

U 盘和 SD 卡高速文件管理控制芯片 CH378

手册

版本: 4B

<http://wch.cn>

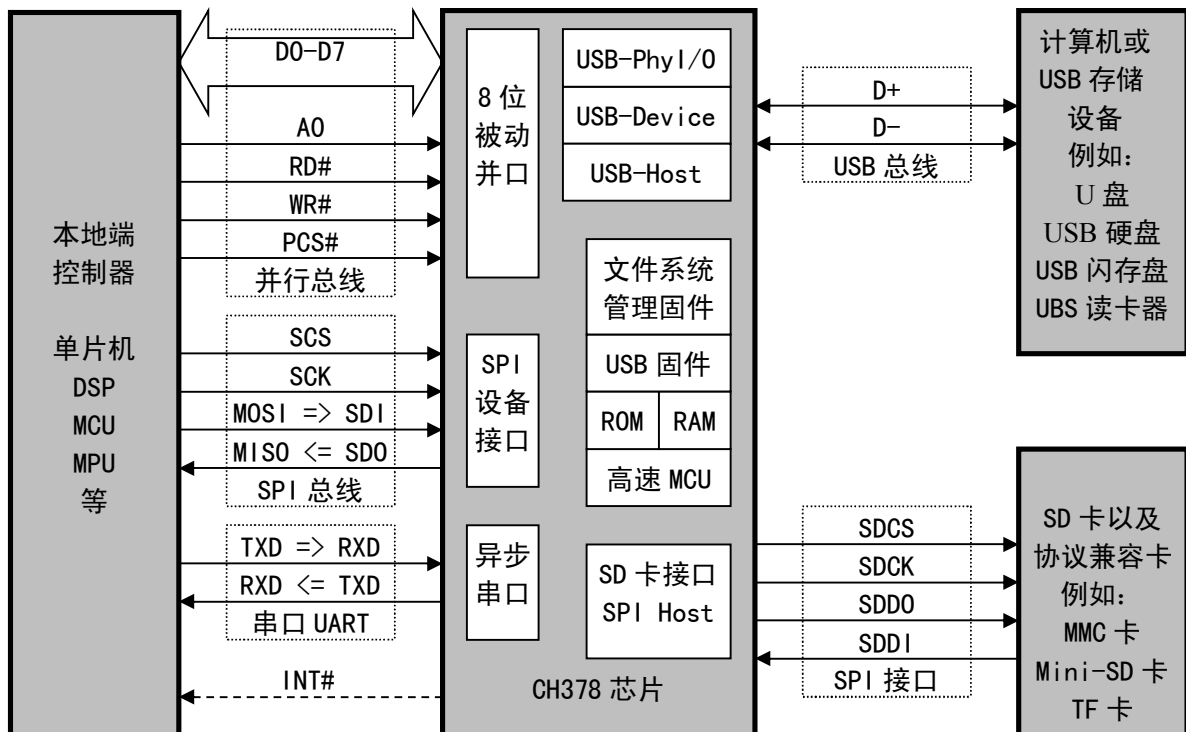
1、概述

CH378 是文件管理控制芯片, 用于单片机系统快速读写 U 盘或者 SD 卡中的文件。

CH378 支持 USB 设备方式和 USB 主机方式, 并且内置了 USB 通讯协议的基本固件, 内置了处理 Mass-Storage 海量存储设备的专用通讯协议的固件, 内置了 SD 卡的通讯接口固件, 内置了 FAT16 和 FAT32 以及 FAT12 文件系统的管理固件, 支持常用的 USB 存储设备 (包括 U 盘/USB 硬盘/USB 闪存盘/USB 读卡器) 和 SD 卡 (包括标准容量 SD 卡和高容量 HC-SD 卡以及协议兼容的 MMC 卡和 TF 卡)。

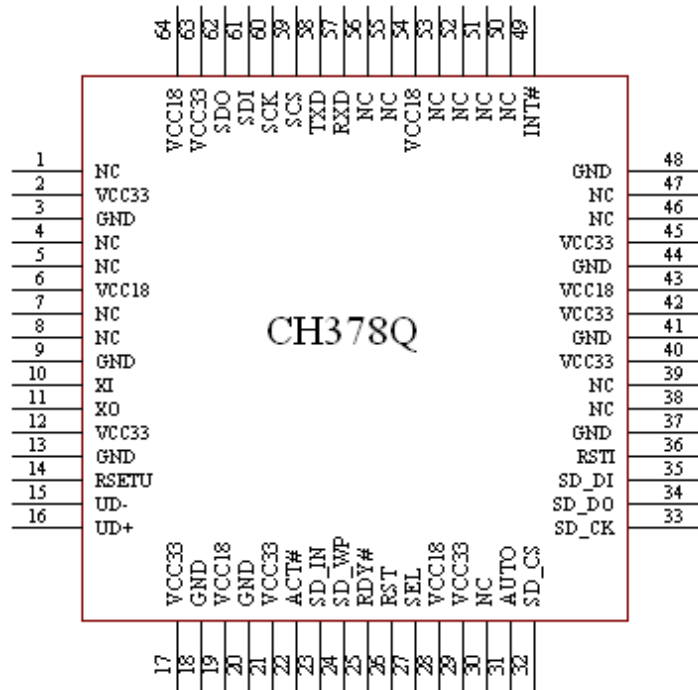
CH378 支持三种通讯接口: 8 位并口、SPI 接口或者异步串口, 单片机/DSP/MCU/MPU 等控制器可以通过上述任何一种通讯接口控制 CH378 芯片, 存取 U 盘或者 SD 卡中的文件或者与计算机通讯。

下图为 CH378 的应用框图。



2、特点

- 支持 12Mbps 全速和 480Mbps 高速 USB 通讯, 外围元器件只需要晶体和电容。
- 支持 USB-HOST 主机接口和 USB-DEVICE 设备接口, 支持动态切换主机方式和设备方式。
- 支持 USB 设备的控制传输、批量传输和中断传输。
- 自动检测 USB 设备或 SD 卡的连接和断开, 提供设备连接和断开的事件通知。
- 提供最高 50MHz 的 SPI 主机接口, 支持 SD 卡以及与其协议兼容的 MMC 卡和 TF 卡等。
- 内置 USB 控制传输的协议处理器, 简化常用的控制传输。
- 内置固件处理海量存储设备的专用通讯协议, 支持 Bulk-Only 传输协议和 SCSI、UFI、RBC 或等效命令集的 USB 存储设备 (包括 U 盘/USB 硬盘/USB 闪存盘/USB 读卡器)。
- 内置 FAT16 和 FAT32 以及 FAT12 文件系统的管理固件, 支持容量高达 32GB 的 U 盘和 SD 卡。
- 提供文件管理功能: 打开、新建或删除文件、枚举和搜索文件、创建子目录、支持长文件名。
- 提供文件读写功能: 以字节为最小单位或者以扇区为单位对多级子目录下的文件进行读写。
- 提供磁盘管理功能: 初始化磁盘、查询物理容量、查询剩余空间、物理扇区读写。



芯片型号	芯片封装	
	名称	描述
CH378L	LQFP128	LQFP 封装; 128 脚; 本体 14x14mm
CH378Q	LQFP64M	LQFP 封装; 64 脚; 本体 10x10mm

4、引脚

CH378L 引脚号	CH378Q 引脚号	引脚 名称	类型	引脚说明
7、21、26、39、 56、70、72、83、 92、101、122	2、12、17、 21、29、40、 42、45、63	VCC33	电源	3.3V 正电源输入端, 外接 0.1uF 电源退耦电容
1、5、12、13、 18、28、30、54、 82、108	6、19、28、 43、54、64	VCC18	电源	1.8V 正电源输入端, 外接 0.1uF 电源退耦电容
2、4、9、16、17、 22、27、29、55、 64、65、71、109	3、9、13、18、 20、37、41、 44、48	GND	电源	公共接地端
3、6、8、10、11、 14、15、31、32、 33、34、35、36、 37、38、40、41、 42、43、44、45、 46、47、57、66、 67、68、69、73、 74、75、76、77、 78、79、80、81、 84、85、86、87、 88、89、90、91、	1、4、5、7、 8、30、38、 39、46、47、 50、51、52、 53、55、56	NC	-	保留引脚, 悬空

93、94、95、96、97、102、103、104、105、110、111、127、128				
23	14	RSETU	输入	外接 12K 电阻到地端
19	10	XI	输入	晶体振荡的输入端, 需要外接 30MHz 晶体
20	11	XO	输出	晶体振荡的反相输出端, 需要外接 30MHz 晶体
24	15	UD-	USB 信号	USB 总线的 D-数据线
25	16	UD+	USB 信号	USB 总线的 D+数据线
48	22	ACT#	输出	状态输出, 低电平有效 在 USB 主机方式下是 USB 设备正在连接状态输出; 在 SD 卡主机方式下是 SD 卡正在连接状态输出; 在内置固件的 USB 设备方式下是 USB 设备配置完成状态输出;
49	23	SD_IN	输入	SD 卡是否插入检测引脚, 内置上拉电阻
50	24	SD_WP	输入	SD 卡是否写保护检测引脚, 内置上拉电阻
51	25	RDY#	输出	状态输出, 低电平有效 在 USB 主机方式下, 是 USB 设备初始化成功可以正常操作状态输出; 在 SD 卡主机方式下, 是 SD 卡初始化成功可以正常操作状态输出; 在自动演示方式下, 是演示状态输出;
52	26	RST	输出	电源上电复位和外部复位输出, 高电平有效
53	27	SEL	输入	在芯片复位期间为接口配置输入, 内置上拉电阻
58	31	AUTO	输入	芯片自动演示配置引脚, 内置上拉电阻 上电正常工作后, 芯片检测到该引脚为低电平则开始进行自动演示
59	32	SD_CS	输出	SD 卡 SPI 接口的片选输出, 低电平有效
60	33	SD_CK	输出	SD 卡 SPI 接口的串行时钟输出
61	34	SD_DO	输出	SD 卡 SPI 接口的串行数据输出
62	35	SD_DI	输入	SD 卡 SPI 接口的串行数据输入, 内置上拉电阻
63	36	RST I	输入	外部复位输入, 低电平有效, 内置上拉电阻
98	49	INT#	输出	中断请求输出, 低电平有效
99	-	A0	输入	并口的地址输入, 区分命令口与数据口, 内置上拉电阻, 当 A0=1 时可以写命令或读状态, 当 A0=0 时可以读写数据
100	-	PCS#	输入	并口的片选控制输入, 低电平有效, 内置上拉电阻
106	-	RD#	输入	并口的读选通输入, 低电平有效, 内置上拉电阻
107	-	WR#	输入	并口的写选通输入, 低电平有效, 内置上拉电阻
112	57	RXD	输入	异步串口的串行数据输入, 内置上拉电阻
113	58	TXD	输入 输出	在芯片复位期间为接口配置输入, 内置上拉电阻, 在芯片复位完成后为异步串口的串行数据输出
114	59	SCS	输入	SPI 接口的片选输入, 低电平有效, 内置上拉电阻
115	60	SCK	输入	SPI 接口的串行时钟输入, 内置上拉电阻
116	61	SDI	输入	SPI 接口的串行数据输入, 内置上拉电阻
117	62	SDO	三态输出	SPI 接口的串行数据输出
118~121	-	D0~D7	双向三态	并口的 8 位双向数据总线, 内置上拉电阻

123~126				
---------	--	--	--	--

（注：本表中标注为灰色的引脚，只能承受 3.3V 输入电压，未标注的引脚可承受 3.3V 和 5V 输入。）

5、命令

本手册中的数据，后缀 B 为二进制数，后缀 H 为十六进制数，否则为十进制数。

低字节在前（Little-Endian）的双字数据（共 32 位）指：先是最低字节（位 7~位 0），接着是较低字节（位 15~位 8），然后是较高字节（位 23~位 16），最后是最高字节（位 31~位 24）。

数据流是指由若干连续字节组成的数据块，数据块的总长度最小为 0，最大为 20480。

下表输入数据和输出数据的括号中的数字为参数的字节数，没有括号则默认为一个字节。

本手册中所指的单片机基本适用于 DSP 或者 MCU/MPU/SCM 等。

本手册中所指的 U 盘包括 U 盘、USB 外置硬盘、USB 闪存盘、USB 读卡器等。

本手册中所指的 SD 卡包括 SD 卡、MMC 卡、HC-SD 卡（大容量 SD 卡）、TF 卡等。

本手册主要提供了常用的文件管理控制类命令，用于 U 盘和 SD 卡。关于一些不太常用的辅助命令和执行 USB 基本事务及控制传输的命令请参考手册（二）CH378DS2.PDF。

CH378 芯片包含的主要常用命令码如下表所示。

代码	命令名称 CMD_	输入数据	输出数据	命令用途
01H	GET_IC_VER		版本号	获取芯片及固件版本
02H	SET_BAUDRATE	波特率系数(3)	(等 1mS) 操作状态	设置串口通讯波特率
03H	ENTER_SLEEP	睡眠模式(1)		进入低功耗睡眠挂起状态
05H	RESET_ALL		(等 50mS)	执行硬件复位
06H	CHECK_EXIST	任意数据	按位取反	测试通讯接口和工作状态
0BH	SET_SDO_INT	数据 16H 中断方式		设置 SPI 的 SDO 引脚的中断方式
0CH	GET_FILE_SIZE	数据 68H	文件长度(4)	获取当前文件长度
0DH	SET_FILE_SIZE	变量地址 0x68 文件长度(4)		临时设置当前文件长度
15H	SET_USB_MODE	模式代码	(等 35mS) 操作状态	设置 USB 工作模式
22H	GET_STATUS		中断状态	获取中断状态并取消中断请求
26H	RD_HOST_OFS_DATA	偏移地址(2) 读取长度(2)	数据流(N)	读取内部缓冲区指定偏移地址数据块
27H	RD_HOST_CUR_DATA	读取长度(2)	数据流(N)	读取内部缓冲区当前偏移地址数据块
28H	RD_HOST_REQ_DATA		数据长度(2) 数据流(N)	读取内部缓冲区当前返回的指定长度数据块
2DH	WR_HOST_OFS_DATA	偏移地址(2) 输入长度(2) 数据流(N)		向内部缓冲区指定偏移地址写入数据块
2EH	WR_HOST_CUR_DATA	输入长度(2) 数据流(N)		向内部缓冲区当前偏移地址写入数据块
2FH	SET_FILE_NAME	字符串(N)		设置将要操作的文件的文件名
30H	DISK_CONNECT		产生中断	检查磁盘是否连接
31H	DISK_MOUNT		产生中断	初始化磁盘并测试磁盘是否就绪
32H	FILE_OPEN		产生中断	打开文件或目录，枚举文件和目录
33H	FILE_ENUM_GO		产生中断	继续枚举文件和目录

34H	FILE_CREATE		产生中断	新建文件
35H	FILE_ERASE		产生中断	删除文件
36H	FILE_CLOSE	是否允许更新	产生中断	关闭当前已经打开的文件或目录
37H	DIR_INFO_READ	目录索引号	产生中断	读取文件的目录信息
38H	DIR_INFO_SAVE	目录索引号	产生中断	保存文件的目录信息
39H	BYTE_LOCATE	偏移字节数(4)	产生中断	以字节为单位移动当前文件指针
3AH	BYTE_READ	请求字节数(2)	产生中断	以字节为单位从当前位置读取数据块
3CH	BYTE_WRITE	请求字节数(2)	产生中断	以字节为单位向当前位置写入数据块
3EH	DISK_CAPACITY		产生中断	查询磁盘物理容量
3FH	DISK_QUERY		产生中断	查询磁盘空间信息
40H	DIR_CREATE		产生中断	新建目录并打开或打开已存在的目录
4AH	SEC_LOCATE	偏移扇区数(4)	产生中断	以扇区为单位移动当前文件指针
4BH	SEC_READ	请求扇区数	产生中断	以扇区为单位从当前位置读取数据块
4CH	SEC_WRITE	请求扇区数	产生中断	以扇区为单位在当前位置写入数据块
50H	DISK_BOC_CMD		产生中断	对 USB 存储器执行 BO 传输协议的命令
54H	DISK_READ	LBA 扇区地址(4) 扇区数	产生中断	从 USB 存储器或 SD 卡读物理扇区
56H	DISK_WRITE	LBA 扇区地址(4) 扇区数	产生中断	向 USB 存储器或 SD 卡写物理扇区
55H	FILE_QUERY		产生中断	查询当前打开的文件的信息
57H	FILE_MODIFY		产生中断	修改当前打开的文件的信息

如果命令的输出数据是操作状态，参考下表。

状态代码	状态名称	状态说明
51H	CMD_RET_SUCCESS	操作成功
5FH	CMD_RET_ABORT	操作失败

表中标为“产生中断”的命令通常需要时间去执行操作。CH378 在命令执行完成后向单片机请求中断，单片机可以读取中断状态作为该命令的操作状态。如果中断状态是 ERR_SUCCESS，那么说明操作成功，有些命令具有返回数据（参考 CH378INC.H 文件中的 CH378_CMD_DATA 结构），可以通过 CMD_RD_HOST_REQ_DATA 命令读取返回数据。

5.1. CMD_GET_IC_VER

该命令用于获取芯片及固件版本。返回的一个字节数据是版本号，其位 7 为 0，位 6 为 1，位 5~位 0 为版本号。如果返回值为 41H，去掉位 7 和位 6，版本号为 01H。

5.2. CMD_SET_BAUDRATE

该命令用于设置 CH378 的串口通讯波特率。在 CH378 工作于串口通讯方式时，复位后默认的通讯波特率由 SDO, SDI 和 SCK 三个引脚的电平组合设定（参考本手册第 6.4 节），这些引脚都悬空时默认是 9600bps。如果单片机支持较高的通讯速度，可以通过该命令动态调整串口通讯波特率。该命令需要输入三个字节数据，分别是波特率系数 0、波特率系数 1、波特率系数 2，下表是系数与波特率的对应关系。

波特率系数 2	波特率系数 1	波特率系数 0	串口通讯波特率 (bps)	误差
00H	12H	00H	4800	0%
00H	25H	80H	9600	0%

00H	4BH	00H	19200	0%
00H	96H	00H	38400	0%
00H	E1H	00H	57600	0%
01H	2CH	00H	76800	0.6%
01H	C2H	00H	115200	0.9%
07H	08H	00H	460800	0%
0EH	10H	00H	921600	7.1%
01H	86H	A0H	100000	0%
0FH	42H	40H	1000000	7.6%
2DH	C6H	C0H	3000000	0%
计算公式：				
$BaudRate = (\text{波特率系数} 2 \ll 16) + (\text{波特率系数} 1 \ll 8) + \text{波特率系数} 0$				

通常情况下，设置串口通讯波特率在 1mS 时间之内完成，完成后 CH378 以新设定的通讯波特率输出操作状态，所以单片机应该在发出命令后及时调整自身的通讯波特率。

5.3. CMD_ENTER_SLEEP

该命令使 CH378 芯片进入低功耗睡眠挂起状态，从而节约电能。该命令需要输入 1 个字节睡眠模式参数，如果输入数据是 0x11，则使能芯片进入半睡眠状态，唤醒之后 USB 功能可以继续使用；如果输入数据是 0x22，使能芯片进入深度睡眠状态。半睡眠模式，CH378 部分时钟仍然处于工作状态，如果操作 SD 卡，大约可以降低 30mA 电流，如果操作 USB 存储设备，大约可以降低 20mA 电流。深度睡眠模式，大约可以降低 80mA 电流，如果操作的 SD 卡，唤醒后可以继续进行后续操作；如果操作的是 USB 设备，则唤醒之后需要全部重新初始化（包括设置工作模式、检测 USB 设备连接、初始化 USB 设备等）。

以下两种情况可以使能 CH378 唤醒，退出低功耗状态：

(1)、单片机向 CH378 写入新命令（没有输入数据的命令，例如 CMD_GET_IC_VER 或 CMD_GET_STATUS 命令等）。

(2)对于 SPI 串行接口或者 8 位并行接口，CS 片选有效也将导致 CH378 退出低功耗状态，所以单片机在发出 CMD_ENTER_SLEEP 命令后，应该立即使 CS 片选无效。

通常情况下，CH378 芯片从低功耗状态退出并恢复到正常工作状态需要几毫秒的唤醒时间，当完全恢复到正常工作状态后，CH378 将产生 USB_INT_WAKE_UP 事件中断。

5.4. CMD_RESET_ALL

该命令使 CH378 执行硬件复位。通常情况下，硬件复位在 35mS 时间之内完成。

5.5. CMD_CHECK_EXIST

该命令用于测试通讯接口和工作状态，以检查 CH378 是否正常工作。该命令需要输入 1 个字节数据，可以是任意数据，如果 CH378 正常工作，那么 CH378 的输出数据是输入数据的按位取反。例如，输入数据是 57H，则输出数据是 A8H。另外，对于并口通讯方式的 CH378，在其复位后未收到任何命令之前，从其并口通常可以读出数据 00H。

5.6. CMD_SET_SDO_INT

该命令用于设置 SPI 接口的 SDO 引脚的中断方式。该命令首先需要输入 1 个字节数据 16H，然后需要输入 1 个字节新的中断方式。中断方式有两种：00H 禁止 SDO 引脚用于中断输出，其在 SCS 片选无效时三态输出禁止，便于与其它设备共享单片机的 SPI 总线；01H 设置 SDO 引脚一直处于输出状态，

在 SCS 片选无效时兼做中断请求输出，等效于 INT#引脚，供单片机查询中断请求状态。

5.7. CMD_GET_FILE_SIZE

该命令用于获取当前文件的长度，即字节数。该命令需要输入 1 个字节数据 68H，输出为当前正在打开的文件的长度，该长度是以低字节在前的 4 个字节表示的双字数据（32 位）。

5.8. CMD_SET_FILE_SIZE

该命令用于设置当前文件的长度，即字节数。该命令首先需要输入 1 个字节的数据 68H，然后需要输入新的文件长度，该长度是以低字节在前的 4 个字节表示的双字数据（32 位）。

该命令只修改 CH378 内存中的文件长度变量，以便进行某些操作，比如以扇区方式读取文件最后不足一个扇区的数据，并没有实际修改文件长度。如果需要实际修改文件长度，还需要执行 CMD_FILE_CLOSE 命令。

5.9. CMD_SET_USB_MODE

该命令用于设置 USB 工作模式。该命令需要输入 1 个数据，该数据是模式代码：

模式代码为 00H 时切换到未启用的 USB 设备方式；

模式代码为 01H 时切换到已启用的 USB 设备方式，外部固件模式；

模式代码为 02H 时切换到已启用的 USB 设备方式，内部固件模式；

模式代码为 03H 时切换到未启用的 SD 卡主机模式，用于管理和存取 SD 卡中的文件；

模式代码为 04H 时切换到已启用的 SD 卡主机模式，用于管理和存取 SD 卡中的文件；

模式代码为 05H 时切换到未启用的主机方式；

模式代码为 06H 时切换到未启用的 USB 主机方式，用于管理和存取 USB 存储设备中的文件；

模式代码为 07H 时切换到已启用的 USB 主机方式，用于管理和存取 USB 存储设备中的文件；

模式代码为 08H 时切换到 USB 读卡器模式，用于计算机 USB 口直接管理 SD 卡中的文件；

其中，未启用是指不自动检测 USB 存储设备或 SD 卡是否连接，所以需要外部单片机检测；启用是指自动检测 USB 存储设备或 SD 卡是否连接，当 USB 存储设备或 SD 卡连接或者断开时都会产生中断通知外部单片机。一般操作 SD 卡选择模式代码 03H，操作 USB 存储设备选择模式代码 06H。另外，异步串口方式下，不支持模式 4。

CH378 工作在模式 8 (版本号大于等于 0x43 时支持该模式) 时，仅作为 SD 卡读卡器使用，外围 MCU 不能操作 SD 卡和 USB 存储设备，如果需要操作，需要重新切换回原工作模式。

通常情况下，设置 SD 卡主机模式在 1mS 之内完成，设置 USB 工作模式在 35mS 时间之内完成，完成后输出操作状态。

5.10. CMD_GET_STATUS

该命令用于获取 CH378 的中断状态并通知 CH378 取消中断请求。当 CH378 向单片机请求中断后，单片机通过该命令获取中断状态，分析中断原因并处理。

中断状态字节	中断状态的分类
10H~1FH	SD 卡或者 USB 主机方式的操作中断状态
20H~3FH	USB 主机方式的通讯失败状态，用于分析操作失败原因
40H~4FH	SD 卡或者 USB 主机文件模式下的文件系统警告代码
80H~BFH	SD 卡或者 USB 主机文件模式下的文件系统错误代码

下面是 SD 卡或者 USB 主机方式的操作中断状态。

状态字节	状态名称	中断状态分析说明
00H	ERR_SUCCESS	操作成功

03H	ERR_PARAMETER_ERROR	参数错误
14H	USB_INT_SUCCESS	SD 卡或者 USB 事务或者传输操作成功
15H	USB_INT_CONNECT	检测到 USB 设备连接事件
16H	USB_INT_DISCONNECT	检测到 USB 设备断开事件
17H	USB_INT_BUF_OVER	传输的数据有误或者数据太多缓冲区溢出
18H	USB_INT_USB_READY	USB 设备已经被初始化（已经分配 USB 地址）
1DH	USB_INT_DISK_READ	USB 存储器请求数据读出
1FH	USB_INT_DISK_ERR	存储设备操作失败

下面是 SD 卡或者 USB 主机文件模式下的文件系统警告代码和错误代码表。

状态字节	状态名称	中断状态分析说明
41H	ERR_OPEN_DIR	指定路径的目录被打开
42H	ERR_MISS_FILE	指定路径的文件没有找到，可能是文件名称错误
43H	ERR_FOUND_NAME	搜索到相匹配的文件名， 或者是要求打开目录而实际结果却打开了文件
44H	ERR_NO_NAME	此短文件名没有长文件名或错误的长文件
45H	ERR_BUF_OVER	长文件缓冲区溢出
46H	ERR_LONG_NAME	错误的长文件名
82H	ERR_DISK_DISCON	磁盘尚未连接，可能磁盘已经断开
84H	ERR_LARGE_SECTOR	磁盘的扇区太大，只支持每扇区 512 字节
92H	ERR_TYPE_ERROR	磁盘分区类型不支持，需由磁盘管理工具重新分区
A1H	ERR_BPB_ERROR	磁盘尚未格式化，或者参数错误， 需要由 WINDOWS 采用默认参数重新格式化
B1H	ERR_DISK_FULL	磁盘文件太满，剩余空间太少或者已经没有
B2H	ERR_FDT_OVER	目录内文件太多，没有空闲目录项，需磁盘整理， FAT12/FAT16 根目录下的文件数应该少于 512 个
B3H	ERR_MISS_DIR	指定路径的某个子目录（文件夹）没有找到，可能是目 录名称错误
B4H	ERR_FILE_CLOSE	文件已经关闭，如果需要使用，应该重新打开文件

更多中断状态可以查看 CH378 芯片对应头文件 CH378INC.H。

5.11. CMD_RD_HOST_OFS_DATA

该命令用于读取内部指定缓冲区指定偏移地址的数据块。首先写入 2 个字节的偏移地址（内部缓冲区起始地址加上该偏移地址得到该命令数据块的读取的起始地址），接着写入 2 个字节要读取的数据块长度，也就是后续数据流的字节数，最后按照长度逐个读取后续数据流。

注意：数据块长度的有效范围为 0 至 20480，如果长度不为 0，则单片机必须将后续数据从 CH378 逐个读取完。

5.12. CMD_RD_HOST_CUR_DATA

该命令用于读取内部指定缓冲区当前偏移地址的数据块。首先写入 2 个字节要读取的数据块长度，也就是后续数据流的字节数，然后按照长度逐个读取后续数据流。

注意：数据块长度的有效范围为 0 至 20480，如果长度不为 0，则单片机必须将后续数据从 CH378 逐个读取完。

该命令与 CMD_RD_HOST_OFS_DATA 命令的区别在于，前者读取的是内部缓冲区指定偏移地址的数据块，需要输入偏移地址；后者读取的是内部缓冲区当前偏移地址的数据块，不需要输入偏移地址。分多次小批量读取数据时，两个命令可以配合使用。

5.13. CMD_RD_HOST_REQ_DATA

该命令用于读取内部指定缓冲区当前偏移地址请求读取的数据块。首先读取的是 2 个字节的数数据块长度，也就是后续数据流的字节数，然后按照长度逐个读取后续数据流。

注意：数据块长度的有效范围为 0 至 20480，如果长度不为 0，则单片机必须将后续数据从 CH378 逐个读取完。

5.14. CMD_WR_HOST_OFS_DATA

该命令用于向内部指定缓冲区指定偏移地址写入数据块。首先写入 2 个字节的偏移地址（内部缓冲区起始地址加上该偏移地址得到该命令数据块的写入的起始地址），接着写入 2 个字节要输入的数据块长度，也就是后续数据流的字节数，最后按照长度逐个输入后续数据流。

注意：数据块长度的有效范围为 0 至 20480，如果长度不为 0，则单片机必须将后续数据逐个写入 CH378。

5.15. CMD_WR_HOST_CUR_DATA

该命令用于向内部指定缓冲区当前偏移地址写入数据块。首先写入 2 个字节要输入的数据块长度，也就是后续数据流的字节数，然后按照长度逐个输入后续数据流。

注意：数据块长度的有效范围为 0 至 20480，如果长度不为 0，则单片机必须将后续数据逐个写入 CH378。

5.16. CMD_SET_FILE_NAME

该命令用于设置将要操作的文件或者目录（文件夹）的文件名或者目录名（路径名）。输入数据是以 0 结束的字符串，并且含结束符 0 在内的长度不得超过 128 个字符。

对于多级子目录下的文件，由于路径名太长，可以分多次逐级打开，首先打开子目录，直到最后再打开文件。其中，首先打开必须是从根目录开始，所以路径名首字符必须是斜杠，以后接着前级再打开时的首字符必须不是斜杠。当文件操作出错时，必须回到根目录下重新逐级打开。具体操作见 CMD_FILE_OPEN 命令。

文件名（或者目录名、路径名）的格式与 DOS 系统的短文件名格式相同，但是不需要盘符和冒号，作为根目录符的左斜杠“/”与右斜杠“\”等效，建议使用左斜杠“/”。所有字符必须是大写字母、数字或者中文汉字以及某些特殊字符，文件名长度不超过 11 个字符，其中主文件名不超过 8 个字符，扩展名不超过 3 个字符，如果有扩展名，那么用小数点与主文件名隔开。参考 EXAM11 例子可以支持长文件名。

字符串中没有字符（但有结束符 0，下同）时，说明是初始化文件系统，不打开任何文件；

字符串中只有一个“/”或者“\”（左斜杠或者右斜杠）时，说明是打开根目录；

字符串首字符为“/”或者“\”并且后续字符是文件名时，说明是根目录下的文件；

字符串直接是文件名时，说明是当前目录下的文件。

例如，对于根目录下的 FILENAME.EXT 文件，可以用字符串“/FILENAME.EXT\0”设置，整个字符串含结束符共 14 个字符，其中“\0”为 C 语言表示的 0，作为字符串结束符，字符串中的“/”表示根目录，在 C 语言中也可以用“\\”（实际是一个\字符）表示根目录。

5.17. CMD_DISK_CONNECT

该命令用于检查磁盘或者 SD 卡是否连接。CH378 在命令执行完成后向单片机请求中断。如果操作状态是 ERR_SUCCESS，那么说明有磁盘或者 USB 设备或者 SD 卡连接。

5.18. CMD_DISK_MOUNT

该命令用于初始化磁盘并测试磁盘是否就绪。新连接的 USB 存储设备或者 SD 卡必须通过该命令进行初始化后，然后才能进行文件操作，有些 USB 存储设备可能需要多次初始化才能返回操作成功状态 ERR_SUCCESS。另外，在文件操作的过程中，该命令也可以随时用于测试磁盘是否就绪。

首次执行 CMD_DISK_MOUNT 命令如果中断状态是 ERR_SUCCESS，那么可以由 CMD_RD_HOST_REQ_DATA 命令获取数据，数据通常是 36 个字节，包括 USB 存储设备的特性以及厂商和产品的识别信息等。

5.19. CMD_FILE_OPEN

该命令用于打开文件或者目录（文件夹），枚举文件和目录（文件夹）。

打开文件（或者目录）是读写文件（或者目录）之前的必要操作。在打开文件命令之前，应该先通过 CMD_SET_FILE_NAME 命令设置将要打开或者枚举的文件的文件名。

如果完整的路径名长度在限制的长度范围内，可以一次性打开。

如果是多级子目录下的文件，路径名较长，那么可以从根目录开始分多次逐级打开。首次打开必须是从根目录开始，所以路径名首字符必须是斜杠“/”或者“\”，以后接着前级再打开时的首字符必须不是“/”或者“\”。

由于路径名有长度限制，所以子目录的深度不宜超过 5 级。

如果成功打开目录，那么中断状态返回 ERR_OPEN_DIR，此时文件长度无效，为 OFFFFFFFFFH。

如果成功打开文件，那么中断状态返回 ERR_SUCCESS，此时文件长度有效。

如果没有找到指定的文件或者目录（文件夹），那么中断状态返回 ERR_MISS_FILE。

例如：

要打开根目录下的文件\TODAY1.TXT，步骤如下：

- ① 用字符串“/TODAY1.TXT\0”通过 CMD_SET_FILE_NAME 命令设置文件名；
- ② 用 CMD_FILE_OPEN 命令打开文件。

要打开三级子目录下的文件\YEAR2004\MONTH05.NEW\DATE18\ADC.TXT，步骤如下：

- ① 用字符串“/YEAR2004/MONTH05.NEW/DATE18/ADC.TXT\0”通过 CMD_SET_FILE_NAME 命令设置文件名；
- ② 用 CMD_FILE_OPEN 命令直接打开文件。

要打开六级子目录下的文件\MODULE\AD_DATA\YEAR2004\MONTH05.NEW\DATE18\HOUR08\ADC.TXT，步骤如下：

- ① 用字符串“/MODULE/AD_DATA/YEAR2004/MONTH05.NEW\0”通过 CMD_SET_FILE_NAME 命令设置文件名；
- ② 用 CMD_FILE_OPEN 命令一次性打开前面四级目录，打开目录后，如果执行 CMD_GET_FILE_SIZE 命令将返回无效的文件长度 OFFFFFFFFFH；
- ③ 用字符串“DATE18/HOUR08/ADC.TXT\0”通过 CMD_SET_FILE_NAME 命令设置文件名；
- ④ 用 CMD_FILE_OPEN 命令打开最终的文件，打开文件后，如果执行 CMD_GET_FILE_SIZE 命令将返回实际的文件长度。

要初始化文件系统，而不打开任何文件，步骤如下：

- ① 用字符串“\0”通过 CMD_SET_FILE_NAME 命令设置文件名；
- ② 执行 CMD_FILE_OPEN 命令，那么将初始化文件系统（如果已经初始化那么直接返回）。

要打开根目录（例如，在处理长文件名时），步骤如下：

- ① 用字符串“\0”通过 CMD_SET_FILE_NAME 命令设置文件名；
- ② 执行 CMD_FILE_OPEN 命令，那么将打开根目录（使用后必须用 CMD_FILE_CLOSE 关闭）。

5.20. CMD_FILE_ENUM_GO

该命令用于继续枚举文件和目录（文件夹）。

如果需要搜索和查询文件，步骤如下：

- ① 用通配符*代替需要查询的文件名中的全部或者部分字符，通配符*后面不能再有字符，通过 CMD_SET_FILE_NAME 命令将含有通配符*的字符串设置为文件名，例如，字符串“/*\0”说明要枚举根目录下的所有文件或者目录，字符串“USB*\0”说明要枚举当前目录下所有名称是

以“USB”三个字符开头的文件或者目录，符合要求的文件名（或者目录名）包括“USB.TXT”、“USB1234”、“USB”、“USBC.H”等，但不包括“XUSB”、“U.SB”、“U2SB”、“MY.USB”等；

- ② 通过 `CMD_FILE_OPEN` 命令开始枚举文件和目录；
- ③ CH378 比较每一个文件名，每当找到一个符合要求的文件，将对单片机产出一中断，中断状态为 `USB_INT_DISK_READ`，请求单片机从 CH378 读取数据；
- ④ 单片机通过 `CMD_RD_HOST_REQ_DATA` 命令读出数据，立即分析处理或者先保存，该数据是枚举到的当前文件的完整路径名；
- ⑤ 单片机发出 `CMD_FILE_ENUM_GO` 命令，通知 CH378 继续枚举；
- ⑥ CH378 继续比较文件名，如果再次找到符合要求的文件，那么转步骤③，否则继续下一步；
- ⑦ CH378 对单片机产出一中断，中断状态为 `ERR_MISS_FILE`，说明没有找到更多的符合要求的文件，整个枚举操作结束。

在上述步骤④中，单片机可以对获得的路径名中的文件名进行分析，查看是否是寻找的文件。例如，对文件扩展名 `DIR_Name[8]`、`[9]`、`[10]` 三个单元的字符与“XLS”进行比较，以过滤特定的 EXCEL 类型的文件。

5.21. `CMD_FILE_CREATE`

该命令用于新建文件，如果文件已经存在那么先删除再新建。

在新建文件命令之前，应该先通过 `CMD_SET_FILE_NAME` 命令设置将要新建的文件的文件名，格式与 `CMD_FILE_OPEN` 命令相同，但不支持通配符。如果存在同名文件，那么该同名文件将首先被删除，然后再新建文件。如果不希望已有文件被删除，那么应该事先通过 `CMD_FILE_OPEN` 命令确认文件不存在后再新建。新建文件的文件日期和时间默认为 2004 年 1 月 1 日 0 时 0 分 0 秒，文件默认长度为 1，如果需要修改这些信息，那么可以通过 `CMD_DIR_INFO_READ` 和 `CMD_DIR_INFO_SAVE` 命令实现。

5.22. `CMD_FILE_ERASE`

该命令用于删除文件，如果文件已经被打开那么将被直接删除，否则，对于文件会自行先打开再删除，而对于子目录则必须先被打开。

对于普通文件，删除步骤如下：

- ① 确认之前的文件或者目录已经关闭，否则将被直接删除，而不受步骤②的影响；
- ② 通过 `CMD_SET_FILE_NAME` 命令设置将要被删除的文件名，不支持通配符；
- ③ 通过 `CMD_FILE_ERASE` 命令自行打开文件并删除。

对于子目录（或者文件）必须参照以下步骤删除：

- ① 对于子目录，必须事先删除子目录中的所有文件以及下级子目录；
- ② 通过 `CMD_SET_FILE_NAME` 命令设置将要被删除的子目录名（或者文件名），不支持通配符；
- ③ 通过 `CMD_FILE_OPEN` 命令打开子目录名（或者文件名）；
- ④ 通过 `CMD_FILE_ERASE` 命令删除步骤②中已经打开的子目录（或者文件）。

5.23. `CMD_FILE_CLOSE`

该命令用于关闭当前已经打开的文件或者目录（文件夹）。该命令需要 1 个输入数据，表明是否允许更新文件长度，为 0 则禁止更新文件长度，为 1 则允许自动更新文件长度。

打开文件或者目录（文件夹）读写完毕后，应该关闭文件。对于根目录的操作，关闭文件是必须的。对于普通文件的读操作，关闭文件是可选操作。对于普通文件的写操作，关闭文件的同时，可以选择是否由 CH378 自动更新文件长度。

如果通过 `CMD_SEC_LOCATE`、`CMD_SEC_READ` 或者 `CMD_SEC_WRITE` 命令对文件进行以扇区为单位的读写，那么 CH378 自动更新的文件长度是以扇区为单位计算的，文件长度通常是扇区大小 512 的倍数。如果希望文件长度不是扇区大小的倍数，那么单片机可以在关闭文件前通过 `CMD_SET_FILE_SIZE` 命令修改文件长度变量，或者通过 `CMD_DIR_INFO_READ` 和 `CMD_DIR_INFO_SAVE` 命令直接修改文件信息。

如果通过 `CMD_BYTE_LOCATE`、`CMD_BYTE_READ` 或者 `CMD_BYTE_WRITE` 命令对文件进行以字节为单位

的读写，那么 CH378 自动更新的文件长度是以字节为单位计算的，所以可以获得适当的长度。

5.24. CMD_DIR_INFO_READ

该命令用于读取文件的目录信息，即 FAT_DIR_INFO 结构。该命令需要输入 1 个字节数据，指定需要读取的目录信息结构在扇区内的索引号，索引号范围为 00H~0FH，索引号 0FFH 则对应当前已经打开的文件。该命令只是将目录信息读取到 CH378 内存缓冲区，之后单片机可以通过 CMD_RD_HOST_REQ_DATA 命令读出数据。

每次打开一个文件后，CH378 从 USB 存储设备或者 SD 卡中取出相邻的 16 个文件目录信息存放于内存，单片机可以指定索引号 0~15 分别对应于各个 FAT_DIR_INFO 结构，也可以指定索引号 0FFH 以获取当前正在打开的文件的 FAT_DIR_INFO 结构，以分析文件日期、时间、长度、属性等信息。

5.25. CMD_DIR_INFO_SAVE

该命令用于保存文件的目录信息。该命令需要输入 1 个字节数据，指定需要写入的目录信息结构在扇区内的索引号，索引号范围为 00H~0FH，索引号 0FFH 则对应当前已经打开的文件。

执行该命令可以将内存中的 16 个文件目录信息刷新保存到 USB 存储设备或者 SD 卡中。修改文件目录信息的步骤如下：

- ① 文件已经打开那么转②，否则通过 CMD_SET_FILE_NAME 和 CMD_FILE_OPEN 命令打开文件；
- ② 通过 CMD_DIR_INFO_READ 命令读取当前文件或相邻文件的 FAT_DIR_INFO 结构到内存缓冲区；
- ③ 通过 CMD_RD_HOST_REQ_DATA 命令从内存缓冲区读出数据，如果不需要修改那么步骤结束；
- ④ 如果需要修改，那么再次通过 CMD_DIR_INFO_READ 命令读取 FAT_DIR_INFO 结构到缓冲区；
- ⑤ 通过 CMD_WR_HOST_OFS_DATA 命令向内部缓冲区写入 32 个字节的目录信息；
- ⑥ 通过 CMD_DIR_INFO_SAVE 命令向 USB 存储设备或者 SD 卡中保存修改后的文件目录信息。

5.26. CMD_BYTE_LOCATE

该命令用于以字节为单位移动当前文件指针。该命令需要输入偏移字节数，该偏移字节数是以低字节在前的 4 个字节表示的双字数据（32 位）。如果命令执行完成后中断状态是 ERR_SUCCESS，那么可以由 CMD_RD_HOST_REQ_DATA 命令获取当前文件指针对应的绝对线性扇区号 LBA（以低字节在前的 4 个字节表示的 32 位双字数据），如果已到文件末尾，那么该值为 0FFFFFFFH。

文件被新建或者被重新打开时，当前文件指针都为 0，移动当前文件指针，通常用于从指定位置开始读写数据。例如，单片机希望跳过文件的前 158 字节再读写数据，那么可以用 CMD_BYTE_LOCATE 命令带上参数 158 作为偏移字节数，命令执行成功后，紧接在后面的读写操作将从第 158 字节开始。对于写操作，如果单片机准备在原文件的尾部继续添加数据，而不希望影响前面的原有数据，那么可以指定很大的字节偏移，例如 0FFFFFFFH，将文件指针移动原文件的末尾，以便追加数据。

5.27. CMD_BYTE_READ

CMD_BYTE_READ 命令用于以字节为单位从当前位置读取数据块。读取成功后，CH378 自动同步移动文件指针，以便下次读写操作能够接着本次读取数据的结束位置开始。该命令需要输入请求读取的字节数，该请求字节数是以低字节在前（Little-Endian）的 2 个字节表示的字数据（16 位），最大为 20480 字节。该命令执行完毕后会以中断通知外部单片机，此时读取到的文件数据保存在 CH378 内部缓冲区中，需要通过 CMD_RD_HOST_OFS_DATA 或 CMD_RD_HOST_CUR_DATA 命令进行实际读取。

完整的字节读操作步骤如下：

- ① 打开文件，并且确认处于合适的位置（文件指针）；
- ② 单片机发出 CMD_BYTE_READ 命令并输入请求读取的字节数，开始读操作；
- ③ CH378 计算从当前文件指针开始到文件结束位置之间的文件剩余长度，并根据请求字节数、文件剩余长度、内部缓冲区状态计算出本次允许读取的字节数，读取文件数据到内部缓冲区中，同时移动当前文件指针，然后中断通知单片机，中断状态为 ERR_SUCCESS；

- ④ 单片机通过发送 `CMD_GET_REAL_LEN` 命令，快速得到本次允许读取的实际长度；
- ⑤ 单片机通过发送 `CMD_RD_HOST_OFS_DATA` 或 `CMD_RD_HOST_CUR_DATA` 命令从 CH378 内部缓冲区读取实际数据。如果单片机 RAM 足够，可以一次性读取完整的全部数据。如果单片机 RAM 相对较少，也可以分多次小批量读取数据，边读取边处理。
- ⑥ 文件结束或者请求读取的字节数全部读取完毕，整个读操作结束。

如果请求读取的长度大于 `CMD_GET_REAL_LEN` 命令返回的允许读取的实际长度，那么说明文件指针已经处于文件末尾。

5.28. `CMD_BYTE_WRITE`

`CMD_BYTE_WRITE` 命令用于以字节为单位向当前位置写入数据块。写入成功后，CH378 自动同步移动文件指针，以便下次读写操作能够接着本次写入数据的结束位置开始。该命令需要输入请求写入的字节数，该请求字节数是以低字节在前（Little-Endian）的 2 个字节表示的字节数据（16 位），最大为 20480 字节。当请求字节数为 0 时，仅用于刷新文件长度。

该命令执行前需要先通过 `CMD_WR_HOST_OFS_DATA` 或 `CMD_WR_HOST_CUR_DATA` 命令将要写入的数据预先输入到 CH378 内部缓冲区中。

完整的字节写操作步骤如下：

- ① 打开或者新建文件，并且确认处于合适的位置（文件指针）；
- ② 单片机通过 `CMD_WR_HOST_OFS_DATA` 或 `CMD_WR_HOST_CUR_DATA` 命令将要写入的数据预先输入到 CH378 内部缓冲区中；
- ③ 单片机发出 `CMD_BYTE_WRITE` 命令并输入请求写入的字节数，开始写操作；
- ④ CH378 检查请求的字节数，如果为 0，那么执行刷新文件长度操作，将内存中文件长度变量保存到 USB 存储设备或者 SD 卡中，完成后输出中断状态为 `ERR_SUCCESS`，结束本次写操作；如果不为 0，则根据请求字节数、内部缓冲区状态计算出本次允许写入的字节数，将内部缓冲区中的数据写入文件，同时移动当前文件指针，然后中断通知单片机，中断状态为 `ERR_SUCCESS`，结束本次写操作。

如果直接向文件末尾追加数据，或者在写操作过程中，自动移动的文件指针超出原文件结束位置，那么 CH378 将自动更新内存中的文件长度变量。在整个写操作完成后，如果短时间内不打算进行另一次写操作，那么单片机应该通知 CH378 刷新内存中的文件长度变量到 USB 存储设备或者 SD 卡中，方法有两种：

- (1)、类似上述步骤②和③的写 0 长度数据；
- (2)、执行 `CMD_FILE_CLOSE` 命令并且允许更新长度。

5.29. `CMD_DISK_CAPACITY`

该命令用于查询磁盘物理容量，支持 USB 存储设备或者 SD 卡。如果命令执行完成后中断状态是 `ERR_SUCCESS`，那么可以由 `CMD_RD_HOST_REQ_DATA` 命令获取磁盘物理容量，即总扇区数，该容量是以低字节在前的 4 个字节表示的双字数据（32 位），如果再乘以扇区大小，那么可以得到以字节为单位的物理总容量。

5.30. `CMD_DISK_QUERY`

该命令用于查询磁盘空间信息，包括总空间、剩余空间和文件系统类型。如果命令执行完成后中断状态是 `ERR_SUCCESS`，那么可以由 `CMD_RD_HOST_REQ_DATA` 命令依次获取逻辑盘的总扇区数（以低字节在前的 4 个字节表示的 32 位双字数据），当前逻辑盘的剩余扇区数（以低字节在前的 4 个字节表示的 32 位双字数据），逻辑盘的 FAT 文件系统类型（参考 CH378 INC. H 文件中的 `CH378_CMD_DATA` 结构）。

5.31. `CMD_DIR_CREATE`

该命令用于新建子目录（文件夹）并打开，如果子目录已经存在那么直接打开。操作方法与

CMD_FILE_CREATE 命令相同。在新建子目录命令之前，应该先通过 CMD_SET_FILE_NAME 命令设置将要新建的子目录的目录名，格式与 CMD_FILE_CREATE 命令相同。如果存在同名的普通文件，那么中断状态为 ERR_FOUND_NAME；如果子目录新建成功或者打开事先已经存在的子目录，那么中断状态为 ERR_SUCCESS。新建子目录的文件日期和时间等信息与 CMD_FILE_CREATE 命令新建文件时相同，修改方法也相同，只是文件属性为 ATTR_DIRECTORY，并且文件长度始终为 0（根据 FAT 规范，子目录的文件长度必须为 0）。

5.32. CMD_SEC_LOCATE

该命令用于以扇区为单位移动当前文件指针，支持 USB 存储设备或者 SD 卡。该命令需要输入偏移扇区数，该偏移扇区数是以低字节在前的 4 个字节表示的双字数据（32 位）。如果命令执行完成后中断状态是 ERR_SUCCESS，那么可以由 CMD_RD_HOST_REQ_DATA 命令获取当前文件指针对应的绝对线性扇区号 LBA（以低字节在前的 4 个字节表示的 32 位双字数据），如果已到文件末尾，那么该值为 0FFFFFFFH。

文件被新建或者被重新打开时，当前文件指针都为 0，移动当前文件指针，通常用于从指定位置开始读写数据。例如，单片机希望跳过文件的前 18 个扇区再读写数据，那么可以用 CMD_SEC_LOCATE 命令带上参数 18 作为偏移扇区数，命令执行成功后，紧接在后面的读写操作将从第 18 扇区开始。对于写操作，如果单片机准备在原文件的尾部继续添加数据，而不希望影响前面的原有数据，那么可以指定很大的字节偏移，例如 0FFFFFFFH，将文件指针移动原文件的末尾，以便追加数据。

5.33. CMD_SEC_READ

该命令用于获取以扇区为单位从当前位置读取数据块的参数信息，支持 USB 存储设备或者 SD 卡。命令成功执行后，CH378 自动同步移动文件指针，以便下次读写操作能够接着本次读取数据的结束位置开始。该命令需要输入 1 个数据，指定请求读取的扇区数，有效值为 1 至 40。如果命令执行完成后中断状态是 ERR_SUCCESS，那么可以由 CMD_GET_REAL_LEN 命令，快速得到本次允许读取的实际扇区数（以低字节在前的 4 个字节表示的 32 位双字数据）。

完整的扇区读操作步骤如下：

- ① 打开文件，并且确认处于合适的位置（文件指针）；
- ② 单片机发出 CMD_SEC_READ 命令并输入请求读取的扇区数，最大扇区数为 40；
- ③ CH378 计算从当前文件指针开始到文件结束位置之间的文件剩余长度，并根据请求扇区数、文件剩余长度、内部缓冲区状态计算出本次允许读取的扇区数，并将文件数据读取到 CH378 内部缓冲区中，同时移动当前文件指针，然后中断通知单片机，中断状态为 ERR_SUCCESS；
- ④ 单片机通过发送 CMD_GET_REAL_LEN 命令，快速得到本次允许读取的实际扇区数；
- ⑤ 单片机通过发送 CMD_RD_HOST_OFS_DATA 或 CMD_RD_HOST_CUR_DATA 命令从 CH378 内部缓冲区读取实际数据。
- ⑥ 允许读取的扇区数全部读取完毕，整个读操作结束。

如果请求读取的扇区数大于 CMD_GET_REAL_LEN 命令返回的允许读取的实际扇区数，那么说明文件指针已经处于文件末尾。

如果文件长度不是扇区的倍数，那么文件尾部不足一个扇区的零碎数据部分将被忽略，如果必须读出文件尾部不足一个扇区的零碎数据，那么可以通过 CMD_SET_FILE_SIZE 命令临时最大文件大小 VAR32_FILE_SIZE，最后一个扇区（实际有效数据不足一个扇区）读取完毕后，再恢复原文件大小 VAR32_FILE_SIZE。

5.34. CMD_SEC_WRITE

该命令用于获取以扇区为单位向当前位置写入数据块的参数信息，支持 USB 存储设备或者 SD 卡。命令成功执行后，CH378 自动同步移动文件指针，以便下次读写操作能够接着本次写入数据的结束位置开始。该命令需要输入 1 个数据，指定请求写入的扇区数，有效值为 0 至 40，请求扇区数为 0 时，仅用于刷新文件长度。

该命令执行前需要先通过 CMD_WR_HOST_OFS_DATA 或 CMD_WR_HOST_CUR_DATA 命令将要写入的数据

预先输入到 CH378 内部缓冲区中。

完整的扇区写操作步骤如下：

- ① 打开或者新建文件，并且确认处于合适的位置（文件指针）；
- ② 单片机通过 `CMD_WR_HOST_OFS_DATA` 或 `CMD_WR_HOST_CUR_DATA` 命令将要写入的数据预先输入到 CH378 内部缓冲区中；
- ③ 单片机发出 `CMD_SEC_WRITE` 命令并输入请求写入的扇区数，最大扇区数为 40；
- ④ CH378 检查请求的扇区数，如果为 0，那么执行刷新文件长度操作，将内存中文件长度变量保存到 USB 存储设备或者 SD 卡中，完成后输出中断状态为 `ERR_SUCCESS`，结束本次写操作；如果不为 0，则根据请求扇区数、内部缓冲区状态计算出本次允许写入的扇区数，将内部缓冲区中的数据写入文件，同时移动当前文件指针，然后中断通知单片机，中断状态为 `ERR_SUCCESS`，结束本次写操作。

如果直接向文件末尾追加数据，或者在写操作过程中，自动移动的文件指针超出原文件结束位置，那么 CH378 将自动更新内存中的文件长度变量。在整个写操作完成后，如果短时间内不打算进行另一次写操作，那么单片机应该通知 CH378 刷新内存中的文件长度变量到 USB 存储设备或者 SD 卡中，方法有两种：

- (1)、类似上述步骤②和③的写 0 长度数据；
- (2)、执行 `CMD_FILE_CLOSE` 命令并且允许更新长度。

5.35. `CMD_DISK_BOC_CMD`

该命令用于对 USB 存储设备执行 BulkOnly 传输协议的命令，不支持 SD 卡。在执行该命令之前，单片机必须先通过 `CMD_WR_HOST_OFS_DATA` 命令向 CH378 写入相应的 CBW 包（参考 CH378 INC. H 文件中的 `BULK_ONLY_CBW` 结构），CH378 在命令执行完成后向单片机请求中断，如果中断状态是 `ERR_SUCCESS` 则说明命令执行成功，对于有返回数据的操作，可以由 `CMD00_RD_HOST_REQ_DATA` 命令获取返回数据。

5.36. `CMD_DISK_READ`

`CMD_DISK_READ` 命令用于从 USB 存储设备读物理扇区，不支持 SD 卡。

`CMD_DISK_READ` 命令需要两组参数：4 个字节表示的扇区起始地址和 1 个字节表示的扇区数，扇区起始地址即线性扇区号 LBA，是以低字节在前的 4 个字节表示的 32 位双字数据。该命令需要 5 个输入数据，依次是 LBA 地址的最低字节，LBA 地址的较低字节，LBA 地址的较高字节，LBA 地址的最高字节，扇区数。该命令一次可以在 USB 存储设备中任意读取最多 40 个扇区的数据。

一次完整的物理扇区读操作步骤如下：

- ① 单片机发出 `CMD_DISK_READ` 命令并指定扇区起始 LBA 和扇区数，启动读操作；
- ② CH378 将指定连续的扇区数据读取到 CH378 内部缓冲区中，然后中断通知单片机，中断状态为 `ERR_SUCCESS`；
- ③ 单片机通过发送 `CMD_RD_HOST_OFS_DATA` 或 `CMD_RD_HOST_CUR_DATA` 命令从 CH378 内部缓冲区读取实际数据。如果单片机 RAM 足够，可以一次性读取完整的全部数据。如果单片机 RAM 相对较少，也可以分多次小批量读取数据，边读取边处理。

5.37. `CMD_DISK_WRITE`

`CMD_DISK_WRITE` 命令用于向 USB 存储设备写物理扇区，不支持 SD 卡。

`CMD_DISK_WRITE` 命令需要两组参数：4 个字节表示的扇区起始地址和 1 个字节表示的扇区数，扇区起始地址即线性扇区号 LBA，是以低字节在前的 4 个字节表示的 32 位双字数据。该命令需要 5 个输入数据，依次是 LBA 地址的最低字节，LBA 地址的较低字节，LBA 地址的较高字节，LBA 地址的最高字节，扇区数。该命令一次可以在 USB 存储设备中任意写入最多 40 个扇区的数据。

一次完整的物理扇区写操作步骤如下：

- ① 单片机通过 `CMD_WR_HOST_OFS_DATA` 或 `CMD_WR_HOST_CUR_DATA` 命令将要写入的数据预先输入到 CH378 内部缓冲区中；

- ② 单片机发出 CMD_DISK_WRITE 命令并指定扇区起始 LBA 和扇区数，启动写操作；

5.38. CMD_FILE_QUERY

该命令用于查询当前打开的文件信息，包括文件的属性、日期、时间、长度等。该命令执行完毕后，CH378 会产生中断通知单片机，单片机通过 CMD_RD_HOST_REQ_DATA 命令读取 9 个字节的返回数据。数据格式为：4 个字节的文件长度、2 个字节的文件日期、2 个字节的文件时间、1 个字节的文件属性。

5.39. CMD_FILE_MODIFY

该命令用于修改当前打开的文件信息，包括文件的属性、日期、时间、长度等。执行该命令前，需要先通过 CMD_WR_HOST_OFS_DATA 命令设置要修改的文件信息。数据格式为：4 个字节的文件长度、2 个字节的文件日期、2 个字节的文件时间、1 个字节的文件属性。若文件长度为 0xFFFFFFFF，文件日期为 0xFFFF，文件时间为 0xFFFF，文件属性为 0xFF，则表示该对应的属性不进行修改。

6. 功能说明

6.1. 单片机通讯接口

CH378 与单片机之间支持三种通讯接口：8 位并行接口、SPI 同步串行接口、异步串口。在芯片上电复位时，CH378 将采样 SEL 和 TXD 引脚的状态，根据这 2 个引脚状态的组合选择通讯接口，参考下表（表中 X 代表不关心此位，0 代表低电平，1 代表高电平或者悬空）。

SEL 引脚	TXD 引脚	选择通讯接口
1	1	异步串口
1	0	SPI 接口
0	1	8 位并口
0	0	错误接口

CH378 芯片 INT#引脚输出的中断请求默认是低电平有效，可以连接到单片机的中断输入引脚或普通输入引脚，单片机可以使用中断方式或查询方式获知 CH378 的中断请求。为了节约引脚，单片机可以不连接 CH378 的 INT#引脚，而通过其它方式获知中断，具体参考第 8 章应用说明中的介绍。

6.2. 并行接口

并口信号线包括：8 位双向数据总线 D7~D0、读选通输入引脚 RD#、写选通输入引脚 WR#、片选输入引脚 PCS#以及地址输入引脚 A0。CH378 芯片的 PCS#由地址译码电路驱动，用于当单片机具有多个外围器件时进行设备选择。通过被动并行接口，CH378 芯片可以很方便地挂接到各种 8 位单片机、DSP、MCU 的系统总线上，并且可以与多个外围器件共存。

对于类似 Intel 并口时序的单片机，CH378 芯片的 RD#引脚和 WR#引脚可以分别连接到单片机的读选通输出引脚和写选通输出引脚。对于类似 Motorola 并口时序的单片机，CH378 芯片的 RD#引脚应该接低电平，并且 WR#引脚连接到单片机的读写方向输出引脚 R/-W。

下表为并口 I/O 操作的真值表（表中 X 代表不关心此位，Z 代表 CH378 三态禁止）。

PCS#	WR#	RD#	A0	D7-D0	对 CH378 芯片的实际操作
1	X	X	X	X/Z	未选中 CH378，不进行任何操作
0	1	1	X	X/Z	虽然选中但无操作，不进行任何操作
0	0	1	1	输入	向 CH378 的命令端口写入命令码
0	0	1	0	输入	向 CH378 的数据端口写入数据
0	1	0	0	输出	从 CH378 的数据端口读出数据
0	1	0	1	输出	从 CH378 的命令端口读取接口状态：

6.4. 异步串口

异步串口信号线包括：串行数据输入引脚 RXD 和串行数据输出引脚 TXD。通过串行接口，CH378 可以用最少的连线与单片机、DSP、MCU 进行较远距离的点对点连接。

CH378 芯片的 RXD 和 TXD 可以分别连接到单片机的串行数据输出引脚和串行数据输入引脚。

CH378 的串行数据格式是标准的字节传输模式，包括 1 个起始位、8 个数据位、1 个停止位。

CH378 既支持硬件设定默认的串行通讯波特率，也支持单片机随时通过 `CMD_SET_BAUDRATE` 命令选择合适的通讯波特率。每次上电复位后，CH378 默认的串行通讯波特率由 SDO、SDI、SCK 三个引脚的电平组合设定，参考下表（表中 0 代表低电平，1 代表高电平或者悬空）。

SDO 引脚	SDI 引脚	SCK 引脚	上电复位后默认的串行通讯波特率
1	1	1	9600 bps
1	1	0	57600 bps
1	0	1	115200 bps
1	0	0	460800 bps
0	1	1	1000000 bps
0	1	0	1843200 bps
0	0	1	3000000 bps
0	0	0	921600 bps

为了区分命令码和数据，CH378 要求单片机先通过串口发送两个同步码字节（57H 和 ABH），然后再发送命令码，接着是发送数据或者接收数据。CH378 会检查上述两个同步码字节之间、同步码与命令码之间的间隔时间，如果间隔时间大于串口输入超时时间 `SER_CMD_TIMEOUT`（约 32ms），那么 CH378 将丢弃该同步码及命令包。串口命令操作步骤如下：

- ① 单片机通过串口向 CH378 发出第一同步码 57H；
- ② 单片机向 CH378 发出第二同步码 ABH；
- ③ 单片机向 CH378 发出命令码；
- ④ 如果该命令具有输入数据，那么依次向 CH378 发出输入数据，每次一个字节；
- ⑤ 如果该命令具有输出数据，那么依次从 CH378 接收输出数据，每次一个字节；
- ⑥ 命令完成，有些命令在执行完成后会产生中断通知并通过串口直接发出中断状态码，单片机可以暂停或者转到①继续执行下一个命令。

6.5. 自动演示

CH378 芯片正常上电后，会自动检测 AUTO 引脚的电平，如果检测到 AUTO 引脚为低电平，则进入自动演示模式。以检测芯片是否正常工作、接口模式是否正确、该 SD 卡或者 USB 存储设备是否支持、文件操作是否能正常。

整个自动演示步骤如下：

- ① 给 CH378 芯片上电，插入 SD 卡或者 USB 存储设备，也可以先插入再上电；
- ② 设置 AUTO 引脚为低电平，或将 AUTO 引脚与 GND 短接；
- ③ CH378 芯片 RDY#引脚输出高电平，自动演示开始；
- ④ CH378 进行内部各种初始化，并在 SD 卡或者 USB 存储设备根目录下新建名为“芯片演示.TXT”的文件，写入当前芯片信息（芯片版本、通信接口模式、插入设备、文件系统格式、扇区大小、总容量、剩余容量等），如果初始化失败或自动演示过程中检测到 AUTO 引脚检测到高电平，跳转到⑤；
- ⑤ 芯片自动演示结束，如果自动演示成功则 RDY#引脚输出低电平，如果自动演示失败则 RDY#引脚输出规则方波。
- ⑥ 等待 AUTO 引脚检测到高电平，退出自动演示模式。

一般在自动演示完毕后，建议单片机发送 `CMD_RESET_ALL` 命令或者通过 RSTI 引脚复位 CH378 芯片，然后再进行其它操作。

6.6. 其它硬件

CH378 芯片内部集成了 USB-SIE 和 Phy-I/O、CRC 数据校验、USB-Host 控制器、USB-Device 控制器、SD 卡 SPI-Host 控制器、被动并行接口、SPI-Slave 控制器、异步串口、SRAM、FIFO、高速 MCU、固件程序、晶体振荡器和 PLL 倍频器、电源上电复位电路等。

CH378 芯片的 ACT#引脚用于状态指示输出。在 USB 主机方式下，当 USB 设备断开后，该引脚输出高电平；当 USB 设备连接后，该引脚输出低电平。在 SD 卡主机方式下，当 SD 卡断开后，该引脚输出高电平；当 SD 卡连接后，该引脚输出低电平。CH378 的 ACT#引脚可以外接串了限流电阻的发光二级管 LED，用于指示连接状态。

CH378 芯片的 RDY#引脚用于状态指示输出。在 USB 主机方式下，当 USB 设备初始化成功可以进行文件操作后，该引脚输出低电平。在 SD 卡主机方式下，当 USB 设备初始化成功可以进行文件操作后，该引脚输出低电平。在自动演示方式下，当自动演示开始后，该引脚输出高电平；当自动演示结束并成功时，该引脚输出低电平。

CH378 芯片的 UD+和 UD-引脚是 USB 信号线，可以直接连接到 USB 设备。如果为了芯片安全而串接保险电阻或者电感或者 ESD 保护器件，那么交直流等效串联电阻应该在 5Ω 之内。

CH378 芯片内置了电源上电复位电路，一般情况下，不需要外部提供复位。RST1 引脚用于从外部输入异步复位信号；当 RST1 引脚为低电平时，CH378 芯片被复位；当 RST1 引脚恢复为高电平后，CH378 会继续延时复位 50ms 左右，然后进入正常工作状态。为了在电源上电期间可靠复位并且减少外部干扰，可以在 RST1 引脚与地之间跨接一个容量为 0.1 μ F 左右的电容。

CH378 芯片的 RST 引脚是高电平有效的复位状态输出引脚，可以用于向外部单片机提供上电复位信号。当 CH378 电源上电复位或者被外部强制复位以及复位延时期间，RST 引脚输出高电平；CH378 复位完成并且通讯接口初始化完成后，RST 引脚恢复到低电平。

CH378 芯片正常工作时需要外部为其提供 30MHz 的时钟信号。CH378 芯片内置了晶体振荡器和振荡电容，一般情况下，时钟信号由 CH378 内置的振荡器通过晶体稳频振荡产生，外围电路需要在 X1 和 X0 引脚之间连接一个标称频率为 30MHz 的晶体。

7. 参数

7.1. 绝对最大值（临界或者超过绝对最大值将可能导致芯片工作不正常甚至损坏）

名称	参数说明		最小值	最大值	单位
TA	工作时的环境温度	VCC33=3.3V VCC18=1.8V	-40	85	°C
TS	储存时的环境温度		-55	125	°C
VCC33	电源电压（VCC33 接电源，GND 接地）		-0.4	4.2	V
VCC18	电源电压（VCC18 接电源，GND 接地）		-0.4	2.3	V
VIO	输入或者输出引脚上的电压		-0.4	VCC33+0.4	V
VIO5	支持 5V 耐压的输入或者输出引脚上的电压		-0.4	5.4V	V

7.2. 电气参数（测试条件：TA=25°C，VCC33=3.3V、VCC18=1.8V，不包括连接 USB 总线的引脚）

名称	参数说明		最小值	典型值	最大值	单位
VCCxx	电源电压	VCC33	2.7	3.3	3.6	V
		VCC18	1.65	1.8	1.95	
ICC	工作时的总电源电流	VCC33=3.3V		90	120	mA

ISLP	低功耗状态的电源电流 I/O 引脚悬空/内部下拉	VCC33=3.3V		15		mA
VIL	低电平输入电压		-0.4		0.7	V
VIH	高电平输入电压		2.0		VCC33+0.4	V
VOL	低电平输出电压 (4mA 吸入电流)				0.4	V
VOH	高电平输出电压 (4mA 输出电流)		VCC33-0.4			V
IUP	内置上拉电阻的输入端的输入电流		20	40	100	μA
IDN	内置下拉电阻的输入端的输入电流		-20	-40	-100	μA
VR	电源上电复位的电压门限		1.0	1.5	1.6	V

7.3. 时序参数 (测试条件: TA=25°C, VCC33=3.3V、VCC18=1.8V, 参考附图)

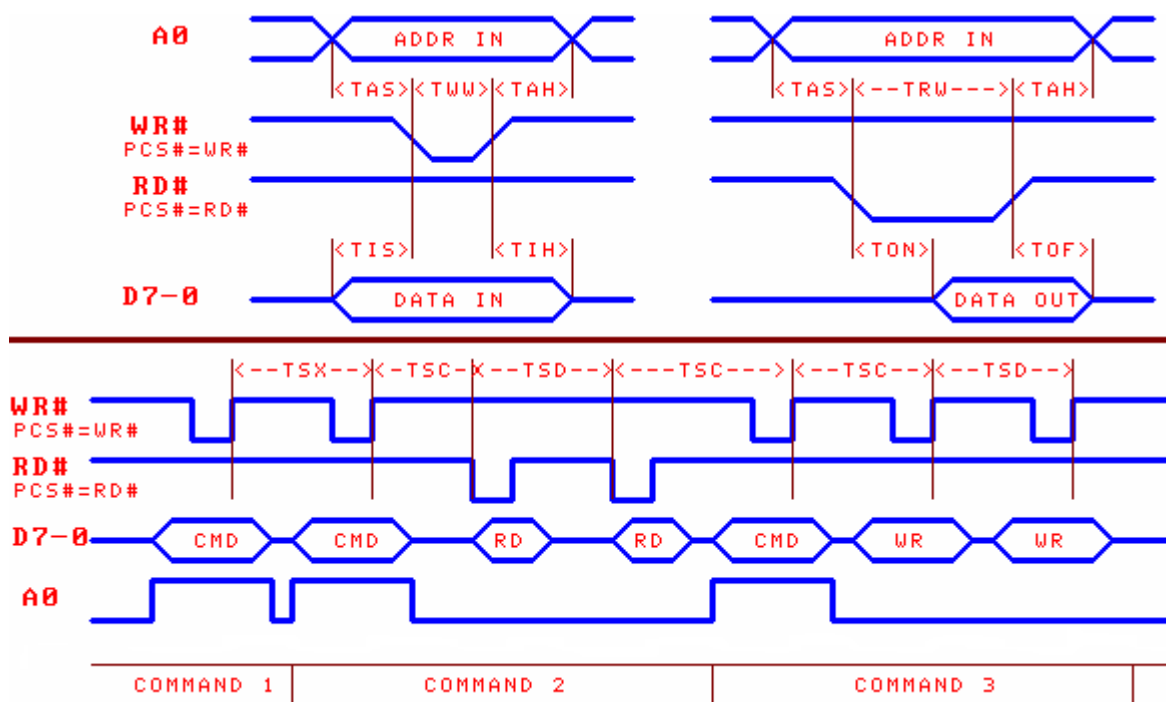
名称	参数说明	最小值	典型值	最大值	单位
FCLK	X1 引脚的输入时钟的频率	29.995	30.00	30.005	MHz
TPR	内部电源上电的复位时间	15	50	60	mS
TRI	外部复位输入的有效信号宽度	100			nS
TRD	外部复位输入后的复位延时	15	50	60	mS
TWAK	从低功耗状态退出的唤醒时间	3	7	15	mS
TE1	CMD_RESET_ALL 命令的执行时间	10	50	60	mS
TE2	CMD_SET_USB_MODE 命令的执行时间	1	35	60	mS
TE3	TEST_CONNECT 命令的执行时间		1	3	mS
TE4	CMD_SET_BAUDRATE 命令的执行时间	200	1000	2000	μS
TE0	其余命令的执行时间		0.8	1.5	μS
TSX	命令码与命令码之间的间隔时间	1.5			μS
TSC	命令码与数据之间的间隔时间	0.6			μS
TSD	数据与数据之间的间隔时间	0.3			μS
TINT	收到 GET_STATUS 命令到 INT#引脚撤消中断		1	2	μS

7.4. 并口时序参数 (测试条件: TA=25°C, VCC33=3.3V、VCC18=1.8V, 参考附图)

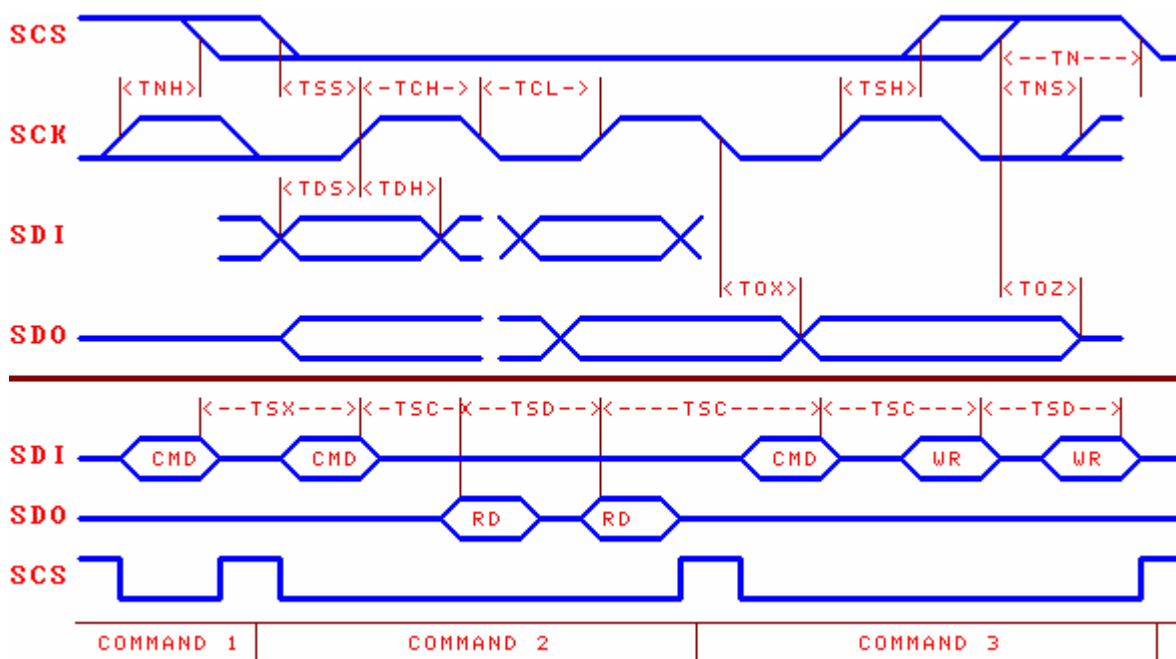
(RD 是指 RD#信号有效并且 PCS#信号有效, RD#=PCS#=0 执行读操作)

(WR 是指 WR#信号有效并且 PCS#信号有效, WR#=PCS#=0 执行写操作)

名称	参数说明	最小值	典型值	最大值	单位
TWW	有效的写选通脉冲 WR 的宽度	30			nS
TRW	有效的读选通脉冲 RD 的宽度	35			nS
TAS	RD 或 WR 前的地址输入建立时间	3			nS
TAH	RD 或 WR 后的地址输入保持时间	3			nS
TIS	写选通 WR 前的数据输入建立时间	0			nS
TIH	写选通 WR 后的数据输入保持时间	3			nS
TON	读选通 RD 有效到数据输出有效	2	7	15	nS
TOF	读选通 RD 无效到数据输出无效	3	12	25	nS



7.5. SPI 时序参数 (测试条件: TA=25°C, VCC33=3.3V、VCC18=1.8V, 参考附图)

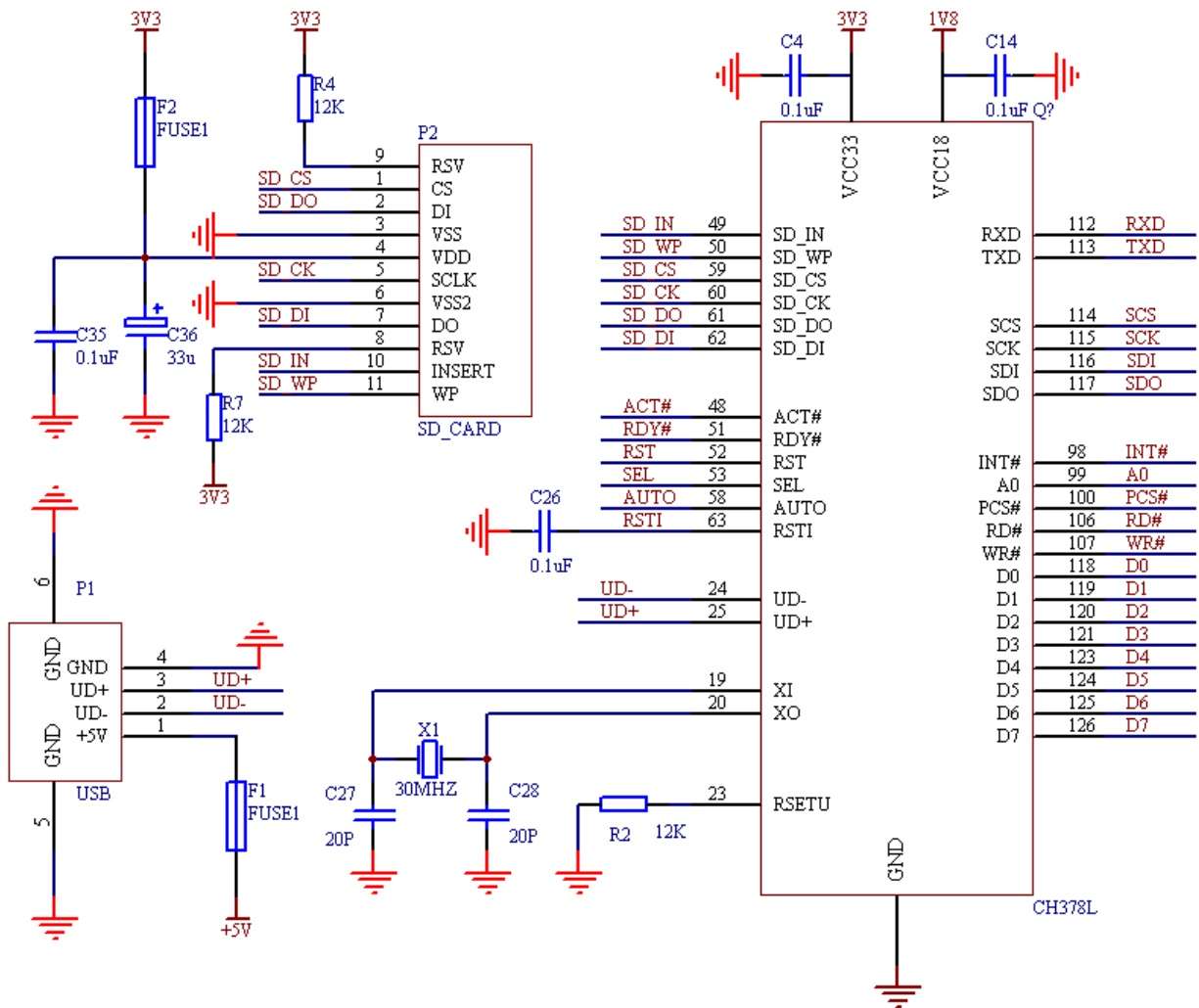


名称	参数说明	最小值	典型值	最大值	单位
TSS	SCK 上升沿之前 SCS 有效的建立时间	25			nS
TSH	SCK 上升沿之后 SCS 有效的保持时间	12			nS
TNS	SCK 上升沿之前 SCS 无效的建立时间	12			nS
TNH	SCK 上升沿之后 SCS 无效的保持时间	12			nS
TN	SCS 无效时间 (SPI 操作间隔时间)	60			nS
TCH	SCK 时钟的高电平时间	15			nS

TCL	SCK 时钟的低电平时间	15			nS
TDS	SCK 上升沿之前 SDI 输入的建立时间	4			nS
TDH	SCK 上升沿之后 SDI 输入的保持时间	2			nS
TOX	SCK 下降沿到 SDO 输出改变	2	6	12	nS
TOZ	SCS 无效到 SDO 输出无效	3	10	18	nS

8、应用

8.1. SD 卡和 U 盘应用 (下图)



如果需要将 CH378 配置为 8 位并口通讯方式 PARALLEL, 那么 SEL 引脚应该接 GND, TXD 引脚悬空。该接口与外围单片机通讯使用的引脚有 A0、PCS#、RD#、WR#、D0-D7、INT#(可选)、RSTI(可选)。

如果需要将 CH378 配置为 SPI 串行通讯方式 SPI, 那么 TXD 引脚应该接 GND, SEL 引脚悬空。该接口与外围单片机通讯使用的引脚有 SCS、SCK、SDO、SDI、INT#(可选)、RSTI(可选)。

如果需要将 CH378 配置为异步串口通讯方式 UART/SERIAL, 那么 SEL 引脚和 TXD 引脚应该都悬空。该接口与外围单片机通讯使用的引脚有 TXD、RXD、INT#(可选)、RSTI(可选)。默认的串口通讯波特率由 SDO、SDI、SCK 三个引脚设定。如果需要动态修改 CH378 串口的通讯波特率, 那么建议由单片机的 I/O 引脚控制 CH378 的 RSTI 引脚, 便于在必要时复位 CH378 以恢复到默认的通讯波特率。

为了节约引脚, 单片机可以不连接 CH378 芯片的 INT#引脚, 获知中断通知的方法如下:

- ① 8 位并口方式下通过查询 CH378 的状态端口（即命令端口）获取接口状态，位 7 是中断标志 `PARA_STATE_INTB`，低有效，等效于查询 `INT#` 引脚，位 7 为 0 时说明有中断请求；位 4 是忙状态标志 `PARA_STATE_BUSY`，高有效；
- ② SPI 接口方式下通过查询 `SDO` 引脚获知中断（上电或者复位后要先通过 `CMD_SET_SDO_INT` 命令设置 `SDO` 引脚在 `SCS` 片选无效时兼做中断请求输出），`SDO` 为低电平时说明有中断请求；
- ③ 串口方式下 CH378 在产生中断通知（`INT#` 变为低电平）的同时，会通过串口直接发出中断状态码，单片机接收到中断状态码说明有中断请求。

P1 为 USB 端口，用来连接 USB 存储设备，USB 总线包括一对 5V 电源线和一对数据信号线。F1 为保险丝，以保护 USB 存储设备。必要时可以串接具有有限流作用的快速电子开关，USB 电源电压必须是 5V。

P2 为 SD 卡插槽，SD 卡通讯使用的 `SCS`、`SCK`、`SDO`、`SDI`、`INSERT` 和 `WP` 引脚连接到 CH378 芯片，其余引脚通过上拉电阻接 3.3V 电源。F2 为保险丝，C35、C36 为 SD 卡供电的退耦电容。

C4 和 C14 为外部电源退耦，可以是容量为 $0.1\mu\text{F}$ 的独石或者高频瓷片电容。CH378 有多组电源引脚，本图中只画了其中一组表示，实际应用中需要全部进行连接。晶体 X1 用于时钟振荡电路，USB-HOST 主机方式要求时钟频率比较准确，晶体 X1 的频率是 $30\text{MHz} \pm 0.4\%$ 。C27 和 C28 为振荡电容，根据晶体 X1 的特性选择，容量在 10pF 到 30pF 之间。

在设计印刷电路板 PCB 时，需要注意：退耦电容 C4 和 C14 尽量靠近 CH378 的相连引脚；使 D+ 和 D- 信号线贴近平行布线，尽量在两侧提供地线或者覆铜，减少来自外界的信号干扰；尽量缩短 X1 和 X0 引脚相关信号线的长度，为了减少高频时钟对外界的干扰，可以在相关元器件周边环绕地线或者覆铜。

8.2. 应用基础

U 盘（或者 SD 卡，下同）提供了若干个物理扇区用于数据存储，每个扇区大小通常是 512 字节。由于计算机通常将 U 盘中的物理扇区组织为 FAT 文件系统，为了方便单片机通过 U 盘或者 SD 卡与计算机之间交换数据，单片机也应该在 FAT 规范下通过文件的形式存取 U 盘中的数据。

一个 U 盘中可以有若干个文件，每个文件都是一组数据的集合，以文件名区分和识别。实际文件数据的存放可能不是连续的，而是通过一组“指针”链接的多个块（也就是分配单元或者簇），从而能够根据需要随时增大文件长度以容纳更多数据。目录（文件夹）是为了便于分类管理，管理者可以人为指定将多个文件归档在一起，例如 2004 年的文件归到一个目录（文件夹）中。

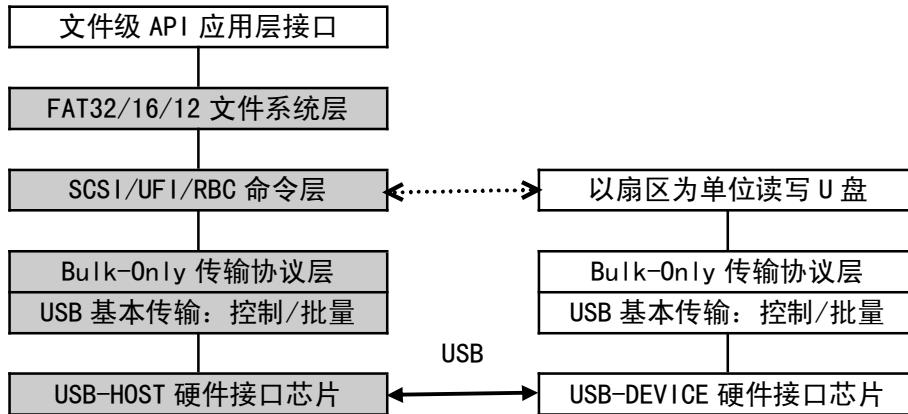
在 FAT 文件系统中，磁盘容量以簇为基本单位进行分配，而簇的大小总是扇区的倍数，所以文件的占用空间总是簇的倍数，也是扇区的倍数。虽然文件占用的空间是簇或者扇区的倍数，但是在实际应用中，保存在文件中的有效数据的长度却不一定是扇区的倍数，所以 FAT 文件系统在文件目录信息 `FAT_DIR_INFO` 中专门记录了当前文件中有效数据的长度，即有效数据的字节数，也就是通常所说的文件长度，文件长度总是小于或者等于文件占用的空间。

在对文件写入数据后，如果是覆盖了原数据，那么文件长度可能不发生变化，当超过原文件长度后，变为追加数据，那么文件长度应该发生变化（增大）。如果向文件追加数据后，没有修改文件目录信息中的文件长度，那么 FAT 文件系统会认为超过文件长度的数据是无效的，正常情况下，计算机无法读出超过文件长度的数据，虽然数据实际存在。

如果数据量少或者数据不连续，那么可以在每次追加数据后立即更新文件目录信息中的文件长度，但是，如果数据量大并且需要连续写入数据，立即更新文件目录信息会降低效率，并且频繁修改文件目录信息也会缩短 U 盘中闪存的使用寿命（因为闪存只能进行有限次擦写），所以在这种情况下，应该在连续写入多组数据后再更新一次文件目录信息中的文件长度，或者一直等到关闭文件时再更新文件长度，`CMD_FILE_CLOSE` 命令可以将内存中的文件长度刷新到 U 盘文件的文件目录信息中。

虽然 CH378 最大支持 4GB 的单个文件，但是为了提高效率，建议单个文件的长度不要超过 100MB，通常在几 KB 到几 MB 范围是比较正常的，数据较多时可以分多个目录，分多个文件存储。

一般情况下，单片机或者嵌入式系统处理 U 盘的文件系统需要实现下图左边的 4 个层次，右边是 U 盘的内部结构层次。由于 CH378 不仅是一个通用的 USB-HOST 硬件接口芯片，还内置了相关的 USB 底层传输固件程序、Bulk-Only 协议传输固件程序、FAT 文件系统管理固件程序，包含了下图左边的 4 个层次（标为灰色部分），所以实际的单片机程序只需要发出文件管理和文件读写命令。



8.3. 快速应用参考步骤

请参考例子程序调用已经将多个命令打包后的子程序。以下步骤使用原始命令码，仅供参考。

8.3.1. 初始化，进行任何一项文件操作之前的必要步骤

- ① 发送 CMD_SET_USB_MODE 命令，进入 USB-HOST 工作方式（模式 6）或者 SD 卡主机工作方式（模式 3）；
- ② 等待 U 盘或者 SD 卡连接，由单片机向 CH378 发送 CMD_DISK_CONNECT 命令定期查询；
- ③ 发送 CMD_DISK_MOUNT 命令，初始化 U 盘或者 SD 卡，并测试磁盘是否就绪，失败后可以重试最多 5 次；
- ④ 上述步骤只需执行一次，除非 U 盘或者 SD 卡断开后重新连接，那么必须回到步骤②；

8.3.2. 顺序读文件

- ① 发送 CMD_SET_FILE_NAME 命令+CMD_FILE_OPEN 命令，打开文件；
- ② 多次发送 CMD_BYTE_READ 命令+CMD_RD_HOST_CUR_DATA 命令，读取数据
- ③ 发送 CMD_FILE_CLOSE 命令，关闭文件，可选操作

8.3.3. 顺序改写文件（覆盖原数据，超过原文件长度后转变为追加数据）

- ① 发送 CMD_SET_FILE_NAME 命令+CMD_FILE_OPEN 命令，打开文件
- ② 多次发送 CMD_WR_HOST_OFS_DATA 命令+CMD_BYTE_WRITE 命令，写入数据
- ③ 发送 CMD_FILE_CLOSE 命令，参数设置为 1，关闭文件并允许自动更新文件长度

8.3.4. 向已有文件追加数据

- ① 发送 CMD_SET_FILE_NAME 命令+CMD_FILE_OPEN 命令，打开文件
- ② 发送 CMD_BYTE_LOCATE 命令，参数是 0FFFFFFFH，移动文件指针到文件末尾
- ③ 多次发送 CMD_WR_HOST_OFS_DATA 命令+CMD_BYTE_WRITE 命令，写入数据
- ④ 发送 CMD_FILE_CLOSE 命令，参数是 1，关闭文件并允许自动更新文件长度

8.3.5. 新建文件并写入数据

- ① 发送 CMD_SET_FILE_NAME 命令+CMD_FILE_CREATE 命令，新建文件
- ② 多次发送 CMD_WR_HOST_OFS_DATA 命令+CMD_BYTE_WRITE 命令，写入数据
- ③ 发送 CMD_FILE_CLOSE 命令，参数是 1，关闭文件并允许自动更新文件长度

8.3.6. 先读文件再改写文件

- ① 发送 CMD_SET_FILE_NAME 命令+CMD_FILE_OPEN 命令，打开文件
- ② 多次发送 CMD_BYTE_READ 命令+CMD_RD_HOST_CUR_DATA 命令，读取数据
- ③ 发送 CMD_BYTE_LOCATE 命令，参数是 0，移动文件指针到文件头部

- ④ 多次发送 CMD_WR_HOST_OFS_DATA 命令+CMD_BYTE_WRITE 命令, 写入数据
- ⑤ 发送 CMD_FILE_CLOSE 命令, 参数是 1, 关闭文件并允许自动更新文件长度

8.3.7. 如果文件已经存在则追加数据, 如果文件不存在则新建文件再写入数据

- ① 发送 CMD_SET_FILE_NAME 命令+CMD_FILE_OPEN 命令, 打开文件, 如果返回 ERR_MISS_FILE 说明文件不存在, 那么转步骤③
- ② 发送 CMD_BYTE_LOCATE 命令, 参数是 OFFFFFFFFFH, 移动文件指针到文件末尾, 然后转步骤④
- ③ 发送 CMD_FILE_CREATE 命令, 新建文件
- ④ 多次发送 CMD_WR_HOST_OFS_DATA 命令+CMD_BYTE_WRITE 命令, 写入数据
- ⑤ 发送 CMD_FILE_CLOSE 命令, 参数是 1, 关闭文件并允许自动更新文件长度

8.3.8. 修改文件名、文件日期/时间、文件长度等文件目录信息, 请参考有关 EXAM10 例子中的说明

- ① 发送 CMD_SET_FILE_NAME 命令+CMD_FILE_OPEN 命令, 打开文件
- ② 发送 CMD_DIR_INFO_READ 命令, 参数是 OFFH, 将文件目录信息读入内存
- ③ 发送 CMD_RD_HOST_REQ_DATA 命令读出原文件目录信息
- ④ 发送 CMD_DIR_INFO_READ 命令, 参数是 OFFH, 将文件目录信息读入内存
- ⑤ 发送 CMD_WR_HOST_OFS_DATA 命令写入新的文件目录信息
- ⑥ 发送 CMD_DIR_INFO_SAVE 命令, 保存文件目录信息
- ⑦ 发送 CMD_FILE_CLOSE 命令, 参数是 0, 关闭文件并禁止自动更新文件长度, 可选操作

8.3.9. 创建子目录(文件夹), 请参考有关 EXAM9 例子中的说明

- ① 发送 CMD_SET_FILE_NAME 命令+CMD_DIR_CREATE 命令, 新建子目录(文件夹)
- ② 发送 CMD_FILE_CLOSE 命令, 参数是 0, 关闭文件并禁止自动更新文件长度

8.3.10. 处理小写文件名和长文件名, 请参考有关 EXAM11 例子中的说明

8.3.11. 搜索和枚举文件名, 全盘枚举所有文件, 请参考有关 EXAM13 例子中的说明