

Research and Implementation of ZigBee Gateway —Based on Linux System Web Server

ZHANG Yan-feng, LI Mian, LIU Shou-yin

Advanced Communication Technology Laboratory, Huazhong Normal University, Wuhan, 430079, China

1.email:yanfengzh@live.cn 2.email:hr_lm@163.com 3.sy_liu@phy.cnu.edu.cn

Abstract: As the Internet of Things is proposed, people have increasing demand on the intelligentize of devices. Therefore, how to achieve the interconnection between the Internet and WSN becomes a hot topic of research. This article discusses the implementation of Web Server based on HTTP protocol and analyses the operational theory of the AJAX script. Then a dynamic Web server based on ARM9 hardware platform and Linux software is achieved, in which the dynamic interactivity between the browser and the Web Server is realized. Considering the application of ZigBee in remote monitoring, the design of ZigBee Gateway as a Web server based on Linux and ARM9 is researched. Finally, it is verified in the experiment.

Key words: ZigBee Gateway; Internet of Things; Web Server

基于 Linux 系统的 Web 服务器型 ZigBee 网关的研究与实现

张艳峰, 李 勉, 刘守印

华中师范大学 现代通信技术实验室 湖北武汉市 430079

1.email:yanfengzh@live.cn 2.email:hr_lm@163.com 3.sy_liu@phy.cnu.edu.cn

【摘要】随着物联网的提出,人们对设备智能化的要求越来越高,如何实现Internet网络与WSN网络的互连成为人们研究的热点问题之一。本文在详细阐述HTTP协议以及Web服务器的实现原理,分析AJAX脚本工作方法的基础上,以ARM9为硬件平台,设计了一种Web服务器型ZigBee网关。该网关不仅能实现ZigBee网络与Internet网络之间的协议转换,而且使得用户可以通过IE浏览器来访问控制ZigBee网络。针对ZigBee在设备远程监控、管理方面的应用,本文最后通过具体的项目对上述网关进行了实际验证。

【关键字】 ZigBee 网关; 物联网; Web 服务器

0 引言

随着 ZigBee 技术的发展,越来越多的具有 ZigBee 功能的终端设备出现在各种应用领域,实现对各种终端设备进行实时的远程监测与控制。然而, ZigBee 网络在通信能力上的缺陷,使得我们需要建立一个嵌入式 ZigBee 网关,实现 ZigBee 局域网与 Internet 的无缝连接。本文详细阐述了一种基于 Web 服务器的 ZigBee 网关的实现方法。

1 Web 服务器型 ZigBee 网关性能概述

随着物联网[1]概念的提出,对物联网实现的架构和协议的研究成为目前研究的热点问题之一。ZigBee+Internet 可能是简单、方便可实现的候选方案之一。其中网关是实现 ZigBee 与互联网连接的核心设备。

网关是一种建立在传输层之上的协议转换器,通常连接两个或多个相互独立的网络,每接收一种协议的数据包后,在转发之前将它转换为另一种协议的格式。ZigBee 网关实现把 ZigBee 无线网络数据传输至计算机、Internet 网络,实现计算机、Internet 网络与 ZigBee 无线网络的互联。

然而,传统的 ZigBee 网关仅仅对终端设备信息进行简单转发,不能满足物联网提出的智能化要求。本系统所设计的 ZigBee 网关实现了对 ZigBee 终端设备节点的实时状态监测、历史数据存储、信息修改、故障处理及上报等功能。并通过合理的任务调度策略实现了 ZigBee 无线网络的安全、高效、可靠运行。同时, ZigBee 网关作为一种网络服务器设备,需要为远程用户提供各种网络服务。本系统所设计的 ZigBee 网关提供了 Web 服务,使 Internet 远端用户可以通过 IE 浏览器(运行 AJAX 脚

本)来动态的监测、控制ZigBee网络节点。从而,有效的解决了ZigBee无线网络与Internet网络的互操作性问题。

网关与Internet的接入支持以太网、WiFi和GPRS等方式。实现了接入方式的多样性和灵活性。使得用户能根据环境条件选择合适的接入方式进行设备监测。

2 软硬件平台的构建

本系统选用基于ARM920T内核的S3C2440作为ZigBee网关的主处理器,它是一款适于中端应用的微控制器;选用符合802.15.4标准的CC2430作为ZigBee网关的无线收发器。该系统主要包括存储器(64MB SDRAM、256MB NANDFlash、2M NORFlash)、USB主从接口、一个100M的以太网控制芯片(DM9000)、两个UART接口(最高波特率为115200 bps)、一个RTC实时时钟、LCD显示模块,应用按键等硬件资源,满足了串口通信,数据存储和网络传输的需要。系统硬件框图如图1所示:

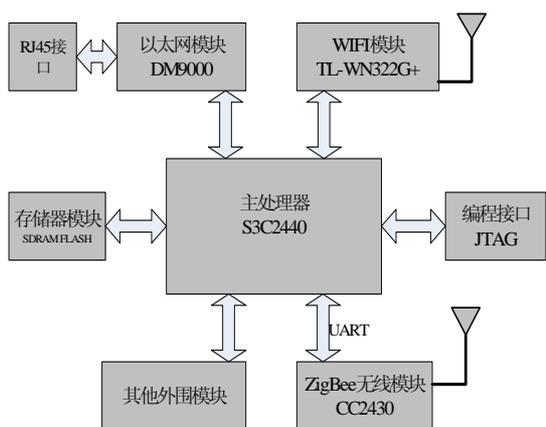


Figure 1.The platform of ZigBee Gateway
图1. ZigBee网关的硬件平台

当前应用较为广泛的嵌入式操作系统有: WinCE、Linux、VxWorks等。其中, Linux遵循GPL协议,具有开放源代码、网络性能优越以及强大的技术支持等优点。因此,本文选用Linux操作系统作为系统设计的软件平台。

3 ZigBee 网关的实现

本文采用 S3C2440 处理器以及 Linux2.6.32.4 内核来实现 Web 型 ZigBee 网关。网关通过串口与 ZigBee 协调器相连来获取 ZigBee 网络信息,然后在 Linux 操作系统下编写 Web 服务器软件来实现信息格式的转换,

并将结果存储到相应的数据库中。这样用户通过 IE 浏览器就能实现对 ZigBee 网络信息的查看、修改和设置。因此,在整个 ZigBee 网关的设计过程中, Web 服务器的设计是核心。

3.1 Web服务器的关键技术

3.1.1 HTTP的工作原理

超文本传输协议^[2](HTTP)是一种应用于分布式、合作式、多媒体信息系统的请求与响应式的应用层协议。HTTP通信一般基于TCP/IP协议族,默认端口号为TCP80。客户机在发出请求之前,必须先申请一个任意的、非占用的、非保留的端口以便进行通信。HTTP是一个“无连接”协议——每次连接只处理一个请求。服务器处理完客户的请求,并收到客户的应答后,即断开连接。通过扩展HTTP协议的请求方式、错误代码以及报头可以完成特定的任务。从而实现客户请求的高效处理。其具体的运行流程如图2所示:

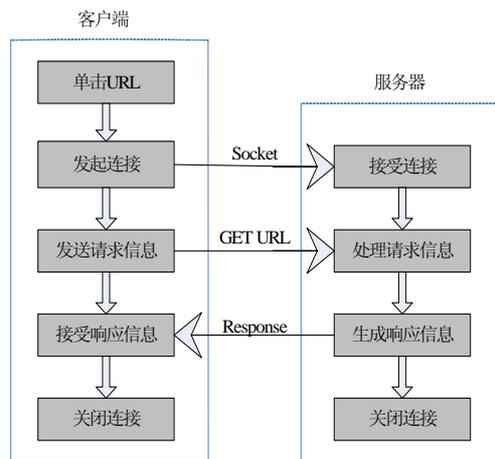


Figure 2.HTTP flow diagram
图2. HTTP流程图

3.1.2 AJAX的基本原理^[3]

AJAX (Asynchronous JavaScript And XML) 指异步JavaScript及XML,是一种在 2005 年由 Google 推广开来的编程模式。传统的Web应用程序模型工作方式是用户在界面上激发一个HTTP请求到Web服务器,服务器做一些处理(获取、处理数据),最后返回整个HTML页面到客户端。这样的模型适合于以超文本为基础的Web应用程序,却并不一定满足交互式应用程序的要求。

AJAX在浏览器与Web服务器之间使用异步数据传

输(HTTP请求),这样就可使客户端从服务器请求少量的信息,而不是整个页面。用户动作的处理由传统的发送一个HTTP请求,变为Javascript调用AJAX引擎。这相当于在用户与服务器之间添加了一个中间层,使用户操作与服务器响应异步化。把一些由服务器负担的工作转嫁到客户端,利用客户端闲置的处理能力来处理,减轻了服务器的负担,从而达到节约ISP空间及带宽租用成本的目的。同时,异步化响应模式极大的改善了用户体验。其基本流程如图3所示:

由事件触发,创建一个XMLHttpRequest对象,把HTTP方法(Get/Post)和目标URL以及请求返回后的回调函数设置到XMLHttpRequest对象,通过XMLHttpRequest向服务器发送请求,请求发送后继续响应用户的界面交互,只有等到请求真正从服务器返回的时候才调用callback()函数,对响应数据进行处理,将处理结果更新至页面的相应位置。从而实现页面的局部数据刷新。

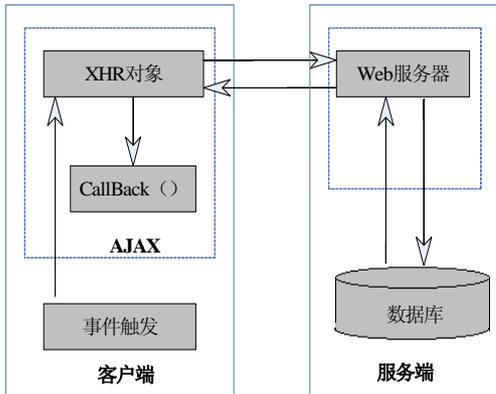


Figure 3.AJAX flow diagram
图3. AJAX工作流程图

3.2 Web服务器设计与实现

本文的Web服务器采用多线程技术,对多用户的并发访问提供了良好的支持。同时,引入了SQLite数据库,实现了对ZigBee节点历史信息、系统操作日志、通信状态的监测查询以及用户认证的功能。

3.2.1 主线程的实现

主线程进行相关软硬件资源的初始化,创建串口处理线程,实现与ZigBee局域网的通信。创建Socket套接字^[4],循环等待客户端的连接请求,接受请求后创建网络处理线程处理客户端请求。具体的流程图如图4所示

```

主线程的关键代码如下:
Sys_Init(); //初始化系统资源
thread=pthread_create(&ser_tid, null,
                    fun_ser_recv,(void*)&fd_serial);
if((socket(AF_INET,SOCK_STREAM,0)) < 0){
    perror( "error: create server socket!");
    exit(1);
}
.....
while(1){
    cli_fd=accept(ser_fd,(struct sockaddr*)&cliaddr,
                (socklen_t*)&addr_len);
    temp= pthread_create(&net_tid,null, fun_net_proc,
                        (void *)& cli_fd);
}
    
```

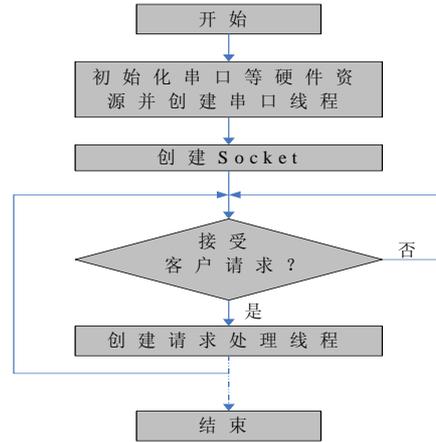


Figure 4.The flow diagram of main thread
图4. 主线程流程图

3.2.2 网络处理线程的实现

网络处理线程接受客户端请求,并依据HTTP协议,对请求路径和参数进行解析,执行相应处理并返回客户端请求的数据。其中核心部分是请求参数的解析以及对对应参数下的响应操作。我们在该线程函数的设计过程中,应用了线程同步技术,实现了代码的线程安全。流程图如图5所示:

```

参数解析关键代码:
switch(http_req_type)
{
    case GET_COM:
        proc_com(cli_tid,req_buff);
        break;
}
    
```

```

case GET_CMD:
    proc_cmd(cli_tid, req_buff);
    break;
case POST:
    proc_post(cli_tid, req_buff);
    break;
case GET_ASP:
    proc_asp(cli_tid, req_buff);
    break;
default:
    break;
break;
}
    
```

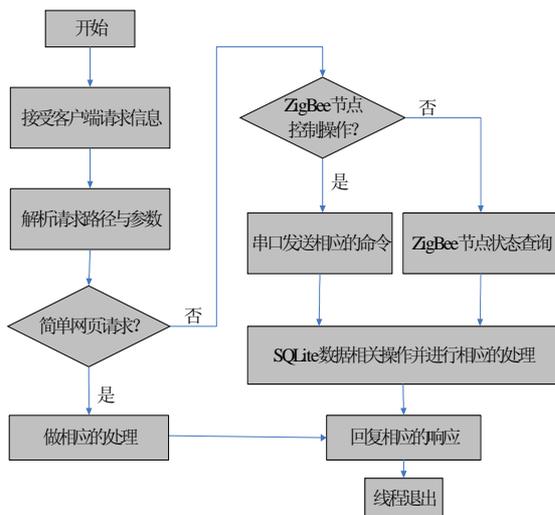


Figure 5.The flow diagram of process thread
图5 请求处理线程流程图

3.2.3 线程同步技术

线程是构造异步应用程序的一种方式，具有并行、并发、共享进程资源等优点，通常用于分布式服务器的设计。但是，当多个线程同时访问或者修改共享资源时，线程的并发性容易导致线程的安全问题，产生灾难性的后果。为了实现线程安全^[5]，本文使用了互斥锁、条件变量、信号量等同步技术，使串口线程与网络处理线程相互协调运行。有关互斥锁的代码如下：

```

char re_buff[MAXBUF+1]
    //线程共享接受数组
pthread_mutex_lock(&mymutex);
    //串口线程中同步相关代码
readn(seri_fd,re_buff,NUM);
pthread_mutex_unlock(&mymutex);
    
```

```

pthread_mutex_lock(&mymutex);
    //网络处理线程中同步相关代码
strcpy(content,re_buff);
pthread_mutex_unlock(&mymutex);
    
```

4 系统测试

以一个监测和控制隧道灯状态的应用项目为例来测试ZigBee网关。

ZigBee网络节点监测系统

华中师范大学现代通信技术实验室
actl.phy.ccnu.edu.cn

节点	节点地址	节点状态	采集时刻	备注	基本操作	
01	0001	OPEN	4.10-19.36	ACTLPHY.CCNU...	open	close
02	0002	OPEN	4.10-19.36	ACTLPHY.CCNU...	open	close
03	0003	OPEN	4.10-19.36	ACTLPHY.CCNU...	open	close
04	0004	CLOSE	4.10-19.36	ACTLPHY.CCNU...	open	close
05	0005	CLOSE	4.10-19.37	ACTLPHY.CCNU...	open	close
06	0006	OPEN	4.10-19.37	ACTLPHY.CCNU...	open	close
07	0007	CLOSE	4.10-19.37	ACTLPHY.CCNU...	open	close
08	0008	OPEN	4.10-19.37	ACTLPHY.CCNU...	open	close
09	0009	CLOSE	4.10-19.38	ACTLPHY.CCNU...	open	close
10	000A	OPEN	4.10-19.38	ACTLPHY.CCNU...	open	close
11	000B	OPEN	4.10-19.38	ACTLPHY.CCNU...	open	close
12	000C	OPEN	4.10-19.38	ACTLPHY.CCNU...	open	close
13	000D	OPEN	4.10-19.39	ACTLPHY.CCNU...	open	close
14	000E	CLOSE	4.10-19.39	ACTLPHY.CCNU...	open	close
15	000F	OPEN	4.10-19.39	ACTLPHY.CCNU...	open	close

Figure 6.The result of system testing

图6 系统测试结果

(1) 构建嵌入式Linux系统：修改supervivi代码中的分区结构、系统晶振和需要传递给内核的参数等部分内容后，编译supervivi，并下载到硬件平台的NANDFLASH中；接着，通过make menuconfig命令来裁剪Linux2.6.32.4内核，将串口驱动、网口驱动等系统需要的部分选中，然后编译内核，生成镜像文件zImage，通过supervivi来下载到硬件平台。

(2) 构建Web服务器：编写服务器代码以及相关的网页，将网页放置到Web目录下，通过arm-linux-gcc来交叉编译服务器代码。将Web目录和交叉编译后的服务器代码下载到硬件平台的同一文件目录下，并运行服务器。

(3) 测试结果：在IE浏览器输入目标板的IP地址，就会进入用户登陆界面进行用户认证，输入用户名和密码无误后才能进入ZigBee网关服务器的主界面。

主界面动态的监测ZigBee网络中各节点的信息(包

含短地址、状态以及数据采集时间)，用户通过相应的操作实现对节点当前状态的控制，并修改、查看节点的详细信息。当出现故障节点时，通过弹出窗口进行提示，并根据用户命令进行相应的故障处理。具体测试结果如图6所示：

5 结束语

本文详细介绍 HTTP 协议、AJAX 实现原理以及多线程技术。并在此基础上设计了一种新型的基于 ArmLinux 的 ZigBee 网关。采用了 B/S 模式，实现远程客户端对本地节点的监测控制，从而实现对 ZigBee 网络的扩展与延伸。该系统平台在隧道灯控制系统中运行良好，具有一定的平台通用性，对物联网的推广、普及具有一定的研究价值和实用价值。

References (参考文献)

- [1] Li Wenqing, Guo Zongliang The review of Internet of Things' growing and development[J]. Network and Information, 2010, 02(24):27-28
李文清, 郭宗良. 物联网的成长与发展综述[J]. 网络与信息, 2010, 02(24):27-28
- [2] Xu Yujie The TCP/IP further analysis[M]. Beijing: Tsinghua university press 2009
徐宇杰. TCP/IP 协议深入分析[M]. 北京:清华大学出版社, 2009
- [3] Ryan Asleson, Nathaniel T. Schutta. The basis tutorial of AJAX[M]
Jin Ling, translate. Beijing: Posts & Telecom Press, 2006
Ryan Asleson, Nathaniel T. Schutta. AJAX 基础教程[M]. 金灵, 译. 北京:人民邮电出版社, 2006
- [4] W. RICHARD STEVENS, BILL FENNER, ANDREW M. RUD-OFF. UNIX Network Programming [M]. Yang Jizhang, translate. Beijing: Tsinghua university press 2006. 133-201
W. RICHARD STEVENS, BILL FENNER, ANDREW M. RUD-OFF. UNIX 网络编程[M]. 杨继张, 译. 北京:清华大学出版社, 2006. 133-201
- [5] David R. Butenhof. Programming With POSIX Threads[M]. Yu Lei, ZengGang, translate. Beijing. China electric power press, 2003. 37-74
David R. Butenhof. POSIX多线程程序设计[M]. 于磊, 曾刚, 译. 北京:中国电力出版社, 2003. 37-74