
库函数调用说明

库版本： M3F20xm_DLL_V01

发行日期： 10/08/2017

目录

库函数调用说明	1
目录	2
结构体	3
➤ SPI_CONFIG:	3
➤ TRIG_CONFIG	4
调用顺序.....	5
函数说明.....	6
➤ M3F20XM_SetUSBNOTIFY	6
➤ M3F20XM_GetSERIALNO	6
➤ M3F20XM_OpenDEVICE	6
➤ M3F20XM_OpenDEVICEByNUMBER	6
➤ M3F20XM_GetVERSION	7
➤ M3F20XM_CloseDEVICE	7
➤ M3F20XM_CloseDEVICEByNUMBER	7
➤ M3F20XM_SPIGetCONFIG	7
➤ M3F20XM_SPISetCONFIG	8
➤ M3F20XM_SPIRead	8
➤ M3F20XM_SPIWrite	8
➤ M3F20XM_TRIGGetCONFIG	9
➤ M3F20XM_TRIGSetCONFIG	9
➤ M3F20XM_EnableTRIG	9
➤ M3F20XM_Verify	10
➤ M3F20XM_BuffRead	10
➤ M3F20XM_BuffWrite	10
➤ M3F20XM_InitFIFO	11
➤ M3F20XM_ReadFIFO	11
➤ M3F20XM_GetFIFOLeft	11
➤ M3F20XM_GPIOSetCONFIG	12
➤ M3F20XM_GPIOGetCONFIG	12
➤ M3F20XM_GPIOREad	12
➤ M3F20XM_GPIOWrite	13

结构体

➤ SPI_CONFIG:

```
typedef struct
{
    WORD wSPIConfig;          //SPI config
    WORD wDelayCSLow;         //the delay time after CS becomes low
    WORD wDelayComEnd;        //the delay time after command sends
    WORD wDelayByteEnd;       //the delay time after a byte clks
    WORD wDelayDataEnd;       //the delay time after all bytes clks
    WORD wReserved;           //unused
} SPI_CONFIG;
```

结构名称: SPI_CONFIG			
序号	数据类型	名称	说明
1	WORD	wSPIConfig	Bit15: SPI 读写模式 1-Direct, 0- Buffered Bit12: SPI 帧长度 1-16bit, 0-8bit Bit7: 发送顺序 1-LSB, 0-MSB Bit5~3:分频系数 SPI 频率 = 60M/(分频系数+1) Bit2: 模式 1-master, 0-slaver Bit1: 极性 CLK 空闲时电平 1-High, 0-Low Bit0: 相位 CLK 采样沿 1-第二个, 0-第一个
2	WORD	wDelayCSLow	CS 变低后延时时间, 单位微秒
3	WORD	wDelayComEnd	Command 发送后延时时间, 单位微秒
4	WORD	wDelayFrameEnd	SPI 每帧之间的延时时间,单位微秒
5	WORD	wDelayDataEnd	读写数据传输完后之间的延时时间, 单位微秒
6	WORD	wReserved	未用的
备注:		参数 2~5 只有在调用 Direct 模式中有作用	

➤ TRIG_CONFIG

```
typedef struct
{
    BYTE  byTrigCon;
    BYTE  byIOTrigOptions;
    BYTE  byUnused;
    BYTE  byActions;
    DWORD dwPeriod;
    BYTE  byRCmdSize;
    BYTE  byWCmdSize;
    BYTE  byReadCmd[16];
    BYTE  byWriteCmd[16];
    WORD  wReadSize;
    WORD  wTrigSize;
    DWORD dwTrigCnt;
    DWORD dwMaxCnt;
} TRIG_CONFIG;
```

结构名称:		TRIG_CONFIG	
序号	数据类型	名称	说明
1	BYTE	byTrigCon	触发状态和选择 Bit7: 1-触发打开, 0-触发关闭 Bit0: 1-定时触发, 0-IO 触发
2	BYTE	byIOTrigOptions	IO 触发类型: 0: 上升沿触发 1: 下降沿触发 2: 上升和下降沿触发
3	BYTE	byUnused	未用的
4	BYTE	byActions	触发后的动作 0: 执行读 1: 先写后读 2: 先读后写
5	DWORD	dwPeriod	定时触发周期, 单位微秒
6	BYTE	byRCmdSize	读操作命令参数长度
7	BYTE	byWCmdSize	写操作命令参数长度
8	BYTE	byReadCmd[16]	读操作命令参数
9	BYTE	byWriteCmd[16]	写操作命令参数
10	wReadSize	wReadSize	读操作数据长度
11	wTrigSize	wTrigSize	触发缓存数据长度, 必须小于 3072
12	DWORD	dwTrigCnt	触发次数计数器, 只读
13	DWORD	dwMaxCnt	最大触发次数, 达到此次数后触发自动关闭, 如果为 0, 则不限次数
备注:			

调用顺序

装载库文件后，不管是静态装载还是动态装载，请按下面步骤调用库函数

调用步骤	第一种方案(适合一个进程打开一个 M3F20xm 设备)	第二种方案适合一个进程打开多个 M3F20xm 设备)
1	调用 M3F20xm_SetUSBNotify,用于监控 USB 插拔事件	调用 M3F20xm_SetUSBNotify,用于监控 USB 插拔事件
2	调用 M3F20xm_OpenDevice 取得设备号	调用 M3F20xm_GetMaxNumofDev 函数取得最大可以同时打开的设备数
3	根据设备号，调用 M3F20xm_Verify 取得授权	根据设备序列号调用 M3F20xm_OpenDeviceByNumber 取得设备号
4	根据设备号，调用其他库读写函数	根据设备号，调用 M3F20xm_Verify 取得授权
5	调用 M3F20xm_CloseDevice 关闭设备	根据设备号，调用其他读写函数
6		调用 M3F20xm_CloseDevice 关闭设备

函数说明

➤ M3F20xm_SetUSBNotify

函数原型	bool M3F20xm_SetUSBNotify(bool bLog, USB_DLL_CALLBACK pUSB_Callback)	
功能说明	设置一个回调函数给库函数，库函数检测到 USB 插拔后调用此函数	
参数说明	bLog	Log 文件记录，不记录如果 false
	pUSB_Callback	函数指针，必须是 bool function(BYTE iDevIndex, DWORD iDevStatus)类型的函数
		iDevIndex: 设备号
		iDevStatus: 设备状态，0x80,设备插上;0,设备拔出
返回值	true	

➤ M3F20xm_GetSerialNo

函数原型	BYTE M3F20xm_GetSerialNo(BYTE byIndex, char* lpBuff)	
功能说明	根据设备号取得设备序列号	
参数说明	byIndex	适配器设备号
	lpBuff	保存设备序列号的缓存，长度必须大于 10 个 BYTE
返回值	BYTE	如果是 0，表示该设备不存在，1 未使用，2 使用中
备注	每个设备都有一个唯一的设备序列号	

➤ M3F20xm_OpenDevice

函数原型	BYTE M3F20xm_OpenDevice(void)	
功能说明	查找一个可用的设备，并打开该设备，返回设备号	
参数说明	无	
返回值	BYTE	如果是 0xFF，表示打开设备不成功

➤ M3F20xm_OpenDeviceByNumber

函数原型	BYTE M3F20xm_OpenDeviceByNumber(char* pSerialString)	
功能说明	根据指定的序列号打开设备，返回设备号	
参数说明	pSerialString	保存设备序列号的缓存
返回值	BYTE	如果是 0，表示该设备不存在，1 未使用，2 使用中
备注	每个设备都有一个唯一的设备序列号	

➤ M3F20xm_GetVersion

函数原型	bool M3F20xm_GetVersion(BYTE byIndex,BYTE byType,char* lpBuffer)	
功能说明	取得软件的版本信息	
参数说明	byIndex	设备号
	byType	软件类型; 0-库版本信息 1-驱动版本信息 2-固件版本信息
	lpBuffer	保存版本信息的缓存, 至少大于 50 个 BYTE
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_CloseDevice

函数原型	bool M3F20xm_CloseDevice(BYTE byIndex)	
功能说明	关闭指定的设备	
参数说明	byIndex	设备号
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_CloseDeviceByNumber

函数原型	BYTE M3F20xm_CloseDeviceByNumber (char* pSerialString)	
功能说明	根据指定的序列号关闭设备, 返回设备号	
参数说明	pSerialString	保存设备序列号的缓存
返回值	bool	成功返回 true,否则返回 false
备注	每个设备都有一个唯一的设备序列号	

➤ M3F20xm_SPIGetConfig

函数原型	bool M3F20xm_SPISetConfig (BYTE byIndex,SPI_CONFIG* psConfig) /*++)	
功能说明	配置指定设备 SPI 接口	
参数说明	byIndex	设备号
	psConfig	见 SPI_CONFIG 定义
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_SPISetConfig

函数原型	bool M3F20xm_SPISetConfig (BYTE byIndex,SPI_CONFIG* psConfig)	
功能说明	配置指定设备 SPI 接口	
参数说明	byIndex	设备号
	psConfig	见 SPI_CONFIG 定义
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_SPIRead

函数原型	bool M3F20xm_SPIRead(BYTE byIndex,BYTE* lpComBuffer,BYTE byComSize,BYTE* lpReadBuffer,DWORD dwBuffSize,DWORD dwTimeout)	
功能说明	通过指定的设备的 SPI 接口读数据	
参数说明	byIndex	设备号
	lpComBuffer	保存开始读数据前写入的寄存器地址或命令的缓存, 如果没有可以设置 byComSize 的值为 0
	byComSize	lpComBuffer 的长度,不能超过 56
	lpReadBuffer	接收读入的数据的缓存,
	dwBuffSize	要读的数据长度, Buffer 方式时 dwBuffSize 最大值 2M, Direct 方式时, 最大值 48K
	dwTimeout	读超时时间,单位: 毫秒
返回值	bool	成功返回 true,否则返回 false
备注	Buffer 方式时,每 8192 个 byte 间有延时; Direct 模式时, 延时时间由 SPI config 决定	

➤ M3F20xm_SPIWrite

函数原型	bool M3F20xm_SPIWrite(BYTE byIndex, BYTE* lpComBuffer, BYTE byComSize, BYTE* lpWriteBuffer,DWORD dwWriteSize,DWORD dwTimeout)	
功能说明	通过指定的设备的 SPI 接口写数据	
参数说明	byIndex	设备号
	lpComBuffer	保存开始写数据前写入的寄存器地址或命令的缓存, 如果没有可以设置 byComSize 的值为 0
	byComSize	lpComBuffer 的长度
	lpWriteBuffer	写入的数据的缓存,不能超过 56
	dwWriteSize	要写的数据长度, Buffer 方式时 dwBuffSize 最大值 2M, Direct 方式时, 最大值 48K
	dwTimeout	写超时时间,单位: 毫秒
返回值	bool	成功返回 true,否则返回 false

备注	Buffer 模式时,每 8192 个 byte 间有延时; Direct 模式时, 延时时间由 SPI config 决定
----	--

M3F20xm_SPITransfer

函数原型	bool M3F20xm_SPITransfer (BYTE byIndex,BYTE* lpWriteBuffer,BYTE* lpReadBuffer,WORD wBuffSize,DWORD dwTimeout)	
功能说明	通过指定的设备的 SPI 接口,写并且读数据 (双工模式)	
参数说明	byIndex	设备号
	lpWriteBuffer	写入数据的缓存
	lpReadBuffer	读入的数据的缓存
	wReadSize	读写的数据长度,不能超过 48K
	dwTimeout	读写超时时间,单位: 毫秒
返回值	bool	成功返回 true,否则返回 false
备注	Buffer 模式时,每 8192 个 byte 间无延时; Direct 模式时, 延时时间由 SPI config 决定	

➤ M3F20xm_TrigGetConfig

函数原型	bool M3F20xm_TrigGetConfig(BYTE byIndex,TRIG_CONFIG* pCfg)	
功能说明	取得触发配置	
参数说明	byIndex	设备号
	pCfg	见 TRIG_CONFIG 定义
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_TrigSetConfig

函数原型	bool M3F20xm_TrigSetConfig(BYTE byIndex,TRIG_CONFIG* pCfg)	
功能说明	配置触发	
参数说明	byIndex	设备号
	pCfg	见 TRIG_CONFIG 定义
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_EnableTrig

函数原型	bool M3F20xm_EnableTrig (BYTE byIndex, BYTE byOn)
------	---

功能说明	启动或停止触发功能	
参数说明	byIndex	设备号
	byOn	1-Enable trig, 0- Disable trig
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_Verify

函数原型	M3F20xm_Verify(BYTE byIndex,BYTE* pResult)	
功能说明	设备授权认证	
参数说明	byIndex	设备号
	pResult	认证结果, 1 验证通过, 0 验证不通过
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_BuffRead

函数原型	M3F20xm_BuffRead(BYTE byIndex,WORD wAddr,BYTE* lpBuffer,WORD wBuffSize,DWORD dwTimeout)	
功能说明	从指定地址读取 SPI 设备内部缓存的内容	
参数说明	byIndex	设备号
	wAddr	Buff 的偏移地址,必须小于 0xC00
	lpBuffer	接收读入的数据的缓存
	wBuffSize	读取的数据长度,wAddr+wBuffSize 必须小于 0xC00
	dwTimeout	读取超时,单位毫秒
返回值	bool	成功返回 true,否则返回 false
备注	SPI 内部有一个 0xC00(3K)的 buff 可供读写	

➤ M3F20xm_BuffWrite

函数原型	bool M3F20xm_BuffWrite(BYTE byIndex,WORD wAddr,BYTE* lpBuffer,WORD wBuffSize,DWORD dwTimeout)	
功能说明	从指定地址写入内容到 SPI 设备内部缓存	
参数说明	byIndex	设备号
	wAddr	Buff 的偏移地址,必须小于 0xC00
	lpBuffer	写入内容的缓存
	wBuffSize	写入的数据长度,wAddr+wBuffSize 必须小于 0xC00
	dwTimeout	写入超时,单位毫秒
返回值	bool	成功返回 true,否则返回 false
备注	SPI 内部有一个 0xC000(3K)的 buff 可供读写	

➤ M3F20xm_InitFIFO

函数原型	bool bool M3F20xm_InitFIFO(BYTE byIndex)	
功能说明	初始化 FIFO,清空 FIFO 的所有内容	
参数说明	byIndex	设备号
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_ReadFIFO

函数原型	bool M3F20xm_ReadFIFO(BYTE byIndex,BYTE* lpBuffer,DWORD dwBuffSize,DWORD* pdwRealSize)	
功能说明	获取指定长度的 FIFO 未读数据	
参数说明	byIndex	设备号
	lpBuffer	用来保存读取内容的缓存
	dwBuffSize	请求读取的数据长度
	pdwRealSize	实际读入的数据长度的指针
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_GetFIFOLeft

函数原型	bool M3F20xm_GetFIFOLeft(BYTE byIndex,DWORD* pdwBuffsize)	
功能说明	取得 FIFO 未读数据的长度	
参数说明	byIndex	设备号
	pdwBuffsize	FIFO 未读的数据长度的指针
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_GPIOSetConfig

函数原型	bool M3F20xm_GPIOSetConfig(BYTE byIndex,DWORD dwValue)	
功能说明	配置 GPIO	
参数说明	byIndex	设备号
	dwValue	GPIO 配置值;bit15~12: IO4 bit11~8:IO3 bit7~4:IO2 bit3~0:IO1 0-IN_NONE(输入浮空, 相当于高阻抗); 1-IN_PU(输入带上拉) 2-IN_PD(输入带下拉) 3-PP_NONE(推挽输出不带上拉下拉) 4-PP_PU(推挽输出带上拉) 5-IN_PD(推挽输出带下拉) 6-OD_NONE(开漏输出不带上拉下拉) 7-OD_PU(开漏输出带上拉) 8-OD_PD(开漏输出带下拉)
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_GPIOGetConfig

函数原型	bool M3F20xm_GPIOGetConfig(BYTE byIndex,DWORD* pdwValue)	
功能说明	获取 GPIO 配置	
参数说明	byIndex	设备号
	pdwValue	保存 GPIO 配置值的指针, 值含义见 M3F20xm_GPIOSetConfig
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_GPIORead

函数原型	bool M3F20xm_GPIORead(BYTE byIndex,BYTE* pbyValue)	
功能说明	读 GPIO 端口, 1 为高电平,0 为低电平	
参数说明	byIndex	设备号
	pbyValue	用来保存 GPIO 值的指针, bit0~bit3 分别对应 IO1,IO2,IO3,IO4
返回值	bool	成功返回 true,否则返回 false
备注		

➤ M3F20xm_GPIOWrite

函数原型	bool M3F20xm_GPIOWrite(BYTE byIndex,BYTE byValue,BYTE byMask)	
功能说明	写 GPIO 端口，1 为高电平,0 为低电平	
参数说明	byIndex	设备号
	byValue	GPIO 端口值，bit0~bit3 分别对应 IO1,IO2,IO3,IO4
	byMask	屏蔽值，如果该 bit 为 1，这该 bit 对应的 GPIO 不做改变
返回值	bool	成功返回 true,否则返回 false
备注		