 由於有兩條軌跡同時上傳，每條軌跡需要6位元組的空間，每組資料前會再加上一個位元組，存放Report ID。所以一次回傳的資料長度總共14位元組。

```
/* two lines demo in Paint */
uint32_t u32ReportCount = 0;
uint8_t u8IsX1Send04 = 0, u8IsX2Send04 = 0;
void HID_UpdateTouchData(void)
{
    uint8_t *pu8buf;
    static uint16_t u16X1 = 0x01f0, u16Y1 = 0x0100;

    if(g_u8EP2Ready)
    {

        pu8buf = (uint8_t *)(USBD_BUF_BASE + USBD_GET_EP_BUF_ADDR(EP2));

        /* Report ID 1 */
        pu8buf[0] = 1;
        pu8buf[2] = 0;
        pu8buf[8] = 1;
```

以下是畫線的實作，把欲執行的觸控資料填入對應的欄位，使用者可以根據自己的需求更改程式，來模擬不同的軌跡效果。

```
        u32ReportCount++;

        if ((u16X1 >= 0x0200) && (u16X1 <= 0x0400)) /* touchDown */
```

```
        {
            u8IsX1Send04 = 1;

            pu8buf[1] = 0x07;
            pu8buf[3] = u16X1 & 0xff;
            pu8buf[4] = (u16X1 >> 8) & 0xff;
            pu8buf[5] = u16Y1 & 0xff;
            pu8buf[6] = (u16Y1 >> 8) & 0xff;
            pu8buf[7] = 0x07;
            pu8buf[9] = u16X1 & 0xff;
            pu8buf[10] = (u16X1 >> 8) & 0xff;
            pu8buf[11] = (u16Y1+0x20) & 0xff;
            pu8buf[12] = ((u16Y1+0x20) >> 8) & 0xff;
            pu8buf[13] = 2;
        }
        else if (u8IsX1Send04)     /* touchUp */
        {
            u8IsX1Send04 = 0;
            pu8buf[1] = 0x04;
            pu8buf[3] = 0x00;
            pu8buf[4] = 0x04;
            pu8buf[5] = u16Y1 & 0xff;
            pu8buf[6] = (u16Y1 >> 8) & 0xff;
            pu8buf[7] = 0x04;
            pu8buf[9] = 0x00;
            pu8buf[10] = 0x04;
            pu8buf[11] = (u16Y1+0x20) & 0xff;
            pu8buf[12] = ((u16Y1+0x20) >> 8) & 0xff;
            pu8buf[13] = 2;

            if (u16Y1 == 0x0100)
            {
                pu8buf[5] = 0xE0;
                pu8buf[6] = 0x02;
                pu8buf[11] = 0x00;
                pu8buf[12] = 0x03;
            }
            else
            {
                pu8buf[5] = (u16Y1-0x50) & 0xff;
                pu8buf[6] = ((u16Y1-0x50) >> 8) & 0xff;
                pu8buf[11] = ((u16Y1+0x20)-0x50) & 0xff;
                pu8buf[12] = (((u16Y1+0x20)-0x50) >> 8) & 0xff;
            }
        }
        else
        {
            pu8buf[1] = 0;
            pu8buf[3] = 0;
            pu8buf[4] = 0;
            pu8buf[5] = 0;
            pu8buf[6] = 0;
            pu8buf[7] = 0;
            pu8buf[9] = 0;
            pu8buf[10] = 0;
            pu8buf[11] = 0;
            pu8buf[12] = 0;
            pu8buf[13] = 0;
        }

        if ((u32ReportCount % 6) == 0)
            u16X1 += 0x3;

        if (u16X1 > 0x0400)
        {
```

```
            u16X1 = 0x01f0;
            u16Y1 += 0x50;
            if (u16Y1 > 0x0300)
                u16Y1 = 0x0100;
        }
```

最後設定端點2的資料長度來回傳資料。

```
        g_u8EP2Ready = 0;
        USBD_SET_PAYLOAD_LEN(EP2, 14);
    }
}
```

2.3 Descriptor.c

2.3.1 HID Report Descriptor

HID_DigitizerReportDescriptor是此觸控裝置的HID報告描述元。此描述元定義了表 2中的資料格式。

欄位的相關資料可以參考微軟公司官方網站和USB開發者論壇(USB-IF)的HID Usage Table。

```
/*!<USB HID Report Descriptor */
const uint8_t HID_DigitizerReportDescriptor[] =
{
    0x05, 0x0D,         // USAGE_PAGE (Digitizers)
    0x09, 0x04,         // USAGE (Touch Screen)
    0xA1, 0x01,         // COLLECTION (Application)
    0x85, 0x01,         // REPORT_ID (1)
    0x09, 0x22,         // USAGE (Finger)

    0xA1, 0x02,         // COLLECTION (Logical)
    0x09, 0x42,         // USAGE (Tip Switch)
    0x15, 0x00,         // LOGICAL_MINIMUM (0)
    0x25, 0x01,         // LOGICAL_MAXIMUM (1)
    0x75, 0x01,         // REPORT_SIZE (1)
    0x95, 0x01,         // REPORT_COUNT (1)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x09, 0x32,         // USAGE (In Range)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x09, 0x47,         // USAGE (Touch Valid)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x95, 0x05,         // REPORT_COUNT (5)
    0x81, 0x03,         // INPUT (Cnst,Ary,Abs)
    0x75, 0x08,         // REPORT_SIZE (8)
    0x09, 0x51,         // USAGE (Contact Identifier)
    0x95, 0x01,         // REPORT_COUNT (1)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x05, 0x01,         // USAGE_PAGE (Generic Desktop)
    0x75, 0x10,         // REPORT_SIZE (16)
    0x55, 0x0E,         // UNIT_EXPONENT (-2)
    0x65, 0x11,         // UNIT (Centimeter)
    0x09, 0x30,         // USAGE (X)
    0x35, 0x00,         // PHYSICAL_MINIMUM (0)
    0x46, 0xE3, 0x13,   // PHYSICAL_MAXIMUM (0x13E3)
    0x26, 0x7F, 0x07,   // LOGICAL_MAXIMUM (0x77F)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x09, 0x31,         // USAGE (Y)
    0x46, 0x2F, 0x0B,   // PHYSICAL_MAXIMUM (0xB2F)
    0x26, 0x37, 0x04,   // LOGICAL_MAXIMUM (0x437)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
```

```
    0xC0,               // END_COLLECTION

    0xA1, 0x02,         // COLLECTION (Logical)
    0x05, 0x0D,         // USAGE_PAGE (Digitizers)
    0x09, 0x42,         // USAGE (Tip Switch)
    0x15, 0x00,         // LOGICAL_MINIMUM (0)
    0x25, 0x01,         // LOGICAL_MAXIMUM (1)
    0x75, 0x01,         // REPORT_SIZE (1)
    0x95, 0x01,         // REPORT_COUNT (1)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x09, 0x32,         // USAGE (In Range)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x09, 0x47,         // USAGE (Touch Valid)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x95, 0x05,         // REPORT_COUNT (5)
    0x81, 0x03,         // INPUT (Cnst,Ary,Abs)
    0x75, 0x08,         // REPORT_SIZE (8)
    0x09, 0x51,         // USAGE (Contact Identifier)
    0x95, 0x01,         // REPORT_COUNT (1)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x05, 0x01,         // USAGE_PAGE (Generic Desktop)
    0x75, 0x10,         // REPORT_SIZE (16)
    0x55, 0x0E,         // UNIT_EXPONENT (-2)
    0x65, 0x11,         // UNIT (Centimeter)
    0x09, 0x30,         // USAGE (X)
    0x35, 0x00,         // PHYSICAL_MINIMUM (0)
    0x46, 0xE3, 0x13,   // PHYSICAL_MAXIMUM (0x13E3)
    0x26, 0x7F, 0x07,   // LOGICAL_MAXIMUM (0x77F)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x46, 0x2F, 0x0B,   // PHYSICAL_MAXIMUM (0xB2F)
    0x26, 0x37, 0x04,   // LOGICAL_MAXIMUM (0x437)
    0x09, 0x31,         // USAGE (Y)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0xC0,               // END_COLLECTION

    0x05, 0x0D,         // USAGE_PAGE (Digitizers)
    0x09, 0x54,         // USAGE (Contact Count)
    0x15, 0x00,         // LOGICAL_MINIMUM (0)
    0x26, 0xFF, 0x00,       // LOGICAL_MAXIMUM (0x00FF)
    0x95, 0x01,         // REPORT_COUNT (1)
    0x75, 0x08,         // REPORT_SIZE (8)
    0x81, 0x02,         // INPUT (Data,Var,Abs)
    0x09, 0x55,         // USAGE (Contact Count Maximum)
    0x25, 0x02,         // LOGICAL_MAXIMUM (2)
    0x95, 0x01,         // REPORT_COUNT (1)
    0x85, 0x02,         // REPORT_ID (2)
    0xB1, 0x02,         // FEATURE (Data,Var,Abs)
    0xC0                // END_COLLECTION
};
```

2.3.2 Configuration Descriptor

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1組介面的描述。“bNumInterfaces”欄位設定為1，讓USB主機知道此裝置有1組介面。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。

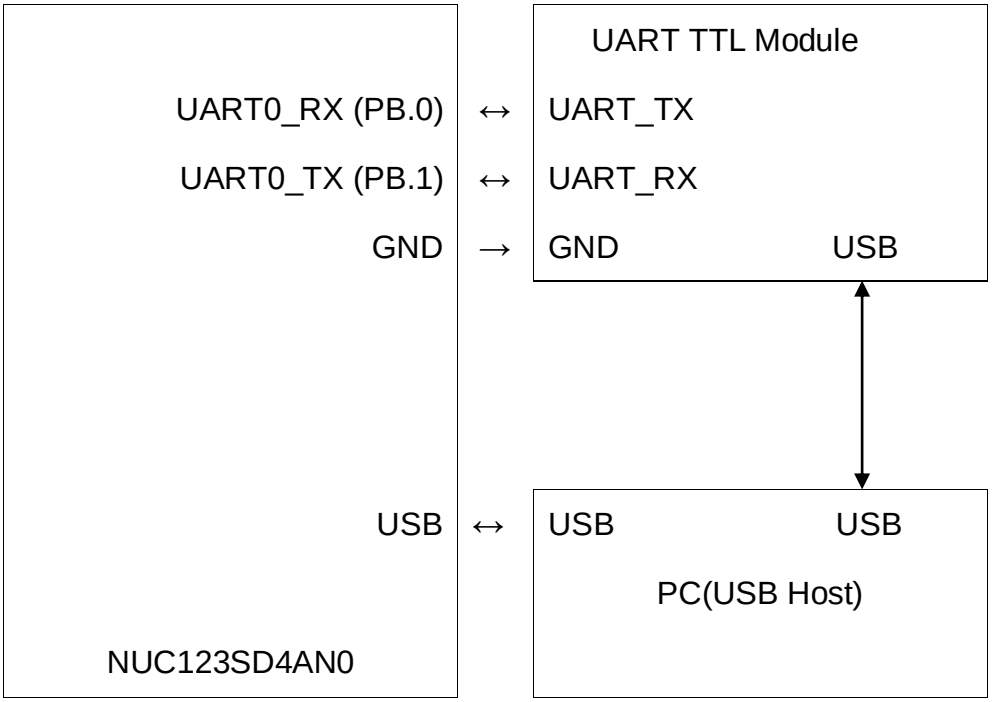
```
/*!<USB Configure Descriptor */
const uint8_t gu8ConfigDescriptor[] =
{
    LEN_CONFIG,     /* bLength */
```

```
    DESC_CONFIG,    /* bDescriptorType */
    /* wTotalLength */
    LEN_CONFIG_AND_SUBORDINATE & 0x00FF,
    ((LEN_CONFIG_AND_SUBORDINATE & 0xFF00) >> 8),
    0x01,           /* 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_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,   /* Descriptor type. */
    /* Total length of report descriptor. */
    sizeof(HID_DigitizerReportDescriptor) & 0x00FF,
    ((sizeof(HID_DigitizerReportDescriptor) & 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 */
};
```

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)
    - USB packet analyzer(Optional)

■ 示意圖


UART TTL Module
UART0_RX (PB.0)
↔
UART_TX
UART0_TX (PB.1)
↔
UART_RX
GND
→
GND
USB
USB
↔
USB
USB
PC(USB Host)
NUC123SD4AN0

4 目錄資訊

- EC_NUC123_USBD_HID_Touch_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

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_Touch.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
Nov. 01, 2019	1.00	1. 初始發佈.

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.