

## SSPA 电源工作状态远程监测系统设计

李长春, 武 丽

(西南科技大学 信息工程学院, 四川 绵阳 621000)

**摘要:** 设计了一种功率放大器电源工作状态远程监测系统, 实现了 137 路电源的监测。该系统以 ARM7 处理器、 $\mu\text{C}/\text{OS-II}$  实时操作系统和 uIP 协议栈为核心, 并在 PC 上设计了上位机监控软件。经验证, 系统能正确接收多点监测, 可以及时反映电源工作情况, 并提供报警信息。

**关键词:** 电源监测; ARM7;  $\mu\text{C}/\text{OS-II}$ ; uIP 协议; VC

中图分类号: TP39

文献标识码: A

文章编号: 1674-7720(2013)18-0047-02

## Design of remote monitoring system for working state of SSPA power supply

Li Changchun, Wu Li

(Information Engineering College, Southwest University of Science and Technology, Mianyang 621000, China)

**Abstract:** Here is the design of a remote monitoring system for power status of the power amplifier, realizing 137-channel power monitoring. This system used ARM7 processor,  $\mu\text{C}/\text{OS-II}$  real time operating system and uIP protocol stack as the core, and designed monitoring software on the PC. After the test, this system can correctly receive multi-point monitoring, timely reflect the power status and provide alarm information.

**Key words:** power monitoring; ARM7;  $\mu\text{C}/\text{OS-II}$ ; uIP agreement; VC

固态功率放大器 SSPA (Solid State Power Amplifier) 具有体积小、重量轻、工作电压低、稳定性高等优点, 被广泛用于各个领域, 在通信系统中也占据着越来越重要的地位。固态功率放大器需要将多路功率器件进行合成, 这就需要多路电源进行稳定供电, 在工作中需要对每一路电源的工作状态进行监测, 主要监测电源的工作电流。对固态功率放大器工作时的电源工作状态进行监测, 并通过网络发送至远程主机进行分析和处理, 以确保系统能够正常工作。

## 1 系统硬件结构

SSPA 远程监测系统由电源模块、模拟开关、嵌入式处理器 AT91SAM7X256 和 DM9161 网卡等构成。多路电源的工作电流经过电路转换为电压, 输入到系统中, 系统通过模拟开关的切换实现分时采样, 得到 137 路电源的工作电流数据, 并将数据打包, 然后 DM9161 通过网络将数据发送至远程主机, 实现了远程采集和监测功能, 当采集的工作参数超过设定值时实现报警, 确保系统工作正常。监测系统构成如图 1 所示。

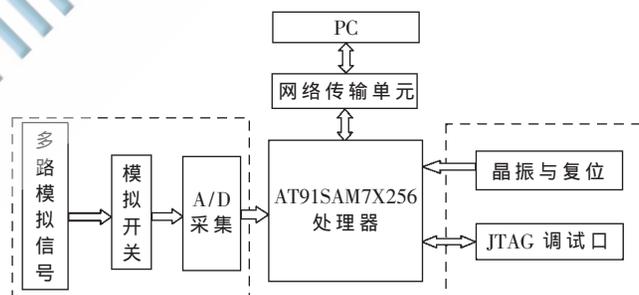


图 1 系统硬件结构图

## 2 系统软件设计

2.1  $\mu\text{C}/\text{OS-II}$  系统移植

$\mu\text{C}/\text{OS-II}$  是一个优秀的嵌入式实时操作系统, 应用中首先应完成系统的移植。所谓移植, 就是使一个实时内核运行在某个微控制器或微处理器上, 并对与处理器相关的代码进行修改。系统移植主要包括以下几个内容:

(1) 在 OS\_CPU.H 头文件中完成配置和定义。其中主要包括与编译无关的数据类型的定义; OS\_ENTER\_CRITICAL、OS\_EXIT\_CRITICAL、OS\_TASK\_SW 3 个宏的定义, 这 3 个宏定义分别表示进入中断、退出中断和任务切换; 还需要完成堆栈生长方向的定义<sup>[1]</sup>。

欢迎网上投稿 [www.pcachina.com](http://www.pcachina.com) 47

## 网络与通信 Network and Communication

(2) 编写 OS\_CPU\_C.C, 其中主要的工作是编写 6 个 C 语言的函数。其中 OSTaskStkInit() 是任务堆栈初始化函数, 这是进行移植时必须重视的函数, 其他 5 个函数都是 Hook 函数, 供系统内核扩展。

(3) 编写 OS\_CPU\_A.ASM, 主要需要编写 3 个汇编语言函数, 分别是高优先级就绪任务启动函数 OSStartHighRdy()、任务级的切换函数 OSCtxSw() 和中断下的任务切换函数 OSIntCtxSw()<sup>[2]</sup>。

### 2.2 uIP 网络协议移植

uIP 由瑞典计算机科学学院(网络嵌入式系统小组)的 Adam Dunkels 开发, 是一种免费的、可实现的、极小的 TCP/IP 协议栈。它可以看作是一个代码库, 通过一系列函数为系统底层和应用程序之间提供通信, 对于系统本身, 它内部的协议是透明的, 提高了协议的适用性<sup>[3]</sup>。

uIP 使用前需要进行移植, 移植的主要内容就是修改网络驱动部分的内容, 包括以下几个步骤:

(1) 编写 EMAC 网络接口库, 该库文件中主要提供了一系列对 ARM 控制器中 EMAC 接口进行操作的函数, 控制器对网卡的操作都是通过该接口进行, 包括了 EMAC 模块的初始化工作以及与网卡芯片的通信函数等。

```
EMAC_Init(AT91C_ID_EMAC, MacAddress.addr, EMAC_CAF_ENABLE, EMAC_NBC_DISABLE); //初始化 EMAC 接口
.....
```

(2) 编写网卡芯片 DM9161 的驱动程序, 包括了 DM9161 的芯片初始化以及数据发送和数据接收操作等。

```
MACB_Init(pMacb, BOARD_EMAC_PHY_ADDR);
//初始化 DM9161
.....
```

(3) 完成本系统 uIP 的配置选项, 包括数据存储方式、缓存区大小、IP 地址、MAC 地址、默认路由、子网掩码、uIP 1 s 时间所需要的中断数和一些基本数据类型的定义, 以及最大连接数、数据包长度以及应用层协议的支持等<sup>[4]</sup>。

(4) 编写应用层函数 UIP\_APPCALL, uIP 协议在接收到底层数据之后, 需要上层函数处理时, 会调用函数 UIP\_APPCALL。本系统中通过使用宏定义 #define UIP\_APPCALL sample\_appcall 将该函数定向到 sample\_appcall。该函数通过查询当前状态来实现各种处理机制。当远程主机建立一个连接后, 初始化缓存区给当前连接使用; 有新数据到达时读取数据包, 分析数据包的信息后, 返回给远程主机所需要的信息。当连接轮询次数达到设定值时, 中止当前连接, 当远程主机发出断开连接命令或连接超时后, 断开当前连接<sup>[5]</sup>。uIP 协议栈通过调用该函数实现网络数据的收发。

### 2.3 系统任务设计

系统的实际任务有启动任务 AppStartTask()、AD 采集任务 AD\_sampleTask()、网络主任务 Network\_MainTask()、网络周期任务 Network\_PerioTask() 及网络数据包接收任

务 Network\_PollTask(), 其优先级从高到低。

AppStartTask() 是在开始时启动其他任务以及在系统运行时通过 LED 灯指示系统是否正常工作。AD\_sampleTask() 是采样任务, 负责各监测点数据的获取和处理工作。Network\_PollTask()、Network\_MainTask() 和 Network\_PerioTask() 3 个任务是网络部分的内容, 并且使用一个信号量 pNetwork\_newdata 来实现任务之间的通信。

数据包接收任务 Network\_PollTask() 是 3 个网络任务中优先级最低的一个, 负责监听在指定的端口上是否有连接请求或数据请求, 当一个新的请求到达该监听端口后, 该任务将收到的数据包缓存到 uip\_buf 中, 设置数据长度变量 uip\_len, 然后该任务会发送一个信号量 pNetwork\_newdata 给网络主任务 Network\_MainTask()。

网络主任务 Network\_MainTask() 是网络任务中优先级最高的, 完成网络数据处理并向应用层分发。平时一直处于挂起的状态, 当收到数据包发送任务发送的信号量之后, 该任务调用函数 uip\_process() 处理数据包信息, 根据不同信息完成数据包的发送, 然后继续被挂起。

Network\_PerioTask() 则是周期性运行的任务, 其优先级处于另两个任务之间, 用于对连接的管理, 处理连接超时数据包重发。这 3 个任务互相协作, 实现了网络的通信<sup>[6]</sup>。

### 3 上位机设计

上位机部分是基于 VC6.0 编写的, 主要运用网络套接字和多线程编程技术等实现。上位机运行时, 首先读取注册表中保存的设定值, 然后对保存的指定 IP 的指定端口发送连接请求, 连接建立后, 向下位机发送数据请求包, 并且在主线程之外创建一个接收线程, 接收线程通过 Socket 套接字接收上传的网络数据, 主线程中实现各个监测点的数据显示以及报警信息等, 并可以发送相应的功能命令给下位机, 以达到一些简单的控制功能。上位机运行界面如图 2 所示。



图 2 上位机运行界面

本系统实现了  $\mu\text{C}/\text{OS-II}$  操作系统和 uIP 网络协议的结合, 并运用到电源监测系统中。本系统经调试, 软硬

件工作均正常。实际运行结果证明,本监测系统在稳定性和实时性方面都有很好的表现,完全适用于实际系统的应用。随着嵌入式技术以及网络技术的发展,未来数据监测系统的发展和应用将会有更大的空间。

#### 参考文献

- [1] 邵贝贝.  $\mu\text{C}/\text{OS-II}$ —源代码公开的实时嵌入式操作系统[M].北京:中国电力出版社,2001.
- [2] 赵伟国,李文军,梁国伟.实时嵌入式操作系统  $\mu\text{C}/\text{OS-II}$  在 AT91 上的移植[J].中国计量学院学报,2005,16(2):137-139.
- [3] DUNKELS A.The uIP 1.0 reference manual[D].Swedish:

Swedish Institute of Computer Science,2006.

- [4] DUNKELS A.uIP-A free small TCP/IP stack[Z].2004.
- [5] 张永涛,黄丹丹,李欧.uIP 协议分析及其应用[J].信息工程大学学报,2006,7(2):147-149.
- [6] 刘春风,张代远. $\mu\text{C}/\text{OS-II}$  下协议栈 uIP 的移植与应用[J].计算机技术与发展,2012,22(9):143-145.

(收稿日期:2013-04-21)

#### 作者简介:

李长春,男,1989年生,硕士研究生,主要研究方向:嵌入式系统。

武丽,女,1968年生,副教授,主要研究方向:嵌入式系统,系统集成与综合自动化。

电子技术应用  
APPLICATION OF ELECTRONIC TECHNIQUE  
www.ChinaAET.com