



## 机械电子工程教育部“卓越工程师” 企业实习报告

题    目： 智能网关设备

学    院： 机械与自动控制学院

班    级： 14 机电二班

学    号： 2014330300129

学生姓名： 郑江湖

机械与自动控制学院

2018 年 05 月 17 日

## 摘 要

根据智能家居中主机与其他受控设备的通信要求,本文提出了一种基于 WiFi 和 Zigbee 之间的通信协议转换网关设备。同时给出了软硬件设计的方法,设计过程包括硬件电路设计和程序设计两大步骤,对于嵌入式开发中可能遇到的重要技术问题都有一定的涉猎。本文通过对 WiFi 和 Zigbee 两种不同的通信协议的解决方案的学习,成功将两种芯片集成在同一个设备上,两者之间采用串口通信,和家庭主机或 APP 之间采用 WiFi 的 JsonRpc 通信,和其他传感设备之间采用 Zigbee 通信。在设计中先用 Cadence 绘制出了原理图及 pcb 图,并进行了调试板打样,然后利用 Eclipse 进行了 c 语言的编写,成功调试出了设计要求的程序,并用实物将其表现了出来。

**关键词:** 嵌入式; 网关; c 语言

# 目 录

## 摘 要

第 1 章 企业实习的目标及系统设计提出.....	1
1.1 企业实习的目标及意义 .....	1
1.2 系统设计的背景意义 .....	1
第 2 章 网关的设计方案及硬件电路.....	2
2.1 系统设计方案.....	2
2.2 电源部分 .....	4
2.3 WiFi 部分.....	5
2.4 Zigbee 部分 .....	6
2.5 基本外设部分.....	7
第 3 章 系统软件的设计.....	10
3.1 软件框架 .....	10
3.2 WiFi 程序设计 .....	11
3.3 Zigbee 程序设计 .....	12
第 4 章 调试.....	15
4.1 组建网络调试.....	15
4.2 删除网络调试.....	16
第 5 章 企业实习总结与展望.....	18
参考文献 .....	19
附录 1、教育部“卓越计划”企业实习成绩表 .....	20

## 第 1 章 企业实习的目标及系统设计提出

### 1.1 企业实习的目标及意义

在大学期间，虽然已经进行了大量电子专业的理论知识学习，但是如果不将其进行实践应用，则永远不会有任何实际价值，因此在毕业之前在相关行业的企业进行一段时间的实习会显得很有必要，这可以帮助我们在将理论知识转换为实际动手能力的同时，也能帮助我们快速适应即将到来的社会生活。而江苏英索纳智能科技有限公司，是一家以研发智能家居产品为主的科技型公司，在该公司研发中心实习期间，使自己的专业技能与知识得到了质的提升，同时也对嵌入式行业有了一定的认识，并且借助实习单位的现有资源与研发情况，完成了以下的课题研究。

### 1.2 系统设计的背景意义

智能家居行业属于一个新兴行业，它借助一系列高新技术与产品，将家庭生活进行智慧化与简易化，而目前大多的智能家居系统中，一般都是以 APP 或主机作为控制中心，以 Zigbee 设备作为控制终端。这就出现了矛盾之处，控制中心基于 WiFi，而控制终端则基于 Zigbee，二者是两种完全不同的通信协议。因此如果想要实现系统的整合和可移植性的塑造，就不得不存在一个协议转换器，通过该协议转换器，实现 WiFi 和 Zigbee 之间的桥梁沟通，从而完成从控制中心到控制终端的命令执行与数据交换。

## 第2章 网关的设计方案及硬件电路

### 2.1 系统设计方案

ZAP（Zigbee 外扩 AP 设备）作为本系统中极其重要的设备，致力于上层主机或用户与下层端点传感器设备之间的连接，用于实现从 WiFi 到 Zigbee 的网络协议转换功能。框架示意图如下：

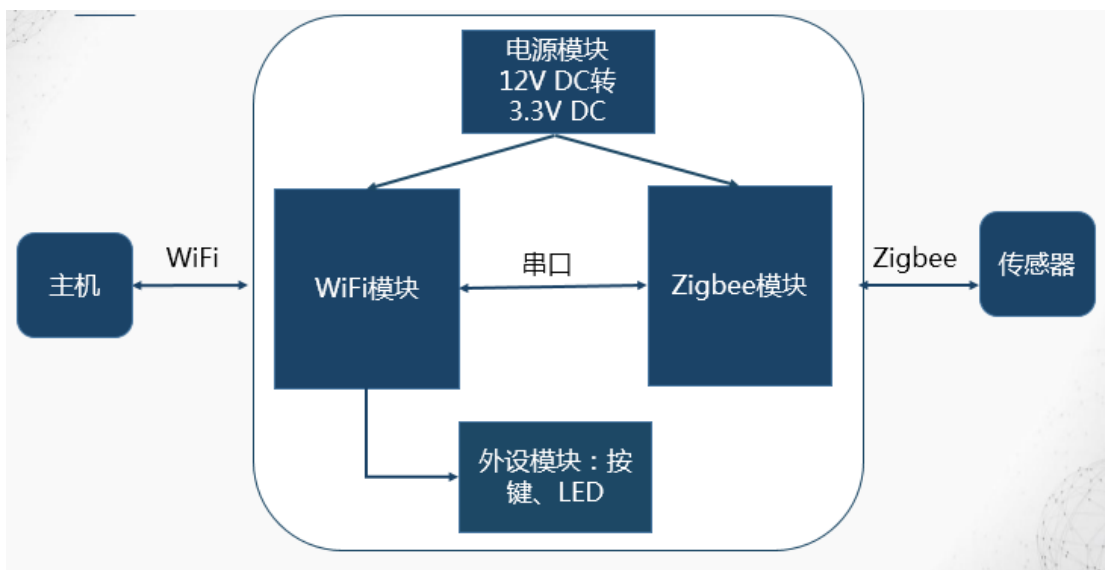


图 2-1 系统方案

因此该 ZAP 网关需要具备以下功能：

- 1：可以无线加入局域网并完成和主机之间的 WiFi 通信；
- 2：具备 Zigbee 组网/删除网络功能；
- 3：设备内部的 WiFi 和 Zigbee 之间可以互相通信；
- 4：低功耗、信号稳定；

鉴于以上要求，决定采用 WiFi 模组+Zigbee 模组的方案来搭建 ZAP 的功能框架，硬件芯片选型如下：

**Zigbee 模块：**Zigbee 领域应用最广的以 TI 的 CC25630、Silicon labs（芯科）的 EM357 为主。其中 CC2530 芯片协议简单，成本低，但是可开发空间不大，不能集成太多设备。而 EM357 虽然相比较于 CC2530 贵了一点，但是弥补了

CC2530 的缺点，信号的传输速率快、信号稳定。因此最终决定采用 EM357 芯片作为 Zigbee 解决方案。

EM35x 系列芯片具备高性能的 ZigBee/802.15.4 解决方案，是一款系统级芯片 (SoC)，其内核为 ARM Cortex-M3，内置 2.4GHz IEEE 802.15.4 无线收发器、32 位微处理器、闪存 Flash 和 RAM。主要应用于智能家居、智能能源、工业自动化等领域。

EM357 特征可概括如下：

- 32 位的 ARM Cortex-M3 微处理器
- 2.4GHz 的 IEEE802.15.4 收发器
- 128KB 的 Falsh
- 12KB RAM
- 灵活的外设和串口支持
- 24 高度可配置的 GPIO 与施密特触发器输入
- 高频 RC 晶振
- 串行线/JTAG 接口
- **WiFi 模块：**采用深圳安信可科技有限公司推出的 WiFi 模组 ESP8266 系列中的 ESP-12S 芯片。ESP8266EX 拥有高性能的无线 SOC 特性，成本低，功能强大，位嵌入式 WiFi 应用量身定制。
- ESP-12SWiFi 的核心处理器 ESP8266 采用了低功耗的 32 位微处理器 Tensilica L106，其具有 16 位精简模式，支持 80MHz 和 160MHz 主频,支持 RTOS，集成 WiFi 等特征，同时具有功放 MAC/BB/RF/PA/LA=NA，贴片式 S 型天线增强了信号强度。支持标准的 IEEE802.11 b/g/h 协议与完整的 TCP/IP 协议栈。
- ESP8266EX 具备完整的 WiFi 处理能力，既可单独使用，也可以搭配其他 MCU 使用。当作为主处理器时，可直接通过外接 Flash 来启动。ESP8266EX 具备一定的处理与存储能力，提供一定数量的 GPIO 口来操控其他的比如 LED、按键等外设。

- ESP8266EX 模组特点可概括如下：
  - 最小的 802.11b/g/n Wi-Fi SOC 模块
  - 采用低功耗为 CPU，可兼做应用处理器
  - 主频最高可达 160MHz
  - 支持 UART/GPIO/IIC/PWM/ADC 等接口
  - 采用 SMD-16 封装，方便焊接与测试
  - 集成 Wi-Fi MAC/BB/RF/PA/LAN
  - 支持多种休眠模式，深度睡眠电流可达 20uA
  - 内嵌 Lwi 协议栈
  - 支持 STA/AP/STA+AP 工作模式
  - 通用 AT 指令可快速上手

## 2.2 电源部分

电源模块采用 12V DC 电源端子接插件的方式接入电路板。由于电路中既需要 3.3V 的直流电源，又不能出现太大的纹波，因此选择开关稳压电源（DC/DC）配合线性电源（LDO）的解决方案。其中开关稳压电源允许输入输出电压有较大差值，并且由于是通过开关电路输出占空比或频率可调输出电压，因此其转换效率高，但是纹波略大，而线性电源则虽然输出输出压差小，但是最终输出的 3.3V 电压纹波很小，可以用来给 ESP8266 和 EM357 供电，原理图如下：

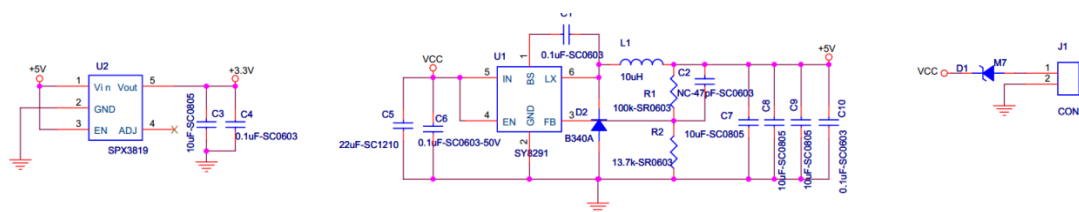


图 2-2 电源

D1：利用防反接二极管对整个电路的电源进行防反接保护。

SY8291：DC/DC 电源芯片，输入 12V，输出 5V，其中

C5、C6：对输入的电源进行滤波；

L1、C2、C7、C8、C9、C10：对输出电源进行滤波；

R1/R2：采用大电阻来最小化二极管负载的功耗；

D2：肖特基二极管，具备极低的正向导通电压和快速的转换频率

开关电源次级线圈上的输出由于是高频开关，因此是脉冲状态，而最终需要的则是可供电路使用的直流电，因此需要利用整流二极管和滤波电容来对其进行平滑处理。

SPX3819：LDO 电源，压降为 1.7V；

C3、C4：输出电源的滤波电容，除去纹波与噪声；

## 2.3 WiFi 部分

采用 ESP-12S 配备其他外设的方案来完成系统软件要求，具备 WiFi 重启按键、两路三色 LED 功能显示灯控制、一个功能按键、TXD/RXD 程序烧写/Zigbee 通信、GPIO2 串口调试、nRST 引脚负责 Zigbee 重启等功能，采用接插件的形式，将相关引脚集成并引出，方便前期开发进行调试：

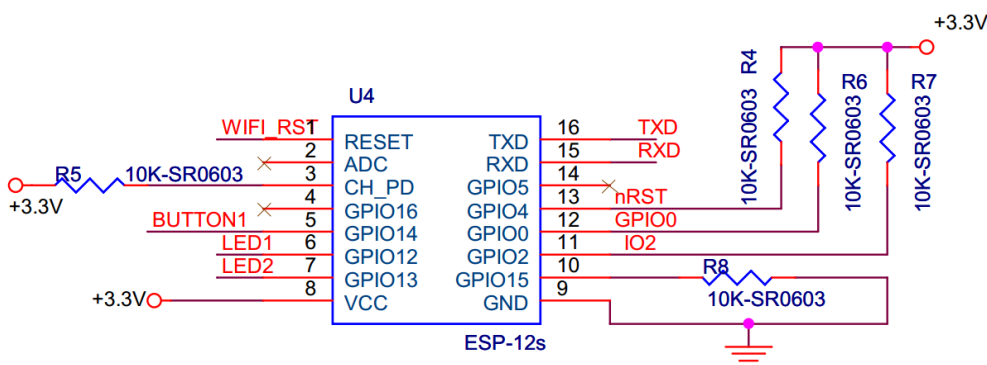


图 2- 3 ESP12S

GPIO4：EM357 的复位引脚，平时被上拉电阻钳在高电平，在和 EM357 串口通信中需要被拉低；





JTMS/JTCK: JLink 固件烧写引脚;

RXD/TXD: 和外界其他 MCU 模块串口通信引脚;

+3.3V: 电源电源输入;

RF\_N/RF\_P/RF\_TX\_ALT\_P/ RF\_TX\_ALT\_N: 功放芯片的输出以及天线信号来源;

PA3/PA5/PC6/: 功放模式配置引脚;

nRST: 芯片重启引脚;

MOSI/MISO/CLK: SPI 外接 Flash 引脚;

X1/X2: 外接晶振;

## 2.5 基本外设部分

LED:

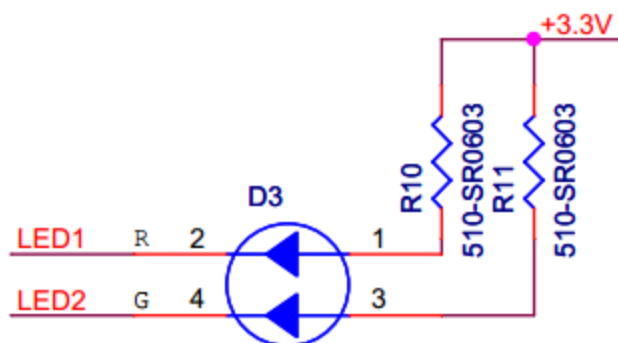


图 2- 5 LED

按键:

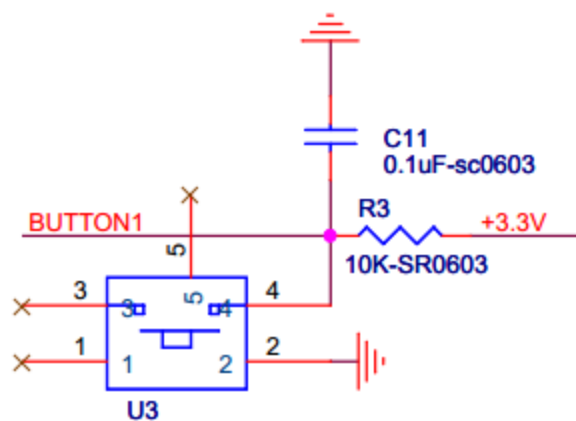


图 2- 6 按键

UART:

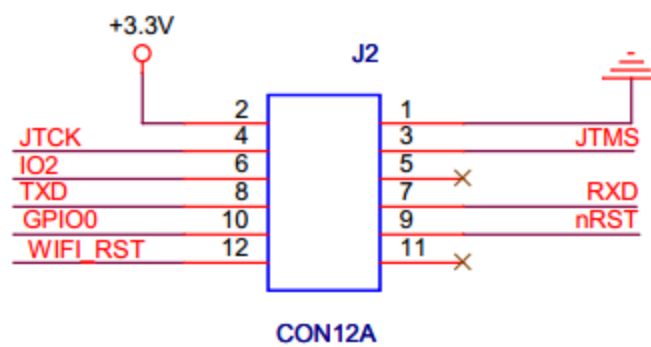


图 2- 7 UART



## 第 3 章 系统软件的设计

### 3.1 软件框架

ZAP 的软件功能需求如下：

- 支持 TCP/IP 局域网无线通信
- 和主机之间进行 WiFi 报文接收，能执行来自主机的功能指令，并将指令执行结构进行反馈
- 可以进行 Zigbee 组网、允许设备入网、删除子节点设备
- 可以进行正常的休眠模式

综上，本文采用商家提供的开发环境与开发包进行软件设计，而 EM357 则直接采用固有的标准化 Zigbee 固件，另外将二者之间进行串口通信使用的 EZSP 协议文件“ncp”进行修改并作为库文件静态链接至 ESP8266 的软件工程，这样 ESP8266 便可以调用 EM357 中的 Zigbee 相关功能。

ZAP 软件功能分层图示如下：

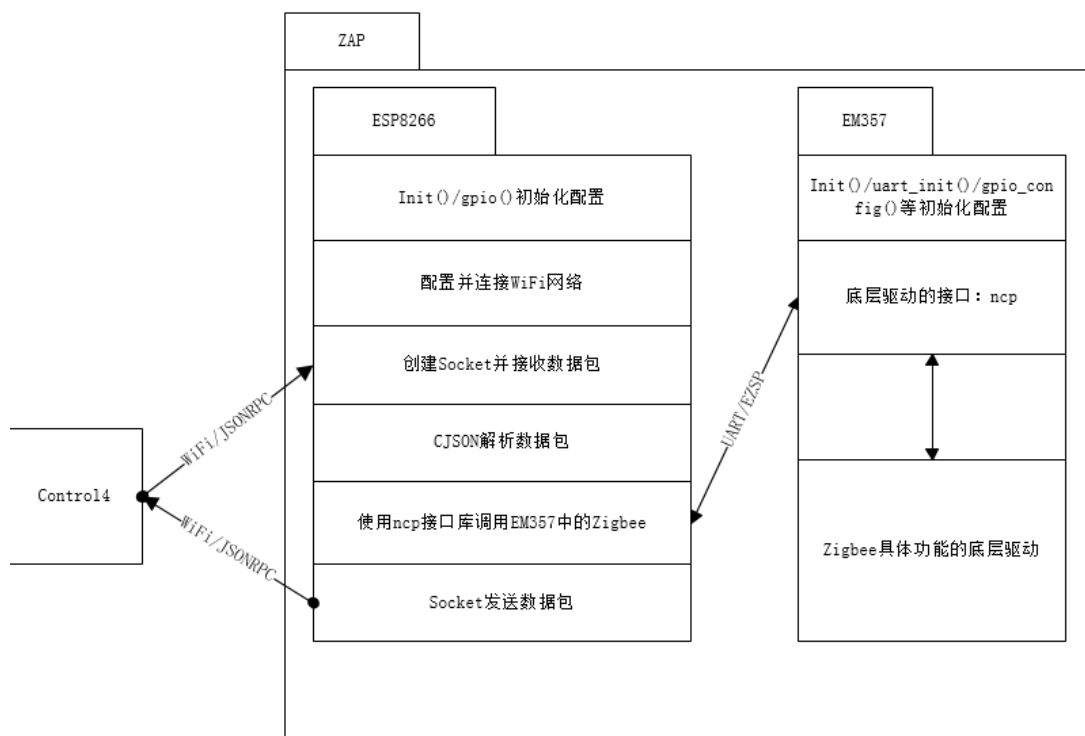


图 3- 1 软件框架

### 3.2 WiFi 程序设计

借助 ESP8266 官方的 SDK 开发包 ESP8266\_RTOS\_SDK-1.5 来完成 ZAP 的无线入网与相关 WiFi 工作模式配置。开发包内容如下：



app	2018\5\5 星期六 ...	文件夹	
bin	2018\5\5 星期六 ...	文件夹	
documents	2018\5\5 星期六 ...	文件夹	
driver_lib	2018\5\5 星期六 ...	文件夹	
examples	2018\5\5 星期六 ...	文件夹	
extra_include	2018\5\5 星期六 ...	文件夹	
include	2018\5\5 星期六 ...	文件夹	
ld	2018\5\5 星期六 ...	文件夹	
lib	2018\5\5 星期六 ...	文件夹	
third_party	2018\5\5 星期六 ...	文件夹	
tools	2018\5\5 星期六 ...	文件夹	
.cproject	2018\5\5 星期六 ...	CPROJECT 文件	12 KB
.project	2018\5\5 星期六 ...	PROJECT 文件	3 KB
LICENSE	2018\5\2 星期三 ...	文件	4 KB
Makefile	2018\5\2 星期三 ...	文件	11 KB
README.md	2018\5\2 星期三 ...	MD 文件	2 KB
VERSION	2018\5\2 星期三 ...	文件	1 KB

图 3- 2 布局

app: 用户工程存放文件夹;

bin: 存放最终生成的 bin 格式可执行文件;

documents: 工程说明文档;

driver\_lib: 外设驱动的库文件, 例如 SPI、IIC、GPIO 等;

examples: 实例工程;

include: 存放工程所需子模块的头文件;

ld: 连接时所需的脚本文件;

lib: 工程所需的库文件;

tools: 工程所需编译工具;

按照提供的标准工程布局, 可将所需文件与程序分别放置在正确的文件夹, 然后在主函数文件中进行代码添加与开发, 最后在 Linux 下通过修改 makefile 文件来添加需要链接的库文件, 运用 cmake 指令生成最终的可执行 bin 文件。

### 3.3 Zigbee 程序设计

致。

### ncp 接口层:

ESP8266 和 EM357 之间采用基于 ncp 静态库的 EZSP 串口协议进行通信。ncp 静态库作为 Zigbee 协议标准的上层接口应用，其底层代码关闭，上层接口开放，随 EM357 芯片的解决方案一同发行，它集成了 ESP8266 期望 EM357 所要实现的功能函数接口以及部分用户根据开发手册自定义编写的代码，比如设置 EM357 芯片的功放模式、打印 EM357 版本号、其他初始化处理机制等，因此只有静态库 ncp 被编译进 ZAP 的工程中之后，当收到来自主机的具体功能函数调用时，ESP8266 便可以通过查询该 ncp 来使得 EM357 执行相关功能。

在开发该软件工程时，可以将各模块比如 jsonrpc、调试日志、ncp 等分别进行开发与编译，然后统一生成.a 格式的静态库，将静态库添加进 ESP8266 官方开发环境中的库文件夹路径下，在 Makefile 中设置相应的静态库添加，这样可最终直接生成可执行文件从而起到源代码保护作用：

```
DEPENDS_eagle.app.v6 = \  
$(LD_FILE) \  
$(LDDIR)/eagle.rom.addr.v6.ld \  
lib/libxncp2ESP8266_RTOS.a \  
lib/libjsonrpcrtosesp8266xtensad.a \  
lib/libcjsonrtosesp8266xtensad.a \  
lib/libstdprtosesp8266xtensad.a \  
lib/libloggerrtosesp8266xtensad.a \  
lib/libringbuf16rtosesp8266xtensad.a
```

图 3- 3 连接库

利用官方提供的 Linux 下的 ESP8266 开发环境，搭配 Vbox 虚拟机在 Linux 下采用 Cmake 完成编译并生成 bin 格式固件，然后采用配套提供的烧录工具进行程序烧录：



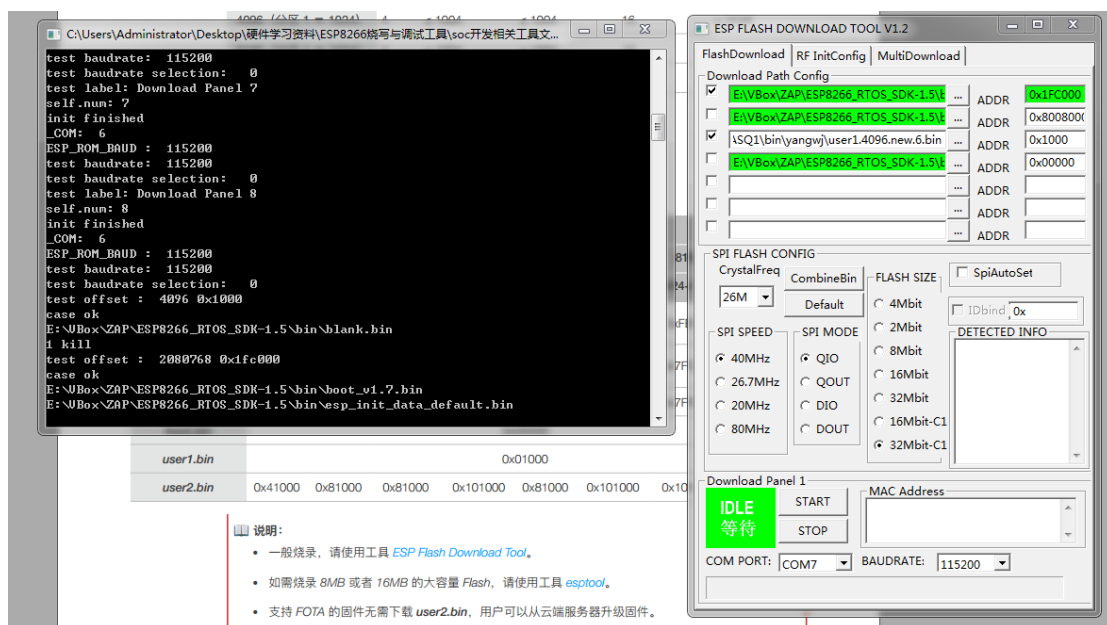


图 3- 4 烧写

## 第4章 调试

该系统旨在实现 WiFi 和 Zigbee 之间的协议转换，因此该设备的调试就需要同时具备 WiFi 和 Zigbee，针对此要求，此次调试决定采用 Control4 主机的调试软件 composer 来和 ZAP 之间进行交互，由 composer 对 ZAP 进行命令与报文的发送。命令可以分为 Zigbee 的两个基本功能“组网”“删除网络”来组成，分别以 WiFi 的形式发送，然后观察 ZAP 的执行结果。

### 4.1 组建网络调试

ZAP 初始化完毕：

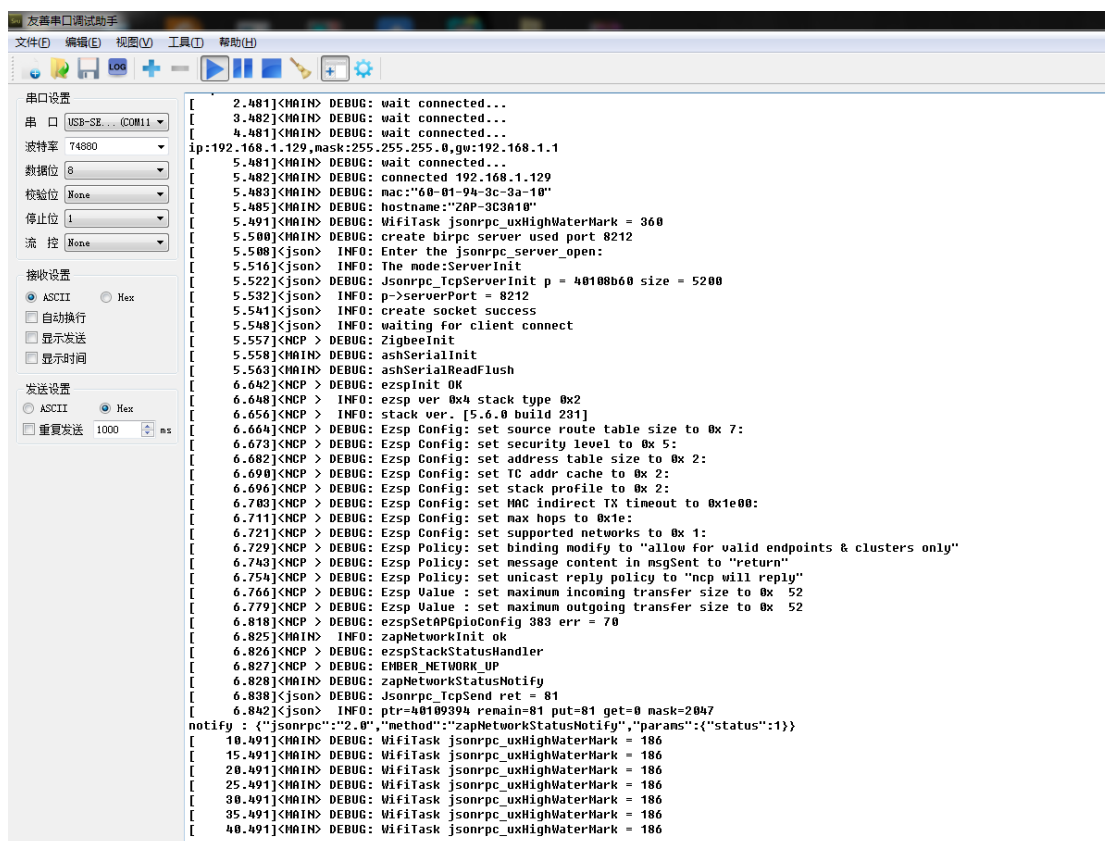


图 4-1 测试初始化

Composer 发送 WiFi 命令:

```
Lua Output
ExecuteCommand(LUA_ACTION)
ACTION: Form_Network
Call Function EX_CMD.Form_Network()
2018-05-16 15:02:58 ---- Send raw frame: {"id":39,"jsonrpc":"2.0","method":"zapFormNetwork","params":
{"netType":"Control4","panId":65535,"radioChannel":14,"radioTxPower":3}}
```

图 4-2 发送命令

ZAP 收到数据并成功执行 Zigbee 命令:

```
[ 110.492]<HMI> DEBUG: WifiTask jsonrpc_uxHighWaterMark = 186
[ 115.491]<HMI> DEBUG: WifiTask jsonrpc_uxHighWaterMark = 186
[ 119.555]<json> INFO: client_socket readable*****
[ 119.555]<json> INFO: ptr=00100b94 maxlen=2047 put=0 get=0 mask=2047
[ 119.557]<json> DEBUG: recv from client 0 0x00100b94 (140 bytes)
0c cd 02 05 00 00 00 04 7b 22 69 6a 22 3a 33 39
2c 22 6a 73 6f 6e 72 70 63 22 3a 22 32 2e 30 22
2c 22 6d 65 74 68 6f 64 22 3a 22 7a 61 70 a6 6f
72 6d 4e 05 74 77 6f 72 6b 22 2c 22 70 61 72 61
6d 73 22 3a 7b 22 6e 65 74 54 79 70 65 22 3a 22
43 6f 6e 74 72 6f 6c 34 22 2c 22 70 61 6e 49 64
22 3a 36 35 35 33 35 2c 22 72 61 64 69 6f 43 68
61 6e 6e 05 6c 22 3a 31 34 2c 22 72 61 64 69 6f
54 78 50 4f 77 65 72 22 3a 33 74 7d
[ 119.623]<json> DEBUG: FrameParse 16
[ 119.628]<json> DEBUG: FrameParse 23
[ 119.634]<json> DEBUG: FrameParse 35
[ 119.639]<json> DEBUG: FrameParse 45
[ 119.645]<json> DEBUG: FrameParse 61
[ 119.650]<json> DEBUG: FrameParse 66
[ 119.656]<json> DEBUG: FrameParse 71
[ 119.662]<json> DEBUG: FrameParse pduLength = 132
[ 119.670]<json> DEBUG: FrameParse 85
[ 119.674]<json> DEBUG: Jsonrpc_TcpTick 671
[ 119.680]<json> DEBUG: req : {"id":39,"jsonrpc":"2.0","method":"zapFormNetwork","params":{"netType":"Control4","panId":65535,"radioChannel":14,"radioTxPower":3}}信qr橙棒L0o12奥
INFO: jsonrpc_server_execute
[ 119.713]<json> INFO: Enter the execute
[ 119.722]<json> INFO: Enter the execute_request
[ 119.726]<json> INFO: Enter the parse_request
[ 119.733]<json> INFO: Parse the version OK
[ 119.740]<json> INFO: Parse the method OK
[ 119.746]<json> INFO: Parse the params OK
```

图 4-3 执行

## 4.2 删除网络调试

Composer 发送删除网络命令:

```
ACTION: Remove_Network
Call Function EX_CMD.Remove_Network()
2018-05-16 16:04:22 ---- Send raw frame: {"id":40,"jsonrpc":"2.0","method":"zapLeaveNetwork","params":{"netType":"Control4"}}
```

图 4-4 发送

ZAP 收到命令并成功执行 Zigbee 命令, 并将执行结果 WiFi 返回:

```
[ 204.290]<json> DEBUG: recv from client @ 0x40100c20 (92 bytes)
dc cd 02 03 00 00 00 54 7b 22 69 64 22 3a 34 30
2c 22 6a 73 6f 6e 72 70 63 22 3a 22 32 2e 30 22
2c 22 6d 65 74 68 6f 64 22 3a 22 7a 61 70 4c 65
61 76 65 4e 65 74 77 6f 72 6b 22 2c 22 70 61 72
61 6d 73 22 3a 7b 22 6e 65 74 54 79 70 65 22 3a
22 43 6f 6e 74 72 6f 6c 34 22 7d 7d
[ 204.336]<json> DEBUG: FrameParse 16
[ 204.342]<json> DEBUG: FrameParse 23
[ 204.347]<json> DEBUG: FrameParse 35
[ 204.353]<json> DEBUG: FrameParse 45
[ 204.358]<json> DEBUG: FrameParse 61
[ 204.363]<json> DEBUG: FrameParse 66
[ 204.369]<json> DEBUG: FrameParse 71
[ 204.374]<json> DEBUG: FrameParse pduLength = 84
[ 204.383]<json> DEBUG: FrameParse 85
[ 204.387]<json> DEBUG: Jsonrpc_TcpTick 671
[ 204.393]<json> DEBUG: req : {{"id":40,"jsonrpc":"2.0","method":"zapLeaveNetwork","params":{"netType":"Control4"}}anId":65535,"radioChannel":14,"radioTxPower":3}}
INFO: jsonrpc_server_execute
[ 204.426]<json> INFO: Enter the execute
[ 204.434]<json> INFO: Enter the execute_request
```

图 4- 5 执行

以上测试结果表明 ZAP 可以成功执行 WiFi 和 Zigbee 协议之间的转换，起到智能家居中的网关作用。

## 第5章 企业实习总结与展望

在进行此次课题研究的同时，也在江苏英索纳智能科技有限公司杭州研发中心进行一年的实习工作。在做研究期间遇到了诸多的问题与困难，比如刚开始时绘图软件使用不熟悉、电路原理理解不够、软件工程不会编译、设计方案行不通等，而每次这个时候实习单位的同事虽然每个人都有自己繁忙的工作要完成，但当我咨询时都还是会第一时间热心提供帮助，教会我很多解决问题的技巧与工具，学会如何快速、高效的解决遇到的困难，这不仅大大提高了我的毕业设计完成速度，也逐渐帮我养成了良好的思维方式与工作习惯，同时也启发了我对很多事物的观察角度，让我在生活习惯上和思想上完成了从在校学生到正式社会从业者的转变。

## 参考文献

- [1] 强静仁, 张珣, 王斌. 智能家居基本原理及应用[M]. 湖北: 华中科技大学出版社, 2017. 32~75
- [2] 郑静. 物联网+智能家居[M]. 北京: 化学工业出版社, 2017. 84~154
- [3] 王怡, 鄂旭. 基于物联网无线传感的智能家居研究[J]. 计算机技术与发展, 2015, 25(2): 234-237
- [4] 陈长友. 智能家居的设计与施工研究[D]. 陕西: 西北农林科技大学, 2009
- [5] 刘畅. 物联智能家居家庭安防系统关键技术研究[D]. 武汉: 武汉工程大学, 2014
- [6] 刘冀鹏, 张洪沛. 智能家居安防技术专利分析[J]. 软件, 2018, 39(01): 165-168
- [7] 林凡东. 智能家居控制技术及应用[M]. 北京: 机械工业出版社, 2017. 69~98
- [8] 朱敏玲, 李宁. 智能家居发展现状及未来浅析[J]. 电视技术, 2015, 39(4): 82-96
- [9] 何鹏举. 无线嵌入式家庭安防系统的设计与实现[D]. 南京: 南京邮电大学, 2017
- [10] 姜涛. 基于 ARM+WIFI 的智能家居系统的设计[D]. 吉林: 吉林大学, 2017
- [11] 翟亚芳, 张天鹏等. 基于 STM32 的家庭智能安防控制系统设计[J]. 黑龙江大学自然科学学报, 2016, 33(01): 118-123
- [12] 范鹏. 基于 ARM-Linux 平台的家庭智能安防系统的研究与设计[D]. 南京: 南京理工大学, 2014

### 附录 1、教育部“卓越计划”企业实习成绩表

题    目		智能网关设备					
姓 名	郑江湖	学 号	2014330300129	专 业	机械电子工程		
成    果 (40%)		报    告 (30%)		表    现 (30%)		合 计	
<div>企业指导教师（签章）</div> <div>校内指导教师（签章）</div> <div>年    月    日</div>							

