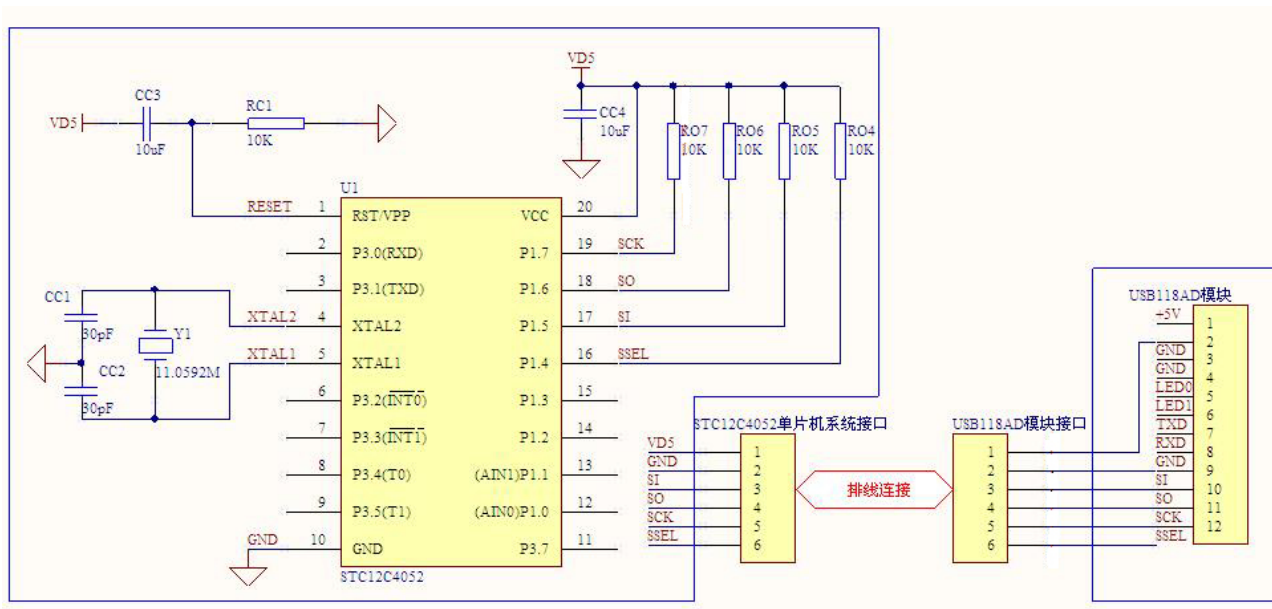


## 1、USB118AD模块SPI接口的硬件接线图



由于单片机的类型特别多，对于可以仿真或者支持在线调试的单片机来说可能会好一些，但是对于没有这两项功能中的任何一种来说（如STC系列）就比较麻烦，因为我们看着也摸不着，但是我们可以借助于串口调试助手（建议使用产品附赠光盘中的“串口调试助手V2.1.exe”）来解决这个问题。

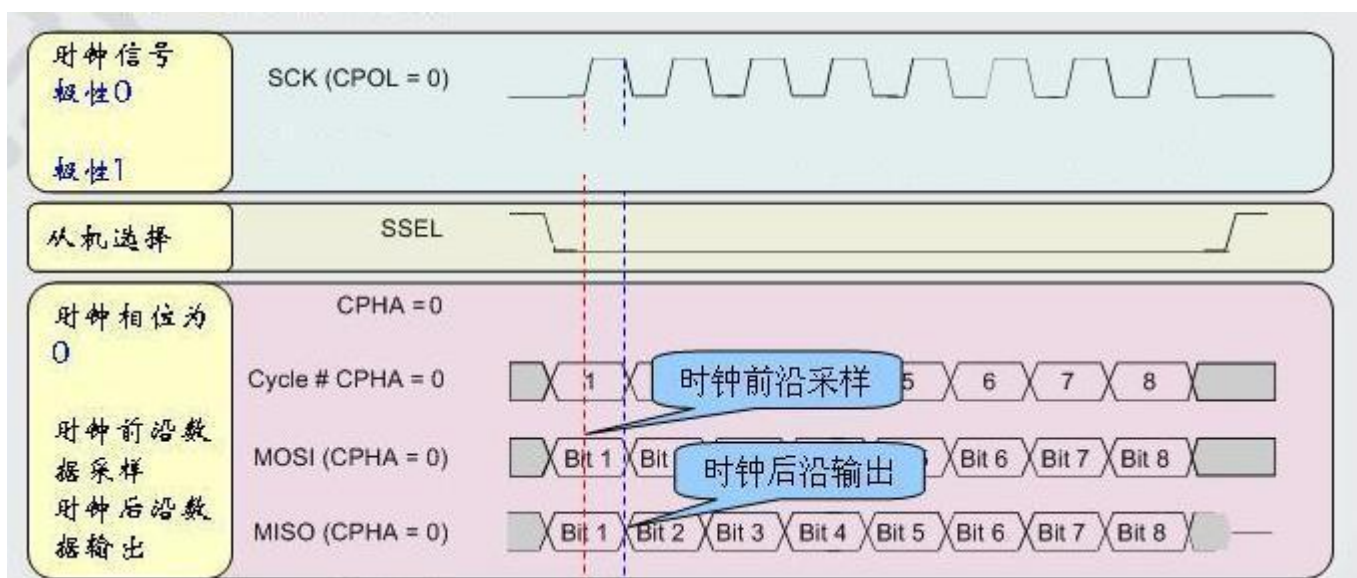
- ◆ **检查硬件连接是否正确**，单片机 SO 对应 USB118AD 模块 SO，单片机 SI 对应 USB118AD 模块 SI，参考上述硬件连接图正确连接。其中 SSEL、SCK、SI、SO 信号电平符合 3.3V 电源系统，同时还兼容 5V 电源系统。也就是说用户单片机无论是 3.3V 还是 5V 系统，信号线都可以直接连接。USB118AD 模块波特率的设置，详细参考《USB118AD 型 USB2.0 接口 U 盘读写模块用户手册 V4.0.pdf》中第 8 页。
- ◆ 对上述所说不能仿真和不支持在线调试的单片机，我们也可以通过模块的串口与 PC 机通信利用串口调试助手来监视单片机发送和接收的数据。

- ◆ 单片机是否能够发送数据，可以设定单片机一直为发状态，然后用示波器观察 SI 和 SCK 线的波形，就可以确认单片机是否能够正常发送数据。
- ◆ 单片机是否能够接收数据，可以设定单片机一直为发状态，然后用示波器观察 USB118AD 模块的 SO

和 SCK 线的波形, 如果没有波形出现, 就说明单片机不能接收到数据, 这是单片机程序接收部分有问题, 我们可以检查相关代码, 找到问题加以修改。

- ◆ 接收到的数据总是 0xFF, 因为 SPI 通信是“挤过来挤过去”的。此时不能发送下一个命令, 只能发送一些无意义的数, 直到读到的数据不是 0xFF, 才表示命令已经处理完成。

## 2.3 USB118AD模块SPI接口通信时序图



注: USB118AD 模块选择 CPOL=0、CPHA=0, 即 8 位数据传输、高位在前、时钟上升沿有效。时钟上升沿数据有效。USB118AD 模块工作在从(Slave)模式下, 所以单片机为主(Master)模式。模块的高速率为 2Mbps。

## 2.4 嵌入USB118AD模块的程序设计

### 2.4.1 用户的程序流程

- 写一个命令给USB118AD 模块。发送命令时, 发送命令中第一个数据之后, 这里注意需要延迟一段时间(10us)再发送第二个数据。也就是发送完0xAA后, 就延迟一段时间, 再发送后面的数据的时候就不用延迟了。
- 循环读数据, 等待命令处理完毕。读数据过程中, 起始时接收到的都是0xFF, 此时不能发送下一个命令, 只能发送一些无意义的数, 比如写0xFF。直到读到的数据不是0xFF时, 才表示命令已经处理完成。

以下是常用步骤简述, 可以根据实际情况进行调整

### 2.4.2 初始化

- 调用SPI初始化函数
- 用户单片机上电后延时5~10秒与模块通信, 强烈建议执行此步骤

### 2.4.3 发送检测命令

- 发送检测设备命令USB118R\_Detect

- 等待模块回复USB118T\_Detect命令，确认U盘连接正常

#### 2.4.4 新建文件

- 发送USB118R\_Create命令，新建文件，文件名称为“8+3”的格式。USB118AD模块只支持“8+3”格式，即文件名最多为8个字符（字母或者数字），文件扩展名最多为3个字符（常用的扩展名txt、doc和xls）。例如：

1、建立一个“TEST.TXT”文件，如下：

//建立TEST.TXT文件命令

```
uchar code Create[66]={0xaa,0xbb,
0x07, 0x54, 0x45, 0x53, 0x54, 0x20, 0x20, 0x20, 0x20, 0x54, 0x58, 0x54, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x63, 0x4F, 0xFD, 0x32, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

//打开TEST.TXT文件命令

```
uchar code Open[66] = {0xaa, 0xbb,
0x06, 0x54, 0x45, 0x53, 0x54, 0x20, 0x20, 0x20, 0x20, 0x54, 0x58, 0x54, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

2、建立一个“TSUNAMI1.XLS”文件，如下：

//建立TSUNAMI1.XLS文件命令

```
uchar code Create[66]={0xaa,0xbb,
0x07, 0x54, 0x53, 0x55, 0x4E, 0x41, 0x4D, 0x49, 0x31, 0x54, 0x58, 0x54, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x63, 0x4F, 0xFD, 0x32, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

//打开TSUNAMI1.XLS文件命令

```
uchar code Open[66] = {0xaa, 0xbb,
0x06, 0x54, 0x53, 0x55, 0x4E, 0x41, 0x4D, 0x49, 0x31, 0x54, 0x58, 0x54, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

- 等待模块回复USB118T\_Create命令，判断是否成功

#### 2.4.5 写入数据

- 文件创建成功后默认是打开的，此时无需再发送USB118R\_Open命令，这样可以节省时间，直接发送USB118R\_Write命令写入数据（每次最大4Kbyte）。注意在USB118R\_Write命令中，数据紧接在指令的后面，即“命令+数据”，例如要将下面表格Data中的数据写入文件中，如下：

//写文件命令（48，0为数据长度 48个字节（低位在前））

```
uchar code Write[66]={0xaa,0xbb,
0x09, 0x00, 48, 0, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20,
0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20,
0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20,
0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20};
```

```
uchar code Data[48]={  
0xB4, 0xEF, 0x09, 0xCC, 0xA9, 0x09, 0xB5, 0xE7, 0x09, 0xD7, 0xD3, 0x0D, 0x55, 0x09, 0x53, 0x09,  
0x42, 0x09, 0x31, 0x0D, 0x31, 0x09, 0x38, 0x09, 0x41, 0x09, 0x44, 0x0D, 0x0D, 0x31, 0x09, 0x32,  
0x09, 0x33, 0x09, 0x34, 0x0D, 0x35, 0x09, 0x36, 0x09, 0x37, 0x09, 0x38, 0x0D, 0x0D, 0x0D, 0x0D};  
程序设计如下:
```

```
Send_Data(Write);           //写文件  
Senddata(Data);            //写入Data表中数据  
if(Receive_Data(0x0C,0) == 1) //0C写数据命令代码, 命令是否成功执行, 是则继续执行  
{  
    delay_xms(200);  
    break;  
}
```

- 写入数据命令可以多次连续发送

#### 2.4.6 连续写入数据

- 数据写入文件后指针自动移动到文件末尾, 再次写入数据时, 直接发送USB118R\_Write命令写入数据, 方式同第一次写入数据一样
- 写入数据命令可以多次连续写入, 例程中我们连续写入了两次

#### 2.4.7 保存文件

- 保存文件, 文件写入完毕后, 发送USB118R\_Detect命令, 判断是否成功
- 切记此步骤必须有, 否则可能文件中没有数据

### 3、USB118AD模块SPI接口通信程序流程图

注：如果只存储一次数据的话，直接保存数据，如图所示。例程中我们连续写入两次。

