USB bus interface chip CH372

English DataSheet (the second): external firmware mode

Version: 1A

http://wch.cn

1. External firmware's additional command

Code	Command name	Input data	Output data	Command function
13H	SET_USB_ADDR	Address value		Set USB address
18H	SET_ENDP2	Work mode	(Wait for 4uS)	Set receiver in USB endpoint 0
19H	SET_ENDP3	Work mode	(Wait for 4uS)	Set transfer in USB endpoint 0
1AH	SET_ENDP4	Work mode	(Wait for 4uS)	Set receiver in USB endpoint 1
1BH	SET_ENDP5	Work mode	(Wait for 4uS)	Set transfer in USB endpoint 1
1CH	SET_ENDP6	Work mode	(Wait for 4uS)	Set receiver in USB endpoint 2
1DH	SET_ENDP7	Work mode	(Wait for 4uS)	Set transfer in USB endpoint 2
0AH	GET_TOGGLE	Data 1AH	Synchronous state	Get current OUT transaction synchronous state
29H	WR_USB_DATA3	Data size		Write data block to USB endpoint 0 upstream buffer
		Data flow		

If the input data is work mode of USB endpoint transceiver, please consult the following table.

Work mode byte	Name	Bit explanation of work mode	
Bit 7~bit 6	Synchronous trigger token	Bit 6 is new synchronous trigger token if bit 7 is 1	
		00 or 01=keep current synchronous trigger	
		10=set synchronous token as 0	11=set synchronous token as 1
Bit 5~bit 4	(Reserved bit)	Undefined, must be set as 0	
Bit 3~bit 0	Transaction acknowledge mode	1101=(Reserved mode, forbid to use)	
		1110=device is busy, return NAK	
		1111=device is error, return STALL	
		0000~1000=device is ready, for transfer in endpoint 0 and endpoint 1, the value appoints the data size which is ready to transfer, 0000~1000 indicates the size is 0~8, for OUT, return ACK, for IN return DATA	

1.1. Command SET_USB_ADDR

This command uses to set USB address and needs to input one data. In external firmware mode, external MCU must immediately write USB device address which is distributed by USB host to CH372 after external MCU deals with USB standard device request SET_ADDRESS. So CH372 can start new USB address to communicate with USB host.

1.2. Command SET_ENDP2

The command sets USB endpoint 0 receiver. The command needs to input one data to appoint new work mode. For example, if USB device doesn't handle with received data in control write operation in time, then through this command to set receiver transaction acknowledge mode in USB endpoint 0. Receiver can return NAK for OUT transaction to make USB host wait and retransfer data. The relevant work mode byte is 0EH. Usually, the command is finished within 4uS.

1.3. Command SET_ENDP3

The command sets USB endpoint 0 transfer. The command needs to input one data to appoint new work mode. For example, if USB device isn't supporting USB standard device request SET_INTERFACE, then through this command to set endpoint 0 transfer transaction acknowledge mode after receiving request. Transfer can return STALL for IN transaction. The relevant work mode byte is 0FH. Usually, the command is finished within 4uS.

1.4. Command SET_ENDP4

The command sets USB endpoint 1 receiver. The command needs to input one data to appoint new work mode. For example, if endpoint 1 receiver is wrong, then through this command to set receiver transaction acknowledge mode in USB endpoint 1. Receiver can return STALL for OUT transaction. The relevant work mode byte is 0FH. Usually, the command is finished within 4uS.

1.5. Command SET_ENDP5

The command sets USB endpoint 1 transfer and needs to input one data to appoint new work mode. For example, if endpoint 1 transfer has no data to transfer, through this command to set transfer transaction acknowledge mode in USB endpoint 1. Transfer can return NAK for IN transaction to make USB host wait and retransfer data. The relevant work mode byte is 0EH. Usually, the command is finished within 4uS.

1.6. Command SET_ENDP6

The command sets USB endpoint 2 receiver. And needs to input one data to appoint new work mode. For example, if aimed at endpoint 2 receiver's ENDPOINT_HALT, and receives USB standard device request SET_CONFIG, or CLEAR_FEATURE then through this command to set synchronous trigger token as 0 in endpoint 2 receiver. The relevant work mode byte is 80H. Usually, the command is finished within 4uS.

1.7. Command SET_ENDP7

The command sets USB endpoint 2 transfer. Needs to input one data to appoint new work mode. For example, if endpoint 2 transfer has no data to transfer, then through this command to set transfer transaction acknowledge mode in USB endpoint 2. Transfer can return NAK for IN transaction to make USB host wait and retransfer data. The relevant work mode byte is 0EH. Usually, the command is finished within 4uS.

1.8. Command GET_TOGGLE

The command gets the synchronous state of current OUT transaction. Needs to input one data 1AH, and the output data is synchronous state. When the bit 4 is 1 indicates the OUT transaction is synchronous while bit 4 is 0 indicates the OUT transaction is asynchronous. In control write operation, if CH372 requests OUT interrupt successfully to external MCU, MCU determines the current OUT transaction whether synchronous or not via the command. If the current OUT transaction isn't synchronous, then ignore the transaction.

1.9. Command WR_USB_DATA3

The command uses to write data block to upstream buffer in USB endpoint 0. The data first written is the size of data block, i.e. the byte number of following data flow. The virtual value of data block length is from 0 to 64.When the length isn't zero, MCU writes the following data one by one to CH375. For example, return the former 8-byte of USB description symbol to USB host through this command. After finishing execute this command for many times to return the following data of USB description symbol.

2. External firmware explanation

2.1. Endpoint 0

The receiver and transfer buffer in endpoint 0 occupy 8-byte respectively. The SETUP and OUT transaction use the same receive buffer but different transaction acknowledge manners.

CH372 will automatically set synchronous flag as 1 of receiver and transfer in endpoint 0 after CH372 successfully completes SETUP transaction in endpoint 0. Then CH372 inform the external MCU to read SETUP data and handling it via interrupt manner.

CH372 will automatically trigger receiver device synchronous trigger token in endpoint 0 after CH372 finishes OUT transaction in endpoint 0. The synchronous trigger token is triggered from o to 1 or from 1 to 0.

CH372 will automatically trigger transfer device synchronous trigger token in endpoint 0 after CH372 finishes IN transaction in endpoint 0. The synchronous trigger token is triggered from 0 to 1 or from 1 to 0.

For control read operation, CH372 automatically sets synchronous trigger token as 1 after finishing SETUP transaction, so in default CH372 transfer the first data is DATA1, and the following is DATA0, then DATA1 to analogy. Generally, external MCU only to prepare data and transfer no considering of synchronous trigger token.

CH372 finishes OUT transaction in endpoint 0 without analyzing synchronous trigger token and notify external MCU in interrupt manner. For control write operation, MCU judges current OUT transaction whether synchronous or not first via GET_TOGGLE and then handle it.

SET_ENDP2 command has no influence on SETUP transaction acknowledge manner. The receiver device returns NAK for SETUP transaction in endpoint 0 when USB buffer is not released. Return ACK while released USB buffer.

If executes WR_USB_DATA3 after SET_ENDP3 command, the transfer in endpoint 0 will automatically set transaction acknowledge manner as return DATA for IN transaction. The data size is determined by WR_USB_DATA3.

If executes SET_ENDP3 after WR_USB_DATA3, the transfer in endpoint 0 will keep data without change but deals with transaction according to transaction acknowledge manner set by SET_ENDP3 command. If the transaction acknowledge manner is set as return DATA for IN, the transfer data size is re-determined by SET_ENDP3.

2.2. Endpoint 1

The receiver and transfer buffer in endpoint 1 occupy 8-byte respectively.

CH372 will automatically trigger receiver device synchronous trigger token in endpoint 1 after CH372 successfully finishes OUT transaction in endpoint 1. The synchronous trigger token is triggered from 0 to 1 or from 1 to 0.

CH372 will automatically trigger transfer device synchronous trigger token in endpoint 1 after CH372 successfully finishes IN transaction in endpoint 1. The synchronous trigger token is triggered from 0 to 1 or from 1 to 0.

After finishing OUT transaction in endpoint 1, CH372 will automatically analyze synchronous trigger token. If the data isn’t synchronous, CH372 will inform the external MCU via interrupt manner. The external MCU only receives the OUT transaction interrupt with data synchronous.

If executes WR_USB_DATA5 after SET_ENDP5 command, the transfer in endpoint 1 will automatically set transaction acknowledge manner as return DATA for IN transaction. The data size is determined by WR_USB_DATA5.

If executes SET_ENDP5 after WR_USB_DATA5, the transfer in endpoint 1 will keep data without change but deals with transaction according to transaction acknowledge manner set by SET_ENDP5 command. If the transaction acknowledge manner is set as return DATA for IN, the transfer data size is re-determined by SET_ENDP5.

2.3. Endpoint 2

The receiver and transfer buffer in endpoint 2 occupy 64-byte respectively.

CH372 will automatically trigger receiver device synchronous trigger token in endpoint 2 after CH372 successfully finishes OUT transaction in endpoint 2. The synchronous trigger token is triggered from 0 to 1 or from 1 to 0.

CH372 will automatically trigger transfer device synchronous trigger token in endpoint 2 after CH372 finishes IN transaction in endpoint 2. The synchronous trigger token is triggered from 0 to 1 or from 1 to 0.

After finishing OUT transaction in endpoint 2, CH372 will automatically analyze synchronous trigger token. If the data isn't synchronous, CH372 will not inform the external MCU via interrupt manner. The external MCU only receives the OUT transaction interrupt with data synchronous.

If executes WR_USB_DATA7 after SET_ENDP7 command, the transfer in endpoint 2 will automatically set transaction acknowledge manner as return DATA for IN transaction. The data size is determined by WR_USB_DATA7.

If executes SET_ENDP7 after WR_USB_DATA7, the transfer in endpoint 2 will keep data and data size without change but deals with transaction according to transaction acknowledge manner set by SET_ENDP5 command.

2.4. External firmware consulting flow

The MCU source program in external firmware mode is supplied in CH372 evaluation board. The following flow is consulted when external MCU deals with USB standard device's request.

㈠ Initialize CH372 as USB device of external firmware mode and set interrupt after starting MCU.

㈡ Use GET_STATUS to obtain interrupt status after MCU receives interrupt. The analyses are as follow:

⑴ If the OUT is successful in endpoint 2 or 1, use RD_USB_DATA to read data and notify the main program to handle.

⑵ If the IN is successful in endpoint 2 or 1, use UNLOCK_USB to release buffer and notify the main program to continue.

⑶ If the SETUP is successful in endpoint 0, use RD_USB_DATA to read data, the following are the analyses:

① If USB request CLEAR_FEATURE, handles according the FEATURE and endpoint number in request. Use SET_ENDP to deal with ENDPOINT_HALT.

② If USB request GET_DESCRIPTOR, use WR_USB_DATA3 to return the former 8-byte of description symbol. And save USB request command besides with current description symbol counter for return continue.

③ If USB request SET_ADDRESS, save USB request command.

④ If USB request SET_CONFIG, save configuration value and notify the main program whether USB initialize is successful or not.

⑤ If USB request GET_CONFIN, use WR_USB_DATA3 to return current configuration value.

⑥ If USB request GET_INTERFACE, use WR_USB_DATA3 to return current interface value.

⑦ If USB request GET_STATUS, use SET_DATA3 to return current status value.

⑧ The other USB request is handled according demand, use SET_ENDP2 or SET_ENDP3 to set acknowledge as STALL when the request isn’t supported.

⑷ If the OUT is successful in endpoint 0, use RD_USB_DATA to read data, abandon data is allowed.

⑸ If the IN is successful in endpoint 0, handles it according to saved USB request command are follow:

① If USB request GET_DESCRIPTOR, use WR_USB_DATA3 to continue return the leaving description symbol.

② If USB request SET_ADDRESS, use SET_USB_ADDR to set USB address.

③ Any USB request including the upper requests, use UNLOCK_USB command before interrupt exit.

⑹ if USB bus reset, clear configuration value and so on. CH372 will automatically clear USB address and synchronous trigger token.

⑺ For any CH372 interrupt, must corresponding to a exclusive UNLOCK_USB or RD_USB_DATA command.