

The Principle and Realization of WeChat Remote Control

Yunxi Zhang^{1,2}, Pengzhan Li¹, Jia Liu^{1,2}

¹School of Automation and Electrical Engineering, Tianjin University of Technology and Education, Tianjin

²Tianjin Key Laboratory of Information Sensing and Intelligent Control, Tianjin University of Technology and Education, Tianjin

Email: liujia0704@126.com

Received: Jun. 3rd, 2018; accepted: Jun. 20th, 2018; published: Jun. 27th, 2018

Abstract

At present, the domestic smart home system usually requires specific software to control, the actual use is more complex, and the user learning cost is high. This paper mainly studies the principle and implementation method of remote control smart home via WeChat and network, and achieves the control of smart home easily and conveniently.

Keywords

WeChat, Remote Control, Smart Home, Server Response, Database Management

微信远程控制的原理与实现

张运喜^{1,2}, 李鹏展¹, 刘佳^{1,2}

¹天津职业技术师范大学自动化与电气工程学院, 天津

²天津职业技术师范大学天津市信息传感与智能控制重点实验室, 天津

Email: liujia0704@126.com

收稿日期: 2018年6月3日; 录用日期: 2018年6月20日; 发布日期: 2018年6月27日

摘要

目前国内的智能家居系统通常需要特定的软件进行控制¹, 实际使用比较复杂, 用户学习成本高。本文主要研究通过微信以及网络来实现远程控制智能家居的原理和实现方法, 做到随时随地, 简单便捷的完成对智能家居的控制。

关键词

微信, 远程控制, 智能家居, 服务器应答, 数据库管理

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

智能家居是近两年来比较热门的研究方向[1] [2], 微信是我们日常生活中经常用到的通信软件, 将智能家居和微信结合起来, 可以实现智能家居控制的即时化和简易化, 使得大多数智能家居项目不在仅仅停留在实验室阶段, 通过微信来拉近普通民众和智能家居的距离。

2. 系统的原理与构架

系统可以分为三个部分: 微信端、服务器端以及控制端, 微信端即我们日常使用的微信软件, 通过所申请的公众测试号向服务器发送命令, 服务器端本文以新浪云为例做介绍, 负责微信命令的处理以及数据储存工作, 控制端采取 ESP8266 模块加 ARDUINO MEGA2560 开发板实现对智能家居的控制。

三个部分之间的工作关系如图 1 所示, 当三部分部署完成后, 微信端向服务器发送命令, 例如: “开灯”、“关灯”等等, 服务器解析命令后会打开数据库, 根据命令要求查询或者更改数据库数据, 将结果返回给微信端; 同样的, 控制端也会不断的向服务器发送命令, 查询或者更改数据库中的参数, 同时根据查询到参数对一些设备进行控制。

总的来说, 服务器是一个数据中转站, 同时也带有数据储存功能, 使得微信端和控制端能够通过服务器进行数据交换, 实现远距离实时控制的效果。

3. 准备工作

在微信端我们需要有一个微信测试号[3], 可通过微信微信公众平台进行申请, 申请到测试号之后我们进入测试号的管理界面, 如图 2 所示, 可以看到我们所申请的测试号的 appID、appsecret 以及接口配置信息设置窗口。

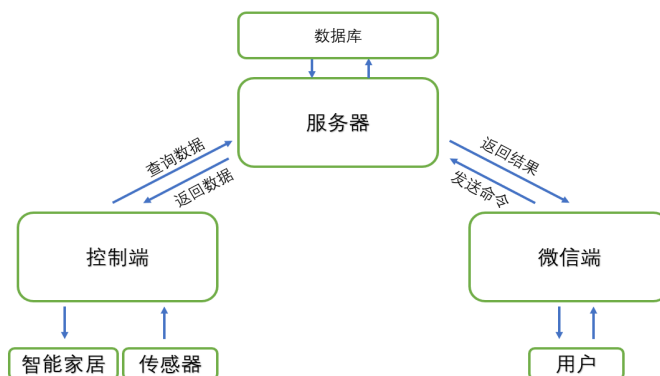


Figure 1. System structure
图 1. 系统构架

图中的接口已经为配置成功的状态，新用户可以根据要求填写 URL 和 Token 进行验证，具体方法可以参照微信公众平台→服务号→开发文档→开始开发中的接入指南进行操作。同时，设置中我们也可以看到测试号二维码以及用户列表，用户可以用手机微信扫描二维码进入测试号，测试号所申请完毕后需要详细阅读微信公众平台的技术文档，如图 3 所示，其中详细介绍了微信测试号的使用规范以及信息交流规范，这些是我们在之后的服务器端进行开发时所应该遵守的，否则服务器将无法和微信测试号进行通信。

服务器端需要申请到一台稳定的服务器以及 MySQL 数据库，本文将以新浪云服务器以及其 MySQL 数据库为例进行讲解，申请步骤以及应用和数据库的创建细节不再过多赘述。

控制端本文以 Arduino mege2560 开发板和 ESP8266 模块为例，控制设备为一颗绿色 LED，传感器使用的是光敏电阻，可以将光照数据返回给数据库。硬件相对比较简单，连接完成之后需要进行检查和短路测试，确定没有问题之后可以上电。



Figure 2. The management interface of test number

图 2. 测试号管理界面



Figure 3. The technical documentation screen of WeChat public platform

图 3. 微信公众平台的技术文档画面

4. 测试与数据交流

完成准备工作后将相应的代码分别部署到服务器端与控制端，测试无误后三部分便可以相互传递数据了。

微信端和服务器端都部署完成之后，便可以开始部署控制端了，控制端主要由单片机开发板和 WiFi 模块以及一些传感器组成，负责将本地传感器的数据上传至数据库，并且查询数据库中的开关数据等，然后做出相应的动作。

本文以 Arduino mege2560 开发板和 ESP8266WiFi 模块为例，控制设备为一颗绿色 LED，传感器使用的是光敏电阻，可以将光照数据返回给数据库。实物如图 4 所示，硬件相对比较简单，连接完成之后需要进行检查和短路测试，确定没有问题之后可以上电。

打开串口监视器查看控制端的返回数据，如图 5 所示，为一个周期的数据记录：

“ON LINE”表示 WiFi 已经连接。

“loops7”表示 loop 函数的第七次循环。

“22813”表示此时的光敏电阻阻值。

“CONNECT done at 14”表示已经建立 TCP 连接。

“GET /httpget.php?light=22813 HTTP/1.1

Host:1.blink.applinzi.com

Cache-Control:no-cache”是发送给服务器的协议内容。

“SENDHTTP done at 58”表示已经完成 HTTP 协议的发送[4]。

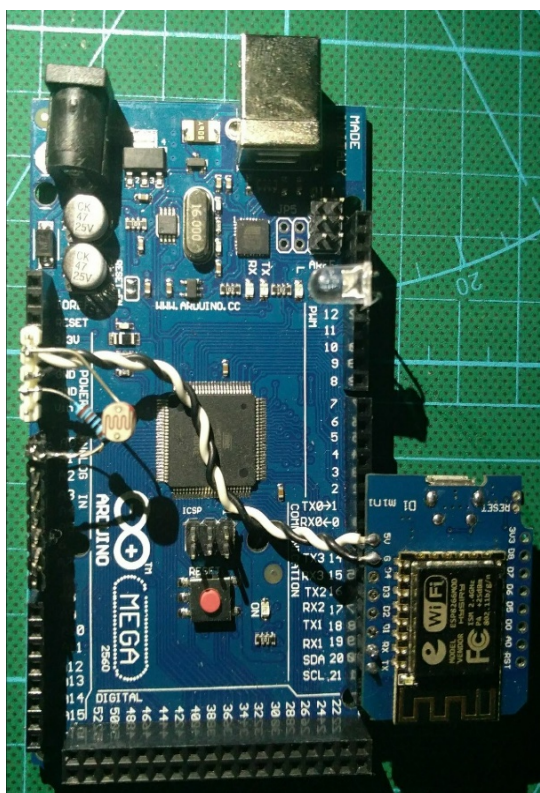


Figure 4. The physical drawing of control terminal
图 4. 控制端实物图

```
ON LINE
loops7
22813
CONNECT done at 14
GET /httpget.php?light=22813 HTTP/1.1
Host:1.blink.applinzi.com
Cache-Control:no-cache

SENDHTTP done at 58

+IPD, 211:HTTP/1.1 200 OK
Server: sae
Date: Fri, 16 Mar 2018 15:07:54 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 45
Connection: keep-alive
Via: 144157

blinkdatapagex135
update
timestamp:152****874
```

Figure 5. Communication procedure

图 5. 通信过程程序

其余数据为服务器的返回数据，可以看到返回数据中有开关数据以及时间戳，当单片机接收到数据时，将会分析其中的“blinkdatapagex135”字符串，如果结尾是“35”，则打开 LED，如果结尾是“34”，则关闭 LED。此时，我们就可以通过微信发送“打开”、“关闭”命令控制 LED 的点亮与熄灭，以及“光照”命令获取光敏电阻的阻值了。

5. 系统特性

完成三部分的部署后，微信端和控制端就可以通过远程服务器进行信息交流了[5]，给予互联网的强大功能，实现使用微信进行远程控制的目的。

5.1. 系统物理构成及介绍

系统中微信端以及服务器端的设备维护分别由腾讯、新浪公司负责，我们不必担心；在接收端采用的是 ESP8266 模块加 ARDUINO MEGA2560 开发板的组合，在正确的接线情况下同样值得信赖。

ESP8266 模块：一块专为移动设备、可穿戴电子产品和物联应用设计的 WIFI 模块，功耗小，稳定性高，可通过串口与其他模块通信。

ARDUINO MEGA2560 开发板：一种 AVR 单片机的开发板，此开发板共有 54 路数字接口，其中有 16 路模拟输入接口，4 路 uart 接口，14 路 PWM 输出接口，具有强大的信息处理、IO 输入、输出能力，能够连接较多的外围设备。

5.2. 系统延迟

1) 微信端与服务器端的延迟：本文以新浪云服务器为例，当关闭微信签名验证时，服务器与微信端

的信息交换十分迅速,但是为了安全起见,建议打开微信签名验证,此时发送命令时有较小的几率出现“命令无应答”的情况,需要重复发送命令,一般两到三次即可。

2) 控制端与服务器端的延迟:系统控制端大约每秒与服务器进行一次通信,在单片机程序中表现为每执行一次通讯语句后存在着一秒延时,即语句“`delay(1000);`”,(此语句的延时时间可以更改,具体延时大小需要实际需求以及服务器要求确定)也就是说,理论上 LED 状态的更新延迟最大为一秒左右,实际的使用效果与网络速度以及微信端命令的发送时间有关,与理论值相差不大。

5.3. 系统可扩展性与兼容性

本系统控制端的 ESP8266 模块与 ARDUINO MEGA2560 开发板之间使用串口通信,可以兼容大多数具有串口功能的单片机以及其他模块。

本文中以 ARDUINO MEGA2560 开发板为例,能够控制非常多的外围设备(参考前文对此模块的介绍),同时也可通过串口通信无限扩展此类模块,使得系统的扩展性极强。

5.4. 系统可靠性

一般来说服务器端与微信端都有较强的可靠性,不必过多担心,而本系统的控制端仅需要一个稳定的 WiFi 热点以及 5v 电源就可以与服务器通信,条件少,效果好,同时单片机中的程序中还加入了在无法连接网络时的处理对策,能够保证控制端在无网络连接时对外围控制设备进行保护性处理,所以系统在正常使用时具有很高的可靠性。

6. 总结

本文简单介绍了通过微信进行远程控制系统中的信息交流原理以及实现方法,以一些简单的硬件元件为例介绍了系统的具体设计方法。

基金项目

国家自然科学基金资助项目(61703307);天津市应用基础与前沿技术研究计划青年项目(15JCQNJC04200);天津市高等学校科技发展基金资助项目(JWK1614);天津市大学生创新创业训练项目(201710066159);天津职业技术师范大学科研启动基金资助项目(KYQD14032)。

参考文献

- [1] 徐锋,刘欣,方加宝.智能家居远程控制系统设计[J].低压电器,2009(4):21-24.
- [2] 魏振春,韩江洪,张建军,张利.智能家居远程控制系统的设计[J].合肥工业大学学报(自然科学版),2005(7):751-754.
- [3] 周桥梁.微信智能家居控制系统设计[J].数字技术与应用,2017(5):158.
- [4] 陈诚斌,苏凯雄,赵阳,刘梦,邱瑾.远程智能家居控制系统设计[J].信息技术与网络安全,2017(14):89-91,94.
- [5] 王富明,柴政.智能家居远程控制模拟系统设计[J].电脑知识与技术,2009(3):698-699.

知网检索的两种方式：

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择：[ISSN]，输入期刊 ISSN：2161-8801，即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：<http://www.hanspub.org/Submission.aspx>

期刊邮箱：csa@hanspub.org