第1章 读写U盘模块驱动使用手册

1.1 读写U盘模块简介

读写U盘模块由硬件及软件两部分组成。硬件使用的是ISP1161BM PACK，软件使用HostMassLib.a驱动库。

ISP1161BM PACK是使用PHILIPS公司生产的USB主控芯片ISP1161BM，该芯片是一款兼容USB 2.0版规范的USB HC（Host Controller：主控器）。ISP1161A1有2个下行口，每个下行口都拥有独立的过流检测输入引脚及电源开关控制输出引脚。ISP1161A1同时也提供唤醒输入引脚及挂机状态输出引脚，使电源供电管理更加灵活。主控器的下行口可以与具有上行口的USB设备及USB集线器连接。

ISP1161A1适合使用于具有USB主机功能的嵌入式系统及便携式设备，它的出现使系统具有更高的灵活性。一个嵌入ISP1161A1的系统可以直接与具有USB上行口的设备连接，如：USB打印机，USB相机，USB键盘，及USB鼠标等。

HostMassLib是基于ZLG USB HOST STACK及ZLG Mass Storage Class驱动封装的ISP1161A1读写U盘驱动库。该库使用RAM资源少于4K，使用CODE资源少于12.4K，可同时支持两个U盘。

硬件功能特点：

- 符合通用串总线2.0规范
- 支持全速（12Mbit/s）及低速（1.5Mbit/s）两种数据模式
- 两个下行端口，可同时操作两个USB设备
- 工业级工作温度-40度到+85度

驱动功能特点：

- 支持μC/OS-II多任务操作系统
- 同时支持两个大容量设备（如：U盘、移动软驱）
- 仅需RAM：4K和CODE：12.4K
- 支持通用文件管理系统（如：ZLG/FS）

1.2 固件驱动说明

1.2.1 初始化配置

HostMassLib库最多使用12个μC/OS-II事件，所以需在OS_CFG.H文件中配置μC/OS-II事件OS_MAX_EVENTS值不能小于12。

创建一个USB主机服务任务，如程序清单 1.1所示。该任务使用的优先级应比其它使用USB主机传输任务的优先级高；主机服务任务推荐使用堆栈长度为256。

程序清单 1.1 USB 主机任务

```
/*********************************************************************************************
**                                    USB 主机任务
*********************************************************************************************/
void Task_USB_Host_Serve(void *pdata)
{
    pdata = pdata;
```

```
    while (1)
    {
        USB_Host_Serve();
    }
}
```

1.2.2 用户编写的接口函数

1．写命令寄存器函数

写命令寄存器函数如表 1.1所示。

表 1.1 写命令寄存器函数

函数名称	***outcommand***	所属文件	
函数原型	void outcommand (unsigned short data)		
功能描述	写 ISP116X 命令寄存器函数。		
编译开关	无		
函数参数	data，需写入的数据。		
函数返回值	无		
特殊说明 和注意要点	无		
范 例	void outcommand(unsigned short data) { * (volatile unsigned short *) HC_COMMAND_ADDRESS = data; }		

2．写数据寄存器函数

写数据寄存器函数如表 1.2所示。

表 1.2 写数据寄存器函数

函数名称	***outdata***	所属文件	
函数原型	void outdata(unsigned short data)		
功能描述	写 ISP116X 数据寄存器函数。		
编译开关	无		
函数参数	data，需写入的数据。		
函数返回值	无		
特殊说明 和注意要点	无		
范 例	void outdata(unsigned short data) { * (volatile unsigned short *) HC_DATA_ADDRESS = data; }		

3．读数据寄存器函数

读数据寄存器函数，如表 1.3所示。

表 1.3 读数据寄存器函数

函数名称	***indata***	所属文件	
函数原型	unsigned short indata(void)		
功能描述	读 ISP116X 数据寄存器函数。		
编译开关	无		
函数参数	无		
函数返回值	从数据寄存器读写的 16 位数位		
特殊说明 和注意要点	无		
范　　例	unsigned short indata(void) {　　return *((volatile unsigned short*) HC_DATA_ADDRESS); }		

4．禁止中断函数

禁止中断函数，如表 1.4所示。

表 1.4 禁止系统中断

函数名称	***disable***	所属文件	
函数原型	void disable(void)		
功能描述	禁止系统中断		
编译开关	无		
函数参数	无		
函数返回值	无		
特殊说明 和注意要点	无		
范　　例	void disable(void) {　　OS_ENTER_CRITICAL();　　// μC/OS-II 关系统中断处理函数 }		

5．使能中断函数

使能中断函数，如表 1.5所示。

表 1.5 使能中断函数

函数名称	***enable***	所属文件	
函数原型	void enable (void)		
功能描述	使能中断函数		
编译开关	无		
函数参数	无		

续上表

函数返回值	无
特殊说明 和注意要点	无
范　　例	void enable (void) {　　OS_EXIT_CRITICAL ();　　// μC/OS-II 关系统中断处理函数 }

6．调用中断服务函数

用户需编写一个中断函数，并在中断函数中调用USB 主机中断服务函数isr_USB_Hc，如表 1.6所示。

表 1.6 主机中断服务函数

函数名称	***isr_USB_Hc***	所属文件	
函数原型	void isr_USB_Hc(void);		
功能描述	主机中断服务函数		
编译开关	无		
函数参数	无		
函数返回值	无		
特殊说明 和注意要点	无		

范　　例	void ISP1160_Exception(void)
	{
	　　isr_USB_Hc();　　// 与 ISP116X 驱动接口的中断处理函数
	　　EXTINT = 0x04;　　// EINT2
	　　VICVectAddr = 0;
	}		

7．硬件复位函数

硬件复位函数，如表 1.7所示。

表 1.7 硬件复位函数

函数名称	***Sys_Hc_RESET***	所属文件	
函数原型	void Sys_Hc_RESET (void)		
功能描述	使能中断函数		
编译开关	无		
函数参数	无		
函数返回值	无		
特殊说明 和注意要点	该函数需在 ISP116X 初始化之前被调用一次。如果使用 ISP116X 上电复位则可以不使用该函数。		

续上表

范　　例	void Sys_Hc_RESET(void)
	{
	IO1CLR = HcRESET; // 置低与 ISP116X 复位引脚连接的 IO 口
	Sys_WaitinMS(2);
	IO1SET = HcRESET; // 置高与 ISP116X 复位引脚连接的 IO 口
	Sys_WaitinMS(2);
	}

1.2.3 API 函数

表 1.8、表 1.9、表 1.10、表 1.11、表 1.12、表 1.13、表 1.14和表 1.15分别描述了大容量类驱动提供的API函数。使用以这些API函数前需包含“HostStack.h”头文件。

表 1.8 USB 主机初始化函数

函数名称	***USB_Host_Stack_Initialize***	所属文件	
函数原型	void USB_Host_Stack_Initialize(void);		
功能描述	USB 主机驱动初始化。		
编译开关	无		
函数参数	无		
函数返回值	无		
特殊说明 和注意要点			
范　　例			

表 1.9 查找大容量设备

函数名称	***Creat_Medium***	所属文件	
函数原型	device_instance *find_mass_device(unsigned char Index)		
功能描述	查找大容量类设备，。		
编译开关	无		
函数参数	Index，逻辑单元索引，0 为第 1 个逻辑单元。		
函数返回值	dvi_ptr，大容量设备的设备描述信息结构指针； 设备不存在，返回一个空指针。		
特殊说明 和注意要点			
范　　例			

表 1.10 Creat_Medium()函数

函数名称	***Creat_Medium***	所属文件	
函数原型	hMedLUN * Creat_Medium(device_instance *dvi_ptr ,unsigned char LUNIndex);		
功能描述	创建（获取）大容量设备逻辑单元描述符指针。		

续上表

编译开关	无
函数参数	dvi_ptr，大容量设备的设备描述信息结构指针； LUNIndex，逻辑单元索引，0 为第 1 个逻辑单元。
函数返回值	逻辑单元存在，返回逻辑单元描述符指针； 逻辑单元不存在，返回一个容指针。
特殊说明 和注意要点	
范　　例	

表 1.11 关闭大容量设备逻辑单元

函数名称	***Close_Medium***	所属文件	
函数原型	void Close_Medium(hMedLUN *hMedLUNPtr,unsigned char LUNIndex)		
功能描述	关闭大容量设备逻辑单元		
编译开关	无		
函数参数	hMedLUNPtr，大容量设备的逻辑单元描述符指针； LUNIndex，逻辑单元索引，0 为第 1 个逻辑单元。		
函数返回值	无		
特殊说明 和注意要点	如果再次打开一个大容量逻辑单元前，必须先使用该函数关闭该逻辑单元。否则将创建失败。		
范　　例			

表 1.12 获取设备总扇区数

函数名称	***GetLastSectorAddr***	所属文件	
函数原型	unsigned int GetLastSectorAddr(hMedLUN *hMedLUNPtr)		
功能描述	获取逻辑单元最后一个扇区地址。		
编译开关	无		
函数参数	hMedLUNPtr，大容量设备的逻辑单元描述符指针		
函数返回值	最后一个扇区地址		
特殊说明 和注意要点	使用该函数前需成功创建一个逻辑单元，关闭逻辑单元其返回值为 0。		
范　　例			

表 1.13 获取扇区大小

函数名称	***GetSectorSize(hMedLUN *hMedLUNPtr)***	所属文件	
函数原型	unsigned short GetSectorSize***(***hMedLUN *hMedLUNPtr***)***		
功能描述	获取大容量存储类设备逻辑单元扇区大小，单位为字节		
编译开关	无		
函数参数	hMedLUNPtr，大容量设备的逻辑单元描述符指针		
函数返回值	扇区大小字节数。		

续上表

特殊说明 和注意要点	使用该函数前需成功创建一个逻辑单元，关闭逻辑单元其返回值为 0。
范　　例	

表 1.14 ReadBlockData()函数

函数名称	***ReadBlockData***	所属文件	
函数原型	unsigned short ReadBlockData(hMedLUN *MediumPtr, unsigned char *BufferPtr, unsigned int LBA, unsigned short TrBLength);		
功能描述	从逻辑单元设备指定的逻辑块地址读出数据。		
编译开关	无		
函数参数	MediumPtr，逻辑单元描述符指针； BufferPtr，数据缓冲区指针； LBA，逻辑块地址； TrBLength，读取逻辑块个数，值范围为：1~127。		
函数返回值	实际读入的字节数，设备不存在或未准备好返回 NULL。		
特殊说明 和注意要点	TrBLength 不能大于 127。		
范　　例			

表 1.15 WriteBlockData()函数

函数名称	***WriteBlockData***	所属文件	
函数原型	unsigned short WriteBlockData(hMedLUN *MediumPtr, unsigned char *BufferPtr, unsigned int LBA, unsigned short TrBLength);		
功能描述	往逻辑单元设备指定的逻辑块地址写入数据。		
编译开关	无		
函数参数	MediumPtr，逻辑单元描述符指针； BufferPtr，数据缓冲区指针； LBA，逻辑块地址； TrBLength，写逻辑块个数，值范围为：1~127。		
函数返回值	实际读入的字节数，设备不存在或未准备好返回 NULL。		
特殊说明 和注意要点	TrBLength 不能大于 127。		
范　　例			