

# 基于 $\mu\text{C}/\text{OS-II}$ 和 LwIP 的嵌入式 Web 服务器实现

杨俊 吕建平 徐峰柳

(苏州大学 电子信息学院 江苏苏州 215006)

**摘要:** 采用以 ARM Cortex-M3 为内核的 32 位微控制器 LPC1768 利用其内置以太网控制器搭建 web 服务器。web 服务器以  $\mu\text{C}/\text{OS-II}$  为操作系统,并在其基础上成功移植了 LwIP 协议栈,通过该协议栈实现了 HTTP(超文本传输协议)服务。文中介绍了该系统的硬件设计和软件开发过程,涉及  $\mu\text{C}/\text{OS-II}$  的移植、LwIP 协议栈的总体架构和移植、LwIP 协议栈数据包处理流程、以及网络应用层程序的编写。

**关键词:** ARM Cortex-M3  $\mu\text{C}/\text{OS-II}$  LwIP 浏览器

[中图分类号] TP332 [文献标志码] A [文章编号] 1000-3886(2011)03-0062-03

## Realizing of Embedded Web Server Based on Light Weight Protocol Stack LwIP and $\mu\text{C}/\text{OS-II}$

Yang Jun Lu Jianping Xu Fengliu

(School of Electronics and Information Engineering, Soochow University, Suzhou Jiangsu 215006, China)

**Abstract:** A 32bit-microcontroller LPC1768 based on the core of ARM Cortex-M3 was adopted in this server. Its embedded Ethernet controller was used to construct a web server with  $\mu\text{C}/\text{OS-II}$  as the operating system(OS). On the basis of  $\mu\text{C}/\text{OS-II}$ , a LwIP protocol stack was transplanted successfully and HTTP(Hyper Text Transfer Protocol) service was realized. The process of hardware designing and software developing was introduced in the paper. Adding to it, the paper included transplanting of  $\mu\text{C}/\text{OS-II}$ , the general structure and transplanting of LwIP protocol stack, the handling proceeding of LwIP protocol stack packet, as well as programming of application layer.

**Keywords:** ARM Cortex-M3  $\mu\text{C}/\text{OS-II}$  LwIP browser

## 0 引言

随着嵌入式系统和单片机技术的发展,嵌入式以太网的设计越来越受关注。美国贝尔实验室总裁 Arun Netravali 等一批科学家曾经作出了预测:未来"将会产生比 PC 时代多出成百上千倍的瘦服务器和超级嵌入式瘦服务器,这些瘦服务器将与我们这个世界你能想象到的各种物理信息、生物信息相连接,通过 Internet 自动地、实时地、方便地、简单地提供给需要这些信息的对象"<sup>[1]</sup>。因此,研究和设计嵌入式网络通信将具有十分重要的价值。本文以 LPC1768、DP83848CVVP 芯片搭建硬件平台,构建以  $\mu\text{C}/\text{OS-II}$ 、LwIP 为基础的软件系统,在此基础上实现 web 服务,从而达到通过浏览器对其进行访问的目的。

## 1 $\mu\text{C}/\text{OS-II}$ 系统分析及移植

$\mu\text{C}/\text{OS-II}$  是由美国人 Jean J. Labrosse 编写的一个开源代码的微内核嵌入式实时操作系统,其实时性能和内核的健壮性早已在大量的实际应用中得到了证实。具有如下特点:①开源代码 ②移植性好 ③可裁剪 ④稳定性和可靠性高。

### 1.1 $\mu\text{C}/\text{OS-II}$ 系统分析

$\mu\text{C}/\text{OS-II}$  是多任务系统,内核负责管理各个任务,每个任务都有其优先级, $\mu\text{C}/\text{OS-II}$  最多可以管理 64 个任务,其每个任务都拥有自己独立的堆栈。 $\mu\text{C}/\text{OS-II}$  提供了非常丰富的系统服务功能,比如信号量、消息邮箱、消息队列、事件标志、内存管理和时

间管理等,这些功能可以帮助用户实现非常复杂的应用。

### 1.2 $\mu\text{C}/\text{OS-II}$ 移植

$\mu\text{C}/\text{OS-II}$  作为操作系统,全部源代码大约是 6 000 ~ 7 000 行,文件结构如图 1 所示,移植主要是处理与处理器相关的源文件,即 os\_cpu.c、os\_cpu\_a.s、os\_cpu.h。其中 os\_cpu.h 主要包含编译器相关的数据类型的定义、堆栈类型的定义以及几个宏定义和函数说明,为了便于移植,须重新定义数据类型,因为不同的编译器所提供的同一数据类型的数据长度并不相同。os\_cpu.c 文件中主要实现任务堆栈的初始化函数 OSTaskStkInit(),任务创建函数通过调用函数 OSTaskStkInit() 初始化任务堆栈结构。在 os\_cpu\_a.s 文件中需要根据具体的硬件处理器实现几个汇编函数 OSStartHighRdy()、OSCtxSw()、OSIntCtxSw 以及临界处理函数 OS\_CPU\_SR\_Save、OS\_CPU\_SR\_Restore。尤其需要注意的是,由于 LPC1768 是 cortex-m3 内核,故 OSCtxSw()、OSIntCtxSw 函数

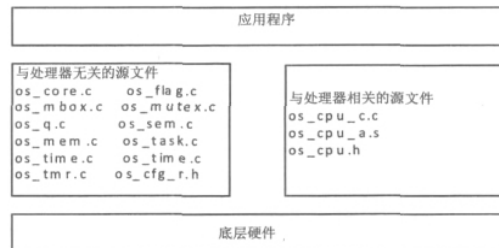


图 1  $\mu\text{C}/\text{OS-II}$  系统文件结构

收稿日期: 2010-11-29

都是通过触发 pendsv(可挂起中断服务)来完成,它负责  $\mu\text{C}/\text{OS}-\text{II}$  的全部上下文切换。

## 2 LPC1768 的网络接口设计

由于 LPC1768 的以太网控制器只集成了介质访问控制器(MAC),缺少物理层(PHY),故本系统选用的 PHY 芯片为 DP83848CVV 芯片,由 National Semiconductor 公司生产,RJ45 接口选用 HanRun 公司生产的 HR911105A 接口。DP83848CVV、HR911105A 与 LPC1768 的硬件连接示意图如图 2 所示。1768 的以太网模块使用 RMII(简化的媒体独立接口)协议和片上 MII(M 媒体独立接口管理)串行总线、MDIO(管理数据输入/输出)来实现与片外 PHY 之间的连接,MII 接口可对 PHY 寄存器进行访问。RMII 对应的引脚为 ENET\_TX\_EN、ENET\_TXD[1:0]、ENET\_RXD[1:0]、ENET\_RX\_ER、ENET\_CRD、ENET\_REF\_CLK,接口 MII 对应的引脚为 ENET\_MDC、ENET\_MDIO。需要注意的是使用 DP83848CVV 芯片的 RMII 模式必须外接 50MHz 的有源晶振。

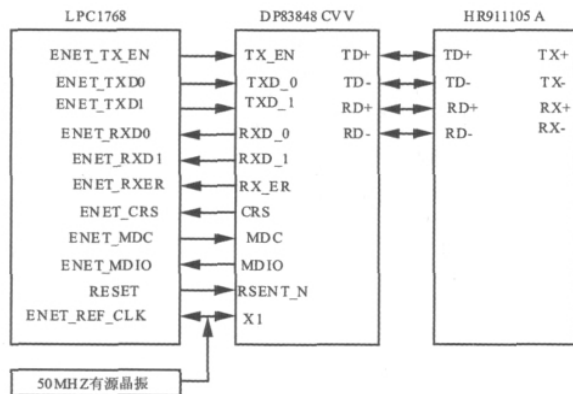


图 2 LPC1768 网络连接示意图

## 3 LwIP 介绍

LwIP 是瑞士计算机科学院(Swedish Institute of Computer Science)Adam Dunkels 等开发的一套用于嵌入式系统的开放源代码 TCP/IP 协议栈。

### 3.1 LwIP 协议栈的体系结构

LwIP 的体系结构如图 3 所示,由图可知:LwIP 由一些相对独立的模块组成,除 TCP/IP 协议的实现模块(IP、ICMP、UDP、TCP)外,还包括许多相关的支持模块,这些支持模块包括操作系统模拟层、缓冲与内存管理模块、网络接口函数以及一组 Internet 校验和计算函数、API 接口函数。

### 3.2 LwIP 移植:实现操作系统模拟层

LwIP 的操作系统模拟层(sys\_arch)是底层操作系统和 LwIP 之间的一个接口,为 LwIP 提供信号量和邮箱 2 种进程间通信方式。当用户移植 LwIP 时,只需修改这个接口即可。主要实现以下几种类型的函数,分别是信号量操作函数、邮箱操作函数、临界保护函数、sys\_thread\_new() 函数、sys\_arch\_timeouts() 函数。由于  $\mu\text{C}/\text{OS}-\text{II}$  提供了创建任务函数、临界保护函数以及丰富的信号量和邮箱操作函数,所以只需对这些函数稍加修改,便可实现 LwIP 操作系统模拟层的函数。

### 3.3 LwIP 协议栈的 TCP 通信流程



图 3 LwIP 协议栈体系结构

TCP 属于传输层协议,它为应用层提供可靠的字节流服务,建立在 TCP 协议之上的应用层协议有 HTTP 等。本文采用的是 HTTP 协议。基本的 TCP 处理过程被分割为 6 个功能函数来实现:tcp\_input

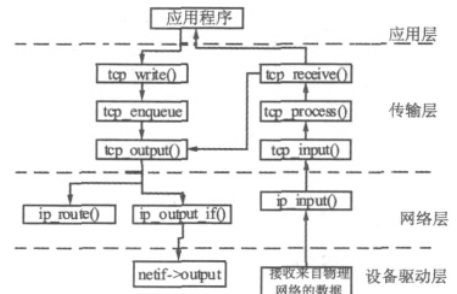


图 4 TCP 层通信流程

()、tcp\_process()、tcp\_receive()、tcp\_write、tcp\_enqueue、tcp\_output。实现框图如图 4 所示。

### 3.4 LwIP 主线程 tcpip\_thread

主线程 tcpip\_thread 是一个主控循环,底层接口函数通过 tcpip\_thread 和 lwip 的内部函数进行通信,比如在进行 tcp 传输时,底层接口函数通过传递邮箱 mbox 的方式进入到 tcpip\_thread,然后在该函数中,进入 tcp 层的 tcpip\_input 函数中,实现 tcp 层的传输。

### 3.5 LwIP API

LwIP API 专为 LwIP 设计,所以它可以充分利用 LwIP 的内部结构以实现其设计目标,不需要在应用程序和协议栈之间复制数据,因此应用程序可以巧妙地直接处理内部缓冲区。LwIP API 使用 2 种数据类型,分别为:①、netbuf,描述网络缓存的数据类型,对应的函数为 netbuf\_new()、netbuf\_delete()、netbuf\_alloc() 等;②、netconn,描述网络连接的数据类型,对应的函数为 netconn\_new、netconn\_delete()、netconn\_bind、netconn\_connect()、netconn\_accept() 等。LwIP 的上层应用程序是基于 LwIP API 编写的。

## 4 HTTP 协议

超文本传送协议(HTTP)是实现 www 的应用层协议,HTTP 负责 web 服务器与 web 客户(浏览器)之间的实际超文本文档的传送。

### 4.1 HTML 语言

网页就是一个 HTML 语法格式的纯文本文档,HTML 是网络所基于的标准的格式化语言。HTML 文档与网页在浏览器的显示并不相同,浏览器显示的是对 HTML 文档的解释。

### 4.2 HTTP 报文格式

一个 HTTP 事物包含一个 HTTP 请求和一个 HTTP 应答报文,这两种报文都遵循一个通用结构,被称为 HTTP 报文格式,如图 5 所示。由图可知,请求行是 HTTP 请求报文的起始行,用于指示命令或客户要求服务器执行的动作、方法、要引用的资源以及客户使用的 HTTP 版本。该行格式如下: <IMETHOD> <I 请求的 URI> <IHTTP 版本>,其中方法(method)告知服务器要执行的动作。方法的种类有"get"、"post"、"HEAD"等。本系统实现的是利用"get"请求向服务器提取网页。

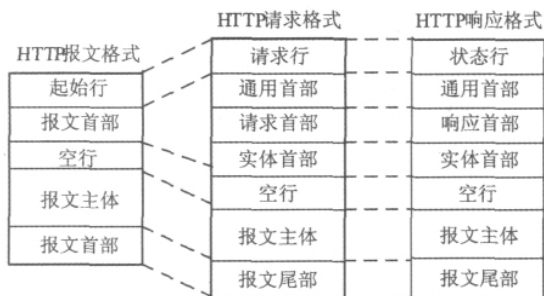


图 5 HTTP 报文格式

## 5 系统实现

本系统的实现是通过在  $\mu C/OS-II$  下建立的 4 个任务完成,分别是:① 起始任务  $task\_start()$ , 优先级为 0, 用于创建其他 3 个任务并进行 LwIP、EMAC 的初始化, 执行之后被挂起, 永不再执行;② 底层硬件接收任务  $ethernetif\_input()$ , 优先级为 4;③  $tcpip\_thread()$  任务, 优先级为 1;④  $http\_server$  任务, 优先级为 10。任务②、③、④依靠  $\mu C/OS-II$  下的信号量  $sem$  和邮箱  $mbox$  进入就绪态, 否则被挂起。如图 6 所示:

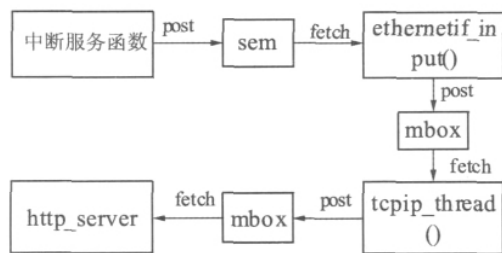


图 6 ethernetif\_input、tcpip\_thread、http\_server 通信流程

当本系统上电时,由于此时浏览器并没有发送数据,网络接口没有数据,故没用中断发生,sem 和 mbox 无法投递,所以此时这 3 个任务完全由  $\mu C/OS-II$  内核的系统定时器中断依据优先级而发生切换,程序流程如图 7。tcpip\_thread 的优先级最高, tcpip\_thread 首先被运行,但由于没有邮箱被投递,故该任务随之被挂起,在内核调度的作用下,ethernetif\_input() 任务开始运行,同理,sem 也没有投递,随之挂起,然后进入到 http\_server 任务,随之也被挂起,此时系统进入空闲任务,等待中断的发生,即等待浏览器的访问。

web 服务器的实现:

$http\_server$  任务是基于 lwip API 编程的,程序流程如图 8 所示,由图可以看出,在进入到  $netconn\_accept$  函数时,等待邮箱  $accept\_mbox$  的到来,否则被挂起。 $accept\_mbox$  邮箱是在浏览器和

服务器三次握手之后由  $accept\_function()$  投递,此时程序进入  $netconn\_recv()$  函数,等待  $recv\_mbox$  的到来。 $recv\_mbox$  的投递发生在数据被传至 tcp 层时,由  $recv\_tcp$  函数投递的。一旦  $recv\_mbox$  被 post,则进入到  $netbuf\_data()$  函数,用于获得接收数据指针,这个指针指向 netbuf 的第一个数据片段且这个片段包含 HTTP 请求,若是"get"请求,则发送网页,若不是则关闭连接,等待下一个连接到来。

## 6 结束语

本文详细介绍了利用 LPC1768 以太网控制模块,在  $\mu C/OS-II$  操作系统的基础上,移植 LwIP 协议栈,搭建嵌入式 web 服务器的过程。由于搭建的 web 服务器具有网页数据动态显示功能,可以满足一般的嵌入式设备进行远程监控,所以本系统实现对嵌入式设备的远程检测和控制具有很好的借鉴意义。

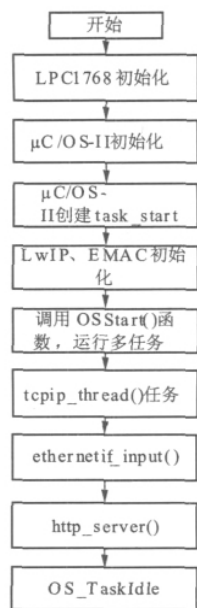


图 7 未发生中断时,系统程序流程图

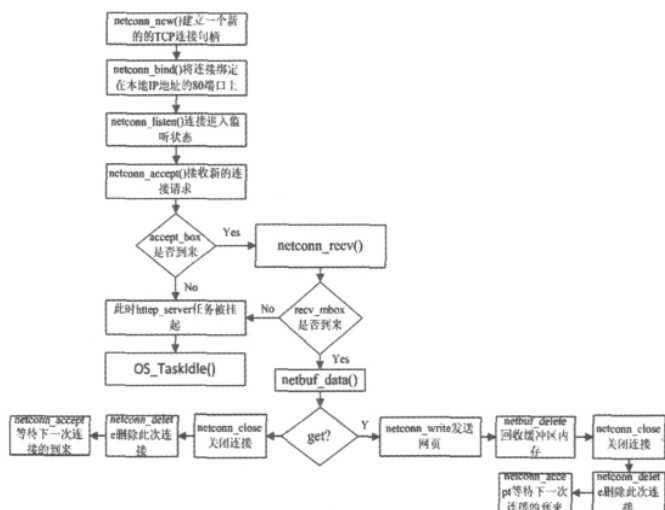


图 8 http\_server 任务流程图

## 参考文献

- [1] 李纪栓. 嵌入式 INTERNET 接入模块的设计[J]. 机械管理开发, 2006, 5.
- [2] 焦海波, 刘健康. 嵌入式网络系统设计—基于 Atmel ARM7 系列[M]. 北京: 北京航空航天大学出版社, 2008. 4.
- [3] (美) JEAN J. LABROSSE. 嵌入式实时操作系统  $\mu C/OS-II$  (第 2 版)[M]. 邵贝贝译. 北京: 北京航空航天大学出版社, 2003. 5.
- [4] 郑巨明, 张和生, 贾利民等. 基于  $\mu C/OS-II$  和 LwIP 的嵌入式以太网接口设计[J]. 计算机测量与控制, 2009, 17(11).
- [5] 熊海泉.  $\mu C/OS-II$  下 LwIP 协议的移植实现[J]. 科技广场, 2005. 2.
- [6] 苏义鑫, 毛晓磊. uIP 在 S3C44B0X 上的移植研究及 webserver 实现[J]. 微型电脑应用, 2008, 24(2).

【作者简介】 杨俊(1987-),男,电子信息学院 信号与信息处理。