



广州大学华软软件学院
South China Institute of Software Engineering.GU

毕业论文(设计)

课题名称 基于树莓派的
多功能家庭环境控制系统的设计

系 别 电子系

专业(方向) 通信工程

班 级 16级通信工程2班

学生姓名 卢 跃

指导教师 李 芳

完成日期 2020年4月

教务处 制

毕业论文（设计）真实性诚信保证书

本人所撰写的毕业论文（设计）是在老师指导下独立完成，没有弄虚作假，没有抄袭或拷贝他人研究成果。我郑重承诺：文责自负。

签 名：李跃

2020年 4月 30日

广州大学华软软件学院

本科毕业设计任务书

设计题目 基于树莓派的多功能

家庭环境控制系统的设计

系 别 电子系

专 业 通信工程

班 级 16 通信(2)班

学 号 1640915223

学生姓名 卢跃

指导教师 李芳

下发时间： 2019 年 10 月 10 日

毕业设计须知

1、认真学习和执行广州大学华软软件学院学生毕业论文（设计）工作管理规程；

2、努力学习、勤于实践、勇于创新，保质保量地完成任务书规定的任务；

3、遵守纪律，保证出勤，因事、因病离岗，应事先向指导教师请假，否则作为缺席处理。凡随机抽查三次不到，总分降低 10 分。累计缺席时间达到全过程 1 / 4 者，取消答辩资格，成绩按不及格处理；

4、独立完成规定的工作任务，不弄虚作假，不抄袭和拷贝别人的工作内容。否则毕业设计成绩按不及格处理；

5、毕业设计必须符合《广州大学华软软件学院普通本科生毕业论文（设计）规范化要求》，否则不能取得参加答辩的资格；

6、实验时，爱护仪器设备，节约材料，严格遵守操作规程及实验室有关制度。

7、妥善保存《广州大学华软软件学院本科毕业设计任务书》。

8、定期打扫卫生，保持良好的学习和工作环境。

9、毕业设计成果、资料按规定要求装订好后交指导教师。凡涉及到国家机密、知识产权、技术专利、商业利益的成果，学生不得擅自带离学校。如需发表，必须在保守国家秘密的前提下，经指导教师推荐和院领导批准。

课题名称	基于树莓派的多功能家庭环境控制系统的设计
完成日期：	2020 年 4 月 2 日
<p>一、题目来源及原始数据资料：</p> <p>家庭环境多种多样，其中居住环境最为重要，一个良好的居住环境能让人健康舒适的生活。若居住环境遭到污染，那么居住者的身心健康将会受到有害的影响。所以对家庭居住环境的监控和控制非常重要。常见影响居住环境的因素有甲醛、挥发性有机化合物、温度、湿度、光强、二氧化碳等排量等。</p> <p>树莓派是一个只有信用卡大小的、基于 Linux 系统的微型计算机，具有很强的扩展性能。树莓派小巧的体积和强大的扩展性非常适合完成本课题的开发。</p> <p>本课题所要设计的家庭环境控制系统，是使用树莓派为服务器，传感器为外围设备来监控家庭环境状况的系统。能实现在环境数据异常的情况下完成预警、提示和模拟控制外围家电来改善当前的环境状况。系统使用 Python、C、HTML5、PHP 等语言完成开发并实现远程监控，还可以利用树莓派的扩展性完成一些实用性的功能，使生活更加安全、舒适、方便。</p>	
<p>二、毕业设计要求：</p> <p>本课题要求设计一个环境监控系统。该系统能监测家中光强、温度、湿度、甲醛浓度、TVOC、CO₂eq 等数据并作出对应的处理。要求如下：</p> <ol style="list-style-type: none"> 1. 设计一个传感器集成板，使用 ATmega328P 系列单片机主控板控制各种传感器。设计相关程序，处理并统一输出数据，主控板以插件形式嵌入电路板中。 2. 使用 Python 完成监控系统服务程序的开发，系统能即时读取环境数据，需要对不同的数据做出处理，并能根据数据自动控制外部电器。 3. 用户可主动通过网页主动查看环境信息，能在数据异常时被动接收消息。 4. 使用 HTML5 和 PHP 开发一个网页，网页可操控监控系统，同时能实现对局域网内电脑的远程开机等实用功能。要求网页和 Python 监控系统服务程序能够协同工作。同时配置树莓派的系统，使网页可以在非局域网访问，考验学生对 Linux 的配置能力。 <p>本项目主要考验学生的软件设计能力，以软件为主，硬件为辅，结合各类软硬件知识并应用到实际中，考核学生所学的编程知识和让不同语言协同工作的能力。论文要求学生独立完成，要有足够的工作量；要按照规定格式写出设计说明书。</p>	

三、进度安排、应完成的工作量:

- 1、10月30日前上交论文任务书,11月12日前论文任务书全部通过;
- 2、期末19周前完成论文初稿50%;
- 3、2020年2月25日提交论文初稿电子板;
- 4、2020年3月30日前完成论文终稿,《毕业实鉴定表》,并准备答辩。

四、主要参考文献

- [1] 勃鲁姆/布莱斯纳罕. 树莓派 Python 编程入门与实战. 北京:人民邮电出版社. 2016.
- [2] 蒙克. Raspberry Pi 开发实战. 北京:机械工业出版社. 2015.
- [3] 赵英杰. 完美图解物联网 IoT 实操 ESP8266 Arduino, Cordova 物联网移动 App, JavaScript 微控制器编程. 北京:电子工业出版社. 2018.
- [4] 董付国. Python 程序设计. 北京:清华大学出版社. 2016.
- [5] 谢平. PCB 设计与加工. 北京:北京理工大学出版社. 2017.
- [6] 蒙克. 电子创客案例手册. Arduino 和 Raspberry Pi 电子制作实战 北京:清华大学出版社. 2018.
- [7] 唐四新. PHP 动态网站开发. 北京:清华大学出版社. 2015.
- [8] 黄小平. Web 页面开发. 广州:华南理工大学出版社. 2017.

指导教师(签名):

李第

系(教研室)主任(签名):

夏国清

李第

广州大学华软软件学院
本科毕业论文（设计）成绩单

姓 名	卢 跃	学 号	1640915223	专业（方向）	通信工程
题 目	基于树莓派的多功能家庭环境控制系统的设计				
指导教师评语 （占论文总成绩的60%）	<p>卢跃同学在毕业设计期间，学习态度认真，设计并制作了“基于树莓派的多功能家庭环境控制系统”。其程序设计思路清楚，实现了对家庭环境数据采集及无线数据传输功能。系统具有一定的创新点。</p> <p>论文结构符合学院要求，条理清楚，语言流畅。根据该生完成毕业设计论文及其硬件作品的实际情况，同意卢跃同学参加毕业设计论文答辩。</p> <p>成绩：57</p> <p style="text-align: right;">签名：李芳 夏国清 2020 年 4 月 10 日 邓革</p>				
评阅教师评语 （占论文总成绩的20%）	<p>该生论文格式基本符合学院的格式规范，条理清楚，语言通顺，设计的系统具有一定的特点及实用性。根据该生的论文完成情况，同意卢跃参加毕业设计论文答辩。</p> <p>成绩：17</p> <p style="text-align: right;">签名：马青云 夏国清 2020 年 4 月 15 日 邓革</p>				
答辩组评语 （占论文总成绩的20%）	<p>作品功能要求完全实现，有一定的创新性，实物制作质量好。论文内容完整，结构合理，条理清晰。答辩时讲述的系统性逻辑性强，语言表达能力强，回答问题正确。</p> <p>成绩：18.3</p> <p style="text-align: right;">夏国清 邓革 陈华兴 何苏如 答辩组成员： 2020 年 4 月 30 日 邓革</p>				
总成绩	<p style="text-align: center;">优</p> <p style="text-align: right;">系主任签名：夏国清 系（章） 2020 年 5 月 10 日 电子系</p>				

摘要 为了能够直观的了解居住环境中挥发性有机化合物、温度、湿度、光强、二氧化碳等排量等信息以保障居住者居住环境的安全，本项目基于B/S架构，采用了Python、PHP、HTML5、Bootstrap、jQuery、Vue.js等技术，使用树莓派和Arduino平台，实现了一个多功能的家庭环境监控系统。该系统能让用户随时了解家庭环境的实时状况，能在环境不安全时控制一些设备，并能对居住者进行提醒，还有一些实用性的功能。产品具有界面美观，实用性强、后续扩展性好等特点。

关键词 智能家居；物联网；树莓派

ABSTRACT In order to intuitively understand the TVOC, temperature, humidity, light intensity, carbon dioxide and other displacement information in the living environment to ensure the safety of the living environment of the residents, the project adopts python, PHP, HTML5, bootstrap, jQuery, vue.js and other technologies based on the B / S architecture, uses Raspberry pi and Arduino platform to realize a multi-functional monitoring system of the family environment. It can let users know the real-time situation of the family environment at any time, control some devices when the environment is not safe, and remind the residents, and have some practical functions. The product has a good interface, strong practicability and good subsequent expansion.

KEY WORDS Smart home; IoT; Raspberry Pi

目 录

1. 绪 论.....	1
1.1 研究背景与现状.....	1
1.2 研究目的与意义.....	1
1.3 研究方法.....	1
2. 硬件及运行开发环境.....	2
2.1 硬件及模块.....	2
2.1.1 树莓派 2B.....	2
2.1.2 CF-WU810N 无线网卡.....	2
2.1.3 Arduino Nano.....	2
2.1.4 WZ-S-K 甲醛传感器.....	3
2.1.5 DHT11 温湿度传感器.....	3
2.1.6 SGP30 多像素气体传感器.....	3
2.1.7 微雪 7 英寸 C 型触摸屏.....	3
2.2 树莓派运行环境.....	3
2.2.1 树莓派操作系统.....	3
2.2.2 驱动触摸屏.....	3
2.2.3 驱动无线网卡.....	4
2.2.4 Python 运行环境.....	4
2.2.5 PHP 环境.....	4
2.2.6 网页服务器程序.....	5
2.3 Arduino 开发环境.....	5
3. 系统介绍及设计方案.....	6
3.1 总体设计及功能简介.....	6
3.2 系统定位及设计原则.....	7
4. 硬件设计.....	8
4.1 传感器集成板的硬件设计.....	8
4.1.1 集成板简介.....	8

4.2.1 协议及引脚的选定.....	8
4.2.2 SGP30 传感器电路设计.....	9
4.2.3 光线传感器电路设计.....	9
4.2.4 线路选择开关的设计.....	10
4.2.5 PCB 及原理图设计.....	11
5. 软件设计.....	12
5.1 环境控制系统的设计.....	12
5.1.1 环境控制系统的主程序流程.....	12
5.1.2 传感器数据获取线程.....	13
5.1.3 数据推送线程.....	13
5.1.4 安全警报线程.....	14
5.1.5 外围设备控制线程.....	17
5.1.6 显示屏控制线程.....	17
5.1.7 程序控制线程.....	18
5.1.8 实时天气获取的实现.....	19
5.1.9 全局变量的解决.....	20
5.2 控制端的实现.....	21
5.2.1 控制端简介及界面.....	21
5.2.2 数据获取功能的统一架构.....	22
5.2.3 自适应页面布局.....	22
5.2.4 Vue 实例.....	24
5.2.5 传感器数据显示的实现.....	25
5.2.6 树莓派状态获取功能的实现.....	26
5.2.7 本地 IP 获取功能的实现.....	27
5.2.8 计算机远程唤醒功能的实现.....	27
5.2.9 控制端公网访问及权限认证.....	27
5.3 集成板软件设计.....	28
5.4 数据库设计.....	29
6. 系统测试.....	31

7. 总 结.....	33
参考文献.....	34
附 录.....	35
致 谢.....	42

1. 绪 论

1.1 研究背景与现状

现代社会中，人们对居住环境的要求较高，各类家具材料进入了人们的家中。但若使用了不安全的材料，家庭环境将会被材料中散发的化学物质所污染，居住者的人身安全也将受到威胁，而这类污染物质需要几年的时间才能完全挥发。在不了解当前居住环境质量的情况下，人们住进房总会伴随着或多或少的恐慌。

1.2 研究目的与意义

本项目的研究目的主要是让居住者能够清晰的了解当前的家庭环境，并在环境的某些数值异常时提醒居住者或打开相应的应对设备，给居住者应对的空间，让居住者能够安心的居住，这就是本项目的意义所在。

1.3 研究方法

本项目是一个多技术结合的系统。硬件方面：使用树莓派和基于 ATmega328P 的 Arduino Nano 开发板；软件方面：前端用到的技术有 HTML5、CSS、JavaScript、Vue.js、Bootstrap、Ajax，后端使用了 Python 和 PHP 技术，数据库选用了无需搭建环境的 Sqlite3。前端通过 Bootstrap 实现了 PC 端、移动端的自适应显示，通过 Vue.js 控制内容的显示逻辑，避免了不同终端需要分开设计不同页面和 HTML5 中常见的代码高重复率的问题。后端使用 Python 完成对硬件数据的读取、控制。

2. 硬件及运行开发环境

2.1 硬件及模块

2.1.1 树莓派 2B

树莓派是一个小型的 PC，功能强大。常见树莓派的配置如表 2-1 所示。

表 2-1 常见树莓派型号的对比

型号	CPU	RAM	USB	网络接口	供电电压	闲置电流	估价
4B	BCM2711	1-4G	USB3*4	千兆有线+WIFI	5V	0.6	270-440
3B	BCM2837	1G	USB2*4	百兆有线+WIFI		0.30	230
2B	BCM2836	1G	USB2*4	百兆有线		0.22	100
Zero	BCM2835	512MB	microUSB*1	WIFI		0.10	70

本项目对 USB 接口、网口和 CPU 的性能没有多大要求，所以选择了价格相对便宜、闲置功耗较低、拥有标准 USB 和百兆有线接口的树莓派 2B 加上 USB 无线网卡来完成项目。

2.1.2 CF-WU810N 无线网卡

无线网卡选用 COMFAST 公司生产的 CF-WU810N 网卡，这款网卡支持 Linux、Windows、MacOS 等操作系统，是一款单频 2.4G 支持 150Mbps 传输速率的 USB2.0 网卡，使用芯片为 RTL8188EUS。选择原因：多系统支持、价格低廉。

2.1.3 Arduino Nano

Arduino 是开源的电子原型平台，有多种型号的 Arduino 供开发者选用。本项目选用 Arduino Nano 作为主控板。选择原因：体积小巧。

2.1.4 WZ-S-K 甲醛传感器

甲醛传感器选用英国 Dart 公司生产的 WZ-S 系列的 WZ-S-K 型传感器,使用串口传输进行通信。选择原因:精度相对国产传感器较为准确,免标定,方便更换。

2.1.5 DHT11 温湿度传感器

温湿度传感器选用奥松科技的 DHT11。这是一款使用单总线传输数据的电容式免标定温湿度传感器模块。选择原因:使用方便。

2.1.6 SGP30 多像素气体传感器

SGP30 是瑞士 Sensirion 公司生产的多像素气体传感器,使用 I²C 传输数据,能够检测 TVOC 及 eCO₂ 的值,并且支持温度湿度补偿。SGP30 长期稳定性好,漂移低,易于集成。选择原因:体积小巧,支持温湿度补偿。

2.1.7 微雪 7 英寸 C 型触摸屏

触摸屏是本项目主要的输入输出设备,使用了微雪的 7inch HDMI LCD (C) 型的触摸屏。这是一款 HDMI+USB 方案的触摸屏,支持 PWM 背光调节。选择原因:配备的外壳支持安装树莓派。

2.2 树莓派运行环境

2.2.1 树莓派操作系统

Raspbian 是官方适配的系统,基于 Debian。Raspbian 目前最新的版本是 Buster。选择原因:官方系统,性能稍弱,但兼容性强。

2.2.2 驱动触摸屏

微雪的显示屏在树莓派上使用时,需要设置分辨率并解除树莓派 usb 供电不足的问题,否则树莓派启动后将导致花屏、闪屏、显示不全、供电不足等问题。为解决问题,需要在树莓派的/boot/config.txt 中加入如下代码,同时需要注

释掉 `dtoverlay=vc4-fkms-V3D`。

2.2.3 驱动无线网卡

由于 WU810N 使用 RTL8188EUS 驱动在树莓派内核中未编译^[1]，所以驱动需要手动安装。本项目使用 aircrack-ng 开发的 `rtl8188eus` 驱动（项目地址：<https://github.com/aircrack-ng/rtl8188eus>）。

本次驱动的安装使用 DKMS（Dynamic Kernel Module Support）。由于 Linux 的驱动对内核版本的依赖关系很大，当内核版本变动时，之前编译的驱动将失效，需要重新编译。DKMS 是 Dell 开发的项目，使用 DKMS 方式安装的驱动，能够在 Linux 内核版本变动时，自动重新编译驱动。

为了编译驱动，还需要安装 Linux 内核 headers，它提供了编译时所需的各种函数和结构定义。树莓派的 headers 包名为 `raspberrypi-kernel-headers`。

2.2.4 Python 运行环境

本项目用到的 Python 版本为 Python3。主要用到的包如表 2-2 所示：

表 2-2 本项目主要用到的 Python 包

Package name	简介
<code>configobj</code>	配置文件读取、写入和验证
<code>pyserial</code>	Python 串行端口扩展
<code>logging</code>	输出日志
<code>traceback</code>	异常信息打印
<code>sqlite3</code>	Sqlite3 数据库支持
<code>json</code>	json 处理
<code>email</code>	邮件处理

2.2.5 PHP 环境

本项目用到的 PHP 版本为 PHP7，用到的模块有 `php-sqlite3`。

2.2.6 网页服务器程序

本项目用到的网页服务器有两个，开发时使用 PHP Built-in web server，非开发时使用 Apache HTTP Server。

本着简易化原则，Apache 使用默认配置，将网页直接软链接到 Apache 的默认网页目录下：

```
ln -s /home/pi/rpi-envsys/python/web/ /var/www/html/
```

2.3 Arduino 开发环境

本系统主要用到的 Arduino 库如表 2-3 所示。打开 Arduino->工具->管理库可下载需要的库。

表 2-3 用到的 Arduino 库

库名	简介
ArduinoJson	Arduino 的 Json 处理库
DHT-sensor-library	DHT 驱动，依赖于 Adafruit_Sensor 库
Adafruit_Sensor	Adafruit 统一传感器驱动
Adafruit_SGP30	SGP30 传感器驱动
SoftwareSerial	软串口模拟库

3. 系统介绍及设计方案

3.1 总体设计及功能简介

本系统的实际成品图如图 3-1 所示：整体硬件由 7 英寸可触摸显示屏、树莓派、本项目设计的 Arduino 传感器集成板构成。

系统结构图如 3-1 所示：

数据收集模块：由传感器集成板负责获取传感器的数据，将数据由串口发送给树莓派。

数据处理模块：能够完成传感器数据、联网实时天气数据的处理，并能根据不同的请求发送不同的数据。

数据监控模块：判断传感器数值是否正常，并根据其状态发送警报或控制电器。

用户交互模块：能够通过各种支持 Browser 的设备查看环境信息、控制局域网设备、修改系统设置。

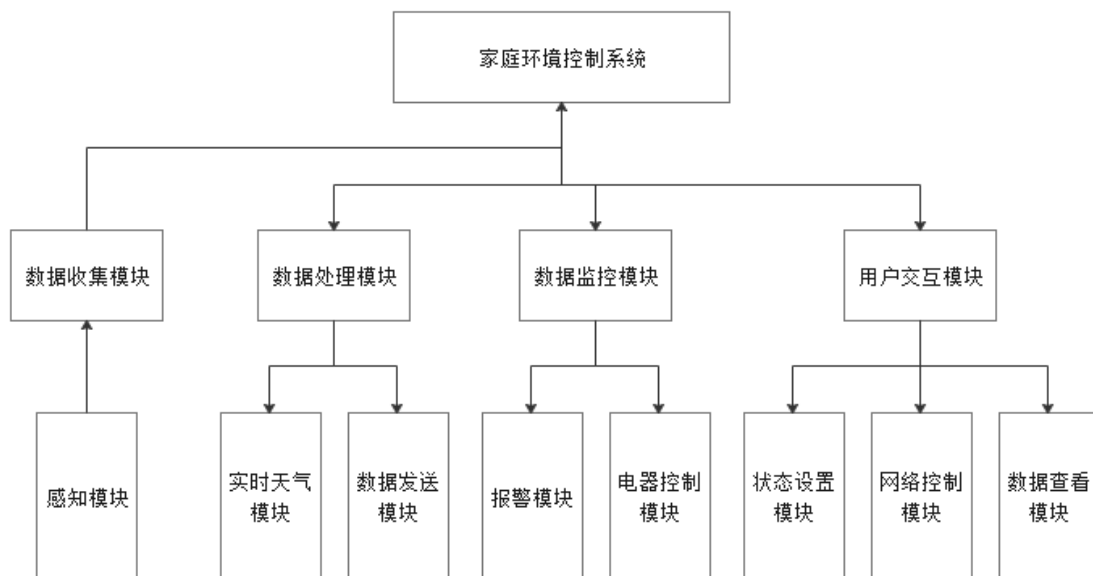


图 3-1 系统结构图

3.2 系统定位及设计原则

本产品定位为单用户室内 24 小时运作项目，故设计系统时需要遵循以下原则：

简化：由于产品为单用户使用，使用简单的模型和数据库设计可以降低 bug 的产生。

低能耗：本产品需要 24 小时不停的运行，需要尽可能的降低产品能耗以节约电能。

多功能：由于家庭环境多数情况下趋于安全，需要一些实用的功能增加产品的平均可用性。

舒适性：系统在室内使用，应尽可能的减少对生活带来的影响。

4. 硬件设计

4.1 传感器集成板的硬件设计

4.1.1 集成板简介

传感器集成板是本系统最重要的一个部分，它使用 **Arduino** 作为核心控制模块，集中处理了传感器的数据，完成了对外部数据的读取和发送。作为树莓派的扩展 **HAT**，其大小完全对应树莓派的大小。

4.2.1 协议及引脚的选定

根据实际用到的传感器，需要用到的协议及选定的引脚如表 4-1 所示：。

表 4-1 协议和引脚表

传感器	协议类型	Arduino 中可用针脚
DHT11	单总线	任意数字信号引脚
SGP30	I ² C	A4 (SDA)、A5 (SCL)
WZ-S	串口通讯	RX0、TX1
光线传感器	-	任意模拟信号引脚

Arduino 的 RX0、TX1 引脚需要连接树莓派的板载串口，无法用于连接 WZ-S 传感器，所以使用了 **SoftwareSerial** 库将 D2、D3 两个数字接口模拟成软串口。最终各传感器使用的引脚如表 4-2 所示。

表 4-2 实际使用的引脚

传感器	实际引脚
DHT11	D4
SGP30	A4、A5
WZ-S	D2、D3
光线传感器	A0

4.2.2 SGP30 传感器电路设计

SGP30 使用 1.8V 供电，1.8V 使用 AMS1117-1.8 模块进行电平转换，电路图如图 4-1 所示：R2 和 R3 将 SDA 和 SCL 默认拉到高电平，使模块正常工作，AMS1117-1.8 将 5V 转换成 1.8V，为模块供电。

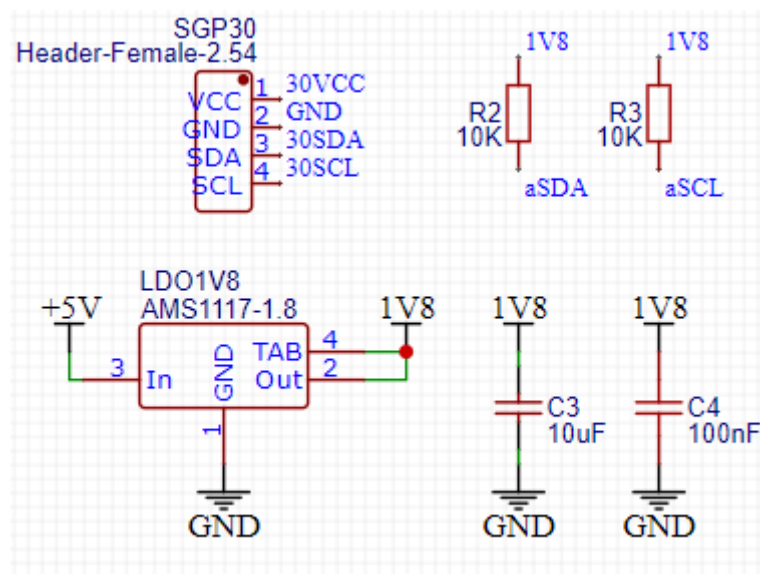


图 4-1 SGP30 传感器电路设计

4.2.3 光线传感器电路设计

光线传感器主要由 LM393 电压比较器、电位器、电阻组成，支持模拟输出和数字输出两种方式，模拟输出由支持数模转换的 Arduino 读取，数字输出可由线路选择开关让树莓派的 I026 读取，其电路图如图 4-2 所示：光敏电阻与一个 10K 的电阻串联后再与电位器并联，电位器充当滑动变阻器的作用。模拟输出将直接读取 R1 和光敏电阻之间的电压，光线强度越强，读到的数据越小。数字输出则是将 R1 和光敏电阻之间的电压和电位器所分的电压做对比，当 R1 和光敏电阻之间的电压大于电位器的电压时，输出高电平。

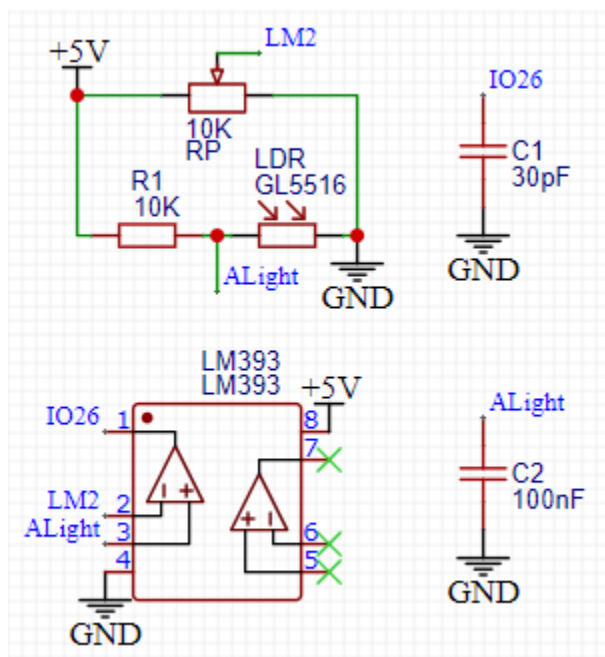


图 4-2 光线传感器电路设计

4.2.4 线路选择开关的设计

为了方便开发过程中对电路连线的修改，传感器集成板上设计了一个线路选择开关，方便传感器使用不同的供电和不同的连接方式，如图 4-3 所示，具体使用方式如表 4-3 所示。

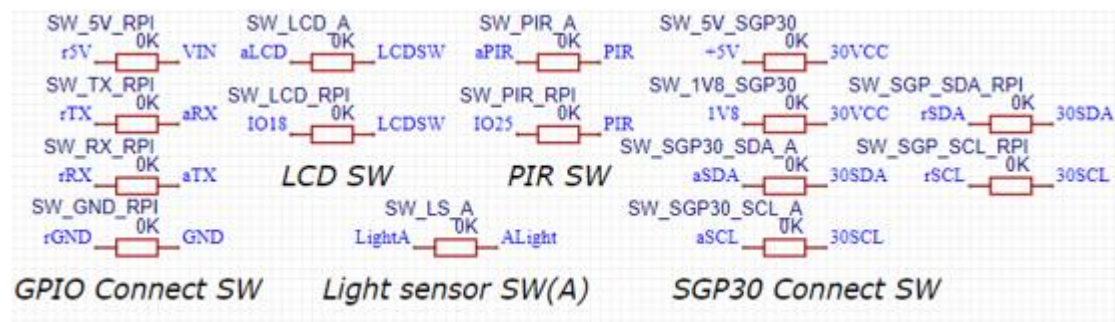


图 4-3 线路选择开关

表 4-3 实际使用的引脚

开关名称	连接作用	断开作用
SW_5V_RPI/SW_GND_RPI	使用树莓派供电	使用 USB 供电
SW_TX_RPI/SW_RX_RPI	使用树莓派板载串口	使用 USB 串口
SW_LCD_A	Arduino 控制显示屏	-
SW_LCD_RPI	树莓派控制显示屏	-

续表 4-3

SW_PIR_A	PIR 传感器连接 Arduino	-
SW_PIR_RPI	PIR 传感器连接树莓派	-
SW_5V_SGP30	使用 5V 为 SGP30 供电	-
SW_1V8_SGP30	使用 1.8V 为 SGP30 供电	-
SW_SGP30_SDA_A/SW_SGP30_SCL_A	SGP30 连接到树莓派	-
SW_SGP30_SDA_RPI/SW_SGP30_SCL_RPI	SGP30 连接到 Arduino	-
SW_LS_A	连接光线传感器模拟信号	-

4.2.5 PCB 及原理图设计

PCB 的设计及原理图的详细设计见附录 1 和附录 2。

5.1.2 传感器数据获取线程

传感器数据获取线程的流程如图 5-2 所示：程序初始化串口成功后，将循环查询串口缓冲区中是否有数据。无数据则记一次错误，有数据则读取数据，清零错误数。当错误次数大于 10 次时，程序将抛出错误，认为集成板未连接，将重新初始化串口。读取数据后，程序尝试将接收到的数据转换为字典，若转换成功，则将解析后的数据写入全局变量。若转换失败，则说明串口接收数据不完整，开始重新初始化串口。

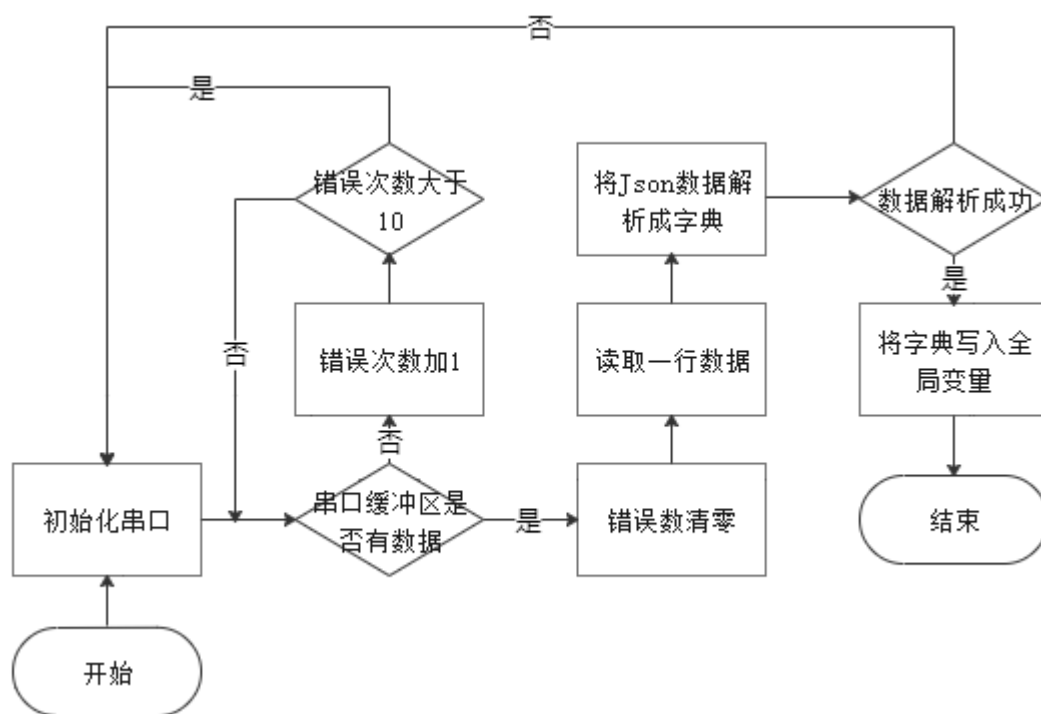


图 5-2 传感器数据获取线程流程图

5.1.3 数据推送线程

数据推送线程实现了发送传感器信息以及发送实时天气预报的功能，其程序流程如图 5-3 所示：线程将监听本地 16868 端口，根据客户端发来的不同信息处理不同的数据，最后再将处理好的数据转换为 Json 并发回客户端。

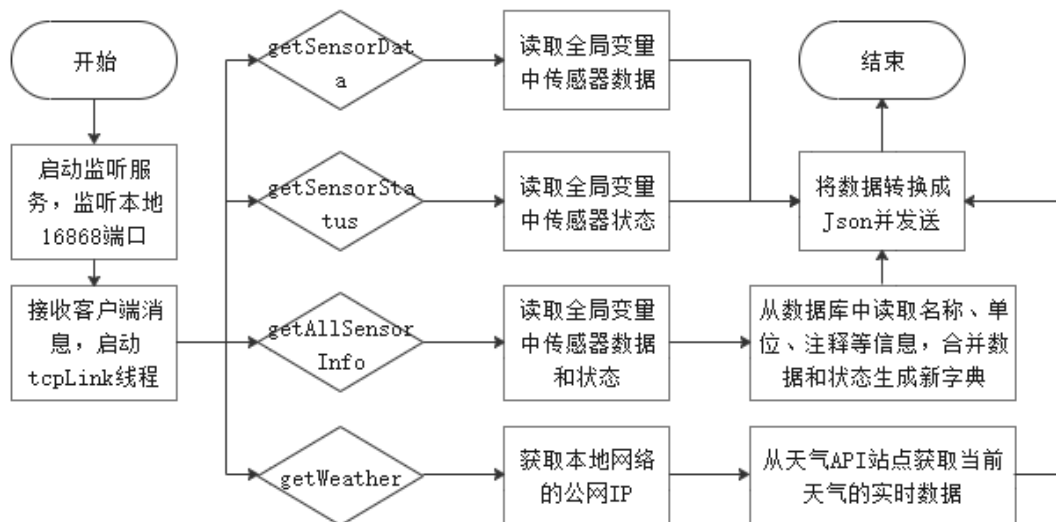


图 5-3 数据推送线程流程图

5.1.4 安全警报线程

安全警报线程实现了当传感器数据异常时对管理员进行提醒，每一个传感器都将启动一个安全警报线程，其程序流程图如图 5-4 所示：程序启动后判断是否需要使用此线程，若需要使用，程序将读取该传感器的数据，判断其运行模式。模式有两种：

模式一：将传感器的实时数据与数据库中的 lowVar、highVar 做对比，小于 lowVar 或大于 highVar 时，将状态设置为警告。

模式二：将传感器的实时数据与数据库中的 lowVar、highVar 做对比，大于 lowVar 小于 highVa 时，将状态设置为预警；大于 highVar 时，将状态设置为警告。

读取完当前状态后，调用报警函数。

报警函数流程如图 5-5 所示：首先判断本次状态是否和上一次调用函数的状态一致。当本次状态和上次状态一致时，只记录出错状态，当连续出错次数达到 10 次以上时，系统认为该传感器出现故障，并调用邮件函数发送邮件通知管理员。当本次状态和上次状态不一致时，将根据实际的状态变化调用邮件函数发送邮件给管理员。

邮件发送函数流程如图 5-6 所示：首先将读取上次发送邮件的时间与本次时间的的时间差是否大于配置文件中定义的发送间隔时间值，避免数据波动时反复发

送邮件对管理员造成骚扰。若时间差大于发送间隔时间，读取数据库中的数据判断该传感器是否开启了邮件通知，没开启则不发送邮件，开启了则根据本次状态的变化生成 HTML 模板，发送邮件。

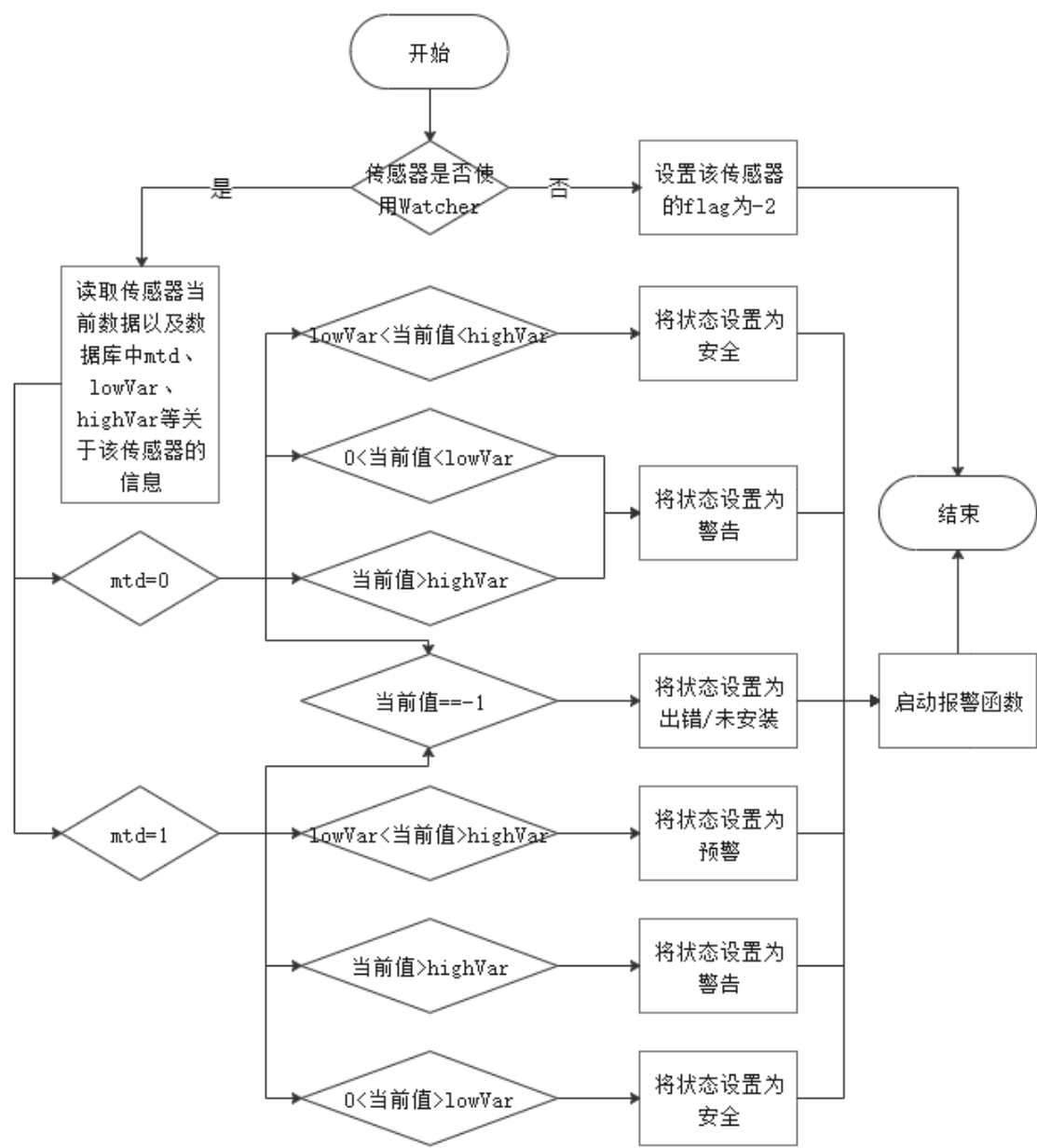


图 5-4 安全警报线程流程图

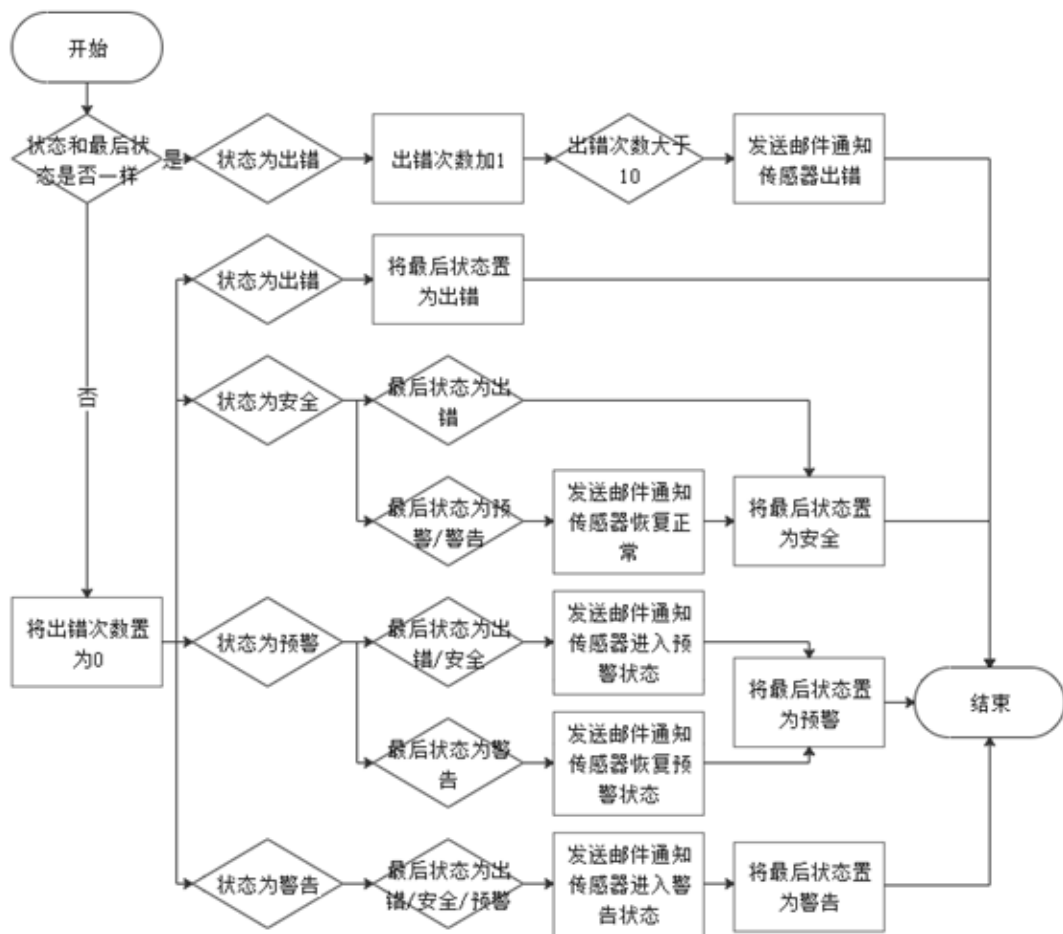


图 5-5 报警函数流程图

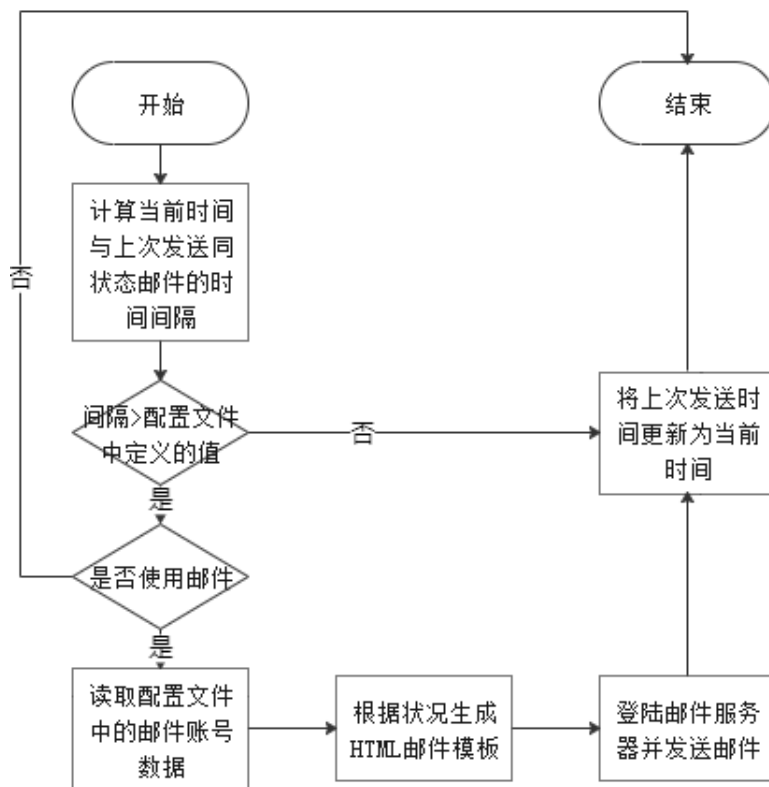


图 5-6 邮件发送函数流程图

5.1.5 外围设备控制线程

外围控制线程实现了对家用电器的智能控制，其流程如图 5-7 所示：当某数值的状态为危险时，若有能控制的设备，则打开，待状态恢复正常时则关闭设备。本系统中树莓派预留了 12 个 IO 口供设备使用，最多能控制 12 个外围设备。

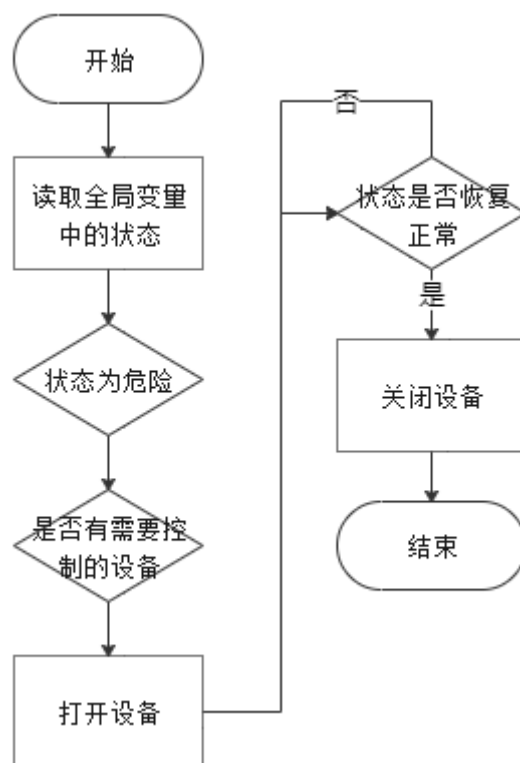


图 5-7 外围设备控制线程流程图

5.1.6 显示屏控制线程

显示屏控制线程主要实现在没有人体活动时关闭显示屏以达到节约用电的目的。显示屏控制线程的流程如图 5-8 所示：开启设备时，显示屏默认打开一分钟，每当 PIR 传感器检测到人体活动时，显示屏以当前时间为起点延长显示 1 分钟。当无法检测到人体活动时，显示屏将在 1 分钟后自动关闭，直到下次检测到人体活动时亮起。

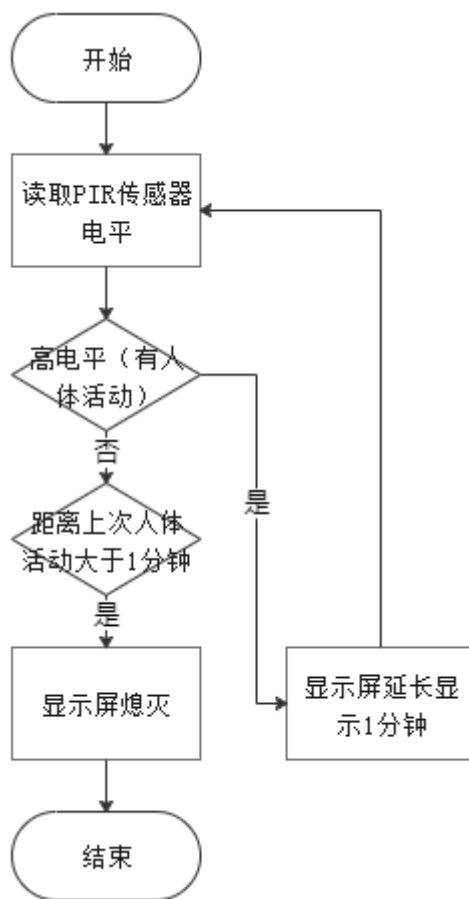


图 5-8 显示屏控制线程流程图

5.1.7 程序控制线程

程序控制线程主要控制程序的重启、关闭，整个程序将在该线程结束后关闭。其流程图如图 5-9 所示。通常情况下，本线程将不会结束。

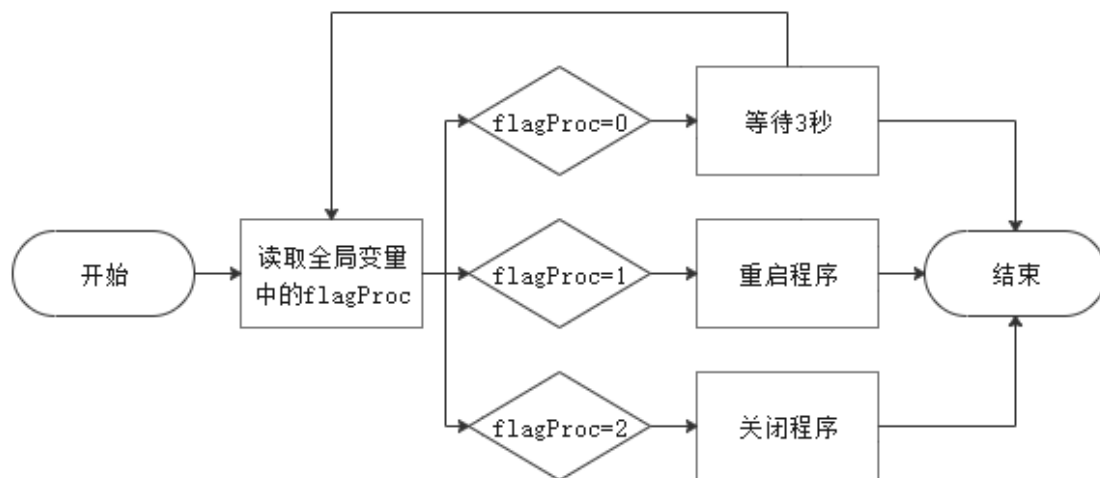


图 5-9 程序控制线程流程图

5.1.8 实时天气获取的实现

实时天气使用天气 API (<http://www.tianqiapi.com/>) 获取, 天气 API 支持使用城市、IP 等多种方式获取当前的实况天气。由于树莓派没有 GPS, 程序主要通过 IP 地址判断当前位置从而获取天气。由于局域网的存在, 此处使用公网 IP 获取天气。公网 IP 的使用 ip-api.com 提供的接口获得, 地址为 <http://ip-api.com/json/>, ip-api 的接口返回了一个 Json 字符串, 其中 query 键的值即为本地公网 IP 的地址。

天气 API 的请求地址为 <https://tianqiapi.com/api>, 参数如表 5-1 所示: 获取实时天气只需提供 appid、appsecret、version、ip 这四个参数。appid 和 appsecret 单独在配置文件中配置, version 值选择 v6, 代表免费实时天气接口, IP 从 ip-api 获取。天气 API 返回的数据中各键含义如表 5-2 所示。程序将把获得的数据直接通过 socket 发送给 PHP, 由前端处理。

表 5-1 天气 API 参数表

参数名	必选	类型	说明
appid	是	string	用户 appid
appsecret	是	string	用户 appsecret
version	是	string	接口版本标识
cityid	否	string	城市 ID
city	否	string	城市名称
ip	否	string	IP 地址
callback	否	string	jsonp 参数
vue	否	string	跨域参数

表 5-2 天气 API 返回值参表

参数名	类型	说明
cityid	String	城市 ID
date	String	当前日期
week	String	当前星期

update_time	String	气象台更新时间
city	String	城市名称
cityEn	String	城市英文名称
country	String	国家名称
countryEn	String	国家英文名称
wea	String	天气情况
wea_img	String	天气对应图标
tem	String	实时温度
tem1	String	高温
tem2	String	低温
win	String	风向
win_speed	String	风力等级
win_meter	String	风速
humidity	String	湿度
visibility	String	能见度
pressure	String	气压 hPa
air	String	空气质量
air_level	String	空气质量等级
air_tips	String	空气质量描述

5.1.9 全局变量的解决

选用多线程编程很大一部分原因是 Python 下不同线程能共享全局变量，然而 Python 多线程的全局变量只对一个文件内的变量有效，不同文件中的全局变量是无法相互访问的，然而将所有代码写在一个文件内也是不合理的。为了解决这个问题，项目内增加了一个专门存储全局变量的文件。文件内预定义了一个 `globalVar` 类用于存储全局变量。还定义了一些常用方法方法，对 `globalVar` 中的变量进行操作。不同文件中需要用到全局变量时，只需要导入该文件并执行特

定的方法就可以对全局变量进行读写。定义的方法如表 5-3 所示。

表 5-3 操作全局变量的方法

方法	说明
setGlobalVar(name, val)	设置一个全局变量的值
setFlagVar(name, val)	设置类中预定义的 flag 变量的值
getGlobalVar(name)	获取全局变量的值
getFlagVar(name)	获取 flag 中某个变量的值
removeGlobalVar(name)	删除一个全局变量

5.2 控制端的实现

5.2.1 控制端简介及界面

控制端端采用 Vue、Bootstrap、PHP 等技术完成，界面使用黑白配色，能够自适应手机、平板、电脑等设备，是环境监控系统的控制端。如图 5-10 所示：状态概览页显示时间、当前天气；情景设置页配置传感器的邮件、监控等相关信息；系统设置页配置和管理程序的设置；设备管理页能查看树莓派信息，并能远程唤醒局域网电脑。每一个页面都是一个 section，通过 js 控制显示。



图 5-10 控制端界面

5.2.2 数据获取功能的统一架构

控制端获取数据统一使用 AJAX 获取，其架构如图 5-11 所示。



图 5-11 数据获取架构

5.2.3 自适应页面布局

控制端使用 Bootstrap 完成对不同分辨率设备的适配，Bootstrap 使用栅格系统实现响应式页面，一行分为 12 列，在不同分辨率下的控制端的占位情况有所不同。网格尺寸为 Extra small（宽度小于 576px）时的布局如图 5-12 所示，此时时钟模块不显示。网格尺寸为 Small（宽度 ≥ 576px）和 Medium（宽度 ≥ 768px）时的布局如图 5-13 所示。网格尺寸为 Large（宽度 ≥ 992px）时的布局如图 5-14 所示。网格尺寸为 Extra large（宽度 ≥ 1200px）时的布局如图 5-15 所示。



图 5-12 Extra small 布局

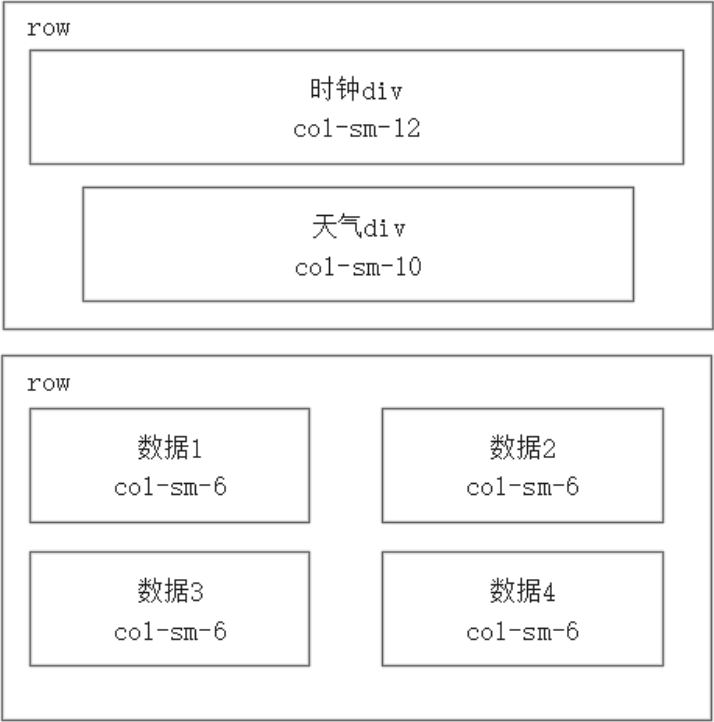


图 5-13 small 和 medium 布局

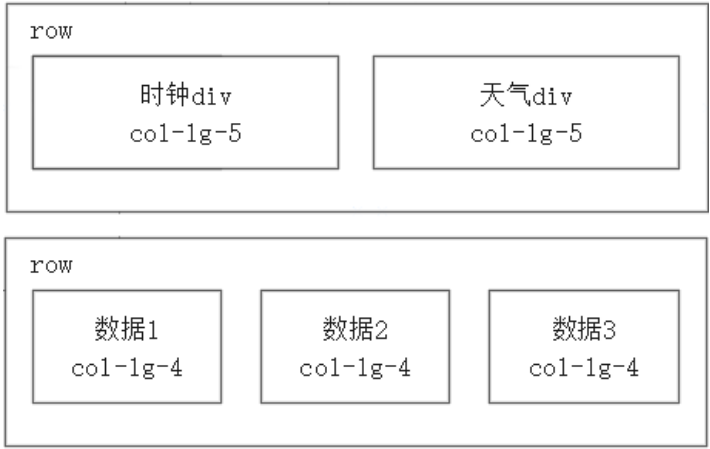


图 5-14 Large 布局

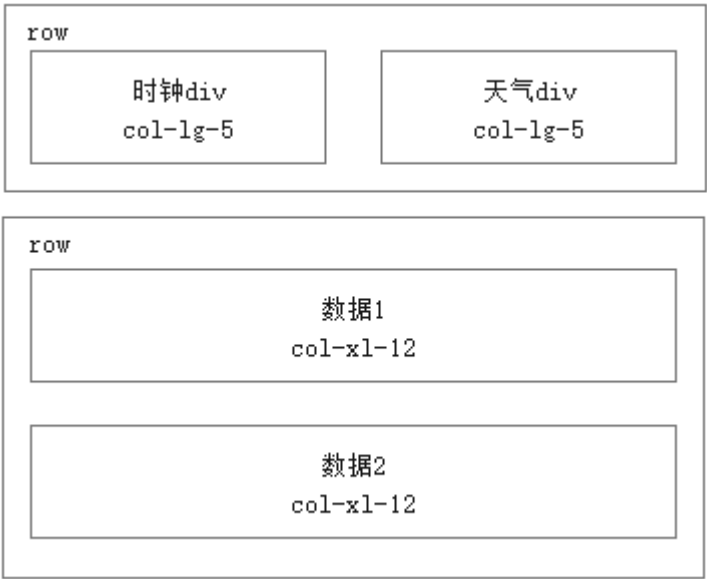


图 5-15 Extra large 布局

5. 2. 4 Vue 实例

控制端创建了 5 个 Vue 实例，分别控制导航栏、状态概览、情景设置、系统设置、设备管理这四个 section，如表 5-4 所示。

表 5-4 Vue 实例

实例名	methods	methods 注释	mounted	watch
navData	-	-	-	控制 section 显示
pageOverviewData	zeroPadding	时间格式化	设置自动更新时间、天气、传感器数据	-
	updateTime	更新时间		
	getData	获取传感器数据		
	getWeather	获取天气		

sceneData	enableEmail	开启邮件通知	-	-
	disableEmail	关闭邮件通知		
	enableIO	开启外围设备		
		控制		
sysSettingData	disableIO	关闭外围设备		
		控制		
	getData	获取树莓派信	设置自动更新	-
		息	树莓派信息	
	getIP	获取 IP		
	reloadPage	刷新页面		
	reload	重启程序		
	reboot	重启树莓派		
	poweroff	关闭树莓派		
	wolAdd	增加 wol 数据		
	wolDel	删除 wol 数据		
	wolUpdate	更新 wol 数据		
	wolSelect	查询 wol 数据		
	wolRun	唤醒计算机		

5.2.5 传感器数据显示的实现

前端数据的显示通过 v-for 遍历 ajax 获取到的数据并自动生成 HTML 代码，每次生成两个关联的 div，可以做到增加数个传感器的数据但不修改前端，其流程图如图 5-16 所示：其中 tid、tids 为 Bootstrap 中 modal 组件的参数，将遍历生成的数据 div 和说明 div 关联起来，实现点击数据 div 就能弹窗显示其说明信息的功能，这两个参数通过 v-bind 绑定至生成的 div 上。数据每隔一秒刷新一次。

PHP 端通过 socket 连接 Python 端的数据推送线程得到数据，当传入参数 \$_GET['mode'] 为 data 时，程序将获取标准数据，当 \$_GET['mode'] 为 status

时，PHP 将连接 SQLite 数据库，根据状态查询数据库中的建议操作，并返回状态和建议值。数据获取成功则返回数据，获取失败则返回-1。

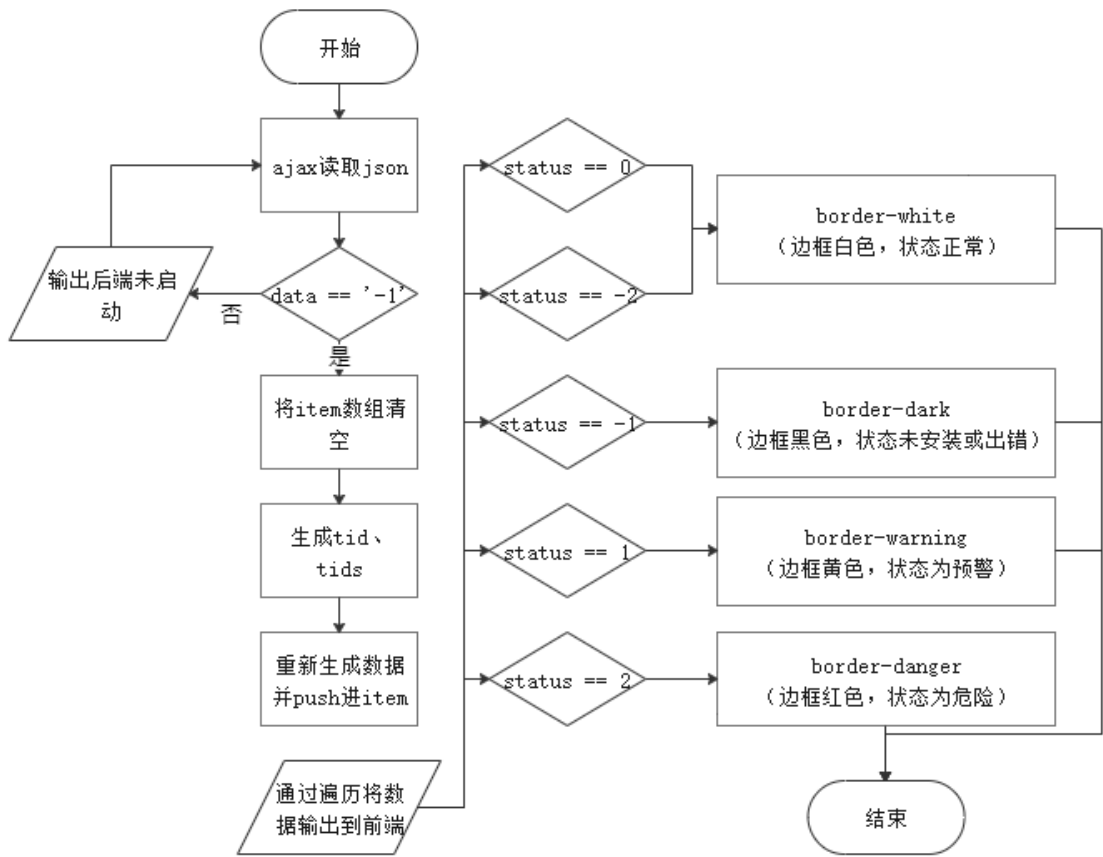


图 5-16 前端显示数据流程

5.2.6 树莓派状态获取功能的实现

树莓派状态获取功能主要由 PHP 端实现，通过调用 Linux 系统工具获取树莓派的 CPU 温度、内存使用情况、系统运行时间，处理完后将数据格式化为 Json 序列后通过 ajax 输出给前端。用到的工具如表 5-5 所示。

表 5-5 获取树莓派状态用到的系统工具

命令名称	作用
cat	用于将文件打印到标准输出设备上
awk	文本处理工具
free	用于查看系统内存信息

CPU 温度获取: raspbian 系统中, `/sys/class/thermal/thermal_zone0/temp` 文件记录了 CPU 当前的温度。使用 `cat` 读取该文件, 得到一个 5 位的十进制数据, 将此数据除以 1000, 才能得到树莓派的实时温度。这里使用 `awk '{print $1/1000}'` 命令, 将数据中的第一个变量除以 1000 并打印, 就得到了树莓派的实时温度。

实时内存使用情况获取: `free` 命令返回了物理内存、虚拟内存的总量、使用量、未使用量等信息。通过 `awk '/Mem/'` 能获取到物理内存使用情况的数据 (Mem 行), 再通过 `awk '{print $3/$2*100}'` 命令, 将 Mem 行中的第三个数据 (used) 除以第二个数据 (total) 并打印, 就能得到内存已使用量的百分比。

系统运行时间获取: `/proc/uptime` 文件记录了系统的启动时间和负载效率, 如图所示, 第一列为系统启动的时间 (以秒为单位), 第二列为系统空闲时间 (以秒为单位), 只需要第一个数据, 并通过 `awk` 加以处理, 就可以得到系统启动的时间。

5.2.7 本地 IP 获取功能的实现

由于 DHCP 服务器的存在, IP 不是固定不变的, 此处使用 PHP 的 `shell_exec` 函数运行 Linux 中 `iproute2` 套件的 `ip` 工具来获取本机的全部 ip。命令为“`ip -4 -j a`”, `-4` 为显示 IPv4 地址, `-j` 则为返回 json 数据。运行该命令后会得到一串很长的 json 数据。通过 `json_decode` 将其转化为 PHP 数组后, 使用 `foreach` 遍历过滤掉 169 开头的无效 ip, 最后将得到数据转化为 json 数据, 传递给前端。管理员可以通过该 IP 访问控制端。

5.2.8 计算机远程唤醒功能的实现

计算机远程唤醒功能使用 Wake on LAN 功能实现, 前端通过输入计算机名和 MAC 将信息记录进数据库, 然后显示在控制端界面上, 点击开机时, PHP 将调用 `wol` 程序向指定 MAC 地址发送一个魔术包, 完成远程开机动作。

5.2.9 控制端公网访问及权限认证

公网访问时, 有公网 IP 则使用端口映射直接访问, 无公网 IP 则使用 `frp` 内

网穿透通过外置服务器代理访问控制端。权限认证方面，由于单用户家用产品，未做用户权限的处理，通过 apache 自带的用户密码认证完成管理员认定操作。

5.3 集成板软件设计

集成板软件流程图如图 5-17 所示：Arduino 将按顺序读取甲醛传感器、温湿度传感器、多像素气体传感器、光线传感器的值，然后将其打包成 Json 数据并通过串口发送出去。

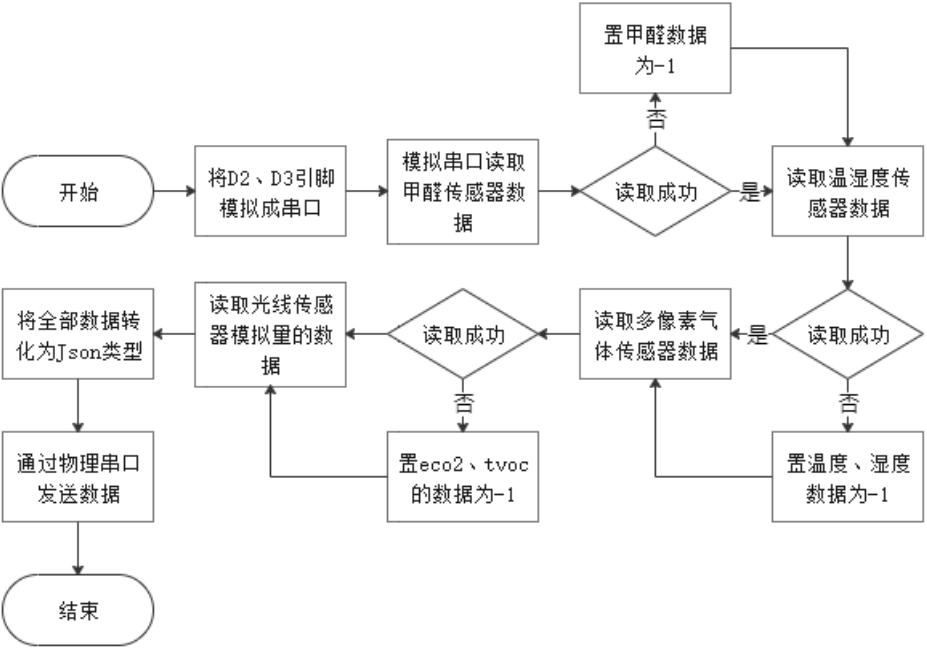


图 5-17 传感器集成板软件流程图

传感器集成板通过串口完成数据的发送，传输协议如表 5-6 所示。发送的数据为 Json 类型, 能够很方便的被各类程序所读取，其内容如表 5-7 所示。

表 5-6 集成板传输协议

键名	含义
波特率	115200
发送数据	Json 字符串

表 5-7 集成板发送数据的内容

键名	含义	单位
temperature	温度	℃
humditiy	湿度	%RH
eco2	二氧化碳等排量	ppm
formaldehyde	甲醛浓度	SGP30 传感器驱动
tvoc	总挥发性有机物浓度	ppb
light	光线强度（越暗越高）	-

5.4 数据库设计

数据库的设计比较简单, 使用 SQLite3 数据库。SQLite 数据库并没有和其他数据库一样强大的功能, 很多字段设置后就不能修改, 但好处在于不用搭建复杂的环境。数据库中定义了两个表, 一个是 sensor 表, 另一个是 wol 表。

sensor 表用于存储传感器相关数据, 如表 5-6 所示: uid 定义了传感器的 ID, 前六条 uid 为传感器集成板固定占用; name 一般用程序处理时对数据的鉴别, 常用于 json 键于; lowVar、highVar 用于判断传感器数据安全; comment 记录了数据的说明, 用于前端显示; suggest 用于生成 Email 的 HTML 模板, 生成对应的建议。

表 5-6 sensor 表结构

字段名	数据类型	Length	非空	自动递增	缺省	描述
uid	INTEGER	-1	true	false	-	传感器唯一 ID
name	TEXT	-1	false	false	-	传感器英文名字段
lowVar	INTEGER	-1	false	false	-	低值
highVar	INTEGER	-1	false	false	-	高值
unit	TEXT	-1	false	false	-	数据单位
method	INTEGER	-1	true	false	0	安全警报模式
alarmMode	INTEGER	-1	true	false	0	提醒模式
comment	TEXT	-1	false	false	-	传感器数据解释说明

续表 5-6

cname	TEXT	-1	false	false	-	传感器中文名
useEmail	INTEGER	-1	true	false	1	使用 Email
useBuz	INTEGER	-1	true	false	0	使用蜂鸣器
suggest	TEXT	-1	false	false	-	危险时的建议做法
useWatcher	INTEGER	-1	true	false	1	使用安全警报线程
recVal	INTEGER	-1	true	false	5	从危险恢复安全的过渡值

wol 表用于存储远程开机的设备信息，如表 5-7 所示：通过该表记录计算机名称和 MAC 的对应关系。

表 5-7 wol 表结构

字段名	数据类型	Length	非空	自动递增	缺省	描述
uid	INTEGER	-1	true	false	-	wol 唯一 id
name	TEXT	-1	false	false	-	计算机名
mac	TEXT	-1	true	false	-	计算机 MAC 地址

6. 系统测试

系统总体测试情况如表 6-1 所示：

表 6-1 系统测试结果及处理

测试内容	测试结果	处理方式
长时间运行稳定性测试	系统稳定，未见明显异常	—
传感器串口数据获取	使用 USB 连接时正常，控制端 2 秒能更新一次数据；使用树莓派 GPIO 连接时，接收到的数据错乱	为避免增加连接线，依旧使用 GPIO 连接，增加 try except 语句解决，但受制于接收失败的次数，控制端更新数据的时间存在较大的波动
光线传感器	数值变动正常	—
SGP30 传感器测试	使用常规电路图焊接时，SGP30 能读到 I ² C 地址但无数 据，经过查询，SGP30 对焊接设备有要求	电路图不变，更换已焊接好的 SGP30 模块
W-Z-S 传感器	未见明显异常	—
DHT11 传感器	误差 2 度，属于误差范围内的最大误差	忽略，可更换模块解决
蜂鸣器	异常，蜂鸣器无声，但电压测试正常，考虑模块损坏	放弃蜂鸣器
时钟模块	未见明显异常	—
PIR 传感器控制显示屏	成功控制显示屏，未见明显异常	—
集成板总体测试	线路选择开关中 SGP30 连接树莓派 GPIO 的开关网络错误	忽略，使用 Arduino 连接传感器，仅修改原理图和 PCB 图，无需再次打板

续表 6-1

邮件发送测试	成功接收到传感器各类预 警、恢复安全等邮件，但部 分数值波动较大的数值发送 邮件比较频繁	关闭发特定数据的邮件发送 功能
传感器状态改变测试	未见明显异常	—
控制端功能测试	未见明显异常	—
控制端树莓派信息获取测试	获取的信息正确	—
有效 IP 获取测试	获取的有效 IP 正确	—
远程计算机唤醒测试	目标计算机成功唤醒	—

7. 总 结

本次设计使用了多种技术，对软硬件都有涉及，完成了传感器集成板，避免了各种传感器之间的复杂连线，实现了对家庭环境的监控。困难的地方在于对不同语言、不同设备之间的连通。系统端代码可靠性高，扩展性强，出错处理做的到位。本设计有以下优点：

- (1) 设计上：交互界面简洁，交互方便，能够很好的融入生活环境。
- (2) 实用性上：在家庭环境相对安全的情况下，本设计的其他功能可以使项目依旧保持实用性。
- (3) 语言上：选择 Json 作为程序与程序、软件与硬件直接桥梁，对后续开发比较友好。
- (4) 扩展性上：使用完整的 Linux 系统，在实现项目本身功能的前提下还可以扩展其他服务功能。

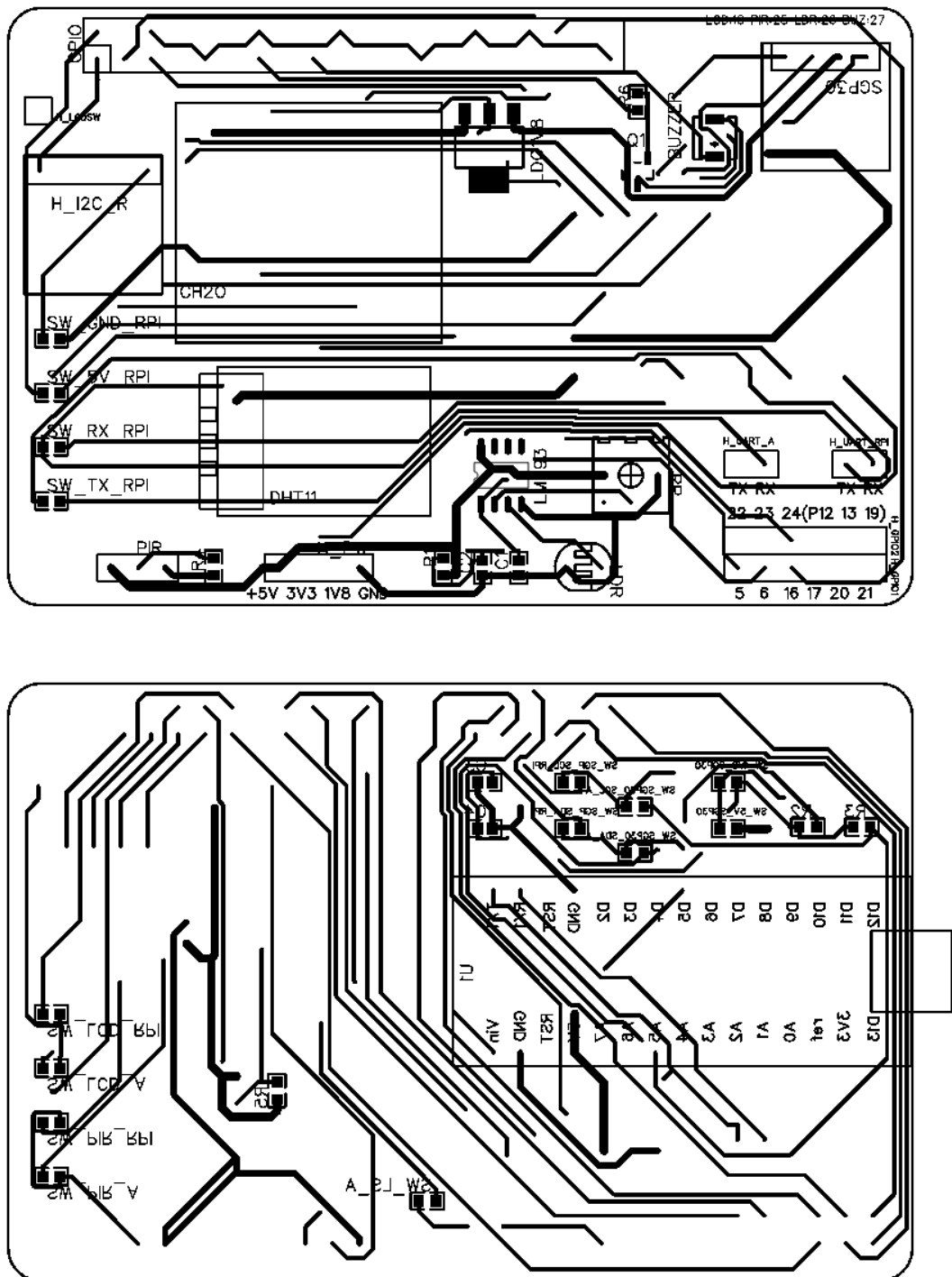
本项目还有很多值得开发的功能和可以解决的问题，但是因为时间、经验、财力等方面的问题导致其无法完成。而且控制端的设计理念存在着一些问题，追求极致的前后端分离导致了 ajax 的高度复用，控制端扩展性不强，无法完成更高级的处理。为支持开源生态，本次设计源码已在 GitHub 上开源，希望对其他人有所帮助：<https://github.com/leaplu/rpi-envsys/>

参考文献

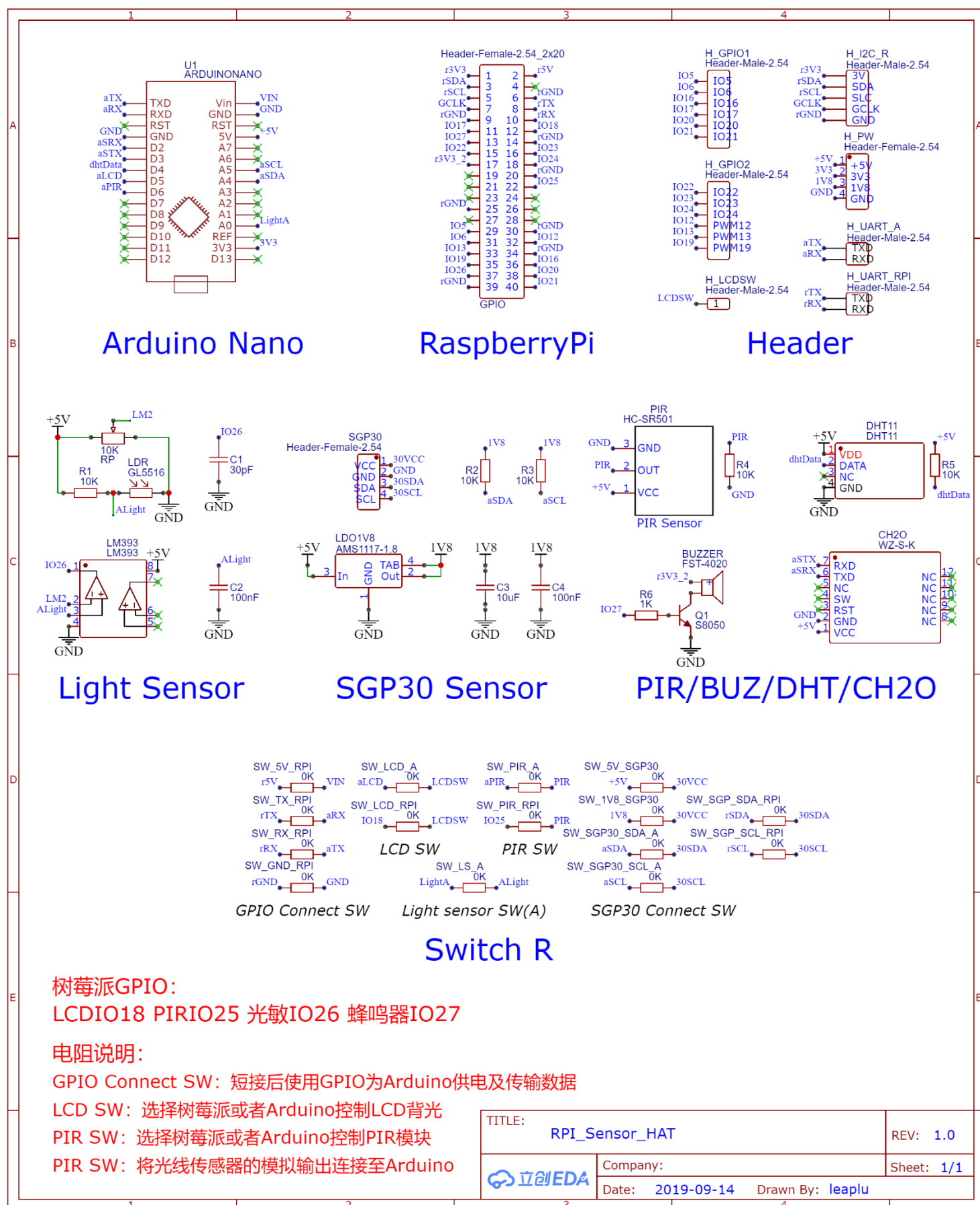
- [1] Embedded Linux Wiki. RPi USB Wi-Fi Adapters[Z/OL]. 2019:
https://elinux.org/RPi_USB_Wi-Fi_Adapters.
- [2] Python Software Foundation. Python 3.8.2 documentation[Z/OL]. 2020:
<https://docs.python.org/3/>.
- [3] Raspberry Pi Foundation. Raspberry Pi Documentation[Z/OL]. 2020:
<https://www.raspberrypi.org/documentation/>.
- [4] 勃鲁姆/布莱斯纳罕. 树莓派Python编程入门与实战[M]. 北京:人民邮电出版社. 2016. 1-460.
- [5] 蒙克. Raspberry Pi 开发实战[M]. 北京:机械工业出版社. 2015. 1-365.
- [6] 赵英杰. 完美图解物联网 IoT 实操 ESP8266 Arduino, Cordova 物联网移动 App, JavaScript 微控制器编程[M]. 北京:电子工业出版社. 2018. 1-376.
- [7] 谢平. PCB 设计与加工[M]. 北京:北京理工大学出版社. 2017. 135-201.
- [8] 蒙克. 电子创客案例手册 Arduino 和 Raspberry Pi 电子制作实战[M]. 北京:清华大学出版社. 2018. 1-373.
- [9] 陈锐. 基于树莓派和 Arduino 智能家居控制系统研究和设计[D]. 天津:天津职业技术师范大学. 2018. 13-54
- [10] 张欣. 基于树莓派的智能家居控制系统设计研究[D]. 黑龙江:东北石油大学. 2018. 13-36

附录

附录 1：传感器集成板 PCB 图



附录 2：传感器集成板电路原理图



附录 3：传感器集成板 BOM 表（部分）

元件名称	编号	数量	制造商料号	制造商
DHT11 传感器	DHT11	1	DHT11	Aosong
FST-4020 蜂鸣器	BUZZER	1	FST-4020	FUET
ARDUINONANO	U1	1	—	—
10uF 电容	C3	1	CC0603KRX5R5BB106	YAGEO
1K 电阻	R6	1	RMC06031K5%N	Tyohm
WZ-S-K 传感器	CH20	1	—	—
AMS1117-1.8	LD01V8	1	AMS1117-1.8	AMS
S8050	Q1	1	S8050	Hottech
GL5516 光敏电阻	LDR	1	GL5516	Senba
10K 电阻	R2, R3, R5, R1, R4	5	0603WAF1002T5E	UNI-ROYAL
30pF 电容	C1	1	CC0603GRNP09BN300	YAGEO
LM393 电压比较器	LM393	1	LM393	PUOLOP
100nF 电容	C4, C2	2	CC0603KRX7R0BB104	YAGEO
10K 电阻	RP	1	3362P-1-103	ReliaPro
HC-SR501 传感器	PIR	1	—	—

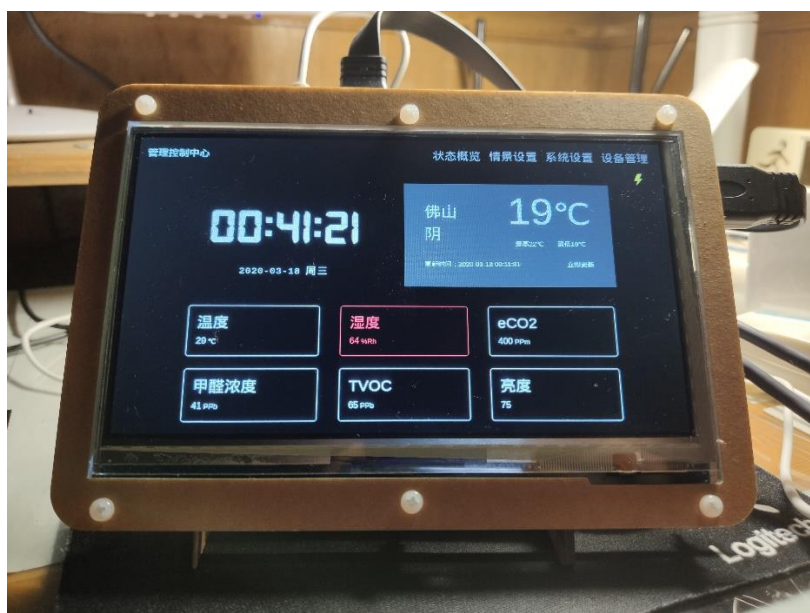
附录 4：树莓派 2B 实物图



附录 5：传感器集成板成品图



附录 6：项目成品图



附录 7: CPU 温度获取方式

```
pi@raspberrypi:~$ cat /sys/class/thermal/thermal_zone0/temp
44388
pi@raspberrypi:~$ cat /sys/class/thermal/thermal_zone0/temp | awk '{print $1/1000}'
44.926
```

附录 8: 内存使用率获取方式

```
pi@raspberrypi:~$ free
              total        used         free       shared  buff/cache   available
Mem:           765600      283352         56388         37604        425860        386732
Swap:          102396         31488         70908
pi@raspberrypi:~$ free | awk '/Mem/' | awk '{print $3/$2*100}'
37.1839
```

附录 9: 系统开机时间获取方式

```
pi@raspberrypi:~$ cat /proc/uptime
35552.57 113438.83
pi@raspberrypi:~$ cat /proc/uptime | awk -F. '{run_days=$1 / 86400;run_hour=($1 % 86400)/3600;run_minute=($1 % 3600)/60;run_second=$1 % 60;printf("%d天%d时%d分%d秒",run_days,run_hour,run_minute,run_second)}'
0天9时52分35秒pi@raspberrypi:~$ _
```

附录 10: ip -4 -j a 返回数据

```
pi@raspberrypi:~$ ip -4 -j a
[{"ifindex":1,"ifname":"lo","flags":["LOOPBACK","UP","LOWER_UP"],"mtu":65536,"qdisc":"noqueue","operstate":"UNKNOWN","group":"default","txqlen":1000,"addr_info":[{"family":"inet","local":"127.0.0.1","prefixlen":8,"scope":"host","label":"lo","valid_life_time":4294967295,"preferred_life_time":4294967295}],{"ifindex":3,"ifname":"wlan0","flags":["BROADCAST","MULTICAST","UP","LOWER_UP"],"mtu":1500,"qdisc":"mq","operstate":"UP","group":"default","txqlen":1000,"addr_info":[{"family":"inet","local":"192.168.2.105","prefixlen":24,"broadcast":"192.168.2.255","scope":"global","dynamic":true,"noprfixroute":true,"label":"wlan0","valid_life_time":4731,"preferred_life_time":3831}]]]
```

附录 11：IP 和系统信息显示结果



附录 12：ip-api 返回数据

```
{"status": "success", "country": "China", "countryCode": "CN", "region": "GD", "regionName": "Guangdong", "city": "Foshan", "zip": "", "lat": 23.0287, "lon": 113.1341, "timezone": "Asia/Shanghai", "isp": "China Unicom Guangdong Province Network", "org": "", "as": "AS17816 China Unicom IP network Chinal69 Guangdong province", "query": "27.36.132.166"}
```

附录 13：天气 API 返回数据

```
{"cityid": "101010100", "date": "2020-03-24", "week": "\u661f\u671f\u4e8c", "update_time": "2020-03-24 12:50:04", "city": "\u5317\u4eac", "cityEn": "beijing", "country": "\u4e2d\u56fd", "countryEn": "China", "wea": "\u591a\u4e91", "wea_img": "yun", "tem": "16", "tem1": "17", "tem2": "5", "win": "\u4e1c\u5357\u98ce", "win_speed": "2\u7ea7", "win_meter": "\u5c0f\u4e8e12km/h", "humidity": "27%", "visibility": "13.27km", "pressure": "1012", "air": "54", "air_pm25": "54", "air_level": "\u826f", "air_tips": "\u7a7a\u6c14\u597d\u7279\u522b\u6d4f\u611f\u7684\u4eba\u7fa4\u4ee5\u5916\u53ef\u5bf9\u516c\u4f17\u6c14\u6709\u5371\u5bb3\u5f01", "alarm": {"alarm_type": "", "alarm_level": "", "alarm_content": ""}}
```


附录 14：集成板数据获取接口返回数据

```
{ "temperature": { "cname": "\u0629\u05ea6", "data": 32, "status": 0, "unit": "\u02103", "comment": null, "humidity": { "cname": "\u06e7f\u05ea6", "data": 59, "status": 0, "unit": "%Rh", "comment": null, "eco2": { "cname": "eCO2", "data": 467, "status": 0, "unit": "PPm", "comment": "\u04eba\u04f53\u05bf9\u07a7a\u06c14\u04e2d\u04e8c\u06c27\u05316\u07b3\u07684\u0589e\u0957f\u0975e\u05e38\u0654f\u0611f\u01000 PPM)\u065f6\u0ff0c\u04eba\u04eec\u04f1a\u0611f\u05230\u06c89\u0957f\u0ff0c\u06ce8\u0610f\u0529b\u05f00\u059cb\u04e0d\u096c6 PPM\u0ff0c\u0800c\u06211\u04eec\u08fde\u07eed\u07761\u089c98\u04e2a\u05c0f\u065f6\u0ff0c\u065e9\u04e0a\u08d77\u05e8a\u065ff 2000 PPM\u065f6\u0ff0c\u04eba\u04eec\u04f1a\u0611f\u05230\u06c14\u05598\u03001\u05934\u075db\u03001\u07729\u06655\u03002\u04e2 PPM.\u0529e\u0516c\u05ba4\u07684\u07a7a\u06c14\u04e2dCO2\u06d53\u05ea6\u08fbe\u052302000PPM\u065f6\u0ff0c\u05458\u05de5\u0 formaldehyde": { "cname": "\u07532\u0919b\u06d53\u05ea6", "data": 6, "status": 0, "unit": "PPb", "comment": "\u07532\u0919b\u06709\u04e24\u04e2a\u06807\u051c6\u0ff0c\u0300a\u0300a\u0611\u07528\u05efa\u07b51\u05ba4\u05185\u073af\u05883 m\u00b3\u0ff08\u07ea6\u04e3a66PPb\u0ff09\u0ff0c\u0300a\u05ba4\u05185\u07a7a\u06c14\u08d28\u091cf\u06807\u051c6\u0300b\u05 m\u00b3\u0ff08\u07ea6\u04e3a82PPb\u0ff09\u0ff0c\u04e00\u065e6\u08d85\u08fc70.08mg/m\u00b3\u05219\u089c6\u04e3a\u08d8 0.07mg/m\u00b3\u065f6\u0ff0c\u053ef\u080fd\u04f1a\u04f7f\u04eba\u04f53\u09020\u06210\u08f7b\u05fae\u06c14\u05598\u03002\u0 tvoc": { "cname": "TVOC", "data": 87, "status": 0, "unit": "PPb", "comment": "\u056fd\u05bb6\u06807\u051c6\u05ba4\u051850.5mg/m3 = 833 * 0.5 = 441.5ppb"} , "light": { "cname": "\u04eae\u05ea6", "data": 389, "status": -2, "unit": null, "comment": null}}}
```

附录 15：天气数据获取接口返回数据

```
{ "cityid": "101280800", "date": "2020-03-29", "week": "\u0661\u0671\u065e5", "update_time": "2020-03-29 22:05:48", "city": "\u045b\u045c71", "cityEn": "foshan", "country": "\u0442d\u056d", "countryEn": "China", "wea": "\u09634", "wea_img": "yin", "tem": "16", "tem1": "18", "tem2": "17", "win": "\u0531\u098ce", "win_speed": "2\u07ea7", "win_meter": "\u05c0f\u04e8e12km/h", "humidity": "97%", "visibility": "2.99km", "pressure": "1008", "air": "55", "air_pm25": "55", "air_level": "\u0826f", "air_tips": "\u07a7a\u06c14\u0597d\u060c\u053ef\u04ee5\u05916\u051fa\u06d3b\u052a8\u060c\u09664\u06781\u05c11\u06570\u05b9f\u06c61\u067d3", "alarm": { "alarm_type": "", "alarm_level": "", "alarm_content": "" } }
```

附录 16: 传感器集成板发送的数据

```
{ "temperature":33,"humditiy":59,"eco2":400,"formaldehyde":95,"tvoc":0,"light":705}
{ "temperature":33,"humditiy":59,"eco2":400,"formaldehyde":90,"tvoc":0,"light":707}
{ "temperature":33,"humditiy":59,"eco2":400,"formaldehyde":86,"tvoc":0,"light":709}
```


致 谢

感谢我师兄为我介绍工具和平台，为我完成本次毕业设计打下了基础

感谢我隔壁宿舍的李澳鑫同学为我送宵夜，解决温饱问题

感谢我的舍友在我需要的时候为我提供帮助

感谢我指导老师对我的细心教导和严格要求

感谢我自己对自己的大力支持