

基于STM32的UART-WiFi模块的设计与实现

Design and implementation of UART-WiFi module based on STM32

潘琢金, 李冰, 罗振, 杨华

PAN Zhuo-jin, LI Bing, LUO Zhen, YANG Hua

(沈阳航空航天大学 计算机学院, 沈阳 110136)

摘要: 传统工业控制领域中存在嵌入式设备无法联网、远程控制不方便、数据无法远程传输等问题。在分析了传统解决方案串口转以太网方案的基础上, 提出了基于STM32微控制器和88w8385无线网络控制器的UART-WiFi接口模块解决方案, 该方案中将STM32F103微控制器与88w8385无线网络控制器通过SDIO接口连接, 并移植LwIP开源协议栈。利用该UART-WiFi模块, 传统的串口设备就能够无线接入网络并接收数据, 从而实现向串口设备传输数据、对串口设备远程无线控制的功能。实验结果表明该模块无线通信性能好、稳定性强。

关键词: STM32微控制器; UART-WiFi模块; 88w8385网络控制器; LwIP协议栈

中图分类号: TP368

文献标识码: A

文章编号: 1009-0134(2015)04(上)-0127-04

Doi: 10.3969/j.issn.1009-0134.2015.04(上).38

0 引言

随着物联网的日益普及、信息共享程度的不断提高, 嵌入式设备接入网络的需求越来越高。然而传统工业控制领域中, 带有串口的嵌入式设备并不具备联网能力。要使此类设备接入网络, 目前解决办法是采用串口转以太网服务器的方案, 通过串口转以太网服务器将嵌入式设备产生的串口数据转换为以太网帧, 进行远程传输。但以太网存在布线困难、灵活性差等缺点, 对于作业环境复杂的工业控制领域, 铺设有线以太网比较困难。WiFi作为一种无线局域网形式, 能够克服以太网布线困难、灵活性差的缺点。综合考虑上述问题, 本文提出并设计实现了基于STM32F103微控制器和88W8385无线网络控制器的UART-WiFi无线模块, 完成了系统硬件的设计, 并实现了LwIP(lightweight IP)协议栈和WiFi驱动的移植。

1 系统整体架构

UART-WiFi模块在用户侧留有标准RS232接口, 可使串口嵌入式设备便捷的接入无线网络。该UART-WiFi模块由三部分组成: 串口数据收发模块、无线网络数据收发模块和数据处理模块。系统模块组成原理框图如图1所示。

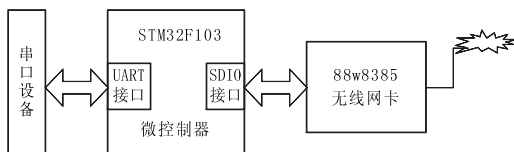


图1 系统架构框图

带有串口的嵌入式设备产生的数据, 通过UART串口传给UART-WiFi模块, 该模块将串口数据打包成WiFi数据包, 通过无线网络数据收发模块传输WiFi数据包, 实现串口数据的网络化传输。反之, 也可远程无线控制嵌入式串口设备。

2 系统硬件设计

2.1 系统设计硬件框图

系统硬件主要由核心微控制器、WiFi无线网络控制器模块、RS232串口模块、电源模块、时钟模块、JTAG调试接口以及可扩展接口等部分组成。其硬件组成结构框图如图2所示。其中微控制器采用STM32F103RET6芯片, WiFi无线网络控制器采用Marvell的88w8385芯片。

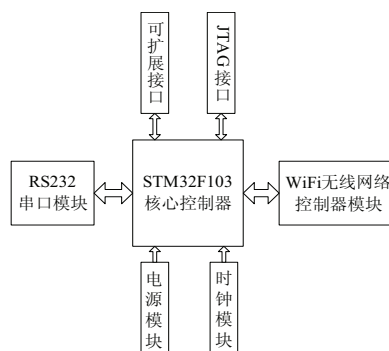


图2 系统硬件组成结构框图

2.2 STM32F103微控制器

STM32是意法半导体基于ARM 32位Cortex-M处理器架构设计的一系列微控制器的总称。该系列微控制

收稿日期: 2014-11-19

基金项目: 辽宁省自然科学基金(2013024002); 辽宁省教育厅基金(L2013063)

作者简介: 潘琢金(1962-), 男, 教授, 硕士, 研究方向为计算机检测与控制、嵌入式系统、嵌入式Internet等。

器具具有功耗小、运行稳定、成本低、速度快等优点，被广泛用于嵌入式开发。本文设计的UART-WiFi模块选用该系列中的STM32F103RET6微控制器芯片作为核心。它是同类产品中性能最高的产品，时钟频率达到了72MHz。它有丰富的外设接口，如定时器、ADC、SPI、SDIO、USB、IIC等。本模块主要关注的部分有：

1) 3个USART接口，该模块中只需其中的1个，外接RS232标准的串口模块，用作与嵌入式设备的串口相连；

2) 3个通用16位定时器，用于定时；

3) 片内512K的Flash存储器可用于存储程序和数据，多达64KB的SRAM能够以CPU的时钟速度进行读写；

4) SDIO接口，用于外接88w8385无线网络控制器，实现无线收发数据的功能。

本设计方案中，STM32F103核心控制器具有控制和数据处理的作用。作为控制器，它控制接收数据的转发。在数据处理方面，它将串口接收的数据进行解包、分组，然后将其打包成所需的数据帧格式。

2.3 88w8385 WiFi无线网络控制器

Marvell 88w8385是全球首个成功将中央处理器和大量片上存储器嵌入单一芯片进行完成主机卸载处理的片上系统。同时，88w8385体积小、适合小体积系统集成。该芯片集成了IEEE802.11a/b/g MAC/基带/射频WLAN，降低了使用的复杂度，节省了开发成本。88w8385 WiFi无线网络控制器具有优异的电源管理、灵活的系统配置、工作在全球通用的2.4GHz频段等特性。含有丰富的外部接口，如CF接口、USB接口、SDIO接口、SPI接口等。本设计方案中，采用88w8385无线网络控制器的SDIO接口，将其与STM32F103控制器连接，微控制器STM32F103可直接访问SDIO接口的寄存器，通过SDIO接口进行固件的下载，及数据的传输。

2.4 RS232接口芯片MAX3232

STM32F103芯片内部集成了3个串口模块USART1、USART2和USART3，本设计方案使用USART1接收RS232串口数据。嵌入式串口设备的RS232数据线通过MAXIM公司的MAX3232芯片，连接到微控制器STM32F103的USART1上。MAX3232芯片完成TTL信号和RS232信号的相互转换。

2.5 电源及时钟模块

电源模块是用来给STM32F103控制器供电的电路。本设计方案选用AMS1117-3.3芯片，输入5.5V电压，输出3.3V稳定电压，用于给STM32F103供电。时钟模块为STM32F103输入时钟信号。STM32F103的电源管理模块内嵌8MHz的RC振荡器，因此选用8MHz的晶振。

此外，外接JTAG接口用于调试和下载；利用可扩展接口外接指示灯用于指示UART-WiFi模块的工作状态。

3 系统软件设计

软件系统的设计主要包括三部分：系统的初始化、LwIP协议栈和WiFi驱动的移植、数据收发过程的设计。

3.1 系统初始化

系统初始化主要包括STM32F103RET6微控制器的初始化、串口的初始化、SDIO的初始化、88w8385芯片的初始化及LwIP协议栈的初始化。系统初始化过程如图3所示。

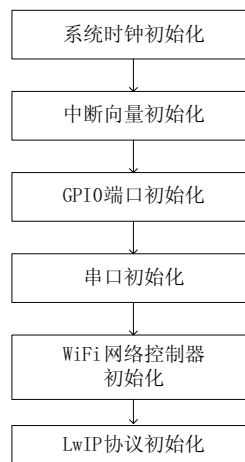


图3 初始化过程

STM32F103RET6微控制器的初始化包括时钟的初始化、中断向量的初始化、端口引脚的初始化等。初始化过程主要包括：调用RCC_Configuration()来初始化系统时钟；调用GPIO_Configuration()来初始化各端口引脚；调用NVIC_Configuration()来映射中断处理函数。

串口初始化由函数USART_Configuration(USART1)完成，通过初始化完成配置串口1到控制台的工作。方案中，串口主要用于连接外部嵌入式设备。同时，将串口设置为全双工异步串行传输模式，并使用定时器产生的波特率时钟。

WiFi网络控制器的初始化主要包括两部分：SDIO设备枚举、加载设备固件，由函数init_marvell_driver()完成。其中stm32_probe()函数完成SDIO设备的枚举，if_sdio_prog_firmware()函数通过SDIO口完成固件的下载。

LwIP协议栈的初始化由Init_lwIP()完成，包括初始化内存堆函数mem_init()，初始化内存池函数memp_init()，数据包管理初始化函数pbuf_init()，网络接口管理初始化函数netif_init()等。

3.2 LwIP协议栈和WiFi驱动移植

LwIP协议栈及WiFi驱动层次结构模型如图4所示。

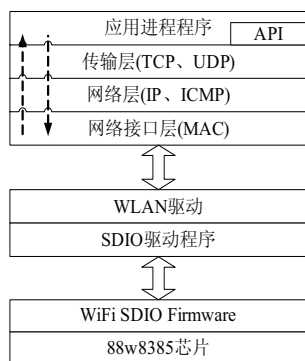


图4 LwIP协议栈及WiFi结构层次图

3.2.1 LwIP协议栈的移植

LwIP是用于嵌入式系统的开源裁剪版TCP/IP协议栈，可运行在无操作系统的平台上，整个协议栈实现了基本的IP、ICMP、UDP、TCP协议。它只需要十几KB的RAM和40K的ROM就可运行，尤其适用于内存资源有限的嵌入式系统。在初始化LwIP协议栈之前，需要将各协议移植到开发板上，使其支持TCP和UDP两种方式。

LwIP源码主要包括以下子目录：存放应用程序接口文件的api子目录、提供核心代码文件的core子目录、提供整个协议栈使用头文件的include子目录、提供网络接口驱动程序基本框架的netif子目录及移植过程中自定义的arch目录。arch包含网络设备驱动、移植需要修改的文件。移植LwIP协议栈，需完成lwipopts.h、perf.h和cc.h三个头文件的定义：

- 1) cc.h: 根据所选编译器和处理器特性完成协议栈内部使用数据类型的定义；
- 2) lwipopts.h: 完成协议栈内核的参数配置，若对某个参数未配置，则使用opt.h文件中的默认值；
- 3) perf.h: 定义与系统统计和测量的宏，该宏与平台处理器密切相关，本设计方案不需要使用任何统计和测量功能，因此文件中的两个宏直接定义为空。

协议栈移植完成后，由于服务器在每次收发数据中都要调用底层驱动，因此该部分代码需要有良好的健壮性和稳定性。本设计方案将发送流程与接收流程分开实现，发送与接收流程如图5所示。

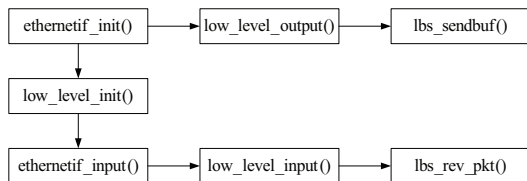


图5 底层程序框架图

发送流程从初始化函数ethernetif_init()开始，它完成网络接口抽象netif的填充并调用链路层发送函数low_level_output()，再调用发送函数lbs_sendbuf()发送数据。

接收流程除初始化无线网络控制器外，需要调用链路层初始化函数low_level_init()。收到的数据经过函数ethernetif_input()，根据帧头携带的上层协议类型判断交给IP协议还是ARP协议处理。

3.2.2 WiFi驱动程序

Marvell 88w8385WiFi模块驱动主要包括WLAN驱动和SDIO接口驱动两部分。WLAN驱动负责将数据通过SDIO接口转发到WiFi硬件或响应WiFi硬件中断，并从硬件中断的缓冲区读取数据，通过驱动程序注册的接口函数，发送到上层应用。

WiFi Firmware是WiFi设备硬件中执行的一段程序，由WiFi驱动将其通过SDIO接口下载到WiFi模块中，主要功能是实现WiFi硬件接口控制、数据缓冲、数据帧的转换、MAC层管理和中断管理等。

3.3 收发过程

串口数据可连续发送，而WiFi数据则必须以数据包为单位发送。因此，本方案涉及到一个数据包长度选取的问题，即：将多少字节的串口数据封装成一个WiFi数据包。针对该问题有两种解决方案，一种是设定数据包长度，当数据达到指定长度时产生中断，将数据打包发送。但如果所产生的数据较少，此种方式会有严重的延时现象。第二种是通过数据包间隔的方式。当串口转WiFi转发器发现串口数据中出现了T毫秒的空闲时间，则认为收到的串口数据可以作为一个WiFi数据包发送了。

综合考虑以上两种方案，本方案采用硬件中断与定时器中断相结合的方式。串口数据到达时产生串口中断，在串口中断处理程序中将串口数据放到缓冲区中，并开始计时，当计时器到达指定值，就将数据发送出去。网口接到数据时，将数据放到另一个缓冲区中，计时器到，将数据从串口发送出去。如图6所示。

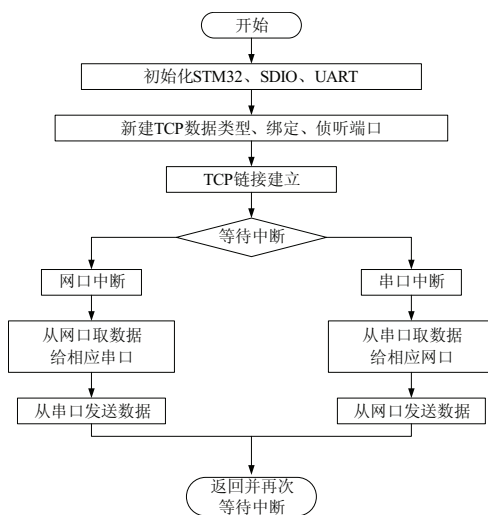


图6 收发流程图

4 测试结果显示

软硬件设计完成之后,对整个UART-WiFi网关模块进行数据的收发测试。首先,当WiFi模块工作在adhoc模式下,成功为笔记本分配ip地址,使用ping命令测试网络,能够正确接收数据包,表明网络正常。其次,使用TCP/UDP Socket调试工具,对UART-WiFi模块外接的工作状态指示灯进行远程控制,TCP/UDP测试结果如图7所示。



图7 TCP/UDP测试

从图7可以看出,WiFi模块服务器的ip地址是192.168.10.10,分配给笔记本的ip地址是192.168.10.100。UDP的端口号均为8080,笔记本发送的LED_OPEN1可以打开WiFi模块上的指示灯,并回送给调试工具LED_OPEN1以表示测试成功。TCP测试与之类似。最后,将Web服务器嵌入到系统中,进行了Web服务器的测试,在网页中输入192.168.10.10对模块进行控制,控制结果通过串口成功显示,从而实现了WiFi转串口的功能。串口显示测试结果如图8所示。

5 结束语

本文针对串口设备数据远程传输不便问题,提出了一种将串口数据转换成WiFi数据包,将数据传入无线传入网络的解决方案,完成了UART-WiFi模块的设计与实现。本方案成本低、功耗小、性能稳定,测试结果表明,该模块能够满足工业控制领域中串口数据无线传入网络的需求。对于作业环境复杂的工业现场,由于其无线的特性,克服布线困难的问题。由于硬件设备成本

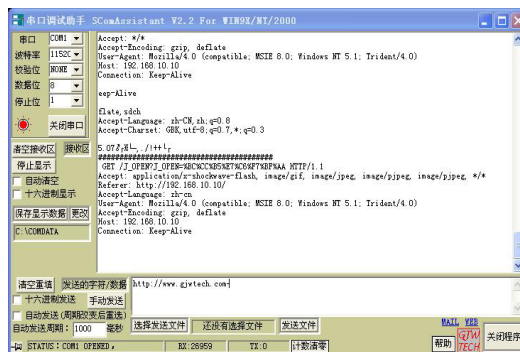


图8 串口显示测试

低、模块性能稳定、应用范围广泛,因此具有广阔的应用前景。

参考文献:

- [1] 项雷军.基于ZigBee、GPRS和TCP/IP协议的无线网关设计[J].微型机与应用,2013,32(8):51-54.
- [2] 王希朝,张毅,程鹏.面向物联网应用的UART-WIFI网关设计[J].微型机与应用,2013,32(8):45.
- [3] 董武,陆利坤,李业丽.基于W5300的以太网和串口数据相互转换系统设计[J].制造业自动化,2013,35(1):139.
- [4] 汤宏萍,王竹平.嵌入式TCP/IP协议栈的设计与实现[J].微电子学与计算机,2008,25(6):168.
- [5] 刘跃军,苏静.基于串口通讯的打包数据的接收方案[J].微计算机信息,2007,24(53):118.
- [6] 范炜,徐洪泽.基于TCP/IP协议的嵌入式多串口网关的设计[J].计算机工程与设计,2008,29(1):80.
- [7] 王晓菊,潘琢金,杜睿.嵌入式网络接口模块设计及协议栈实现[J].沈阳航空工业学院学报,2007,24(1):49.
- [8] Jiang Xiangguo,Qi Yulin,Yang Jiancheng A method of streamline the TCP/IP Protocol Stack at embedded systems[J].2010.147.
- [9] W.Richard Stevens.范建华,译.TCP/IP详解卷1:协议[M].机械工业出版社,2013.
- [10] Scaglia Sergio.潘琢金,译.嵌入式internet tcp/ip基础实现及应用[M].北京航空航天大学出版社,2008.
- [11] In-Su Yoon,Sang-Hwa Chung,Jeong-Soo Kim.Implementation of Lightweight TCP/IP for Small,Wireless Embedded Systems.[J]2009.53.