

U 盘和 SD 卡高速文件管理控制芯片 CH378

手册 (二): 辅助命令和 USB 基本传输命令

版本: 4B

<http://wch.cn>

1、附加命令

代码	命令名称 CMD_	输入数据	输出数据	命令用途
以下命令为 USB 主机模式相关辅助命令				
07H	BULK_WR_TEST	输入长度 (2) 数据流 (N)		批量写数据测试
08H	BULK_RD_TEST	输入长度 (2)	数据流 (N)	批量读数据测试
0AH	READ_VAR8	变量地址	数据	读取指定的 8 位 CH378 系统变量
0BH	WRITE_VAR8	变量地址 数据		设置指定的 8 位 CH378 系统变量
0CH	READ_VAR32	变量地址	数据 (4)	读取指定的 32 位 CH378 系统变量
0DH	WRITE_VAR32	变量地址 数据 (4)		设置指定的 32 位 CH378 系统变量
0EH	GET_REAL_LEN		数据 (4)	快速返回上一命令执行实际长度
16H	TEST_CONNECT		连接状态	检测 USB 设备或 SD 卡连接状态
25H	DIRTY_BUFFER			清除内部的磁盘和文件缓冲区
41H	CLR_STALL	端点号	产生中断	控制传输: 清除端点错误
45H	SET_ADDRESS	地址值	产生中断	控制传输: 设置 USB 地址
46H	GET_DESCR	描述符类型	产生中断	控制传输: 获取描述符
49H	SET_CONFIG	配置值	产生中断	控制传输: 设置 USB 配置
4DH	AUTO_SETUP		产生中断	自动配置 USB 设备
4EH	ISSUE_CTRL_TRAN		产生中断	执行控制传输
51H	DISK_INIT		产生中断	初始化 USB 存储器
52H	DISK_RESET		产生中断	复位 USB 存储设备
53H	DISK_SIZE		产生中断	获取 USB 存储器的容量
58H	DISK_INQUIRY		产生中断	查询 USB 存储器特性
59H	DISK_READY		产生中断	检查 USB 存储器就绪
5AH	DISK_R_SENSE		产生中断	检查 USB 存储器错误
5DH	DISK_MAX_LUN		产生中断	获取 USB 存储器设备的最大单元号
60H	SET_LONG_FILE_NAME	以 2 个 0 结束的字符串		设置将要操作的长文件的文件名
61H	GET_LONG_FILE_NAME		产生中断	由完整短文件名路径获取相应长文件名
62H	LONG_FILE_CREATE		产生中断	新建长文件名文件
63H	LONG_DIR_CREATE		产生中断	新建长文件名目录并打开
64H	AUTO_DEMO		产生中断	芯片自动演示
65H	GET_SHORT_FILE_NAME	以 2 个 0 结束的字符串	产生中断	通过长文件名获取相应短文件名路径
66H	LONG_FILE_OPEN	以 2 个 0 结束的字符串	产生中断	通过长文件名打开文件
67H	LONG_FILE_ERASE	以 2 个 0 结束的字符串	产生中断	通过长文件名删除文件

以下命令为 USB 设备模式相关命令				
11H	SET_USB_ID	VID 低字节		设置 USB 的厂商识别码 VID 和产品识别码 PID
		VID 高字节		
		PID 低字节		
		PID 高字节		
13H	SET_USB_SPEED	USB 速度值		设置当前 USB 设备速度
14H	GET_USB_SPEED		USB 速度值	获取当前 USB 设备速度
17H	INIT_ENDPx	端点索引		初始化 USB 端点 (外置固件)
		端点号		
		端点类型		
		端点方向		
		端点包大小高字节		
		端点包大小低字节		
	中断状态码			
18H	SET_INDEXx_IN	端点索引		设置 IN 端点工作方式
		工作方式		
19H	SET_INDEXx_OUT	端点索引		设置 OUT 端点工作方式
		工作方式		
23H	UNLOCK_USB	端点索引		释放当前 USB 缓冲区
29H	RD_USB_DATA	端点索引	数据长度	从指定 USB 端点缓冲区读取数据块，并释放缓冲区
			数据流	
2AH	WR_USB_DATA	端点索引		向指定 USB 端点发送缓冲区写入数据块，并进行数据发送
		数据长度		
		数据流		
2BH	WR_USB_DATA0	数据长度		向 USB 控制端点发送缓冲区写入数据块，并进行数据发送
		数据流		

1.1. CMD_BULK_WR_TEST

该命令用于芯片进行批量数据写测试。首先写入 2 个字节的数据长度，接着按照长度逐个输入后续数据流。

1.2. CMD_BULK_RD_TEST

该命令用于芯片进行批量数据读测试。首先写入 2 个字节的数据长度，然后按照长度逐个读取后续数据流。该命令一般与 CMD_BULK_WR_TEST 命令配合使用，用于测试外部系统与芯片进行大批量数据传输时的正确性。

1.3. CMD_READ_VAR8

该命令用于读取指定的 8 位（单字节）文件系统变量。该命令需要输入 1 个字节的指定变量地址，输出为该变量的数据。

1.4. CMD_WRITE_VAR8

该命令设置指定的 8 位（单字节）文件系统变量。该命令需要输入两个字节数据，分别是指定变

量地址和指定变量数据。

1.5. CMD_READ_VAR32

该命令用于读取指定的 32 位（4 字节）文件系统变量。该命令需要输入 1 个字节的指定变量地址，输出为该变量的数据，变量数据共 4 个字节，依次是数据最低字节、数据较低字节、数据较高字节、数据最高字节。

1.6. CMD_WRITE_VAR32

该命令设置指定的 32 位（4 字节）文件系统变量。该命令需要输入 5 个数据，分别是指定变量地址、变量数据的最低字节、数据较低字节、数据较高字节、数据最高字节。

1.7. CMD_GET_REAL_LEN

该命令用于快速返回上一命令执行完毕后实际返回的长度。输出 4 个字节数据，依次为实际长度的最低字节、实际长度的较低字节、实际长度的较高字节、实际长度的最高字节。比如发送 CMD_BYTE_READ 或 CMD_SEC_READ 命令执行成功产生中断后，可发送该命令来获取实际读取的字节数或扇区数。

1.8. CMD_TEST_CONNECT

该命令用于查询当前 USB 设备或者 SD 卡的连接状态。完成后输出 USB_INT_CONNECT、USB_INT_DISCONNECT 或者 USB_INT_USB_READY 三种状态之一，状态 USB_INT_CONNECT 说明 USB 设备刚连接或者已经连接但尚未初始化，状态 USB_INT_DISCONNECT 说明 USB 设备尚未连接或者已经断开，状态 USB_INT_USB_READY 说明 USB 设备已经连接并且已经被初始化（已经分配 USB 地址），输出 0 说明该命令尚未完成，可以稍后再读取状态。

1.9. CMD_DIRTY_BUFFER

该命令用于主机文件模式下清除内部的磁盘和文件缓冲区。进入主机文件模式后，CH378 总是在内部的磁盘缓冲区中存放一些经常使用的数据，但是有些命令（例如 CMD_RD_DISK_SEC 或者 CMD_WR_DISK_SEC 等）在执行时也会使用该缓冲区，导致缓冲数据无效，为了避免 CH378 误用无效数据，需要在其它命令使用完内部缓冲区后通知 CH378 清除内部缓冲区。

1.10. CMD_CLR_STALL

该命令是清除端点错误的控制传输命令。该命令需要输入 1 个字节数据，指定将被清除错误的 USB 设备的端点号，对于 OUT 端点，有效值是 01H~0FH，对于 IN 端点，有效值是 81H~8FH。该命令用于简化标准 USB 请求 CLEAR_FEATURE，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。

1.11. CMD_SET_ADDRESS

该命令是设置 USB 地址的控制传输命令。该命令需要输入 1 个字节数据，指定新的 USB 设备地址，有效地址是 00H~7FH。该命令用于简化标准 USB 请求 SET_ADDRESS，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。

1.12. CMD_GET_DESCR

该命令是获取描述符的控制传输命令。该命令需要输入 1 个字节数据，指定将要获取的描述符的类型，有效类型是 1 或者 2，分别对应于 DEVICE 设备描述符和 CONFIGURATION 配置描述符，其中，配置描述符还包括接口描述符和端点描述符。该命令用于简化标准 USB 请求 GET_DESCRIPTOR，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，单片机可以通过 CMD_RD_HOST_REQ_DATA 命令获取描述符数据。

1.13. CMD_SET_CONFIG

该命令是设置 USB 配置的控制传输命令。该命令需要输入 1 个字节数据，指定新的 USB 配置值，配置值为 0 则取消配置，否则应该取自该 USB 设备的配置描述符中。该命令用于简化标准 USB 请求 SET_CONFIGURATION，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。

1.14. CMD_AUTO_SETUP

该命令用于自动配置 USB 设备，不支持 SD 卡。该命令用于简化普通 USB 设备的初始化步骤，相当于 GET_DESCR、SET_ADDRESS、SET_CONFIGURATION 等多个命令序列。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。

1.15. CMD_ISSUE_CTRL_TRAN

该命令使能 CH378 执行 USB 控制传输。发起该命令前，必须先通过 CMD_WR_HOST_OFS_DATA 命令将 8 个字节的 SETUP 包写入到内部缓冲区中。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。如果操作状态是 ERR_SUCCESS，则说明命令执行成功，否则说明命令执行失败，单片机可以根据操作状态进一步分析失败原因。

对于发送数据的 OUT 事务，需要先通过 CMD_WR_HOST_OFS_DATA 命令将要发送的后续数据包，紧接着 8 个字节的 SETUP 包写入到内部缓冲区中；对于接收数据的 IN 事务，执行成功后，再通过 CMD_RD_HOST_REQ_DATA 命令读出已经接收的数据。

1.16. CMD_DISK_INIT

该命令用于初始化 USB 存储设备，不支持 SD 卡。执行该命令前需要先对 USB 设备进行枚举，如果该设备是 USB 存储设备，可以通过该命令直接对 USB 存储设备进行初始化。初始化成功之后，即可对 USB 存储设备进行读写操作。CH378 在命令执行完成后向单片机请求中断，如果 USB 设备已经断开，那么中断状态可能是 USB_INT_DISCONNECT；如果 USB 设备不能识别或者该 USB 存储设备不被支持，那么中断状态通常是 USB_INT_DISK_ERR；如果 USB 存储设备初始化成功，那么中断状态将是 ERR_SUCCESS。

1.17. CMD_DISK_RESET

该命令用于通过控制传输复位 USB 存储设备，不支持 SD 卡。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。

完整的复位过程包括：通过该命令复位 USB 存储设备，通过 CLR_STALL 命令复位 Bulk-IN 端点，通过 CLR_STALL 命令复位 Bulk-OUT 端点。

当 USB 存储设备发生错误时，CH378 会分析错误原因并根据需要自动选择是否复位 USB 设备。

1.18. CMD_DISK_SIZE

该命令用于获取 USB 存储器的容量，不支持 SD 卡。在成功初始化 USB 存储设备后，该命令可以获取 USB 存储设备的总容量。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, 那么可以由 CMD_RD_HOST_REQ_DATA 命令获取数据, 数据通常是 8 个字节, 前 4 个字节以高字节在前组成的双字数据是 USB 存储设备的总扇区数, 后 4 个字节以高字节在前组成的双字数据是每个扇区的字节数, 两个数据相乘的结果就是以字节为单位的 USB 存储设备的总容量。

1. 19. CMD_DISK_INQUIRY

该命令用于查询 USB 存储设备的特性, 不支持 SD 卡。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, 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。该命令一般不需要用到, 除非是分析新的逻辑单元。

1. 20. CMD_DISK_READY

该命令用于检查 USB 存储设备是否就绪, 不支持 SD 卡。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, 那么说明 USB 存储设备当前已经就绪。

1. 21. CMD_DISK_R_SENSE

该命令用于检查 USB 存储设备的错误。CH378 在命令执行完成后向单片机请求中断, 正常情况下中断状态是 ERR_SUCCESS, 可以由 CMD_RD_HOST_REQ_DATA 命令获取数据后分析错误。

1. 22. CMD_DISK_MAX_LUN

该命令通过控制传输获取 USB 存储设备的最大逻辑单元号。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, 那么可以由 CMD_RD_HOST_REQ_DATA 命令获取数据, 数据通常是 1 个字节。

1. 23. CMD_SET_LONG_FILE_NAME

该命令用于设置将要操作的长文件的文件名或者目录名。输入数据是以 2 个 0 结束的字符串, 长度不得超过 520 个字节。

1. 24. CMD_GET_LONG_FILE_NAME

该命令用于由完整短文件名路径(可以是文件或文件夹)得到相应的长文件名。在发起该命令之前, 需要先通过 CMD_SET_FILE_NAME 命令设置短文件名完整的路径名。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, 那么可以由 CMD_RD_HOST_REQ_DATA 命令获取数据。输出数据依次为 2 个字节的长文件名总长度(最大为 520 字节)、后续长文件名字符。

1. 25. CMD_LONG_FILE_CREATE

该命令用于新建长文件名文件, 如果文件已经存在那么先删除再新建。在发起该命令之前, 需要先通过 CMD_SET_FILE_NAME 命令设置短文件名完整的路径名, 再通过 CMD_SET_LONG_FILE_NAME 命令设置相应长文件名。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, 那么说明长文件名文件新建成功。

1. 26. CMD_LONG_DIR_CREATE

该命令用于新建长文件名目录, 如果目录已经存在那么先删除再新建。在发起该命令之前, 需要先通过 CMD_SET_FILE_NAME 命令设置短目录名完整的路径名, 再通过 CMD_SET_LONG_FILE_NAME 命令

设置相应长文件名。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，那么说明长文件名目录新建成功。

1.27. CMD_AUTO_DEMO

该命令用于芯片进行自动演示功能。CH378 在自动演示过程中不能继续接收命令码，整个自动演示步骤如下：

- ① 芯片 RDY#引脚输出高电平，自动演示开始；
- ② CH378 进行内部各种初始化；
- ③ 检测 SD 卡或者 USB 存储设备是否插入，优先选择 SD 卡；
- ④ 初始化 SD 卡或者 USB 存储设备，如果初始化失败，跳转到⑥；
- ⑤ 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）；
- ⑥ 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。

一般在 CMD_CHECK_EXIST 命令检测到通讯连接正常之后，即可发送自动演示命令。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，那么说明自动演示成功。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。

1.28. CMD_GET_SHORT_FILE_NAME

该命令用于由短文件名所在文件路径及完整的长文件名获取到相应的短文件名及路径。在发起该命令之前，需要先通过 CMD_SET_FILE_NAME 命令设置短文件名的所在路径名，再通过 CMD_SET_LONG_FILE_NAME 命令设置相应长文件名。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，那么说明短文件名及路径获取成功，可以通过 CMD_RD_HOST_REQ_DATA 命令获取数据。输出数据依次为 2 个字节的短文件名总长度（最大为 300 字节）、后续短文件名字符。

1.29. CMD_LONG_FILE_OPEN

该命令用于由短文件名所在文件路径及完整的长文件名打开对应的文件。在发起该命令之前，需要先通过 CMD_SET_FILE_NAME 命令设置短文件名的所在路径名，再通过 CMD_SET_LONG_FILE_NAME 命令设置相应长文件名。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，那么说明文件打开成功。

1.30. CMD_LONG_FILE_ERASE

该命令用于由短文件名所在文件路径及完整的长文件名删除对应的文件。在发起该命令之前，需要先通过 CMD_SET_FILE_NAME 命令设置短文件名的所在路径名，再通过 CMD_SET_LONG_FILE_NAME 命令设置相应长文件名。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，那么说明文件删除成功。

1.31. CMD_SET_USB_ID

该命令设置 USB 的厂商识别码 Vendor-ID 和产品识别码 Product-ID。该命令需要输入 4 个数据，依次为厂商 VID 的低 8 位、厂商 VID 的高 8 位，产品 VID 的低 8 位、产品 VID 的高 8 位。如果需要设置 ID，那么该命令必须在 SET_USB_MODE 命令之前执行。

1.32. CMD_SET_USB_SPEED

该命令设置当前 USB 设备传输速度，需要输入 1 个字节的 USB 速度码，0x00 表示全速，0x02 表

示高速，其它值无效。如果需要设置 USB 传输速度，那么该命令必须在 SET_USB_MODE 命令之前执行，否则默认配置为高速设备。

1.33. CMD_GET_USB_SPEED

该命令获取当前 USB 设备传输速度，返回 1 个字节数据，该数据为当前 USB 设备实际速度码，0x00 表示全速，0x02 表示高速，其它值无效。外置固件模式一般需要在获取设备描述符时发送该命令，以便根据实际速度重新配置端点大小等信息。

1.34. CMD_INIT_ENDPx

该命令初始化 USB 端点，仅用于外置固件模式。该命令需要输入 7 个字节数据，依次为端点索引、端点号、端点类型、端点方向、端点包大小高字节、端点包大小低字节、中断状态码。如果需要初始化端点，那么该命令必须在 SET_USB_MODE 命令之前执行。

外置固件模式下，默认 CH378 的 4 个 USB 端点分别配置成：端点 1 中断上传端点 (64 字节)、端点 1 批量上传端点 (512 字节)、端点 2 批量上传端点 (512 字节) 和端点 2 批量上传端点 (512 字节)。根据实际需要可以将 4 个端点分别配置成中断或批量端点，上传或上传端点。即最大可以配置成 4 个上传端点或 4 个上传端点。如果需要重新初始化端点，那么必须同时初始化全部 4 个端点。

1.35. CMD_SET_INDEXx_IN

该命令设置 IN 端点工作方式，需要输入 2 个字节数据，依次为端点索引和工作方式。例如，如果 USB 设备不支持 USB 标准请求 SET_INTERFACE，那么在接收到请求后，可以通过该命令设置端点 0 的发送器的事务响应方式，使其对 IN 事务返回 STALL，相应的工作方式字节为 0x40。通常情况下，该命令在 2uS 时间内完成。

IN 端点工作方式

工作方式字节	名称	工作方式位说明
位 7	复位同步位	如果该位为 1，则复位该端点的同步位
位 6	STALL 设置位	如果该位为 1，则设置该端点响应方式为 STALL
位 5	STALL 清除位	如果该位为 1，则清除该端点之前的 STALL 特性
位 4-位 0	无效	其它情况无数据上传，则自动应答 NAK

1.36. CMD_SET_INDEXx_OUT

该命令设置 OUT 端点工作方式，需要输入 2 个字节数据，依次为端点索引和工作方式。例如，如果端点 1 的接收器发生错误，那么可以通过该命令设置端点 1 的接收器的事务响应方式，使其对 OUT 事务返回 STALL，相应的工作方式字节为 0x40。通常情况下，该命令在 2uS 时间内完成。

OUT 端点工作方式

工作方式字节	名称	工作方式位说明
位 7	复位同步位	如果该位为 1，则复位该端点的同步位
位 6	STALL 设置位	如果该位为 1，则设置该端点响应方式为 STALL
位 5	STALL 清除位	如果该位为 1，则清除该端点之前的 STALL 特性
位 4-位 0	无效	其它情况接收到数据下传，则自动应答 ACK

1.37. CMD_UNLOCK_USB

该命令释放当前 USB 下传端点缓冲区，需要输入一个字节端点索引。CH378 的下传端点成功接收到 USB 主机下传的数据包后，为防止缓冲区覆盖，在向单片机请求中断前首先锁定当前缓冲区，暂停该端点的 USB 通讯，直到单片机通过 UNLOCK_USB 命令释放当前端点缓冲区，或者通过 RD_USB_DATA

命令读取数据后才会释放当前端点缓冲区。该命令不能多执行，也不能少执行。

1.38. CMD_RD_USB_DATA

该命令从当前 USB 中断的端点缓冲区中读取数据块并释放当前缓冲区。该命令需要输入 1 个字节端点索引。首先读取的输出数据是数据块长度，也就是后续数据流的字节数，占 2 个字节（低字节在前），有效值为 0 至 512。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。数据读取完毕后，CH378 自动释放当前 USB 端点缓冲区，从而可以继续接收 USB 主机发来的数据。

1.39. CMD_WR_USB_DATA

该命令向指定 USB 端点发送缓冲区写入数据块，并进行数据发送。该命令需要输入若干个字节数据，首先写入的输入数据是端点索引，其次写入的输入数据是数据长度，也就是后续数据流的字节数，占 2 个字节（低字节在前），有效值为 0 至 512。如果长度不为 0，则单片机必须将后续数据逐个写入 CH378。

1.40. CMD_WR_USB_DATA0

该命令向 USB 控制端点发送缓冲区写入数据块，并进行数据发送。该命令需要输入若干个字节数据，首先写入的输入数据是数据长度，也就是后续数据流的字节数，占 1 个字节，有效值为 0 至 64。如果长度不为 0，则单片机必须将后续数据逐个写入 CH378。

2、USB 设备模式

CH378 内置了 USB 通讯中的底层协议，具有省事的内置固件模式和灵活的外置固件模式。在内置固件模式下，CH378 自动处理默认端点 0（控制端点）的所有事务，本地端单片机只要负责数据交换，所以单片机程序非常简洁。在外置固件模式下，由外部单片机根据需要自行处理各种 USB 请求，从而实现符合各种 USB 类规范的设备。

CH378 芯片内部集成了 PLL 倍频器、USB 接口 SIE、数据缓冲区、命令解释器、通用的固件程序等主要部件。USB 接口 SIE 用于完成物理的 USB 数据接收和发送，自动处理位跟踪和同步、NRZI 编码和解码、位填充、并行数据与串行数据之间的转换、CRC 数据校验、事务握手、出错重试、USB 总线状态检测等。数据缓冲区用于缓冲 USB 接口 SIE 收发的数据。命令解释器用于分析自动处理单片机/DSP/MCU 提交的各种命令。通用的固件程序用于自动处理 USB 默认端点 0 的各种标准事务等。

CH378 芯片内部具有 5 个物理端点，以端点索引命名，分别为索引号 0 端点、索引号 1 端点、索引号 2 端点、索引号 3 端点和索引号 4 端点。外置固件模式下，如果没有通过发送 CMD_INIT_ENDPx 重新初始化端点，则各个端点功能和内置固件相同。

CH378 向单片机请求中断后，单片机通过 CMD_GET_STATUS 命令获取中断状态，中断状态分析如下：

中断状态字节	名称	中断状态分析说明
位 7-位 5	保留	总是 000
位 4-位 3	当前事务	00=OUT 事务 10=IN 事务 11=SETUP 事务
位 2-位 0	当前端点	000=0 号端点 001=1 号端点 010=2 号端点 011=3 号端点 100=4 号端点 101=USB 挂起

110=USB 总线复位

下面为中断状态值说明。在内置固件模式的 USB 设备方式下，单片机只需要处理表中标注为灰色的中断状态，CH378 内部自动处理了其它中断状态。

中断状态值	状态名称	中断原因分析说明
0x18	USB_INT_EPO_SETUP	控制端点的接收器接收到数据，SETUP 成功
0x00	USB_INT_EPO_OUT	控制端点的接收器接收到数据，OUT 成功
0x10	USB_INT_EPO_IN	控制端点的发送器发送完数据，IN 成功
0x01	USB_INT_INDEX1_OUT	索引号 1 端点的接收器接收到数据，OUT 成功
0x11	USB_INT_INDEX1_IN	索引号 1 端点的发送器发送完数据，IN 成功
0x02	USB_INT_INDEX2_OUT	索引号 2 端点的接收器接收到数据，OUT 成功
0x12	USB_INT_INDEX2_IN	索引号 2 端点的发送器发送完数据，IN 成功
0x03	USB_INT_INDEX3_OUT	索引号 3 端点的接收器接收到数据，OUT 成功
0x13	USB_INT_INDEX3_IN	索引号 3 端点的发送器发送完数据，IN 成功
0x04	USB_INT_INDEX4_OUT	索引号 4 端点的接收器接收到数据，OUT 成功
0x14	USB_INT_INDEX4_IN	索引号 4 端点的发送器发送完数据，IN 成功
0x05	USB_INT_BUS_SUSP	USB 总线挂起事件
0x06	USB_INT_BUS_RESET	监测到 USB 总线复位
0x07	USB_INT_SET_CONFIG	监测到 USB 设备接收到 SET_CONFIG 命令

注：对于 USB 设备中断：USB_INT_BUS_SUSP、USB_INT_BUS_RESET 和 USB_INT_SET_CONFIG，芯片通知外部系统后，会自动进行清除。

2.1. 内部固件说明

2.1.1. 索引号 0 端点

索引号 0 端点即控制端点，该端点为双向端的，支持上传和下传，端点最大包大小为 64 字节。

当 CH378 成功完成 SETUP 事务后，CH378 将自动设置该端点的接收器和发送器的同步触发标志为 1，并自动处理 SETUP 包。

当 CH378 成功完成 OUT 事务后，CH378 将自动触发该端点的接收器的同步触发标志，由 0 触发为 1，由 1 触发为 0。

当 CH378 成功完成 IN 事务后，CH378 将自动触发该端点的发送器的同步触发标志，由 0 触发为 1，由 1 触发为 0。

2.1.2. 索引号 1 端点

该端点为中断上传端点，端点号为 0x81，端点最大包大小为 64 字节。

当 CH378 成功完成端点的 IN 事务后，CH378 将自动触发该端点的发送器的同步触发标志，由 0 触发为 1，由 1 触发为 0。然后以 USB_INT_INDEX1_IN 为中断状态通知外部单片机。

2.1.3. 索引号 2 端点

该端点为批量下传端点，端点号为 0x01，端点最大包大小为 512 字节。

当 CH378 成功完成端点的 OUT 事务后，CH378 将自动触发该端点的接收器的同步触发标志，由 0 触发为 1，由 1 触发为 0。然后以 USB_INT_INDEX2_OUT 为中断状态通知外部单片机。

2.1.4. 索引号 3 端点

该端点为批量上传端点，端点号为 0x82，端点最大包大小为 512 字节。

当 CH378 成功完成端点的 IN 事务后，CH378 将自动触发该端点的发送器的同步触发标志，由 0 触发为 1，由 1 触发为 0。然后以 USB_INT_INDEX3_IN 为中断状态通知外部单片机。

2.1.5. 索引号 4 端点

该端点为批量下传端点，端点号为 0x02，端点最大包大小为 512 字节。

当 CH378 成功完成端点的 OUT 事务后，CH378 将自动触发该端点的接收器的同步触发标志，由 0 触发为 1，由 1 触发为 0。然后以 USB_INT_INDEX4_OUT 为中断状态通知外部单片机。

2.1.6. 参考流程

在 CH378 评估板资料中提供了内置固件模式下的单片机程序，下述流程供外部单片机处理请求时参考。

(一)、单片机启动后，首先初始化 CH378 为使用内部固件的 USB 设备方式，然后设置中断。

(二)、当单片机收到中断后，使用 GET_STATUS 命令获取中断状态，分析处理如下：

(1)、如果是索引号 1 端点或索引号 3 端点的 IN 事务成功，则通知主程序继续，比如通过 CMD_WR_USB_DATA 命令写入下一包数据进行上传。

(2)、如果是索引号 2 端点或索引号 4 端点的 OUT 事务成功，则使用 CMD_RD_USB_DATA 命令读取数据再通知主程序处理。

(3)、如果是 USB 总线复位，CH378 会自动清除 USB 地址和同步触发标志等。

2.2. 外部固件说明

2.2.1. 索引号 0 端点

索引号 0 端点即控制端点，该端点为双向端的，支持上传和下传，端点最大包大小为 64 字节。

当 CH378 成功完成 SETUP 事务后，CH378 将自动设置该端点的接收器和发送器的同步触发标志为 1，然后以 USB_INT_EPO_SETUP 为中断状态通知外部单片机读取 SETUP 数据并处理。

当 CH378 成功完成 OUT 事务后，CH378 将自动触发该端点的接收器的同步触发标志，由 0 触发为 1，由 1 触发为 0，然后以 USB_INT_EPO_OUT 为中断状态通知外部单片机读取数据并处理。

当 CH378 成功完成 IN 事务后，CH378 将自动触发该端点的发送器的同步触发标志，由 0 触发为 1，由 1 触发为 0，然后以 USB_INT_EPO_IN 为中断状态通知外部单片机继续处理。

2.2.2. 索引号 1 端点

该端点为单向端点，可以配置成上传或下传，端点号有效范围为 1 至 8，端点最大包大小为 512 字节。

该端点可以通过 CMD_INIT_ENDPx 命令根据实际需要灵活配置，如果未配置则默认为中断上传端点，端点号为 0x81，端点最大包大小为 64 字节，产生的中断状态为 USB_INT_INDEX1_IN。

2.2.3. 索引号 2 端点

该端点为单向端点，可以配置成上传或下传，端点号有效范围为 1 至 8，端点最大包大小为 512 字节。

该端点可以通过 CMD_INIT_ENDPx 命令根据实际需要灵活配置，如果未配置则默认为批量下传端点，端点号为 0x01，端点最大包大小为 512 字节，产生的中断状态为 USB_INT_INDEX2_OUT。

2.2.4. 索引号 3 端点

该端点为单向端点，可以配置成上传或下传，端点号有效范围为 1 至 8，端点最大包大小为 512 字节。

该端点可以通过 CMD_INIT_ENDPx 命令根据实际需要灵活配置，如果未配置则默认为批量上传端点，端点号为 0x82，端点最大包大小为 512 字节，产生的中断状态为 USB_INT_INDEX3_IN。

2.2.5. 索引号 4 端点

该端点为单向端点，可以配置成上传或下传，端点号有效范围为 1 至 8，端点最大包大小为 512 字节。

该端点可以通过 CMD_INIT_ENDPx 命令根据实际需要灵活配置，如果未配置则默认为批量下传端点，端点号为 0x02，端点最大包大小为 512 字节，产生的中断状态为 USB_INT_INDEX4_OUT。

2.2.6. 参考流程

在 CH378 评估板资料中提供了外置固件模式下的单片机程序，下述流程供外部单片机处理请求时参考。

(一)、单片机启动后，首先初始化 CH378 为使用外部固件的 USB 设备方式，根据需要可以重新配置 CH378 的 4 个索引端点，该操作必须在发送 SET_USB_MODE 命令之前，并且必须同时配置 4 个索引端点，然后设置中断。

(二)、当单片机收到中断后，使用 GET_STATUS 命令获取中断状态，分析处理如下：

- (1)、如果是索引号 x(有效值为 1-4) 端点的 IN 事务成功，则通知主程序继续，比如通过 CMD_WR_USB_DATA 命令写入下一包数据进行上传。
- (2)、如果是索引号 x(有效值为 1-4) 端点的 OUT 事务成功，则使用 CMD_RD_USB_DATA 命令读取数据再通知主程序处理。
- (3)、如果是索引号 0(控制端点)的 SETUP 成功，则使用 CMD_RD_USB_DATA 命令读取数据，分析处理如下。
 - (a)、如果是 USB 请求 CLEAR_FETURE，那么根据请求中的 FETURE 以及端点号分析处理，对于需要返回 STALL 的请求，可以通过 SET_INDEXx_IN 命令设置。
 - (b)、如果是 USB 请求 GET_DESCRIPTOR，那么使用 CMD_WR_USB_DATA0 命令返回全部描述符或描述符的前 64 个字节，并保存该 USB 请求以及当前描述符计数，以便后面继续返回。
 - (c)、如果是 USB 请求 SET_ADDRESS，那么保存主机所设置的 USB 地址值，不需要做其它处理，该命令由 CH378 芯片内部自动处理。
 - (d)、如果是 USB 请求 SET_CONFIG，那么保存设置值，通知主程序 USB 初始化成功与否。
 - (e)、如果是 USB 请求 GET_CONFIG，那么使用 CMD_WR_USB_DATA0 命令返回当前配置值。
 - (f)、如果是 USB 请求 GET_INTERFACE，那么使用 CMD_WR_USB_DATA0 命令返回当前接口值。
 - (g)、如果是 USB 请求 GET_STATUS，那么使用 CMD_WR_USB_DATA0 命令返回当前状态值。
 - (h)、其余 USB 请求根据需要处理，不支持则使用 SET_INDEXx_IN 命令设置应答为 STALL。
- (4)、如果是索引号 0(控制端点)的 OUT 成功，则使用 CMD_RD_USB_DATA 命令读取数据，可以放弃数据。
- (5)、如果是索引号 0(控制端点)的 IN 成功，如果之前 USB 请求是 GET_DESCRIPTOR，那么使用 CMD_WR_USB_DATA0 命令返回剩余描述符。
- (6)、如果是 USB 总线复位，CH378 会自动清除 USB 地址和同步触发标志等。