

基于 STM32 微处理器的 GPRS 数据传输技术的研究

郑优讯, 李宗伯

(国防科学技术大学, 湖南 长沙 410000)

摘要: 为了实现用微处理器进行 GPRS 数据传输的应用, 可以采用微处理器搭载 GPRS 模块并进行配置的解决方案。介绍了用 STM32 微处理器搭载 SIM900A 的 GPRS 模块实现数据传输, 并简要描述 STM32 底层配置以及数据传输的实现方法。

关键词: STM32; SIM900A; USART; 发送检验; 接收检验

中图分类号: TP393

文献标识码: A

文章编号: 1674-7720(2012)21-0060-02

Research of GPRS data transmission technique based on STM32 MCU

Zheng Youxun, Li Zongbo

(National University of Defense Technology, Changsha 410000, China)

Abstract: For the purpose of implementing the application of GPRS data transmission using MCU, MCU equipped with GPRS module and configure it can be used as a solution. This article realizes data transmission based on STM32 MCU equipped with GPRS module of SIM900A, and briefly describes the underlying configuration of STM32 and the implementation of data transmission.

Key words: STM32; SIM900A; USART; send verify; receive verify

STM32 是意法半导体 (ST) 公司推出的基于 ARM Cortex-M3 内核的主流产品之一, 它是 ST 公司专门针对要求高性能、低功耗和低成本嵌入式应用而设计的, 目前已在各领域得到广泛的应用。SIM900A 是 SIMCom 公司主推的一款紧凑型的 GSM/GPRS 双频模块产品, 它以性能稳定、外观精巧及性价比高的优点深受广大工程师青睐。

本文通过对 STM32 底层配置以及数据传输的研究, 介绍 STM32 主要的底层配置, 并着重介绍数据传输的实现。通过关键步骤的程序源代码的介绍, 阐述实现数据传输的细节以及注意事项。该方法对其他项目或芯片有一定的实现价值和参考价值, 且简单可靠, 具有普遍性和通用性。

1 STM32 底层配置

为了实现 STM32 单片机与 SIM900A 模块之间的数据命令的传输, 本文以串口为例, 先搭建开发平台, 在工程中加入相应的库函数以及配置文件, 然后配置时钟以及串口相应的输入输出 GPIO 接口。在配置的同时, 需要针对自身的原理图进行编写, 才能保证配置正确无误。

这样, 基本的开发平台就搭建起来了。

1.1 串口配置

在开发平台搭建起来之后, 就可以对串口进行配置了。配置速率为 115 200 b/s, 字长为 8 bit, 1 bit 停止位, 串口模式为输入与输出模式, 最后, 初始化相对应的串口。初始化串口之后, 打开串口的中断响应函数, 即 USART_ITConfig (USART2, USART_IT_RXNE, ENABLE) (以串口 2 为例), 然后使能相对应的串口, 这样串口函数就基本配置完成了。需要注意一点, 有些程序可能在传输的时候出现首位丢失。这个问题涉及到 USART 的机制。硬件复位之后, USART 的状态位是置位的 (置 1, 表示已经发送完毕), 而此时数据可以进行正常发送。当一帧数据发送后, 由硬件将该位置位。而清除 TC 位 (置 0) 是由软件来完成的, 通过先读 USART_SR, 再写 USART_DR 将该位清除。但是程序在发送第一帧数据的时候, 并没有进行读 USART_SR, 而是直接进行写 USART_DR, 因此 TC 标志位还是置 1, 并没有清除。当发送第一帧数据之后, 用 USART_GetFlagStatus () 检测状态返回的是已经发送完毕, 程序就会马上发送下一帧数

网络与通信 Network and Communication

据,因此第一帧数据就会被第二帧数据覆盖了,这样就看不到首位数据。根据这种情况,可以在每次传输之前或之后清除传输完成标志位,即 USART_ClearFlag(USART2, USART_FLAG_TC)。

1.2 中断配置

配置完串口之后,将对 NVIC 进行配置。首次配置中断分组,然后选择串口的中断,即 NVIC_InitStructure.NVIC_IRQChannel=USART2_IRQn(以所使用固件库的定义为主)。

再设置抢占式中断优先级和响应式中断优先级,然后使能中断和初始化。以上的配置必须结合自身的情况,设计出最优的中断分组和优先级,以保证程序响应中断的速度。中断后所做的内容在 stm32f10x_it.c 文件里配置,下文将会详细阐述。

2 实现细节

实现 GPRS 数据传输的原理是:STM32 解析一串数据或命令,然后通过串口或其他方式一个字符一个字符地发送给 SIM900A 模块, SIM900A 接收到数据之后再通过 SIM 卡发送到服务器。当 SIM900A 接收到数据时,立即响应中断,按照中断所设置的方式进行数据处理。此时,就需要通过发送检验和接收检验来控制数据的传输。

2.1 发送检验

由于 STM32 逐个字符地将数据发送给 SIM900A 模块,因此必须保证数据的正确性与连贯性。如果在发送的时候响应中断或者进行任务调度,则发送将作废,从而导致程序出错,所以开发者必须警惕该类的错误出现。

发送数据或者命令的时候,可以将数据通过参数传给发送函数,由发送函数统一控制,发送完成之后再返回一个发送完成标志位,告知调用函数者发送已完成。源程序如下:

```
void USART_Send_Byte(char MyData){ //发送字符函数
    USART_ClearFlag(USART2, USART_FLAG_TC);
    //清除标志位,如上所述
    USART_SendData(USART2, MyData); //发送数据
    while (USART_GetFlagStatus(USART2, USART_FLAG_TC)
    ==RESET); //等待发送完成
}
void USART_Send_Str(char*s){ //发送字符串
    int i;
    int len = strlen(s)-1; //字符串长度
    for(int i=0; i<len; i++)
    USART_Send_Byte(s[i]); //循环将字符串发送出去
    if(s[i]==0x0a){ //判断发送是否结束
        SendCFFlag=TRUE;
        //如果为真,则发送完成标志位置为真
    }else{
```

```
        USART_Send_Byte(s[i]); //如果为假,则发送出去
    }
}
```

2.2 接收检验

当 SIM900A 有数据返回或者有数据通过 SIM900A 接收到下位机时, STM32 会立即响应中断来接收数据。此时就要在中断函数中进行一系列处理。以 SIM900A 为例, SIM900A 模块返回的命令都是以 "\r" + "\n" + "\0" 结尾,因此检验传输结束可以根据它进行判断。在中断响应函数(即 stm32f10x_it.c 文件里)的 USART2_IRQHandler 函数可以设置如下:

```
void USART2_IRQHandler(void)
{
    if (USART_GetITStatus(USART2, USART_IT_RXNE) !=
    RESET){
        //将接收字符存入接收缓冲区 RxBuffer
        RxBuffer [ReceCounter ++ ] = (char)USART_ReceiveData
        (USART2);
        //判断是否接收结束
        if (RxBuffer [ReceCounter] == '\0' && RxBuffer
        [ReceCounter-1]==0x0A &&
        RxBuffer[ReceCounter-2]==0x0D){
            ReceCFFlag=TRUE;
        }
        USART_ClearITPendingBit(USART2, USART_IT_RXNE);
    }
}
```

该函数的基本思路是:将 USART 接收到的字符逐个存入缓冲区,然后判断缓冲区最后 3 个字符是否为 SIM900A 的结束标识符。如果为假,继续接收;如果为真,则将接收完成标识符置为真。当接收完成标识符为真时,说明接收完成,接下来就可以进行数据处理了。

2.3 命令函数实现方法

下面将以 AT+CIPSEND 为例,阐述发送数据的细节。通过初始化模块、开启网络、建立接入点和建立 TCP 连接之后,就可以开始发送数据。实现源代码如下:

```
u8 GPRS_Send(void){
    u8 i=0;
    u8*p;
    USART_SendToGPRS("AT+CIPSEND\r\n"); //发送命令
    Delay_ms(500); //延时 500 ms
    p=LookFor_Str(RxBuffer, ">");
    //查找是否有">"符号,如果有,则可以发送数据
    if(p!=0){
        p=0;
        memset(RxBuffer, 0, BufferSize); //清空接收缓冲区
        USART_SendToGPRS(GPRSSendData); //发送数据
        Delay_ms(500);
```

```

Delay_ms(500);
Delay_ms(500);
p=LookFor_Str(RxBuffer,"SEND OK");
if(p!=0){ //判断是否发送成功
//发送成功操作
return 1;
}else {
//发送失败操作
return 0;
}
}
}

```

该函数的基本思路是：首先发送命令，然后查找是否有“>”符号，如果有，则说明可以开始发送数据。一段延时之后查找接收缓冲区是否有“SEND OK”字眼，有则说明发送成功，没有则表示发送失败。可以根据判断作进一步的操作。命令的用法详见 SIM900A 配套的 AT 命令手册。有以下三点需要注意：

(1) 在本文测试程序中需要先获取 IP 然后才能建立 TCP 连接，这是由 SIM900A 机制决定的。所以如果开发者不能建立 TCP 连接，除了测试网络是否正常、服务器是否正确配置之外，还需在程序中先获取 IP，命令为 AT+CIFSR。

(2) 可以先获取 SIM900A 的状态，命令为 AT+CIPSTATUS。根据状态来判定进行哪些操作，可以减少运行量，简化代码，从而减少运行时间，提高运行效率。详见 SIM900A 配套的 AT 命令手册。

(3) 延时的设定需要具体问题具体分析。例如，初始化 SIM900A 模块的时候，只需延时 500 ms，模块返回的信息就接收到了，而接收来自服务器的信息时，有时由

于信号问题或者巨大的数据量可能要延时久一点，而此时代就需要开发者自行进行测试。延时的准确设置，可以在保证数据正确性的同时减少延时时间，从而提高程序的运行效率。

本文通过对 STM32 微处理器串口的设置以及中断的配置来阐述 STM32 微处理器底层的配置，再通过 SIM900A 的发送和接收数据实现 GPRS 的数据传输技术，从而实现 STM32 微处理器接入互联网。在接收检验实现中，可以根据接收是否完成只作一次判断，从而减少中断运行时间。而 SIM900A 是 GSM/GPRS 双频模块，还可以实现通话、收发短信、HTTP 及 FTP 传输等诸多功能，通过更深入的研究，可以最大限度地挖掘出该模块的实用价值，从而为电子产品提供更多的应用功能。

参考文献

- [1] 谭浩强.C 程序设计(第三版)[M].北京:清华大学出版社,2005.
- [2] 刘波文.ARM Cortex-M3 应用开发实例详解[M],北京:电子工业出版社,2011.
- [3] 意法半导体STM32F10x-StdPeriph-Lib[EB/OL].http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/stm32f10x_stdperiph_lib.zip,[2012-3-02].
- [4] 芯讯通无线科技有限公司.SIM900_AT command manual_V1.05 [EB/OL].http://wm.sim.com/Sim/FrontShow_en/wireless/detail.aspx?cid=6&nid=770,[2011-10-31].

(收稿日期:2012-05-02)

作者简介:

郑优讯,男,1988年生,硕士研究生,主要研究方向:嵌入式系统和物联网。

李宗伯,男,1969年生,副教授,主要研究方向:体系结构和嵌入式系统。