USBD HID SetReport GetReport

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Demonstrate how to send the command and data of set/get report request in control endpoint
BSP Version	NUC200 Series BSP CMSIS v3.00.003
Hardware	NuTiny-EVB-NUC220 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 Function Description

1.1 Introduction

In the USB communication, the interrupt and bulk endpoints are usually used for data transfer. This example code implements a HID (Human Interface Devices) class device, which uses control endpoint to do command and data transfer of Set_Report and Get_Report request. Moreover, a PC-side application is also implemented to communicate with that device.

1.2 Principle

USB HID class device communicates with the HID class driver using either the control pipe or an interrupt pipe as shown in Figure 1-1. In this example code, the control pipe is used for transmitting the input report from device using Get_Report request when polled by the HID class driver, and the output report is transmitted to device through the control endpoint using Set_Report request.

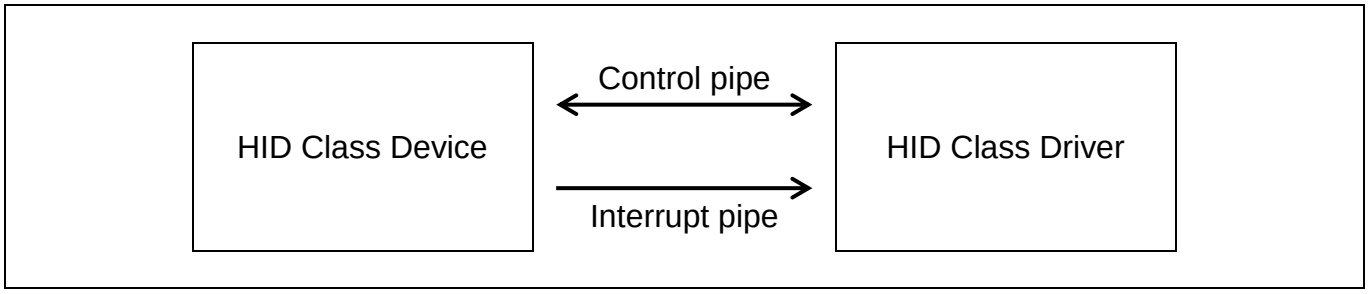

Figure 1-1 HID Class Communication Diagram

The Get_Report request is one of the class-specific requests; it allows the host to receive a report from the device. The Get_Report Request format is listed in Table 1. Among of them, the *wValue* field specifies Report Type in the high byte, and the Report Type is specified as Table 2.

Part	Offset/Size (Bytes)	Description
bmRequestType	0/1	10100001
bRequest	1/1	GET_REPORT (0x01)
wValue	2/2	Report Type and Report ID
wIndex	4/2	Interface
wLength	6/2	Report Length

Table 1 Get_Report Request Format

Value	Report Type
0x1	Input

0x2	Output
0x3	Feature
0x4-0xFF	Reserved

Table 2 Report Type

Similarly, the Set_Report request is also one of the class-specific requests; it allows the host to send a report to the device, possibly setting the state of input, output or feature controls. The meaning of the request fields for the Set_Report request is the same as for the Get_Report request, but the data direction is reversed. The Set_Report Request format is listed in Table 3. No matter which kind of request in class-specific requests, the transactions are done through the control pipe; therefore, they must follow the control pipe request format defined in the USB specification.

Part	Offset/Size (Bytes)	Description
bmRequestType	0/1	00100001
bRequest	1/1	SET_REPORT (0x09)
wValue	2/2	Report Type and Report ID
wIndex	4/2	Interface
wLength	6/2	Report Length

Table 3 Set_Report Request Format

Furthermore, this example code provides PC-side with report application to let user easily receive/send report from/to USB HID device. The APIs used in this application provided by Microsoft are listed in Table 4. For more information about the function description, incoming parameter and return value of each API, please refer to the header file, hidsdi, located in HIDTransfer_Report/HIDTransferTest folder.

Report APIs
BOOLEAN __stdcall HidD_SetOutputReport(HANDLE HidDeviceObject, PVOID ReportBuffer, ULONG ReportBufferLength)
BOOLEAN __stdcall HidD_GetInputReport(HANDLE HidDeviceObject, PVOID ReportBuffer, ULONG ReportBufferLength)

Table 4 Report APIs

The report application mentioned above is placed in the directory of EC_NUC220_USBD_HID_SetReport_GetReport_V1.00\SampleCode\ExampleCode\NUC220_USBD_HID_SetReport_GetReport.

1.3 Demo Result

The following execution results will be output to window command processor.
After you build the report application, you can open HIDTransferTest.exe located in Debug

folder via window command processor. When you execute the execution file, PC receives an input report from the USB HID device firstly, and then sends an output report to it. The execution result as shown in Figure 1-2.

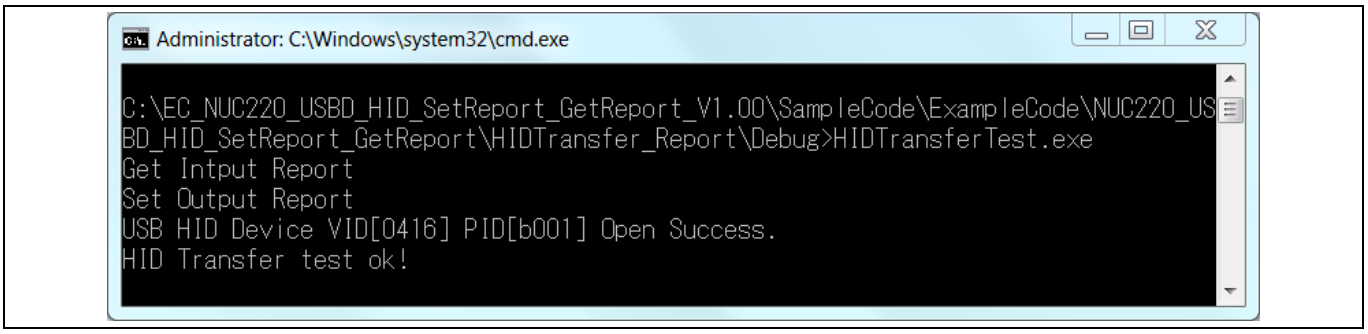

Figure 1-2 Execution Result

2 Code Description

In the function of HID_ClassRequest(), PC will issue a SET_REPORT request within HID class-specific requests to device and receive a GET_REPORT request within HID class-specific requests from device.

```
void HID_ClassRequest(void)
{
    uint8_t buf[8];

    USBD_GetSetupPacket(buf);

    if (buf[0] & 0x80)    /* request data transfer direction */
    {
        // Device to host
        switch (buf[1])
        {
            case GET_REPORT:
            {
                HID_CtrlSetupGetReport();
                break;
            }

            ...

            default:
            {
                /* Setup error, stall the device */
                USBD_SetStall(EP0);
                USBD_SetStall(EP1);
                break;
            }
        }
    }
    else
    {
        // Host to device
        switch (buf[1])
        {
            case SET_REPORT:
            {
```

```
                HID_CtrlSetupSetReport();
                break;
            }

            ...

            default:
            {
                /* Setup error, stall the device */
                USBD_SetStall(EP0);
                USBD_SetStall(EP1);
                break;
            }
        }
    }
}
```

3 Software and Hardware Environment

Software Environment

- BSP version
  - NUC200 Series BSP CMSIS v3.00.003
- IDE version
  - Keil uVision V5.28

Hardware Environment

- Circuit components
  - NuTiny-EVB-NUC220
- Diagram
  - As shown in Figure 3-1, connect NUC220 USBD to PC via a mini USB cable; connect NUC220 pins: UART0_RX (PB.0) and UART0_TX (PB.1) to PC: UART_TX and UART_RX respectively.

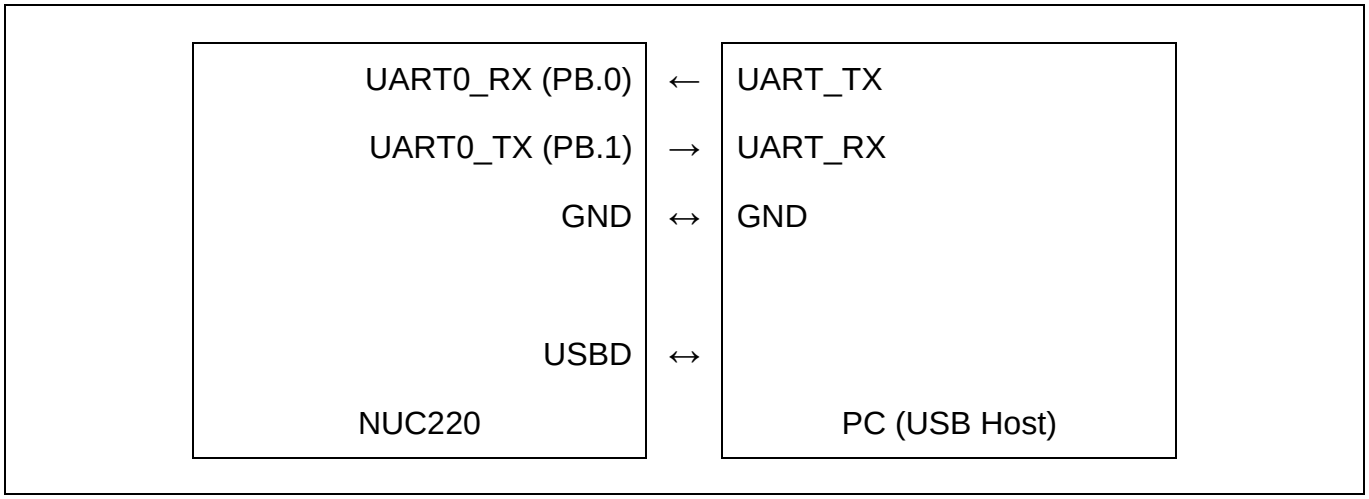

Figure 3-1 NUC220 and PC Connection

4 Directory Information

- EC_NUC220_USBD_HID_SetReport_GetReport_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 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click NUC220_USBD_HID_SetReport_GetReport.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Sep. 23, 2019	1.00	1. Initially issued.

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.