

# 基于 Socket 的 Linux 与 Windows 下 WiFi 模块间通信的实现

王云亮<sup>1</sup>, 王光林<sup>2</sup>

(1.天津理工大学 天津市复杂系统控制理论及应用重点实验室,天津 300384;  
2.天津理工大学 自动化学院,天津 300384)

**摘要:** 为了实现 Linux 开发环境下的总控制器与 Windows 开发环境下分节点之间的数据进行无线传输,提出用 Socket 的方法解决它们所带 WiFi 模块之间的无线通信。该方法能够实现数据的正常传输,并能把数值准确地显示出来。Socket 的使用比较简单,不会对整体系统增加太多代码量。这样开发出来的总控制器与分节点之间的通信,传输速率快、误码率低、容易实现,具有很强的实用价值。

**关键词:** 嵌入式 Linux; WiFi; Socket; 总控制器; 分节点

中图分类号: TP39

文献标识码: A

文章编号: 1674-7720(2014)02-0057-03

## Realization of communication between WiFi module based on Socket under Linux and Windows

Wang Yunliang<sup>1</sup>, Wang Guanglin<sup>2</sup>

(1.Tianjin Key Laboratory for Control Theory & Application in Complicated System, Tianjin University of Technology,  
Tianjin 300384, China;

2.School of Automation, Tianjin University of Technology, Tianjin 300384, China)

**Abstract:** To realize the data's wireless transmission between master controller under Linux development environment and partial node under Windows, using the method of Socket solves wireless communication between their wifi modules. The method can achieve the normal transmission of data and put the value displayed accurately. The using of socket is simple, which won't add too much system code. The result has a higher transmission rate, lower bit error rate and is easy to implement and has a strong practical value.

**Key words:** embedded Linux; WiFi; Socket; master control; partial node

现如今无线通信的方式越来越多,组网模式也千姿百态。一般研究同种操作系统下的 Socket 或 WiFi 通信的比较多,但在两种系统开发环境下 WiFi 之间的通信研究却并不多。众所周知, Linux 操作系统由于其系统源码的自由和开放性,以及系统性能的稳定性和被广泛用作服务器的操作系统。而 Windows 由于其友好的图形界面和强大的编程环境,常被广泛用作客户端的开发环境和操作系统<sup>[1]</sup>。因此,研究两大操作系统的实时通信问题具有一定的现实意义。

本文主要研究嵌入式 Linux 下的 ARM 与 Windows 环境开发下的 STC80C51 单片机之间的无线通信问题,通过套接字编程由 WiFi 进行传输,成功实现了两大开发环境下数据的无线传输,基本解决了两者之间的通信

问题。

### 1 Socket 原理

Socket 通常称作“套接字”,用于描述 IP 地址和端口,是一个通信链的句柄。在客户端/服务器通信模型中, Socket 是网络通信的基本操作单元,一个 Socket 描述就是通信连接的一端,在一个网络通信连接中,两端通信程序应各有一个 Socket 对其进行描述,不同的进程通过各自的 Socket 发送和接收消息,从而实现了网络通信<sup>[2-3]</sup>。

Linux Socket 支持以下 3 种常见的套接字类型:(1)流式 Socket(SOCK\_STREAM),一种最常用的套接字类型,可以保证数据传输的可靠性,不会出现数据丢失、破损或重复出现等差错;(2)数据报 Socket(SOCK\_DGRAM)是一种无连接和不可靠的双工数据传输服务,数据通过独

欢迎网上投稿 [www.pcachina.com](http://www.pcachina.com) 59

## 网络与通信 Network and Communication

立的报文进行传输,是无序的,并且不保证数据的可靠性和无重复性,适用于可靠性要求不高的一些场合;(3)原始 Socket(SOCK\_RAW),这种类型的 Socket 允许对底层协议(如 IP 或 ICMP)直接访问,它功能强大但使用不太方便,一般用于新协议的开发。

Socket 的编程方法很多,但主要有两种方式——面向连接的 TCP 协议方式和无连接的 UDP 协议方式。本文采用的是 TCP 协议方式。

### 2 通信平台的搭建

本文研究的是两个平台的通信,服务器是内有嵌入式 Linux 操作系统的 ARM 开发板,客户端是运行于 Windows 操作系统开发环境下的单片机。本文在实际研究中,服务器端选用 ARM11 系列的 S3C6410 为开发平台,内部嵌入 Linux2.6.28 内核版本,目标代码的编译平台选用装有 Ubuntu12.04 操作系统的普通 PC,它们通过交叉编译和 NFS 挂载的方式实现了服务器端编译平台和运行平台的正常运行;客户端是在 Windows XP 系统开发环境下的带有各种传感器的 51 单片机。

为了实现二者之间的通信,WiFi 模块在此过程中起到枢纽的作用,因此通信两端都有一个 WiFi 模块。在服务器端的 WiFi 模块选用的是 Marvell 88w8686<sup>[4]</sup>,它是一款低成本、低功耗、自带数据链路层的 WiFi 协议栈,内部集成了两个工作频率为 128 MHz 的 ARM7 Core,一个 Core 负责网络数据的处理,另一个 Core 负责处理 RF 方面的工作。该模块通过 SDIO 接口与 ARM 开发板相连。

客户端的 WiFi 模块选用海凌科电子有限公司新推出的 HLK-RM04 模块<sup>[5]</sup>,是基于通用串行接口的符合网络标准的低成本嵌入式模块。通过该模块,传统的串口设备在不需更改任何配置的情况下,即可通过 Internet 网络传输自己的数据。模块通过双 9 针工头串口线与单片机相连,功能结构图如图 1 所示,由于该模块内置了 TCP/IP 协议栈,能够实现用户串口、以太网、无线网(WiFi)3 个接口之间的转换。

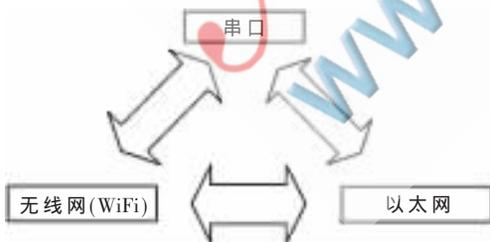


图 1 HLK-RM04 模块的功能结构图

至此,整个系统的通信平台的搭建已经完成,系统的整体结构功能框图如图 2 所示,从图中可以看出 WiFi 模块的组网模式。

### 3 通信过程的实现

本文以采集到的温度值的传输来说明实现的过程。下面分别从服务器端和客户端进行介绍。

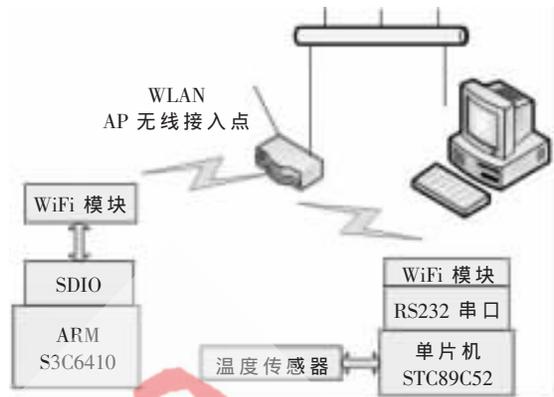


图 2 系统的整体功能结构框图

#### 3.1 服务器端的实现

虽然在举例中采用的是单一客户端与服务器进行通信,但考虑到实际中往往会遇到多个客户端连接服务器的情况,像常用的 `recv`、`send` 都是阻塞性函数,若资源没有准备好,则调用该函数的进程将进入睡眠状态,这样就无法处理 I/O 多路复用的情况。鉴于此,使用 `select` 函数既可以实现非阻塞 I/O 或信号驱动 I/O,还可以设置等待时间。服务器端实现的程序流程图如图 3 所示。

接下来对其中一些关键性的步骤进行阐述。

##### (1) 建立 Socket

```
socketfd = socket(AF_INET, SOCK_STREAM, 0);
```

`AF_INET` 表示的是针对 Internet 的通信协议,允许在

图 3 服务器端的程序流程图

```

graph TD
    Start([开始]) --> CreateSocket[创建 Socket]
    CreateSocket --> Bind[绑定套接字与端口]
    Bind --> Wait[等待客户连接]
    Wait --> HasClient{有客户?}
    HasClient -- N --> Wait
    HasClient -- Y --> Select[调用 select 函数]
    Select --> Receive[接收数据并显示]
    Receive --> End([结束])
  
```

##### (2) 绑定和监听

```
bind(socketfd, (struct sockaddr*)&server_sockaddr,
      sizeof(struct sockaddr)); // 绑定端口号和地址
listen(socketfd, n); // 监听, n 表示允许与服务器的最大连接数
```

它们都是成功时返回 0, 出错时返回 -1。

##### (3) 调用 `select` 函数与客户端连接

```
fd_set readfd, writefd; // 文件描述集的声明
FD_ZERO(&readfd); // 清空 readfd 与所有文件句柄的联系
FD_SET(socketfd, &readfd); // 建立文件句柄与 readfd 的联系
select(MAX_CONNECTED_NO, &readfd, NULL, NULL,
        (struct timeval*)0); // select 函数的调用
FD_ISSET(socketfd, &readfd); // 检查 readfd 联系的文件句柄 socketfd 是否可读写, 当大于 0 时表示可读写
```

## 网络与通信 Network and Communication

### 3.2 客户端的实现

客户端的实现实际上分两部分进行:一是单片机通过 DALLAS 公司的 DS18B20 温度传感器将采集的数据传送给串口;二是与之用串口线相连的 RM04 模块,通过 WiFi 与服务器建立连接,具体实现的程序流程如图 4 所示。

第一部分采集数据传给串口的过程需要注意数据在发送的过程中应该把数据以 ASCII 形式发送,这样服务器端才能以想要的十进制的形式显示出来,其他的比较简单,不再赘述。主要介绍第二部分中 RM04 模块的配置。

(1)调到相应的端口,搜索到该模块。

(2)以无线网卡的模式配置该模块,选择 TCP 客户端模式,远程端口的 IP 和端口号,无线局域网的名称与密码,本模块的 IP 等,具体配置图如图 5 所示。



图 5 RM04 模块配置图

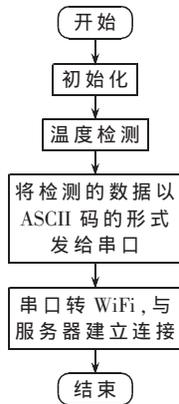


图 4 客户端的程序流程图

(3)在命令窗口下 ping 192.168.1.120,如果 ping 通,说明该模块成功。

在 Linux 系统中进入开发板,先启动服务器端,再运行客户端程序,服务器端出现如图 6 所示的结果。

```

[root@urbetter /mnt]# ./server3
socket success,sockfd=3
bind success!
listening...
received a connection :5
27.35
27.35
27.35
27.35
27.35
  
```

图 6 服务器端的运行结果图

通过图 6 可以看出,客户端与服务器端之间能够正常通信,并能正确显示出当前测量的温度是 27.35℃。该项研究主要针对总控制器与分节点之间的无线通信,对于需要这种模式通信的系统有很大的实际使用价值。

#### 参考文献

- [1] Li Fei, Yuan Lin, Wang Zhihuo, et al. Realization of communication between Linux and Windows based on Socket[C]. Proceedings of the 2010 International Conference on Information Technology and Scientific Management, 2010.
- [2] 王远洋,周渊平,郭焕丽. Linux 下基于 Socket 多线程并发通信的实现[J]. 微计算机信息, 2009, 25(5-3): 70-72.
- [3] 郭东升,田秀华. Linux 环境下基于 Socket 的网络通信[J]. 软件导刊, 2009, 8(1): 116-118.
- [4] 王云亮,李莹. 嵌入式安防远程监控系统的设计[J]. 化工自动化及仪表, 2013, 40(2): 237-240.
- [5] 深圳市海凌科电子有限公司. HLK-RM04 应用手册 1 串口转以太网应用[K]. 2012.

(收稿日期: 2013-09-25)

#### 作者简介:

王云亮,男,1963 年生,教授,主要研究方向:电力电子技术、自动控制和微机控制技术。

王光林,男,1987 年生,硕士研究生,主要研究方向:无线通信和嵌入式的开发。