

CH375 Chinese Manual (2)

1

## USB bus interface chip CH375

Chinese manual (2): USB basic transfer command

Version: 4

[Http://wch.cn](http://wch.cn)

## 1, additional commands

Code	Command name	Input data	Output Data	Command purpose
04H	SET_USB_SPEED	Bus speed		Set the USB bus speed
0AH	GET_DEV_RATE	Data 07H	Data rate	Get the data rate type of the USB device
0BH	SET_RETRY	Data 25H number of retries		Set the number of retries for USB transaction operations
0FH	DELAY_100US		Delay state	Delay 100uS
13H	SET_USB_ADDR	Address value		Set the USB address
1CH	SET_ENDP6	Way of working	(etc 3uS)	Set the receiver of the USB host endpoint
1DH	SET_ENDP7	Way of working	(etc 3uS)	Set the sender of the USB host endpoint
27H	RD_USB_DATA0		Data length data flow	Endpoint buffer from current USB interrupt Read data block
41H	CLR_STALL	Endpoint number	Generate an interruption	Control Transfer: Clear Endpoint Error
45H	SET_ADDRESS	Address value	Generate an interruption	Control Transfer: Set USB Address
46H	GET_DESCR	Descriptor type	Generate an interruption	Control Transfer: Get Descriptor
49H	SET_CONFIG	Configuration value	Generate an interruption	Control Transfer: Set USB Configuration
4DH	AUTO_SETUP		Generate an interruption	Automatically configure USB devices
4EH	ISSUE_TKN_X	Sync flag Transaction attribute	Generate an interruption	Issue a synchronization token to execute the transaction
4FH	ISSUE_TOKEN	Transaction attribute	Generate an interruption	Issue a token, execute a transaction
50H	DISK_BOC_CMD		Generate an interruption	Command to execute the BulkOnly transport protocol
52H	DISK_RESET		Generate an interruption	Reset USB storage device
5DH	DISK_MAX_LUN		Generate an interruption	Set the maximum unit number of the USB storage device

If the input data is the way the transceiver of the USB endpoint works, refer to the table below.

Working mode byte	name	Bit analysis of working methods
		If bit 7 is 1, then bit 6 is the new sync trigger flag:
Bit 7 to 6	Synchronous trigger flag	00 or 01=Keep the current sync trigger flag unchanged 10=Sync trigger flag is set to 0    11=Sync trigger flag set 1
Bit 5 to 4	(reserved bit)	(undefined, must be 0)
Bit 3 to 0	Transaction response	Must be 0000

## 1.1. Command SET\_USB\_SPEED

This command sets the USB bus speed (this feature is not supported on some models). This command requires inputting 1 data for selection. Select USB bus speed, 00H corresponds to 12Mbps full speed mode, and 02H corresponds to 1.5Mbps low speed mode. USB bus speed of CH375 The default is 12Mbps full speed mode, and it will automatically recover after setting the USB working mode by executing the SET\_USB\_MODE command. 12Mbps full speed mode.

## 1.2. Command GET\_DEV\_RATE

This command gets the data rate type of the currently connected USB device. This command needs to input 1 data 07H, and the output is data speed.

CH375 Chinese Manual (2)

2

The rate type, whose bit 4 is 1, is a 1.5Mbps low-speed USB device, otherwise it is a 12Mbps full-speed USB device. This command is only in the USB mode It is valid in the state of Equation 5 (enabled USB host mode, no SOF packet is generated).

### 1.3. Command SET\_RETRY

This command sets the number of retries for USB transaction operations. This command requires two data inputs, data 25H and number of retries.

Bits 7 and 6 of the number of retries specify how the CH375 will be processed when it receives a NAK response. Bit 7 is 1 and Bit 6 is 0.

Try (you can temporarily abandon the current retry with the ABORT\_NAK command), bit 7 is 1 and bit 6 is 1 and limited retry 200mS to 2S left

Right, bit 7 is 0, the NAK is notified to the microcontroller as a result or treated as an error. Bit 5 to bit 0 of the number of retries specified when USB is set

The number of retries for CH375 after the standby response times out. If it is 0, it will not be retried after timeout.

The default number of retries after the chip is reset or after resetting the USB mode is 85H, so it will be retried infinitely after receiving the NAK response.

The USB device will retry 5 times after the response times out.

### 1.4. Command DELAY\_100US

This command is used to delay 100uS and only supports the parallel port mode. Output data 0 at the parallel port during the delay, and output the number of parallel ports after the delay. According to the non-zero (usually the chip version number), the MCU determines whether the delay is over based on the read data.

### 1.5. Command SET\_USB\_ADDR

This command sets the USB device address. This command requires inputting 1 data to select the address of the USB device being operated. Reset

After the USB device is connected or disconnected, the USB device address is always 00H, and the MCU communicates with the USB device through the default address 00H.

If the address of the USB device is set via a standard USB request, the same USB device address must also be set with this command.

The CH375 communicates with the USB device via the new address.

### 1.6. Command SET\_ENDP6

This command sets the USB host endpoint or the receiver of endpoint 2. This command requires you to enter 1 data and specify a new way of working.

For example, if you execute an IN transaction and want to receive DATA0 and discard DATA1, you must set the host endpoint's receive with this command.

The synchronization trigger flag of the device is 0, and the corresponding working mode byte is 80H. Normally, this command is completed within 3uS.

### 1.7. Command SET\_ENDP7

This command sets the USB host endpoint or the sender of endpoint 2. This command requires you to enter 1 data and specify a new way of working.

For example, if you execute a SETUP or OUT transaction and want to send DATA0, you must set the sender of the host endpoint with this command.

The synchronization trigger flag is 0, and the corresponding working mode byte is 80H. If you want to send DATA1, the working mode byte is C0H. through

Normally, this command is completed within 3uS.

### 1.8. Command RD\_USB\_DATA0

This command reads the data block from the endpoint buffer of the current USB interrupt. In USB host mode, this command is in RD\_USB\_DATA

The function of the command is identical, the only difference is that the command is slightly more efficient.

### 1.9. Command CLR\_STALL

This command is a control transfer command that clears the endpoint error. This command requires inputting 1 data and specifying the USB settings that will be cleared.

The endpoint address of the standby address is 01H to 0FH for the OUT endpoint and 81H to 8FH for the IN endpoint. The order

Used to simplify the standard USB request CLEAR\_FEATURE, CH375 requests an interrupt to the microcontroller after the command is executed, the microcontroller can read

Take the interrupt status as the operational status of the command. If the operation status is USB\_INT\_SUCCESS, the command execution is successful, otherwise

The command execution failed.

### 1.10. Command SET\_ADDRESS

This command is a control transfer command that sets the USB address. This command requires inputting 1 data and specifying a new USB device address.

The effective address is 00H to 7FH. This command is used to simplify the standard USB request SET\_ADDRESS, CH375 to the microcontroller after the command is executed

Requesting an interrupt, the microcontroller can read the interrupt status as the operating state of the command. If the operational status is USB\_INT\_SUCCESS, then

The command execution succeeds. Otherwise, the command execution fails.

### 1.11. Command GET\_DESCR

This command is a control transfer command that gets the descriptor. This command requires inputting 1 data, specifying the type of descriptor to be obtained.

Valid types are 1 or 2, corresponding to the DEVICE device descriptor and the CONFIGURATION configuration descriptor, respectively.

The descriptor also includes an interface descriptor and an endpoint descriptor. This command is used to simplify the standard USB request GET\_DESCRIPTOR, CH375 in the command

After the execution is completed, the microcontroller is requested to interrupt, and the microcontroller can read the interrupt status as the operation status of the command. If the operating s

USB\_INT\_SUCCESS indicates that the command was executed successfully. Otherwise, the command execution failed. Due to the CH375 control transfer buffer only

There are 64 bytes, so when the length of the descriptor exceeds 64 bytes, CH375 will return the operation status USB\_INT\_BUF\_OVER, for

For this USB device, the microcontroller can handle the control transfer by the ISSUE\_TOKEN or ISSUE\_TKN\_X commands.

### 1.12. Command SET\_CONFIG

This command is a control transfer command that sets the USB configuration. This command requires inputting 1 data, specifying a new USB configuration value, configuration

A value of 0 cancels the configuration, otherwise it should be taken from the configuration descriptor of the USB device. This command is used to simplify standard USB requests SET\_CONFIGURATION, CH375 requests an interrupt to the microcontroller after the command execution is completed, and the microcontroller can read the interrupt status as the operational status of the command. If the operation status is USB\_INT\_SUCCESS, the command execution is successful, otherwise the command execution fails.

### 1.13. Command AUTO\_SETUP

This command is used to automatically configure a USB device. This command is used to simplify the initialization steps of a normal USB device, which is equivalent to GET\_DESC. Multiple command sequences such as SET\_ADDRESS, SET\_CONFIGURATION. CH375 requests interrupt to the microcontroller after the command execution is completed, single The slicer can read the interrupt status as the operational status of the command. If the operation status is USB\_INT\_SUCCESS, the command execution is executed. Successful, otherwise the command execution failed.

### 1.14. Command ISSUE\_TKN\_X

This command causes CH375 to issue a synchronization token to execute the transaction. The command needs to input two data, which are the synchronization flag and the transaction. Bit 7 of the synchronization flag is the synchronization trigger flag of the receiver of the host endpoint, and bit 6 is the synchronization trigger flag of the transmitter of the host endpoint. Bit 5 to Bit 0 must be 0. The lower 4 bits of the transaction attribute specify the token PID of the transaction, and the upper 4 bits specify the destination endpoint number of the USB device. CH375 requests an interrupt from the microcontroller after the command execution is completed. The microcontroller can read the interrupt status as the operation status of the command. If the operation status is USB\_INT\_SUCCESS, the command is executed successfully. Otherwise, the command execution fails. This command is with ISSUE\_TOKEN. The only difference between commands is that the command always sets the sync trigger flag (equivalent to the SET\_ENDP? command) before executing the transaction.

### 1.15. Command ISSUE\_TOKEN

This command causes CH375 to issue a token and execute the transaction. This command requires 1 data to be entered as a transaction attribute. Low transaction attribute. The 4-bit specified transaction's token PID, the upper 4 bits specify the destination endpoint number of the USB device. CH375 please ask the microcontroller after the command is executed. To find an interrupt, the microcontroller can read the interrupt status as the operating state of the command. If the operation status is USB\_INT\_SUCCESS, then say. The execution of the command is successful. Otherwise, the command execution fails. The MCU can further analyze the cause of the failure according to the operation status.

For SETUP transactions and OUT transactions that send data, the data to be sent should be written first by the WR\_USB\_DATA7 command. Then execute the transaction with the ISSUE\_TOKEN command; for the IN transaction that receives the data, it should first be executed by the ISSUE\_TOKEN command. Row transaction, when the transaction is executed successfully, read the received data through the RD\_USB\_DATA command.

For example, when the transaction attribute byte is 09H, CH375 receives data from the default endpoint 0 of the USB device; the transaction attribute byte is At 21H, CH375 sends data to endpoint 2 of the USB device; when the transaction attribute byte is 29H, then CH375 is from the end of the USB device. Point 2 receives the data, and the address of the endpoint is 82H.

Below is the USB Token PID supported by CH375.

PID byte	name	Description
0DH	DEF_USB_PID_SETUP	Initiate control transfer, send setup data
01H	DEF_USB_PID_OUT	Execute OUT transaction, send data
09H	DEF_USB_PID_IN	Execute IN transaction, receive data

### 1.16. Command DISK\_BOC\_CMD

This command executes the BulkOnly Transfer Protocol command for a USB storage device. Before executing this command, the microcontroller must pass The WR\_USB\_DATA7 command writes the corresponding CBW packet to CH375. After the command is executed, CH375 requests an interrupt from the microcontroller. The interrupt status can be read as the operational status of the command. If the operation status is USB\_INT\_SUCCESS, the command execution is successful. For operations that have returned data, the returned data can be obtained by the RD\_USB\_DATA command.

### 1.17. Command DISK\_RESET

This command resets the USB storage device by controlling the transfer. CH375 requests an interrupt from the microcontroller after the command execution is completed. The read interrupt status is used as the operational status of the command. If the operation status is USB\_INT\_SUCCESS, the command execution is successful. when. When an error occurs in the USB storage device, the CH375 analyzes the cause of the error and automatically selects whether to reset the USB device as needed.

The complete reset process consists of resetting the USB storage device with this command and resetting the Bulk-IN endpoint with the CLR\_STALL command. The Bulk-OUT endpoint is reset by the CLR\_STALL command.

### 1.18. Command DISK\_MAX\_LUN

This command gets the maximum logical unit number of the USB storage device by controlling the transfer. CH375 please ask the microcontroller after the command is executed. To find an interrupt, the microcontroller can read the interrupt status as the operating state of the command. If the operational status is USB\_INT\_SUCCESS, then Data can be obtained by the RD\_USB\_DATA command, which is usually 1 byte.

## 2, external firmware

### 2.1. Overview

The ISSUE\_TOKEN command or the ISSUE\_TKN\_X command is used to perform basic USB transfer transactions and is firmware in USB host mode. The most basic operation in programming.

On this basis, the external MCU can handle the CH375 itself according to the requirements of the USB protocol. System transmission. Further, the MCU can handle the USB protocol of various specific types of devices according to the requirements of the USB protocol, and realize the

USB device control and data exchange. The CH375 has a built-in Bulk-Only transport protocol for Mass-Storage mass storage devices.

For USB storage devices that use the CBI transport protocol, external microcontrollers are still required to be based on the ISSUE\_TOKEN command or ISSUE\_TKN\_X. The command and control transfer commands are processed by themselves.

## 2.2. External firmware reference flow

A reference program for the external firmware is provided in the CH375 evaluation board data. The following procedure is performed by the external microcontroller through the control Quasi USB requests GET\_STATUS to get the status of the USB device for reference when designing the firmware for the external microcontroller.

### (1) The establishment phase of control transmission

- 1 Issue the WR\_USB\_DATA7 command to write 8 bytes of request data to the output buffer. The request data is 80H in order.  
00H, 00H, 00H, 00H, 00H, 02H, 00H, length is 8.

---

## Page 5

CH375 Chinese Manual (2)

5

- 2 Issue the ISSUE\_TKN\_X command to execute the transaction, the synchronization flag is 00H, the transaction attribute byte is 0DH, and the default endpoint is 0.

Issue a SETUP token and send DATA0. If you use the ISSUE\_TOKEN command to execute a transaction, you must first work

The SET\_ENDP7 command with mode byte 80H sets the sync trigger flag of the transmitter of the host endpoint to 0.

- 3 The microcontroller waits for a transaction to complete the interrupt or wait for an interrupt notification.
- 4 After the transaction is completed, CH375 sets the INT# pin low to request an interrupt from the microcontroller;
- 5 The MCU enters the interrupt service routine, or exits waiting after receiving the interrupt notification in the main program.
- 6 Issue the GET\_STATUS command to get the interrupt status.
- 7 CH375 Resets the INT# pin high after the GET\_STATUS command is completed, canceling the interrupt request.
- 8 The interrupt status obtained by the MCU analysis. If it is not USB\_INT\_SUCCESS, the operation fails and the exception processing is performed;  
If it is USB\_INT\_SUCCESS, the transaction is executed successfully and the setup phase is completed.

### (2) Control the data phase of the transmission

- 1 Issue the ISSUE\_TKN\_X command to execute the transaction, the synchronization flag is 80H, the transaction attribute byte is 09H, and the default endpoint is 0.  
Issue an IN token and receive DATA1.

- 2 The MCU waits for the transaction to complete; after the transaction is completed, CH375 requests the MCU to interrupt.
- 3 Issue the GET\_STATUS command to get the interrupt status and CH375 to cancel the interrupt request.
- 4 The interrupt status obtained by the MCU analysis, if the operation fails, the exception processing; if it is USB\_INT\_SUCCESS  
Then the transaction is executed successfully.

- 5 Issue the RD\_USB\_DATA0 command to get the data returned by the USB device and save the result as a control transfer.
- 6 Because the control transfer requires only one IN transaction, the data phase is complete.

### (3) Control the state of the transmission

- 1 Issue the WR\_USB\_DATA7 command to write the status data of length 0 to the output buffer with a length of 0.
- 2 Issue the ISSUE\_TKN\_X command to execute the transaction, the synchronization flag is 40H, the transaction attribute byte is 01H, and the default endpoint is 0.  
Issue an OUT token and send DATA1.

- 3 The MCU waits for the transaction to complete; after the transaction is completed, CH375 requests the MCU to interrupt.
- 4 Issue the GET\_STATUS command to get the interrupt status and CH375 to cancel the interrupt request.
- 5 The interrupt status obtained by the MCU analysis, if the operation fails, the exception processing; if it is USB\_INT\_SUCCESS  
The transaction is executed successfully and the status phase is completed.

- (4) Control transfer is completed, the data returned in the data phase is returned as standard USB request GET\_STATUS, usually  
The length of the returned data is 2 bytes.