

基于 ARM-Linux 的 Ad-hoc 组网^①

王晓华, 贾继洋

(西安工程大学 电子信息学院, 西安 710048)

摘要: 针对传统通信系统不能满足在无任何网络情况下的通信需求现状, 设计并实现了一种由 ARM-Linux 系统及其外围部件组成的无线自组网(Ad-hoc)的通信方案. 在 ARM11 平台上完成了 Linux2.6.36 操作系统和 RT3070 无线通信模块的驱动程序移植. 利用基于 TCP/IP 协议的 socket 编程, 编写测试程序, 进行节点间无线通信传输实验. 实验结果表明: 本文搭建的 Ad-hoc 网络, 可以用最少的资源和成本, 实现节点间可靠的无线高速率通信, 具有现实应用意义.

关键词: ARM-Linux 系统; Ad-hoc 网络; TCP/IP; socket; Wi-Fi

Ad-hoc Network Based on ARM-Linux

WANG Xiao-Hua, JIA Ji-Yang

(School of Electronic and Information, Xi'an Polytechnic University, Xi'an 710048, China)

Abstract: Aiming to solve the problem of the contemporary communication system in the absence of any network, a new scheme of the Ad-hoc network based on ARM-Linux system is designed. Firstly, this paper introduces the transportation of the Linux2.6.36 and the driver of RT3070 on ARM11 platform. Then based on socket programming in TCP/IP, program designing of communication and tests are conducted on the ARM-Linux platform. Results have proved that the Ad-hoc network can use the least resources and costs to achieve reliable high rate communication. It is significant for practical application.

Key words: ARM-Linux system; Ad-hoc network; TCP/IP; socket; Wi-Fi

随着网络技术和通信技术的发展, 使得 WLAN 成为了人们研究的热点之一, 其中, 最为明显的是 Wi-Fi 无线技术^[1]. 目前, 人们对无线局域网技术的应用仅仅局限在 PC 机^[2], 通过一个访问节点(Access Point, AP)、无线路由或无线网桥接入 Internet 实现不同设备通信; 但是, 在无任何网络情况下, 上述方式无法实现设备间通信. 无线局域网技术中的 Ad-hoc 网络是一种由移动节点组成的临时性的自组网络, 它不依赖于任何固定的网络基础设施, 主要靠节点间的相互协作来实现网络互联.

ARM 嵌入式微处理器凭借其低功耗高性价比的优势, 向各个传统微控制器的应用领域提出了挑战^[3]. 操作系统是嵌入式系统的灵魂, 用于管理嵌入式系统

的资源和控制应用程序的运行. Linux 是开放的源码, 具有遍布全球的支持者, 内核小, 效率高^[4], 网络功能强大^[5]等优势, 使得它在众多的操作系统如 WINCE、uC/OS、vxworks 等中脱颖而出, 现在已成为嵌入式操作的理想选择. 本文应用嵌入式 ARM-Linux 技术, 完成了 Ad-hoc 组网测试平台的设计.

1 Ad-hoc网络组网原理和模型

在 Ad-hoc 网络中, 每个主机都配有一个由嵌入式软件充当或一个路由器或是无线射频调制解调器的路由, 这样的主机称为一个“节点”. Ad-hoc 网络中的各个“节点”能够在没有中央控制单元作用时, 可以对等的通过无线连接方式进行通信^[6]. 网络中的两个主机

① 基金项目:陕西省教育厅科学研究计划项目(12JK0518),西安工程大学博士科研启动基金(BS1207);西安工程大学研究生创新基金(chx131121)

收稿时间:2013-11-25;收到修改稿时间:2013-12-30

在单点通信中可以通过无线网络直接通信,而在多点通信(multi-hop fashion)中,由于每个主机同时也是一个访问节点,这时候就需要设置一个公共访问节点,让公共访问节点充当“路由器”功能.图 1 为基于 TCP/IP 网络^[7]模型的 Ad-hoc 网络的一个典型多点通信模型.

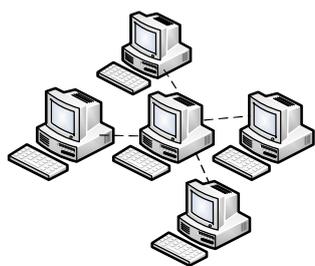


图 1 多节点 Ad-hoc 通信模型

本文采用了 USB WIFI 无线通信模块,在物理层(physical)和数据链路层(data link layer)使用 IEEE 802.11b/g/n 协议标准,在网络层采用 IP 协议,传输层采用 TCP 协议进行通信.

2 网络节点的构建

Ad-hoc 网络节点的构建,包括嵌入式硬件系统和软件系统的设计两个方面,其构建示意图如图 2 所示.图中,网络节点的硬件部分是由 OK6410A ARM 嵌入式硬件开发板和无线网络设备组成;而网络节点的软件部分是有嵌入式操作系统内核和应用程序两部分.

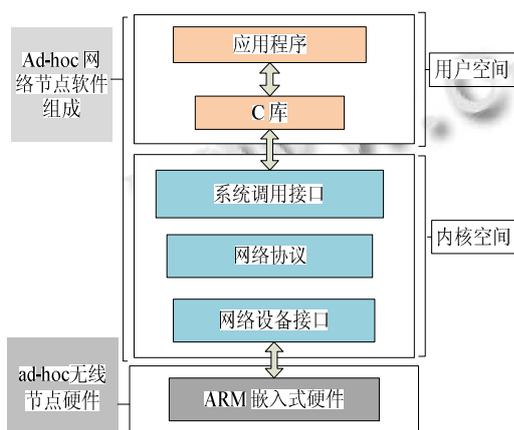


图 2 网络节点结构方框图

2.1 ARM 嵌入式平台的节点设计

基于 ARM11 的嵌入式硬件开发板和无线网络设

备硬件方框图如图 3 所示.其中,WIFI 模块、ARM 嵌入式开发板及外围部件共同组成了 Ad-hoc 网络的一个节点.个人笔记本电脑负责通过串口查看节点主机的信息收发情况,充当了超级终端的功能;WIFI 模块负责短程无线局域网传输^[8];ARM 嵌入式开发板及外围部件作为主控制器,负责嵌入式操作系统的文件管理和数据存储等功能.表格 1 是图 3 中主要元器件参数列表.

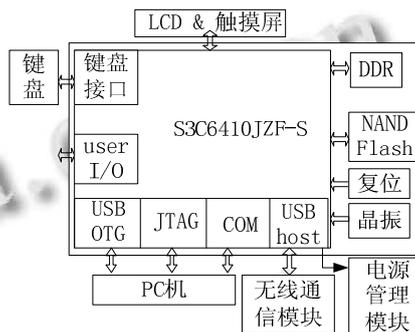


图 3 系统设计方框图

2.2 网络节点的软件设计

2.2.1 嵌入式 Linux 操作系统的移植

本文采用了目前比较稳定的一款 Linux 内核版本 ----Linux 2.6.36. 利用 Busybox 1.15.2 来制作 linux 2.6.36 所需要的 cramfs 根文件系统.在 PC 机的开发环境下,对 linux 2.6 内核进行配置,添加了与 S3C6410 相关的架构文件、平台文件,去掉了触摸屏模块,添加了无线网卡配置等.然后,利用交叉编译工具链 arm-linux-gcc 4.2.2 重新编译,并将生成的镜像 zImage 移植到 ARM 平台上^[9].实验表明裁剪后的 Linux 在硬件平台上运行良好.

linux 2.6.36 操作系统移植步骤如下:

① uboot 移植.在网上下载 uboot 1.1.6 源码,拷贝到 PC 机的开发环境下,然后执行 make smdk6410_config、make all,完成 U-Boot 的配置、编译、连接;最后将生成的烧写映像文件 u-boot.bin 烧写到 Nandflash 中.

② linux 2.6.36 移植.首先对内核进行配置,使其支持 S3C6410;其次,修改 MTD 分区,完成对 NAND Flash 分区;然后,配置文件系统,选择对 CRAMFS 文件系统的支持;最后执行 make clean 删除原来编译的结果,使用 make zImage 对内核进行编译,在 arch/arm/boot 目录下生成 zImage.

表 1 主要元器件参数列表

元件名	型号选择和功能描述
主处理器	S3C6410JZF-S 是一款 16/32 位的精简指令微处理器, 其工作频率可达 533/667MHz.
NAND Flash	NANDFlash 采用 2G k9GAG08U0D 芯片, 主要用于存储内核代码、应用程序和数据.
DDR SDRAM	采用两片 128M K4X1G163PC 扩充了内存, 组成了 256M Mobile DDR
LCD & 触摸屏	4.3 寸 LCD 触摸屏设备
COM	该端口遵循 RS232 通讯协议
USB Host	Port USB 1.1 Host 接收速率可达 12Mbps
电源	Port USB 2.0 OTG 支持高速设备
JTAG	通过 JTAG 接口, 用户可以在 SDRAM 中编程和调试应用程序.
无线通信模块	采用雷凌科技有限公司的一款 USB WIFI 模块, 主控芯片为 RT3070, 可以实现 CPU 和其他设备没有物理连接情况下的通信.

③ 编译、制作 Linux 系统根文件系统^[10]. 首先制作 CRAMFS 文件系统的根目录和各级子目录. 然后在命令行下切换到解压后的 Busybox1.15.2 源码目录下, 利用 make menuconfig 进入 Busybox1.15.2 的可视化配置菜单中, 修改相应配置, 并设定安装目录, 保存; 然后在命令行输入命令 make 进行编译, 用 make install 进行安装. 接着, 将刚才生成的文件拷到用户自己创建的文件系统对应的目录下, 并分别编译对应的文件. 最后, 利用 mkcramfs 工具制作文件系统镜像文件, 它能将所有文件压缩为一个小的镜像文件. 在终端输入命令 sudo ./mkcramfs /home/rootfs/ cramfs, 在 mkcramfs 工具所在的目录下生成了新的 cramfs 根文件系统.

④ SD 卡一键烧写系统. 设置开发板从 SD 卡启动, 利用 SD 卡将制作好引导程序 u-boot.bin、zImage 和 cramfs 拷贝到 SD 卡, 完成 linux2.6.36 操作系统的移植.

⑤ 设置开发板从 NAND Flash 启动, 重启 ARM 开发板, 系统可以正常运行.

2.2.2 USB WIFI 驱动移植

Linux2.6.36 操作系统提供了对 USB 接口的驱动, 为了实现无线通信模块的通信功能, 需要系统能够正常识别到 RT3070 USB WIFI 模块并能够正常被调用.

因此, 就要在 Linux 操作系统下完成驱动配置, 移植和动态加载. 基本流程如图 4 所示.

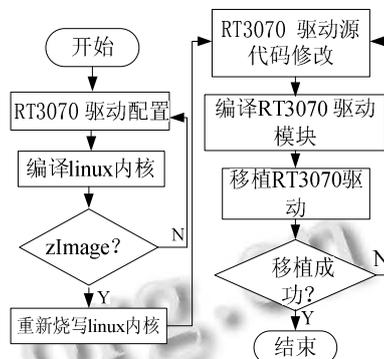


图 4 RT3070 驱动移植流程图

① Linux 内核预配置:

利用内核编译命令 make menuconfig, 在弹出的可视化菜单中, 要完成下边两个步骤: Networking support --->Wireless, 选中 “Wireless extensions sysfs files”; Networking support --->Device Drivers --->选中 Device Drivers support, 选中 Wireless LAN 并进入无线局域网子目录, 配置 Ralink driver support 为模块驱动, 保存内核配置.

② 重新编译内核, make 生成 zImage 映像文件.

③ 在网上下载最新的 RT3070 驱动源码版本, 然后将所有文件拷贝到 PC 机虚拟机和 Ubuntu10.10 构建的开发环境中的 /home/kernel/ 文件夹下. 在终端中切换到该目录, 解压驱动源文件, 并修改解压后的文件名为 rt3070drv, 并将该文件拷贝到用户自行创建的测试目录, 这里为 /home/jia_test/ 目录.

④ 进入 rt3070drv 目录, 修改 RT3070 驱动.

a) 编辑 Makefile, 修改芯片类型和开发平台、linux2.6.36 内核源码和交叉编译工具链的路径.
 ifeq (\$(CHIPSET),)
 CHIPSET = 3070
 endif

#PLATFORM:Target platform
 PLATFORM = SMDK

ifeq (\$(PLATFORM),SMDK)
 LINUX_SRC=/home/kernel/linux-2.6.36.2-v1.05

```
CROSS_COMPILE=/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-
endif
```

b)修改 os/linux/config.mk 文件,将下边几项设置如下:

```
# Support Wpa_Supplicant
HAS_WPA_SUPPLICANT=y
```

```
# Support Native WpaSupplicant for Network Manager
HAS_NATIVE_WPA_SUPPLICANT_SUPPORT=y
```

⑤ 编译 rt3070drv 源码,在 rt3070drv 驱动文件夹下生成了 RT2870STA.dat,并在/os/linux/下生成了 rt3070sta.ko,并编写脚本文件 rt3070drv.sh.

脚本文件内容如下:

```
#!/bin/bash
mkdir -p /etc/Wireless/RT2870STA
cp RT2870STA.dat /etc/Wireless/RT2870STA
insmod rt3070sta.ko #加载 rt3070 驱动模块
ifconfig -a #查看所有的网卡设备
iwlist ra0 scan #扫描无线网络 ra0
```

⑥ 执行 shell 脚本文件,加载无线网卡驱动.将 RT2870STA.dat 和 rt3070sta.ko 拷进 SD 卡,重启开发板.运行脚本文件,完成 USB WIFI 驱动加载,并查看无线网络特性.脚本运行结束,最后出现

```
===>rt_ioctl_giwscan. 2(2) BSS returned,
data->length = 380
```

```
ra0 Scan completed :
```

```
Cell 01 - Address: 5C:63:BF:3F:B6:2C
```

```
Protocol:802.11b/g/n
```

```
ESSID:"jcjshlab"
```

```
Mode:Managed
```

```
Frequency:2.412 GHz (Channel 1)
```

```
Quality=13/100 Signal level=-85 dBm Noise
level=-90
```

说明驱动加载成功.

2.2.3 用户空间应用软件设计

基于 TCP/IP 协议的网络编程方式, WIFI 无线终端程序主要由两部分组成: 客户机端的程序和服务器端的程序. 采用了典型的客户机---服务器模型, 使用服务器来管理客户机的外围设备采集的数据. 本文采用网络套接字(socket)编程^[11], 构建了并发服务器, 使得

服务器可以同时接受 5 个客户端的请求. 实现由客户端输入字符, 服务器端负责接收字符, 并将接收到的字符转换为大写字符后重新发送给客户端, 客户端再次接收转换后的字符.

程序流程图如图 5 所示: 服务器端首先创建套接字, 将其初始化为数据流套接字(SOCKET_STREAM), 绑定端口号, 并开始监听(设置可以接入的客户机的个数), 当客户机端发出连接请求后, 建立连接, 并读取接收到的信息; 同时, 服务器也可以向客户机发送停止数据采集的命令. 客户机端, 先创建套接字, 设置服务器的 IP 地址和端口号, 然后调用建立连接函数 connect(), 将外部输入的数据发送到服务器端.

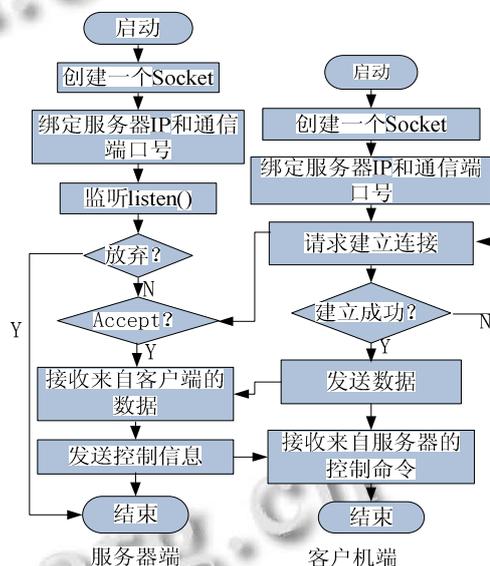


图 5 通信测试程序流程图

3 实验测试

本文在 ARM-Linux 平台上分别进行了两个节点的自组 Ad-hoc 网络测试和节点通信测试^[12], 实验结果很好的验证了自组 Ad-hoc 网络的稳定性和可靠性.

3.1 Ad-hoc 网络测试

分别用两个 Linux2.6.36 内核的操作系统 OK6410A 的开发板和两个 Ralink 科技有限公司研制的 USB WIFI 无线模块和一台 PC 机. 其中 PC 机和 ARM 开发板通过 RS232 串口连接.

两个节点的 IP 配置分别为:A: 192.168.0.100 B:192.168.0.200. 由于这两个节点并没有物理上的任何连接, 这时候 A 节点是没法访问 B 节点的. 为了能够让这两个节点进行信息交互, 需要在每个无线节点

中设置一个相同的 SSID/ESSID(Service Set Identifier),用以区分不同的网络.这里设置通信网络的名称为 WIFI,两个无线节点的工作模式都设置为 ad-hoc 模式.

USB WIFI 驱动移植成功后,使用 shell 脚本命令按上述参数分别配置两块开发板,然后使用 ping 命令测试数据包接收和路由情况(这里在服务器端):ping 192.168.0.200

数据包接收情况如下:

```
PING 192.168.0.200 (192.168.0.200): 56 data bytes
RTMP_TimerListAdd: add timer obj d08c07f4!
64 bytes from 192.168.0.200: seq=0 ttl=64 time=17.184
ms
Rcv Wcid(1) AddBAReq
Start Seq = 00000002
RTMP_TimerListAdd: add timer obj d08c3404!
64 bytes from 192.168.0.200: seq=1 ttl=64 time=2.939
ms
64 bytes from 192.168.0.200: seq=2 ttl=64 time=2.970
ms
64 bytes from 192.168.0.200: seq=3 ttl=64 time=3.005
ms
64 bytes from 192.168.0.200: seq=4 ttl=64 time=4.015
ms
64 bytes from 192.168.0.200: seq=5 ttl=64 time=5.126
ms
```

本实验说明两个节点通信建立成功.

3.2 节点通信测试

在一块 ARM11 开发板上运行服务器程序,另一上边运行客户机程序.其中,用 A 作为服务器端,需要给出端口号;B 作为客户机,需要给出访问的服务器 IP 和通信进程的端口号.这里两个端口号要相同,不妨设置为 6000.

先运行服务器端程序,再运行客户机端的程序,在客户机端输入并查看通信情况:

```
[root@FORLINUX6410]# ./client 192.168.0.100
6000
input message:abc
Message from server:ABC
input message:i love you,honey!
```

Message from server:I LOVE YOU,HONEY!

实验表明 ad-hoc 网络中的节点能够实现可靠通信.

4 结语

本文通过对 ARM11 硬件系统的设计, Linux2.6 内核操作系统和 USB WIFI 驱动的移植,完成了基于 ARM-Linux 开发平台的自组 Ad-hoc 网络的设计.文中,设计的两个节点的客户机服务器软件程序,具备很强的可移植性,可以同时实现 ad-hoc 网络的多点通信,对解决工业现场工作设备的通信问题,具有很好的参考价值.

参考文献

- 1 陈新,翁秋华.基于 Linux+ARM9 的 Wi-Fi 网络图形化设计与实现.通信技术,2012,45(3):60-62,65.
- 2 黄学雷,陈祖爵.嵌入式无线局域网设备的设计与实现.微计算机信息,2006,22(5-2):23-25.
- 3 孙恩元.基于 ARM 的机车监控显示系统研究与设计[硕士学位论文].长沙:国防科学技术大学,2008.
- 4 李新洲,赵力宏.基于 ARM 和嵌入式 Linux 的手持数据分析仪的设计.工业控制计算机,2011,24(1):88-89.
- 5 姜云杰.Linux 在嵌入式系统中的应用优势及前景.中国科技信息,2008,11:110-111.
- 6 Wu J. Ivan Stojmenovic. Ad Hoc Networks. IEEE Computer Socitey.2004:29-31.
- 7 宋敬彬,孙海滨,等.Linux 网络编程.北京:清华大学出版社,2010.
- 8 邹意然,王永威,赵龙.基于 ARM 体系的 WIFI 移植研究.微计算机信息,2010,26(11-2):67-69.
- 9 胡庆烈.Linux 内核的配置与编译.电脑知识与技术,2009,5(3):730-731,735.
- 10 刘超.基于 3G 的嵌入式视频监控系统的设计与实现[硕士学位论文].大连:大连理工大学,2012.
- 11 童永清.Linux C 编程实战.北京:人民邮电出版社,2008.
- 12 王浩,魏明磊,李云,陈前斌,隆克平.Linux 环境下 Ad hoc 网络的实现及性能分析.重庆邮电学院学报(自然科学版),2005,17(5):608-612.