

# 基于 i.MX21 的 AdHoc 网络终端的设计与实现

侯战胜,姚放吾

(南京邮电大学 计算机学院,江苏 南京 210003)

**摘要:**随着嵌入式技术和无线 AdHoc 网络技术的发展,无线 AdHoc 网络应用成为研究的热点。文中设计了基于应用处理器 MC9823MX21 的无线 AdHoc 网络终端系统,论述了各个模块的设计思想,给出了设计结构原理框图。采用基于 Memory 接口的方式,首次将 WT4 无线网卡成功移植到 i.MX21 平台上。在嵌入式 Linux 系统上实现对无线通信模块的支持,成功构建了具有移动自组网和多跳传输的功能的无线移动 AdHoc 网络,并且实现了各个无线网络终端之间的数据传输通信。

**关键词:**AdHoc 网络;嵌入式系统;中断服务程序

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2010)01-0005-04

## Design and Realization of AdHoc Network Terminal Based on i.MX21

HOU Zhan-sheng, YAO Fang-wu

(College of Computer, Nanjing University of Posts & Telecommunication, Nanjing 210003, China)

**Abstract:** The application of wireless AdHoc network become research point with the development of embedded technology and wireless AdHoc network technology. Design a wireless AdHoc network terminal system based on the application processor MC9823MX21, discuss the design idea of each module, and draw the block diagram of design structure. With the memory interface, it first successfully transplants wireless card WT4 to i.MX21 platform and adds the wireless communication module into the embedded Linux system. In this way, it constructs a wireless mobile AdHoc network possessing MANET and multi-hop transmission functions, making the data transmission communication among the wireless network terminals available.

**Key words:** AdHoc network; embedded systems; interrupt service routine

### 0 引言

近年来嵌入式微处理器性能的提高,基于嵌入式系统的无线通信模块不仅能实现数据传输而且还能够满足人们对于多媒体通信的需求。嵌入式系统整合 WLAN 技术实现无线通信已成为应用的热点。

无线网络<sup>[1]</sup>通常可以分为有中心网络和无中心网络,前者需要固定基础设施的支持,移动主机之间通信通常借助基站来完成,例如蜂窝移动通信系统;后者主要是指移动 Adhoc 网络,它不需要固定的基础设施,能够快速自动组网,与有中心网络相比,Adhoc 网络每个移动节点兼备路由器和主机两种功能<sup>[2]</sup>,Adhoc 网

络灵活、健壮、投资少,特别适合于作战指挥、抢险救灾以及应付突发事件和执行临时任务的场合。而文中设计的网络终端是属于后者。

### 1 平台的硬件结构设计

根据整个项目对终端的功能要求,该硬件平台主要由 4 个模块组成:CPU 模块、多媒体模块、终端接口模块以及无线模块。其中核心模块电路主要包括处理器和 Flash 存储器、SDRAM 存储器的连接电路以及启动复位等电路;多媒体模块电路主要包括 LCD 显示电路、摄像头输入电路和音频输出电路等;终端接口模块包括 USB 接口、SD 卡存储接口、串口和以太网口等;无线模块主要 WLAN 芯片和 RF 芯片等,整个终端的硬件结构图如图 1 所示。

#### 1.1 CPU 模块

CPU 模块部分由 freescale 公司的 MC9328MX21 处理器<sup>[3]</sup>,一片 NOR flash 存储器(16M 字节),两片

收稿日期:2009-04-22;修回日期:2009-07-05

基金项目:国家 863 计划资助项目(2006AA01Z208)

作者简介:侯战胜(1983-),男,安徽宿州人,硕士研究生,研究方向为嵌入式技术及其在通信中的应用;姚放吾,教授,硕士生导师,研究方向为嵌入式技术及其在通信中的应用,并行计算机及其体系结构。

SDRAM 存储器(共 32M 字节),复位电路,BOOT 电路以及 JTAG 调试电路等部分组成。

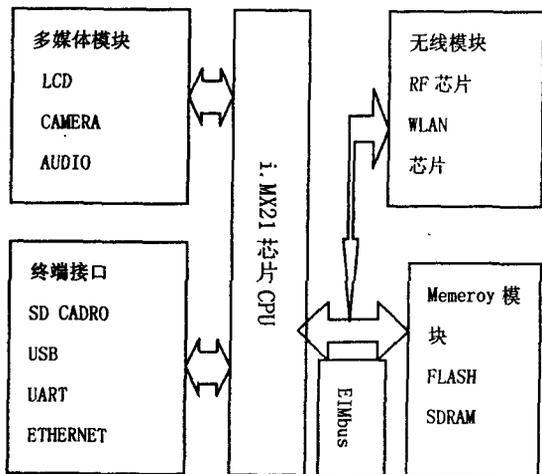


图 1 终端硬件结构图

### 1.2 无线网卡部分

因为 wifi 芯片开发难度较大,所以一般采用无线模块的方式开发。设计采用的无线模块为六合万通的基于 WT4 的 memory 接口的无线模块<sup>[4]</sup>。该模块接口地址线 16 位,数据线 16 位,采用处理器的 cs3 作为片选,接口插座采用 minipci 插座,模块电源为 3.3V。

## 2 无线网卡的硬件设计

因为要构建无线自组织网络,所以无线网卡的硬件设计是该终端设计的重点部分。在标准方面,设计兼顾了目前流行的 IEEE802.11 和 WAPI 两种标准,力争为设计提供更为安全可靠的加解密措施<sup>[5]</sup>。在接口方面采用了与 CPU 总线直接相连接的 memory 接口方式。该无线网卡模块采用 MAC 和基带处理芯片加上射频芯片构建而成,完整实现了 IEEE 标准所规定的 802.11a/b/g 基带信号处理、协议处理以及各种附加增强功能模块<sup>[6]</sup>。完全支持 802.11b 和 802.11g 所要求的各种调制方式,能够完整提供协议规定的 1Mbps 到 54Mbps 的全部传输速率<sup>[7]</sup>。该模块能够完成媒体接入层(MAC)所规定的所有协议,并针对语音通信和 QoS 的要求。

### 2.1 网卡模块设计

无线模块的设计主要可以分为四个部分:MAC, BaseBand, ADC/DAC 以及 RF 芯片<sup>[4]</sup>。该设计采用一个 WLAN 芯片加一个 RF 芯片结构,完整实现了 IEEE 标准所规定的 802.11a/b/g 基带信号处理、协议处理以及各种附加增强功能,集成了 MAC, BaseBand, ADC/DAC 功能。

### 2.2 网卡模块的工作模式

该模块的工作模式<sup>[4]</sup>可以分为 4 种:

(1) 芯片复位时 Chip-Power-Save 寄存器 3'b000 处于 power-up 模式,此时 MAC 和 BB 的时钟都打开, ADC/DAC 在相应的收发通路使能的时候工作,RF 由寄存器 RF-Interface-Controller-Register 控制是否工作。

(2) 软件在芯片复位后立即设置寄存器。

Chip-Power-Save 为 3'b100,使芯片进入 normal-work 模式,此时门控时钟有效,此为推荐的操作模式。

(3) power-save 模式时,BaseBand 和 DAC/ADC 的时钟都被停掉了,此时不能收发,但软件可以正常的访问和配置寄存器,读写 memory。

(4) sleep-mode 模式时,MAC,BaseBand 和主机接口的时钟都停止,仅唤醒模块工作,主机可执行唤醒操作。从 sleep-mode 唤醒,芯片即进入 power-save 模式。各个状态跳转及条件如图 2 所示。

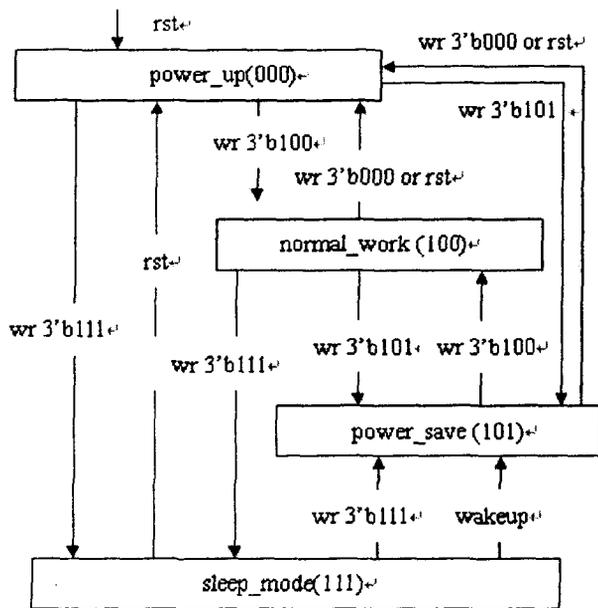


图 2 网卡模式状态跳转图

其中 rst 表示复位 Wr3'bxxx 向 Chip-Power-Save 写入相应的数值,用二进制表示

## 3 无线网卡驱动程序设计

Wt4 驱动源码中主要包括类型定义,物理寄存器配置相关函数,链路操作相关函数,发送/接收帧管理,发送接收包相关函数,设备相关结构及函数,I/O 控制结构及相关函数,mac 层相关结构以及函数,物理层寄存器管理相关结构和函数,初始化相关结构和函数等等<sup>[8]</sup>。

其中模块加载时,执行的第一个函数是 wt\_hotplug\_mem.c 中的 WtModuleInit(void) 函数。

```
static int _init WtModuleInit(void)
{ int err;
  struct net_device * ndev = alloc_etherdev(sizeof(WT_ADAPTER));
  if (! ndev)
    return ndev;
  FN_ENTER;
  printk("Loaded %s driver, version %s\n", DRV_NAME, DRV_VERSION);
  _bug_on_wrong_struct_sizes();
  err = WtProbe(ndev, *wtwlan_mem_addr);
  if (err)
    goto out;
  dev_wlanwt4 = ndev;
  FN_EXIT(0,0);
  return 0;
out:
  return err;
}
```

其中, WtProbe(ndev, \*wtwlan\_mem\_addr) 函数是寻找网卡设备以及进行相关初始化的函数, 它的第二个参数为系统分配给无线网卡的内存区域的首地址。这个值由 static unsigned int wtwlan\_mem\_addr[] = {IMX21\_WTWLAN\_BASE, 0} 中的宏 IMX21\_WTWLAN\_BASE 决定。

在 WtProbe 函数执行的工作主要有:

(1) 申请设备的 I/O 区域, 通过 request\_mem\_region(mem\_addr, 64 \* 1024, DRV\_NAME) 函数完成, 其参数 mem\_addr 是申请 I/O 区域的首地址, 由 WtProbe 通过参数传递获得。这里分配给无线网卡的片选是 3, 查看 mx21 芯片手册对应的物理地址是 0xd1000000, 但是, 操作系统内核运行后, 该物理地址被映射成虚拟地址 0xE4500000。驱动程序申请内存时必须提供一个起始地址, 而这个地址正是映射而成的虚拟地址。故需将宏 IMX21\_WTWLAN\_BASE 改成 0xE4500000。

(2) 建立网络设备接口以及它的结构体, 并对其结构体赋初值, 这其中包括填充设备结构体成员以及函数指针值, 初始化工作队列, 初始化所有用到的锁机制等等。在配置网络设备初始参数过程中, 采用芯片通过 spi 总线读取片上 eeprom 中默认设置值的方式获取各种发送接收 buffer、软件参数、管理参数、mac 地址、发送以及接收错误极限等等。

(3) 申请中断号, 挂载中断服务程序, 通过 request-

irq(IMX21\_WTWLAN\_ETH\_IRQ, &WtInterrupt, SA\_SHIRQ, ndev->name, Adapter) 函数完成。这是一个回调函数, 中断发生时, 系统调用这个函数, 传入的参数包括硬件中断号 IMX21\_WTWLAN\_ETH\_IRQ, 设备标识符 Adapter, 中断服务程序函数指针 WtInterrupt, SA\_SHIRQ 表明多个设备共享中断。本系统中分配给无线网卡的中断引脚是 GPIO, 而系统中分配给 GPIO 引脚的中断号是 8, 这种多个中断复用中断号的方式在嵌入式 Linux 系统中非常普遍。

## 4 无线网卡驱动的移植

### 4.1 修改无线网卡驱动源码

对无线网卡驱动源码修改<sup>[9]</sup>主要包括三个地方:

1) 配置处理器的片选 3 的控制寄存器, 使其按照 16bit 读写操作, 并给无线网卡中断赋予中断号。具体在 wt\_hotplug\_mem.c 文件中的 wtModuleInit 函数, 做出如下添加:

```
_reg_WEIM_CSL(3) = 0X11118501;
_reg_WEIM_CSU(3) = 0X00002000;
_reg_SYS_FMCR &= 0Xffffffd;
_reg_GPIO_GIUS(3) |= 1 << 20;
_reg_GPIO_DDIR(3) &= ~(1 << 20);
_reg_GPIO_PUEN(3) |= 1 << 20;
_reg_GPIO_ICONFA2(3) &= 0XFFFFFFF;
_reg_GPIO_ICONFA2(3) |= 1 << 8;
_reg_GPIO_ICR2(3) |= 0X00000300;
_reg_GPIO_IMR(3) &= ~(1 << 20);
_reg_GPIO_ISR(3) |= 1 << 20;
_reg_AITC_INTENNUM = 8;
_reg_AITC_INTENABLEL |= 1 << 8;
```

2) 对 wtprecomp.h 中 IMX21\_WTWLAN\_BASE 和 IMX21\_WTWLAN\_ETH\_IRQ 的宏定义进行修改。由于 mx21 系统中片选 3 对应的虚拟地址为 0xe4500000, 故将 IMX21\_WTWLAN\_BASE 改为 0xe4500000, 由于无线网卡中断线引到 mx21 上 gpio 口, 而 gpio 口中断采用复用的方式, 都统一复用到中断 8, 故将 IMX21\_WTWLAN\_ETH\_IRQ 改为 8。

3) 对中断服务程序中添加判断是否为无线网卡中断的语句, 将同样为中断 8 但非无线网卡中断的中断响应直接从中断服务程序返回, 具体修改如下: 在 wt\_dev\_mem.c 文件中的 WtInterrupt(int irq, void \* config, struct pt\_regs \* regs) 函数起始处添加:

```
if(! (_reg_GPIO_ISR(3) & 0x00100000))
  return;
```

```
_reg_GPIO_ISR(3) |= 0x00100000;
```

#### 4.2 修改 u-boot, 配置内核, 编译内核

修改 u-boot 中对无线网卡片选控制寄存器的参数设置, 配置 Linux 内核使内核支持无线选项<sup>[10]</sup>, 交叉编译内核, 并将编译好的内核文件下载并烧到终端中的 flash 中去。

### 5 网络通信实验结果及分析

使用的 WT4 无线网卡支持 802.11a/b/g 模式, 该项目要求用 15 块板子进行通信试验, 在 15 个无线 AdHoc 网络终端上进行网络通信的实验:

使用无线测试软件检测了网络信号质量, 其中 AdHoc 网络“912”是笔者自己建立的, 其他的网络是南京邮电大学校园无线网络, 如图 3 所示。

同时编写了服务器端和客户端的数据传输程序 MyTcpServer 和 MyTcpClient, 在 IBSS 网络组建成功的情况下, 任选一个终端作为服务器端, 其他终端任选择一个作为客户端, 进行通信实验, 来验证网络的传输可靠性和传输时间:

服务器端 # ./MyTcpServer

客户端 # ./MyTcpClient 192.168.2.11 <服务器 IP>

```
/home/myapp/mp3player
Creating Socket success!
Binding success!
mp3player transferring begin.
mp3player transferring finished!
5 seconds used!
```

传输的 mp3player 大小为 10M 左右, 所用的时间为 5s, 在默认 802.11b 模式下, 基本达到了 11Mb/s, 取得了良好的效果。

### 6 结束语

将 WT4 无线网卡成功移植到 i.MX21 平台上, 在嵌入式 Linux 系统上实现对无线 AdHoc 网卡通信模块的支持, 构建一个无线 AdHoc 网络终端, 最终实现无线 AdHoc 网络的组网和数据传输通信。下一步的目标是在此平台上开发基于移动 AdHoc 网络的视频电话系统, 这将进一步提高该移动智能终端的应用范围和市场前景。

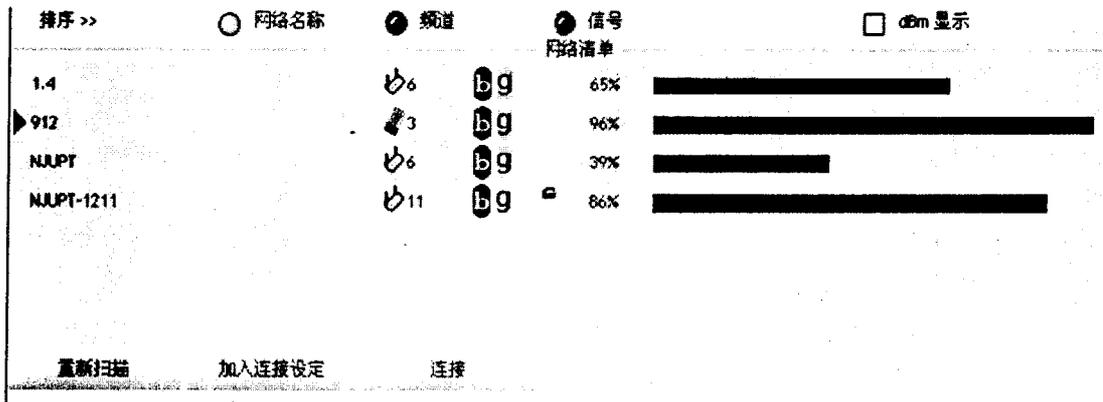


图 3 Ad-Hoc 网络图

#### 参考文献:

[1] Wu J, Stojmenovic I. Ad Hoc Network[J]. Computer, 2004, 37(2):29-31.

[2] Chlamtae I, Conti M, Liu J J N. Mobile Ad Hoc networking: imperatives and challenges[M]. [s. l.]: Ad Hoc Networks, Elsevier Press, 2003:13-64.

[3] Freescale Semiconductor China. MC9328data sheet[M]. 苏州: 飞思卡尔半导体苏州公司, 2006:7-15.

[4] 六合万通. WTIV DATA Sheet, WT4 芯片无线网络技术手册[M]. 北京: 北京六合万通公司, 2006:12-45.

[5] 朱坤华, 王玉芬, 李新丽. 两种无线局域网安全标准 WPI 与 IEEE802.11i 比较[J]. 河南科技学院学报: 自然科学版, 2005, 33(1):83-85.

[6] 孙宏, 杨义先. 无线局域网协议 802.11 安全性分析[J]. 电子学报, 2003, 31(7):1098-1100.

[7] Stubblefield A, Ioannidis J, Rubin A D. A Key Recovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP)[J]. ACM Transactions on Information and System Security, 2004, 7(2):319-332.

[8] Corbet J, Rubini A. Linux 设备驱动程序[M]. 第 3 版. 魏永明, 耿岳, 译. 北京: 中国电力出版社, 2005.

[9] 强新建. 基于 S3C2410x 的 Linux 下 UART 驱动程序实现[J]. 计算机技术与发展, 2006, 16(10):58-60.

[10] 黄信兵. Linux 在嵌入式系统中的应用与设计[J]. 计算机技术与发展, 2006, 16(10):146-148.