

# 嵌入式服务器在智能家居系统中的应用

陈亮, 高宏力, 周伦

(西南交通大学 机械工程学院智能机电研究所, 成都 610031)

**摘要:** 智能家居正逐步融入大众的生活, 改变着我们的生活方式; 嵌入式处理器的性能不断的提高, 使得智能家居系统可加入更多的功能, 进行更复杂的运算; 基于高性能的嵌入式处理器硬件平台, 设计了一种基于 Linux 软件系统的服务器作为智能家居系统的数据交换与监控中心; 并且采用 Nginx 服务器、PHP 编程语言和 SQLite 数据库, 实现了智能家居系统的 Web 远程监控平台, 此 Web 站点符合 HTML5 规范, 并且使用 HTML5 规范中的 WebSocket 协议, 实现了浏览器端和服务器端之间的全双工实时通信; 这是一种更为便捷的跨平台兼容的智能家居系统远程监控平台。

**关键词:** 智能家居; 嵌入式服务器; 远程控制; Web 实时通信; WebSocket

## Application of Embedded Server in Smart Home System

Chen Liang, Gao Hongli, Zhou Lun

(School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China)

**Abstract:** Smart home enters people's life gradually, and it changes our way of life. With the development of embedded processor performance, can more functions join in the smart home system, can perform more complex operations. A server was designed as the data exchanging and monitoring center of the smart home system, it based on Linux system and high-performance embedded processor hardware platform. And realized a remote monitoring platform on web by adopting Nginx, PHP and SQLite, the website conform to the HTML5 specification, and realized full-duplex real-time communication between browser and server by using WebSocket protocol in HTML5 specification. And this is a kind of more convenient and cross-platform compatible remote monitoring platform in smart home system.

**Keywords:** smart home; embedded server; remote control; Web real-time communication; WebSocket

## 0 前言

科技的发展改变了人们的生活方式, 智能家居已经渐渐的走近大众的生活, 使人们的生活变得更为舒适和便捷<sup>[1]</sup>。近几年 ARM 公司和 Intel 公司等处理器公司在嵌入式平台市场上的大力投入, 使得嵌入式处理器的性能有了很大提升, 功耗和价格进一步降低。这也使得嵌入式服务器的性能可以满足作为智能家居系统中数据交换与监控中心的需要。

采用 ARM cortex-a8 内核的处理器基于 Linux 系统实现了嵌入式 Web 服务器以及嵌入式数据库服务器在智能家居系统中的应用。平台实现了对传统家电的集中监控和远程监控, 以及 HTML5 和 CSS3 等技术实现的基于 Web 的远程监控平台, 实现了跨平台的兼容以及媲美客户端的交互性。

## 1 智能家居系统总体结构

智能家居系统主要包括控制核心、各种传感器与检测设备、把各部分连接起来进行协调工作的网关设备以及其他显示和娱乐设备等等。

其中使用嵌入式服务器作为智能家居系统的控制核心, 同时嵌入式服务器在本智能家居系统中作为数据交换中心。嵌入式服务器在作为控制核心担任控制相关设备进行工作的同时,

作为 Web 服务器使用, 搭建基于 Web 的监控平台, 实现对系统的监控, 这样可以实现监控平台在不同终端设备上的统一访问和兼容, 智能家居系统结构的如图 1 所示。用户可以通过电脑、手机、平板电脑等搭载 Web 浏览器的各类终端通过 Internet 或在家庭网络内访问智能家居监控系统。

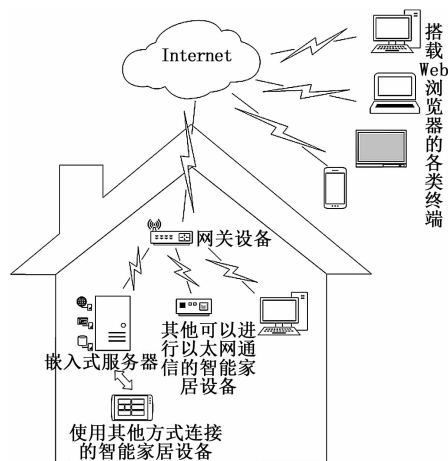


图 1 智能家居系统总体结构

## 2 嵌入式服务器硬件平台设计

本课题在 Linux 系统平台下搭建嵌入式服务器, 目前 Linux 系统支持的处理器众多, 以 ARM 和 Intel x86 构架的处理器为主, 选用基于 ARM cortex-A8 内核的全志 (Allwinner) A10 处理器搭建嵌入式服务器硬件平台, A10 在 ARM cortex-A8 内核的基础上加入了丰富的外围电路, 如 ADC、

收稿日期: 2014-10-09; 修回日期: 2015-03-19。

**作者简介:** 陈亮 (1990-), 男, 陕西人, 硕士研究生, 主要从事机械电子工程方向的研究。

高宏力 (1971-), 男, 河南人, 教授, 博士生导师, 主要从事复杂机电系统智能控制及可靠性设计、机电液一体化智能控制技术等方面的研究。

PWM、红外控制器、LCD 控制器等等, 可以简化硬件平台的设计, 并且 A10 处理器官方提供丰富的固件以及驱动源代码, 适合开发使用。配合相应的外部电路, 实现嵌入式服务器的硬件结构如图 2 所示。

除了必须的外部电源、RAM、Flash 等等之外, A10 处理器内部集成了以太网控制器, 在外部接上 RJ45 接口就可以使用以太网功能。以及在 A10 内部集成的红外控制器 (IR Controller), 在外部接上红外接收器, 既可以使用红外遥控器对服务器进行控制, 以及实现红外学习功能, 通过 GPIO 使用红外发射器对部分家电进行控制。除此之外使用 USB 接口、UART 接口也可使用部分一件设备实现智能家居系统的监测。如 USB 摄像头, 串口 GPRS 模块等。另外, 嵌入式服务器可以通过与其他以太网设备进行 Socket (套接口) 通信很方便的实现智能家居系统的控制和监测。

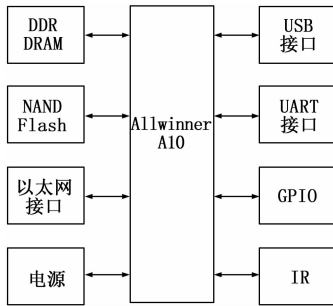


图 2 嵌入式服务器硬件结构

### 3 嵌入式服务器软件平台设计

#### 3.1 软件平台搭建

使用 Linux 系统作为嵌入式服务器的服务器系统, 选用轻量化的 Web 服务器、程序语言环境、数据库等搭建服务器软件环境。

A10 处理器官方提供 U-Boot, 针对 A10 芯片的配置工具 sunxi-tools.git, 以及系统内核源码等等, 可以很方便的进行系统移植。

Linux 环境下常用的 Web 服务器有 Apache、Tomcat、Nginx 等, 相对来说 Nginx 在这些服务器中最为轻量, 因此选用 Nginx 作为 Web 服务器, 动态页面语言选用 PHP, 而 Nginx 不支持对外部程序的直接调用或者解析, 所以 PHP 程序必须通过 FastCGI 接口来调用。Spawn-FCGI 与 PHP-FPM 是支持 PHP 的两个常用的 FastCGI 引擎。在新版的 PHP 中已经集成了 PHP-FPM, 安装变得更为方便。

数据库使用常用的 SQL 数据库, 选用轻型专为嵌入式而设计的 SQLite 数据库。

Linux 环境下软件安装及相关软件配置方法在此不作赘述。嵌入式服务器软件建平台结构如图 3。

#### 3.2 软件总体设计

软件使用 B/S (浏览器/服务器) 模式设计, 用户通过 Web 浏览器访问使用 PHP 编写的网站, 达到监测和控制智能家居系统的目的, 作为监测与控制系统, 智能家居系统要求在 B/S 之间实现实时性较高的通信, 可以通过 Socket 通信、WebSocket 通信以及 HTTP 轮询的方式实现。软件系统总体结构图如图 4 所示。

用户通过浏览器向服务器发出请求, 服务器根据请求进行

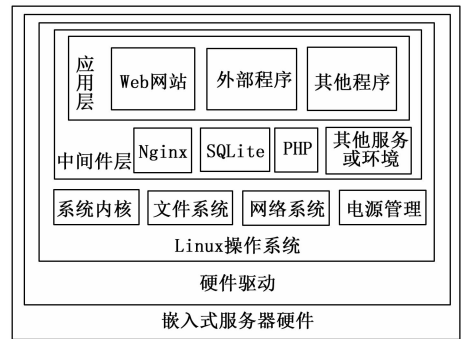


图 3 嵌入式服务器软件结构

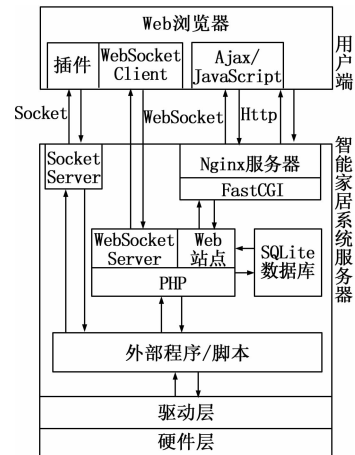


图 4 软件系统结构图

相应操作, 如操作数据库, 调用外部程序等等, 操作完毕后再通过 Socket、WebSocket 或者 HTTP 返回给浏览器相应数据。被调用的外部程序通过驱动层与相关设备进行通信, 实现智能家居监测和控制。

#### 3.3 数据库设计

根据智能家居系统中的功能需求, 设计数据库, E-R (实体-关系) 图如图 5 所示, 所有控制相关指令存储在数据库中, 方便程序调用。由于使用 B/S 模式搭建智能家居监控系统, 为达到较好的交互性与跨平台兼容的可视化效果, 部分界面元素数据储存在数据库中, 方便程序调用以动态绘制用户界面。另外, 除了图中所示的实体和关系以外, 还包含日志、任务等辅助功能的实体和关系未画出, 在这里主要体现出智能家居中的监测和控制功能。

#### 3.4 Web 站点设计

智能家居系统的监控站点使用 PHP 语言编写, 使用 HTML5 和 CSS3 标准, 不仅实现媲美客户端的可交互性和美观的操作界面, 更重要的是通过基于 Web 的实时通信和服务器端外部程序调用与硬件操作实现智能家居系统的管理和监测与控制的功能。

##### 3.4.1 基于 Web 的实时通信程序设计

为达到智能家居系统的实时性要求, 在浏览器和服务器端之间需要实现实时性较高的双向通信, 以及时显示智能家居系统的相关状态。目前可以通过 Socket 通信、WebSocket 通信以及 HTTP 轮询这 3 种主要方式实现<sup>[6-7]</sup>。

1) 基于 HTTP 的 Ajax/JavaScript 轮询: Web 应用程序的

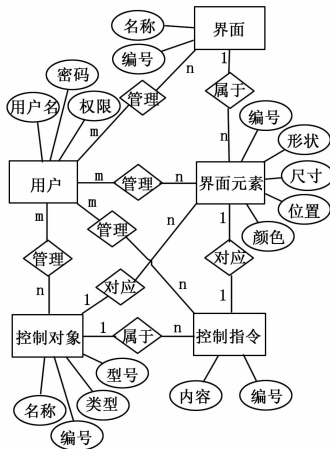


图 5 数据库 E-R 图

工作基于 HTTP 协议，它是一个无状态、单向的协议，由浏览器主动对服务器发出请求，然后服务器做出响应。而服务器并不能主动给浏览器发送信息，如当智能家居系统中有报警信号，服务器不能主动发送给浏览器端。这时可以使用浏览器端定时反复查询的方式来获得相关信息，使用 Ajax/JavaScript 在浏览器后台进行轮询，与服务器进行少量的数据交换，可以使网页实现异步更新，而不用刷新整个网页，以减少频繁访问对服务器造成很大压力，和改善浏览器端的用户体验。在浏览器后台可以使用浏览器端短时间定时轮询的方式，这样时间周期短对服务器照成的压力越大，对于智能家居系统中使用的嵌入式服务器来说，这种方式不可取；另外还可以使用长轮询方式，当浏览器发出请求时，服务器保持请求到数据更新或超时，服务器响应后请求结束，浏览器发出下一次请求，这种方式要求将请求一直保持，这样会影响客户端的数量，但在智能家居系统中，用户数量小，这个影响可以忽略。因此，若使用轮询方式选择长轮询方式较好。

(2) 基于浏览器插件的 Socket 通信：HTTP 并不能实现类似余 Socket 通信的全双工通信，但是借助于浏览器插件可以在 Web 页面上实现 Socket 通信。常用的方式有使用 Flash 插件的 Flash XMLSocket 和使用 JAVA 的 Java Applet 等，这种方式和普通 Socket 通信并无太大差别，但是会受客户端系统限制，如 Apple iOS、Android 等并不支持 Java Applet，插件运行受客户端系统权限限制等等。

(3) 基于 HTML5 的 WebSocket 通信：HTML5 WebSockets 规范定义了 WebSockets API，支持 Web 页面使用 WebSocket 协议与远程主机进行全双工的通信。它引入了 WebSocket 接口并且定义了一个全双工的通信通道，通过一个单一的套接字在 Web 上进行操作。HTML5 WebSockets 以最小的开销高效地提供了 Web 连接。相较于经常需要使用推送实时数据到客户端甚至通过维护两个 HTTP 连接来模拟全双工连接的旧的轮询或长轮询来说，这就极大的减少了不必要的网络流量与延迟。

要使用 HTML5 WebSockets 从一个 Web 客户端连接到一个远程端点，需要创建一个新的 WebSocket 实例并为之提供一个 URL 来表示想要连接到的远程端点。该规范定义了 ws:// 以及 wss:// 模式来分别表示 WebSocket 和安全 WebSocket 连接。一个 WebSocket 连接是在客户端与服务器之间

HTTP 协议的初始握手阶段将其升级到 WebSocket 协议来建立的，其底层仍是 TCP/IP 连接。

目前 PC 平台和移动平台的主流 Web 浏览器都支持 HTML5 标准以及 WebSocket，所以这里主要选择 WebSocket 协议来实现服务器与浏览器之间的实时通信，但为了兼容更多的平台，可以同时使用三种方式，如图 4 所示。在 WebSocket 连接建立以前，客户端和服务器之间有一个如图 6 所示的握手过程。

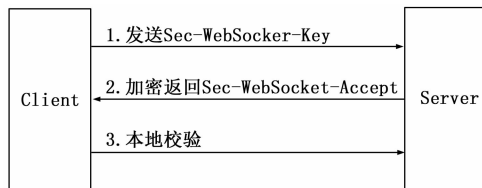


图 6 客户端主界面

服务器需要对握手协议的内容进行解析并返回给客户端相应内容，服务器端使用 PHP 语言实现，部分代码示例如下。

```
function dohandshake(user,buffer){
    list (resource, host, origin, strkey1, strkey2, data) = getheaders
    (buffer);
    .....
    upgrade="HTTP/1.1 101 WebSocket Protocol Handshake/r/n" .
    "Upgrade: WebSocket/r/n" .
    "Connection: Upgrade/r/n" .
    "Sec-WebSocket-Origin: " . origin . "/r/n" .
    "Sec-WebSocket-Location: ws://" . host . resource . "/r/n" .
    "/r/n" .
    hash_data;
    socket_write(user->socket, upgrade, chr(0), strlen(upgrade, chr
    (0)));
    user->handshake=true;
    return true;
}
```

浏览器端 WebSocket 的握手自动完成，客户端代码较简单，主要功能实现代码如下，其中 MozWebSocket 为 Mozilla Firefox 浏览器中的 WebSocket 类。

```
<script type="text/javascript">
var ws;
.....
function init()
{
    Try{
        if ("WebSocket" in window) {
            ws = new WebSocket("ws://" + .....);
        }
        else if("MozWebSocket" in window) {
            ws = new MozWebSocket("ws://" + .....);
        }
    }
    catch (ex) {.....}
    ws.onopen = functinon() {.....};
    ws.onmessage = functinon(e) {.....};
    ws.onclose = functinon() {.....};
    ws.onerror = functinon() {.....};
    .....
```

```

}
.....
function wsSend(ms) {
ws.send(ms);
.....
};
.....
function wsClose() {
ws.close();
.....
};
.....
</script>

```

浏览器端直接使用 `ws.send(ms)`; 就可以直接使用 WebSocket 发送数据。`ws.onmessage = function(e)` {.....}; 为数据接收的事件函数。

#### 3.4.2 服务器端外部程序调用

在智能家居系统中, 服务器承担主要工作, 对系统中各个部分进行监控, 而 PHP 功能有限, 并不能完成所有操作, 所以需要其他语言或脚本来实现服务器端的具体操作, 如监控服务器端 GPIO 等等, 可以使用 C、Python 等编写程序, 或者直接使用 Shell 命令或脚本程序进行操作。可以通过 PHP 调用这些外部程序或脚本来实现相应工能。在 PHP 中, 可以使用 `exec`、`system`、`passthru`、`popen`、`proc_open` 等函数来实现此功能。其中 `exec` 的使用方法如下:

```
string exec ( string command [, array &output [, int &return_var]] )
```

其中返回值保存最后的输出结果, 而所有输出结果将会保存到 `output` 数组, `return_var` 用来保存命令执行的状态码 (用来检测成功或失败)。例如通过 shell 命令来使一个 GPIO 引脚状态称为高电平:

```
Exec ( "echo 1 > /sys/class/gpio/gpio17_pg9/Value");
```

其他函数使用方法类似, 需要注意的是这些命令的使用需要对 PHP 进行安全相关的设置。

#### 3.4.3 用户界面

用户界面使用 HTML5、CSS3 以及 JavaScript 等编写, 尽量少的使用图片, 以提高页面的加载速度。截稿前并未对页面进行电脑浏览器的适配, 使用 Android 手机浏览器访问的 Web 页面的效果如图 7。



图 7 手机 Web 端访问效果

除此之外, 客户端还包括用户身份验证, 记录管理等功能。后续将会进一步完善和开发。

## 4 系统测试结果与分析

系统研制完成后, 对嵌入式服务器性能, Web 端响应等方面做了测试, 在正常访问使用的情况下服务 CPU 空闲百分比保持在 90%~95%左右, 内存占用在 120 MB 左右, 这得益于 linux 系统占用资源少和所选硬件平台性能较高, 使得系统还有更大的扩展空间。在局域网内使用局域网 IP 地址测试 Web 站点, 页面获取时间在 100 ms 以内, 这主要取决于页面内容的多少和服务器的处理速度。在实时控制环节 WebSocket 连接建立需要 30 ms 左右, 连接建立完成后每次单向通信时间在 5 ms 以内, 由于服务器只在系统状态有变化的时候向 Web 端发送数据, 所以网络吞吐量很小, 由于规定的控制指令很短, 最少只有 5 个字符, 所以在使用 Web 端进行频繁操作控制时, 网络吞吐量平均在 30 Byte/s。Web 端在接收到服务器数据后界面会发生相应改变, 在测试过程中, 使用灯的开关控制功能, Web 界面变化与实际控制结果几乎感觉不到延迟。从测试结果看来, 本系统达到了实时控制的预期目标, 并且实现了良好的交互性, 另外 Web 端也实现了跨平台的兼容, 使智能家居控制更为便捷。

## 5 结束语

使用 Linux+ Nginx+ SQLite+ PHP 的软件方案在 ARM cortex-a8 (Allwinner A10) 的硬件平台上构建了嵌入式服务器, 使用最新的 HTML5 和 CSS3 标准构建智能家居系统远程监控平台, 实现了基于 WebSocket 的实时通信, 和友好的人机交互界面。用户可以通过手机、电脑等任意平台上的浏览器访问系统界面实现对系统的操作。信息技术发展日新月异, 智能家居系统正渐渐融入大众的日常生活, 使我们的生活变得更为舒适、安全、便捷。今后将会对智能家居系统进行更深入的研究。

#### 参考文献:

- [1] 郭稳涛, 何怡刚. 智能家居远程监控系统的设计与实现 [J]. 计算机测量与控制, 2011, 19 (9): 2109-2112.
- [2] 邵鹏飞, 张宝儒, 王 喆. 面向移动互联网的智能家居系统研究 [J]. 计算机测量与控制, 2012, 20 (2): 474-476, 479.
- [3] 孙宇虹, 崔少彬. 家庭网络中嵌入式服务器的研究 [J]. 计算机工程与设计, 2008, 29 (6): 1394-1396.
- [4] 明 航. 智能家居系统中嵌入式系统的应用 [D]. 武汉: 湖北大学, 2013.
- [5] 马培粤. 智能家居 web 远程浏览控制软件的设计与实现 [D]. 武汉: 华中科技大学, 2006.
- [6] 李代立, 陈 榕. WebSocket 在 Web 实时通信领域的研究 [J]. 电脑知识与技术, 2010, 28 (6): 7923-7925.
- [7] 柳 纲, 李佳阳, 张 毅. WebSocket 技术在 EMS 信息实时发布中的应用 [J]. 电力信息与通信技术, 2013, 11 (10): 30-33.
- [8] 孙奎全, 陈向东, 张传武. 基于嵌入式服务器的实验室安防系统设计 [J]. 通信技术, 2013, 46 (7): 60-62.