USBD Bulk Communication

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	示范如何以批量传输型态做 USB 通讯
BSP 版本	NUC230/NUC240 Series BSP CMSIS v3.01.002
开发平台	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 功能介绍

1.1 简介

本范例实现了一个USB批量装置与一个用来与此装置沟通的PC端应用程序。PC端程序藉由调用批量应用程序编程接口实现对USB批量装置的读写控制。USB批量装置可以从PC取得数据并将该数据原样返回，同时，该数据也会显示于窗口命令处理器窗口。

1.2 原理

当USB批量装置连接至主机时，主机需要安装放置在Driver_BulkUSB文件夹中的驱动程序来辨识USB装置。该驱动程序可以使对应的装置运作，且让中央处理器进行各项应用及操作。当装置与主机正确连接时，它们就可以开始传输数据以及互相沟通了。
本范例为PC端提供了批量应用程序让用户可以轻易地存取USB批量装置的IN/OUT端点。此应用程序所提供的应用程序编程接口列出如表 1所示。如想得知更多关于各个应用程序编程接口之功能描述、引入参数与回传数值的信息，请参考App_BulkUSB文件夹中BulkUSB源文件。

批量应用程序编程接口
BOOL CBulkUSB::OpenDevice(void)
BOOL CBulkUSB::WriteUSB(LPCVOID lpBuffer, DWORD nNumberOfBytesToWrite)
BOOL CBulkUSB::ReadUSB(LPVOID lpBuffer, DWORD nNumberOfBytesToRead)
void CBulkUSB::CloseDevice(void)

表 1 批量应用程序编程接口

上述所提到之驱动程序与批量应用程序放置路径如下：
EC_NUC240_USBD_Bulk_Communication_V1.00\SampleCode\ExampleCode\NUC240_USBD_Bulk_Communication

1.3 执行结果

以下执行结果会输出于窗口命令处理器窗口。
在编译完批量应用程序后，用户可以藉由窗口命令处理器将位在Debug文件夹下的BulkUSB.exe打开。当使用者执行此执行檔时，PC会先读取USB批量装置的供货商标识符与制造商标识符，接着再对此装置写64-byte的数据，最后再对其读取64-byte的数据，如图 1-1所示。

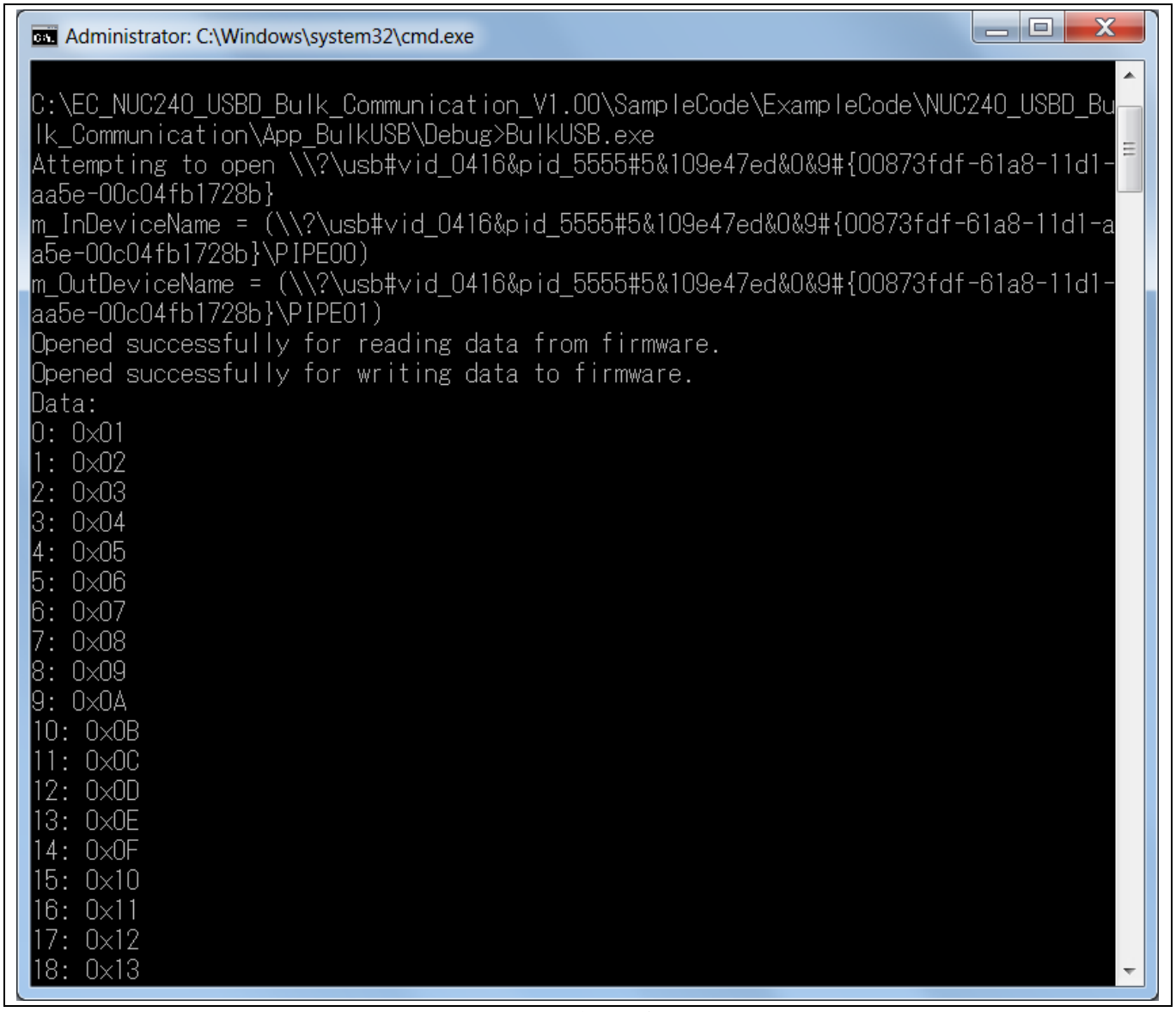
Administrator: C:\Windows\system32\cmd.exe
C:\EC_NUC240_USBD_Bulk_Communication_V1.00\SampleCode\ExampleCode\NUC240_USBD_Bulk_Communication\App_BulkUSB\Debug>BulkUSB.exe
Attempting to open \\?\usb#vid_0416&pid_5555#5&109e47ed&0&9#{00873fdf-61a8-11d1-aa5e-00c04fb1728b}
m_InDeviceName = (\\?\usb#vid_0416&pid_5555#5&109e47ed&0&9#{00873fdf-61a8-11d1-aa5e-00c04fb1728b}\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

图 1-1 执行结果

2 代码介绍

VCOM_Init()函式用来做端点的配置，包括端点型态、端点缓冲区等等。在本范例中，端点 2 被配置为批量 IN 端点，而端点 3 被配置为批量 OUT 端点：

```
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);
}
```

当装置接收到 PC 所发出用来发送数据的 OUT token 时，端点 3 中断处理程序中的 EP3_Handler()函式将会被执行。在 EP3_Handler()函式中，装置会根据从 PC 接收来的数据准备数据在端点 2 的缓冲区。而后，当装置接收到 IN token 时，储存在端点 2 缓冲区的数据就会被送回给 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 软件与硬件环境

- 软件环境
  - BSP 版本
    - NUC230/NUC240 Series BSP CMSIS v3.01.002
  - IDE 版本
    - Keil uVision V5.28
- 硬件环境
  - 电路组件
    - NuTiny-EVB-NUC240
  - 示意图
    - 如图 3-1 所示，将 NUC240 USBD 藉由 mini USB 传输线连接至 PC；分别将 NUC240 的引脚：UART0_RX (PB.0)、UART0_TX (PB.1)与 PC 的 UART_TX、UART_RX 进行连接以在 PC 显示执行结果。

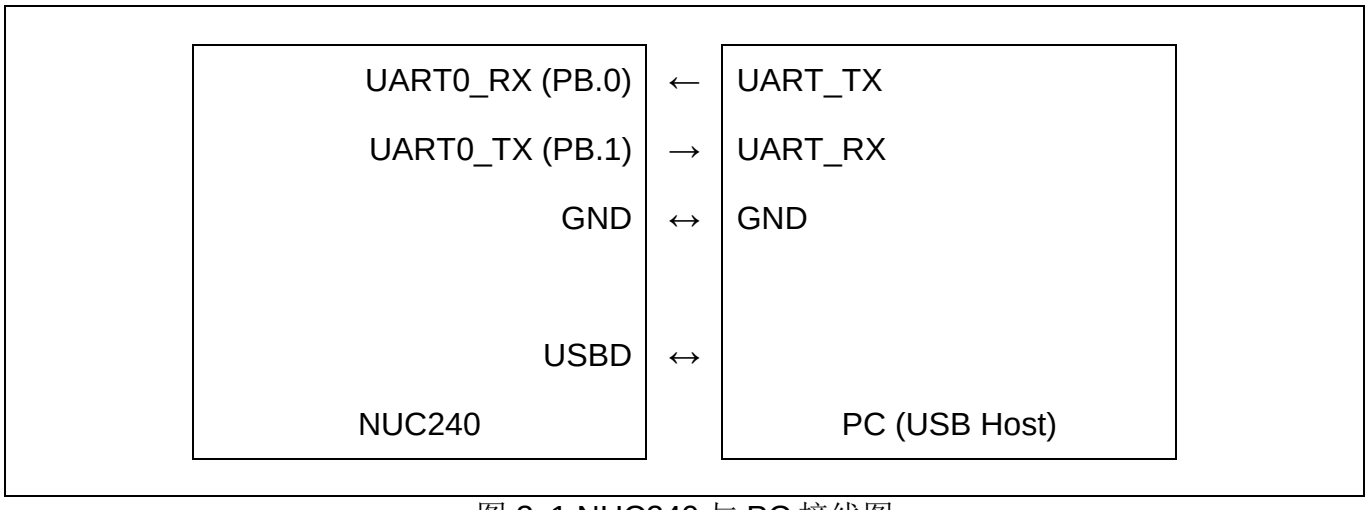

图 3-1 NUC240 与 PC 接线图

4 目录信息

- 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 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 NUC240_USBD_Bulk_Communication.uvproj。
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码

6 修订纪录

Date	Revision	Description
Sep. 19, 2019	1.00	1. 初始发布.

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.