

您設計產品時的好朋友！



[Forum: 16-bit PIC/dsPIC](#)

Topic: USB - HID使用EPO的IN回傳資料給Host?

Subject: USB - HID使用EPO的IN回傳資料給Host?

作者: srxrrrrr

2017年04月14日 17:37:33

MCU: PIC24FJ256GB110

Board: 開發板APP1632

IDE: MPLAB X IDE v3.51

Compiler: XC16 v1.26

參考MLA: v2017_03_06appsusbdevicehid_customfirmwareexp16_pic24fj256gb210_pim.x

透過MLA範例建立了一個hid_custom想透過預設的Control Pipe - Endpoint 0 來IN/OUT data;

參考了同樣是MLA上的hid_keyboard Project,

已可從Host端透過EPO OUT來下指令給MCU, 但卻無法透過EPO IN將資料回傳給Host...

經過研究, 認為應該是透過USBEP0SendRAMPtr函數來回傳資料給Host, 但卻不會使用(或者根本不對因)

希望大家能提供點建議, 感激不盡!!

app_device_custom_hid.c裡的相關Code(其他檔案的相關Code也有做相對應撰寫):

```
unsigned char ToSendDataBuffer_EP0[2];
```

```
static void USBHIDCBSetReportComplete(void)
```

```
{
```

```
    switch(CtrlTrfData[0]) //Look at the data the host sent, to see what kind of  
application specific command it sent.
```

```
{
```

```
    case 0x80: //Toggle LEDs command
```

```
        LED_Toggle(LED_USB_DEVICE_HID_CUSTOM);
```

```
        break;
```

```
    case 0x81:
```

```
        if(!HIDTxHandleBusy(USBInHandle))
```

```
        {
```

```
            ToSendDataBuffer_EP0[0]=1;
```

```
            ToSendDataBuffer_EP0[1]=2;
```

```
            //使用中斷型EP1的IN pipe來傳送資料給Host => Success!
```

```
            //USBInHandle = HIDTxPacket(CUSTOM_DEVICE_HID_EP,
```

```
(uint8_t*)&ToSendDataBuffer_EP0[0],2);
```

```
            //使用控制型EPO的IN pipe來傳送資料給Host => Fail...
```

```
            USBEP0SendRAMPtr((uint8_t*)&ToSendDataBuffer_EP0[0],2,USB_EPO_RAM);
```

```
        }
```

```
        break;
```

```
    default:
```

```
        break;
```

```

    }
}

void USBHIDCBSetReportHandler(void)
{
    /* Prepare to receive the keyboard LED state data through a SET_REPORT
    * control transfer on endpoint 0. The host should only send 1 byte,
    * since this is all that the report descriptor allows it to send. */
    USBEP0Receive((uint8_t*)&CtrlTrfData, USB_EPO_BUFF_SIZE, USBHIDCBSetReportComplete);
}

```

ps. 附檔為該段CODE的截圖，以便檢視

附加檔案：

EPO - IN.jpg(133.27 KB)

```

static void USBHIDCBSetReportComplete(void)
{
    switch(CtrlTrfData[0]) //Look at the data the host sent, to see what kind of application specific command it sent.
    {
        case COMMAND_TOGGLE_LED: //Toggle LEDs command
            LED_Toggle(LED_USB_DEVICE_HID_CUSTOM);
            break;
        case COMMAND_GET_BUTTON_STATUS:
            if(!HIDTxHandleBusy(USBInHandle))
            {
                ToSendDataBuffer_EP0[0]=1;
                ToSendDataBuffer_EP0[1]=2;

                //使用中斷型EP1的IN pipe來傳送資料給Host => Success!
                USBInHandle = HIDTxPacket(CUSTOM_DEVICE_HID_EP, (uint8_t*)&ToSendDataBuffer_EP0[0],2);

                //使用控制型EPO的IN pipe來傳送資料給Host => Fail...
                USBEP0SendRAMPtr((uint8_t*)&ToSendDataBuffer_EP0[0],2,USB_EPO_RAM);
            }
            break;
        default:
            break;
    }
}

void USBHIDCBSetReportHandler(void)
{
    /* Prepare to receive the keyboard LED state data through a SET_REPORT
    * control transfer on endpoint 0. The host should only send 1 byte,
    * since this is all that the report descriptor allows it to send. */
    USBEP0Receive((uint8_t*)&CtrlTrfData, USB_EPO_BUFF_SIZE, USBHIDCBSetReportComplete);
}

```

