

USB bus interface chip CH375

English manual: USB basic control command

Edition: 3C

<http://wch.cn>

1. Additional command

Code	Command name	Input data	Output data	Function
13H	SET_USB_ADDR	Address		Set USB address
1CH	SET_ENDP6	Work mode	Wait for 4uS	Set USB host endpoint receiver
1DH	SET_ENDP7	Work mode	Wait for 4uS	Set USB host endpoint transfer
0BH	SET_RETRY	25H		Set USB transaction retry times
		Retry times		
4FH	ISSUE_TOKEN	Issue property	Produce interruption	Send token, execute issue
41H	CLR_STALL	Endpoint number	Produce interruption	Control transfer: clear endpoint error
45H	SET_ADDRESS	Address	Produce interruption	Control transfer: set USB address
46H	GET_DESCR	Description sign type	Produce interruption	Control transfer: get description sign
49H	SET_CONFIG	Configuration value	Produce interruption	Control transfer: set USB configuration
27H	RD_USB_DATA0		Date size	Read data block from current USB interrupt endpoint buffer
			Data flow	
0FH	DELAY_100US		Delay statue	Delay 100uS
16H	TEST_CONNECT		(Wait for 20uS) connection status	Test USB device connection statue
4EH	ISSUE_TKN_X	Synchronous flag	Produce interruption	Send synchronous flag, execute transaction
		Transaction property		
4DH	AUTO_SETUP		Produce interruption	Configure USB device automatically

If the input data is USB endpoint receiver work mode, please refer to the following table.

Work mode byte	Name	Explanation bit of work mode
Bit 7~bit 6	Synchronous trigger flag	Bit 7=1, then bit 6 is new synchronous trigger flag
		00 or 01, keep current synchronous trigger flag

		10,then set synchronous flag=0	11,then set synchronous flag =1
Bit 5~bit 4	(Reserved)	(Undefined, must set 0)	
Bit 3~bit 0	Transaction respond manner	Always are 0000	

1.1. Command SET_ADDR

The command sets USB device address. One data is needed to select the address of operated USB device. The USB device address is always 00H after resetting, USB device connects or disconnects. MCU communicates with USB device via default address 00H. If sets USB device address through standard USB request, sets the same USB device address through this command so that CH375 can communicate with USB device via new address.

1.2. Command SET_ENDP6

The command sets USB host endpoint or receiver of endpoint 2. One data is input to specify new work mode. E.g. If executes IN transaction and expects to receive DATA0 to abandon DATA1, sets the host endpoint receiver's synchronous flag as 0 and relevant work mode bytes as 80H through this command. Usually, finishes the command costs 4uS below.

1.3. Command SET_ENDP7

The command sets USB host endpoint or transfer of endpoint 2. One data is input to specify new work mode. E.g. If executes SETUP or OUT transaction and expects to send DATA0, sets the host endpoint transfer's synchronous flag as 0 and relevant work mode bytes as 80H. If expects to send DATA1, the work mode bytes are C0H. Usually, finishes this command costs 4uS below.

1.4. Command SET_RETRY

The command sets retry times of USB transaction operation. The command needs to input two data such as 25H and retry times.

The bit 7 and bit 6 are to define transact mode when CH375 receives NAK. When bit 7 is 1 and bit 6 is 0 the retry times is limitless (uses ABORT_NAK to give up current retry temporary). Bit 7 is 1 and bit 6 is 1 the retry time is about 200uS. When bit 7 is 0, takes NAK as result to inform MCU or as error to handle. Bit 5 to bit 0 are to specify the retry times when USB device answer is overtime. If the data is 0, there is no retry when overtime.

The default retry times are 85H when chip reset or reconfigure USB mode, so receives NAK will retry limitless and USB device will retry five times when answer overtime.

1.5. Command ISSUE_TOKEN

The command makes CH375 to send token and execute issue. One data is to input as issue property. The lower 4 bits of issue property to specify issue's token PID while the upper 4 bits to specify intent endpoint number of USB device. After executing the command, CH375 requests interrupt to MCU. MCU can read interrupt status as this command operation status. If the operation status is USB_INT_SUCCESS, the command is executed successfully otherwise is unsuccessfully. The MCU can analyze reason according operation status.

Refer to send data command SETUP and OUT, writes data those are prepared to send through WR_USB_DATA7 command firstly, then executes issue through ISSUE_TOKEN. Toward receive

data command IN, executes issue via ISSUE-TOKEN firstly. After the issue is successful, reads the data that has received through RD_USB_DATA.

For example, the issue property bytes is 09H,CH375 receives data via the default endpoint 0 of USB device. The issue property bytes is 21H,CH375 sends data to endpoint 2 of USB device. While the issue property bytes are 29H,CH375 receives data from endpoint 2 of USB device and the endpoint address is 82H.

The following are USB token PID they are supported by CH375.

PID byte	Name	Explanation
0DH	DEF_USB_PID_SETUP	Initiate control transfer, end establish data
01H	DEF_USB_PID_OUT	Execute OUT, send data
09H	DEF_USB_PID-IN	Execute IN, receive data

1.6. Command CLR_STALL

The command is a control transfer order to clear endpoint error. One data to input to specify USB device endpoint address which will be clear error .For OUT endpoint, the effective address is vary from 01H to 0FH while the effective address varies from 81H to 8FH of IN endpoint. The command used to simplify standard USB request CLEAR_FEATURE. After executing command, CH375 requests interrupt to MCU.MCU can read interrupt statue as this command operation statue. If the operation statue is USB_INT_SUCCESS, the command is executed successfully, otherwise is failure.

1.7. Command SET_ADDRESS

The command is a control transfer order to set USB address. One data to input to specify new USB device address and the effective address varies from 00H to 7FH.The command used to simplify standard USB request SET_ADDRESS. After executing command, CH375 requests interrupt to MCU.MCU can read interrupt statue as this command operation statue. If the operation statue is USB_INT_SUCCESS, the command is executed successfully, otherwise is failure.

1.8. Command GET_DESCR

The command is a control transfer to obtain description sign. One data is input to specify the type of description sign which will obtain and the effective type is 1 or 2 corresponding to device description sign and configuration description sign respectively. The configuration description sign contains interface description sign and endpoint description sign. The command used to simplify standard USB request GET_DESCRIPTION. CH375 requests interrupt to MCU.MCU can read interrupt statue as this command operation statue. If the operation statue is USB_INT_SUCCESS, the command is executed successfully, otherwise is failure. Because the control transfer buffer is only 64-byte of CH375, CH375 will return operation statue USB_INT_OVER when the size of description sign is over 64-byte. MUC can deal control transfer through ISSUE_TOKEN command itself.

1.9. Command SET_CONFIG

The command is a control transfer order to set USB configuration. One data is input to define new USB configuration value. Cancel configuration when the configuration value is 0, else get new USB configuration value from USB device configuration description sign. The command used to simplify USB request SET_CONFIGURATION. CH375 requests interrupt to MCU.MCU can read

interrupt statue as this command operation statue. If the operation statue is USB_INT_SUCCESS, the command is executed successfully, otherwise is failure.

1.10. Command RD_USB_DATA0

The command reads data block from current USB interrupt endpoint buffer. On the USB-HOST, the function of this command is the same with RD_USB_DATA except for higher efficient of this command.

1.11. Command DELAY_100US

The command used to delay 100uS only supported parallel endpoint mode. During delay time, the parallel endpoint output 0. After delay time, the parallel endpoint output data is not 0. MCU judges the delay time over or not according the read data.

1.12. Command TEST_CONNECT

The command used to test current USB device connection state on USB-HOST. Usually, finishes this command needs 20uS. The output state is USB_INT_CONNECT、USB_INT_DISCONNECT or others after finishing the activity. The USB_INT_CONNECT implies USB device connected just a few minutes or had connected. The USB_INT_DISCONNECT implies USB device is disconnect or have cut down. If the out data is 0, the command has not finished. While other value implies USB device have connected and initialized.

1.13. Command ISSUE_TKN_X

This command makes CH375 send synchronous token to execute issue. The synchronous flag and issue property are two data that the command needed. The bit 7 of synchronous flag is receiver synchronous trigger symbol of host endpoint. The bit 6 of synchronous flag is transfer synchronous trigger symbol of host endpoint. The bit 5 to bit 0 must set as 0. The lower 4-bit of issue property imply issue token PID while the upper 4-bit imply the intention endpoint number. CH375 requests interrupt to MCU after finishing the command. MCU can read interrupt statue as this command operation statue. If the operation statue is USB_INT_SUCCESS, the command is executed successfully, otherwise is failure. The only difference from ISSUE_TOKEN is the command sets synchronous trigger flag before executing issue (equivalent add SET_ENDP? command).

1.14. Command AUTO_SETUP

The command used to set USB device automatically and simplify common USB device initialization process, equivalent with GET_DESCR、SET_ADDRESS、SET_CONFIGURATION etc serial commands. CH375 requests interrupt to MCU after finishing the command. MCU can read interrupt statue as this command operation statue. If the operation statue is USB_INT_SUCCESS, the command is executed successfully, otherwise is failure.

2. External firmware

2.1. Summarize

The command of ISSUE_TOKEN or ISSUE_TKN_X is basic operation in firmware program of USB-HOST, used to execute basic USB transfer transaction.

On the basement, the external MCU can deal with control transfer itself that CH375 does not

directly supply command to simplify according USB protocol requests. Further more, MCU deals with various special device USB protocol itself to realize control of USB device and data exchange according the USB protocol requests. CH375 sets Bulk-Only transfer protocol of Mass-Storage device internal. Refer to USB storage which uses CBI transfer protocol needs external MCU to deal with USB protocol itself according to ISSUE_TOKEN or ISSUE_TKN_X command and control transfer command.

2.2. External firmware reference flow

The resource of CH375 evaluated board supplies external firmware reference flow. The following flow is external MCU via control transfer executes standard USB request GET_STATUS and get USB device status. The flow is consulted when designing firmware program of external MCU.

(1) The foundation moment of control transfer

- ① Send WR_USB_DATA7 command to write request data of 8-byte to output buffer. The request data are 80H、00H、00H、00H、00H、00H、02H and 00H, the size is 8-byte.
- ② Send ISSUE_TKN_X command to execute issue. The synchronous flag is 00H, and the issue property byte is 0DH. Send SETUP token and DATA0 to default endpoint 0. Use ISSUE_TOKEN to execute issue, sets host endpoint transfer synchronous flag as 0 through SET_ENDP7 with work mode byte 80H.
- ③ MCU wait the interrupt of finishing issue or wait interrupt information.
- ④ After finishing the issue, CH375 sets INT# point as low-level and request interrupt to MCU.
- ⑤ MCU step into interrupt service program or quit wait when receiving interrupt information in main program.
- ⑥ Send GET_STATUS to get interrupt status.
- ⑦ After finishing the GET_STATUS, CH375 set INT# point as high-level to cancel interrupt request.
- ⑧ MCU analyze received interrupt status. If the status is not USB_INT_SUCCESS, the operation is failure and deal with the abnormality. If the status is USB_INT_SUCCESS, the issue executes successfully. The foundation moment is over.

(2) The data moment of control transfer

- ① Send ISSUE_TKN_X to execute issue. The synchronous flag is 80H. The issue property byte is 09H. Send IN token to default endpoint 0 and receive DATA1.
- ② MCU wait until finish issue and then CH375 request interrupt to MCU.
- ③ Send GET_STATUS to get interrupt status, CH375 cancel interrupt request.
- ④ MCU analyze received interrupt status. If the status is not USB_INT_SUCCESS, the operation is failure and deal with the abnormality. If the status is USB_INT_SUCCESS, the issue executes successfully.
- ⑤ Send RD_USB-DATA0 to get USB device return data and save return result that as control transfer.
- ⑥ The control transfer need IN transaction only once a time, so the data moment is over.

(3) The status moment of control transfer

- ① Send WR_USB_DATA7 write status data of size 0 to output buffer, the size is 0 byte.
- ② Send ISSUE_TKN_X to execute issue. The synchronous flag is 40H. The issue property byte is 01H. Send OUT token to default endpoint and transfer DATA1.
- ③ MCU wait until finish issue and then CH375 request interrupt to MCU.
- ④ Send GET_STATUS to get interrupt status, CH375 cancel interrupt request.
- ⑤ MCU analyze received interrupt status. If the status is not USB_INT_SUCCESS, the operation is failure and deal with the abnormality. If the status is USB_INT_SUCCESS, the issue executes successfully. The status moment is over.

(4) The control transfer is finished. The data returned during data moment see as standard USB request GET_STATUS return data. Usually, the size of return data is 2 bytes.