

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：李雪华

学 号：15115061021

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 11 月—— 2019 年 1 月

（教务处制）

# 智能家居系统控制终端的设计和实现

**摘要：**随着近年来物联网的兴起与快速发展，物联网技术和传感器技术等智能家居系统中的应用不断增加。本文主要介绍了一套基于 Cortex-A8 的智能家居系统的终端控制设计，在硬件上，控制终端模块通过 USB Host (2.0) 接口与协调器模块的 Zigbee 通信模块相连，进行数据传输；通过网线，与电脑（服务器）直接相连。软件是在 Linux 环境下使用 Qt Create 软件编写出来的，通过交叉编译下载到开发板上。

本系统能够实时监测房间环境电器数据，加强家居智能化，更好的管控，因此对于实现现代化家具产业的智能化管理非常可行。

**关键词：**智能家居；Qt Creator

# **Design and implementation of control terminal for intelligent home system**

**Abstract:** With the rise and rapid development of the Internet of things in recent years, the application of IoT technology and sensor technology in smart home systems has been increasing. This paper mainly introduces a set of terminal control design of intelligent home system based on CORTEX-A8, on the hardware, the control terminal module is connected with the ZigBee communication module of the Coordinator module through the USB Host (2.0) interface for data transmission, and is directly connected with the computer (server) through the network cable. The software was written in a Linux environment using QT create software and downloaded to the Development Board through cross-compilation. This system can monitor the room environment electrical data in real time, strengthen home intelligence, better control, so it is very feasible to realize the intelligent management of modern furniture industry.

**Key words:** Smart home; Qt Creator

## 目录

一、 智能家居控制系统的设计相关技术.....	5
1.1 ZigBee 无线传输技术.....	5
1.2 Qt 图形界面应用.....	6
1.3 Sqlite 数据库技术.....	7
1.4 安卓 APP 技术.....	8
二、 系统设计总体方案.....	8
2.1 S5PV210 处理器.....	8
2.2 CC2530 模块.....	9
2.3 控制终端模块.....	10
2.4 数据采集模块.....	11
2.4.1 DHT11 温湿度模块.....	11
2.4.2 BH1750FVI 光照传感模块.....	12
2.4.3 MQ-2 烟雾气敏传感器.....	13
2.5 服务器模块.....	14
2.6 移动端控制模块.....	14
三、 控制终端模块设计.....	15
3.1 TCP 网络.....	15
3.2 串口传输数据.....	18
3.3 界面设计.....	21
3.4 其他函数.....	25
四、 系统实现.....	25
4.1 系统功能及硬件分析.....	25
4.2 软件分析.....	26
五、 系统测试.....	27
5.1 ZigBee 模块组网测试.....	27
5.2 串口数据传输测试.....	27
5.3 控制终端.....	28
六、 总结与展望.....	29
6.1 本文的主要成果.....	29
6.2 展望.....	29

## 一、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。

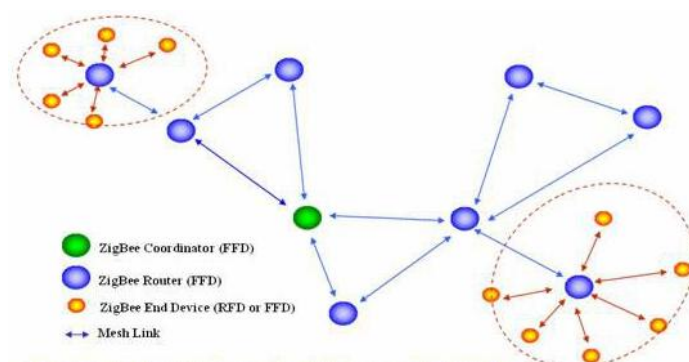


图 1.1 ZigBee 网络节点图

### 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++ 语言图形应用界面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：

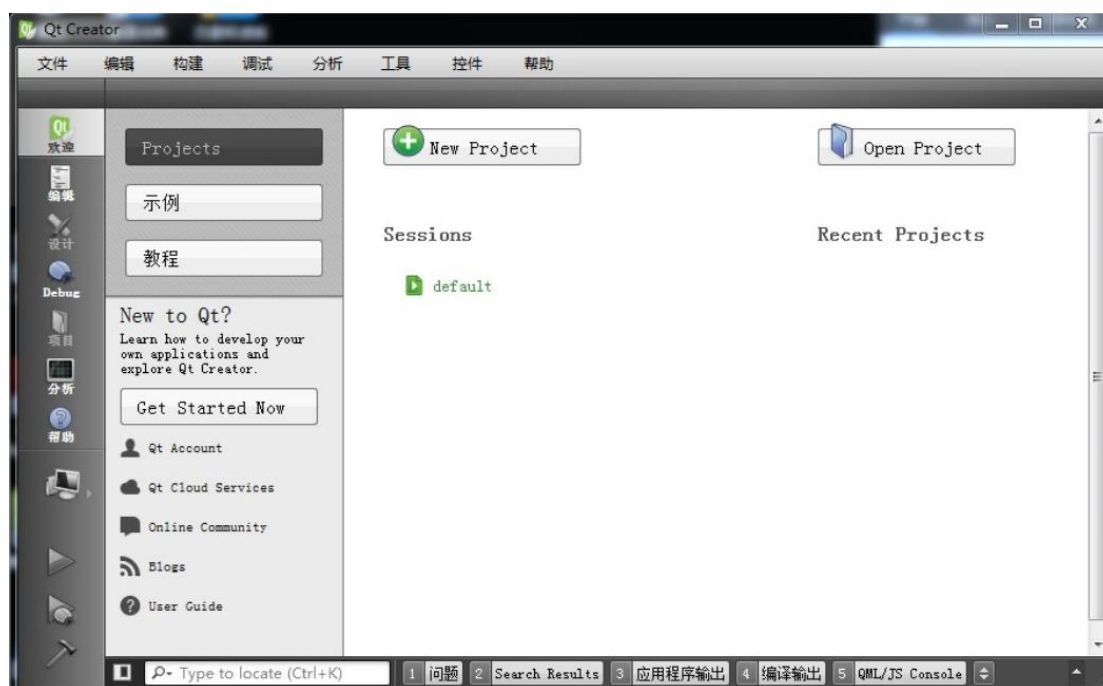


图 1.2 Qt Creator 开发软件界面图

终端控制显示是用 Qt 编写，然后移植到 Cortex™-A8 控制的嵌入式板显示出来。Qt 是一个跨平台 C++ 图形用户界面应用程序开发框架。它既可以开发 GUI 程序，也可用于开发非 GUI 程序，比如控制台工具和服务器，它提供应用程序开发者建立图形用户界面所需求的所有功能。而 Qt Creator 是一个用于 Qt 开发的轻量级跨平台集成开发环境。Qt Creator 可提供首个专为支持跨平台开发而设计的集成开发环境（IDE），并确保首次接触的开发人员能迅速上手和操作。

### 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. Richard Hipp 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：

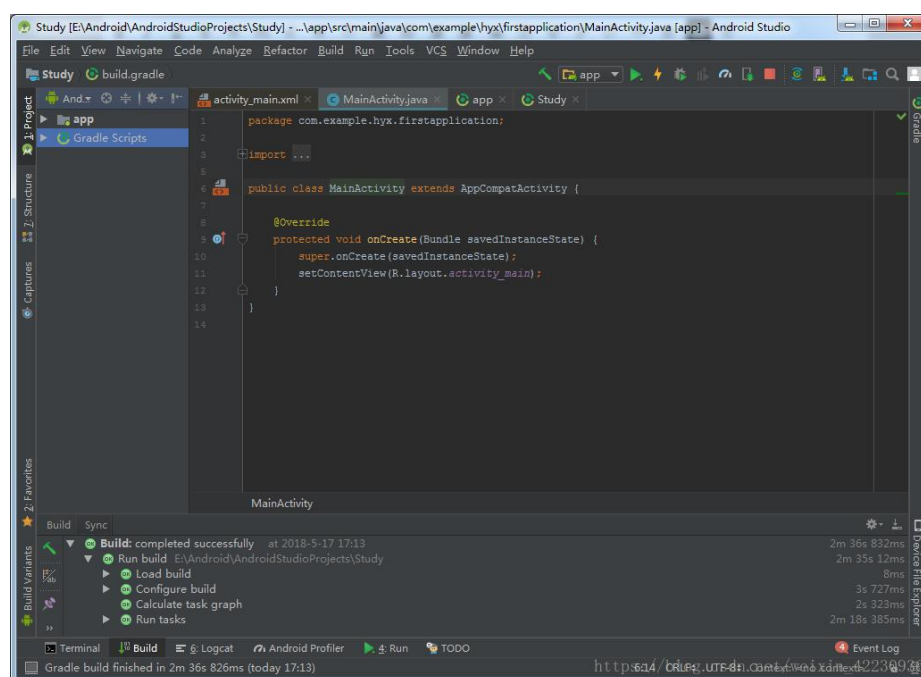


图 1.4 Android Studio 开发工具界面图

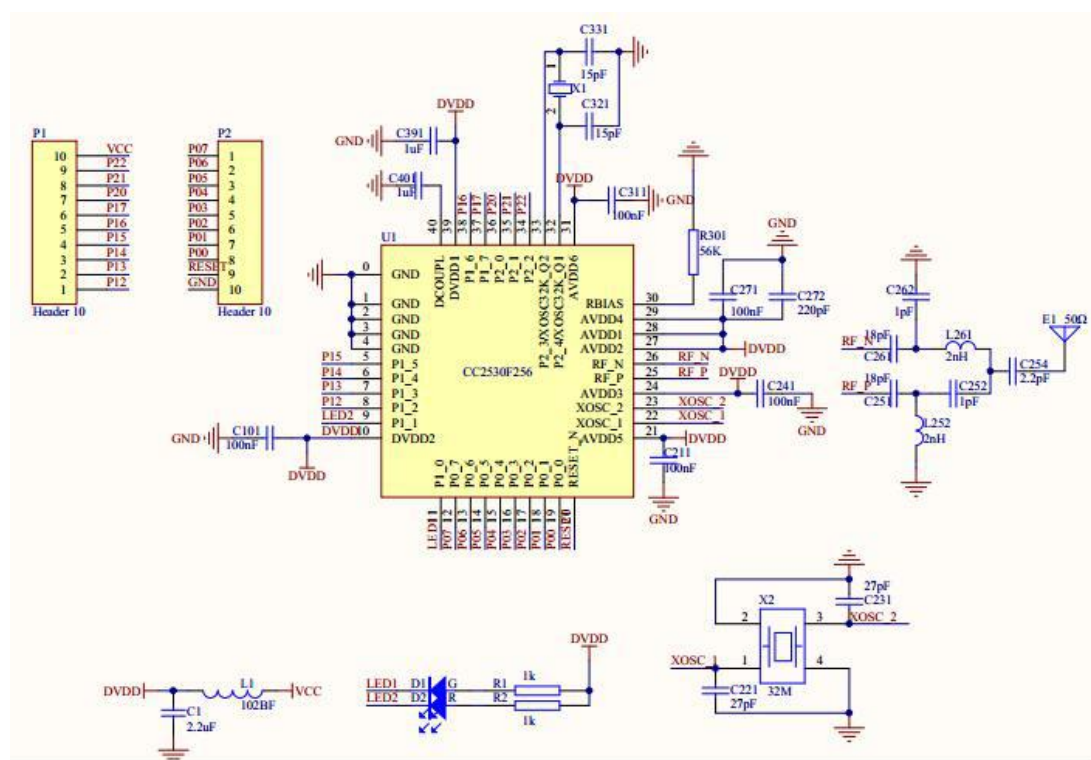
## 二、系统设计总体方案

### 2.1 S5PV210 处理器

选用 S5PV210（以 Cortex<sup>TM</sup>-A8 为核心）作为处理器，其有很多强大的阴硬件编解码功能。所以本智能家居控制系统是以 S5PV210 处理器为核心 GEC21 作为控制终端，选择 Linux 操作系统作为软件平台，运行在 Qt 开发的用户界面，进行接收、处理、上传数据一系列操作，实现对智能家居系统的监控与控制。

### 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司（TI）生产的 CC2530 ZigBee 模块负责数据和控制指令的传输。





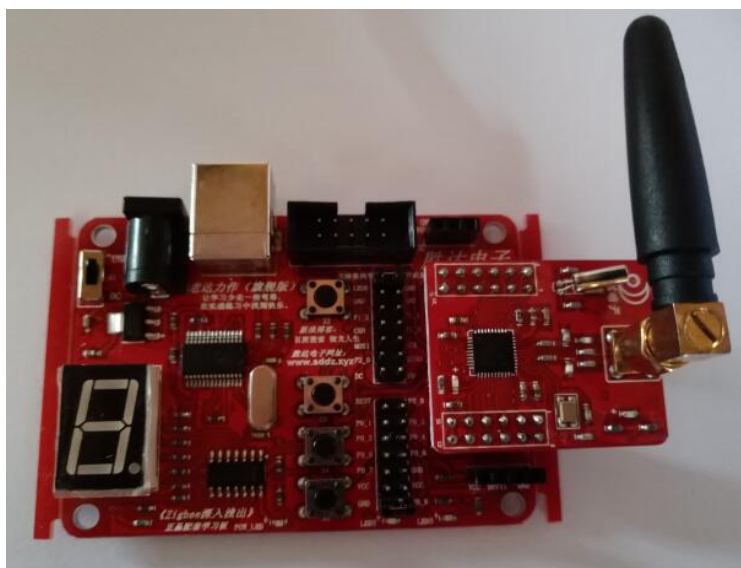


图 2.2 CC2530 模块

### 2.3 控制终端模块



图 2.3 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。

智能家居系统需要一个 GEC210 的开发板来设计一个中央控制系统，用它来实现监控和控制。GEC210 是一款高性能的 Cortex A8 核心板，它采用三星 S5PV210

为主处理器，运行主频可高达 1GHz。S5PV210 内部集成了 PowerVR SGX540 高性能图形引擎，支持 2D/3D 图形加速，其完美的性能非常适合开发物联网高级智能家居。

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 2.4 I2S0 接口

## 2.4 数据采集模块

### 2.4.1 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。

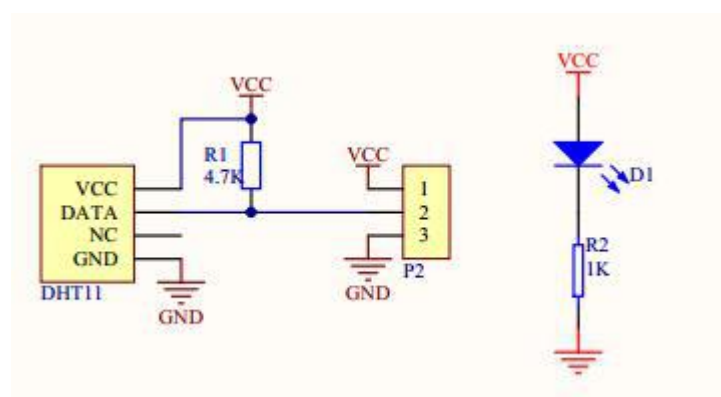


图 2.5 DHT11 模块原理图

DHT11 温湿度模块实物图如下：

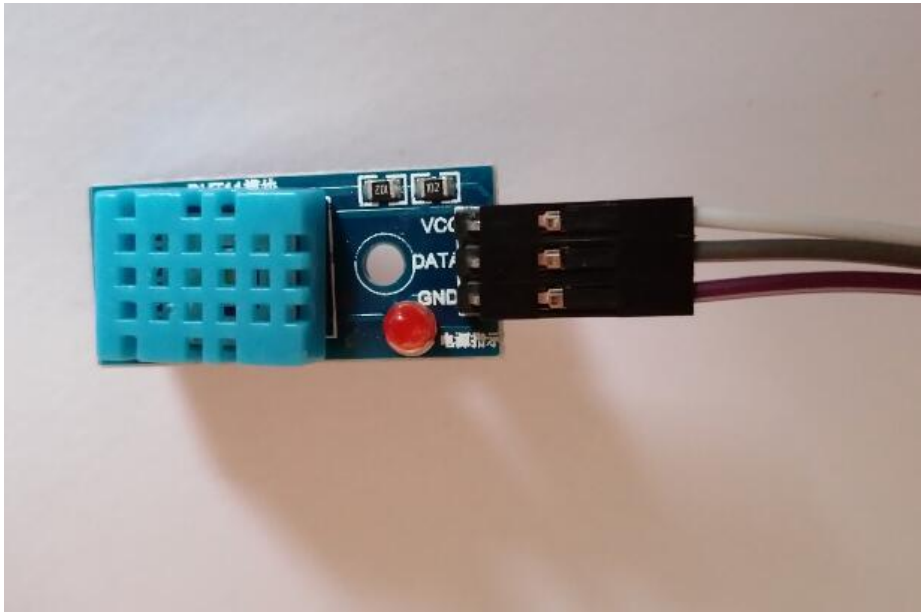


图 2.6 DHT11 模块

2.4.2 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。

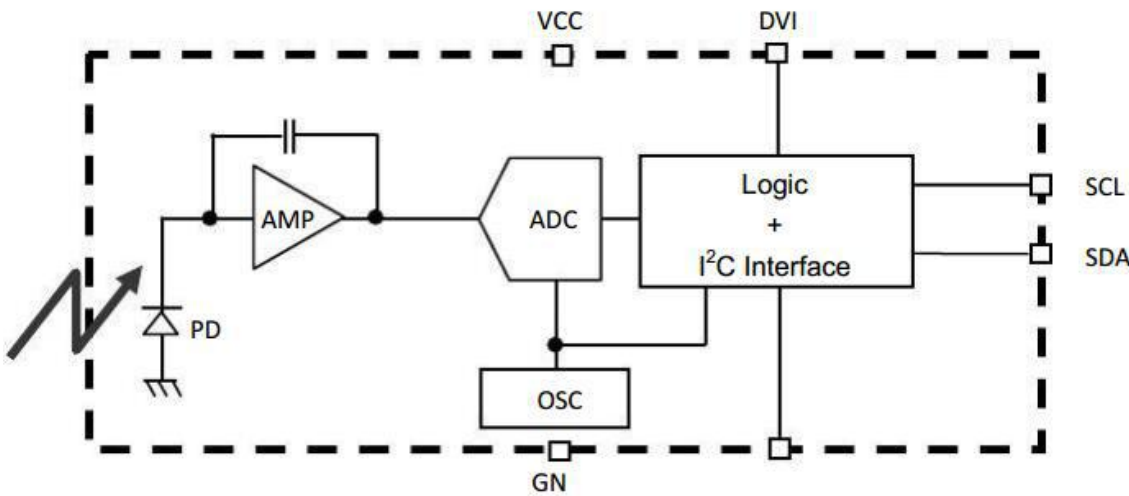


图 2.8 BH1750FV 框图

BH1750FV 光照传感模块实物如下图：

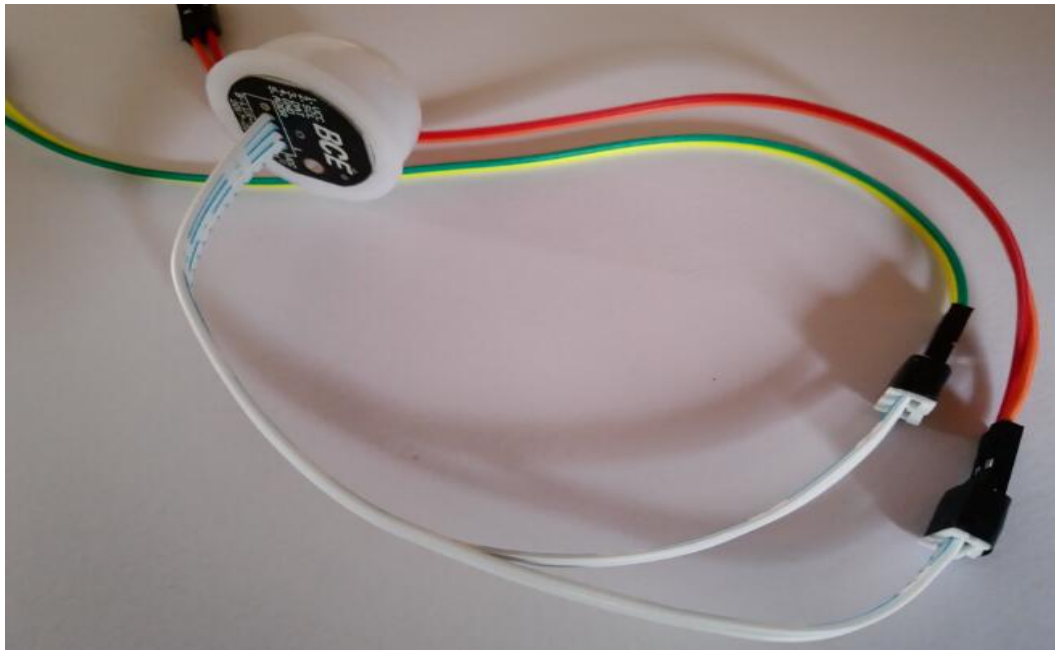


图 2.7 BH1750FV 光照传感模块

#### 2.4.3 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（ $\text{SnO}_2$ ）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。

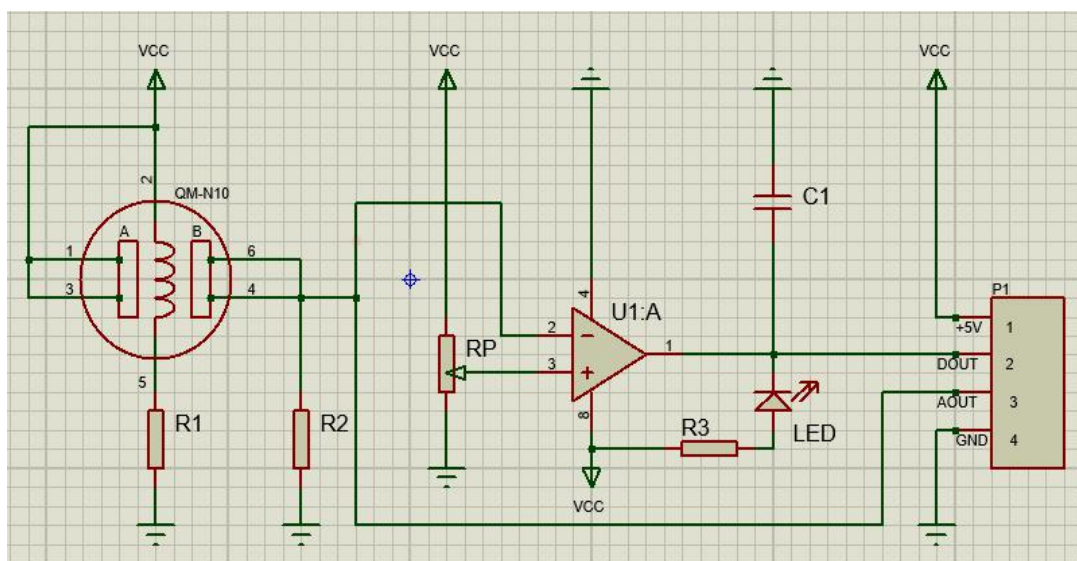


图 2.9 MQ-2 烟雾气敏传感器原理图

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。

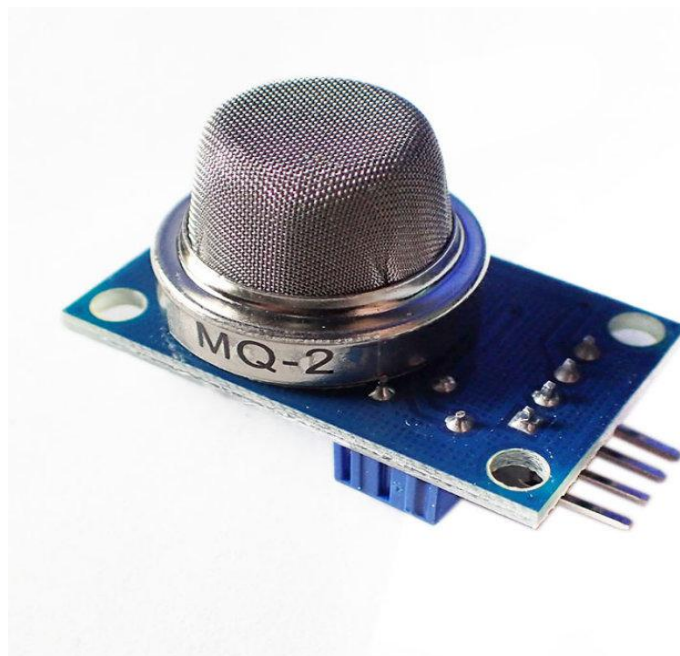


图 2.10 MQ-2 烟雾气敏传感器

## 2.5 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。

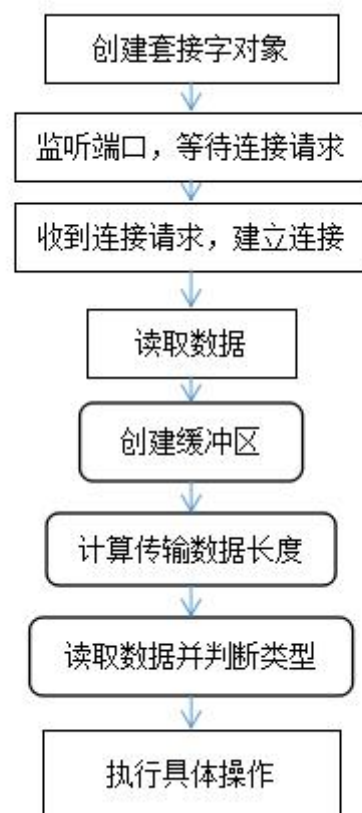
## 2.6 移动端控制模块

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。

### 三、控制终端模块设计

#### 3.1 TCP 网络

控制终端通过网线与服务器（电脑）相连，这里使用 TCP 网络，采用 S/C 模式进行通信。首先，在建立连接之前，先设置好确定好端口号、IP 地址。由于开发板是 Client（客户端），而客户端被设计为等待连接，所以，设置端口号之后等待连接。当服务器那边手动点击连接之后，才开始建立连接，接收运载数据的套接字。



创建套接字对象，并且通过 Listen（）函数进行监听端口  
当监听到消息时，触发 imcoming（）函数，建立连接，  
以下是套接字对象的创建：

```
socket=new QTcpSocket(this);  
client = new QTcpServer(this);
```

监听端口，等待网络连接请求：

```
if(!client->listen(QHostAddress::Any, 9090))
```

若接收到连接请求，调用以下函数：

```
void MainWindow::imcoming()
{
    把客户端的 socket 端口存入到服务器端的 socket 端口：
    socket = client->nextPendingConnection();
    ui->socktextBrowser->setText("已连接");
}
```

连接完毕后，调用读取函数、进行网络数据读取：

首先，清空在头文件中已经定义好的 buff，防止其他数据干扰

其次，计算传送数据长度，将数据从管道中读取出来并进行判断

判断命令信息并实现具体操作

/\*读信息函数\*/

```
void MainWindow::onread()
{
    char buf[5];
    memset(buf, 0, sizeof(buf));
    int length = socket->bytesAvailable();
    socket->read(buf, length);

    QString databuf=buf;
    if(databuf=="HL0")
        on_light_hall_on_clicked();
    if(databuf=="HLF")
        on_light_hall_down_clicked();
    if(databuf=="KL0")
        on_light_kitchen_on_clicked();
    if(databuf=="KLF")
```



```

        on_light_kitchen_down_clicked();
    if(databuf=="RLO")
        on_light_room_on_clicked();
    if(databuf=="RLF")
        on_light_room_down_clicked();
    if(databuf=="CLO")
        on_light_corridor_on_clicked();
    if(databuf=="CLF")
        on_light_corridor_down_clicked();
    if(databuf=="RFO")
        on_fan_room_on_clicked();
    if(databuf=="RFF")
        on_fan_room_down_clicked();
    if(databuf=="KFO")
        on_fan_kitchen_on_clicked();
    if(databuf=="KFF")
        on_fan_kitchen_down_clicked();
}

```

控制终端除了数据上传之外，还负责上位机的命令传递，所以需要用到 Write（）函数将数据写入串口。

同时，创建一个新进程，单独为报警服务，新进程通过启动 Linux 系统自带的 mplayer 软件，mplayer 软件播放音频文件实现报警功能。

//播放音频

```

mplayerProcess = new QProcess(this); //新进程
mplayerProcess->setProcessChannelMode(QProcess::MergedChannels);

```

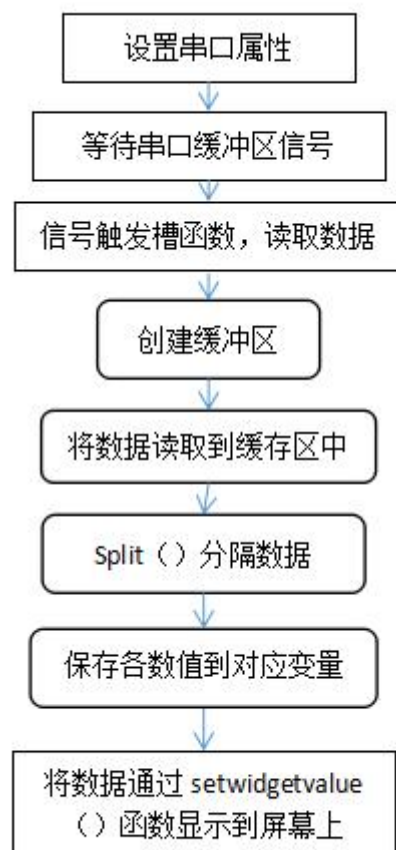


本设计中，设计有警报声提醒，包括设置更改提醒声，以下就是通过一个新创建的进程，运行 Linux 自带的一个音乐播放器，并使用它对某些音乐文件进行播放：

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

void MainWindow:: stopMusic()
{
    mplayerProcess->kill();
    musicflag=0;
}
```

### 3.2 串口传输数据



串口：在 <http://sourceforge.net/projects/qextserialport/> 下载  
posix\_qextserialport.cpp 和 posix\_qextserialport.h  
qextserialbase.cpp 和 qextserialbase.h

添加到工程之中，创建对象，设置串口相关参数：

波特率、数据位、奇偶效验、停止位、数据流控制

检测串口缓冲区，当有数据的时候触发信号，通过信号与槽的连接

触发串口数据读取函数

//定义串口对象，并传递参数，在构造函数里对其进行初始化

```
myCom = new
```

```
Posix_QextSerialPort("/dev/ttyUSB0", QextSerialBase::Polling);
```

```
//以可读写方式打开串口
```

```
myCom->open(QIODevice::ReadWrite);
```

```
myCom->setBaudRate(BAUD9600); //波特率设置，我们设置为 9600
```

```
myCom->setDataBits(DATA_8); //数据位设置，我们设置为 8 位数据位
```

```
myCom->setParity(PAR_NONE); //奇偶校验设置，我们设置为无校验
```

```
myCom->setStopBits(STOP_1); //停止位设置，我们设置为 1 位停止位
```

```
myCom->setFlowControl(FLOW_OFF); //数据流控制设置，我们设置为无数
```

据流控制

```
myCom->setTimeout(500);
```

```
//信号和槽函数关联，当串口缓冲区有数据时，进行读串口操作
```

```
connect(time2, SIGNAL(timeout()), this, SLOT(readMyCom()));
```

```
time2->start(1200);
```

同样的，创建一个 buff，用来存放读取的数据

创建一个列表，通过 split（）函数，判断分隔符号，并将字符串分割逐一  
存进列表中

通过判断第零位、第五位的字符，确定数据类型，并存放对应变量中  
最后通过 setWidgetvalue() 输出到屏幕上

```
void MainWindow::readMyCom() //读串口函数
{
    //读取串口缓冲区的所有数据给临时变量 data
    QString data="iiii";

    data =myCom->readAll();
    Sdatabuf=data;
    if(data!=NULL)
    {
        qDebug()<<data<<endl;
        QStringList sections = data.split(QRegExp(":"));
//把每一个块装进一个 QStringList 中
        if(sections.at(0)=="R")
        {
            Rtemp=sections.at(1);
            Rhum=sections.at(2);
            Rlightvalue=sections.at(3);
            Rlightstyle=sections.at(4);
        }

        if(sections.at(5)=="K")
        {
            Ktemp=sections.at(6);
            Khum=sections.at(7);
            Ksmoke=sections.at(8);
            Kfanstyle=sections.at(9);
        }
    }
}
```

```

    }

    setWidgetvalue();
    //qDebug()<<Rtemp<<Rhum<<Rlightvalue<<Rlightstyle<<endl;
    //qDebug()<<Ktemp<<Khum<<Ksmoke<<Kfanstyle<<endl;
    if(socketflag==1 && Sdatabuf!=NULL)
    {
        onSend();
    }
}

}

/*点击发送按钮后相应操作函数*/
void MainWindow::onSend()
{
    //QString data=":223";
    char sendbuf[100]={0};
    QByteArray ba = Sdatabuf.toLatin1();
    strcpy(sendbuf, ba.data());
    socket->write(sendbuf, strlen(sendbuf));    //把相关数据送入管道
    socket->flush();                            //防止数据较少不能自动发送,
    所以推一把
}

```

### 3.3 界面设计

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。

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 3.1 主界面



图 3.2 TCP 网络连接界面

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 3.3 灯光控制界面



图 3.4 风扇控制界面





图 3.5 温湿度监控界面



图 3.6 红外烟雾监控界面



图 3.7 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。

### 3.4 其他函数

main ()：设置文本显示属性、设置窗口标题等，显示 UI 界面

Mainwindow ()：

- 1) 里面含有许多跟按钮(button)相关的单击按钮信号和与信号相连的槽函数。
  1. 通过点击按钮，可以发出控制实物状态的命令（置 1、置 0，高低电平），同时改变标签（label）显示的状态。如：声音开与关、自动模式的开与关、风扇（大厅、房间、厨房）的开与关、灯（大厅、房间、厨房、走廊）的开与关。同时给每个操作配上了音乐。
  2. 或者是控制页面的按钮：返回按钮（调用 Back（）函数：显示已隐藏按钮，隐藏本 Widget）、不同界面的进入按钮（灯光、风扇、温度、设置、安全、联网）
  - 2) 关联窗口小部件（Widget）显示情况与变量所存数值的关系
  - 3) 同时也在这里设置控件属性：参数变量初始化、按钮控件聚焦情况（无聚焦）、文本表示属性（大小、水平居中）

## 四、系统实现

### 4.1 系统功能及硬件分析

控制终端模块下接感知层、上接上位机应用层，可以说这是整个系统的核心，也是连接感知层和上位机之间的一个中介。

硬件是 GEC-210 开发板，它的处理器是 S5PN210，以 CortexTM-A8 为核心；

在硬件上，控制终端模块通过 USB Host (2.0) 接口与协调器模块的 Zigbee 通信模块相连，进行数据传输；通过网线，与电脑（服务器）直接相连。



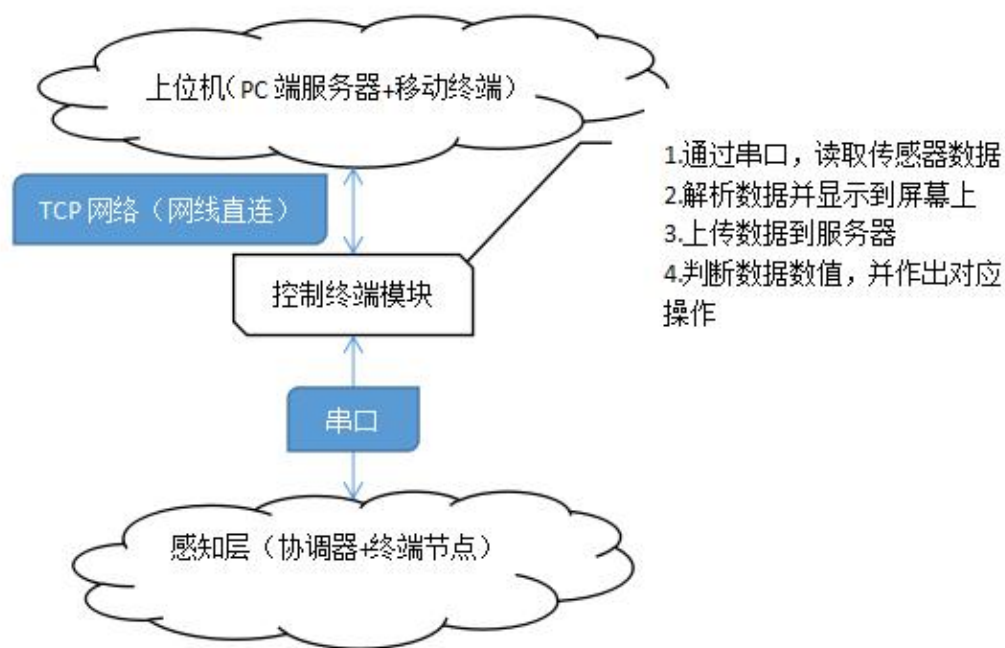


图 4.1 控制终端功能

## 4.2 软件分析

软件是在 Linux 环境下使用 Qt Create 软件编写出来的, 通过交叉编译下载到开发板上。

在软件上, 通过创建 `posix_QextSerialport` 窗口类的对象, 对串口属性进行设置。使用定时器 `Qtime` 进行定时, 以便按一定的时间间隔调用串口读取函数 `ReadMyCom()` 读取串口数据。数据, 如温湿度、光照强度等, 皆是按顺序排序, 通过一特定符号分隔开来。所以, 通过调用解析函数, 通过对特定符号的识别, 对数据进行解析, 最后调用 `SetWidgetvalue()` 函数显示到显示器上。

# 五、系统测试

## 5.1 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 5.1 ZigBee 组网测试

## 5.2 串口数据传输测试



图 5.2 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

### 5.3 控制终端

控制终端上进行处理、反馈和发送。

经过测试，控制终端能够与服务器正常进行 TCP 连接及数据传输；

并且，界面能正常显示传感数据；

通过设置、手动控制，都很好控制家电状态。



控制终端显示的界面，可以通过不同需求查看不同的按键，如房间温湿度、光照强等，以此来控制。

经过多次试验测试，观察数据，检测结果基本准确，控制基本实现。

## 六、总结与展望

### 6.1 本文的主要成果

本人主要负责控制终端这部分，使用 S5PV210 为核心的 GEC210 开发板作为控制中断，

本文的设计成果如下：

- (1) 介绍了智能家居控制系统的设计与开发，同时调研了国内外智能家居方向的发展现状，看到了极为广阔乐观的市场前景，明确了进行基于 S5PV210 的智能家居控制系统开发的重要性。

- (2) 通过比较和分析控制终端模块通过 USB 接口与协调器模块的 Zigbee 通信模块相连，进行数据传输。

## 6.2 展望

由于本人经验欠缺，时间有限，学习物联网及传感器技术的知识不够深入，在本智能家居控制系统还有改进的空间。其中，我对数据存在疑问，通过看书查资料以及寻求团队的帮助，最终解决疑难点。伴随着不断的数据处理、上传、管理、控制等需求，并且还有更多的技术和硬件发展，智能家居控制系统将会应用新技术增加新的功能，需要完善以下几点：

- (1) 本系统控制终端控制的是特定的模块，未来本系统还将开发控制多个关联性的物体模块进行高端智能控制，实现更加人性化、智能化的监测与控制。
- (2) 完善服务器、数据库及客户端组成，方便系统维护与管理，提升用户对家居控制的远程交互使用感。

## 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