USBD HID To EEPROM

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	实作从 PC 端透过 USB HID 介面控制 NUC240 端的 I2C 传输协定以存取 EEPROM 装置
BSP 版本	NUC230/240 Series BSP v3.01.002
开发平台	NuTiny–EVB–NUC240–LQFP100 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 功能介绍

1.1 简介

此范例程式演示如何在PC端透过USB HID(Human Interface Device)的连接来控制NUC240的I2C资料传输，以进行对EEPROM装置的存取。

1.2 原理

此范例程式的运作流程，是让使用者可从PC端下命令，经由USB HID介面的中断传输，将命令传到NUC240，进而存取EEPROM装置。而NUC240端则等待USB连线，以USB的端点3 (End Point 3) 来控制传输。当端点3发出传输中断，亦即USB连线成功后，NUC240端开始接收命令并执行I2C读写的动作，以进行对EEPROM装置的存取。如图1-1所示：

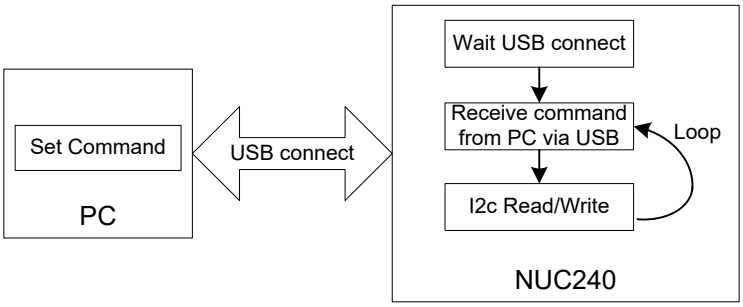

图 1-1 PC 与 NUC240 的连线与传输流程控制

1.3 执行结果

在此范例程式中需使用到两个windows的工具，分别为HIDTransferTest和HID_I2C_TOOL。以下为取得这两个open source code的连结：

HIDTransferTest的连结：
https://github.com/OpenNuvoton/NUC472_442BSP/tree/master/SampleCode/StdDriver/USBD_HID_Transfer/WindowsTool/HIDTransferTest

HID_I2C_TOOL的连结：
http://bbs.21ic.com/icview-1103894-1-1.html

将 PC 与 NUC240 的 USB Port 连接后，执行此范例程式，在 PC 的 UART 视窗上会出现初始讯息。如图 1-2 所示：

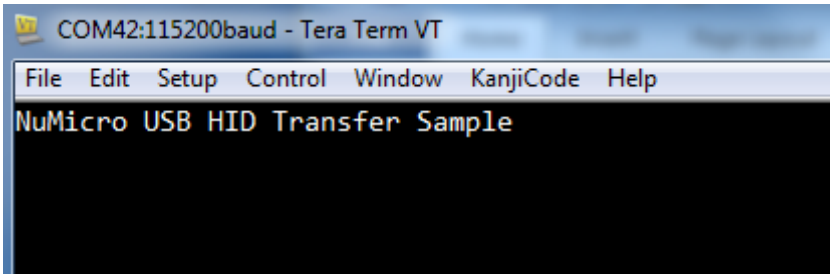

图 1-2 执行范例程式的初始讯息

接着开启执行档 HIDTransferTest.exe，会出现一个视窗显示连线成功的讯息。如图 1-3 所示：

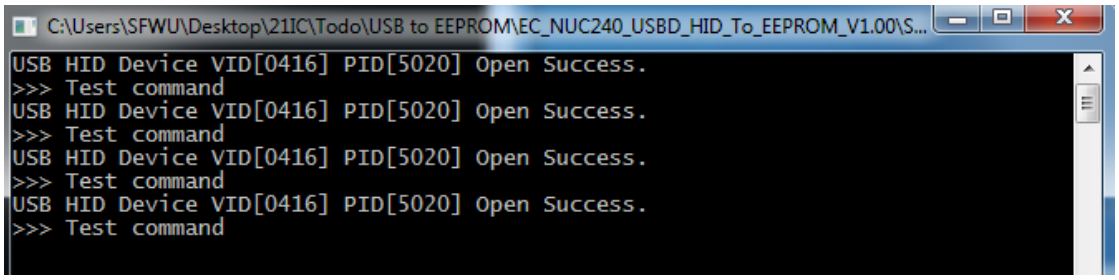

图 1-3 侦测 USB HID 连线成功的讯息

在开启执行档 HIDTransferTest.exe 显示成功讯息后，在 PC 的 UART 视窗上会显示出 I2C 传输资料的讯息，表示 I2C 已启动资料传输。如图 1-4 所示：

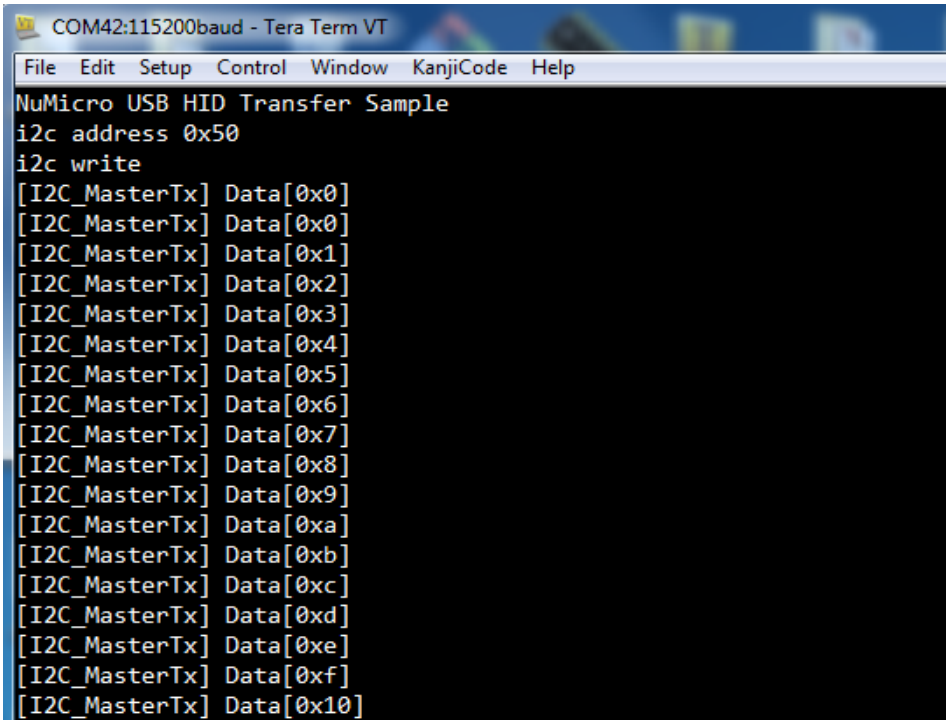

图 1-4 USB HID 连线后，与 EEPROM 传输资料的讯息

接着开启执行档 HID_I2C_TOOL.exe，会出现一个视窗如图 1-5 所示。使用者可在此介面上设定传输资料的起始位置(OFFSET 栏位)、长度(LENGTH 栏位)以及传输资料的值(Value 栏位)。

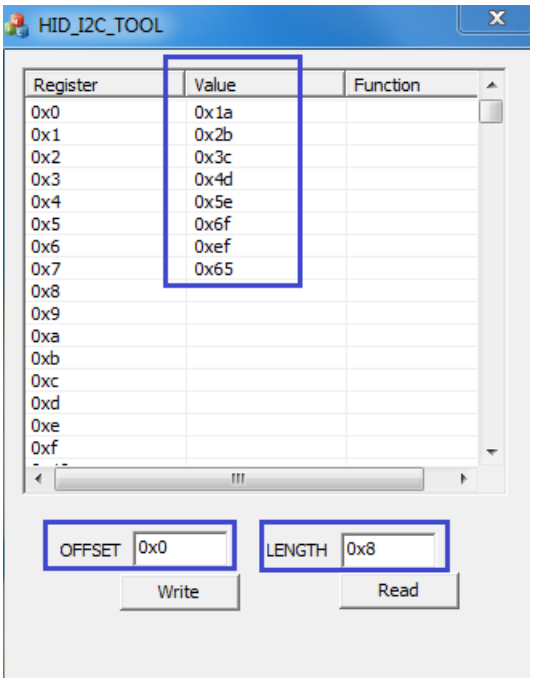

图 1-5 HID_I2C_TOOL 介面与初始设定

初始设定完成后，可选择执行读取(Read 按键)或写入(Write 按键)。若选择写入，则按下 Write 按键。如图 1-6 所示：

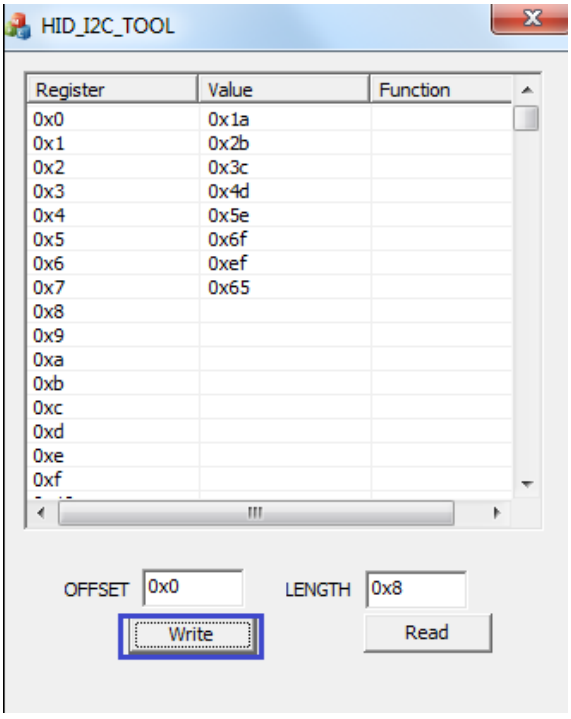

图 1-6 在 HID_I2C_TOOL 介面上执行 I2C 写入资料

按下 Write 按键，PC 上的 UART 视窗则会显示写入 EEPROM 装置的资料。如图 1-7 所示：

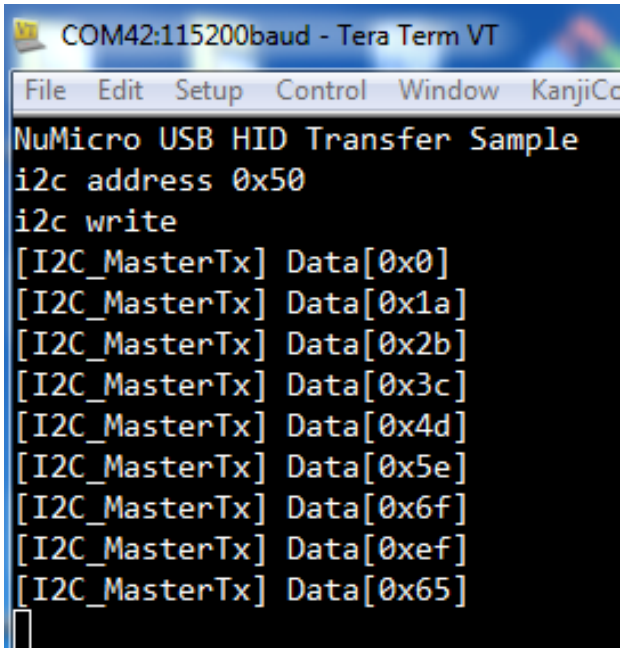

图 1-7 在 UART 视窗上显示 I2C 写入的资料

若选择读取，则按下 Read 按键。如图 1-8 所示：

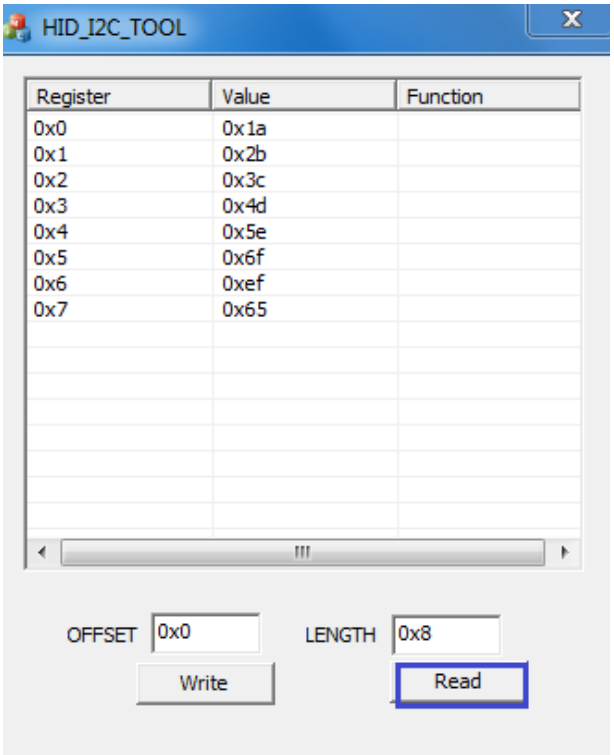

图 1-8 在 HID_I2C_TOOL 介面上执行 I2C 读取资料

按下 Read 按键，PC 上的 UART 视窗则会显示读取 EEPROM 装置的资料。如图 1-9 所示：

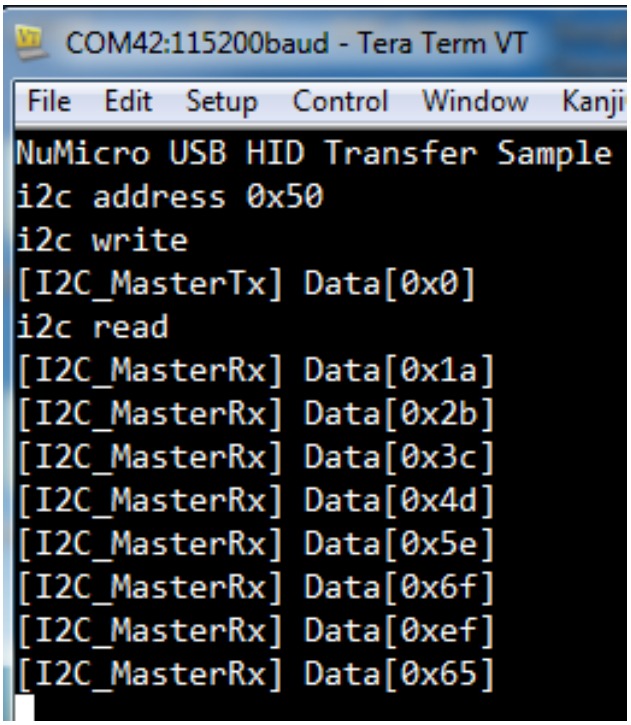

图 1-9 在 UART 视窗上显示 I2C 读出的资料

2 代码介绍

下方的函式为 EEPROM 的初始设定，其中包含设定 I2C 的 SCL、SDA 的 PIN 脚、I2C 传输频率、以及 I2C 中断的开启：

```
/*---------------------------------------------------------------------------*/
/*  EEPROM Initial Function                                                  */
/*---------------------------------------------------------------------------*/
void I2C_EEPROM_Init(void)
{
    /* Init I2C PIN setting */
    I2C_PIN_Init();

    /* Init I2C */
    I2C_Open(I2C_EEPROM, 100000);

    /* Enable I2C interrupt and set corresponding NVIC bit */
    I2C_EEPROM->I2CON |= I2C_I2CON_EI_Msk;
    NVIC_EnableIRQ(I2C_EEPROM_IRQn);
}
```

下方的函式为 Write I2C EEPROM，其流程为设定 I2C Master TX 的 callback 函式后，启动 I2C 的传输：

```
/*---------------------------------------------------------------------------*/
/*  Write I2C EEPROM                                                         */
/*---------------------------------------------------------------------------*/
void I2C_EEPROM_Write(void)
{
    g_u8DataLen = 0;
    g_u8EndFlag = 0;

    /* I2C function to write data to slave */
    s_I2CHandlerFn = (I2C_FUNC)I2C_MasterTx;

    /* I2C as master sends START signal */
    I2C_SET_CONTROL_REG(I2C_EEPROM, I2C_I2CON_STA);

    /* Wait I2C Tx Finish */
    while (g_u8EndFlag == 0);
}
```

下方的函式为 Read I2C EEPROM，其流程为设定 I2C Master RX 的 callback 函式后，启动 I2C 的传输：

```
/*---------------------------------------------------------------------------*/
/*  Read I2C EEPROM                                                          */
/*---------------------------------------------------------------------------*/
uint8_t I2C_EEPROM_Read(void)
{
    g_u8DataLen = 0;
    g_u8EndFlag = 0;

    /* I2C function to write data to slave */
    s_I2CHandlerFn = (I2C_FUNC)I2C_MasterRx;

    /* I2C as master sends START signal */
    I2C_SET_CONTROL_REG(I2C_EEPROM, I2C_I2CON_STA);

    /* Wait I2C Tx Finish */
    while (g_u8EndFlag == 0);

    return g_u8RxData;
}
```

以下则为主程式，其流程为等待USB连线，以NUC240中USB的端点3来控制传输，当由端点3发出传输中断后，EEPRON装置端开始接收I2C传输命令(Read，Write)并执行：

```
/*---------------------------------------------------------------------------*/
/*  Main Function                                                            */
/*---------------------------------------------------------------------------*/
int32_t main(void)
{
    uint8_t *pu8PtrOut, *pu8PtrIn;

    /* initial settings ... */
    /*..................... */


    /* Open USB controller */
    USBD_Open(&gsInfo, HID_ClassRequest, NULL);

    /*Init Endpoint configuration for HID */
    HID_Init();
```

```
/* Start USB device */
USBD_Start();

/* Enable USB device interrupt */
NVIC_EnableIRQ(USBD_IRQn);

while(1)
{
    if(g_u8EP3Ready==1)
    {
        g_u8EP3Ready=0;

        /* Interrupt OUT */
        pu8PtrOut = (uint8_t *)(USBD_BUF_BASE + USBD_GET_EP_BUF_ADDR(EP3));
        pu8PtrIn = (uint8_t *)(USBD_BUF_BASE + USBD_GET_EP_BUF_ADDR(EP2));

        /* Get command form HID and copy to command queue */
        USBD_MemCopy((uint8_t *)&g_tCmd, pu8PtrOut, 64);

        /* Set EEPROM device address */
        if(g_tCmd.u8Command==I2C_SET_ADDRESS)
        {
            g_u8DeviceAddr=g_tCmd.u8Size;
            printf("i2c address 0x%x\n\r",g_tCmd.u8Size);
        }

        /* Write data to EEPROM device */
        if(g_tCmd.u8Command==I2C_WRITE)
        {
            printf("i2c write\n\r");
            I2C_EEPROM_Write();
        }

        /* Read data from EEPROM device */
        if(g_tCmd.u8Command==I2C_READ)
        {
            printf("i2c read\n\r");
            I2C_EEPROM_Read();
        }
        g_tCmd.u8PID+=1;
```

```
            /* Write command to HID */
            USBD_MemCopy(pu8PtrIn, (void *)&g_tCmd, 64);
            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
    - NUC230/240 Series BSP CMSIS v3.01.002
  - IDE 版本
    - Keil uVersion 5.28
- 硬件环境
  - 电路组件
    - NuTiny–EVB–NUC240–LQFP100 V1.0
  - 示意图
    - 将 UART TX（PB.1）pin 脚连接到 PC UART RX，以显示示例代码的执行结果于 PC 上，并将 NUC240 USB 藉由 mini USB 传输线连接至 PC。
    - 将 NUC240 的 I2C1_SDA 和 I2C1_SCL 分别与 EEPROM 的 SDA 和 SCL 连接。

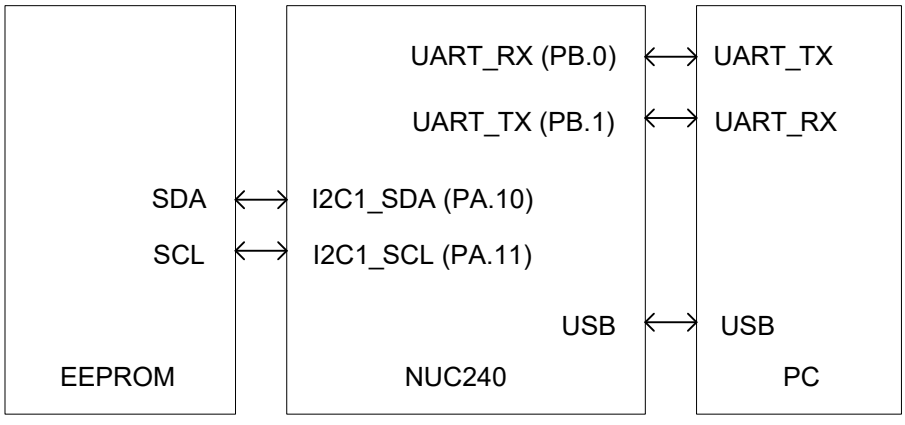

图 3-1 NUC240 和 PC、EEPROM连接图

4 目录信息

- EC_NUC240_USBD_HID_To_EEPROM_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 USBD_HID_To_EEPROM.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. 16, 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.