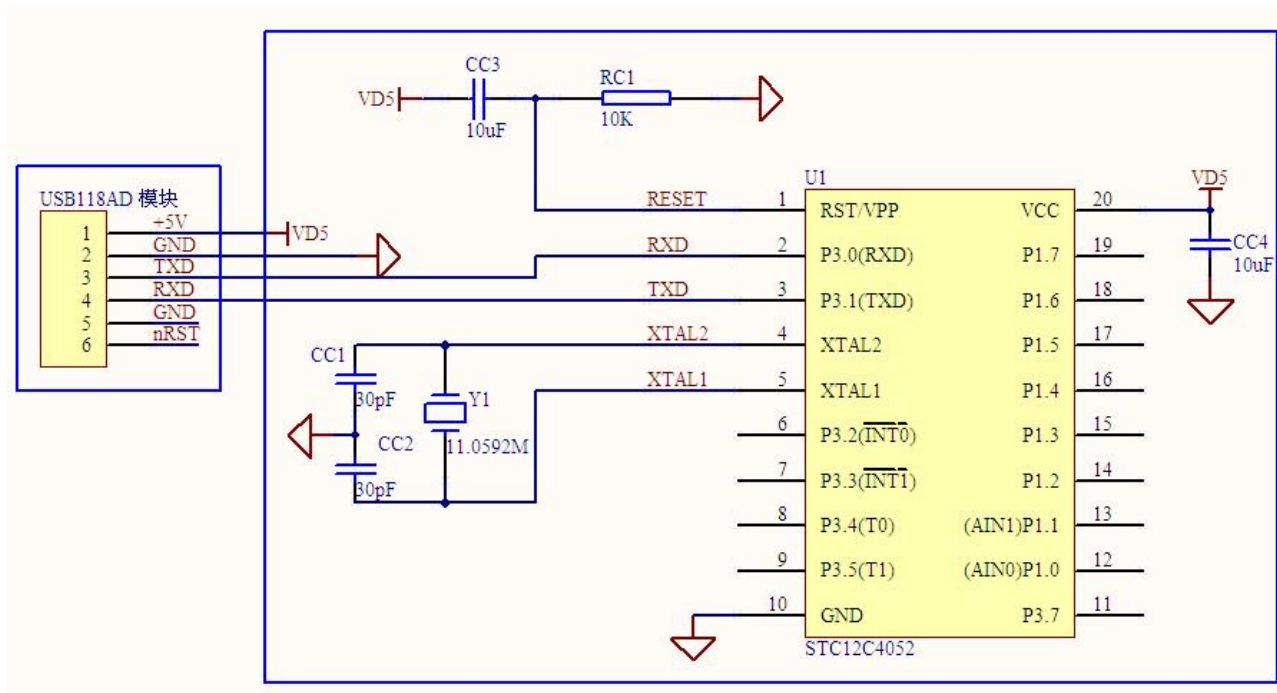


USB118AD 嵌入式 U 盘读写模块 TTL 串口接口应用

1、USB118AD模块TTL串口接口的硬件接线图



2、关于USB118AD模块TTL串口通信编程的一些建议

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，因为我们看着不着也摸不着，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。

2.1 调试之前首先排除硬件问题

- ◆ **检查硬件连接是否正确**，单片机 TXD 对应 USB118AD 模块 RXD，单片机 RXD 对应 USB118AD 模块 TXD，其中注意电平的转换，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。
- ◆ 由于有些用户用的单片机是+3V 供电，并且其 I/O 口不能承受+5V 的电压，所以可能开始能够收到，后来就收不到数据了，这很可能已经将您的单片机的串口接收 RXD 口烧坏了，检测方法可以借助与串口调试助手检测 I/O 好坏，参考上述硬件连接图正确连接。
- ◆ **串口被并联使用，可能收不到数据**。对上述所说的单片机不能仿真和不支持在线调试的单片机，我们有些用户在利用串口调试助手来监视单片机发送的数据的时候，把 PC 机上的 RS232 和 USB118AD 并联在了单片机的串口上，而串口是不能够进行并联使用的。所以我们在使用 PC 机上的串口调试软件监视时，应做如下操作：

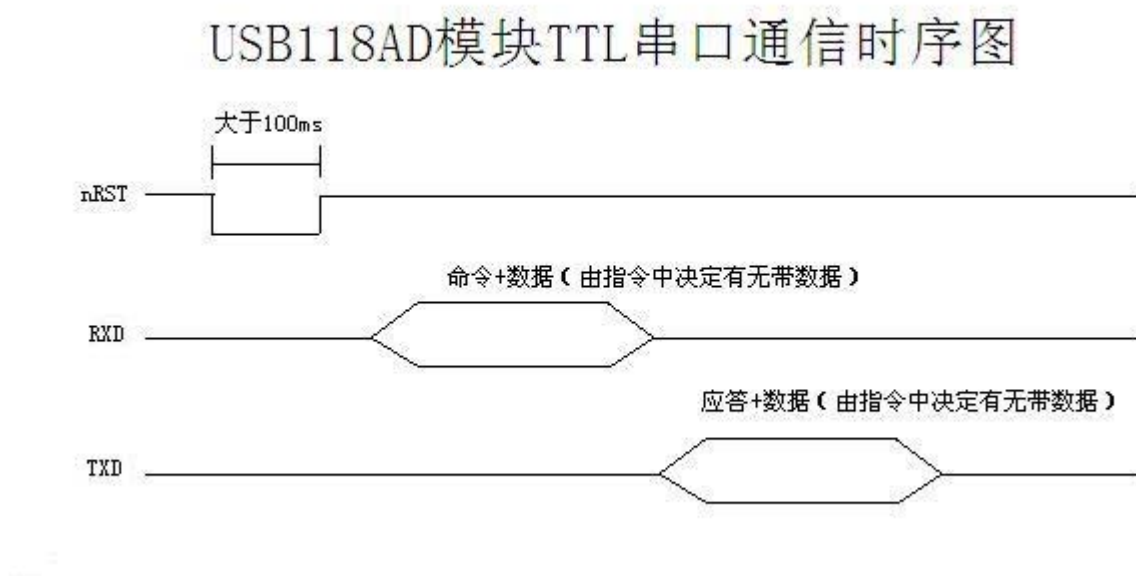
a、监视单片机发送数据。把单片机与USB118AD连接好，然后RS232发送不接悬空，只把接收（RXD）连接在单片机的发送（TXD）数据线上，即发送命令（由单片机发往 USB118AD模块，USB118AD模块的串口接收），这样您就可以监视您的单片机发送的命令是什么和您软件设计的是否一样。

b、监视单片机接收数据。把RS232的接收（RXD）连接在单片机的接收（RXD）数据线上，这样您就可以监视USB118AD在您发送命令之后返回的命令参数，即应答命令（由USB118AD模块发往单片机，USB118AD模块的串口发送）。

2.2 检查排除基本的软件问题

- ◆ 您的软件波特率设置是否正确（可以利用串口调试助手检验）。
- ◆ 单片机的接收可能存在问题，可以利用串口调试助手查看，单片机有没有发送数据或者发送数据是否正确，也可以查看有没有数据返回或者返回的数据是多少，如果串口调试助手中有数据返回而单片机接收不到，那么在确保硬件接线正常的情况下，很可能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。

2.3 USB118AD模块TTL串口通信时序图



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
- 循环读数据，等待命令处理完毕

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

- 调用串口初始化函数
- 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, 0x0D, 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  
Send_Data(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TTL串口通信程序流程图

