

韶关学院

课 程 设 计

课 程：物联网应用开发实践

题 目：基于 Cortex-A8 的智能家居控制系统的设计与实现

学生姓名：陈俊杰

学 号：15115061002

二级学院：信息科学与工程学院

专 业：物联网工程

班 级：2015 级 01 班

指导教师姓名及职称：刘晓樑 讲师

起止时间：2018 年 9 月—— 2019 年 1 月

（教务处制）

1 智能家居控制系统的设计相关技术.....	1
1.1 ZigBee 无线传输技术.....	1
1.2 Qt 图形界面应用.....	1
1.3 Sqlite 数据库技术.....	2
1.4 安卓 APP 技术.....	3
2 系统设计总体方案.....	4
2.1 系统总体架构.....	4
2.2 各模块功能.....	4
2.2.1 数据采集模块.....	4
2.2.2 CC2530 模块.....	7
2.2.3 控制终端模块.....	9
2.2.4 服务器模块.....	10
2.2.5 移动端控制模块.....	10
3 系统软件设计与实现.....	10
3.1 CC2530 程序设计.....	10
3.2 控制终端程序设计.....	13
3.3 服务器程序设计.....	16
3.4 移动控制端设计.....	17
3.4.1 ESP8266 WIFI 模块控制网络的搭建.....	17
3.4.2 SIM800C 短信模块控制环境搭建.....	18
3.4.3 Android 移动端 APP 界面设计.....	20
3.4.4 移动控制功能的实现.....	22
3.4.4.1 WIFI 控制方式.....	22
3.4.4.2 短信控制方式.....	24
4 系统测试.....	26
4.1 ZigBee 模块组网测试.....	26
4.2 串口数据传输测试.....	27
4.3 服务器端数据存储测试.....	28
4.4 Android 手机 APP 测试.....	28
5 总结与展望.....	30
5.1 本文的主要成果.....	30
5.2 展望.....	31
参考文献.....	32

1 智能家居控制系统的设计相关技术

1.1 ZigBee 无线传输技术

ZigBee 技术是一种近距离、低复杂度、低功耗、低速率、低成本的双向无线通讯技术。主要用于距离短、功耗低且传输速率不高的各种电子设备之间进行数据传输以及典型的有周期性数据、间歇性数据和低反应时间数据传输的应用。

ZigBee 模块是一种物联网无线数据终端，利用 ZigBee 网络为用户提供无线数据传输功能。ZigBee 是一个由可多到 65535 个无线数传模块组成的一个无线数传网络平台，在整个网络范围内，每一个 ZigBee 网络数传模块之间可以相互通信，每个网络节点间的距离可以从标准的 75m 无限扩展。

ZigBee 已广泛应用于物联网产业链中的 M2M 行业，如智能电网、智能交通、智能家居、金融、移动 POS 终端、供应链自动化、工业自动化、智能建筑、消防、公共安全、环境保护、气象、数字化医疗、遥感勘测、农业、林业、水务、煤矿、石化等领域。

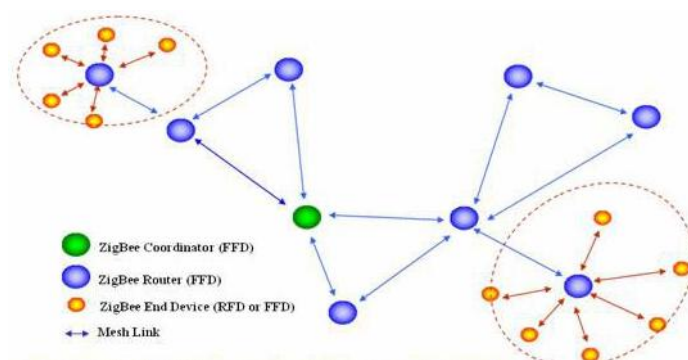


图 1.1 ZigBee 网络节点图

1.2 Qt 图形界面应用

Qt 是一个由 Qt Company 开发的跨平台 C++ 图形用户界面应用程序开发框架，它是一个面向对象的 C++ 架构，方便扩展，被普遍应用于 linux 平台、Windows 平台应用程序的开发上，全面支持 iOS、Android、WP，它提供给应用程序开发者建立艺术级的图形用户界面所需的所有功能。

Qt Creator 是由美国某企业研发的一个多方向的 C++ 语言图形应用程序的集成开发软件。它能够开发 GUI（图形用户界面）程序，还能开发服务器等其他非 GUI 的应用。本设计使用 Qt Creator 开发控制终端以及服务器用户交互界面。Qt Creator 开发软件界面如下：

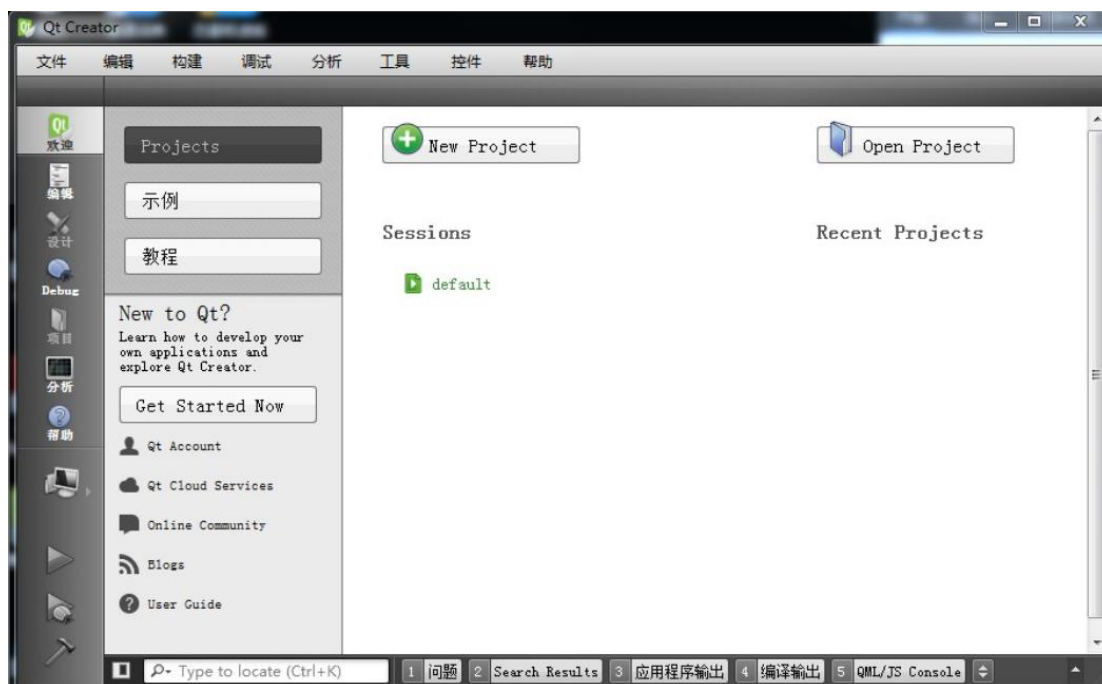


图 1.2 Qt Creator 开发软件界面图

1.3 Sqlite 数据库技术

本设计使用的数据库为 SQLite 数据库，用于存储 PC 服务器端接收到的网络数据。

SQLite 是一款轻型的数据库，是遵守 ACID 的关系型数据库管理系统，它包含在一个相对小的 C 库中。它是 D.RichardHipp 建立的公有领域项目。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百 K 的内存就够了。它能够支持 Windows/Linux/Unix 等等主流的操作系统，同时能够跟很多程序语言相结合，比如 Tcl、C#、PHP、Java 等，还有 ODBC 接口，同样比起 Mysql、PostgreSQL 这两款开源的世界著名数据库管理系统来讲，它的处理速度比他们都快。

1.4 安卓 APP 技术

Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。该平台由操作系统、中间件、用户界面和应用软件组成。开发人员可以利用 API 框架很方便地进行 Android 应用程序的开发。Android 应用程序框架包含了丰富的程序组件，可以用来构建应用程序，它包括列表（Lists），网格（Grids），文本框（Text boxes），按钮（Buttons），甚至可嵌入的 web 浏览器。

Android Studio 是谷歌推出一个 Android 集成开发工具，基于 IntelliJ IDEA。类似 Eclipse ADT，Android Studio 提供了集成的 Android 开发工具用于开发和调试。本设计受用 Android Studio 开发手机端 APP 程序，主要实现 WiFi 控制和短信远程控制功能。Android Studio 开发工具界面如下：

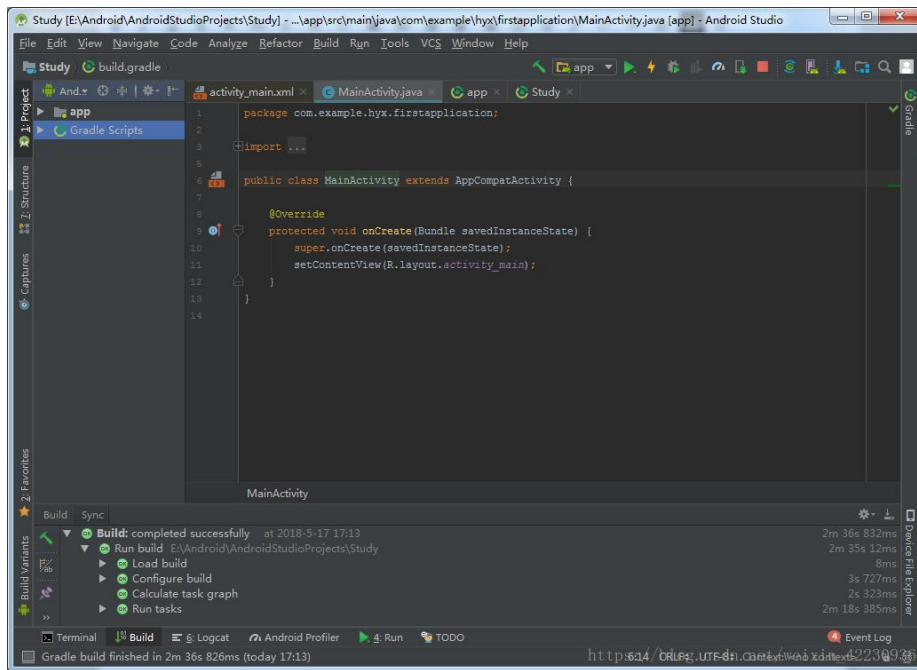


图 1.4 Android Studio 开发工具界面图

2 系统设计总体方案

2.1 系统总体架构

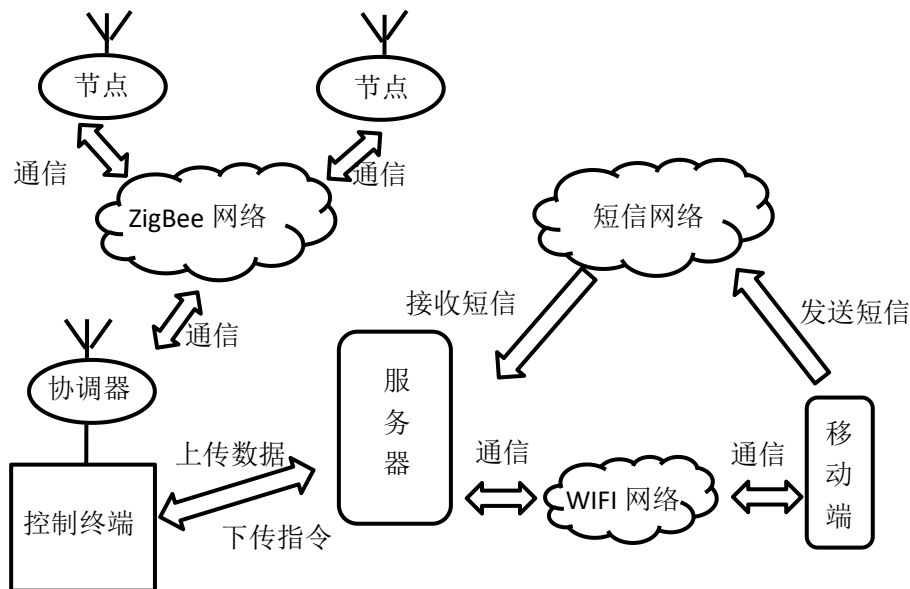


图 2.1 系统总体架构图

2.2 各模块功能

2.2.1 数据采集模块

DHT11 温湿度模块

本系统使用的 DHT11 温湿度模块是一款有已校准数字信号输出的温湿度复合传感器，该模块具有品质卓越、超快响应、抗干扰能力强、性价比极高、体积小、功耗极低等优点。相对于其它的温湿度传感器，DHT11 模块无需额外部件，可直接获得外界温湿度测量值，无需繁琐的计算和 AD 转换，测量结果更为精确，可用于需要获得精确温湿度值的场合中。

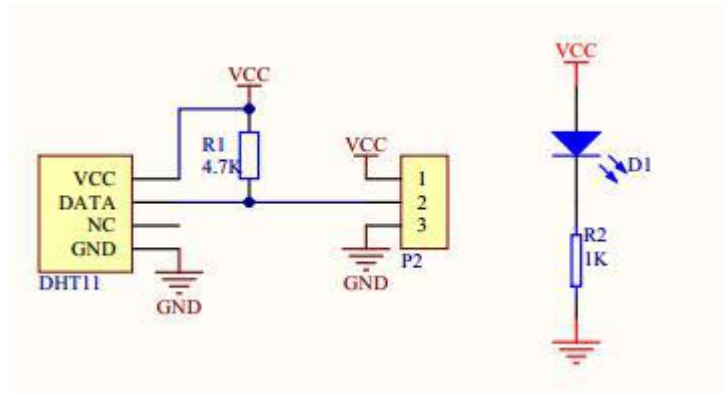


图 2. 2. 1. 1 DHT11 模块原理图

DHT11 温湿度模块实物图如下：

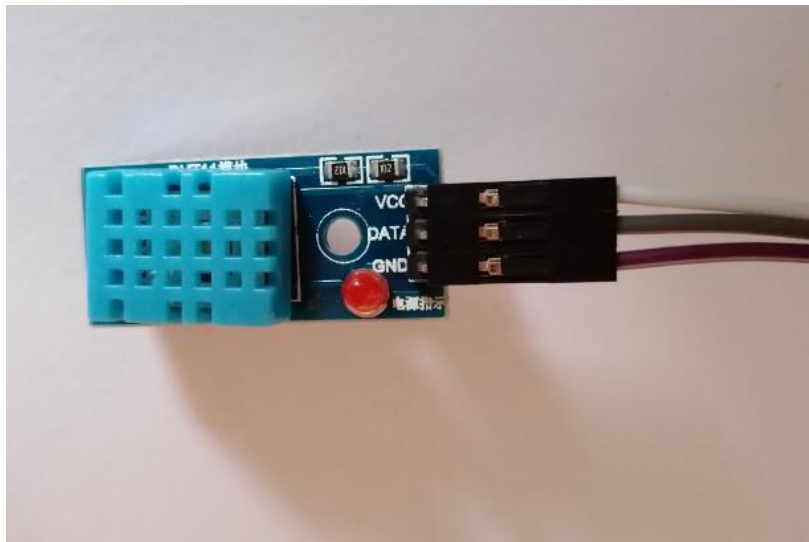


图 2. 2. 1. 2 DHT11 模块

BH1750FVI 光照传感模块

BH1750FVI 光照传感模块是一款以 BH1750FVI 芯片为核心的数字式光照传感器，传感器采用标准 NXP IIC 通信协议，内置 16bitAD 转换器，直接数字输出光照强度值。BH1750FVI 光照传感器具有接近于视觉灵敏度的分光特性，可对广泛的亮度进行 1 勒克斯的高精度测定，测量范围在 1-65535 勒克斯之间。此外，该传感器受红外线影响较小，可应用场合更为广泛。

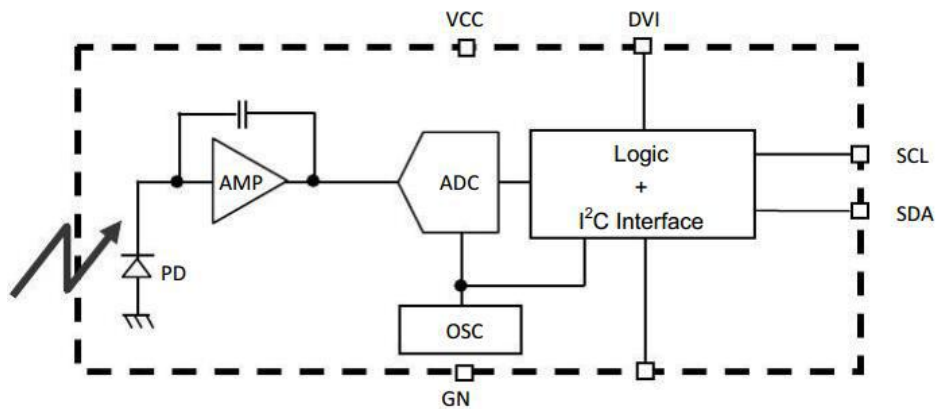


图 2. 2. 1. 3 BH1750FV 框图

BH1750FV 光照传感模块实物如下图：

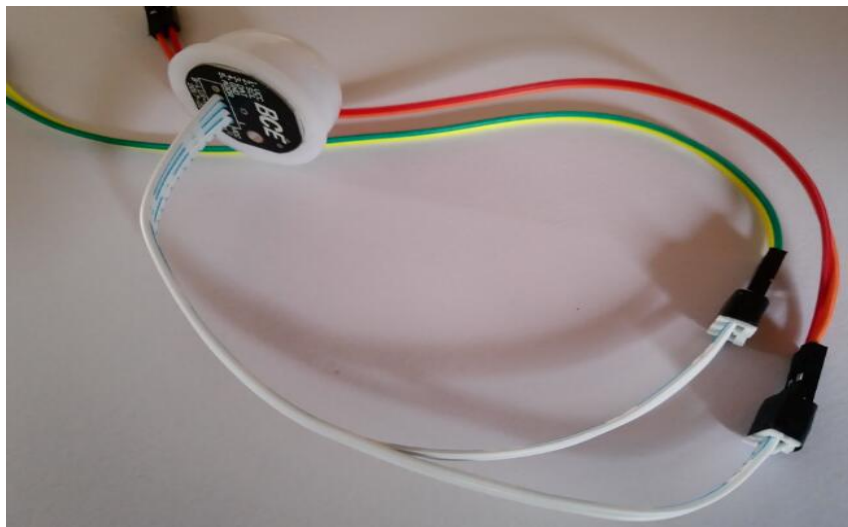


图 2. 2. 1. 4 BH1750FV 光照传感模块

MQ-2 烟雾气敏传感器

MQ-2 烟雾气敏传感器是一款使用二氧化锡 (SnO_2) 半导体作为气敏材料，属于表面离子式 N 型半导体的传感器。当传感器所在位置的烟雾的浓度越大，其导电率就越大，输出的电阻值就越低，输出的模拟信号就越大。

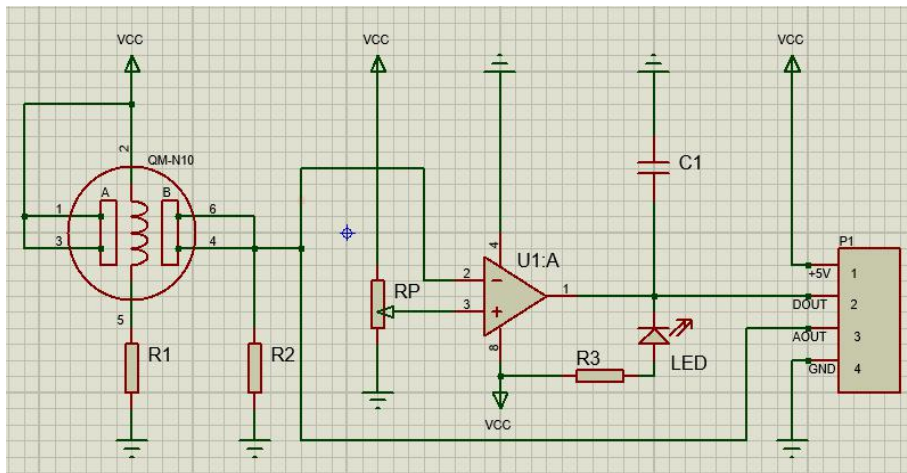


图 2. 2. 1. 5 MQ-2 烟雾气敏传感器原理图

MQ-2 烟雾气敏传感器对液化气、氢气、丙烷等可燃性气体的灵敏度很高，并且具有良好的抗干扰性，可以使用在多种场合，例如使用在有以天然气为燃料的热水器的浴室中，可防止可燃性气体泄漏导致气体中毒等事故的发生；使用在厨房可通过检测油烟的浓度是否自动开启抽油烟机，保证厨房空气的清洁。

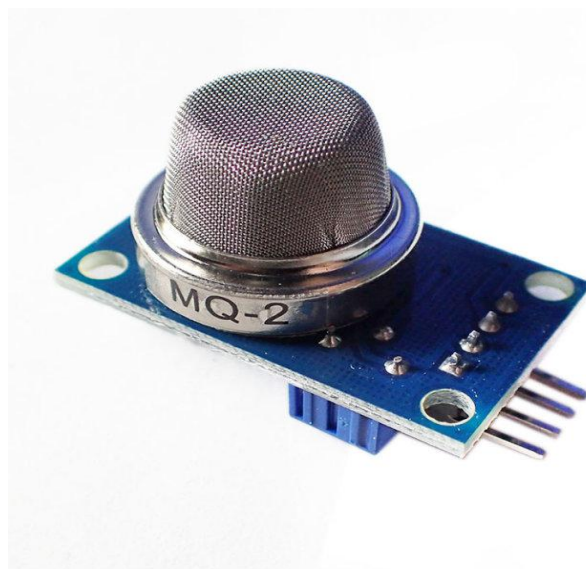


图 2. 2. 1. 6 MQ-2 烟雾气敏传感器

2.2.2 CC2530 模块

CC2530 是用于 2.4-GHz IEEE 802.15.4、ZigBee 和 RF4CE 应用的一个真正的片上系统（SoC）解决方案。它能够以非常低的总的材料成本建立强大的网络节点。CC2530 结合了领先的 RF 收发器的优良性能，业界标准的增强型 8051 CPU，系统内可编程闪存，8-KB RAM 和许多其它强大的功能。本系统使用德州仪器公

2.2.3 控制终端模块



图 2.2.3.1 GEC210 开发板

本系统 GEC210 开发板作为控制终端模块，GEC210 是一款高性能的 Cortex-A8 核心板，它采用三星 S5PV210 作为主处理器，运行主频可高达 1GHz。GEC210 开发板可流畅运行 Android、Linux 和 WinCE6 等高级操作系统。

GEC-210 具有两种 USB 接口，一种是 USB Host (2.0) 接口，共 2 个，它和普通 PC 的 USB 接口是一样的，可以接 USB 摄像头、USB 键盘、USB 鼠标、优盘等常见的 USB 外设；另外一种 miniUSB (2.0)，主要用于 Android 系统下的 ADB 功能，用于软件安装和程序调试。本文的智能家居控制系统中 ZigBee 协调器模块通过 USB Host (2.0) 接口于控制终端相连。

此外除了 TV-OUT 输出，S5PV210 支持 I2S/PCM/AC97 等音频接口，GEC-210 开发板采用的是 I2S0 接口，它外接了 WM8960 作为 CODEC 解码芯片，可支持 HDMI 音视频同步输出。本文的语音报警功能使用了音频解码和音频输出进行语音播报。I2S0 接口如下图所示：

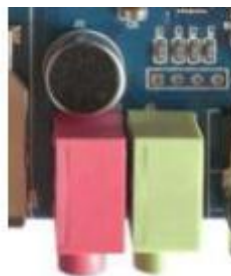


图 2.2.3.2 I2S0 接口

2.2.4 服务器模块

本智能家居控制系统使用 PC 作为服务器程序的运行环境，运行 Qt 图形界面程序，连接 Sqlite 数据库作为服务器端，采用网线连接方式与控制终端建立 TCP 连接，接收并储存来自控制终端的数据信息。除此之外，在 Windows 环境下对 Qt 服务器程序进行开发、更新和维护都比在 Linux 系统更为简单。

2.2.5 移动端控制模块

本设计选择 Android 手机作为移动控制终端，使用 Java 语言开发移动端 APP，用户可以通过 WIFI 网络或者发送短信对家居的电器进行控制。

3 系统软件设计与实现

3.1 CC2530 程序设计

ZStack 是适用于 TI 公司推出的 CC2530 芯片的一款协议栈，ZStack 协议栈包括物理层（PHY）、ZigBee 设备对象层（ZDO）、介质接入控制子层（MAC）、网络层（NWK）、应用支持层（APS）和应用程序框架（AF）。开发者通过协议栈开源的应用支持层（APS）与 ZigBee 设备对象层（ZDO）相应的 API 接口进行开发。



图 3.1.1 ZigBee 协议栈框架

CC2530 数据采集模块协议栈软件流程图如下：

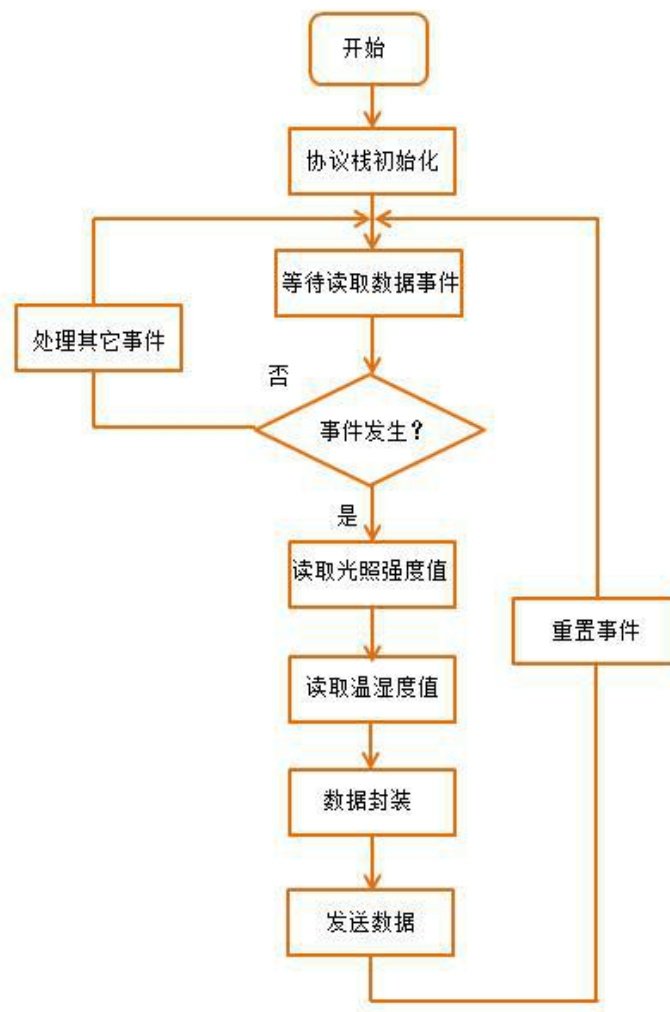


图 3.1.2 CC2530 协议栈流程图

CC2530 协调器模块协议栈软件流程图如下：

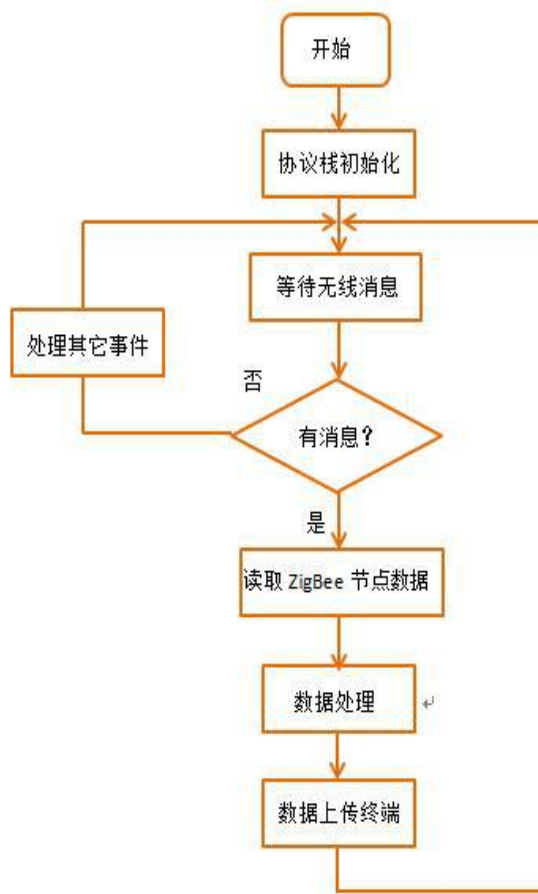


图 3.1.3 CC2530 协调器软件流程图

CC2530 数据采集模块上电后进行协议栈的初始化，并设置读取数据事件，与协调器组网后每隔固定时间触发该事件，获取到室内湿度、温度和光照强度值后将数据进行封装，然后通过单播形式将数据发往协调器。协调器与数据采集模块组网后进入低功耗模式，每当接收到节点发送的消息时退出低功耗模式，将数据进行拆分，分别给加上温度、湿度和光照强度值加上标记，然后通过串口上传至控制终端。

下图为 IAR8.0 环境下 CC2530 数据采集模块协议栈软件设计：

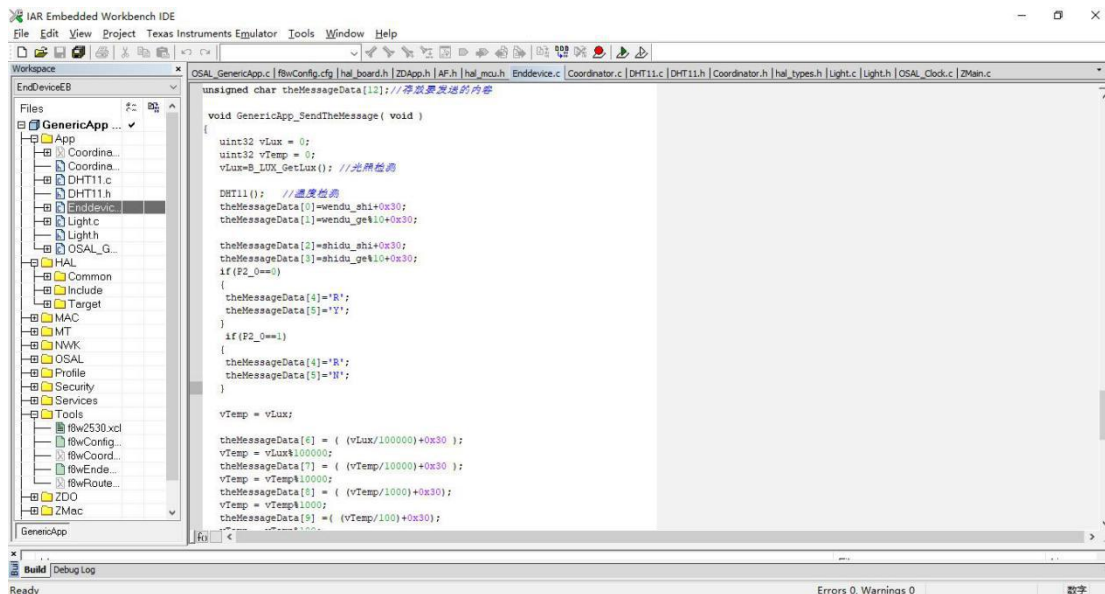


图 3.1.4 CC2530 协议栈

3.2 控制终端程序设计

控制终端 Qt 程序由主界面、TCP 连接界面、灯光控制界面、风扇控制界面、温湿度监测界面、安全监测界面和设置界面组成。下面介绍主界面、TCP 连接界面和设置界面，其它界面不作介绍：



图 3.2.1 主界面

主界面是 Qt 程序运行后的第一个界面，主要由 6 个 QPushButton 组成，点击即可切换到对应的界面。



图 3.2.2 TCP 连接界面

TCP 连接界面主要显示控制终端与 PC 端服务器的 TCP 连接的状态，当用户想设置控制终端不再上传数据到服务器，用户可以主动断开 TCP 连接。TCP 连接中，控制终端作为 Server，服务器作为 Client，创建连接需由服务器主动连接。



图 3.2.3 设置界面

设置界面主要提供用户设置 Qt 程序的工作模式、声音开关以及温度、湿度和光照强度的阈值。在自动模式下，当温度、湿度、光照强度超过阈值，或者检测到有烟雾，程序会自动发送控制指令采取相应措施。烟雾安全监控功能固定为自动模式。

控制终端做为整个智能家居系统的核心，是传感器数据的汇集点，也是最主要的控制端。控制终端上的 Qt 程序主要实现以下功能：

1) 数据接收、解析、显示、更新

接收数据功能主要 `posix_qextserialport` 串口类，通过创建该类的对象，设置固定串口属性，利用 Qt 定时器 `QTimer` 每隔固定时间读取串口数据。读取到数据后，程序以固定字符分割数据，得到相应的温度、湿度、灯光状态、风扇状态等对应值，最后调用 `setWidgetvalue()` 界面上的信息。读取串口数据函数如下：

```
void MainWindow::readMyCom() //读串口函数
{
    QString data="iiii";
    data =myCom->readAll();
}
```

2) 发送控制命令控制相应电器设备

用户通过点击按钮触发事件，通过串口类对象调用 `write()` 往串口写指令，经由协调器转发指令实现电器控制。

3) 语音报警

程序通过 `QProcess` 类创建新的进程，通过启动 Linux 环境下的 `mplayer` 软件达到播放音频文件的功能。当采集到的某项数据超过设定阈值，程序会更新音频播放进程的启动参数，使 `mplayer` 播放对应音频文件进行语音报警。函数如下：

```
//启动播放音频进程
void MainWindow::startMusic()
{
    QString mplayerPath("/mytest/music/./mplayer");
    mplayerProcess->start(mplayerPath, args);
    musicflag=1; //标志进程启动
}
```

4) 通过 TCP 连接上传数据到服务器

程序通过 `QTcpServer` 创建 TCP 连接的服务器端，并监听指定端口号，当接收到 PC 服务器端发送的连接请求后，建立连接并取得 `Socket` 套接字。成

功连接后，程序会将每次收集到的数据通过网络发往服务器。

3.3 服务器程序设计

PC 端服务器程序也是使用 Qt 语言进行开发，由 TCP 连接界面、灯光控制界面、风扇控制界面、温湿度监测界面、安全监测界面和数据库界面组成。程序主要实现以下功能：

- 1) 建立 TCP 连接，接收控制终端上传的数据，写入数据库
- 2) 将网络控制指令转发（来自移动终端）



图 3.3.1 服务器 TCP 连接界面



图 3.3.2 服务器数据库界面

3.4 移动控制端设计

3.4.1 ESP8266 WIFI 模块控制网络的搭建

ESP8266-01 是一款搭载 ESP8266 芯片的，体积小、功耗低、价格低、支持透传，丢包现象不严重的 WiFi 模块。该模块支持 STA/AP/STA+AP 三种工作模式，支持串口透传、PWM 调控和 GPIO 控制，广泛应用于工业、消防、安防智能一体化管理、WiFi 远程监控/控制、智能卡终端，无线 POS 机，Wi-Fi 摄像头，手持设备等。该 WiFi 模块处于 AP 模式，传输模式为透传模式。AP 模式下，ESP8266 模块作为 WiFi 热点，Android 手机通过直接与模块通信，实现局域网无线控制。ESP8266-01 无线 WiFi 模块实物图如下：



图 3. 4. 1 ESP8266-01 无线 WiFi 模块

设置步骤大概如下：

- (1) AT+CIPSERVER =0,关闭 server 服务
- (2) AT+RST 重启 WIFI 模块
- (3) 手机端 APP 创建服务器 IP:192.168.4.2 端口 8080
- (4) AT+CIPMUX=1, 开启多路连接模式
- (5) AT+CIPSTART=2, "TCP","192.168.4.2",8080 建立 TCP 连接
- (6) AT+SAVETRANSLINK=1,"192.168.4.2",8080,"TCP",100 连接服务器，保存透传到 FLASH，下次上电自动建立 TCP 连接并进入透传。

3.4.2 SIM800C 短信模块控制环境搭建

PZ-SIM800C GSM-GPRS 模块是深圳普中科技推出的一款高性能工业级 GSM/GPRS 模块（开发板）。PZ-SIM800C GSM-GPRS 模块板载 SIMCOM 公司的工业级四频 GSM/GPRS 模块：SIM800C，工作频段四频：850/900/1800/1900MHz，可以低功耗实现语音、SMS(短信)、MMS(彩信)、蓝牙数据信息的传输。PZ-SIM800C 模块支持 RS232 串口和 LVTTTL 串口(即支持 3.3V/5V 系统)，并带硬件流控制，支持 5V~24V 的超宽工作范围，使得本模块可以非常方便的与产品进行连接，从而给您的产品提供包括语音、短信、彩信、蓝牙和 GPRS 数据传输等功能。本系统使用 SIM800C 模块作为短信的接收站，负责读取 Android 移动控制端发来的含有控制指令的短信，并将短信内容通过串口送往 PC 端服务

器，再经由服务器程序解析短信内容，并将得到的控制指令送往控制终端。

SIM800C 模块实物图如下：

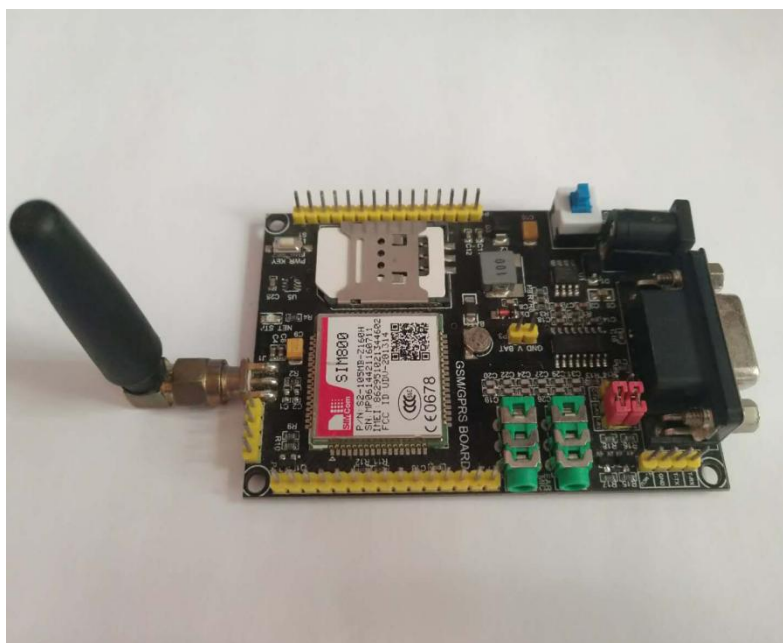


图 3.4.2 SIM800C 模块

SIM800C 模块设置步骤如下：

- (1) AT+CMGF=1，设置短消息模式为文本模式。
- (2) AT+CMGR=1 当接收到消息时，使用该命令读取信息，读取的信息为以下形式：
+CMGR: "REC UNREAD", "1881386453", "短信内容", "17/08/09, 10:11:29+32"
- (3) AT+CMGS="18813864534" 用于发送短信，本系统未用到

3.4.3 Android 移动端 APP 界面设计



图 3.4.3.1 控制界面



图 3.4.3.2 数据显示界面

如以上两图所示，该 APP 主要由控制界面和数据显示界面两个界面组成。控制界面分为两个部分，分别对应一种实现移动端控制的方式，上部分为 WIFI 控制方式。当用户处于 WIFI（服务器放置在家中，WiFi 与服务器相连）信号覆盖范围内，用户通过连接该 WIFI，在点击控制界面上的连接服务器按钮即可在移动手机和服务器之间创建通信连接，用户即可通过点击相应按钮发送控制指令控制相应的家居设备。在 WiFi 连接过程中，该 APP 会接收来自服务器返回的家居环境的各项数据，通过特定的解包处理解析数据，最终在页面上显示数据。此外，点击显示数据按钮即可查看最近的数据。下部分为 SIM 短信控制方式，用户通过向连接服务器的 SIM800C 模块上的 SIM 卡发送特定的指令，可以实现远程控制，解决了 WiFi 控制方式距离短的问题。

3.4.4 移动控制功能的实现

3.4.4.1 WIFI 控制方式

(1) 手机 APP 端创建服务器，用于连接 WIFI，IP 地址为 192.168.4.2，与 ESP8266 无线模块创建 TCP 连接的 IP 一致，创建成功后点击连接服务器按钮即可接受来自 WIFI 模块的 TCP 连接请求，当 APP 创建好 TCP 连接后会返回 Socket 对象，通过调用对应的方法即可获得 Socket 中输入流和输出流的对象。部分函数如下：

```
mServerSocket = new ServerSocket(8080,
5, InetAddress.getByName("192.168.4.2"));

connect.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try { //开启服务、指定端口号
            if((connectflag==false) && wificonnectflag){
                try{
                    mSocket = mServerSocket.accept();
                } catch (IOException e){
                    e.printStackTrace();
                }
            }
            if(mSocket ==null) {
                Log.i("test", "连接失败");
            }
            else{
                Log.i("test", "连接成功");

                connect.setText("连接成功");

                mOutputStream = mSocket.getOutputStream();
                mInputStream = mSocket.getInputStream();
                ReadThread.start();
                showValuethread.start();
                connectflag=true;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
```



```

    }
}
});

```

(2) 发送控制指令。当点击开关按钮时会触发点击事件，然后使用输出流对象往 Socket 写入控制指令，Socket 随后将指令通过网络送往服务器，发送指令函数如下：

```

    sendMSG = "RLO";
    try {
        if(connectflag) {
            mOutputStream.write(sendMSG.getBytes());
            mOutputStream.flush();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

控制指令集如下表所示：

	开灯	关灯	开风扇	关风扇
房间	RLO	RLF	RFO	RFF
厨房	KLO	KLF	KFO	KFF

表 3.4.4.1 WIFI 控制指令

(3) 服务器读取控制指令，转发控制指令

```

void MainWindow::readMyCom2() //读串口函数
{
    if(mycom2flag ==1)
    {
        //读取串口缓冲区的所有数据给临时变量 temp
        QString buffer;
        int i, fg=0;
        buffer =myCom2->readAll();

        for(i=0;i<buffer.length();i++)
            if(buffer[i]=='R' || buffer[i]=='K')
            {
                fg=i;break;
            }
        if(buffer!=NULL)
        {

```

```

        qDebug() << buffer.mid(fg, 3);
    if(socketflag==1)
    {
        ByteArray ba = buffer.toLatin1();
        strcpy(sendbuf, ba.data());
        socket->write(sendbuf); //转发控制指令
    }
}
}
}
}

```

3.4.4.2 短信控制方式

(1) 通过调用系统自带的发送短信功能发送特定指令信息，再由服务器解析出短信内容中的指令部分，转发给控制终端（ARM 板）。

```

//用 SmsManager.getDefault() 得到一个 smsManager 实例
SmsManager smsManager = SmsManager.getDefault();
//自定义发出广播的内容
Intent sendintent = new Intent("SENT_SMS_ACTION");
//发出广播
PendingIntent pi =
PendingIntent.getBroadcast(MainActivity.this, 0,
sendintent, 0);
//调用 sendTextMessage 去发送短信 第一个参数是 手机号码
第三个是短信内容,
smsManager.sendTextMessage(editText1.getText().toString(), null, editText2.getText().toString(), pi, null)
;

```

短信控制指令集如下表所示：

	开灯	关灯	开风扇	关风扇
房间	#RLO#	#RLF#	#RFO#	#RFF#
厨房	#KLO#	#KLF#	#KFO3	#KFF#

表 3.4.4.2 短信控制指令

(2) 服务器读取短信内容，并以“#”符号为分割标志分割提取出控制指令，然后发往控制终端。读取短信内容部分函数如下

```
void MainWindow::readMyCom() //读串口函数
{
    temp =myCom->readAll();

    if(temp!=NULL) {
        for(i=0;i<temp.length();i++)
            if(temp[i]=='#')
                count++;
        for(i=0;i<temp.length();i++)
            if(temp[i]=='S' && temp[i+1]=='M')
                readflag=1;
    }
    if(count==0 && readflag==1)
    {
        if(temp !=NULL)
        {
            for(i=0;i<temp.length();i++)
                if(temp[i]==',' )
                {
                    fg1=i;
                    break;
                }
            for(i;i<temp.length();i++)
                if(temp[i]=='\r')
                {
                    fg2=i;
                    break;
                }
            c=temp.mid(fg1+1, fg2-fg1-1);
            Mes_count=c;
            cmd="AT+CMGR="+c+"\r\n";
            myCom->write(cmd.toLatin1());
            readflag=0;
            //qDebug()<<"sendCmd"<<endl;
        }
    }

    if(count!=0)
    {
        for(i=0;i<temp.length();i++)
```

```

        if(temp[i]=='#')
        {
            if(fg1==0)
                fg1=i;
            else
                fg2=i;
            if(fg1!=0&&fg2!=0)
                break;
        }
    if(temp!=NULL)
    {
        message=temp.mid(fg1+1, fg2-fg1-1);
        qDebug()<<message<<endl;

        if(socketflag==1)
        {
            QByteArray ba = message.toLatin1();
            strcpy(sendbuf, ba.data());
            qDebug()<<sendbuf<<endl;
            socket->write(sendbuf);
        }

        if(Mes_count=="30")
        {
            cmd="AT+CMGD=1,4\r\n";
            myCom->write(cmd.toLatin1());
        }
    }
}

```

4 系统测试

4.1 ZigBee 模块组网测试

组网测试主要测试 ZigBee 数据采集模块与协调器无线组网情况。协调器先上电，ZigBee 子节点随后上电，子节点 LED1 闪烁表明组网成功。



图 4.1 ZigBee 组网测试

4.2 串口数据传输测试



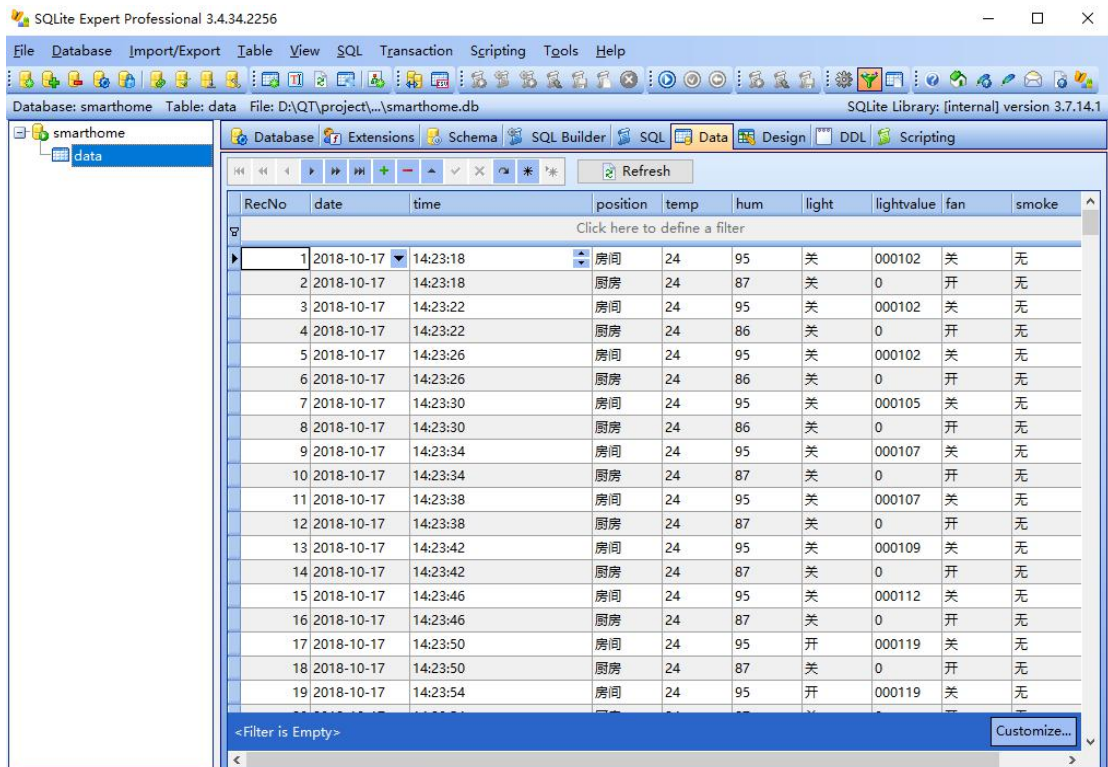
图 4.2 串口数据传输测试

如上图所示，串口助手可以接收到来自两个 ZigBee 节点传回来的数据，串口数据传送正常。数据格式如下：

R : 19 : 93 : 000704 : LF : K : 18 : 89 : SN : FF : 000 : ORN

房 温 湿 光 灯 厨 温 湿 烟 风 占 红
间 度 度 强 光 房 度 度 雾 扇 位 外

4.3 服务器端数据存储测试



SQLite Expert Professional 3.4.34.2256

File Database Import/Export Table View SQL Transaction Scripting Tools Help

Database: smarthome Table: data File: D:\QT\project\...\smarthome.db SQLite Library: [internal] version 3.7.14.1

RecNo	date	time	position	temp	hum	light	lightvalue	fan	smoke
1	2018-10-17	14:23:18	房间	24	95	关	000102	关	无
2	2018-10-17	14:23:18	厨房	24	87	关	0	开	无
3	2018-10-17	14:23:22	房间	24	95	关	000102	关	无
4	2018-10-17	14:23:22	厨房	24	86	关	0	开	无
5	2018-10-17	14:23:26	房间	24	95	关	000102	关	无
6	2018-10-17	14:23:26	厨房	24	86	关	0	开	无
7	2018-10-17	14:23:30	房间	24	95	关	000105	关	无
8	2018-10-17	14:23:30	厨房	24	86	关	0	开	无
9	2018-10-17	14:23:34	房间	24	95	关	000107	关	无
10	2018-10-17	14:23:34	厨房	24	87	关	0	开	无
11	2018-10-17	14:23:38	房间	24	95	关	000107	关	无
12	2018-10-17	14:23:38	厨房	24	87	关	0	开	无
13	2018-10-17	14:23:42	房间	24	95	关	000109	关	无
14	2018-10-17	14:23:42	厨房	24	87	关	0	开	无
15	2018-10-17	14:23:46	房间	24	95	关	000112	关	无
16	2018-10-17	14:23:46	厨房	24	87	关	0	开	无
17	2018-10-17	14:23:50	房间	24	95	开	000119	关	无
18	2018-10-17	14:23:50	厨房	24	87	关	0	开	无
19	2018-10-17	14:23:54	房间	24	95	开	000119	关	无

<Filter is Empty> Customize...

图 4.3 服务器端数据库

如上图所示，服务器端的 SQLite 数据库保存了控制终端通过网络传送回来的数据，服务器端功能正常。

4.4 Android 手机 APP 测试

手机 APP 主要测试以下三部分：

- (1) WIFI 网络命令传输测试

(3) 短信命令发送测试



图 4. 4. 2 短信命令发送测试

如以上图片所示，WIFI 网络命令传输、WIFI 接收数据、短信命令发送功能正常。

5 总结与展望

5.1 本文的主要成果

物联网技术的快速发展为智能家居行业带来了新的契机与挑战。本文使用 Cortex-A8 为核心的 GEC-210 开发板作为控制终端，ZigBee 模块作为数据采集和数据传输设备开发了一套智能家居控制系统。

本文的设计成果如下：

（1）选用 CC2530 ZigBee 模块进行以协议栈为基础进行开发，实现了数据采集以及数据传输以及往串口写数据的功能。

（2）在 Linux 环境下开发了具有读取串口数据和往串口写数据功能的 Qt 界面程序，实现了对程序进行交叉编译并移植到 GEC-210 开发板上

（3）测试了控制终端显示采集到的数据，以及发送控制指令远程控制灯光功能。

5.2 展望

由于本人经验欠缺，时间有限，学习 Linux 嵌入式以及 ZigBee 无线组网技术的时间尚短，在 ZigBee 无线数据传输以及 Qt 界面程序开发方面还有一定的改进空间。本系统需要完善以下几点：

（1）本文智能家居控制系统 ZigBee 数据采集精度不高，数据采集程序需要进一步调整。

（2）本文只测试 2 个 ZigBee 模块之间的无线传输功能，传输距离在 5-10 米，未来本系统将进行多个 ZigBee 模块无线组网，实现远距离传输。

（3）控制终端 Qt 界面流畅度不高，需要进行多线程优化，提高程序的流畅程度。

（4）本文只使用了温湿度模块、光照强度模块、烟雾传感器进行数据采集，未来本系统需要使用多种传感器，实现对家居环境全方面检测。

参考文献

- [1]基于 Android 的智能家居终端设计与实现[J]. 周时伟,谢维波. 微型机与应用. 2012(14)
- [2 基于 ZigBee 的智能家居控制节点设计与实现[D]. 罗凯.电子科技大学 2013
- [3]ZigBee 无线传感器网络设计与实现[M]. 化学工业出版社 , 王小强, 2012
- [4]基于 zigbee 技术和 android 系统的智能家居系统设计[D]. 沈淀.武汉理工大学 2011