在usbh\_core.c中有一个函数USBH\_Process（），这个函数很重要，main()函数每循环一次，它就执行一次，  
  
它处理所有USB内核状态的变化，  
typedef enum {  
HOST\_IDLE =0,  
HOST\_DEV\_ATTACHED,  
HOST\_DEV\_DISCONNECTED,   
HOST\_DETECT\_DEVICE\_SPEED,  
HOST\_ENUMERATION,  
HOST\_CLASS\_REQUEST,   
HOST\_CLASS,  
HOST\_CTRL\_XFER,  
HOST\_USR\_INPUT,  
HOST\_SUSPENDED,  
HOST\_ERROR\_STATE   
}HOST\_State;   
根据这个当前状态(phost->gState)处于以上哪个状态的不同，做出不同的处理过程。其中最重要的要属这两个状态：HOST\_ENUMERATION,，HOST\_CLASS。  
HOST\_ENUMERATION状态，不言而喻，这就是大名鼎鼎的枚举状态，经过空闲，连接状态后面就要对设备进行枚举了，枚举过程其实也是对状态机的处理过程，用  
  
的是这个USBH\_HandleEnum(pdev , phost)函数，也在在usbh\_core.c中，因为枚举过程也是USB过程的核心过程。枚举过程根据时间先后顺序依次经过   
typedef enum {  
ENUM\_IDLE = 0,  
ENUM\_GET\_FULL\_DEV\_DESC,  
ENUM\_SET\_ADDR,  
ENUM\_GET\_CFG\_DESC,  
ENUM\_GET\_FULL\_CFG\_DESC,  
ENUM\_GET\_MFC\_STRING\_DESC,  
ENUM\_GET\_PRODUCT\_STRING\_DESC,  
ENUM\_GET\_SERIALNUM\_STRING\_DESC,  
ENUM\_SET\_CONFIGURATION,  
ENUM\_DEV\_CONFIGURED  
} ENUM\_State;   
以上几个过程，如果都得到正确的处理，且返回值也为OK的话，枚举过程结束，此时已经获得了USB设备的基本信息。比如：设备描述符，配置描述符等等。下一  
  
步就该加载设备驱动了，根据枚举过程获得到的设备类型和ID号，比如大容量存储设备，HID,VIDIO等，选择不同的设备驱动加载，并初始化。这一步是在USBH\_Process（）处  
  
理函数在的HOST\_USR\_INPUT状态下执行的。正确执行后进入下一个状态HOST\_CLASS\_REQUEST，  
在这个HOST\_CLASS\_REQUEST状态，主要是初始化设备驱动类的状态机（host class state machine），为一步做准备。  
下面就到了另一个重要的处理函数，status = phost->class\_cb->Machine(pdev, phost);其实这个函数最终会调用USBH\_MSC\_Handle（），这个函数在  
  
usbh\_msc\_core.c中，usbh\_msc\_core.c顾名思义，这就是USB主机端mass storage类的驱动程序，核心文件，  
USBH\_MSC\_Handle（）函数根据当前所处的状态，作出相应的处理。  
typedef enum  
{  
USBH\_MSC\_BOT\_INIT\_STATE = 0,   
USBH\_MSC\_BOT\_RESET,   
USBH\_MSC\_GET\_MAX\_LUN,   
USBH\_MSC\_TEST\_UNIT\_READY,   
USBH\_MSC\_READ\_CAPACITY10,  
USBH\_MSC\_MODE\_SENSE6,  
USBH\_MSC\_REQUEST\_SENSE,   
USBH\_MSC\_BOT\_USB\_TRANSFERS,   
USBH\_MSC\_DEFAULT\_APPLI\_STATE,   
USBH\_MSC\_CTRL\_ERROR\_STATE,  
USBH\_MSC\_UNRECOVERED\_STATE  
}  
MSCState;  
USBH\_MSC\_DEFAULT\_APPLI\_STATE, 这个状态就是我们编写的应用程序将要执行的状态。最为不爽的是，连STM提供的上层应用程序也是按状态机的方式编写的。  
USBH\_USR\_MSC\_Application（），就是STM提供的应用程序，在usbh\_usr.c中，分为4个状态：  
#define USH\_USR\_FS\_INIT 0  
#define USH\_USR\_FS\_READLIST 1  
#define USH\_USR\_FS\_WRITEFILE 2  
#define USH\_USR\_FS\_DRAW 3  
如果我要编写我自己的应用程序的话，应该就是修改这个函数吧，