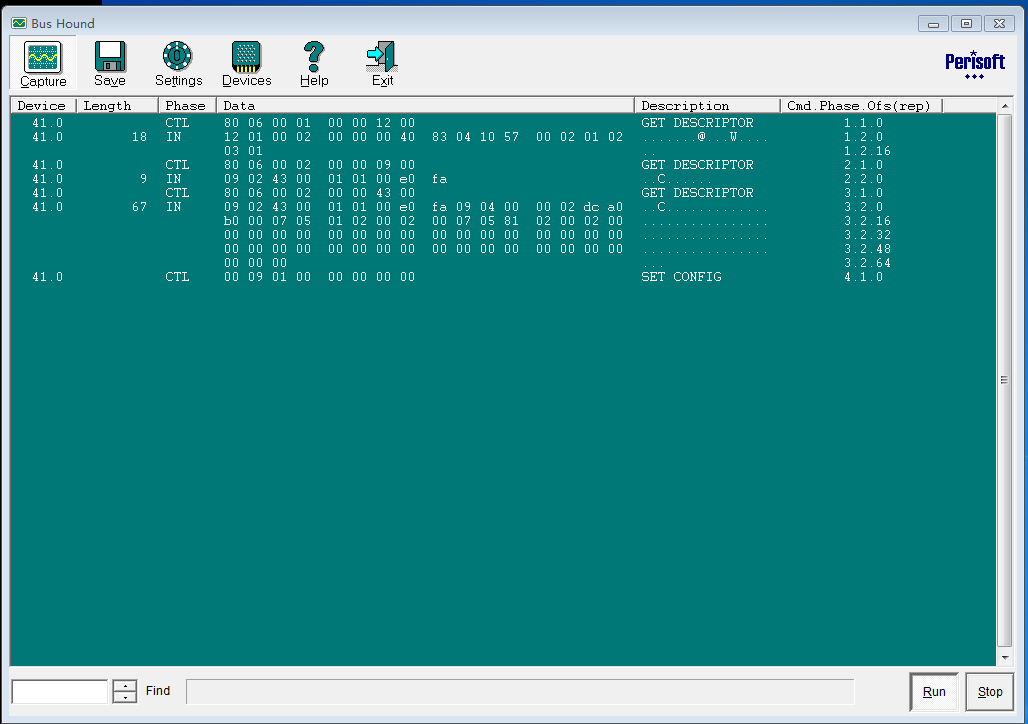
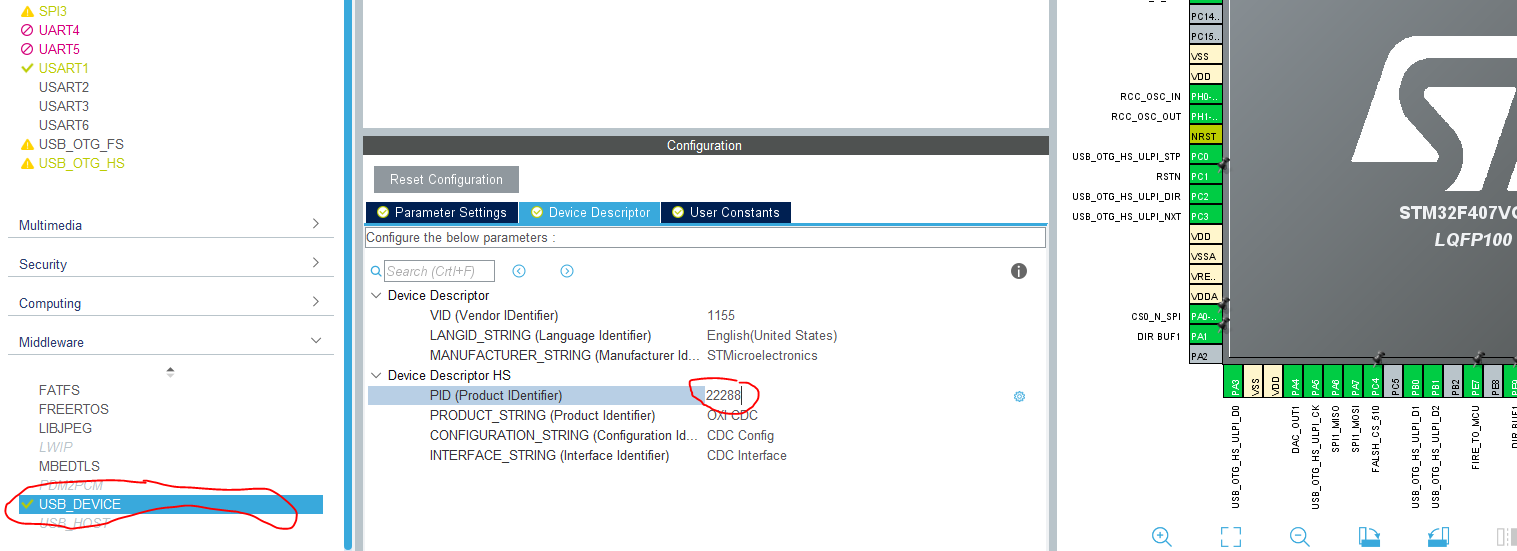
## USB HS枚举

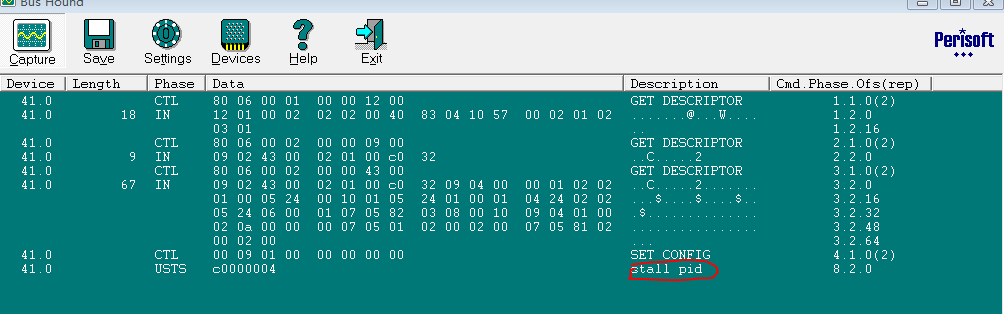
USB HS正常枚举（用之前测试版标准库code）bus hound抓取数据：



1. 刚开始枚举不上，因为PID不对，上位机的驱动程序将PID写死为22288，而stm32cubeMx默认的不是这个值，所以上位机不认识这个设备，只要将PID改为22288就可以枚举上了：



1. 另外，当ST heap设置较小时（小于0x400）,枚举时会出现stall pid（pid溢出），需将heap设置为大于或等于0x400：



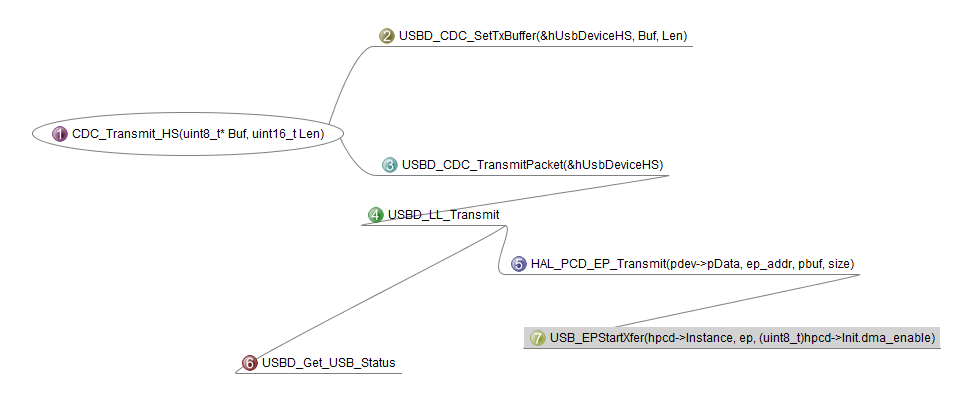
1. 将USB设置为HS / ULPI时，只用设置与USB ULPI有关的管脚，不用设置USB FS的管脚。

## USB HS receive

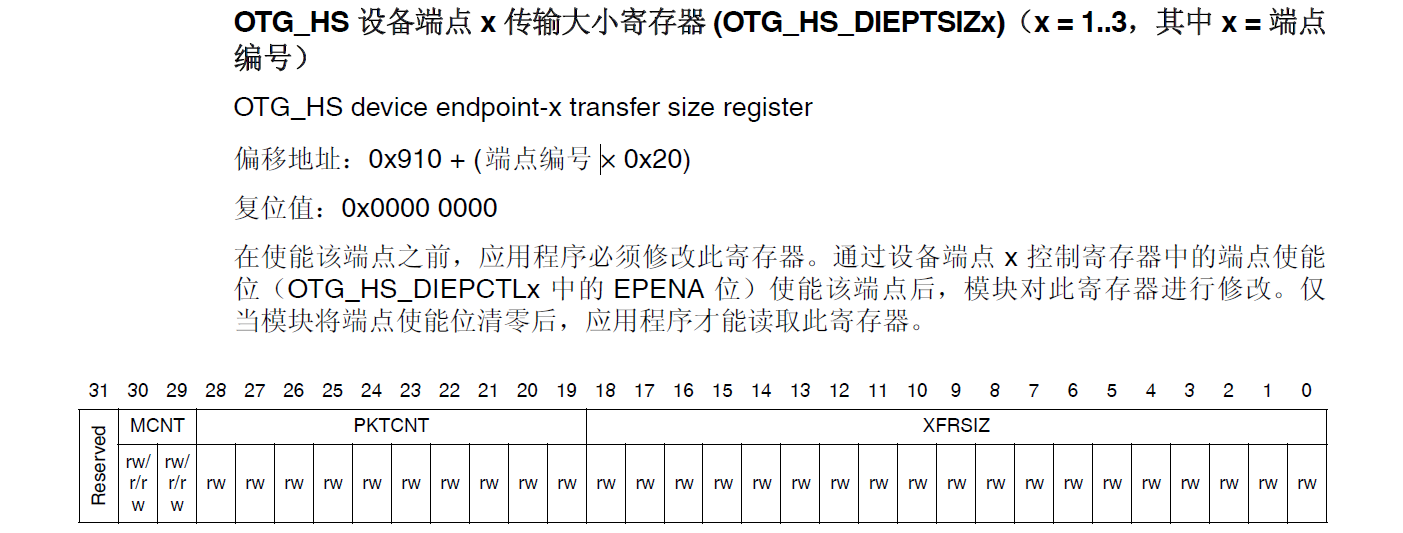
收不到数据

## USB HS send(transmist)

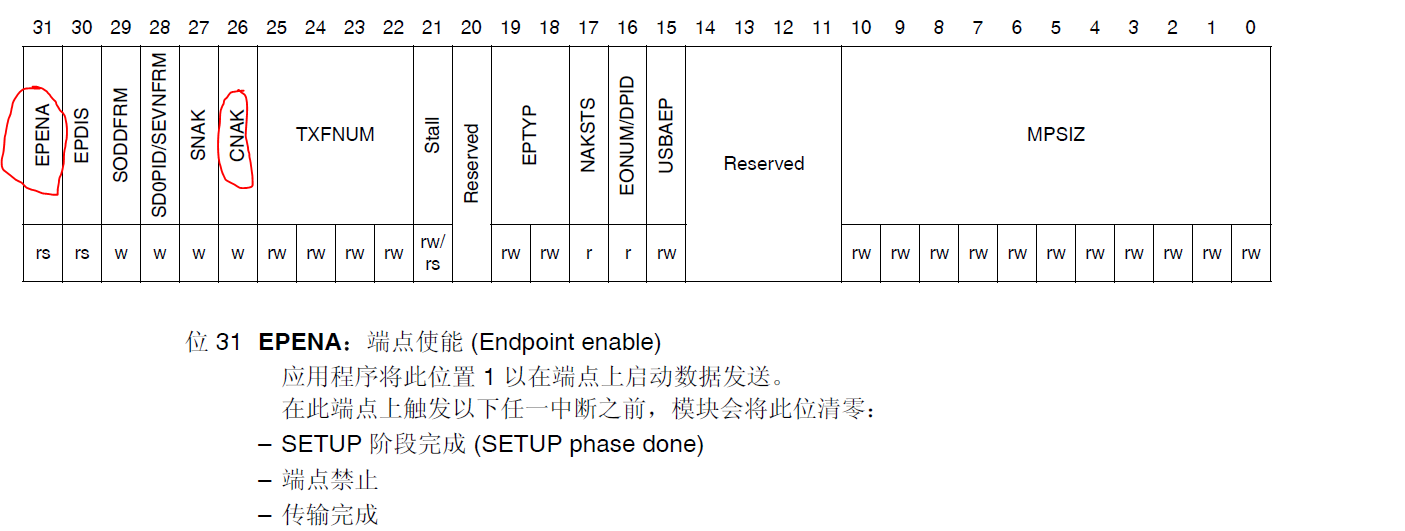
1. 函数调用关系：



其中为将buf和len映射到结构体hUsbDeviceHs中，一路调用到 USB\_EPStartXfer,其主要作用是设置端点传输大小和传输的包数（P1128）：

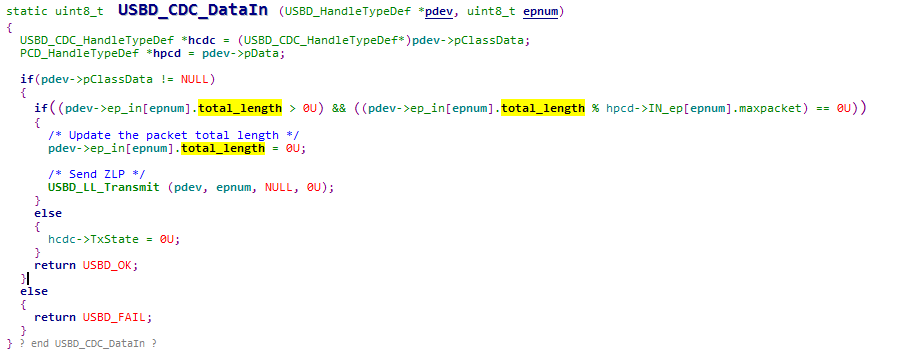


其中大小为XFRSIZ，为总的大小（USB HS为18bit，理论上一次最多可传输256K bytes数据），而PKTCNT为总的要发送的size相对于Ep size对齐的包数（LEN+EpSize-1）/EpSize。这样设置好后，再enable EP（P1118）：

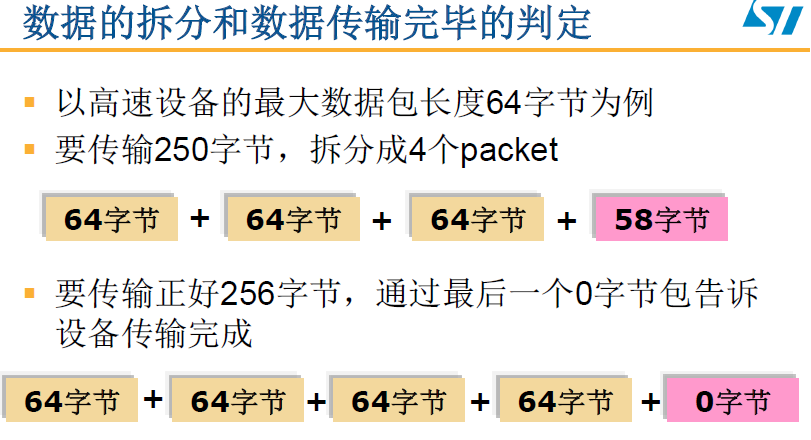


这样USB EP会以EP size为单位一包一包的将数据发送出去（纯硬件行为）。

发送完成后，调用发送完成的callback函数：HAL\_PCD\_IRQHandler，其中发送完成后USB\_OTG\_DIEPINT\_XFRC会被置位，然后调用HAL\_PCD\_DataInStageCallback（），而DataInStageCallback调用USBD\_LL\_DataInStage，在此函数中对非端点0的处理为：调用USBD\_CDC\_DataIn，而在此USBD\_CDC\_DataIn函数中判断如果发送的数据的size为非0且是EP size的整数倍，就会再发送一个size为0的IN包。而后等待此包发送完成后再次进入回调而进入此函数，此时USBD\_CDC\_DataIn将USBD TxState清0（即为准备状态），此时一次完整的IN事件完成。



Note：为什么传输是端点size的整数倍时，要在数据传输完成后再传一包0呢，这是USB协议规定的，为了告诉接收数据方传输完成了（P43:USB培训\_part1）：



所以即USB传输时的发送方都得这么做。

技巧：用USB协议分析仪查看数据时，USB协议分析仪上的host数据线要接到电脑的USB端口上，不要接到hub，否则会抓到大量的乱的数据。