Application Note

32-bit Cortex™-M0 MCU
NuMicro® Family

Creating USB HID (Human Interface Device) in NUC120/140

Application Note

Table of Contents-

1 INTRODUCTION

This document explains the sample code, **"**`Smpl_HID`**"** which is included in the ***AN_1001_EN.ZIP*** file and demonstrates how to create an USB HID (Human Interface Device) device through the related libraries.

1.1 Scope

This article is provided for programmers applying the USB IP to USB applications. It is assumed that the reader is familiar with the Universal Serial Bus (USB) Specification 1.1 and the Device Class Definition for Human Interface Devices (HID).

1.2 Features

- Support customization of VID (Vendor ID) / PID (Product ID).
- Support customization of a HID device.

1.3 Limitation

- Interrupt-Out transfer is not supported.
- Limit the total configuration descriptor size to Control packet size.(64 bytes)

2 CODE SECTION

2.1 Conditional Compiling

The sample code "Smpl_USB_HID" supports HID keyboard or HID mouse by using conditional compiling. To configure the Smpl_USB_HID to be HID keyboard, the macro HID_FUNCTION should be defined as HID_KEYBOARD otherwise it should be defined as HID_MOUSE.

```
#define HID_KEYBOARD      1
#define HID_MOUSE         2
#define HID_FUNCTION      HID_MOUSE
```

2.2 Main Function (in Smpl_HID.c)

Refer chapter of Calling Sequence for its calling sequence.

In the main function, the PLL clock should be set to 48MHz for USB PHY clock. The system clock is set to external 12MHz to handle all USB & HID events.

After the hardware initialization is finished, the DrvUSB_Open() will be called to initialize USB endpoints information structures for HID device. Now the USB interrupt is enabled and ready to wait for USB plug-in. The status will be changed from eDRVUSB_DETACHED to eDRVUSB_ATTACHED after USB is plugged in. When the USB is attached, the HID_Init will be called to initialize the HID vendor descriptor to be **HID keyboard** or **HID mouse**.

```
// Poll and handle USB events.
while(1)
{
        eUsbState = DrvUSB_GetUsbState();
        DrvUSB_DispatchEvent();
        if (eUsbState == eDRVUSB_DETACHED)
        {
                break;
        }
}
```

2.3 Example to Define a Mouse

2.3.1 Mouse Report Descriptor (in HID_API.c)

A Report Descriptor defined as a mouse.

2.3.2 Get In Mouse Report (in HID_API.c)

For key activities, the global variable `g_au8KeyboardReport` is used to store the input keys. The definitions of the keys are defined in USB Usage Tables document of USB HID standard. The mouse activities are reported by the global variable `g_au8MouseReport` and the function definition of the variable is as follows:

g_au8MouseReport[0]							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0**
Reserved					Middle Button	Right Button	Left Button

g_au8MouseReport[1]							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0**
Offset of X axis. Must be -127 ~ +127.							

g_au8MouseReport[2]							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0**
Offset of Y axis. Must be -127 ~ +127.							

3 CALLING SEQUENCE

3.1 Run as an USB Device

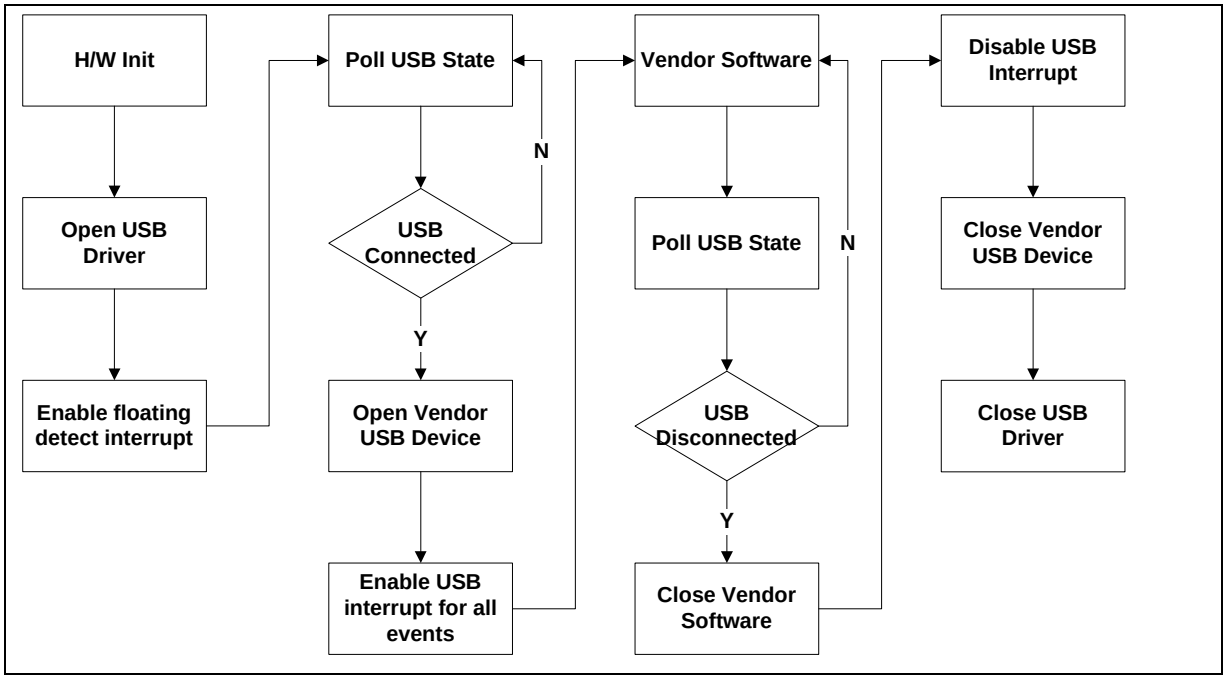

Figure 3-1 Control Flow of Running as an USB Device

1. Initialize Clock Control settings (`PLLCON` and `CLKSEL0` registers)
2. Open USB driver (Prepare USB buffer, setup information and enable NVIC IRQ)
3. Initialize USB interrupt and enable floating-detect
4. Poll USB until it is connected to Host
5. Open Vendor USB device (In this case, it's a HID device)
6. Vendor software connects.
7. Poll USB connection and dispatch USB-related event if disconnected.
8. Remove Vendor software.
9. Disable USB Interrupt.
10. Close Vendor USB Device.
11. Close USB driver.

3.2 Customize a USB Device

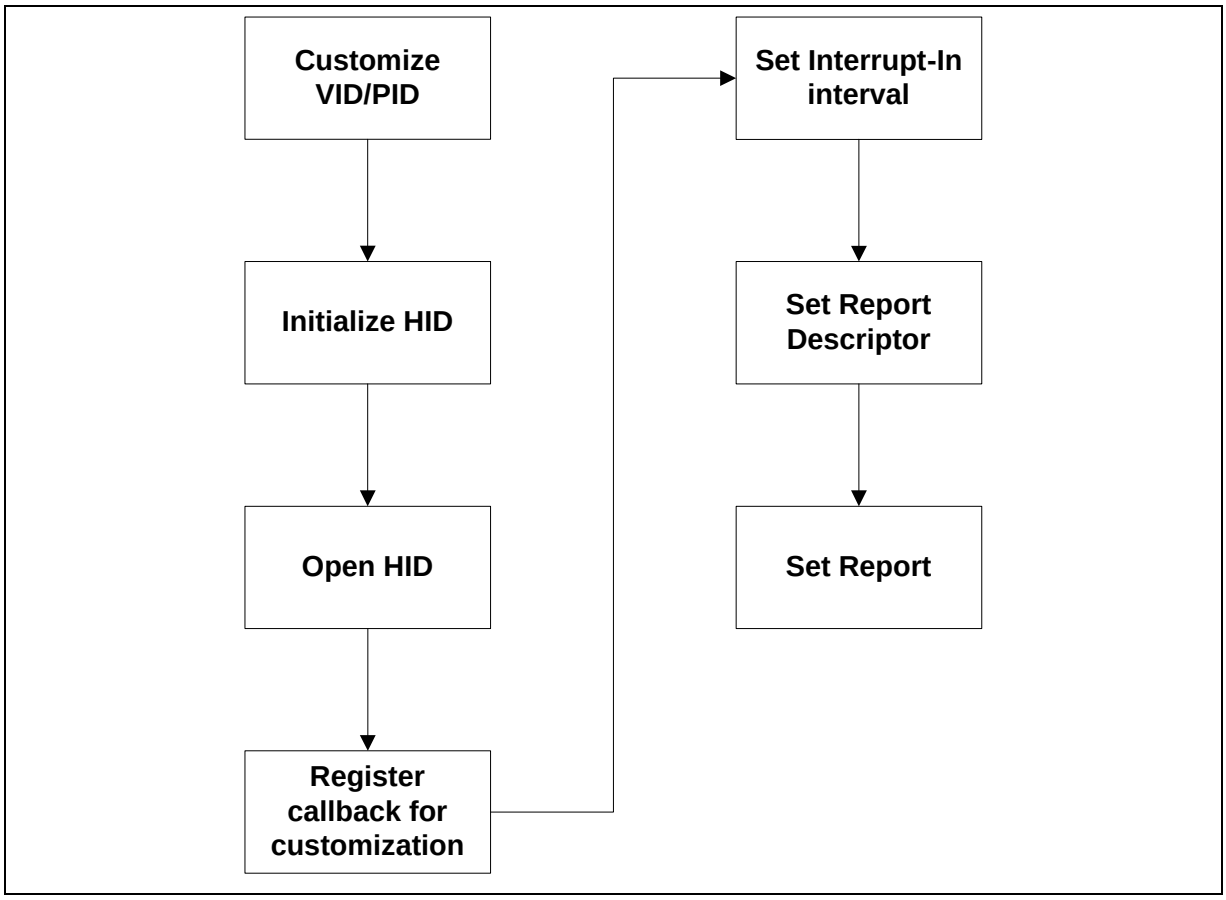

Figure 3-2 Control Flow of Customize Vendor USB Device as HID

1. Customize VID / PID if needed. Such as,0x0416/0xC1401
2. Open HID library to Initialize.
3. Create a HID Device after HID_Open.
4. Register USB-related callback functions for customization if needed. For example HID_CtrlSetupGetReport
5. Set customized Interrupt-In interval if needed.
6. Set customized Report Descriptor.
7. Set customized Report.

3.3 API Usage Reference

- `USB Driver Reference Guide.doc`

4 EXECUTION ENVIRONMENT SETUP AND RESULT

4.1 Test Smpl_USB_HID

The HID sample code, `Smpl_USB_HID`, could be built by Keil MDK tool and download to NUC1xx series DEV Board through ICE. Then user is able to execute the code in ICE environment or reset the DEV board to execute the code which had been programmed in on-chip Program Flash to verify the `Smpl_USB_HID` code.

4.2 Result

4.2.1 HID_MOUSE

When the DEV board executes this sample code, the DEV board performs the behavior as a mouse. Once the USB port is connected to the PC with Windows OS, there will be a mouse cursor moving to right on the screen..

4.2.2 HID_KEYBOARD

When the DEV board executes this sample code, the DEV board performs the behavior as keyboard. User can open an editor tool, for example the Microsoft Notepad, to test it and character “A” will repeatedly appear on the editor because this sample code emulates that key “A” is pressed continuously.

5 REVISION HISTORY

REV.	DATE	DESCRIPTION
0.01	February 25, 2010	1. Initially issued.

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

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.