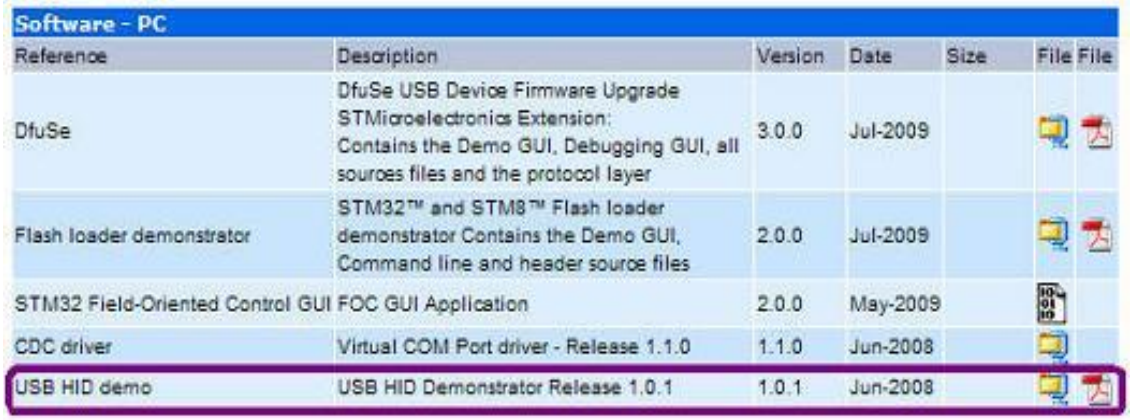
USB HID 실습

2009.11.20

USB HID demo 다운로드

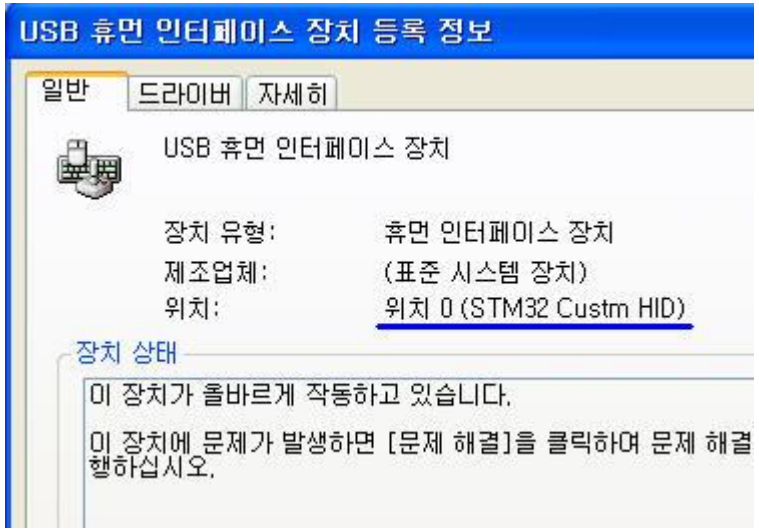


Software - PC

Reference	Description	Version	Date	Size	File	File
DfuSe	DfuSe USB Device Firmware Upgrade STMicroelectronics Extension: Contains the Demo GUI, Debugging GUI, all sources files and the protocol layer	3.0.0	Jul-2009			
Flash loader demonstrator	STM32™ and STM8™ Flash loader demonstrator Contains the Demo GUI, Command line and header source files	2.0.0	Jul-2009			
STM32 Field-Oriented Control GUI	FOC GUI Application	2.0.0	May-2009			
CDC driver	Virtual COM Port driver - Release 1.1.0	1.1.0	Jun-2008			
USB HID demo	USB HID Demonstrator Release 1.0.1	1.0.1	Jun-2008			

- http://www.st.com/mcu/familiesdocs-110.html
- 문서 파일 14711.pdf는 프로그램에 대한 간단한 설명
- 압축 파일 um0551.zip를 풀면 HIDDemo_V1.0.1_Setup.exe 파일이 생긴다. 이것을 실행하면 윈도우 프로그램을 설치할 수 있다.

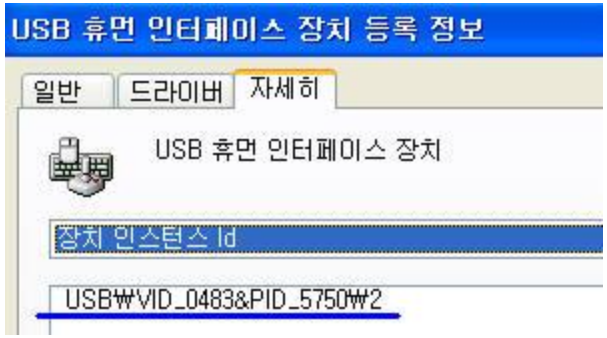
Device Description (1)



<usb_desc.c>

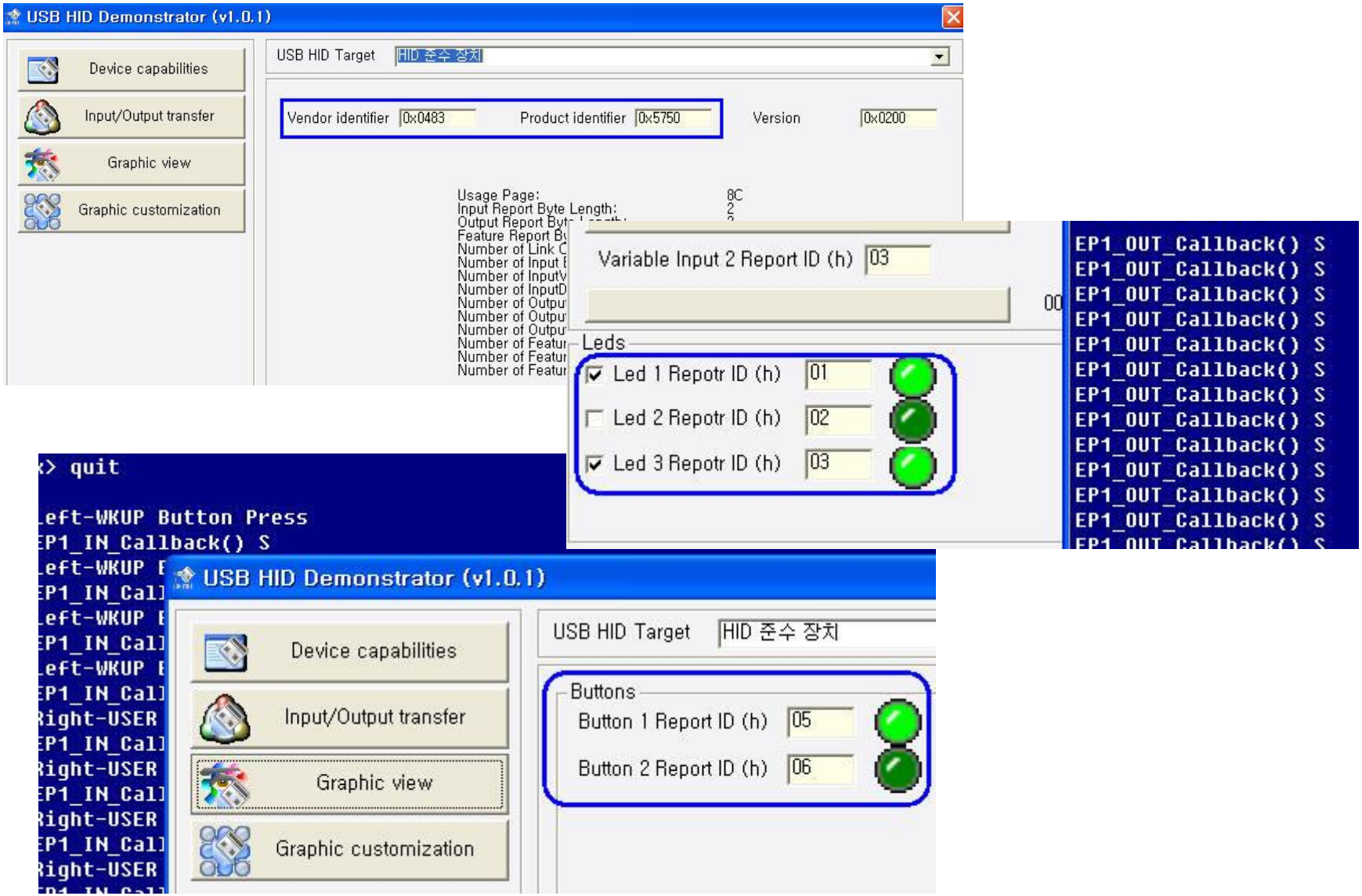
```
const uint8_t CustomHID_StringProduct[CUSTOMHID_SIZ_STRING_PRODUCT] =
  {
    CUSTOMHID_SIZ_STRING_PRODUCT,          /* bLength */
    USB_STRING_DESCRIPTOR_TYPE,        /* bDescriptorType */
    'S', 0, 'T', 0, 'M', 0, '3', 0, '2', 0, ' ', 0, 'C', 0,
    'u', 0, 's', 0, 't', 0, 'm', 0, ' ', 0, 'H', 0, 'I', 0,
    'D', 0
  };
```

Device Description (2)




```
/* USB Standard Device Descriptor */
const uint8_t CustomHID_DeviceDescriptor
                        [CUSTOMHID_SIZ_DEVICE_DESC] =
  {
    0x12,                       /*bLength */
    USB_DEVICE_DESCRIPTOR_TYPE, /*bDescriptorType*/
    0x00,                       /*bcdUSB */
    0x02,
    0x00,                       /*bDeviceClass*/
    0x00,                       /*bDeviceSubClass*/
    0x00,                       /*bDeviceProtocol*/
    0x40,                       /*bMaxPacketSize40*/
    0x83,                       /*idVendor (0x0483)*/
    0x04,
    0x50,                       /*idProduct = 0x5750*/
    0x57,
    0x00,                       /*bcdDevice rel. 2.00*/
    0x02,
    1,              /*Index of string descriptor describing manufacturer */
    2,              /*Index of string descriptor describing product*/
    3,              /*Index of string descriptor describing
                      the device serial number */
    0x01                        /*bNumConfigurations*/
  }
  ; /* CustomHID_DeviceDescriptor */
```

실행 결과


USB HID Demonstrator (v1.0.1)
Device capabilities
Input/Output transfer
Graphic view
Graphic customization
USB HID Target
HID 준수 장치
Vendor identifier 0x0483
Product identifier 0x5750
Version 0x0200
Usage Page: 8C
Input Report Byte Length: 2
Variable Input 2 Report ID (h) 03
Leds
Led 1 Repotr ID (h) 01
Led 2 Repotr ID (h) 02
Led 3 Repotr ID (h) 03
EP1_OUT_Callback() S
> quit
Left-WKUP Button Press
EP1_IN_Callback() S
USB HID Demonstrator (v1.0.1)
Device capabilities
Input/Output transfer
Graphic view
Graphic customization
USB HID Target
HID 준수 장치
Buttons
Button 1 Report ID (h) 05
Button 2 Report ID (h) 06

EXTI1_IRQHandler

```
static bool toggle_data_key2 = FALSE;

void EXTI1_IRQHandler(void)
{
   if(EXTI_GetITStatus(GPIO_KEY2_EXTI_Line) != RESET) {
      Send_Buffer[0] = 0x06;  // Button 2를 가리킴

      if(toggle_data_key2) {
         toggle_data_key2 = FALSE;  Send_Buffer[1] = 0x01;
      } else {
         toggle_data_key2 = TRUE;  Send_Buffer[1] = 0x00;
      }

      UserToPMABufferCopy(Send_Buffer, ENDP1_TXADDR, 2);
      SetEPTxCount(ENDP1, 2);
      SetEPTxValid(ENDP1);
      EXTI_ClearITPendingBit(GPIO_KEY2_EXTI_Line);
   }
}
```

EP1_OUT_Callback

```
void EP1_OUT_Callback(void)
{
   BitAction Led_State;
   PMAToUserBufferCopy(Receive_Buffer, ENDP1_RXADDR, 2);
   if (Receive_Buffer[1] == 0)    {  Led_State = Bit_SET; }
   else                           {  Led_State = Bit_RESET;  }

   switch (Receive_Buffer[0]) {
   case 1: GPIO_WriteBit(GPIO_LED, GPIO_LED1_PIN, Led_State);  break;
   case 2: GPIO_WriteBit(GPIO_LED, GPIO_LED2_PIN, Led_State);  break;
   case 3: GPIO_WriteBit(GPIO_LED, GPIO_LED3_PIN, Led_State);  break;
   default:
      GPIO_Write(GPIO_LED,
            ~(uint16_t)(GPIO_LED1_PIN | GPIO_LED2_PIN | GPIO_LED3_PIN ));
      break;
   }

   SetEPRxStatus(ENDP1, EP_RX_VALID);
}
```