M463 使用 TinyUSB 實現滑鼠與鍵盤

NuMicro® 32 位元系列微控制器範例代碼介紹

文件資訊

應用簡述	本範例代碼使用 M463 MCU 基於 TinyUSB 架構實現滑鼠與鍵盤的功能
BSP 版本	M460_Series_BSP_CMSIS_V3.00.004
開發平台	NuMaker-M463KG 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 and microprocessor 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. 概述

本範例代碼使用 Nuvoton M463 系列微控制器基於 TinyUSB Library 來實現 USB HID（Human Interface Device）滑鼠與鍵盤的功能。透過本範例，使用者可以了解如何初始化 MCU、處理 HID 類別的滑鼠與鍵盤事件。本範例適用於對 USB HID 協定有初步認識，並希望快速建立應用雛型的開發者。本應用採用 M463 的 USB 高速（USB HS）介面與 GPIO 來模擬滑鼠與鍵盤輸入，並透過 TinyUSB 提供的 API 進行 HID 資料傳輸。

1.1 原理

Nuvoton M463 系列微控制器具備 USB High-Speed Device，支援 USB 2.0 裝置功能。此範例透過 TinyUSB 對 USB 裝置初始化、設定描述符，讓開發板在接上電腦後被辨識為 HID 類型的滑鼠與鍵盤。

TinyUSB 是一個開源、跨平台的 USB Host/Device Stack，設計目標是提供簡單易用、可攜性高且資源占用小的 USB 協定支援，適合嵌入式系統開發。TinyUSB 架構模組化，依功能可分為以下幾個主要部分：

- **Device Stack**：支援 USB 裝置端功能，包含多種 USB 類別如 CDC（虛擬串口）、MSC（大容量儲存裝置）、HID（人機介面裝置）、MIDI、DFU 等。
- **Host Stack**：支援 USB 主機端功能，可控制外接裝置（如讀取 USB 隨身碟、滑鼠等）。
- **Class Drivers**：每種 USB 類別對應的驅動模組，用於解析資料與處理控制流程。
- **Board Support Layer**：用於將 TinyUSB 連接至 MCU 的 USB 控制器，實作底層傳輸邏輯。

在本範例中，我們主要使用其 Device Stack 中的 HID 類別驅動，使 M463 能夠在與電腦連接後被辨識為 USB 滑鼠與鍵盤裝置。TinyUSB 提供統一且直觀的 API，例如 tud_hid_keyboard_report() 與 tud_hid_mouse_report()，讓開發者無需關心底層傳輸協議，即可輕鬆實現 HID 資料傳輸。

本範例使用 tud_hid_keyboard_report()模擬鍵盤資料的傳輸：

```
bool tud_hid_keyboard_report(uint8_t instance, uint8_t report_id, uint8_t modifier, uint8_t const keycode[6]);
```

詳細的參數說明如下：

- **instance**：HID 執行個體編號，通常為 0。
- **report_id**：若 HID 描述符中啟用了 Report ID，則填入對應值，否則設為 0。
- **modifier**：修飾鍵，如 LEFT SHIFT、RIGHT CTRL 等，亦可設為 0 表示無修飾鍵。
- **keycode[6]**：一次最多傳送 6 個按鍵代碼，若未使用的位置需填入 0。

另外，本範例也有使用 tud_hid_mouse_report()模擬滑鼠資料的傳輸：

bool tud_hid_mouse_report(uint8_t instance, uint8_t report_id, uint8_t buttons, int8_t x, int8_t y, int8_t vertical, int8_t horizontal);

詳細的參數說明如下：

- **instance**：HID 執行個體編號，通常為 0。
- **report_id**：若 HID 描述符中啟用了 Report ID，則需填入對應值，否則設為 0。
- **buttons**：滑鼠按鈕狀態，最多 8 位元（bit 0~2 分別為左、中、右鍵）。
- **x**：游標在 X 軸的相對移動（-127 ~ 127）。
- **y**：游標在 Y 軸的相對移動（-127 ~ 127）。
- **vertical**：垂直滾輪的移動量（可為正負）。
- **horizontal**：水平滾輪的移動量（可為正負）。

1.2 執行結果

本範例代碼執行編譯和燒錄後，通過 UART 串口可以看到範例代碼的相關訊息：

```
+-----------------------------------------------------+
|          TinyUSB_Mouse_Keyboard Sample Code         |
+-----------------------------------------------------+
If PE.8 = 0, report the keyboard is inputting 'a'.
If PE.9 = 0, report the mouse is moving.
```

圖 1-1 UART 訊息

將 USB HS Port 連接到 PC 後，可以根據訊息按下 BTN0(PE.8)模擬鍵盤按下'a'按鍵：

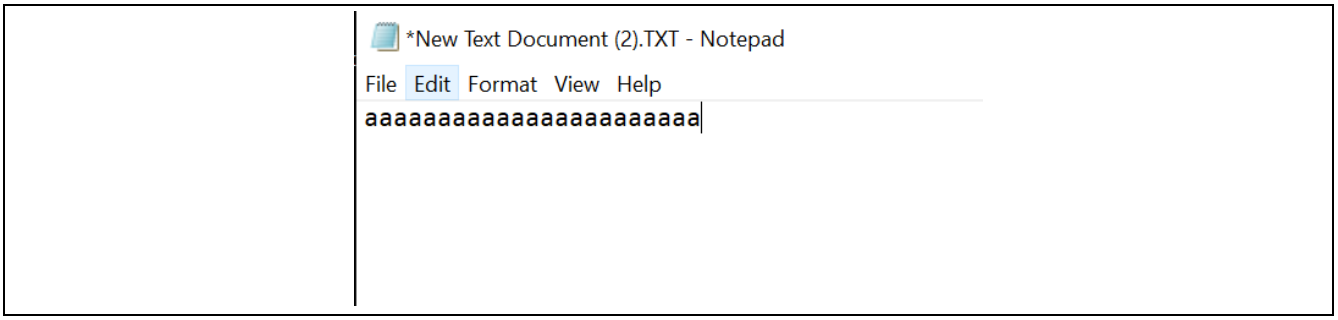


圖 1-2 模擬鍵盤按鍵

按下 BTN1(PE.9)模擬滑鼠向右下角移動：

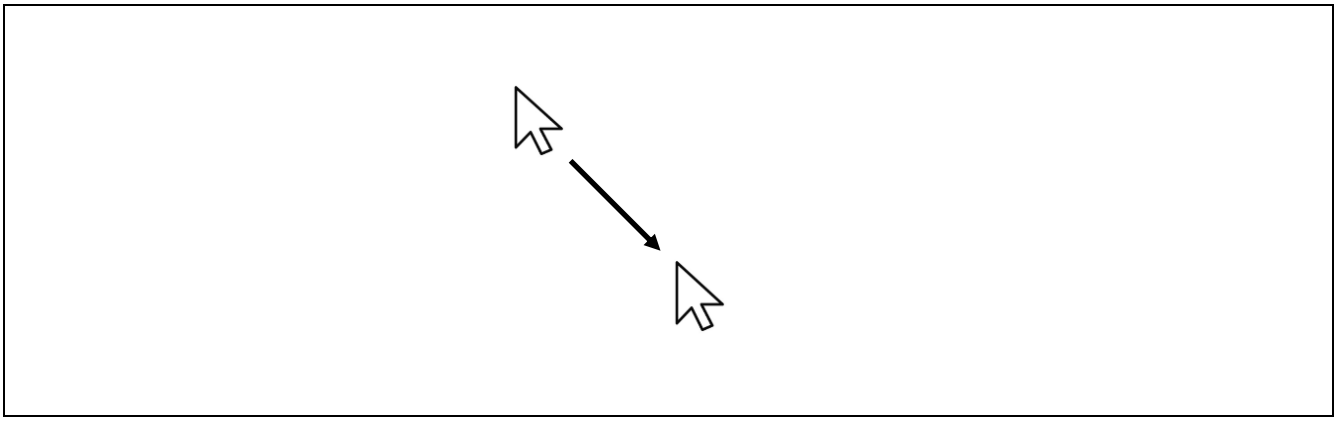


圖 1-3 模擬滑鼠移動

2. 代碼介紹

初始化 MCU 並呼叫 TinyUSB Library 的函式，代碼位於 *main.c*。

```
void SYS_Init(void)
{
    uint32_t volatile i;

    /* Unlock protected registers */
    SYS_UnlockReg();

    /*---------------------------------------------------*/
    /* Init System Clock                                 */
    /*---------------------------------------------------*/

    /* Enable HIRC and HXT clock */
    CLK_EnableXtalRC(CLK_PWRCTL_HIRCEN_Msk | CLK_PWRCTL_HXTEN_Msk);

    /* Wait for HIRC and HXT clock ready */
    CLK_WaitClockReady(CLK_STATUS_HIRCSTB_Msk | CLK_STATUS_HXTSTB_Msk);

    /* Set PCLK0 and PCLK1 to HCLK/2 */
    CLK->PCLKDIV = (CLK_PCLKDIV_APB0DIV_DIV2 | CLK_PCLKDIV_APB1DIV_DIV2);

    /* Set core clock to 200MHz */
    CLK_SetCoreClock(FREQ_200MHZ);

    /* Enable all GPIO clock */
    CLK->AHBCLK0 |= CLK_AHBCLK0_GPACKEN_Msk | CLK_AHBCLK0_GPBCKEN_Msk |
CLK_AHBCLK0_GPCCKEN_Msk | CLK_AHBCLK0_GPDCKEN_Msk |
                    CLK_AHBCLK0_GPECKEN_Msk | CLK_AHBCLK0_GPFCKEN_Msk |
CLK_AHBCLK0_GPGCKEN_Msk | CLK_AHBCLK0_GPHCKEN_Msk;
    CLK->AHBCLK1 |= CLK_AHBCLK1_GPICKEN_Msk | CLK_AHBCLK1_GPJCKEN_Msk;

    /* Enable UART0 module clock */
    CLK_EnableModuleClock(UART0_MODULE);

    /* Select UART0 module clock source as HIRC and UART0 module clock divider as 1 */
    CLK_SetModuleClock(UART0_MODULE, CLK_CLKSEL1_UART0SEL_HIRC, CLK_CLKDIV0_UART0(1));

    /* Select HSUSBD */
    SYS->USBPHY &= ~SYS_USBPHY_HSUSBROLE_Msk;

    /* Enable USB PHY */
    SYS->USBPHY = (SYS->USBPHY & ~(SYS_USBPHY_HSUSBROLE_Msk | SYS_USBPHY_HSUSBACT_Msk)) |
SYS_USBPHY_HSUSBEN_Msk;
    for(i = 0; i < 0x1000; i++);    // delay > 10 us
    SYS->USBPHY |= SYS_USBPHY_HSUSBACT_Msk;

    /* Enable HSUSBD module clock */
    CLK_EnableModuleClock(HSUSBD_MODULE);

    /*---------------------------------------------------*/
    /* Init I/O Multi-function                           */
    /*---------------------------------------------------*/

    /* Set multi-function pins for UART0 RXD and TXD */
    SET_UART0_RXD_PB12();
    SET_UART0_TXD_PB13();

    /* Lock protected registers */
    SYS_LockReg();
}
```

```
/*---------------------------------------------------*/
/*  Main Function                                    */
/*---------------------------------------------------*/
int32_t main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Init System, peripheral clock and multi-function I/O */
    SYS_Init();

    /* Init UART0 to 115200-8n1 for print message */
    UART_Open(UART0, 115200);

    printf("\n");
    printf("+------------------------------------------------------------+\n");
    printf("|          TinyUSB_Mouse_Keyboard Sample Code                |\n");
    printf("+------------------------------------------------------------+\n");
    printf("If PE.8 = 0, report the keyboard is inputting 'a'.\n");
    printf("If PE.9 = 0, report the mouse is moving.\n");

    /* Set PE.8 & PE.9 as Quasi-bidirectional mode */
    GPIO_SetMode(PE, BIT8| BIT9, GPIO_MODE_QUASI);
    PE8 = 1;
    PE9 = 1;

    // 1ms tick timer
    SysTick_Config(200000000 / 1000);

    // init device stack on configured roothub port
    tud_init(BOARD_TUD_RHPORT);

    while(1) {
        tud_task(); // tinyusb device task

        hid_task();
    }
}
```

使用 BTN0(PE.8)及 BTN1(PE.9) 按鍵來觸發滑鼠事件和鍵盤事件，當按下 BTN0(PE.8)呼叫 tud_hid_n_keyboard_report() 傳輸鍵盤事件的數據，當按下 BTN1(PE.9) 呼叫 tud_hid_n_mouse_report()傳輸滑鼠事件的數據，代碼位於 *main.c*。

```
    ……
void hid_task(void)
{
    // Poll every 10ms
    const uint32_t interval_ms = 10;
    static uint32_t start_ms = 0;

    if(board_millis() - start_ms < interval_ms) return;   // not enough time
    start_ms += interval_ms;

    /*------------- Keyboard -------------*/
    if(tud_hid_n_ready(ITF_KEYBOARD)) {
        // use to avoid send multiple consecutive zero report for keyboard
        static bool has_key = false;

        if(PE8==0) {
            uint8_t keycode[6] = { 0 };
```

```
            keycode[0] = HID_KEY_A;

            tud_hid_n_keyboard_report(ITF_KEYBOARD, 0, 0, keycode);

            has_key = true;
        } else {
            // send empty key report if previously has key pressed
            if(has_key) tud_hid_n_keyboard_report(ITF_KEYBOARD, 0, 0, NULL);
            has_key = false;
        }
    }

    /*------------- Mouse -------------*/
    if(tud_hid_n_ready(ITF_MOUSE)) {
        if(PE9==0) {
            int8_t const delta = 5;

            // no button, right + down, no scroll pan
            tud_hid_n_mouse_report(ITF_MOUSE, 0, 0x00, delta, delta, 0, 0);
        }
    }
}
```

Board Support Layer 代碼，用來處理 USB 控制器收發 USB 封包的中斷函式，將封包資料傳遞到 TinyUSB Library 中，位於 *dcd_M463.c*。

```
void dcd_int_handler(uint8_t rhport)
{
    ……
    if(IrqStL & (HSUSBD_GINTSTS_EPAIF_Msk | HSUSBD_GINTSTS_EPBIF_Msk |
HSUSBD_GINTSTS_EPCIF_Msk | HSUSBD_GINTSTS_EPDIF_Msk | HSUSBD_GINTSTS_EPEIF_Msk |
HSUSBD_GINTSTS_EPFIF_Msk | HSUSBD_GINTSTS_EPGIF_Msk | HSUSBD_GINTSTS_EPHIF_Msk |
HSUSBD_GINTSTS_EPIIF_Msk | HSUSBD_GINTSTS_EPJIF_Msk | HSUSBD_GINTSTS_EPKIF_Msk |
HSUSBD_GINTSTS_EPLIF_Msk)) {
        /* service PERIPH_EPA through PERIPH_EPL */
        for(ep_index = PERIPH_EPA, mask = HSUSBD_GINTSTS_EPAIF_Msk, xfer =
&xfer_table[PERIPH_EPA], ep = &HSUSBD->EP[PERIPH_EPA]; ep_index < PERIPH_MAX_EP;
ep_index++, mask <<= 1, xfer++, ep++) {
            if(IrqStL & mask) {
                uint8_t const ep_addr = xfer->ep_addr;
                bool const out_ep = !(ep_addr & TUSB_DIR_IN_MASK);
                uint32_t ep_state = ep->EPINTSTS & ep->EPINTEN;

                if(out_ep) {
                    uint16_t const available_bytes = ep->EPDATCNT &
HSUSBD_EPDATCNT_DATCNT_Msk;
                    /* copy the data from the PC to the previously provided buffer */
                    for(int count = 0; (count < available_bytes) && (xfer-
>out_bytes_so_far < xfer->total_bytes); count++, xfer->out_bytes_so_far++) {
                        *xfer->data_ptr++ = ep->EPDAT_BYTE;
                    }

                    /* when the transfer is finished, alert TinyUSB; otherwise, continue
accepting more data */
                    if((xfer->total_bytes == xfer->out_bytes_so_far) || (available_bytes <
xfer->max_packet_size)) {
```

```
                        dcd_event_xfer_complete(0, ep_addr, xfer->out_bytes_so_far,
XFER_RESULT_SUCCESS, true);
                    }

                } else if(ep_state & HSUSBD_EPINTSTS_BUFEMPTYIF_Msk) {
                    /* send any remaining data */
                    dcd_userEP_in_xfer(xfer, ep);
                } else if(ep_state & HSUSBD_EPINTSTS_TXPKIF_Msk) {
                    /* alert TinyUSB that we've finished */
                    dcd_event_xfer_complete(0, ep_addr, xfer->total_bytes,
XFER_RESULT_SUCCESS, true);
                    ep->EPINTEN = 0;
                }

                ep->EPINTSTS = ep_state;
            }
        }
    }
}
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M460_Series_BSP_CMSIS_V3.00.004
- IDE 版本
  - Keil uVision 5.38

3.2 硬體需求

- 電路元件
  - NuMaker-M463KG V1.0
- 線路示意圖
  - 將 UART0 TX (PB.13) pin 腳連接到 PC UART RX，以顯示範例代碼的訊息。
  - 將 USB 線從 PC 連接到開發板的 USB HS Port，作為 TinyUSB 滑鼠及鍵盤的連接。

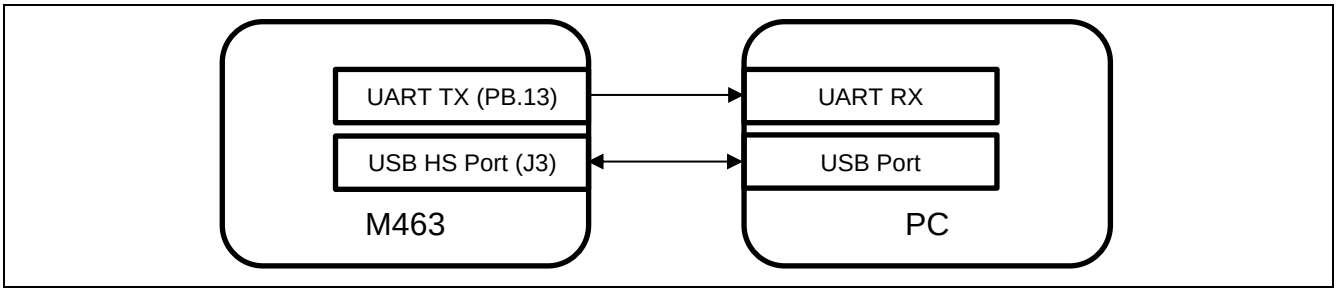


圖 3-1 線路示意圖

4. 目錄資訊

目錄結構如下圖所示。

EC_M463_TinyUSB_Mouse_Keyboard_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	
Project	Source file of example code
ThirdParty	
tinyusb-master	An open-source USB Host/Device stack for embedded system. Its official website is: https://www.tinyusb.org/

圖 4-1 目錄資訊

5. 範例程式執行

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

6. 修訂紀錄

Date	Revision	Description
2025.04.04	1.00	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.