M480 8K Mouse

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This example code uses M480 HS-USBD to simulate a HID mouse with 8K report rate and support remote wake-up.
BSP Version	M480 Series BSP CMSIS v3.03.000
Hardware	NuMaker-ETM-M487 v1.2 NuMaker-PFM-M487 v3.0

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 and microprocessor 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 Overview

This example code uses HS-USBD to simulate a HID mouse and supports these features:

- Supports 8K report rate
- Supports remote wake-up
- Enters Power-down mode when receiving SUSPEND signal, and exits Power-down mode when receiving RESUME signal

The resume packet issues USB bus from J-state to K-state and keeps K-state at least 20ms, makes M480 enter the LLPD (low leakage power-down) mode and wires an I/O (PA.6) to the HSUSBD_D+ and detects the falling edge to wake up the system from LLPD mode.

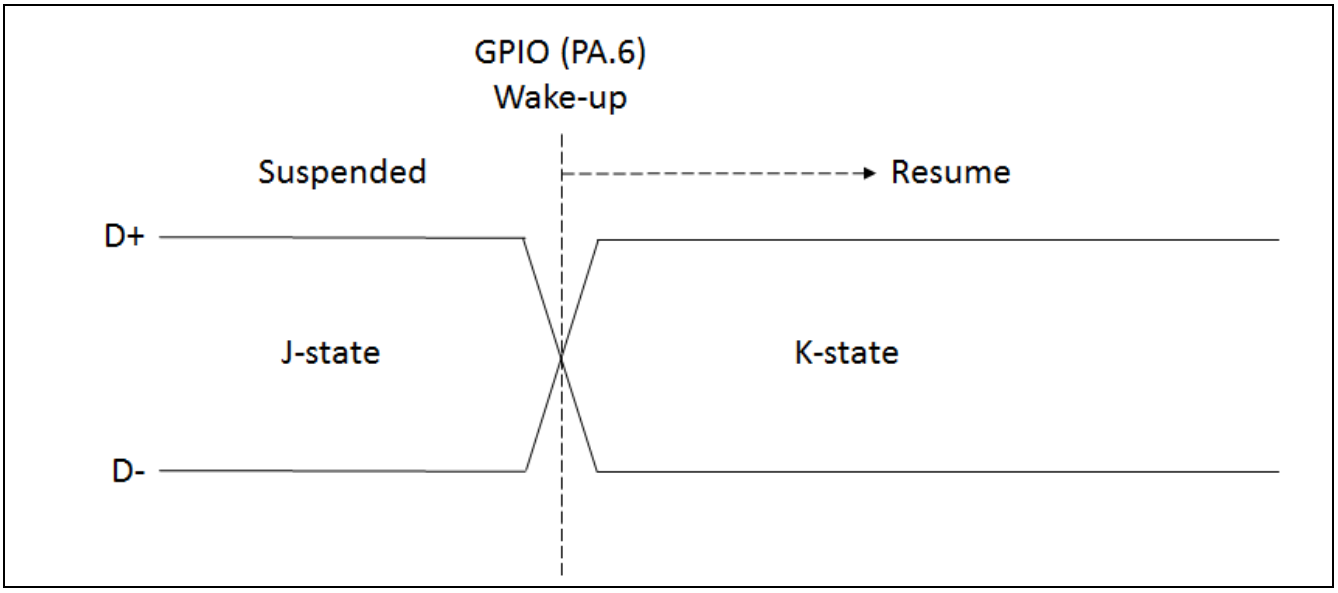
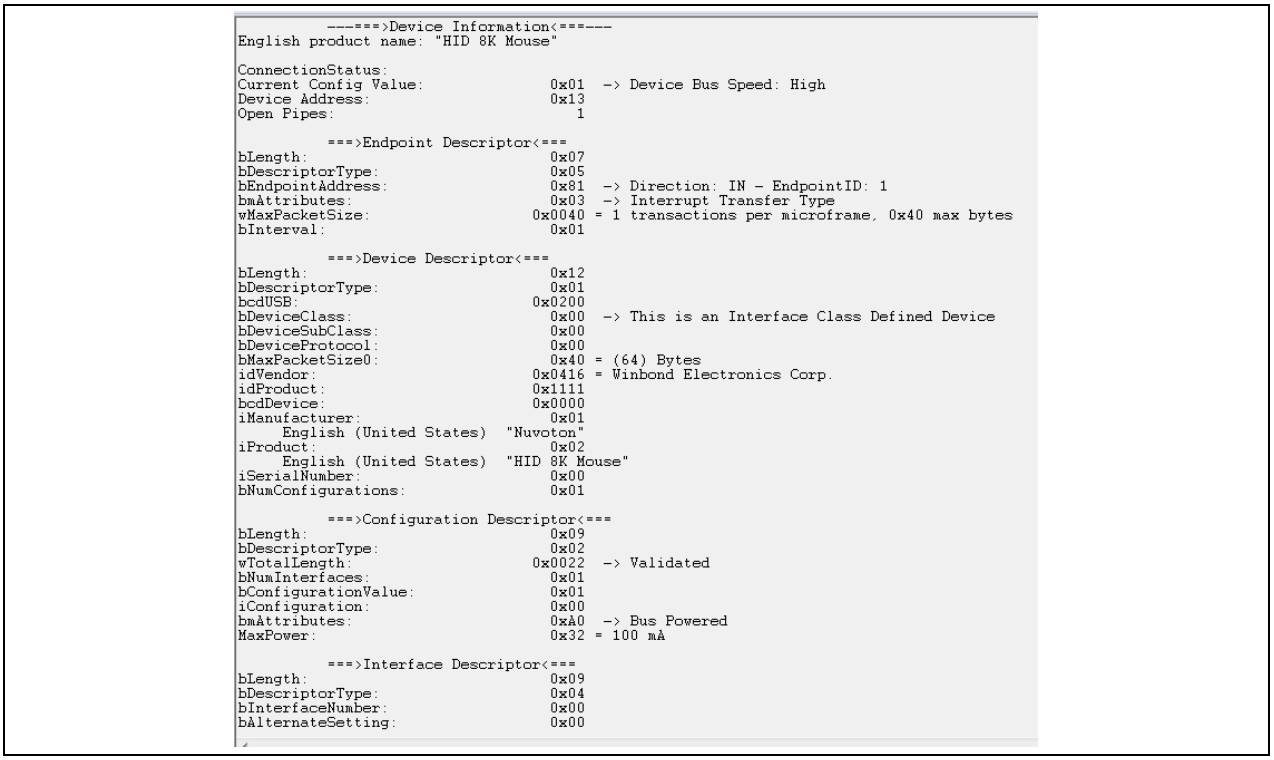


Figure 1-1 USB Signal

1.1 Demo Result

- HID Mouse

  The M480 is played as a HID mouse when connected with a computer, and uses one interrupt-in endpoint whose interval is '0x01' (125us).



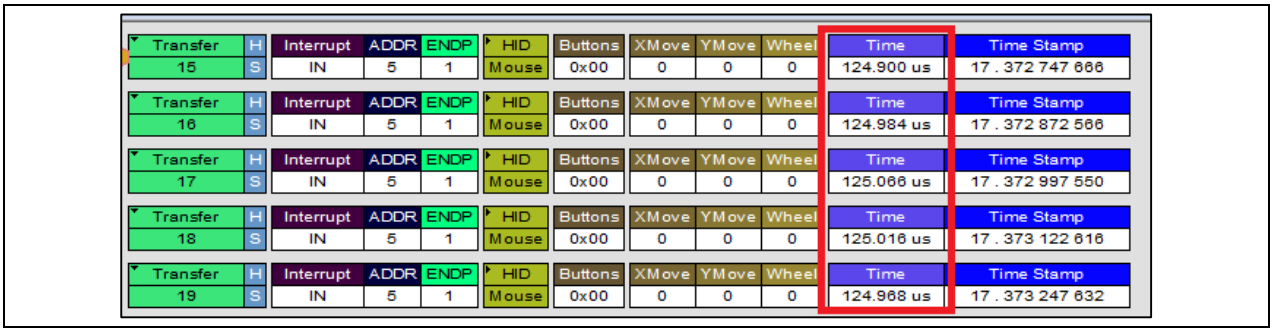
```
          ---===>Device Information<===---
English product name: "HID 8K Mouse"

ConnectionStatus:
Current Config Value:              0x01  -> Device Bus Speed: High
Device Address:                    0x13
Open Pipes:                           1

          ===>Endpoint Descriptor<===
bLength:                           0x07
bDescriptorType:                   0x05
bEndpointAddress:                  0x81  -> Direction: IN - EndpointID: 1
bmAttributes:                      0x03  -> Interrupt Transfer Type
wMaxPacketSize:                  0x0040 = 1 transactions per microframe, 0x40 max bytes
bInterval:                         0x01

          ===>Device Descriptor<===
bLength:                           0x12
bDescriptorType:                   0x01
bcdUSB:                          0x0200
bDeviceClass:                      0x00  -> This is an Interface Class Defined Device
bDeviceSubClass:                   0x00
bDeviceProtocol:                   0x00
bMaxPacketSize0:                   0x40 = (64) Bytes
idVendor:                        0x0416 = Winbond Electronics Corp.
idProduct:                       0x1111
bcdDevice:                       0x0000
iManufacturer:                     0x01
     English (United States)  "Nuvoton"
iProduct:                          0x02
     English (United States)  "HID 8K Mouse"
iSerialNumber:                     0x00
bNumConfigurations:                0x01

          ===>Configuration Descriptor<===
bLength:                           0x09
bDescriptorType:                   0x02
wTotalLength:                    0x0022  -> Validated
bNumInterfaces:                    0x01
bConfigurationValue:               0x01
iConfiguration:                    0x00
bmAttributes:                      0xA0  -> Bus Powered
MaxPower:                          0x32 = 100 mA

          ===>Interface Descriptor<===
bLength:                           0x09
bDescriptorType:                   0x04
bInterfaceNumber:                  0x00
bAlternateSetting:                 0x00
```

Figure 1-2 USB Device Information

Pressing the buttons on the NuMaker-ETM-M487/ NuMaker-PFM-M487 can control the cursor to move right or left.

- 8K Report Rate

  The interval between each interrupt-in packet is around 125us, which means this device has 8K report rate.



Transfer	H/S	Interrupt	ADDR	ENDP	HID	Buttons	XMove	YMove	Wheel	Time	Time Stamp
15	S	IN	5	1	Mouse	0x00	0	0	0	124.900 us	17.372747666
16	S	IN	5	1	Mouse	0x00	0	0	0	124.984 us	17.372872566
17	S	IN	5	1	Mouse	0x00	0	0	0	125.066 us	17.372997550
18	S	IN	5	1	Mouse	0x00	0	0	0	125.016 us	17.373122616
19	S	IN	5	1	Mouse	0x00	0	0	0	124.968 us	17.373247632

Figure 1-3 8K Report Rate

- Remote Wake-up
  Before PC enters the sleep mode, the HOST sends the SET_FEATURE request to enable the remote wake-up function and let the USB bus enter SUSPEND.

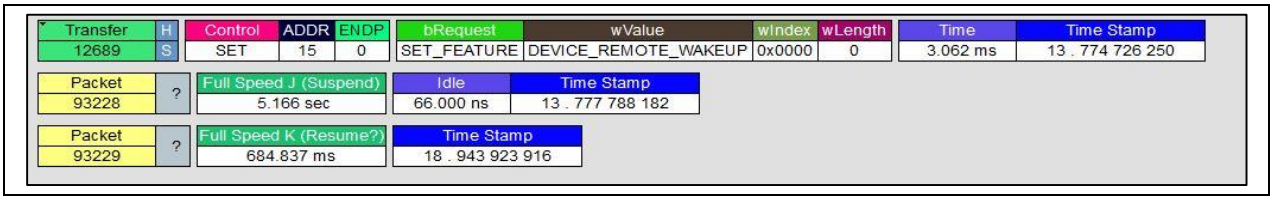


Figure 1-4 Remote Wake-up Packet

2 Code Description

- HID Mouse Report Descriptor

```
//
// HID_MouseReportDescriptor
//
uint8_t HID_MouseReportDescriptor[] __attribute__((aligned(4))) = {
    0x05, 0x01,     /* Usage Page(Generic Desktop Controls) */
    0x09, 0x02,     /* Usage(Mouse) */
    0xA1, 0x01,     /* Collection(Application) */
    0x09, 0x01,     /* Usage(Pointer) */
    0xA1, 0x00,     /* Collection(Physical) */
    0x05, 0x09,     /* Usage Page(Button) */
    0x19, 0x01,     /* Usage Minimum(0x1) */
    0x29, 0x03,     /* Usage Maximum(0x3) */
    0x15, 0x00,     /* Logical Minimum(0x0) */
    0x25, 0x01,     /* Logical Maximum(0x1) */
    0x75, 0x01,     /* Report Size(0x1) */
    0x95, 0x03,     /* Report Count(0x3) */
    0x81, 0x02,     /* Input(3 button bit) */
    0x75, 0x05,     /* Report Size(0x5) */
    0x95, 0x01,     /* Report Count(0x1) */
    0x81, 0x01,     /* Input(5 bit padding) */
    0x05, 0x01,     /* Usage Page(Generic Desktop Controls) */
    0x09, 0x30,     /* Usage(X) */
    0x09, 0x31,     /* Usage(Y) */
    0x09, 0x38,     /* Usage(Wheel) */
    0x15, 0x81,     /* Logical Minimum(0x81)(-127) */
    0x25, 0x7F,     /* Logical Maximum(0x7F)(127) */
    0x75, 0x08,     /* Report Size(0x8) */
    0x95, 0x03,     /* Report Count(0x3) */
    0x81, 0x06,     /* Input(1 byte wheel) */
    0xC0,           /* End Collection */
    0xC0,           /* End Collection */
};
```

- HID Mouse Handler

```
//
// HID_mouseViaReportIf
//
void HID_mouseViaReportIf(void)
{
    uint32_t volatile i;

    if (g_u8EPAReady) {

         switch(user_btn_detect()) {
             case USER_BTN0:
                epa_buf[0] = 0x00;
                epa_buf[1] = 25;
                epa_buf[2] = 0x00;
                epa_buf[3] = 0x00;
                  break;
             case USER_BTN1:
                epa_buf[0] = 0x00;
                epa_buf[1] = (-25);
                epa_buf[2] = 0x00;
                epa_buf[3] = 0x00;
                  break;
             case NONE:
             default:
                epa_buf[0] = 0x00;
                epa_buf[1] = 0x00;
                epa_buf[2] = 0x00;
                epa_buf[3] = 0x00;
                  break;
        }

        g_u8EPAReady = 0;

        for (i = 0; i < 4; i++) {
            HSUSBD->EP[EPA].EPDAT_BYTE = epa_buf[i];
        }

        HSUSBD->EP[EPA].EPRSPCTL = HSUSBD_EP_RSPCTL_SHORTTXEN;
        HSUSBD_ENABLE_EP_INT(EPA, HSUSBD_EPINTEN_INTKIEN_Msk);
    }
}
```

- HID Mouse Power-Down Handler

```
//
// HID_powerDownHandler
//
void HID_powerDownHandler(void)
{
    if (g_u8pwrdwn) {
        g_u8pwrdwn = 0;

        SYS_UnlockReg();

        printf("Enter Power-Down Mode\r\n");
        while(!UART_IS_TX_EMPTY(UART0));

        //
        // Enable IRQ
        //
        NVIC_EnableIRQ(GPA_IRQn);
        NVIC_EnableIRQ(GPF_IRQn);
        NVIC_EnableIRQ(GPG_IRQn);

        //
        // Disable HS-USBD engine
        //
        HSUSBD->PHYCTL &= ~HSUSBD_PHYCTL_PHYEN_Msk;
        /* should delay > 2.5us between USB PHY suspend and system power down */
        CLK_SysTickDelay(50);

        //
        // Enable PA6 interrupt
        //
        GPIO_CLR_INT_FLAG(PA, BIT6);
        GPIO_ENABLE_DIGITAL_PATH(PA, BIT6);
        GPIO_EnableInt(PA, 6, GPIO_INT_FALLING);

        //
        // Enter LLPD mode
        //
        CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_LLPD);
        CLK_PowerDown();

        //
        // Disable PA.6 interrupt
        //
        GPIO_CLR_INT_FLAG(PA, BIT6);
        GPIO_DisableInt(PA, 6);
        GPIO_DISABLE_DIGITAL_PATH(PA, BIT6);

        //
        // Enable HS-USBD engine
        //
        HSUSBD->PHYCTL |= HSUSBD_PHYCTL_PHYEN_Msk;

        //
        // Wait HS-USBD PHY clock ready
        //
        while (1) {
            HSUSBD->EP[EPG].EPMPS = 0x20ul;
            if (HSUSBD->EP[EPG].EPMPS == 0x20ul) {
                break;
            }
        }

        //
        // Disable IRQ
```

```
        //
        NVIC_DisableIRQ(GPF_IRQn);
        NVIC_DisableIRQ(GPG_IRQn);
        NVIC_DisableIRQ(GPA_IRQn);

        printf("Exit Power-Down Mode\r\n");
        switch(g_wakeupSrc) {
        case USB_RESUME:
            printf("Wakeup by USB resume\r\n");
            g_wakeupSrc = SRC_NONE;
            break;
        case USB_RESET:
            printf("Wakeup by USB Reset\r\n");
            g_wakeupSrc = SRC_NONE;
            break;
        case USER_BUTTON:
            printf("Wakeup by I/O button\r\n");
            g_wakeupSrc = SRC_NONE;

            //
            // Perform remote wake-up
            //
            HID_remoteWakeupProcess();
            break;
        case SRC_NONE:
        default:
            break;
        }
        SYS_LockReg();
    } else {
        g_wakeupSrc = SRC_NONE;
    }
}
```

3 Software and Hardware Requirements

3.1 Software Requirements

- BSP version
- M480 Series BSP CMSIS v3.03.000
- IDE version
  - Keil uVersion 5.26

3.2 Hardware Requirements

- Circuit components
  - NuMaker-ETM-M487 / NuMaker-PFM-M487

To wake up the system from Power-down mode via the RESUME signal, you must wire a 100Ω resistor between HSUSBD_D+ and PA.6 pin.

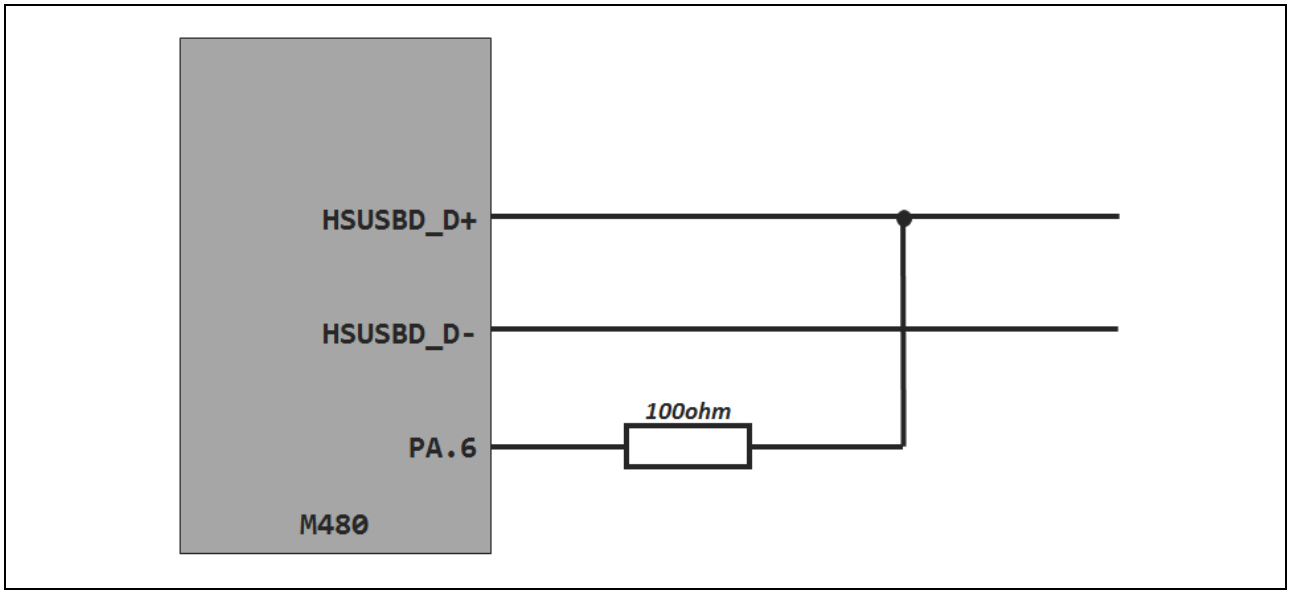


Figure 3-1 USB Circuit

4 Directory Information

The directory structure is shown below.

EC_M480_8K_Mouse_LLPD_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 Example Code Execution

1. Browse the sample code folder as described in the Directory Information section and double-click *M480_8K_Mouse_LLDP.uvproj*.
2. Enter Keil compile mode.
   - Build
   - Download
   - Start/Stop debug session
3. Enter debug mode.
   - Run

6 Revision History

Date	Revision	Description
2021.06.22	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.