USB 总线接口芯片 CH372

中文手册

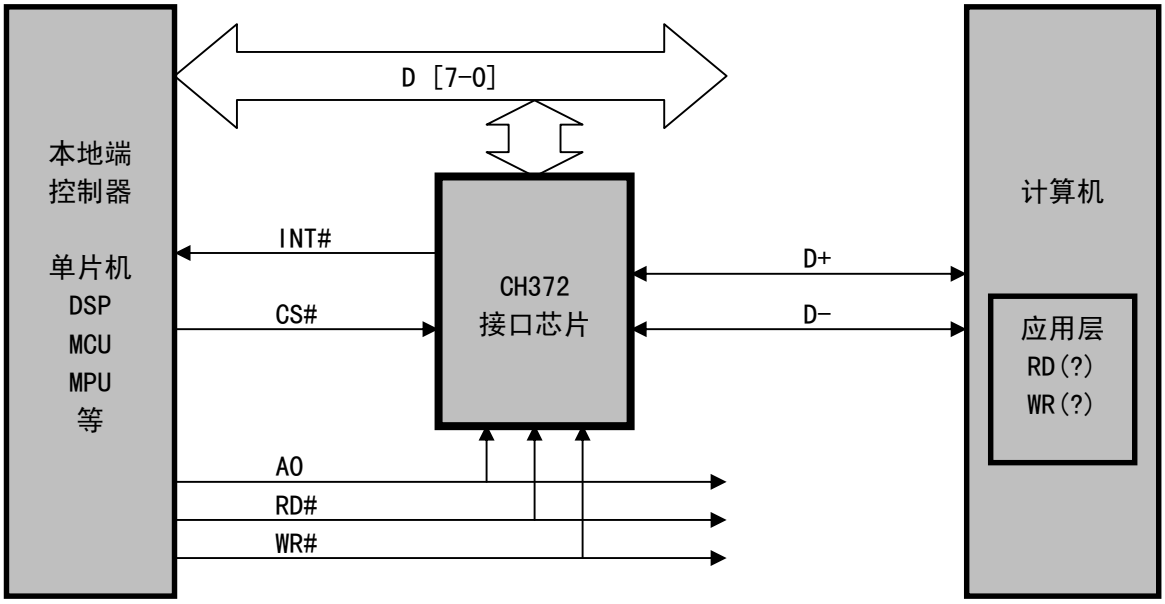
版本： 4

http://wch.cn

1、概述

CH372 是一个 USB 总线的通用设备接口芯片，是 CH371 的升级产品，是 CH375 芯片的功能简化版。在本地端，CH372 具有 8 位数据总线和读、写、片选控制线以及中断输出，可以方便地挂接到单片机/DSP/MCU/MPU 等控制器的系统总线上；在计算机系统中，CH372 的配套软件提供了简洁易用的操作接口，与本地端的单片机通讯就如同读写文件。

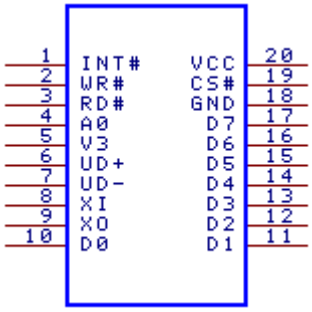
CH372 内置了 USB 通讯中的底层协议，具有省事的内置固件模式和灵活的外置固件模式。在内置固件模式下，CH372 自动处理默认端点 0 的所有事务，本地端单片机只要负责数据交换，所以单片机程序非常简洁。在外置固件模式下，由外部单片机根据需要自行处理各种 USB 请求，从而可以实现符合各种 USB 类规范的设备。



2、特点

- 全速 USB 设备接口，兼容 USB V2.0，即插即用，外围元器件只需要晶体和电容。
- 提供一对主端点和一对辅助端点，支持控制传输、批量传输、中断传输。
- 具有省事的内置固件模式和灵活的外部固件模式。
- 内置固件模式下屏蔽了相关的 USB 协议，自动完成标准的 USB 枚举配置过程，完全不需要本地端控制器作任何处理，简化了单片机的固件编程。
- 通用 Windows 驱动程序提供设备级接口，通过 DLL 提供 API 应用层接口。
- 产品制造商可以自定义厂商标识（Vendor ID）和产品标识（Product ID）。
- 通用的本地 8 位数据总线，4 线控制：读选通、写选通、片选输入、中断输出。
- 主端点上传下传缓冲区各 64 字节，辅助端点上传下传缓冲区各 8 字节。
- 支持 5V 电源电压和 3.3V 电源电压，支持低功耗模式。
- CH372C 芯片还支持内置时钟，无需外部晶体。
- CH372 芯片是 CH375 芯片的功能简化版，CH372 在 CH375 基础上减少了 USB 主机方式和串口通讯方式等功能，所以硬件成本更低，但是其它功能完全兼容 CH375，可以直接使用 CH375 的 WDM 驱动程序和 DLL 动态链接库。
- 采用小型的 SSOP-20 无铅封装，兼容 RoHS，引脚兼容 CH374T 芯片。

3、封装



封装形式	塑体宽度		引脚间距		封装说明	订货型号
SSOP-20	5.30mm	209mil	0.65mm	25mil	缩小型 20 脚贴片	CH372C
SSOP-20	5.30mm	209mil	0.65mm	25mil	缩小型 20 脚贴片	CH372B

区别：CH372B 必须外接晶体及振荡电容；
CH372C 既可以选择外接晶体及电容，也可以选择不外接而直接使用内置时钟。

4、引脚

引脚号	引脚名称	类型	引脚说明
20	VCC	电源	正电源输入端，需要外接 0.1uF 电源退耦电容
18	GND	电源	公共接地端，需要连接 USB 总线的地线
5	V3	电源	在 3.3V 电源电压时连接 VCC 输入外部电源， 在 5V 电源电压时外接容量为 0.01uF～0.1uF 退耦电容
8	XI	输入	晶体振荡的输入端，需要外接晶体及振荡电容。 对于 CH372C 的内置时钟模式，XI 应该接 GND
9	XO	输出	晶体振荡的反相输出端，需要外接晶体及振荡电容。 对于 CH372C 的内置时钟模式，XO 应该悬空
6	UD+	USB 信号	USB 总线的 D+数据线
7	UD-	USB 信号	USB 总线的 D-数据线
17～10	D7～D0	双向三态	8 位双向数据总线，内置弱上拉电阻
3	RD#	输入	读选通输入，低电平有效，内置弱上拉电阻
2	WR#	输入	写选通输入，低电平有效，内置弱上拉电阻
19	CS#	输入	片选控制输入，低电平有效，内置弱上拉电阻
1	INT#	输出	中断请求输出，低电平有效
4	A0	输入	地址线输入，区分命令口与数据口，内置弱上拉电阻， 当 A0=1 时可以写命令，当 A0=0 时可以读写数据

5、命令

本手册中的数据，后缀 B 为二进制数，后缀 H 为十六进制数，否则为十进制数。
本手册中所指的单片机基本适用于 DSP 或者 MCU/MPU/SCM 等。
本手册提供了内置固件模式下可能用到的命令，其它命令请参考手册（二）。

代码	命令名称	输入数据	输出数据	命令用途
01H	GET_IC_VER		版本号	获取芯片及固件版本
03H	ENTER_SLEEP			进入低功耗睡眠挂起状态
05H	RESET_ALL		（等 40mS）	执行硬件复位
06H	CHECK_EXIST	任意数据	按位取反	测试工作状态

0BH	CHK_SUSPEND	数据 10H 检查方式		设置检查 USB 总线 挂起状态的方式
12H	SET_USB_ID	VID 低字节 VID 高字节 PID 低字节 PID 高字节		设置 USB 的厂商识别码 VID 和产品识别码 PID
15H	SET_USB_MODE	模式代码	（等 20uS） 操作状态	设置 USB 工作模式
22H	GET_STATUS		中断状态	获取中断状态并取消请求
23H	UNLOCK_USB			释放当前 USB 缓冲区
27H	RD_USB_DATA0		数据长度 数据流	从当前 USB 中断的端点缓冲区 读取数据块
28H	RD_USB_DATA		数据长度 数据流	从当前 USB 中断的端点缓冲区 读取数据块并释放当前缓冲区
2AH	WR_USB_DATA5	数据长度 数据流		向 USB 端点 1 的上传缓冲区 写入数据块
2BH	WR_USB_DATA7	数据长度 数据流		向 USB 端点 2 的上传缓冲区 写入数据块

如果命令的输出数据是操作状态，参考下表。

状态代码	状态名称	状态说明
51H	CMD_RET_SUCCESS	操作成功
5FH	CMD_RET_ABORT	操作失败

5.1. 命令 GET_IC_VER

该命令获取芯片及固件版本。返回的一个字节数据是版本号，其位 7 为 1，位 6 为 0，位 5～位 0 为版本号。如果返回值为 0B7H，去掉位 7 的 1，实际版本号为 37H。

5.2. 命令 ENTER_SLEEP

该命令使 CH372 芯片进入低功耗睡眠挂起状态（部分型号的芯片不支持该功能）。进入低功耗状态后，CH372 芯片的时钟停振，从而节约电能，直到检测到以下两种情况之一才退出低功耗状态：一是检测到 USB 总线有信号；二是单片机向 CH372 芯片写入新的命令（没有输入数据的命令，例如 GET_IC_VER 命令）。

通常情况下，CH372 芯片从低功耗状态退出并恢复到正常工作状态需要几毫秒的唤醒时间，当完全恢复到正常工作状态后，CH372 将产生 USB_INT_WAKE_UP 事件中断。

5.3. 命令 RESET_ALL

该命令使 CH372 执行硬件复位。通常情况下，硬件复位在 40mS 时间之内完成。

5.4. 命令 CHECK_EXIST

该命令测试工作状态，以检查 CH372 是否正常工作。该命令需要输入 1 个数据，可以是任意数据，如果 CH372 正常工作，那么 CH372 的输出数据是输入数据的按位取反。例如，输入数据是 57H，则输出数据是 A8H。另外，在 CH372 复位后未收到任何命令之前，从其并口通常是读出数据 00H。

5.5. 命令 CHK_SUSPEND

该命令设置检查 USB 总线挂起状态的方式。该命令需要输入两个数据，分别是数据 10H 和检查方式。检查方式有 2 种：00H 说明不检查 USB 挂起（上电或复位后的默认值）；04H 说明以 50mS 为间隔检查 USB 挂起。

USB 总线挂起状态包括两种情况：一是 USB 信号线物理断开，完全没有 USB 信号；二是 USB 主机端停止发送 SOF 信号，也就是 USB 主机端要求 USB 设备进入挂起状态。当检查到 USB 总线挂起状态后，CH372 将产生 USB_INT_USB_SUSPEND 事件中断。

5.6. 命令 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，依次是厂商 VID 的低 8 位、厂商 VID 的高 8 位，产品 PID 的低 8 位，产品 PID 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。

5.7. 命令 SET_USB_MODE

该命令设置 USB 工作模式。该命令需要输入 1 个数据，该数据是模式代码：

模式代码为 00H 时切换到未启用的 USB 设备方式（上电或复位后的默认方式）；

模式代码为 01H 时切换到已启用的 USB 设备方式，外部固件模式；

模式代码为 02H 时切换到已启用的 USB 设备方式，内置固件模式。

在 USB 设备方式下，未启用是指 USB 总线 D+的上拉电阻被禁止，相当于断开 USB 设备；启用是指 USB 总线 D+的上拉电阻有效，相当于连接 USB 设备，从而使 USB 主机能够检测到 USB 设备的存在。通过设置是否启用，可以模拟 USB 设备的插拔事件。

通常情况下，设置 USB 工作模式在 20uS 时间之内完成，完成后输出操作状态。

关于外部固件模式请参考手册（二）。

5.8. 命令 GET_STATUS

该命令获取 CH372 的中断状态并通知 CH372 取消中断请求。当 CH372 向单片机请求中断后，单片机通过该命令获取中断状态，分析中断原因并处理。

中断状态字节	名称	中断状态分析说明		
位 7～位 4	（保留位）	总是 0000		
位 3～位 2	当前事务	00=OUT 事务	10=IN 事务	11=SETUP 事务
位 1～位 0	当前端点	00=端点 0	01=端点 1	10=端点 2
		11=USB 总线复位		

下面是中断状态的分析说明。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，CH372 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
03H、07H、0BH、0FH	USB_INT_BUS_RESET1～USB_INT_BUS_RESET4	检测到 USB 总线复位 （中断状态值的位 1 和位 0 为 11）
0CH	USB_INT_EP0_SETUP	端点 0 的接收器接收到数据，SETUP 成功
00H	USB_INT_EP0_OUT	端点 0 的接收器接收到数据，OUT 成功
08H	USB_INT_EP0_IN	端点 0 的发送器发送完数据，IN 成功
01H	USB_INT_EP1_OUT	辅助端点/端点 1 接收到数据，OUT 成功
09H	USB_INT_EP1_IN	中断端点/端点 1 发送完数据，IN 成功
02H	USB_INT_EP2_OUT	批量端点/端点 2 接收到数据，OUT 成功
0AH	USB_INT_EP2_IN	批量端点/端点 2 发送完数据，IN 成功

05H	USB_INT_USB_SUSPEND	USB 总线挂起事件（如果已 CHK_SUSPEND）
06H	USB_INT_WAKE_UP	从睡眠中被唤醒事件（如果已 ENTER_SLEEP）

5.9. 命令 UNLOCK_USB

该命令释放当前 USB 缓冲区。为了防止缓冲区覆盖，CH372 向单片机请求中断前首先锁定当前缓冲区,暂停所有的 USB 通讯,直到单片机通过 UNLOCK_USB 命令释放当前缓冲区,或者通过 RD_USB_DATA 命令读取数据后才会释放当前缓冲区。该命令不能多执行，也不能少执行。

5.10. 命令 RD_USB_DATA0

该命令从当前 USB 中断的端点缓冲区中读取数据块。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。数据块长度的有效值是 0 至 64，如果长度不为 0，则单片机必须将后续数据从 CH372 逐个读取完。该命令与 RD_USB_DATA 命令的唯一区别是后者在读取完成后还会自动释放当前 USB 缓冲区（相当于再加上 UNLOCK_USB 命令）。

5.11. 命令 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。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。数据块长度的有效值是 0 至 64，如果长度不为 0，则单片机必须将后续数据从 CH372 逐个读取完；读取数据后，CH372 自动释放 USB 当前缓冲区，从而可以继续接收 USB 主机发来的数据。

5.12. 命令 WR_USB_DATA5

该命令向 USB 端点 1 的上传缓冲区写入数据块，在内置固件模式下，USB 端点 1 就是中断端点。首先写入的输入数据是数据块长度，也就是后续数据流的字节数。数据块长度的有效值是 0 至 8，如果长度不为 0，则单片机必须将后续数据逐个写入 CH372。

5.13. 命令 WR_USB_DATA7

该命令向 USB 端点 2 的上传缓冲区写入数据块，在内置固件模式下，USB 端点 2 就是批量端点。首先写入的输入数据是数据块长度，也就是后续数据流的字节数。数据块长度的有效值是 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 CH372。

6、功能说明

6.1. 一般说明

CH372 套件包括 CH372 芯片和计算机端的 CH372 驱动程序。在本地端，CH372 芯片以内置的固件程序自动处理了 USB 通讯中的基本事务；在计算机端，驱动程序以及动态链接库等软件向计算机应用层提供应用层接口。

6.2. 本地端的硬件

CH372 芯片的被动并行接口包括：8 位双向数据总线 D7～D0、读选通输入引脚 RD#、写选通输入引脚 WR#、片选输入引脚 CS#、中断输出引脚 INT#以及地址输入引脚 A0。通过被动并行接口，CH372 芯片可以很方便地挂接到各种 8 位单片机、DSP、MCU 的系统总线上，并且可以与多个外围器件共存。

CH372 芯片的 RD#和 WR#可以分别连接到单片机的读选通输出引脚和写选通输出引脚。CS#由地址

译码电路驱动，用于当单片机具有多个外围器件时进行设备选择。INT#输出的中断请求是低电平有效，可以连接到单片机的中断输入引脚或者普通 I/O 引脚，单片机可以使用中断方式或者查询方式获知中断请求。

当 WR#为高电平并且 CS#和 RD#及 A0 都为低电平时，CH372 中的数据通过 D7～D0 输出；当 RD#为高电平并且 CS#和 WR#及 A0 都为低电平时，D7～D0 上的数据被写入 CH372 芯片中；当 RD#为高电平并且 CS#和 WR#都为低电平而 A1 为高电平时，D7～D0 上的数据被作为命令码写入 CH372 芯片中。

CH372 芯片占用两个地址位，当 A0 引脚为高电平时选择命令端口，可以写入新的命令，或者读出中断标志；当 A0 引脚为低电平时选择数据端口，可以读写数据。

下表为并口 I/O 操作的真值表（表中 X 代表不关心此位，Z 代表 CH372 三态禁止）。

CS#	WR#	RD#	A0	D7-D0	对 CH372 芯片的实际操作
1	X	X	X	X/Z	未选中 CH372，不进行任何操作
0	1	1	X	X/Z	虽然选中但无操作，不进行任何操作
0	0	1/X	1	输入	向 CH372 的命令端口写入命令码
0	0	1/X	0	输入	向 CH372 的数据端口写入数据
0	1	0	0	输出	从 CH372 的数据端口读出数据
0	1	0	1	输出	从 CH372B/C 的命令端口读取中断标志，位 7 等效于 INT#引脚

CH372 芯片的 UD+和 UD-引脚应该直接连接到 USB 总线上。如果为了芯片安全而串接保险电阻或者电感或者 ESD 保护器件，那么交直流等效串联电阻应该在 5Ω之内。

CH372 芯片内置了电源上电复位电路，一般情况下，不需要外部提供复位。

CH372B 芯片正常工作时需要外部为其提供 12MHz 的时钟信号。一般情况下，时钟信号由 CH372B 内置的反相器通过晶体稳频振荡产生。外围电路只需要在 XI 和 XO 引脚之间连接一个标称频率为 12MHz 的晶体，并且分别为 XI 和 XO 引脚对地连接一个高频振荡电容。如果从外部直接输入 12MHz 时钟信号，那么应该从 XI 引脚输入，而 XO 引脚悬空。

CH372C 芯片支持外部时钟和内置时钟两种模式，外部时钟模式参考上述 CH372B 外接 12MHz 晶体及电容；内置时钟模式应该将 XI 引脚接 GND，并悬空 XO 引脚，从而节约外部的晶体及振荡电容。

CH372B 和 C 芯片支持 3.3V 或者 5V 电源电压。当使用 5V 工作电压时，CH372 芯片的 VCC 引脚输入外部 5V 电源，并且 V3 引脚应该外接容量为 0.01uF 到 0.1uF 的电源退耦电容。当使用 3.3V 工作电压时，CH372 芯片的 V3 引脚应该与 VCC 引脚相连接，同时输入外部的 3.3V 电源，并且与 CH372 芯片相连接的其它电路的工作电压不能超过 3.3V。

6.3. 内部结构

CH372 芯片内部集成了 PLL 倍频器、USB 接口 SIE、数据缓冲区、被动并行接口、命令解释器、通用的固件程序等主要部件。

PLL 倍频器用于将外部输入的 12MHz 时钟倍频到 48MHz，作为 USB 接口 SIE 时钟。

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。

被动并行接口用于与外部单片机/DSP/MCU 交换数据。

命令解释器用于分析并执行外部单片机/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。

CH372 芯片内部具有 5 个物理端点：

端点 0 是默认端点，支持上传和下传，上传和下传缓冲区各是 8 个字节；

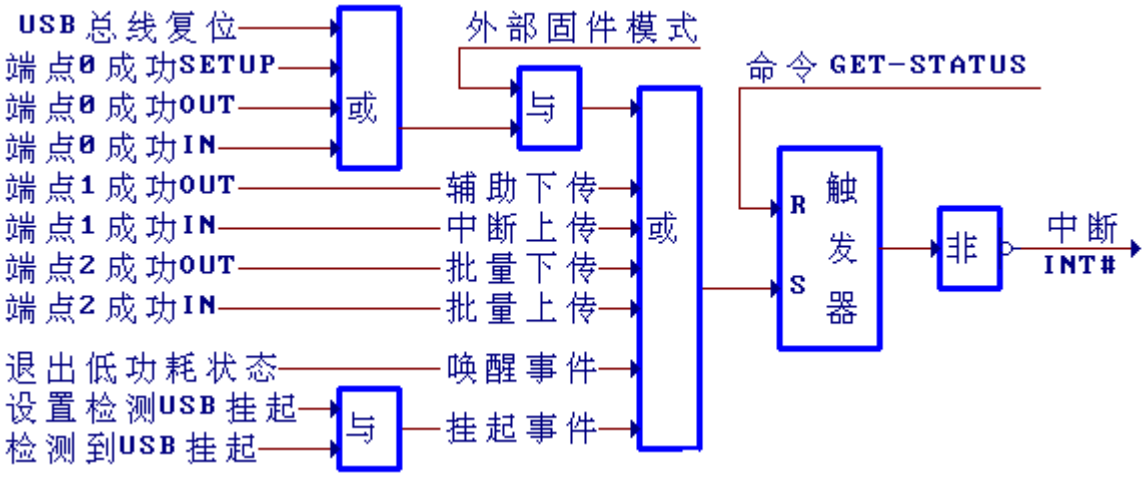
端点 1 包括上传端点和下传端点，上传和下传缓冲区各是 8 个字节，上传端点的端点号是 81H，下传端点的端点号是 01H；

端点 2 包括上传端点和下传端点，上传和下传缓冲区各是 64 个字节，上传端点的端点号是 82H，下传端点的端点号是 02H。

在内置固件模式下，端点 2 的上传端点作为批量数据发送端点，端点 2 的下传端点作为批量数据接收端点，端点 1 的上传端点作为中断端点，端点 1 的下传端点作为辅助端点。

在外部固件模式下，端点 0 作为默认端点，端点 1 和端点 2 可以根据 USB 产品的需要选择使用，并且可以由外部固件提供的描述符分别定义其用途。通常情况下，端点 2 作为数据传输的主端点，如果 USB 产品需要，可以将端点 1 作为辅助端点。

下图为 CH372 芯片内部的中断逻辑图。



6.4. 本地端的单片机软件

CH372 芯片占用两个地址位，当 A0 引脚为高电平时选择命令端口，可以写入命令；当 A0 引脚为低电平时选择数据端口，可以读写数据。

单片机通过 8 位并口对 CH372 芯片进行读写，所有操作都是由一个命令码、若干个输入数据和若干个输出数据组成，部分命令不需要输入数据，部分命令没有输出数据。命令操作步骤如下：

①、 在 A0=1 时向命令端口写入命令代码；

②、 如果该命令具有输入数据，则在 A0=0 时依次写入输入数据，每次一个字节；

③、 如果该命令具有输出数据，则在 A0=0 时依次读取输出数据，每次一个字节；

④、 命令完成，可以暂停或者转到①继续执行下一个命令。

CH372 芯片专门用于处理 USB 通讯，在接收到数据后或者发送完数据后，CH372 以中断方式通知单片机进行处理。

单片机通过 CH372 芯片接收数据的处理步骤如下：

①、 当 CH372 接收到 USB 主机发来的数据后，首先锁定当前 USB 缓冲区，防止被后续数据覆盖，然后将 INT#引脚设置为低电平，向单片机请求中断；

②、 单片机进入中断服务程序，首先执行 GET_STATUS 命令获取中断状态；

③、 CH372 在 GET_STATUS 命令完成后将 INT#引脚恢复为高电平，取消中断请求；

④、 由于通过上述 GET_STATUS 命令获取的中断状态是“下传成功”，所以单片机执行 RD_USB_DATA 命令从 CH372 读取接收到的数据；

⑤、 CH372 在 RD_USB_DATA 命令完成后释放当前缓冲区，从而可以继续 USB 通讯；

⑥、 单片机退出中断服务程序。

单片机通过 CH372 芯片发送数据的处理步骤如下：

①、 单片机执行 WR_USB_DATA 命令向 CH372 写入要发送的数据；

②、 CH372 被动地等待 USB 主机在需要时取走数据；

③、 当 USB 主机取走数据后，CH372 首先锁定当前 USB 缓冲区，防止重复发送数据，然后将 INT#引脚设置为低电平，向单片机请求中断；

④、 单片机进入中断服务程序，首先执行 GET_STATUS 命令获取中断状态；

⑤、 CH372 在 GET_STATUS 命令完成后将 INT#引脚恢复为高电平，取消中断请求；

⑥、 由于通过上述 GET_STATUS 命令获取的中断状态是“上传成功”，所以单片机执行 WR_USB_DATA 命令向 CH372 写入另一组要发送的数据，如果没有后续数据需要发送，那么

单片机不必执行 WR_USB_DATA 命令；

⑦、 单片机执行 UNLOCK_USB 命令；

⑧、 CH372 在 UNLOCK_USB 命令完成后释放当前缓冲区，从而可以继续 USB 通讯；

⑨、 单片机退出中断服务程序；

⑩、 如果单片机已经写入了另一组要发送的数据，那么转到②，否则结束。

6.5. 计算机端的软件接口

CH372 在计算机端提供了应用层接口，应用层接口是由 CH372 动态链接库 DLL 提供的面向功能应用的 API，所有 API 在调用后都有操作状态返回，但不一定有应答数据。

CH372 动态链接库提供的 API 包括：设备管理 API、数据传输 API、中断处理 API。

有关 API 参数的说明请参考 CH372DLL.H 或者 CH375DLL.H。

设备管理 API：

打开设备：CH375OpenDevice

关闭设备：CH375CloseDevice

获取 USB 设备描述符：CH375GetDeviceDescr

获取 USB 配置描述符：CH375GetConfigDescr

复位 USB 设备：CH375ResetDevice

设置 USB 数据读写的超时：CH375SetTimeout

设置独占使用当前 CH375 设备：CH375SetExclusive

设定内部缓冲上传模式：CH375SetBufUpload

查询内部上传缓冲区中的已有数据包个数：CH375QueryBufUpload

设定 USB 设备插入和拔出时的事件通知程序：CH375SetDeviceNotify

数据传输 API：

读取数据块（数据上传）：CH375ReadData

写出数据块（数据下传）：CH375WriteData

放弃数据块读操作：CH375AbortRead

放弃数据块写操作：CH375AbortWrite

写出辅助数据（辅助数据下传）：CH375WriteAuxData

中断处理 API：

读取中断数据：CH375ReadInter

放弃中断数据读操作：CH375AbortInter

设定中断服务程序：CH375SetIntRoutine

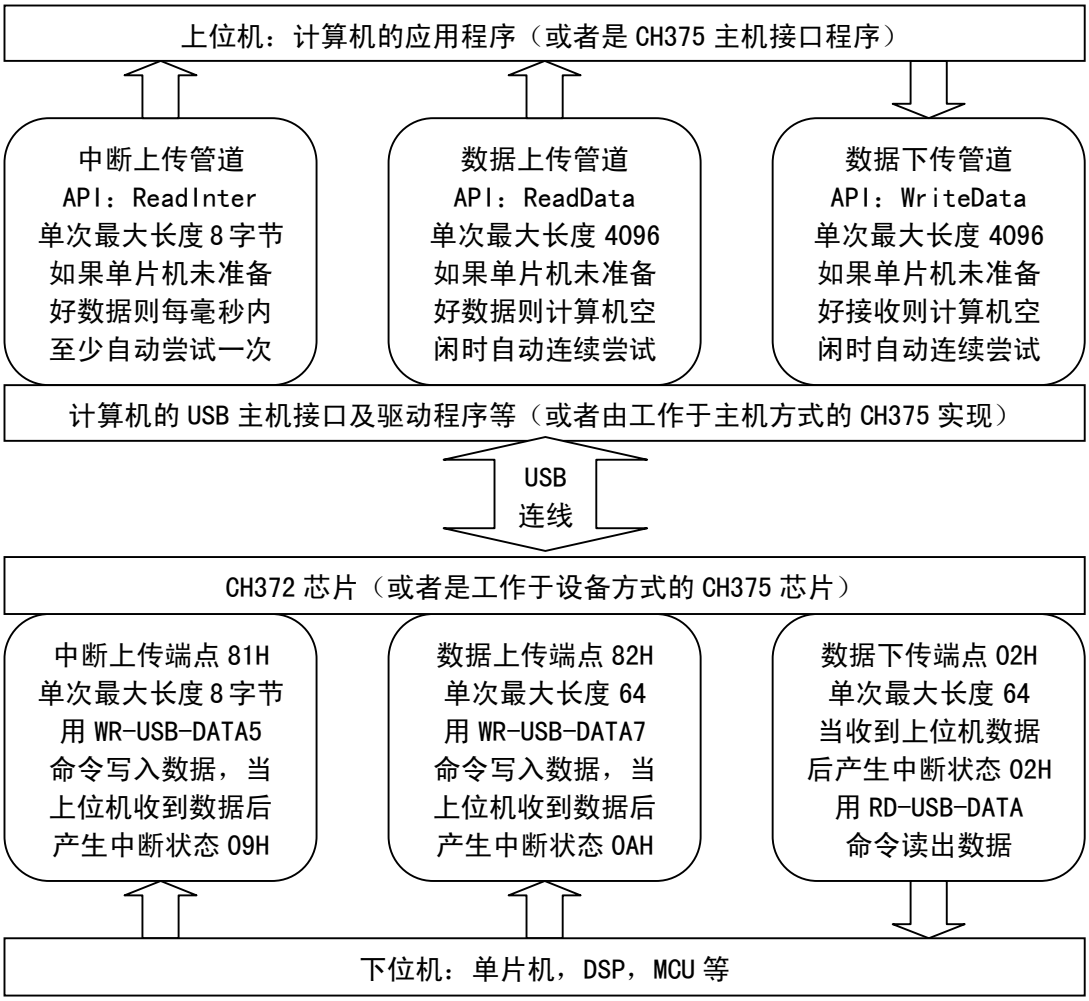
6.6. 端对端的 USB 传输（供应用层设计参考）

CH372 在计算机应用层与本地端单片机之间提供了端对端的连接，在这个基础上，USB 产品的设计人员可以选用两种通讯方式：单向数据流方式、请求加应答方式。前者使用两个方向相反的单向数据流进行通讯，具有相对较高的数据传输速率，但是数据不容易同步；后者使用主动请求和被动应答的查询方式进行通讯，数据自动同步，具有较好的交互性和可控性，程序设计简单，但是数据传输速率相对较低。

6.6.1. 逻辑结构

CH372 提供了 4 个相互独立的端对端的逻辑传输通道，分别称为：数据上传管道、数据下传管道、中断上传管道、辅助数据下传管道。CH372 并未定义各个管道的用途，也未定义其数据格式，所以 USB 产品的设计人员可以根据需要自行定义其用途，并在上位机与下位机之间约定各个传输通道中的数据格式。

下面的逻辑结构图只适用于内置固件模式，所以没有包含默认端点 0。除了图中的 3 个管道之外，CH372 芯片还提供辅助数据下传管道，与数据下传管道类似，但是有 4 个区别：上位机的 API 是 WriteAuxData；下位机的辅助数据下传端点是 01H；单次最大长度是 8 字节；中断状态是 01H。



6.6.2. 单向数据流方式

单向数据流方式使用一个上传数据流和一个下传数据流进行双向数据通讯，两个数据流之间完全独立。

下传数据流是由计算机应用层通过数据下传 API 发起的，CH372 芯片以 64 个字节为一组，将一个较大的数据块分成多组提交给单片机；如果应用层发送 150 个字节的数据块，则单片机会被中断 3 次，前两次各获取 64 个字节，最后一次获取 22 个字节。

上传数据流的发起方式有两种：一种是查询方式，指计算机应用层定期以查询方式发起；另一种是伪中断方式，指单片机以中断数据通知计算机应用层，再由计算机应用层发起。因为 USB 总线是主从式结构，只有在计算机主动联系 USB 设备时，USB 设备才能向计算机上传数据。

上传数据流以查询方式发起的系统中，计算机应用层总是通过数据上传 API 尝试读取数据。当单片机没有数据需要上传时，计算机应用层就会一直等待（如果设置 USB 读超时则会退出），实际上该应用层程序的线程将会被操作系统挂起。当单片机需要上传数据时，应该将数据写入 CH372 批量端点的上传缓冲区中，接着计算机应用层自动取走数据，然后 CH372 芯片以中断方式通知单片机上传成功，以便单片机继续上传后续数据。使用这种方式，建议用 CH375SetBufUpload 设定内部缓冲上传。

上传数据流以伪中断方式发起的系统中，计算机应用层初始化时设置一个伪中断服务程序，然后应用层就不需要再涉及到上传数据流。当单片机需要上传数据时，首先将数据写入批量端点的上传缓冲区中，然后将中断特征数据写入中断端点的上传缓冲区中。在 1 毫秒之内（理论值），与中断特征数据对应的伪中断服务程序被激活，伪中断服务程序通知主程序调用数据上传 API 获得上传数据块。在此期间，单片机将会收到 CH372 芯片通知的两次中断，首先是中断端点上传成功中断，然后是批量端点上传成功中断。

6.6.3. 请求加应答方式

请求加应答方式使用一个下传的主动请求和一个上传的被动应答进行交互式的双向数据通讯，下传与上传一一对应，相互关联。

主动请求是指由计算机应用层下传给单片机的数据请求，被动应答是指在单片机收到数据请求后，上传给计算机应用层的应答数据。所有的通讯都由计算机应用层发起，然后以接收到单片机的应答结束，完整的过程包括：

① 计算机应用层按事先约定的格式将数据请求发送给 CH372 芯片；
② CH372 芯片以中断方式通知单片机；
③ 单片机进入中断服务程序，获取 CH372 的中断状态并分析；
④ 如果是上传，则释放当前 USB 缓冲区，然后退出中断程序；
⑤ 如果是下传，则从数据下传缓冲区中读取数据块；
⑥ 分析接收到的数据块，准备应答数据，也可以先退出中断程序再处理；
⑦ 单片机将应答数据写入批量端点的上传缓冲区中，然后退出中断程序；
⑧ CH372 芯片将应答数据返回给计算机；
⑨ 计算机应用层接收到应答数据。

7、参数

7.1. 绝对最大值（临界或者超过绝对最大值将可能导致芯片工作不正常甚至损坏）

名称	参数说明		最小值	最大值	单位
TA	工作时的环境温度	CH372B 或 C 芯片，VCC=5V，外部时钟	-40	85	℃
		CH372B 或 C 芯片，VCC=3.3V，外部时钟	-40	85	
		CH372C 芯片，VCC=5V 或 3.3V，内置时钟	-20	70	
TS	储存时的环境温度		-55	100	℃
VCC	电源电压（VCC 接电源，GND 接地）		-0.5	6.5	V
VIO	输入或者输出引脚上的电压		-0.5	VCC+0.5	V

7.2. 电气参数（测试条件：TA=25℃，VCC=5V，不包括连接 USB 总线的引脚）

（如果电源电压为 3.3V，则表中所有电流参数需要乘以 40％的系数）

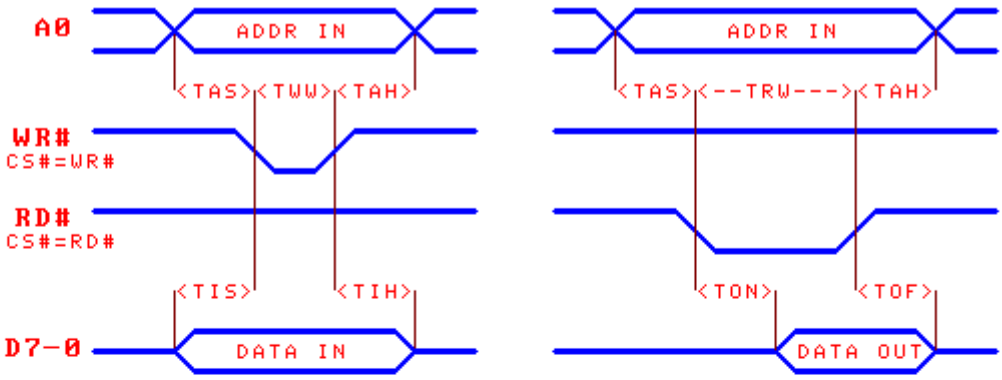
名称	参数说明		最小值	典型值	最大值	单位
VCC	电源电压	CH372B/C 芯片,V3 不连 VCC	4.2	5	5.4	V
		CH372B/C 芯片,V3 连接 VCC	3.1	3.3	3.6	
ICC	工作时的总电源电流	CH372B 芯片，VCC=5V		10	30	mA
		CH372C 芯片，VCC=5V		7	25	
		CH372B 芯片，VCC=3.3V		6	15	
		CH372C 芯片，VCC=3.3V		4	13	
ISLP	低功耗状态的电源电流 I/O 引脚悬空/内部上拉	VCC=5V		0.15	0.25	mA
		VCC=3.3V		0.05	0.15	
VIL	低电平输入电压		-0.5		0.7	V
VIH	高电平输入电压		2.0		VCC+0.5	V
VOL	低电平输出电压（4mA 吸入电流）				0.5	V
VOH	高电平输出电压（4mA 输出电流）		VCC-0.5			V
IUP	内置上拉电阻的输入端的输入电流		40	80	160	uA
VR	电源上电复位的电压门限		2.1	2.6	3.0	V

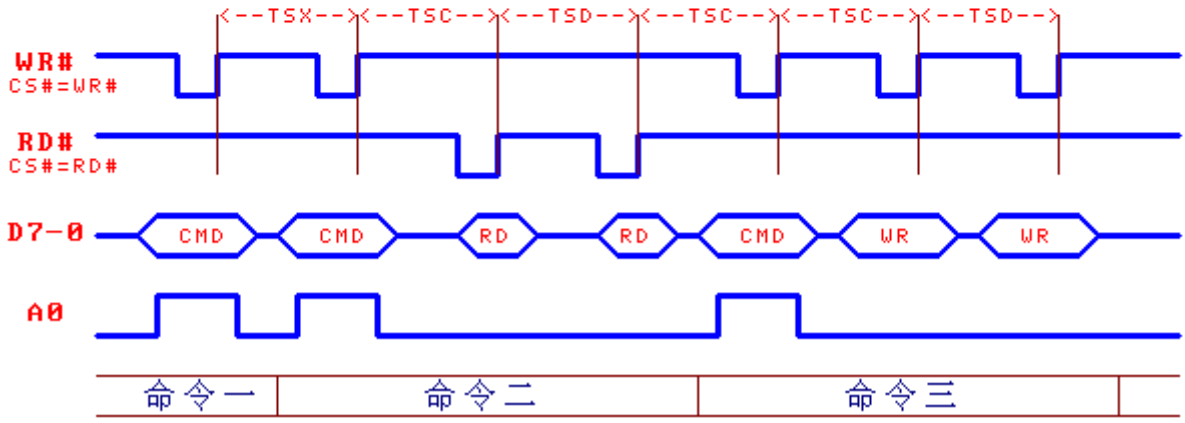
注：在 CH372 芯片复位期间 INT#引脚只能提供 80uA 的高电平输出电流。

7.3. 时序参数（测试条件：TA=25℃，VCC=5V，参考附图）

（RD 是指 RD#信号有效并且 CS#信号有效，RD#=CS#=0 执行读操作）
（WR 是指 WR#信号有效并且 CS#信号有效，WR#=CS#=0 执行写操作）

名称	参数说明	最小值	典型值	最大值	单位
FCLK	XI 引脚的输入时钟信号的频率	11.98	12.00	12.02	MHz
FCLKI	CH372C 内置时钟模式下的内部频率	11.80	12.00	12.20	MHz
TPR	内部电源上电的复位时间	10	30	40	mS
TE1	RESET_ALL 命令的执行时间	0.1	30	40	mS
TE2	SET_USB_MODE 命令的执行时间		10	20	uS
TE3	SET_ENDP?命令的执行时间		2	4	uS
TEO	其余命令的执行时间		1.5	2	uS
TSX	CH372 命令码与命令码之间的间隔时间	1.5			uS
TSC	CH372 命令码与数据之间的间隔时间	1.5		1000	uS
TSD	CH372 数据与数据之间的间隔时间	0.6		1000	uS
TWW	有效的写选通脉冲 WR 的宽度	60		10000	nS
TRW	有效的读选通脉冲 RD 的宽度	60		10000	nS
TAS	RD 或 WR 前的地址输入建立时间	5			nS
TAH	RD 或 WR 后的地址输入保持时间	5			nS
TIS	写选通 WR 前的数据输入建立时间	0			nS
TIH	写选通 WR 后的数据输入保持时间	5			nS
TON	读选通 RD 有效到数据输出有效	0		30	nS
TOF	读选通 RD 无效到数据输出无效	0		20	nS
TINT	收到 GET_STATUS 命令 到 INT#引脚撤消中断请求		1.5	3	uS
TWAK	从低功耗状态退出的唤醒时间	2	6	10	mS





8、应用

8.1. 连接 USB 总线/外部时钟（下图）

P1 是 USB 端口，USB 总线包括一对 5V 电源线和一对数据信号线，通常，+5V 电源线是红色，接地线是黑色，D+信号线是绿色，D-信号线是白色。USB 总线提供的电源电流通常可以达到 500mA，一般情况下，低功耗的 USB 产品可以直接使用 USB 总线提供的 5V 电源。如果 USB 产品通过其它供电方式提供常备电源，那么 CH372 应该与单片机一起使用该常备电源并且断开 USB 总线的电源；如果需要同时使用 USB 总线的电源，那么可以通过阻值约为 1Ω 的电阻 R1 连接 USB 总线的 5V 电源线与 USB 产品的 5V 常备电源，并且两者的接地线直接相连接。

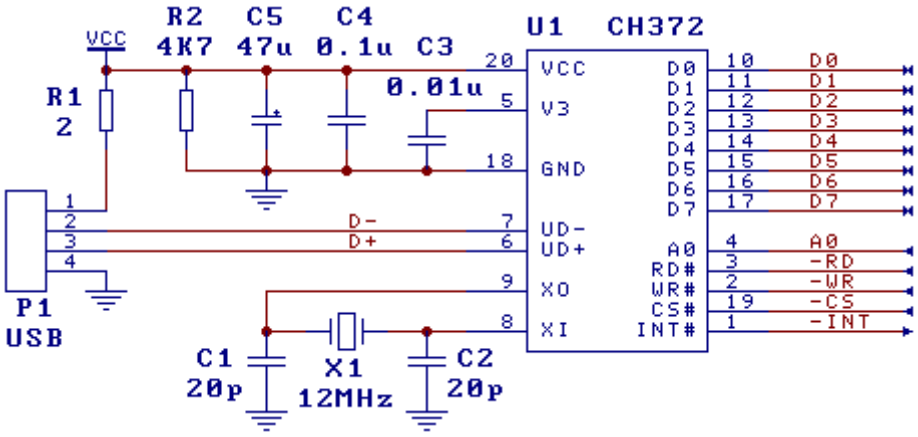
图中，可选电阻 R2 用于在电源断电后将电解电容 C5 中的电能及时释放掉，使 VCC 及时下降到 0V，确保在下次通电时 CH372 能够可靠地上电复位。为使 CH372 能够可靠复位，电源电压从 0V 上升到 5V 的上升时间应该少于 100mS，所以电容 C5 的容量和电阻 R1 的阻值都不能太大。

电容 C3 用于 CH372 内部电源节点退耦，C3 是容量为 0.01μF 到 0.1μF 的独石或者高频瓷片电容。电容 C4 和 C5 用于外部电源退耦，C4 是容量为 0.1μF 的独石或者高频瓷片电容。晶体 X1、电容 C1 和 C2 用于 CH372 的时钟振荡电路。晶体 X1 的频率是 12MHz，C1 和 C2 是容量为 15pF～30pF 的独石或者高频瓷片电容。

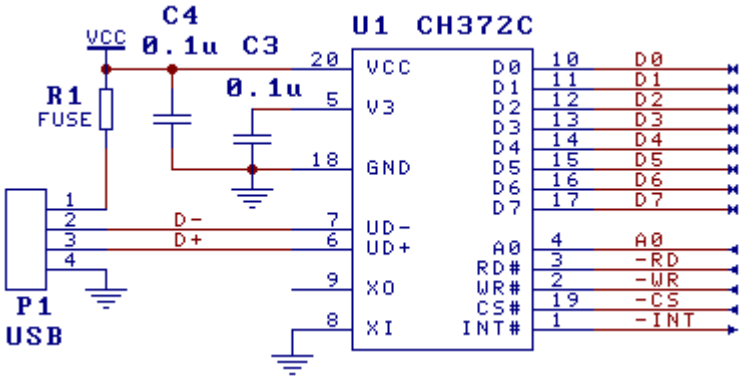
成品板如果用内置时钟的 CH372C 替换 CH372B，那么可以去掉 X1 和 C1，并将 C2 换成 0Ω 电阻。

如果 CH372 的电源电压为 3.3V，那么应该将 V3 引脚与 VCC 引脚短接，共同输入 3.3V 电压，并且电容 C3 可以省掉。

在设计印刷线路板 PCB 时，需要注意：退耦电容 C3 和 C4 尽量靠近 CH372 的相连引脚；使 D+和 D-信号线贴近平行布线，尽量在两侧提供地线或者覆铜，减少来自外界的信号干扰；尽量缩短 XI 和 XO 引脚相关信号线的长度，为了减少干扰，可以在相关元器件周边环绕地线或者覆铜。

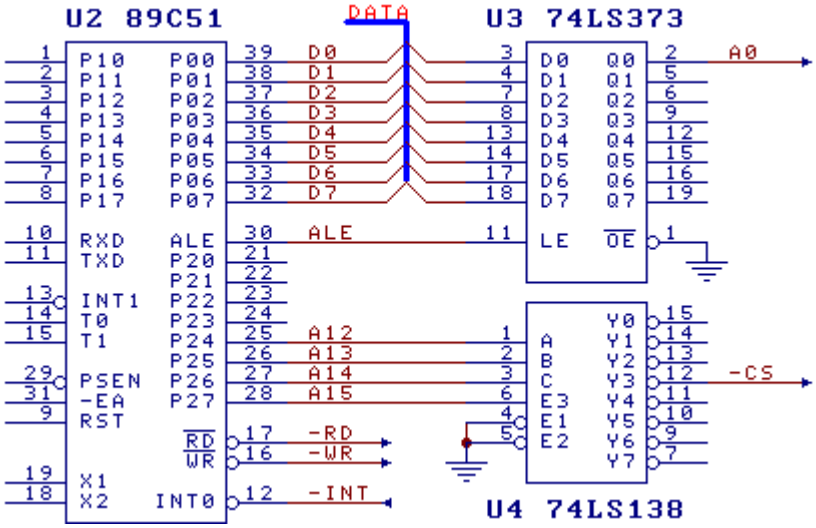


8.2. 连接 USB 总线/内置时钟（下图）



如果使用内置时钟的 CH372C 芯片，那么可以免去外部晶体及振荡电容，电路更简洁。

8.3. 连接单片机总线（总线扩展）



CH372 芯片具有通用的被动并行接口，可以直接连接多种单片机、DSP、MCU 等。在普通的 MCS-51 系列单片机的典型应用电路中，CH372 芯片可以通过 8 位被动并行接口的 D7～D0、-RD、-WR、-CS、A0 直接挂接到单片机 U2 的系统总线上。

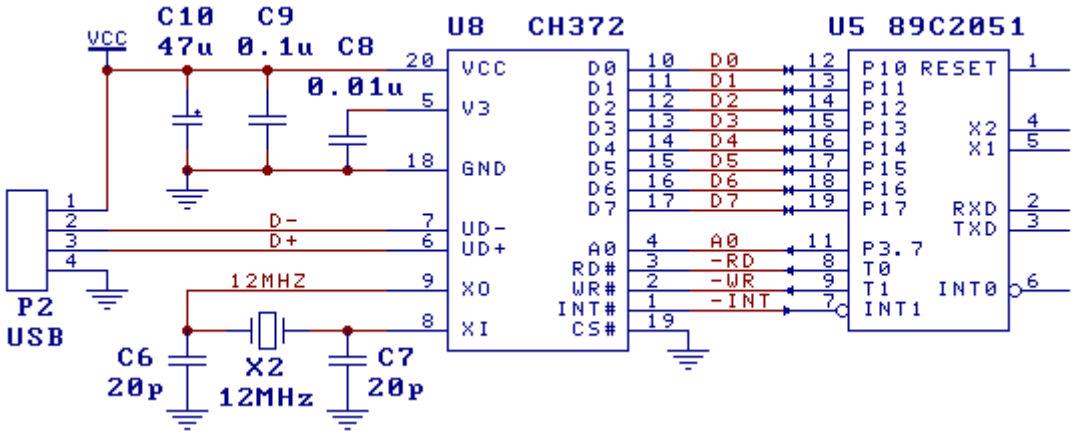
如果 MCS-51 单片机没有用 U3 锁存 A7～A0 地址，那么可以用 U2 的 P20 等引脚驱动 CH372 的地址线 A0，当然单片机程序中的端口地址需要相应修改。U4 用于简单的地址译码，产生所需的片选信号，图中 CH372 芯片的片选地址范围为 B000H-BFFFH，而实际上 CH372 只需要占用两个地址：地址 BXX1H 用于写命令，地址 BXX0H 用于读写数据。

8.4. 连接单片机 I/O（非总线扩展）

在没有外部扩展总线的单片机系统中，单片机也可以用普通的 I/O 引脚模拟出 8 位并口时序操作 CH372 芯片。

在普通的 MCS-51 系列简化单片机的典型应用电路中，CH372 的 CS#固定为低电平，一直处于片选状态，U5 的 P1 端口作为 8 位双向数据总线，在单片机程序中，可以控制各个 I/O 引脚模拟并口时序与 CH372 进行数据交换。

图中电路精简，可以用于软件狗、USB 加密锁、USB 转串口或 RS485 等。



8.5. 单片机接口程序

请参考 CH372 评估板资料中的详细说明。