

USB虚拟串口发送数据上位机不能显示问题

Questions: 在使用 USB 虚拟串口发送数据时，如果每帧发送的字节数一直是当前 BULK IN 端点的最大包长度（如 64Byte），上位机工具不能正常显示数据。

例如：在 V1.3.0 或更低版本的 BSP 中，VirtualComPort_loopback 例程使用上位机工具发送 64Byte 数据给 USB 设备，USB 设备将此 64Byte 数据再返回给上位机，我们在上位机的输出界面可能不能实时接收到 64Byte 数据。

Answer:

此种情况和 USB2.0 协议 bulk 传输的定义有关系，参考 usb2.0 specification 5.8 节 Bulk transfers 有如下描述：

An endpoint must always transmit data payloads with a data field less than or equal to the endpoint's reported *wMaxPacketSize* value. When a bulk IRP involves more data than can fit in one maximum-sized data payload, all data payloads are required to be maximum size except for the last data payload, which will contain the remaining data. A bulk transfer is complete when the endpoint does one of the following:

- Has transferred exactly the amount of data expected
- Transfers a packet with a payload size less than *wMaxPacketSize* or transfers a zero-length packet

简单来说：一个 bulk 传输完成有三种条件：

1. 确切知道要收发多少数据
2. 发送的一帧数据少于最大包长度
3. 发送一个 0 长度数据的包

我们看到前面的问题，USB 发送的数据一直是最大的包长度，会导致主机认为当前的 bulk 传输并没有全部完成，因此在上位机上看不到收到的数据。

解决方法：

1. 将 BSP 更新至最新版本。
2. 在 V1.3.0 或更低版本的 BSP 中，以 AT32F403A 的 BSP DEMO VirtualComPort_loopback 为例，采用条件 3 的方式来解决此问题。

在程序中增加对前面的数据是否传输完成的判断(红色粗体部分)，当前已经没有数据要传输的时候，USB device 发送一个 0 长度的数据包来告诉主机传输已经完成：

```
void EP1_IN_Callback(void)
{
    uint16_t dwSendLen = 0;
    uint32_t SendPtr = 0;
```

```
static uint8_t send0packet = 0;
/*no data need send*/
if (usb_txfifo.wrpointer == usb_txfifo.curpointer )
{
    if ( send0packet == 1 )
    {
        SetEPTxCount(ENDP1, 0);
        SetEPTxValid(ENDP1);
        send0packet = 0;
    }
    return;
}
if ( usb_txfifo.wrpointer > usb_txfifo.curpointer )
{
    dwSendLen = usb_txfifo.wrpointer - usb_txfifo.curpointer;
}
else
{
    dwSendLen = USB_FIFO_MAX - usb_txfifo.curpointer;
}
SendPtr = usb_txfifo.curpointer;
if ( dwSendLen > VIRTUAL_COM_PORT_DATA_SIZE )
{
    dwSendLen = VIRTUAL_COM_PORT_DATA_SIZE;
}
usb_txfifo.curpointer += dwSendLen;
if ( usb_txfifo.curpointer >= USB_FIFO_MAX )
usb_txfifo.curpointer = 0;
send0packet = 1;
/* send packet to PMA*/
UserToPMABufferCopy(&usb_txfifo.fifo[SendPtr], ENDP1_TXADDR, dwSendLen);
SetEPTxCount(ENDP1, dwSendLen);
SetEPTxValid(ENDP1);
}
```

以上修改在 1.3.0 之后版本的 BSP 中均有添加，在 2.x 的 BSP 中修改方式如下（在 BSP 中已添加，此处只是说明，并不需要用户自己添加）

```
while(1)
{
    /* get usb vcp receive data */
    data_len = usb_vcp_get_rxdata(&usb_core_dev, usb_buffer);

    if(data_len > 0 || send_zero_packet == 1)
```

```
{  
  
    /* bulk transfer is complete when the endpoint does one of the following  
    1 has transferred exactly the amount of data expected  
    2 transfers a packet with a payload size less than wMaxPacketSize or transfers a zero-length packet  
    */  
    if(data_len > 0)  
        send_zero_packet = 1;  
  
    if(data_len == 0)  
        send_zero_packet = 0;  
  
    timeout = 50000;  
    do  
    {  
        /* send data to host */  
        if(usb_vcp_send_data(&usb_core_dev, usb_buffer, data_len) == SUCCESS)  
        {  
            break;  
        }  
    }  
    }while(timeout --);  
}
```

按照如上代码逻辑，在每次接受到数据并回发给上位机时，在最后都会发一笔长度为 0 的数据包

类型： MCU 应用

适用型号： AT32F403, AT32F413, AT32F403A, AT32F407

主功能： USB

次功能： 无

文档版本历史

日期	版本	变更
2022.3.3	2.0.0	最初版本
2022.4.28	2.0.1	更新在2.x库中的修改方式

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）航天应用或航天环境；（D）武器，且/或（E）其他可能导致人身伤害、死亡及财产损害的应用。如果采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险及法律责任仍将由采购商单独承担，且采购商应独立负责在前述应用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2022 雅特力科技 保留所有权利