USBD Bulk Communication

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	示範如何以批量傳輸型態做 USB 通訊
BSP 版本	NUC230/NUC240 Series BSP CMSIS v3.01.002
開發平台	NuTiny-EVB-NUC240 V1.2

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

本範例實現了一個USB批量裝置與一個用來與此裝置溝通的PC端應用程序。PC端程序藉由調用批量應用程式介面實現對USB批量裝置的讀寫控制。USB批量裝置可以從PC取得資料並將該資料原樣返回，同時，該資料也會顯示於窗口命令處理器視窗。

1.2 原理

當USB批量裝置連接至主機時，主機需要安裝放置在Driver_BulkUSB資料夾中的驅動程式來辨識USB裝置。該驅動程式可以使對應的裝置運作，且讓中央處理器進行各項應用及操作。當裝置與主機正確連接時，它們就可以開始傳輸資料以及互相溝通了。

本範例為PC端提供了批量應用程式讓使用者可以輕易地存取USB批量裝置的IN/OUT端點。此應用程式所提供的應用程式介面列出如表 1所示。如想得知更多關於各個應用程式介面之功能描述、引入參數與回傳數值的資訊，請參考App_BulkUSB資料夾中BulkUSB源文件。

批量應用程式介面
BOOL CBulkUSB::OpenDevice(void)
BOOL CBulkUSB::WriteUSB(LPCVOID lpBuffer, DWORD nNumberOfBytesToWrite)
BOOL CBulkUSB::ReadUSB(LPVOID lpBuffer, DWORD nNumberOfBytesToRead)
void CBulkUSB::CloseDevice(void)

表 1 批量應用程式介面

上述所提到之驅動程式與批量應用程式放置路徑如下：
EC_NUC240_USBD_Bulk_Communication_V1.00\SampleCode\ExampleCode\NUC240_USBD_Bulk_Communication

1.3 執行結果

以下執行結果會輸出於窗口命令處理器視窗。

在編譯完批量應用程式後，使用者可以藉由窗口命令處理器將位在Debug資料夾下的BulkUSB.exe打開。當使用者執行此執行檔時，PC會先讀取USB批量裝置的供應商識別碼與製造商識別碼，接著再對此裝置寫64-byte的資料，最後再對其讀取64-byte的資料，如圖 1-1所

示。

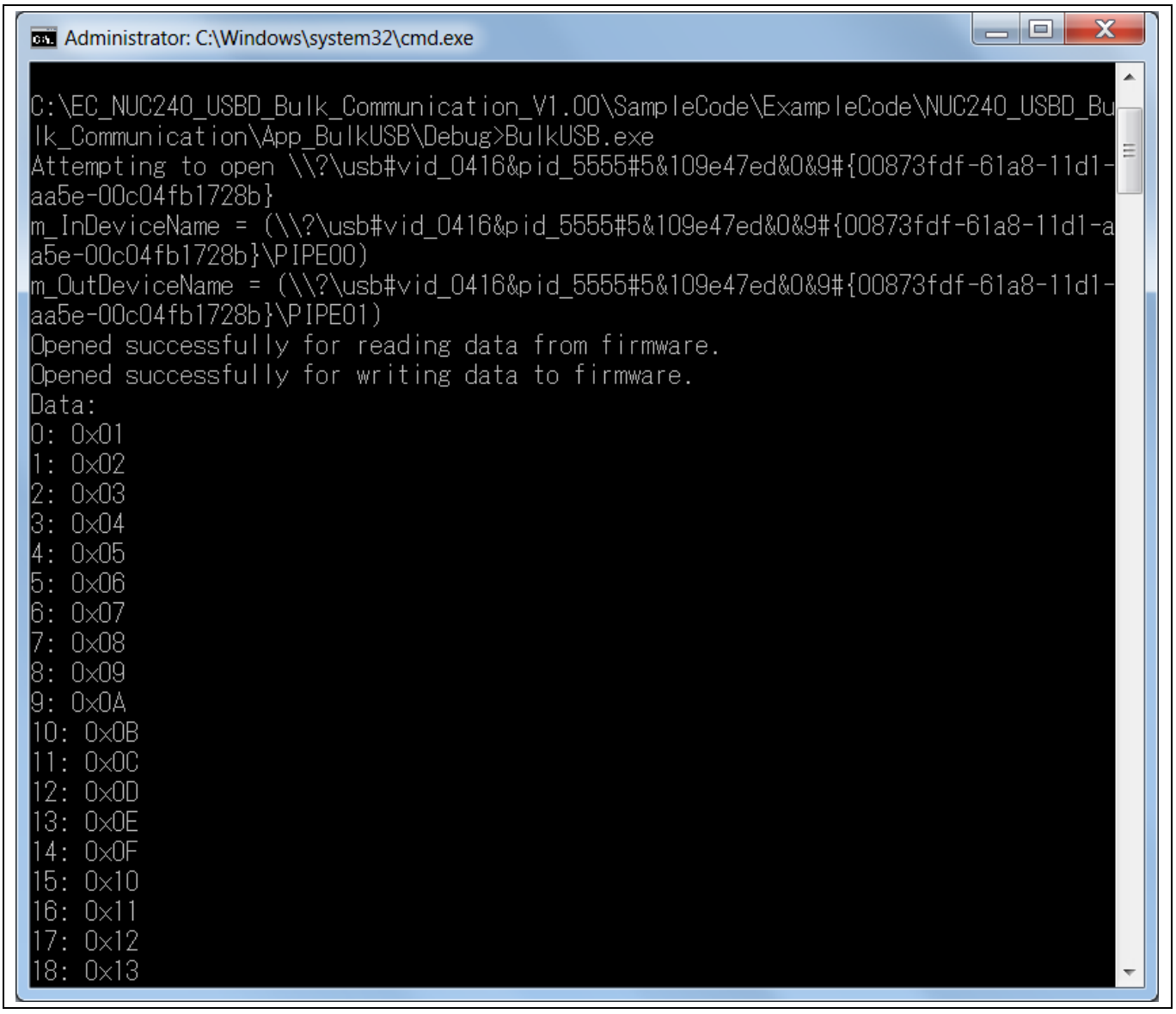
Administrator: C:\Windows\system32\cmd.exe
C:\EC_NUC240_USBD_Bulk_Communication_V1.00\SampleCode\ExampleCode\NUC240_USBD_Bulk_Communication\App_BulkUSB\Debug>BulkUSB.exe
Attempting to open \\?\usb#vid_0416&pid_5555#5&109e47ed&0&9#{00873fdf-61a8-11d1-aa5e-00c04fb1728b}
m_InDeviceName = (\\?\usb#vid_0416&pid_5555#5&109e47ed&0&9#{00873fdf-61a8-11d1-aa5e-00c04fb1728b}\PIPE00)
m_OutDeviceName = (\\?\usb#vid_0416&pid_5555#5&109e47ed&0&9#{00873fdf-61a8-11d1-aa5e-00c04fb1728b}\PIPE01)
Opened successfully for reading data from firmware.
Opened successfully for writing data to firmware.
Data:
0: 0x01
1: 0x02
2: 0x03
3: 0x04
4: 0x05
5: 0x06
6: 0x07
7: 0x08
8: 0x09
9: 0x0A
10: 0x0B
11: 0x0C
12: 0x0D
13: 0x0E
14: 0x0F
15: 0x10
16: 0x11
17: 0x12
18: 0x13

圖 1-1 執行結果

2 代碼介紹

VCOM_Init()函式用來做端點的配置，包括端點型態、端點緩衝區等等。在本範例中，端點 2 被配置為批量 IN 端點，而端點 3 被配置為批量 OUT 端點：

```
void VCOM_Init(void)
{
    ...

    /* EP2 ==> Bulk IN endpoint, address 1 */
    USBD_CONFIG_EP(EP2, USBD_CFG_EPMODE_IN | BULK_IN_EP_NUM);
    /* Buffer offset for EP2 */
    USBD_SET_EP_BUF_ADDR(EP2, EP2_BUF_BASE);

    /* EP3 ==> Bulk Out endpoint, address 2 */
    USBD_CONFIG_EP(EP3, USBD_CFG_EPMODE_OUT | BULK_OUT_EP_NUM);
    /* Buffer offset for EP3 */
    USBD_SET_EP_BUF_ADDR(EP3, EP3_BUF_BASE);
    /* trigger receive OUT data */
    USBD_SET_PAYLOAD_LEN(EP3, EP3_MAX_PKT_SIZE);
}
```

當裝置接收到 PC 所發出用來發送數據的 OUT token 時，端點 3 中斷處理程序中的 EP3_Handler()函式將會被執行。在 EP3_Handler()函式中，裝置會根據從 PC 接收來的資料準備資料在端點 2 的緩衝區。而後，當裝置接收到 IN token 時，儲存在端點 2 緩衝區的資料就會被送回給 PC：

```
void EP3_Handler(void)
{
    int i;

    /* Bulk OUT */
    gu32RxSize = USBD_GET_PAYLOAD_LEN(EP3);
    gpu8TxBuf = (uint8_t *)(USBD_BUF_BASE + USBD_GET_EP_BUF_ADDR(EP2));
    gpu8RxBuf = (uint8_t *)(USBD_BUF_BASE + USBD_GET_EP_BUF_ADDR(EP3));

    for (i = 0; i < 64; i++)
        gpu8TxBuf[i] = gpu8RxBuf[i] + 1;
```

```
    /* Ready to get next BULK out */
    USBD_SET_PAYLOAD_LEN(EP3, EP3_MAX_PKT_SIZE);
    USBD_SET_PAYLOAD_LEN(EP2, EP2_MAX_PKT_SIZE);
}
```

3 軟體與硬體環境

- **軟體環境**
  - BSP 版本
    - NUC230/NUC240 Series BSP CMSIS v3.01.002
  - IDE 版本
    - Keil uVision V5.28
- **硬體環境**
  - 電路元件
    - NuTiny-EVB-NUC240
  - 示意圖
    - 如圖 3-1 所示，將 NUC240 USBD 藉由 mini USB 傳輸線連接至 PC；分別將 NUC240 的引腳：UART0_RX (PB.0)、UART0_TX (PB.1)與 PC 的 UART_TX、UART_RX 進行連接。

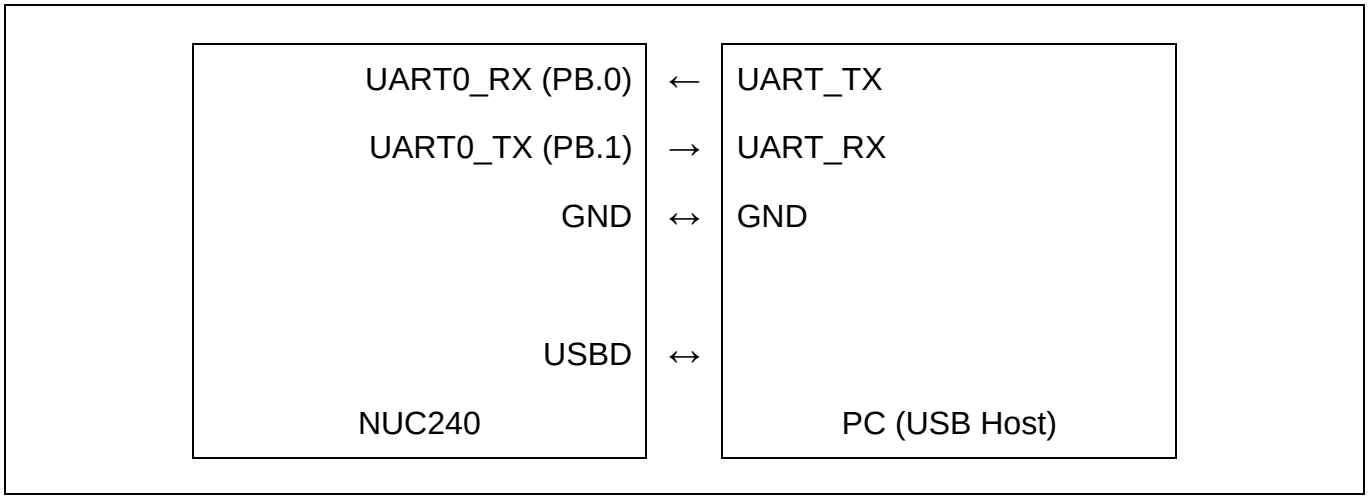

圖 3-1 NUC240 與 PC 接線圖

4 目錄資訊

- EC_NUC240_USBD_Bulk_Communication_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 NUC240_USBD_Bulk_Communication.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
Sep. 19, 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.