

# USB 芯片 CH374 的评估板说明及应用参考

版本: 1C

<http://wch.cn>

## 1、概述

CH374 评估板包括两种: CH374S 评估板 (单 USB 端口, 主机与设备复用)、CH374U 评估板 (共三个 USB 端口, 三个主机端口与一个设备端口复用, 支持内置的 3 端口 USB 根集线器 Root-HUB)。本文只介绍 CH374S 评估板, 有关 CH374U 评估板的信息请联系本公司技术人员。

CH374 评估板本身没有单片机, 只是 USB 总线接口芯片 CH374S 外加一些辅助元器件, 对外预留了 8 位并口、SPI 串口以及电源端口等, 用于连接其它单片机主板, 并由单片机主板控制 CH374 进行功能评估。

在单片机的控制下, 可以用于演示 CH374 的 USB-HOST 主机接口功能和 USB 设备接口功能。可以读写 U 盘 (闪盘、USB 闪存盘、USB 外置硬盘、USB 读卡器等); 可以作为 USB 设备与计算机相连接, 例如自行设计的专用 U 盘等; 通过相关程序, 还可以操作其它 USB 设备, 例如 USB 打印机、USB 摄像头、USB 键盘、USB 鼠标等, 或者与另一块 CH37X 评估板进行对连, 互传数据。

本文中涉及 MCS51 单片机的大多数实验和例子程序都是在 CH375 评估板上进行的, 原则上, 应该使用排线等将 CH374 评估板连接到 CH375 评估板的对外端口 P6。只不过, 由于 CH374S 芯片的引脚基本兼容 CH375 芯片, 所以只需要将 CH375 评估板上的 CH375 换成 CH374 即可进行 CH374 芯片的功能评估, 而并不需要通过 P6 端口连接 CH374 评估板。实际上, 简单的做法就是将 CH375 的 DIP28 转换板换成 CH374S 的 DIP28 转换板 (或者换成 CH374T 的 SSOP20 转 DIP28 转换板), 采用 CH374 的 DIP28 转换板与通过 P6 端口外接 CH374 评估板的唯一区别就是 I/O 地址不同。有关 CH375 评估板的硬件请参考 CH375 评估板的资料 CH375EVT.PDF, 网上的压缩文件是 CH375EVT.ZIP。

## 2、评估板的硬件

评估板的原理图和 PCB 请参考 CH374SCH.PDF 文档。下面是元器件说明。

评估板中的主要器件 U1 是 USB 主机和设备双用接口芯片 CH374, 由于该评估板也可以用于 CH375 芯片, 所以图中可能会同时标出 CH375 的有关信号。

晶体 X1 为标准 24MHz, USB 主机比 USB 设备要求更高的频率精确度, X1 的误差要求小于 0.4%, 普通的 24MHz 晶体基本上可以满足要求, 另外, 振荡电容 C1、C2 的容量也会少量影响振荡频率。强烈建议缩短相关引线的长度, 以减少干扰。CH374 也支持 12MHz 晶体, 但需修改程序。

电容 C3 用于向 CH374 提供额外的上电复位, 容量为 0.01 $\mu$ F 到 1 $\mu$ F, 电容 C3 是可选器件。电容 C4 用于内部电源节点退耦从而降低 USB 传输过程中的 EMI, 容量为 4700pF 到 0.1 $\mu$ F, 可以选用普通的 103 贴片电容 0.01 $\mu$ F。如果实际产品电路中有  $\mu$ P 监控电路, 那么应该为 CH374 和单片机提供同一个复位信号, 并且去掉 C3。

P4 是 USB 端口, 既可以用于 USB HOST 主机方式, 也可以用于 USB DEVICE 设备方式, 电阻 R1 用于限制输出给外部 USB 设备的电流, 避免在 U 盘等 USB 设备刚插入时造成电源电压的短时间下降, 甚至引起 CH374 或者单片机非正常复位或者内部 RAM 数据错误。如果是 USB 外置硬盘, 那么应该将电阻 R1 换成直流电阻较小的电感, 或者另外用一组 5V 电源直接提供更大的工作电流 (500mA 以上) 给外置硬盘。另外, USB-HOST 插座的电源退耦电容 C9 的容量不能太小, 容量大些 (应该大于 100 $\mu$ F) 可以减少在 USB 设备刚插入时的电源电压的波动。

P1 是 8 位并口的信号端口, 用于连接单片机的并口, 并由外部对本评估板提供 5V 电源, 外部也可以向 RSTI 引脚提供可选的硬件复位信号。并口的必要信号包括 D0-D7、A0、RD#、WR#、CS#以及 GND, 而 INT#是可选的。注意, 应该尽量缩短本评估板与单片机之间的信号线的长度, 最长不能超过 15cm, 否则需要使用一个信号间隔一个地线的专用排线。使用长线做临时实验可以降低 CH374 工作电压。

P2 是 SPI 串口的信号端口, 用于连接单片机的 SPI 接口, 并由外部对本评估板提供 5V 电源, 外部也可以向 RSTI 引脚提供可选的硬件复位信号。SPI 的必要信号包括 SS#、SCK、SDI、SDO 以及 GND, 而 INT#是可选的。注意, 应该尽量缩短本评估板与单片机之间的信号线的长度, 最长不能超过 15cm, 否则需要使用一个信号间隔一个地线的专用排线。

P3 是预留的信号端口，用于 CH375 芯片，默认不做任何连接。

J3 用于选择 CH374 芯片工作电压，短接 1-2 脚时为 5V 电压，短接 2-3 脚时为 3.3V 电压。默认为 5V 电压，但当单片机的工作电压等于或者低于 3.3V 时，可以为 CH374 选择 3.3V 电压。当 CH374 芯片的工作电压为 5V 时，J2 必须断开，当 CH374 芯片的工作电压为 3.3V 时，J2 必须短接。

J1 默认是断开或者选择 2-3 脚短接，但是不同的单片机程序可能需要 J1 做出调整。例如，单片机程序可以要求 CH374 仅在 UEN 引脚为高电平时才输出 USB 信号，那么 J1 需选择 1-2 脚短接。

J5 和 J6 默认是断开的，用于选择 CH374 工作于 8 位并口方式。如果将 J5 和 J6 分别短接，那么上电或者硬件复位后的 CH374 将工作于 SPI 串口方式。J4 默认是断开的，仅在 CH374 工作于 3 线 SPI 串口方式时才需要将 J4 短接，便于使 SDO 和 SDI 合用一个引脚。

对于单片机读写 U 盘的应用，单片机系统中必须提供至少 620 字节的 RAM。

有些例子程序可能会用单片机的串口输出调试状态信息，如果需要显示这些监控信息，可以由将单片机串口经过 RS232 电平转换后连接到计算机使用串口监控/调试工具软件查看。如果使用 CH375 评估板，那么可以将 J2 连接到计算机串口；如果计算机没有串口，或者串口已经被其它设备占用，那么可以由 USB 转串口芯片 CH341 提供仿真串口。

CH375 评估板内部器件工作于 5V 电源电压时，必须加上电阻 R0 并去掉 3.3V 稳压器 D4，工作于 3.3V 电源电压时，必须加上稳压器 D4 并去掉电阻 R0。默认是 5V 电源。当使用 CH374 的 DIP28 转换板时，CH375 评估板的 JP6 与 CH374 评估板的 J1 的功能相同，JP7 可以用于由 CH374 芯片的 CK0 引脚为单片机提供时钟信号。

### 3、示例程序

产品光盘或者网站上提供了 CH374 的一些 C 语言示例程序，可以用作设计参考，另外，还可以参考 CH374 的 U 盘文件读写模块中的多个 C 和 ASM 示例程序。这些示例大多数是以 MCS51-C 语言编写，将硬件相关部分局部修改后，基本上可以适用于其它类型的单片机或者 DSP。有关 USB 设备方式的应用可以参考 CH372 评估板资料中的说明。

如果在 CH375 评估板中做 CH374 的 DIP 转换板测试，那么 CH374 的端口地址是（可用其它地址）：

```
unsigned char volatile xdata CH374_IDX_PORT_at_ 0xBDF1; /* CH374 索引端口的 I/O 地址 */
unsigned char volatile xdata CH374_DAT_PORT_at_ 0xBCF0; /* CH374 数据端口的 I/O 地址 */
#define CH374_INT_WIRE INT0 /* 连接 CH374 的 INT#引脚，用于查询中断状态，可选 */
```

如果在 CH375 评估板中做 CH374 评估板的测试，那么 CH374 的端口地址是（也可以用其它地址）：

```
unsigned char volatile xdata CH374_IDX_PORT_at_ 0xEDF1; /* CH374 索引端口的 I/O 地址 */
unsigned char volatile xdata CH374_DAT_PORT_at_ 0xECF0; /* CH374 数据端口的 I/O 地址 */
#define CH374_INT_WIRE INT1 /* 连接 CH374 的 INT#引脚，用于查询中断状态，可选 */
```

如果用于非 MCS51 单片机，需要修改以下与单片机硬件有关的子程序：

可以参考 PARA.C、PARA\_HW.C、PARA\_SW.C、SPI.C、SPI\_HW.C、SPI\_SW.C、SPI3\_SW.C 等例子程序

```
void Write374Index( unsigned char Index ) { /* 向索引端口写入索引地址 */
    CH374_IDX_PORT=Index;
}
void Write374Data( unsigned char dat ) { /* 向数据端口写入数据，索引地址自动加 1 */
    CH374_DAT_PORT=dat;
}
unsigned char Read374Data ( void ) { /* 从数据端口读出数据，索引地址自动加 1 */
    return( CH374_DAT_PORT );
}
```

#### (1) 按照文件格式读写 U 盘

这种方式下，单片机需要分析 U 盘的已有文件系统，并按照其格式进行数据读写，操作复杂，速度也有所下降，很多操作需要数百条语句才能完成。写入数据后的 U 盘插到计算机中，计算机可以直接看到相应的文件，并可以直接读写其中的数据。

为了兼容 WINDOWS 下的文件系统，大多数 USB 闪存盘常用的文件系统是 FAT12 或者 FAT16，部分大容量 U 盘或者 USB 外置硬盘，常用的文件系统是 FAT32。

由于文件系统对于单片机而言，处理起来比较复杂，所以我们通过子程序库提供 U 盘文件操作的 API。有关这些 API 的进一步说明可以参考 CH374HF.PDF 文档。

以下是示例程序，节选自 CH374HFT.C

```
strcpy( mCmdParam.Open.mPathName, "\\C51\\CH374HFT.C" );
/* 要操作的文件名称是 CH374HFT.C，该文件在 C51 子目录下，C 语言的双斜杠实际是单斜杠 */
CH374FileOpen( ); /* 打开文件 */
mCmdParam.ReadX.mSectorCount = 4; /* 读取 4 个扇区（2KB 数据） */
mCmdParam.ReadX.mDataBuffer = 0x0200; /* 将读出数据放到 0200H 开始的缓冲区中 */
CH374FileReadX( ); /* 以扇区为单位从文件读取数据，如果文件比较大，可以多次读取 */
CH374FileClose( ); /* 关闭文件 */
/* 数据处理，示例程序是将原文件中的小写字符转换为大写字符，然后保存到一个新文件中 */
strcpy( mCmdParam.Create.mPathName, "/NEWFILE.TXT" );
/* 新建一个文件，名称是 NEWFILE.TXT，在根目录下，下面以字节为单位读写该文件 */
CH374FileCreate( ); /* 新建文件并打开，如果文件已经存在则先删除后再新建 */
strcpy( mCmdParam.ByteWrite.mByteBuffer, "准备写入新文件中的字符串" ); /* 数据 */
mCmdParam.ByteWrite.mByteCount=strlen(mCmdParam.ByteWrite.mByteBuffer); /* 长度 */
CH374ByteWrite( ); /* 以字节为单位向文件写入数据块，单次长度不超过 MAX_BYTE_IO */
unsigned short Dat = 12345; /* 下面将变量 Dat 格式化为字符串输出到文本文件中 */
mCmdParam.ByteWrite.mByteCount = /* 转换后的字符串长度由下面的 sprintf 返回 */
sprintf(mCmdParam.ByteWrite.mByteBuffer, "变量 Dat=%d", Dat); /* 输出文本 */
CH374ByteWrite( ); /* 以字节为单位向文件写入数据块，单次长度不超过 MAX_BYTE_IO */
CH374FileClose( ); /* 字节模式下关闭文件会自动计算文件长度 */
```

## (2) 操作其它 USB 设备

被操作的 USB 设备应该支持 USB1.1 或者 USB2.0 的 12Mbps 全速或者 1.5Mbps 低速。

CH374 能够自动检测 USB 设备是否连接，当 USB 设备连接后，建议先进行 USB 总线复位，然后读取 USB 描述符、设置 USB 地址、设置 USB 配置、分析 USB 设备等，对于不同的 USB 设备，其操作流程及方式可能不同。

详细的例子可以参考 CH374/EVT/PUB/EXAM 目录下的 HOST 和 HOST\_HUB 程序。

## 4、关于 U 盘文件接口

### 4.1. U 盘文件读写模块（特点：只需要极少的硬件资源，省事，快速应用）

模块中包含 CH374 和一个单片机以及必要的 RAM 缓冲区，系统的主单片机只要将文件名提供给模块（数据流模式可以不需要文件名），就可以读写文件。模块支持 FAT12、FAT16 和 FAT32 文件系统的 U 盘（含闪存盘和外置硬盘，下同），支持文件搜索、读写、查询、删除、新建和子目录新建等操作。

如果单片机系统的 RAM 少于 620 字节，那么只能使用 U 盘文件读写模块。

最新的详细说明请参考 U 盘模块的详细使用说明 CH374HM.PDF。

### 4.2. U 盘文件级子程序库（特点：批量时硬件成本较低，应用灵活）

硬件上只需要在原单片机系统中增加一个 CH374 芯片，综合成本较低。使用 U 盘文件级子程序库，单片机系统需要具备以下硬件资源：不少于 4KB 到 7KB 的程序空间，不少于 620 字节的随机存储器 RAM，不同的子程序库对内部 RAM 的占用稍有不同。

最新的详细说明请参考 U 盘文件级子程序库说明 CH374HF.PDF。

### 4.3. U 盘文件级子程序的 C 源程序（特点：可以结合实际应用做进一步优化）

源程序可以适用于常用的 8 位、16 位、32 位等多种单片机和 DSP，特别针对硬件资源有限的中低端单片机进行优化。目前，源程序只针对批量用户提供免费使用授权，本公司保留相关一切著作权。目前可以提供的普及版源程序，支持 FAT16 文件系统，支持文件读写、删除、新建等，经过简单优化。

另外，还可以提供基于子程序库的 U 盘文件读写模块的 C 语言源程序。

## 5、其它说明

### 5.1. 硬件方面

- (1) 如果用频率计测量 CH374 的振荡频率要考虑探头电容对频率的影响，应该用 10:1 高频探头，普通晶体基本上可以满足晶体 X1 的 0.4% 精度要求。简单应用中也可以使用陶瓷晶体降低成本。
- (2) 为了降低电磁辐射，并减少来自外界的干扰，晶体 X1 的金属外壳应该接地，晶体 X1 以及电容 C1、C2 应该尽量靠近 CH374，相关的 PCB 走线应该尽量短，并且可以在周边环绕接地线或者敷铜。USB 数据线 D+ 和 D- 应该平行布线，长度保持差不多，两侧可以环绕接地线或者敷铜，更详细的说明可以参考 USB 规范。
- (3) 对于需要频繁带电插拔 USB 设备的应用以及静电较强的环境，建议在电路中增加 USB 信号瞬变电压抑制器件，为 CH374 的 USB 引脚 D+ 和 D- 提供进一步的保护。详细说明可以参考 CH375 和 CH374 电路设计注意事项 README.PDF 文档。
- (4) 如果操作 USB 外置硬盘或者耗电较大的 USB 闪存盘，需要考虑其电源供应，确保提供足够的工作电流，否则在其插入过程以及读写过程中会导致电源电压波动，甚至导致 CH374 以及单片机复位。建议在电源与地之间并联较大的电解电容，或者为 USB 插座单独提供一组 5V 电源，或者用直流电阻较小的电感代替电阻以减少对 CH374 的影响。
- (5) 如果需要减小电流消耗，可以在空闲时使 CH374 进入低功耗睡眠挂起状态，当有 USB 设备插拔时 CH374 会自动唤醒。在 CH374 睡眠期间，应该使 CH374 的各个 I/O 引脚（除 RST1 引脚）处于悬空或者高电平状态，避免产生不必要的上拉电流。CH374 的 SLP 引脚还可以用于控制外部电路。

### 5.2. 单片机程序方面

- (1) 有些 USB 设备包括 U 盘，在刚插入 USB 插座后，不能立即进入工作状态，所以主程序可以在检测到 USB 设备连接后，等待数百毫秒再对其进行操作。详细做法可以参考 EXAM1 和 EXAM13 等。
- (2) 单片机的 C 语言效率比汇编语言低，以 MCS51 单片机为例，纯 C 语言数据读写速度可能只有汇编语言速度的一半，所以，根据需要部分子程序可以嵌入汇编。如果是频率为 24MHz 的标准 MCS51 单片机，以“单 DPTR 和 P2+R0 复制”方式传输文件数据（附加说明：MCS-51 单片机每复制一个字节至少需要 8 个机器周期即 4μs），那么传输速度约为每秒 100KB 到 200KB，文件越零碎，传输速度越慢。

### 5.3. 单片机读写 U 盘

- (1) 应用建议：用量不多的用户可以使用 U 盘读写模块，综合成本较低；一般用户可以使用 U 盘文件级子程序库，硬件成本远低于 U 盘读写模块，只需要在原有系统中增加一只 CH374 芯片。
- (2) 以扇区为单位的文件读写子程序，速度较快，操作效率高，但是如果文件长度不是扇区的整数倍，那么就需要自行考虑文件长度的问题。
- (3) 以字节为单位的文件读写子程序，占用 RAM 相对较少，能够自动处理文件长度，使用较为方便，但是速度比以扇区为单位的文件读写慢，并且频繁地向 U 盘中的文件写入零碎的数据，会缩短 U 盘中闪存的使用寿命（因为闪存只能进行有限次擦写）。另外，每次可以读写的最大字节长度还与 mCmdParam 结构的总长度有关。
- (4) 在 WINDOWS 2000 或者 XP 下的控制面板/管理工具/计算机管理中有磁盘管理工具，可以将 U 盘格式化指定为 FAT12、FAT16 或者 FAT32 文件系统，当总容量除以分配单元大小后的结果小于 4085

时是 FAT12，大于 65525 时是 FAT32，否则是 FAT16。对于已经格式化过的 U 盘，可以使用命令行工具 CHKDSK 分析，点击[开始]选择[运行]，输入“CHKDSK 盘符:”分析指定盘符的磁盘，显示分配单元的大小和总数。分配单元较大时，通常读写效率稍高，分配单元较小时，通常会节约磁盘容量。建议：如果 U 盘容量小于 16MB，可以使用 FAT12 文件系统格式；如果 U 盘容量小于 512MB，优先使用 FAT16 文件系统格式；如果 U 盘容量大于 512MB，可以使用 FAT32 文件系统格式。

(5) 关于如何节约 CH374 子程序库所占用的 RAM

CH374 子程序库总共占用约 620 个字节的 RAM，其中对于 MCS51 单片机还分为：

外部 RAM 需 550 字节，其中 512 字节用于磁盘数据缓冲 DISK\_BASE\_BUF（通常是 512，如果需要支持大扇区 U 盘则缓冲区更大），17 字节用于 mBOC，剩余的为一些外部变量

内部 RAM 需 70 字节(概数)，用于各种全局变量和局部变量，

在调用子程序时需要使用 CmdParam 结构，最少为 16 字节内部 RAM，默认为 30，

合计内部 RAM 需要最少 70 字节，默认情况下为 85 字节。

CH374 的子程序库经过多次优化，基本上已经很难再节约 RAM 了，对于 MCS51 单片机的子程序库，已经部分采用汇编语言优化，既优化运行速度，又优化 RAM 占用数

如何节约 CmdParam 结构所占用的 RAM，对于 MCS51 是内部 idata\_RAM

当没有调用 CH374 子程序库时或调用返回后，CmdParam 结构可以用于其它任何用途，从而可以与其它应用程序共用 CmdParam 结构所占用的几十个字节的 RAM。这个方法完全没有副作用，实际上 CmdParam 所占用的 RAM 长度还可以自行定义。

如何节约 mBOC 结构所占用的 RAM，对于 MCS51 是外部 xdata\_RAM

当没有调用 CH374 子程序库时或调用返回后，mBOC 结构可以用于其它任何用途，从而可以与其它应用程序共用 mBOC 结构所占用的 17 个字节的 RAM。这个方法完全没有副作用。

如何节约 DISK\_BASE\_BUF 缓冲区所占用的 RAM，对于 MCS51 是外部 xdata\_RAM

如果定义 DISK\_BASE\_BUF\_LEN 为 0 则禁止独占 DISK\_BASE\_BUF 缓冲区，而是由应用程序在调用 CH374Init 之前将其它缓冲区的起始地址置入 pDISK\_BASE\_BUF 变量。CH374 子程序库所用的 pDISK\_BASE\_BUF 缓冲区可以用于其它任何用途，但是在用完后必须调用 CH374DirtyBuffer 通知子程序库“缓冲区被用过”，以免子程序库仍然以为其中数据有效，通过这种方法就可以让这些 RAM 与其它应用程序共用。副作用是降低效率，当 CH374 子程序库在需要磁盘数据时必须重新从 U 盘中读取，而无法直接使用缓冲区中的数据，因为缓冲区中的数据已经无效。

如何节约 CH374 子程序库占用的所有 RAM，对于 MCS51 则包含内部 RAM 和外部 RAM

在没有调用 CH374 子程序库时或调用返回后，其它应用程序可以使用子程序库占用的任何 RAM，但是在用完后，必须将全局变量 CH374DiskStatus 置为 0(也就是 DISK\_UNKNOWN)，以通知 CH374 子程序库完全重来，这种方法可以节约子程序库占用的所有 RAM，但是效率较低，将 CH374DiskStatus 清零会使 CH374 子程序库完全重新分析 U 盘的文件系统，可能需要多花几十毫秒甚至更多时间。另外，也可以调用 CH374SaveVariable 备份所有变量。

(6) 下表是单片机通过 U 盘文件级子程序库读写 U 盘文件时的速度，是在以 30K 大数据块读取 5MB 大文件时测得的，在小数据块读取时的速度要下降为 70%到 95%。写速度与 U 盘本身的技术方案有关，在大数据块写入时的速度通常是读速度的 70%到 90%，在小数据块写入时的速度通常再下降一半。

单片机硬件类型	文件读写的数据复制方式	读速度/字节每秒
工作于 24MHz 系统时钟的标准 12 时钟 MCS51 单片机 (非 12 时钟需参考计算)	常规方法：单 DPTR 复制	156K
	依赖新硬件：双 DPTR 复制	195K
	单 DPTR 和 P2+R0 复制	234K
工作于 40MHz 系统时钟的标准 12 时钟 MCS51 单片机 (非 12 时钟需参考计算)	常规方法：单 DPTR 复制	258K
	依赖新硬件：双 DPTR 复制	319K
	单 DPTR 和 P2+R0 复制	399K
工作于 16MHz 的 AVR 单片机	常规方法，并口 I/O 总线	700K
工作于 60MHz 的 ARM 单片机	常规方法，并口 I/O 总线	800K
大多数 8 位单片机	手工汇编优化可以提高速度	100K 到 700K
大多数 16 位或 32 位单片机	通常不需要手工汇编优化	200K 到 900K