

Linux 下 Wi-Fi 驱动程序的设计与实现

Design and Implementation of Wi-Fi Driver Under Linux

刘芳华 周凤星

(武汉科技大学冶金自动化与检测技术教育部工程中心,湖北 武汉 430081)

摘要: 针对目前流行的 Wi-Fi 无线通信模块在嵌入式系统开发中的应用要求,设计开发了 Linux 下基于 ARM9 处理器 S3C2440 的 Marvell 88W8686 Wi-Fi 驱动程序。结合 S3C2440 与 88W8686 的硬件连接以及 Wi-Fi 驱动程序体系结构,给出了网络设备初始化和数据传递、转发的实现过程。实验结果表明,该 Wi-Fi 驱动程序能够实现预期目标,并已成功应用于系统开发。

关键词: Wi-Fi 无线通信 Linux ARM 驱动程序 网络设备

中图分类号: TP316.2 **文献标志码:** A

Abstract: Aiming at the applicable requirements of embedded system development using popular Wi-Fi wireless communication module, the Marvell 88W8686 Wi-Fi driver based on ARM9 processor S3C2440 under Linux has been designed and developed. Combining with the hardware connection between S3C2440 and 88W8686, and the architecture of Wi-Fi driver, the implementation processes of the initialization of network device, data transmission and forwarding are given. The result of experiments indicates that this Wi-Fi driver reaches desired target and can be successfully applied in system development.

Keywords: Wi-Fi Wireless communication Linux ARM Driver Network device

0 引言

近年来,Wi-Fi 无线通信技术发展迅速。相比有线接入技术,Wi-Fi 网络连接更具灵活性;而与其他无线技术相比,Wi-Fi 则具有传输数据快、距离远等优势。目前,很多公共场所都提供免费 Wi-Fi 服务,只要随身携带支持 Wi-Fi 功能的电子产品,即可方便地接入因特网^[1]。

由于 ARM 嵌入式技术已广泛应用于各种便携式电子设备的开发,因此,在 ARM 设备上增加 Wi-Fi 无线通信功能已成为热门话题。常见的连接 ARM 与 Wi-Fi 模块的接口有 SPI、SDIO 和 USB 等,其中 SPI 具有硬件连接方便、软件设计简单及节省系统资源等优点^[2]。目前,对 Linux 下 SPI 驱动的研究已较为广泛且深入,但还没有针对 Wi-Fi 驱动的专门研究。本文分析了 Wi-Fi 模块通过 SPI 接口与 ARM 连接时 Wi-Fi 驱动程序的设计与实现,并在此基础上总结出 Linux 下 Marvell 系列 Wi-Fi 模块驱动程序开发的一般方法。

1 Wi-Fi 技术及模块介绍

1.1 Wi-Fi 技术及 88W8686 芯片介绍

Wi-Fi 技术是 IEEE 定义的无线局域网通信工业标准 IEEE 802.11。该标准包括 IEEE 802.11a、802.11b 和 802.11g,其中 802.11b 是使用历史最长的 Wi-Fi 技术,其工作在 2.4 GHz 频段,可提供 11 Mbit/s 的无线传输速率^[3]。

88W8686 是 Marvell 推出的一款面向移动电话、PDA 及数字摄像机等移动设备的高整合 Wi-Fi 芯片。其在单一的芯片上集成了可以工作于 2.4 GHz 和 5 GHz 的双频射频无线收发器、物质层、媒介接入控制器和一个 ARM 处理器,实现了无线局域网通信、电源管理和加密等功能,并支持视频、语音和多媒体应用。

88W8686 向用户提供了 SDIO 和 SPI 数据传输接口。该模块将从 SDIO 或 SPI 接口传过来的用户数据封装成数据帧,通过 WLAN 传送给远程的客户端。本设计采用的是 SPI 接口。

1.2 Wi-Fi 与 ARM 的硬件连接

S3C2440 通过 SPI₀ 接口与 Marvell 88W8686 的 SPI 接口连接,实现数据的收发,如图 1 所示。其中,S3C2440 是数据的发送源,所以将其配置为主设备,88W8686 配置为从设备。SPI 采用全双工通信模式,主设备的 MISO、MOSI 和 CLK 引脚分别与从设备的

湖北省教育厅重点研究基金资助项目(编号:D200711004)。

修改稿收到日期:2010-10-19。

第一作者刘芳华,女,1984年生,现为武汉科技大学电路与系统专业在读硕士研究生;主要从事嵌入式系统与应用方面的研究。

SDO、SDI 和 CLK 引脚相连接;88W8686 的 SPI_SINTn 是低电平中断输出引脚,它与 S3C2440 的 EINT₁ 引脚相连,用于在通信过程中检测 Wi-Fi 的状态;S3C2440 的 nSS₀ 是片选引脚,低电平有效,用于激活从设备。

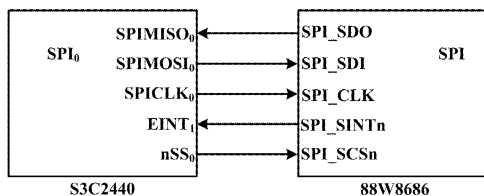


图 1 SPI 接口连接图

Fig. 1 Connection of SPI interface

2 Wi-Fi 驱动程序实现

2.1 Wi-Fi 驱动体系结构

Marvell 88W8686 Wi-Fi 模块正常工作所需的 Host 驱动包括 WLAN 和 SPI 接口驱动两部分。WLAN 驱动在整个数据收发过程中充当数据中转的角色,即接收上层用户应用程序的数据流,通过 SPI 口转发到 Wi-Fi 硬件或响应 Wi-Fi 硬件中断,并从硬件的缓冲区读取数据流,通过驱动程序注册的接口函数,发送到上层应用程序。Wi-Fi 设备在 Linux 下的软件结构层次如图 2 所示。

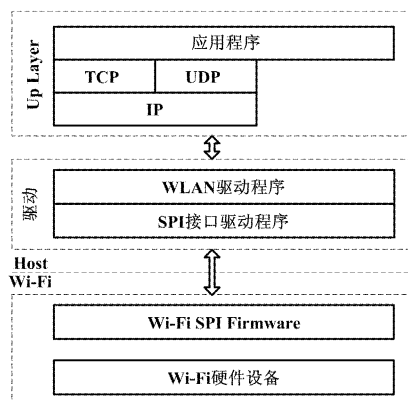


图 2 Wi-Fi 软件结构层次图

Fig. 2 Hierarchical structure of the Wi-Fi software

Firmware 是在 Wi-Fi 设备硬件中执行的一段程序,系统上电后由 WLAN 驱动将其下载到 Wi-Fi 模块中^[4],实现 Wi-Fi 硬件接口控制、数据缓冲、802.11 与 802.3 帧类型转换、802.11 MAC 层管理、WLAN MAC 中断管理以及硬件控制等功能。发送数据时,Host 驱动程序将从上层接收到的标准 802.3 帧发送给 Firmware, Firmware 将收到的数据帧转换成 802.11 帧,再通过无线连接将数据传输出去;接收数据时, Firmware 将

接收到的所有 802.11 帧转换成 802.3 帧后,通过 SPI 口发送给 Host 驱动。由此可见,Wi-Fi 无线网卡设备在 Linux 中是被当作普通的以太网设备对待的,在 Wi-Fi 驱动程序中无需实现 802.11 帧与 802.3 帧之间的类型转换。

2.2 WLAN 驱动程序

2.2.1 WLAN 设备初始化

WLAN 模块初始化函数 wlan_init_module() 的主要工作是注册回调函数^[5],将完成设备添加和删除功能的回调函数分别注册为 wlan_add_card() 和 wlan_remove_card()。wlan_add_card() 实现了 WLAN 设备初始化的所有操作,并依次完成以下几部分工作。

① 通过 sbi_probe_card() 检测设备,探测到无线网卡;然后,和一般 Linux 下网络设备驱动程序一样,调用 alloc_etherdev() 来分配代表网络设备的 struct net_device 结构,并在该结构中初始化内核对网卡操作的接口函数,主要接口初始化如下:

```
dev->open = wlan_open;
dev->hard_start_xmit = wlan_hard_start_xmit;
dev->stop = wlan_close;
dev->do_ioctl = wlan_do_ioctl;
dev->set_mac_address = wlan_set_mac_address。
```

② 调用 wlan_create_thread() 创建主线程 wlan_service_main_thread。该线程处理 WLAN 驱动的主要工作,包括处理 Firmware 产生的事件、接收从 Firmware 发送过来的数据,并发送从内核传递过来的数据。

③ 通过 wlan_create_thread() 创建 wlan_reassociation_thread 线程。该线程负责在连接自动断开时重新连接 AP。

④ 通过 sbi_register_dev() 注册 WLAN 设备,填写代表网卡设备的私有结构体中的网卡硬件设备相关信息和 IRQ 请求。

⑤ 调用 wlan_init_fw() 初始化 Firmware 并下载 Firmware 到 Wi-Fi。

⑥ 调用 register_netdev() 注册网络设备,供上层访问。该函数返回设备的主设备号,之后对网络设备的所有调用都通过这个设备号来实现。

2.2.2 WLAN 数据包发送

WLAN 驱动程序的数据发送函数是 wlan_hard_start_xmit()。该函数已经在 wlan_add_card() 中设置。此外,在 WLAN 驱动程序中还引入了 WMM (Wi-Fi multi-media) 机制来规范数据流量优先权,从而实现无线网络流量的优先级管理。与一般的网络设备驱动程序一样,sk_buff 数据结构用来存放从上层应用程序接

收到的数据。发送函数调用 `wlan_tx_packet()` 把接收到的 `sk_buff` 结构数据添加到 WMM 队列, 然后唤醒主线程来处理数据的发送。

主线程首先为数据传输做了一些准备工作, 包括初始化 WMM 状态信息和数据队列、检查当前设备是否可用以及唤醒网络设备等, 然后依照当前设备的可用性作不同的处理。如果当前设备不可用(即设备忙), 则等待, 直到设备可用时再发送数据; 如果当前设备可用, 则通过 `wmm_process_tx()` 传输 WMM 等待队列中优先级最高的数据包。需要注意的是, `wmm_process_tx()` 要先进行模式判断, 若处于 WMM PS (power save) 模式, 则不发送数据; 若处于 Active 模式, 则调用 `wlan_process_tx()` 检查数据发送条件。准备工作就绪后, 将数据传送给 `SendSinglePacket()`, 以进行单个数据包的发送。`SendSinglePacket()` 需要先对数据包做一些检查, 然后把数据包复制到 TxPD 类型区域中, 最后调用 `sbi_host_to_card()` 将数据包发送到 Wi-Fi 模块硬件, 由固化在 Wi-Fi 模块硬件中的 Firmware 把数据发送出去。

数据成功发送到 Wi-Fi 模块硬件后将产生中断。该中断对应的中断处理函数所完成的主要工作包括清除相应状态位、释放 `sk_buff` 结构和通知系统可以再次发送。如果数据发送不成功, `dev->busy` 置位, 驱动程序会不断尝试重传, 若重传超时(如设备严重堵塞时), 驱动程序会将该数据包丢弃, 同时释放 `sk_buff` 结构。

2.2.3 WLAN 数据包接收

WLAN 驱动程序并没有提供专门的数据接收函数。网络设备在收到数据后会产生一个中断, 中断处理程序通过 `sbi_get_int_status()` 读取当前状态寄存器, 判断出数据接收状态, 申请一块大小合适的 `sk_buff` 缓冲区, 再通过 `sbi_card_to_host()` 把 Wi-Fi 模块硬件中的网络数据包读取到该 `sk_buff` 缓冲区, 并将该 `sk_buff` 添加到接收数据队列。从硬件中成功地把数据读进来后, `wlan_send_rxskbQ()` 调用 `ProcessRxdPacket()` 处理接收到的数据包, 并将其转发到协议层。

2.3 SPI 驱动程序

Host 和 Wi-Fi 模块之间的数据传输最终是通过 SPI 实现的, SPI 属于字符型设备, 应用程序通过字符设备文件来对 SPI 硬件进行操作^[6]。SPI 设备驱动程序的结构与大多数字符型设备驱动相似, 主要包含设备初始化和注销、打开关闭和数据读写等操作。

SPI 设备初始化通过 `gspihost_module_init()` 实现。该函数初始化 S3C2440 中与 SPI 相关的寄存器, 调用 `gspihost_init_hw()` 配置 `SPIMISO0`、`SPIMOSI0`、`SPICLK0`、`EINT1`

和 `nSS0` 引脚所对应的寄存器, 并通过 `register_chrdev` 将 SPI 注册为字符设备; 注销函数则执行相反的操作。

SPI 设备的打开和关闭通过 `gspihost_open()` 和 `gspihost_release()` 实现, 数据读写通过 `gspi_read_data()` 和 `gspi_write_data()` 实现。数据读写最终将分别调用 `gspi_read_data_direct()` 和 `gspi_write_data_direct()`, 按照 SPI 接口时序直接对寄存器进行读写, 实现数据的收发。此外, 这两个函数还会在 WLAN 数据收发中断处理函数中被调用。

3 Wi-Fi 驱动的加载和测试

嵌入式 Linux 设备驱动有内核加载和模块加载两种加载方式^[7], 本 Wi-Fi 设备驱动采用模块加载方式。Wi-Fi 驱动程序编译后生成 `gspi.ko` 和 `gspi8xxx.ko` 两个文件, 将配置好的内核编译下载到连接好 Wi-Fi 模块的 S3C2440 开发板后就可以加载驱动模块, 加载过程如下。

加载 SPI 驱动, 运行指令为 `# insmod gspi.ko`。

加载 WLAN 驱动, 运行指令为 `# insmod gspi8686.ko`
`helper_name = ./helper_gspi.bin`
`fw_name = ./gspi8686.bin`
`mfdmode = 1`。

成功加载 Wi-Fi 驱动后, 需要对该驱动进行测试, 具体操作步骤如下:

- ① 查看是否检测到设备, 运行 `iwconfig` 指令, 结果显示可以检测到;
- ② 配置设备接口, 运行 `# ifconfig eth1 up` 指令, 运行结果显示设置成功;
- ③ 扫描可用 AP, 运行 `# iwlist eth1 scan` 指令, 扫描到可用网络;
- ④ 连接到其中一个 AP, 运行 `# iwconfig eth1 txpower auto essid MyPlace channel 4` 指令;
- ⑤ 获取 IP 地址, 运行 `# udhcpc-I eth1` 指令, 获取 IP 为 192.168.10.216;
- ⑥ 运行 `# ping 192.168.10.1` 指令, 运行结果表明可以 ping 通。

测试结果表明, 该 Wi-Fi 驱动能够成功加载固件、扫描并连接到可用 AP 和网内计算机, 实现信息互通, 达到了预期目标。

4 结束语

作为目前无线接入的主流标准, Wi-Fi 应用将越来越广泛, 因此, 对 Wi-Fi 技术的研究非常具有实际意义。本文介绍了基于 ARM 硬件平台和 Linux 操作系

(下转第 7 页)

集的数据信息,需要把该数据存到临时数组里,依据地址采集下一个节点的数据信息,当整个监测区域的节点数据采集完毕后,根据临时数组里的数据作融合,并把最终结果传给监控中心。同时,在监控中心的上位机中建立一个数据库,分别将各节点的地址和该节点采集的数据一一对应起来,这样就实现了各用户用表数据的准确抄读。

4 结束语

基于 GPRS 无线通信技术的远程自动抄表系统,结合基于 ZigBee 技术的数据采集设备,改变了以往全人工抄表的模式,从本质上提高了抄表人员的工作效率和准确率;集中抄表范围广且安装、维护方便,不需要进行专门布线,同时可对表具设备进行远程控制、参数调整和开关等控制操作。在实际应用中,可根据抄表用户的不同分布,灵活地构建抄表的无线网络,甚至可以将 ZigBee 无线模块集成到电能表、水表和燃气表中,从而完全实现居民区集中抄表、无布线和快速组网以及三表统一抄收的功能。

参考文献

- [1] 沈建华,杨艳琴,翟晓曙. MSP430 系列 16 位超低功耗单片机实践与系统设计[M]. 北京:清华大学出版社,2005.
- [2] 魏小龙. MSP430 系列单片机接口技术及系统设计实例[M]. 北京:北京航空航天大学出版社,2003.
- [3] 许克滨,陈新,胡兰馨,等. 一种基于 GPRS 网络的 GPS 数据传输系统的实现[J]. 福州大学学报:自然科学版,2004(5):541-545.
- [4] 胡大可. MSP430 系列单片机接口技术及系统设计实例[M]. 北京:北京航空航天大学出版社,2002.
- [5] 龙玉湘,章兢,戴瑜兴. 基于 ZigBee 的无线抄表系统的集中器设计[J]. 低压电器,2007(20):14-17.
- [6] 李文仲,段朝玉. ZigBee 无线网络技术入门与实战[M]. 北京:北京航空航天大学出版社,2007:189-371.
- [7] 马永强,李静强,冯立营. 基于 ZigBee 技术的射频芯片 CC2430[J]. 单片机与嵌入式系统应用,2006(3):45-46.
- [8] 王盛慧,邓志东,裴忠民. 无线传感器网络室内空气污染物节点的硬件设计与实现[C]//2007 年中国智能自动化会议论文集. 长沙:中南大学出版社,2007:52-55.
- [9] ZigBee Alliance. ZigBee specification version 1.0[EB/OL]. [2010-06-05]. <http://www.zigbee.org>.
- [10] 高志国,李巧鸽,高新鹏. 视频监控数据远程传输的设计与实现[J]. 石油化工自动化,2010,46(5):51-53.

(上接第 3 页)

统的 Wi-Fi 设备驱动程序结构框架,具体分析了 WLAN 和 SPI 驱动实现的详细过程,并对所实现的驱动代码进行测试。测试结果表明,该驱动程序实现了所要求的所有功能,具有实时性好和加载方便等优点,为类似的 Wi-Fi 设备驱动开发提供了很好的参考价值。

参考文献

- [1] 沈韬,李绍荣. 无线网卡驱动分析与 WLAN 性能测试[J]. 通信技术,2009(10):105-110.
- [2] 黄猛,杜红彬. 移动机器车的 Wi-Fi 接口设计[J]. 自动化仪表,

2010,31(3):50-56.

- [3] 索炜. 基于 S3C2440 + Linux 的无线射频模块的驱动程序设计[D]. 北京:北京邮电大学,2008.
- [4] 范皖勇. 嵌入式 Wi-Fi 技术应用与研究[D]. 上海:华东师范大学,2008.
- [5] 朱轶,赵洁. Linux 内核 802.11 无线网络协议栈的设计与实现[J]. 计算机工程,2008,34(1):140-144.
- [6] 宋宝华. Linux 设备驱动开发详解[M]. 北京:人民邮电出版社,2008.
- [7] Corbet J, Rubini A, Kroah-Hartman G. Linux 设备驱动程序[M]. 魏永明,耿岳,钟书毅,译. 西安:中国电力出版社,2006.

行业信息

贝加莱获大同集团"2010 年度优秀战略合作伙伴"殊荣

近日,大同机械集团 2010 年年在东莞市隆重举行,贝加莱也应邀参加。会上,大同集团向与其合作的供应商颁发了“年度优秀供应商”和“年度优秀战略合作伙伴”两个奖项。凭借多年深入的合作,贝加莱获得了“优秀战略合作伙伴”这一殊荣。

大同集团在技术上追求领先,积极创新,这与贝加莱的创新理念非常契合,也正是因为这个原因,双方在过去 10 年里一直保持着紧密的合作。贝加莱产品的缩短“Time to Market”、降低成本、提升整机性能与竞争力方面充分满足了大同集团对于竞争力塑造的迫切需要;同时,在自动化方案研发理念、系统架构以及服务理念方面也相互认同。双方的合作不仅包括产品供应,同时还涉及方案集成、对未来产品架构设计以及共同服务终端客户等方面。

会上,大同集团展示了“2010 年技术革新成果”,其中,使用了贝加莱巧塑 88 控制器的 S09 注塑机系列是其最重要的一个成就,众多参与此项目的员工也获得了大同集团年度优秀员工的称号。

2011 年,作为长期的战略合作伙伴,贝加莱将与大同的优秀团队继续合作,深入参与未来更具竞争力的产品开发计划,共同研发最具有竞争力的机器系统。相信在不久的将来贝加莱与大同机械将共同取得更加辉煌的战绩。