

USB bus interface chip CH372

English DataSheet

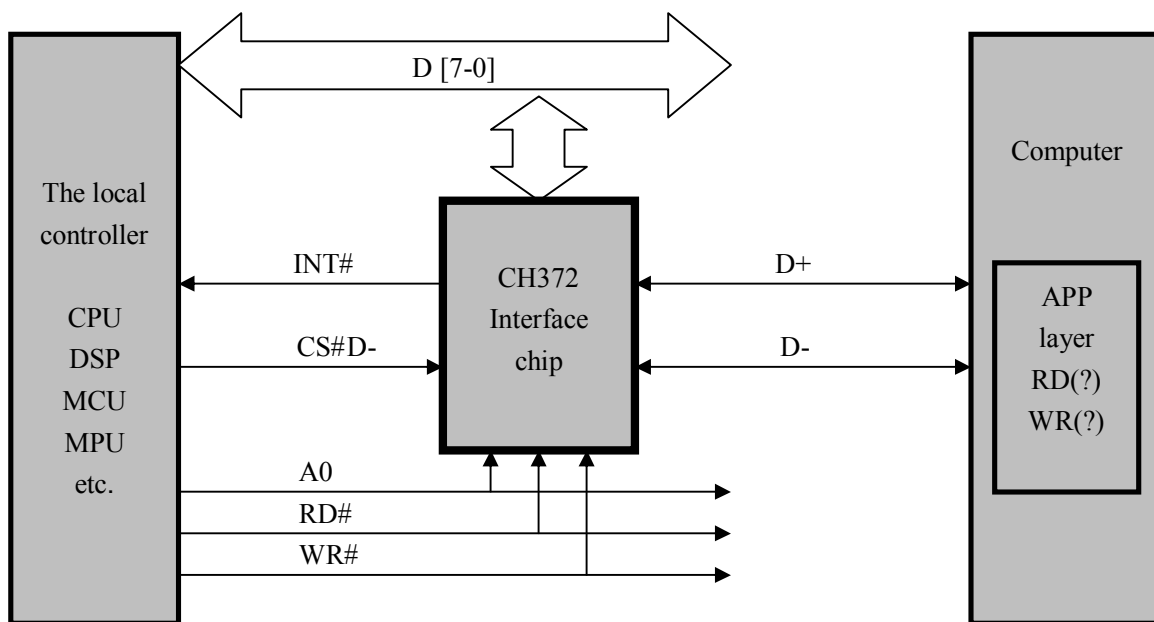
Version: 3E

<http://wch.cn>

1. Introduction

CH372 is a USB bus universal device interface chip. It is upgrade production of CH371, and function predigested edition of CH375. In the location, CH372 has 8-bit data bus and read strobe input, write strobe input and interrupt output. It is convenient to connect to CH372 on controller system bus such as CPU/DSP/MCU/MPU/microprocessors (uPs) and so on. In computer system, the equipped software of CH372 supplies operation interface which is handy and wieldy. When CH372 communicates with local MCU is just like read/write file.

CH372 set up internally lower layer protocol of USB communication. It both has advantageous set firmware up internally mode and flexible set firmware up externally mode. In set firmware up internally mode, CH372 automatically handles with all transaction on default endpoint 0 and local MCU only charges of data conversion, hence the MCU program is very simple. In set firmware externally mode, external MCU deals with various USB requests according to requirements to realize devices in term of various USB analog criterion.

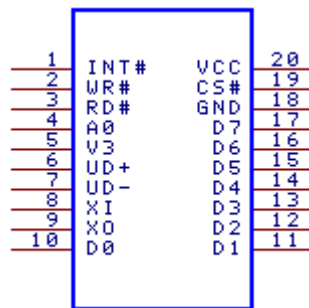


2. Features

- Full-speed USB device interface, complies with Universal Serial Bus Specification Revision 2.0, plug-and-play, only needs crystal and capacitance as peripheral components.
- Supplies a couple of master endpoints and a couple of assistant endpoint, supports control transfer, bulk transfer and interrupt transfer.
- Both has advantageous set firmware up internally mode and flexible set firmware up externally mode.
- In set firmware up internally mode, screens relevant USB protocol, automatically completes standard USB enumerate configuration, totally no needs any process in location controller, simplify firmware program in MCU.

- Universal Windows drive program supplies device layer interface, and supplies API application layer interface via DLL.
- Production vendors can self-define Vendor ID and Product ID.
- Universal local 8-bit data bus, 4-wire to control: read strobe, write strobe, chip select input and interrupt output.
- The master endpoint upstream and downstream buffer each occupies 64-byte while the assistant endpoint upstream and downstream buffer occupy 8-byte respectively.
- Operation at either +5V or +3.3V power supply input, support low-power mode.
- CH372 is function predigested edition of CH375. It gets rid of some components such as USB host mode and serial interface communication mode of CH375, so the hardware cost is lower than CH375. But its other functions are totally compatible with CH375 and use WDM drive program and DLL Dynamic Link Library of CH375 directly.
- SSOP-20 lead-free package, and compatible with RoHS, pins of CH372 are compatible with CH374T.

3. Package



| Package shape | Width of plastic | | Pitch of Pin | | Instruction of package | Ordering type |
|---------------|------------------|--------|--------------|-------|--|---------------|
| SSOP-20 | 5.30mm | 209mil | 0.65mm | 25mil | Shrink small outline package of 20-pin | CH372B |

4. Pins

| Pin No. | Pin Name | Pin Type | Pin Description |
|---------|----------|-----------------------------|---|
| 20 | VCC | POWER | Positive power input port, requires an external 0.1uF power decoupling capacitance |
| 18 | GND | POWER | Public ground, ground connection for USB |
| 5 | V3 | POWER | Attachment of VCC input external power while 3.3V; connects of 0.01uF decoupling capacitance outside while 5V |
| 8 | XI | IN | Input of crystal oscillator, attachment of crystal and crystal oscillator capacitance outside |
| 9 | XO | OUT | Opposite output of crystal oscillator, attachment of crystal and crystal oscillator capacitance outside |
| 6 | UD+ | USB signal | USB Data Signal plus |
| 7 | UD- | USB signal | USB Data Signal minus |
| 17~10 | D7~D0 | Bi-directional of tri-state | 8-bit bi-directional data bus, set up feeble pull-up resistor internal |

| | | | |
|----|------|-----|---|
| 3 | RD# | IN | Read Strobe Input, active with LOW, with feeble pull-up resistor |
| 2 | WR# | IN | Write Strobe Input, active with LOW, with feeble pull-up resistor |
| 19 | CS# | IN | Active LOW CH375 chip select, with pull-up feeble resistor |
| 1 | INT# | OUT | Interrupter request output, active with low-level |
| 4 | A0 | IN | Address wire input to identify command port and data port, with feeble pull-up resistor, A0=1,write order; A0=0,read/write data |

5. Command

The data in this manual has three types. Binary numbers are followed by a “B”. Hexadecimal numbers are followed by an “H”. Numbers without annotations are decimals.

The MCU referred in this handbook are basically applied to DSP or MCU/MPU/SCM.

This manual supplies many commands which may be used in set firmware up internally, and other commands can refer to the second handbook.

| Code | Command name | Input data | Output data | Function |
|------|--------------|----------------|-------------------------------------|--|
| 01H | GET_IC_VER | | Version number | Obtain chip and firmware version number |
| 03H | ENTER_SLEEP | | | Go into low-power and suspending |
| 05H | RERET_ALL | | (Wait for 40mS) | Execute hardware reset |
| 06H | CHECK_EXIST | Any data | OPP. accord bit | Test working status |
| 0BH | CHK_SUSPEND | Data 10H | | Set checking USB bus suspend mode |
| | | Check mode | | |
| 12H | SET_USB_ID | VID lower byte | | Set Vendor ID and Product ID of USB |
| | | VID high byte | | |
| | | PID lower byte | | |
| | | PID high byte | | |
| 15H | SET_USB_MODE | Mode code | (Wait for 20uS) Operation status | Set the work mode of USB |
| 22H | GET_STATUS | | Interruption status | Get interruption status and cancel requirement |
| 23H | UNLOCK_USB | | | Release current USB buffer |
| 27H | RD_USB_DATA0 | | Data length | Read data block from endpoint buffer of current USB interrupt |
| | | | Data stream | |
| 28H | RD_USB_DATA | | Data length | Read data from current interrupt port buffer of USB and release current buffer |
| | | | Data stream | |
| 2AH | WR_USB_DATA5 | Data length | | Write data block to upstream buffer of endpoint 1 |
| | | Data stream | | |
| 2BH | WR_USB_DATA7 | Data length | | Write data block to upstream buffer of endpoint 2 |
| | | Data stream | | |

If the output data of command is operation status, please consult the following table.

| Status code | Status name | State explanation |
|-------------|-----------------|----------------------|
| 51H | CMD_RET_SUCCESS | Operation successful |
| 5FH | CMD_RET_ABORT | Operation failure |

5.1. Command GET_IC_VER

This command can get chip and firmware version number and the return byte data is version number. The bit 7 is 1, bit 6 is 0 and bit 5 to bit 0 is version number. If the return byte data is 0B7H, take the bit 7 out, the actual version number is 37H.

5.2. Command ENTER_SLEEP

The order suspends the CH372 and let it enter into low power (some types don't support the function). The clock of CH372 stops oscillating to economize power when keeping low power. Once detection the signal of USB bus or MCU writes new command without input data (such as GET_IC_VER) to CH372, the CH372 can exit the low-power state.

In general, the time to wake up CH372 from low-power state to work normally is several milliseconds. The CH372 will generate USB_INT_WAKE_UP interrupter while totally recovering to work state.

5.3. Command RESET_ALL

Command of RESET_ALL makes the CH372 reset through hardware. Usually, the hardware reset finishes within 40mS

5.4. Command CHECK_EXIST

Command CHECK_EXIST is used to check the status to examine the CH372. when using the order, one data at random needs to input. The output data is contrary to the input data if the CH372 is working normally. For example, the output data is A8H while the input data is 57H. Additional, CH372 is reset but don't receive any command, the data is 00H read from parallel interface.

5.5. Command CHK_SUSPEND

The command sets checking USB bus suspending mode. Data 10H and check mode are two input data. The check mode has two kinds: 00H means not to check USB suspending (the default value after power-up or reset); 04H means to check USB suspending while taking 50mS as interval.

USB bus suspended status consists of two kinds states: one is the USB signal wire disconnect physically, there has no USB signal. The other is the USB host endpoint stops transferring SOF signal, i.e. USB host endpoint ask USB device enter into suspend status. When detects USB bus suspend, CH372 will generate USB_INT_USB_SUSPEND interrupt

5.6. Command SET_USB_ID

The command sets Vendor-ID and Product-ID of USB. Four data need to input. And they are the lower 8-bit of VID, the upper 8-bit of VID, the lower 8-bit of PID and the upper 8-bit of PID. If needs to set ID, the command must executes before SET_USB_MODE.

5.7. Command SET_USB_MODE

The command of SET_USB_MODE can determine the work mode of USB. The mode code may input. The code of 00H means switch to invalid USB-DEVICE mode (the default mode after power-up or

reset)

The code of 01H means switch to valid USB-DEVICE, set firmware up externally mode.

The code of 02H means switch to valid USB-DEVICE, set firmware up internally mode.

In USB device mode, the pull-up resistor is forbidden in USB bus D+ equivalent to disconnect of USB device is defined as invalid. The pull-up resistor is active in USB bus D+ equivalent to connect of USB device is defined as valid. So the USB host can detect the existence of USB device. We can simulate USB plug-unplug transaction through set whether valid or not.

Ordinarily, the time of set USB work mode is within 20uS and outputs operation station after setting.

Information about setting firmware externally mode can consult the second handbook.

5.8. Command GET_STATUS

The GET_STATUS order can obtain interrupt state of CH372 and notice CH372 to cancel the interrupt requests. MCU receives interrupt status, analyzes interrupt and deal with interrupt after the CH372 sends interrupt to MCU.

| Interrupt state byte | Name | Interrupt state explanation | | |
|----------------------|---------------------|-----------------------------|---------------|---------------|
| Bit 7~bit 4 | (Reserved) | Always be 0000 | | |
| Bit 3~bit 2 | Current transaction | 00=OUT | 10=IN | 11=SETUP |
| Bit 1~bit 0 | Current endpoint | 00=endpoint 0 | 01=endpoint 1 | 10=endpoint 2 |
| | | 11=USB bus reset | | |

The following is explanation of interrupt status. In USB device of set firmware up internally mode, MCU only needs to deal with the gray in the table, and the other interrupt status is handled by CH372 inner, automatically.

| Interrupt state value | Status name | Interrupt reason explanation |
|-----------------------|---------------------|--|
| 03H | USB_INT_BUS_RESET1 | Detection of USB bus reset (the bit 1 and bit 0 of interrupt status value is 11) |
| 07H | USB_INT_BUS_RESET2 | |
| 0BH | USB_INT_BUS_RESET3 | |
| 0FH | USB_INT_BUS_RESET4 | |
| 0CH | USB_INT_EP0_SETUP | Receiver of endpoint 0 successfully receive SETUP |
| 00H | USB_INT_EP0_OUT | Receiver of endpoint 0 successfully receive OUT |
| 08H | USB_INT_EP0_IN | Transfer of endpoint 0 successfully transfer IN |
| 01H | USB_INT_EP1_OUT | Assistant endpoint/endpoint 1 receive data, OUT successfully |
| 09H | USB_INT_EP1_IN | Interrupt endpoint/endpoint 1 transfer data, IN successfully |
| 02H | USB_INT_EP2_OUT | Bulk endpoint/endpoint 2 receive data, OUT successfully |
| 0AH | USB_INT_EP2_IN | Bulk endpoint/endpoint 2 transfer data, IN successfully |
| 05H | USB_INT_USB_SUSPEND | USB bus suspending transaction (if finish CHK_SUSPEND) |
| 06H | USB_INT_WAKE_UP | Wake up from sleep (if finish ENTER_SLEEP) |

5.9. Command UNLOCK_USB

The command used to release current USB buffer. In order to protect buffer from be covered, CH372 lock current buffer before request interrupt to MCU, and pause all USB communication until MCU release current buffer via UNLOCK_USB or via RD_USB_DATA read data. The execute times of this command have strict limit. When executes the command times are excess or low the stated times, there will be error.

5.10. Command RD_USB_DATA0

The command reads data block from current interrupt endpoint buffer of USB. The length of data block is read at first, i.e. the byte number of following data stream. The virtual value of data block length is from 0 to 64. If the length is not Zero, MCU reads following data one by one from CH372. The difference between RD_USB_DATA0 and RD_USB_DATA is after reading, the later release current buffer automatically (equivalent to add UNLOCK_USB),

5.11. Command RD_USB_DATA

The command reads data block from current USB interrupt endpoint buffer and release current buffer. The length of data block is read at first, i.e. the byte number of following data stream. The virtual value of data block length is from 0 to 64. If the length is not Zero, MCU reads following data one by one from CH372. After reading, the CH372 release USB current buffer automatically so as to receive data from USB host continually.

5.12. Command WR_USB_DATA5

The command means to write data block to upstream buffer of USB endpoint 1. In set firmware up internally mode, USB endpoint 1 is interrupt endpoint. The length of data block is written at first, i.e. the byte number of the following data stream. The virtual value of data block length is from 0 to 8. If the length is not Zero, MCU writes following data one by one to CH372.

5.13. Command WR_USB_DATA7

The command means to write data block to upstream buffer of USB endpoint 2. In set firmware up internally mode, USB endpoint 2 is bulk endpoint. The length of data block is written at first, i.e. the byte number of the following data stream. The virtual value of data block length is from 0 to 64. If the length is not Zero, MCU writes following data one by one to CH372.

6. Function explanation

6.1. General explanation

A set of CH372 includes CH372 chip and CH372 drive program in computer end. In local, CH372 deals with USB communication basic transaction automatically via set firmware program internally; in the computer end, drive program and Dynamic Link Library etc software supply application layer interface for computer.

6.2. Local hardware

In local end, CH372 supplies universal passive parallel interface. And the interfaces contain: 8-bit bi-directional data bus D7~D0, read strobe input pin RD#, write strobe input pin WR#, chip select input pin CS#, interrupt output pin INT# and address input pin A0. It is convenient to connect to CH372 on various 8-bit system bus of uPs, DSP and MCU through passive parallel interface. And CH372 can coexist with many peripheral components.

The RD# and WR# in CH372 can each connect to read strobe output pin and write strobe output pin in MCU. CS# is driven by address decode circuit and used to select device when MCU have many peripheral equipments. The INT# output interrupt request is active with low-level, and connects to interrupt input pin or common I/O pin of MCU, the MCU uses interrupt mode or query mode to acknowledge interrupt request.

When WR# is high-level, CS# and RD# are low-level besides with A0 is low-level, the data of CH372 via D7~D0 output. When RD# is high-level, CS# and WR# are low-level besides A0 is low-level, the data in

D7~D0 is written to CH372. When RD# is high-level, CS# and WR# are low-level besides A1 is high-level, the data in D7~D0 is written to CH372 as command code.

The CH372 chip occupies two address bits. When A0 is high-level, choose command port to write new command or read the interrupt token; when A0 is low-level, choose data port to read/write data.

The following is truth value table of I/O (X stands for taking no care of this bit, Z stands for tri-state forbidding of CH372)

| CS# | WR# | RD# | A0 | D7-D0 | The real operation of CH372 |
|-----|-----|-----|----|-------|--|
| 1 | X | X | X | X/Z | Don't choose CH372, any operation is executed |
| 0 | 1 | 1 | X | X/Z | Select but no operation |
| 0 | 0 | 1/X | 1 | IN | Write command code to command port of CH372 |
| 0 | 0 | 1/X | 0 | IN | Write data to data port of CH372 |
| 0 | 1 | 0 | 0 | OUT | Read data from data port of CH372 |
| 0 | 1 | 0 | 1 | OUT | Read interrupt flag from command port of CH372, the bit 7 is equal to INT# |

The UD+ and UD- in CH372 must directly connect to USB bus. If serial connect secure resistor or inductance or ESD to protect CH372, the AC/DC equivalent serial resistor is within 5Ω.

The chip CH372 set power-up reset circuit inside, and external supplies reset is not need in generally.

The CH372 needs outside supplies 12MHZ clock signal to work normally. In common, clock signal is generated by inverter in CH372 through oscillating of crystal keeping frequency. A crystal of 12MHz between XI and XO, XI and XO connect a high frequency oscillator capacitance to ground respectively can compose the peripheral circuit. The 12MHz clock signal directly input to XI while suspending XO.

CH372B can operate at 3.3V or 5V power supply. When uses 5V power supply, the VCC pin in CH372 input 5V power, and V3 connects to decoupling capacitance with capacitance from 4700pF to 0.02uF. When uses 3.3V power supply, the pin of V3 connects to VCC, and input external 3.3V power. The source power of other circuit connects to CH372 can't surpass 3.3V.

6.3. Internal structure

In the inner of CH372 integrates PLL multiplier, USB interface SIE, data buffer, passive parallel interface, command explanation device, firmware program in common etc.

PLL multiplier takes the 12MHz input from clock and generates a 48MHz reference clock for SIE.

USB interface SIE takes charge of physical USB data receive and transfer, deals with bit track and synchronization automatically, coding and decoding of NRZI, bit stuffing, parallel/serial data conversion, CRC data check, transaction handshake, retry when error, detection USB bus state etc.

Data buffer delays data receive and transfer of USB interface SIE.

Passive parallel interface exchanges data with peripheral uPs/DSP/MCU.

Command explanation device analyzes and executes various commands submitted by peripheral uPs/DSP/MCU.

Common firmware program is used to deals with various standard transaction of USB default endpoint 0 automatically.

There are five endpoints in CH372 inner.

The port0 is a default endpoint, supports upstream and downstream. The buffer of upstream and downstream is 8-byte respectively.

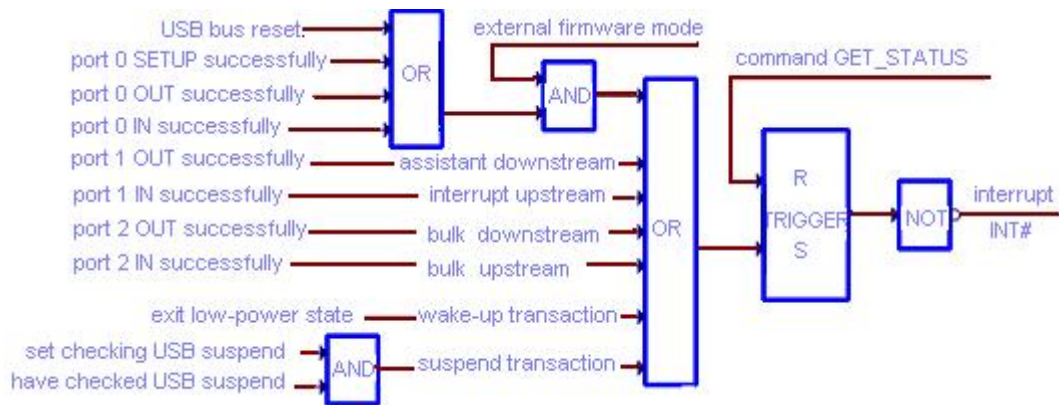
The port1 includes upstream and downstream endpoint and buffer of each is 8-byte. The upstream endpoint number is 81H while the downstream endpoint number is 01H.

The port2 includes upstream and downstream endpoint and buffer of each is 64-byte. The upstream endpoint number is 82H while the downstream endpoint number is 02H.

In set firmware up internally mode, the upstream of endpoint 2 is bulk data transfer endpoint while the downstream of endpoint 2 is bulk data receive endpoint. The upstream of endpoint 1 is interrupt endpoint while downstream of endpoint 1 is assistant endpoint.

In set firmware up externally mode, endpoint 1 as default endpoint while endpoint 1 and endpoint 2 are optional according USB product request, and defines function via external firmware supplies description symbols. In generally, endpoint 2 as data transfer master endpoint, if USB product needs, endpoint 1 may be as assistant endpoint.

The following is interrupt logical image of inner CH372.



6.4. Local MCU software

CH372 occupies 2-bit address. When A0 is high-level, selects command port to write command. When A0 is low-level selects data port to read/write data.

MCU reads from or write to CH372 through 8-bit parallel. All the operation is composed of one command code、several input data and several output data. Some command doesn't need input data while some command without output data. The process is as following:

- ① Write a command code to command port while A0=1.
- ② If the command has input data, then write input data one byte each time when A0=0.
- ③ If the command has output data, then read output data one byte each time when A0=0.
- ④ After the command is finished, pause or return to ① to execute next command.

The CH372 is specially handling USB communication. It will inform the MCU to deal with through interrupt manner when detection the change of USB bus or command is finished.

The process of MCU receive data via CH372 is as following:

- ① When CH372 receives data transfer from USB host, firstly lock current buffer to protect it from be covered by the following data, then set INT# as low-level and request interrupt to MCU;
- ② MCU enters into interrupt service, executes GET_STATUS to get interrupt status at first;
- ③ CH372 resume INT# as high-level after completing GET_STATUS and cancel interrupt request;
- ④ The obtained interrupt status is "downstream successfully" through the upper GET_STATUS, so MCU executes RD_USB_DATA to read data received from CH372.
- ⑤ After finishing GET_USB_DATA , CH372 releases current buffer to continue USB communication;
- ⑥ MCU exit interrupt service program.

The process of MCU transfer data through CH372 is as following:

- ① MCU executes WR_USB_DATA command to write data to CH372;
- ② CH372 is passively waits USB host to get data when needing;
- ③ After USB host takes data away, firstly lock current buffer to prevent retransfer data, then set INT#

as low-level and request interrupt to MCU;

- ④ MCU enters into interrupt service, executes GET_STATUS to get interrupt status at first;
- ⑤ CH372 resume INT# as high-level after completing GET_STATUS and cancel interrupt request;
- ⑥ The obtained interrupt status is “upstream successfully” through the upper GET_STATUS, so MCU executes WR_USB_DATA to write next data to CH372. If there isn't the next data to transfer, the MCU doesn't execute WR_USB_DATA command;
- ⑦ MCU executes UNLOCK_USB command;
- ⑧ After finishing UNLOCK_USB CH372 release current buffer to resume communication;
- ⑨ MCU exits interrupt service program;
- ⑩ If MCU has written the next data to transfer then return to ②, otherwise finish the transfer data.

6.5. Computer port software interface

CH372 supplies application layer interface in computer port. The application layer interface is API which is supplied by Dynamic Link Library DLL face to function of CH372. After use all the API there will return operation status but may not have acknowledge data.

The API supplied by Dynamic Link Library of CH372 contains: device manage API, data transfer API and interrupt handle API.

Information about API parameter can refer to CH372DLL.H or CH375DLL.H.

Device manage API:

Open device: CH375OpenDevice

Close device: CH375CloseDevice

Get USB device description symbol: CH375GetDeviceDescr

Get USB device configuration description symbol: CH375GetConfigDescr

Reset USB device: CH375ResetDevice

Set time out when read/write USB data: CH375SetTimeout

Set exclusive occupy current CH375 device: CH375SetExclusive

Set internal buffer upstream mode: CH375SetBuffUpload

Query the embody data package number in internal upstream buffer: CH375QueryBufUpload

Set notice program when USB device plug and unplug: CH375SetDeviceNotify

Data transfer API:

Read data (data upstream): CH375ReadData

Write data (data downstream): CH375WriteData

Abort reading operation: CH375AbortRead

Abort writing operation: CH375AbortWrite

Write assistant data (assistant data downstream): CH375WriteAuxData

Interrupt handle API:

Read interrupt data: CH375ReadInter

Abort interrupt data read operation: CH375AbortInter

Set interrupt service program: CH375SetRoutine

6.6. Port-to-port USB transfer (supply for designing application layer)

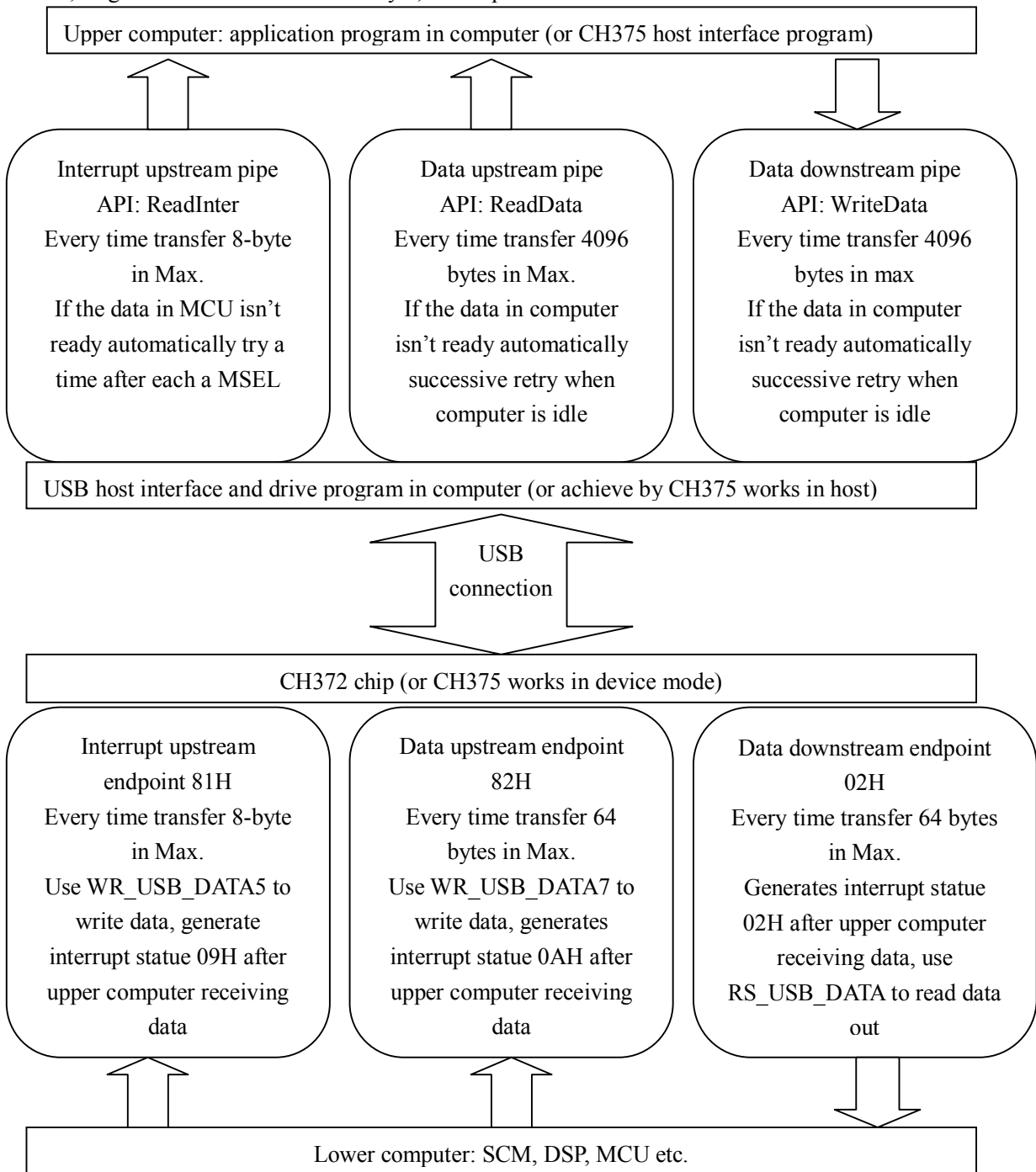
CH372 provides port-to-port connection between computer application layer and local MCU. On the basic, the designers of USB product have two choices of communication manner: single-direction data stream manner and request-acknowledge manner. The former use two opposite single-direction data stream to communicate and has higher data transfer speed, but the data is hard synchronous. The latter use active request and passive acknowledge to communicate, the data automatically synchronous. The manner has

better alternative and easy to control. The program designed is easy but the data transfer is lower.

6.6.1. Logic structure

CH372 offers four separate port-to-port logic transfer pipes. They are data upstream pipe、 data downstream pipe、 interrupt upstream pipe and assistant data downstream pipe. CH372 isn't defines each pipe's function and data structure. As a result, the designer can define functions according requirement and appoint data structure in each pipe between upper computer and lower computer.

The logic configuration in below is apply to set firmware internally mode and there isn't default endpoint 0.CH372 supplies assistant data downstream pipe which is similar with data downstream pipe except the three pipes in picture. The assistant data downstream pipe has four differences from downstream pipe: the API of upper computer is WriteAuxData; the assistant data downstream endpoint in lower computer is 01H; single time maximum size is 8-byte; interrupt statue is 01H.



6.6.2. Single-directional data stream mode

The single-directional data stream mode uses a upstream data stream and a downstream data stream to communicate with double-directional. The two data streams are totally separate.

The downstream is transferred by the computer application layer via data downstream API. The CH372 takes 64-byte as a group to divide a big data bulk into small to transfer to MCU. If the application layer transfers 150 bytes, MCU will be interrupted three times and the former two times each gets 64 bytes, the last time gets 22 bytes.

There are two transfer modes of upstream data stream: one is query mode, computer layer transfers in query mode periodically; the other is fake interrupt mode, MCU notifies the computer application layer in interrupt data, then transfer by computer application layer. Because USB bus is host-slave structure, only MCU keeps in touch with USB device the USB device can upstream data to computer.

In system that the upstream data stream is generated by query mode, the computer is always read data through data upstream API. The computer application layer will wait (if set USB read time out will exit) when the MCU doesn't have any data to upstream. In actually the route process in application layer program will be suspended be operation system. When MCU needs data, write data to CH372 bulk endpoint upstream buffer, then computer application takes the data away automatically, and CH372 notifies MCU that the upstream is successfully in interrupt mode so that the MCU can continue to upstream the following data. When use this manner, recommend to use CH375SetBufUpload to set inner buffer upstream.

In system that the upstream data stream is generated by fake interrupt mode, during initialization the computer application layer set a fake interrupt service program, as a result the application has no touch with upstream data stream. When MCU need to upstream data, writes data to bulk endpoint upstream buffer firstly, then writes the interrupt characteristic value to interrupt endpoint upstream buffer. Between a millisecond (in theory), fake interrupt service program accord to interrupt characteristic value is acted, fake interrupt service program notifies main program to use data upstream API to get upstream data. During this period, MCU will get two interrupt from CH372, the first is interrupt endpoint upstream successfully interrupt and then is the bulk endpoint upstream successfully interrupt.

6.6.3. Request-acknowledge manner

Request-acknowledge manner is a bi-directional data communication with a downstream positive request and a upstream passive acknowledge. The downstream is according to the upstream and they are relative.

Positive request is the data request that computer downstream to MCU. Passive acknowledge is acknowledge data that MCU upstream to computer application layer after MCU receiving data request. All the communication is generated by computer application layer then finished with receive acknowledge of MCU. The complete process is:

- ① Computer application layer transfer data request to CH372 according appointment structure;
- ② CH372 notifies MCU via interrupt manner;
- ③ MCU enter into interrupt service program and gets interrupt status of CH372 and analyses.
- ④ Release the current USB buffer and exit interrupt program if it is upstream;
- ⑤ Read data from data downstream buffer if it is downstream;
- ⑥ Analyses the received data and ready to acknowledge data, or exit interrupt program and the handles;
- ⑦ MCU write acknowledge data to bulk endpoint upstream buffer and exit interrupt program;
- ⑧ CH372 return the acknowledge data to computer;
- ⑨ Computer application layer receives the acknowledge data.

7. Parameter

7.1. Absolute maximum rating (Stresses above those listed can cause permanent damage to the device. Exposure to maximum rated conditions can affect device operation and reliability.)

| Name | Parameter note | | Min. | Max. | Units |
|------|---|------------------|------|---------|-------|
| TA | Ambient operating temperature | CH372A/B, VCC=5V | -40 | 85 | °C |
| | | CH372B, VCC=3.3V | -40 | 85 | |
| TS | Storage temperature | | -55 | 125 | °C |
| VCC | Voltage source (VCC connects to power, GND to ground) | | -0.5 | 6.5 | V |
| VIO | The voltage of input or output pin | | -0.5 | VCC+0.5 | V |

7.2. Electrical parameter (test conditions: TA=25°C, VCC=5V, exclude pin connection of USB bus)
(The every current parameter must multiply the coefficient of 40% when the power is 3.3V)

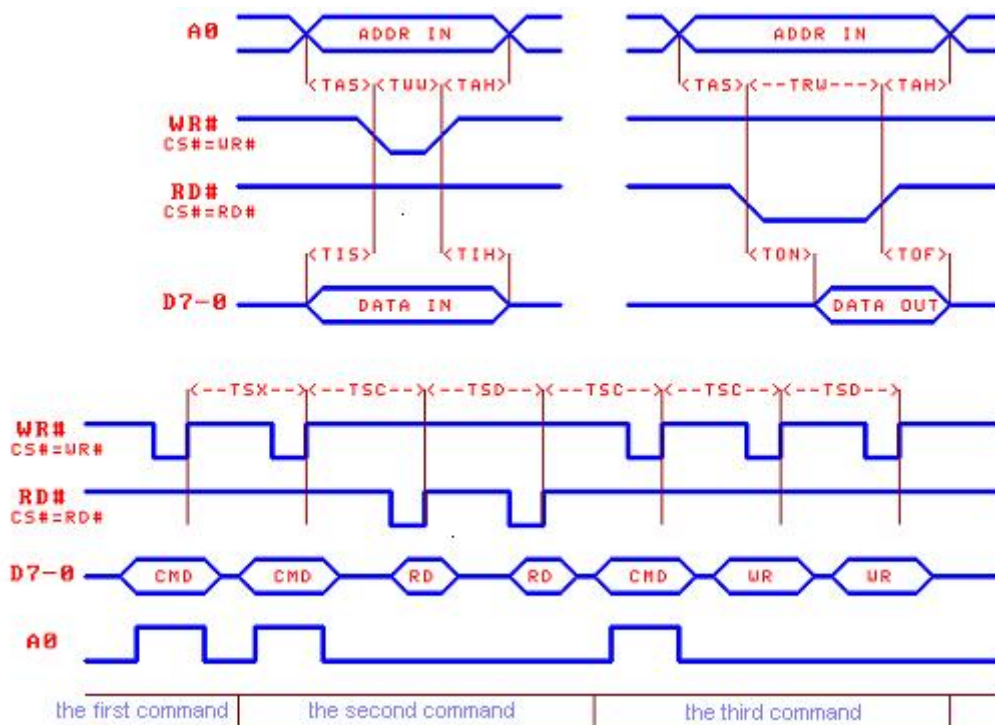
| Name | Note of parameter | | Min. | Typical | Max. | Units |
|------|---|-----------------------------------|---------|---------|---------|-------|
| VCC | Power | CH372A, V3 doesn't connect to VCC | 4.5 | 5 | 5.3 | V |
| | | CH372B, V3 connects to VCC | 3.3 | 3.3 | 3.6 | |
| ICC | Total source current when working | VCC=5V | | 12 | 30 | mA |
| | | VCC=3.3V | | 6 | 15 | |
| ISLP | Source current with low-power, I/O pin in suspend /internal pull up | VCC=5V | | 0.15 | | mA |
| | | VCC=3.3V | | 0.05 | | |
| VIL | Input Voltage LOW | | -0.5 | | 0.7 | V |
| VIH | Input Voltage HIGH | | 2.0 | | VCC+0.5 | V |
| VOL | Output Voltage LOW (draw 4mA current) | | | | 0.5 | V |
| VOH | Output Voltage HIGH (output 4mA current) | | VCC-0.5 | | | V |
| IUP | Input current in input port with internal pull-up resistor | | 40 | 80 | 160 | uA |
| VR | Edge power when power-up reset | | 2.4 | 2.7 | 3.0 | V |

Note: INT# only supplies 80uA high-level output current during CH375 reset.

7.3. Time sequence parameter (test conditions: TA=25°C, VCC=5V, refer the following picture)
(RD implies RD# and CS# are active, execute read operation when RD#=CS#=0)
(WR implies WR# and CS# are active, execute write operation when WR#=CS#=0)

| Name | Explanation of parameter | Min. | Typical | Max. | Units |
|------|--------------------------------------|-------|---------|-------|-------|
| FCLK | The input clock signal req of XI pin | 11.98 | 12.00 | 12.02 | MHz |
| TPR | Reset time of internal power-up | 18 | 30 | 40 | mS |

| | | | | | |
|------|--|-----|-----|-------|----|
| TE1 | The execute time of RESET_ALL | | 30 | 40 | mS |
| TE2 | The execute time of SET_USB_MODE | | 10 | 20 | uS |
| TE3 | The execute time of SET_ENDP? | | 2 | 4 | uS |
| TE0 | The execute time of other commands | | 1.5 | 2 | uS |
| TSX | The interval time between commands code of CH372B | 1.5 | | | uS |
| TSC | The interval time between command code and data of CH372B | 1.5 | | 1000 | uS |
| TSD | The interval time between data of CH372B | 0.6 | | 1000 | uS |
| TWW | Write pulse width | 60 | | 10000 | nS |
| TRW | Read pulse width | 60 | | 10000 | nS |
| TAS | Address to Read HIGH or Write HIGH SET-UP TIME | 5 | | | nS |
| TAH | Address hold time after Read HIGH or Write HIGH | 5 | | | nS |
| TIS | Data to Write HIGH set-up time | 0 | | | nS |
| TIH | Data hold time after Write HIGH | 5 | | | nS |
| TON | Data valid after Read LOW | 0 | | 30 | nS |
| TOF | Data hold after Read HIGH | 0 | | 20 | nS |
| TINT | The time from receiving GET_STATUS to INT# cancel interruption | | 1.5 | 3 | uS |
| TWAK | The woke-up time from low-power state | 2 | 6 | 10 | mS |



8. Application

8.1. Connect USB bus (the following picture)

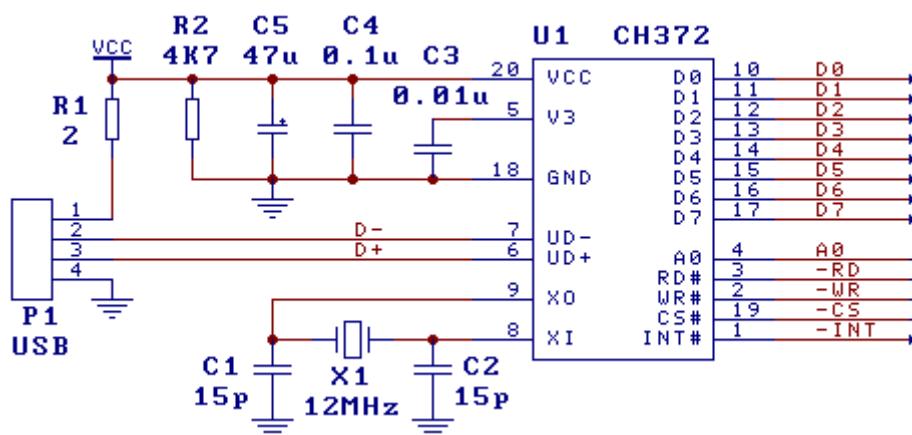
The P1 is USB port, USB bus contains a pair of 5V source wires and a pair of data signal wires. Usually, +5V source wire is red, ground wire is black, D+ signal wire is green and D- is white. The maximum source current is up to 500mA of USB bus. In generally, the low-power USB product can directly use 5V source power supplied by USB bus. If USB products use other manner to supply common power, the CH372 must use the common power with MCU and disconnect power of USB bus. When use USB bus power and common power at the same time, connects to R1 about 1Ω between 5V source wire in USB bus and 5V common power, and the ground wires of each other are connected directly.

In the picture, the optional resistor R2 is used to eliminate the power of C5 in time when disconnect with source power so that the VCC can descend to 0V without delay. Through this CH372 will power-up reset reliable the next time. In order to reset credible, the time of source power change from 0V to 5V must below 100ms, so the value of C5 and R1 may not too high.

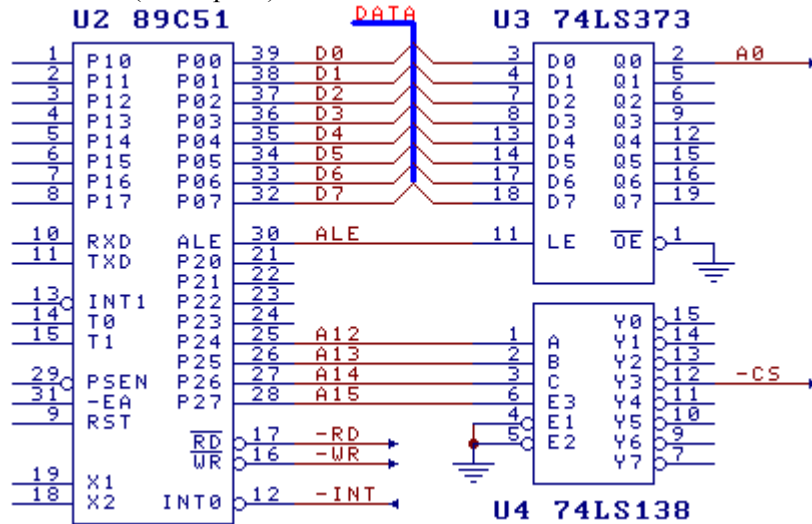
The capacitance C3 eliminates the coupling of inner power of CH372. The capacity of C3 is 4700pF to 0.02uF. It is made of monolithic or high frequency ceramic. If there is no request to EMI, the C3 is not in need. The C4 and C5 are used to decoupling of external power. The C4 is 0.1uF and made of monolithic or high frequency ceramic. The crystal X1, capacitance C1 and C2 are composed of clock oscillating circuit of CH372. The frequency of X1 is 12MHz, C1 and C2 are monolithic or high frequency ceramic capacitors of 15pF~30pF.

If the source voltage is 3.3V of CH372, connect the V3 with VCC, and input 3.3V voltage, the C3 can be taken out of the application.

When designing the PCB, pay much attention to some notes: decoupling capacitance C3 and C4 must keep near to connection pin of CH372; makes sure D+ and D- are parallel and supply ground or covering copper besides to decrease the disturb from outside signal; the relevant signal leads between XI and XO must be kept as short as possible. In order to lessen the high frequency clock disturb outside, setting ground wire on the circle or covering copper to the relative equipment.



8.2. Connect MCU bus (bus expend)



The CH372 has the common passive parallel interface, connects to various uPs, DSP and MCU directly. In common MCS-51 typical application circuit, CH372 can connects to system bus of MCU U2 through 8-bit passive parallel interfaces D7~D0, -RD, -WR, -CS and A0.

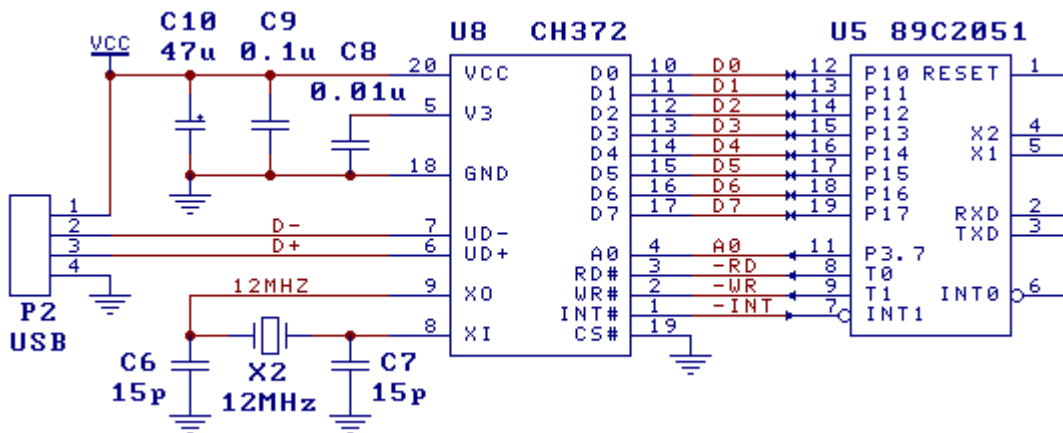
If the MCS-51 has not lock A7 ~A0 through U3,uses the P20 of U2 to drive the address wire A0 of CH372 and modify the port address of MCU program.U4 takes charge of simple address encoding, produces chip selection signal in need. The chip selection address varies from B000H to BFFFH of CH372 in the image while occupies two address in actually. The BXX1H used to write commend as the BXX0H used to read/write data.

8.3. Connect MCU I/O (not bus enlarge)

In MCU system without external enlarge bus, MCU can use common I/O pin to simulate 8-bit parallel interface timing sequence to operate CH372.

In common MCS-51 typical application circuit to simply MCU, the CS# in CH372 is fixed as low-level and on chip select statue. In MCU program, P1 port in U5 is served as 8-bit bi-directional data bus to control each I/O pin simulates parallel timing sequence exchange data with CH372.

The circuit in the picture is simple and can used to software dog、 USB encrypt lock、 USB change to serial interface or RS485 and so on.



8.4. MCU interface program

The detail about MCU interface program can refer to CH372 evaluation board.