支持视觉身份验证的智能家居系统设计

详细设计说明书

V1.0.4

2014 年 5 月

文档历史

序号	版本号	作者	修订
1	1.0.0	卓灿辉	起稿
2	1.0.1	徐阳	修改
3	1.0.2	许文勇	修改
4	1.0.3	季蕴青	修改
5	1.0.4	田燕	修改、审阅

目录

一、概述

本项目着眼于家庭生活环境，以智能家居系统为研究对象，设计了一种以传统住宅为基础的智能家居综合控制系统。系统以 ARM 处理器为核心，使用了 Linux 操作系统，采用模块化的设计方法，在外围模块的配合下，实现了诸如温度采集、人脸检测、灯光控制等功能。

本控制器的主要功能可以划分为三个部分：

1.1 PC 端功能

该部分的功能主要包括登陆控制器的 Web 界面，检测控制器环境的相应数据，发送相应指令对相关部分形成控制。项目实施的过程，各项功能都是先在 PC 机开发完成，然后再移植到控制器端。该部分也主要用于开发过程的调试，因此，该步骤的实施完成对于整个项目具有重要意义！

1.2 控制器端功能

该部分是整个项目的核心，各项功能的开发都围绕该部分展开。该部分的功能主要包括：

1) 运行 Web 服务器，解释执行客户端的指令；
2) 采用 Zibgee 构建传感器网络，接收并分析传感器数据，本项目采用的传感器是温度传感器；
3) 监控摄像头端的数据，实现人脸检测。

1.3 客户端功能

该部分包括来访客户的身份验证、环境数据的检测与通信、远程检测与控制，是基于安卓环境，以 Java 语言开发的一个客户端应用。

二、系统架构

2.1 系统功能模块描述

本系统主要核心为一台中央控制器，系统上直接能过电缆连接着的设备有：

1) 基于 Zigbee 的无线通信模块,其中 Zigbee 组网模块的另一模块连接着检测环境信息所需要的传感器；
2) 以太网电缆，PC 机或者智能手机可通过网络连接到该控制器；
3) 摄像头。

2.2 基本工作环境描述

基本的工作环境描述如下所示：

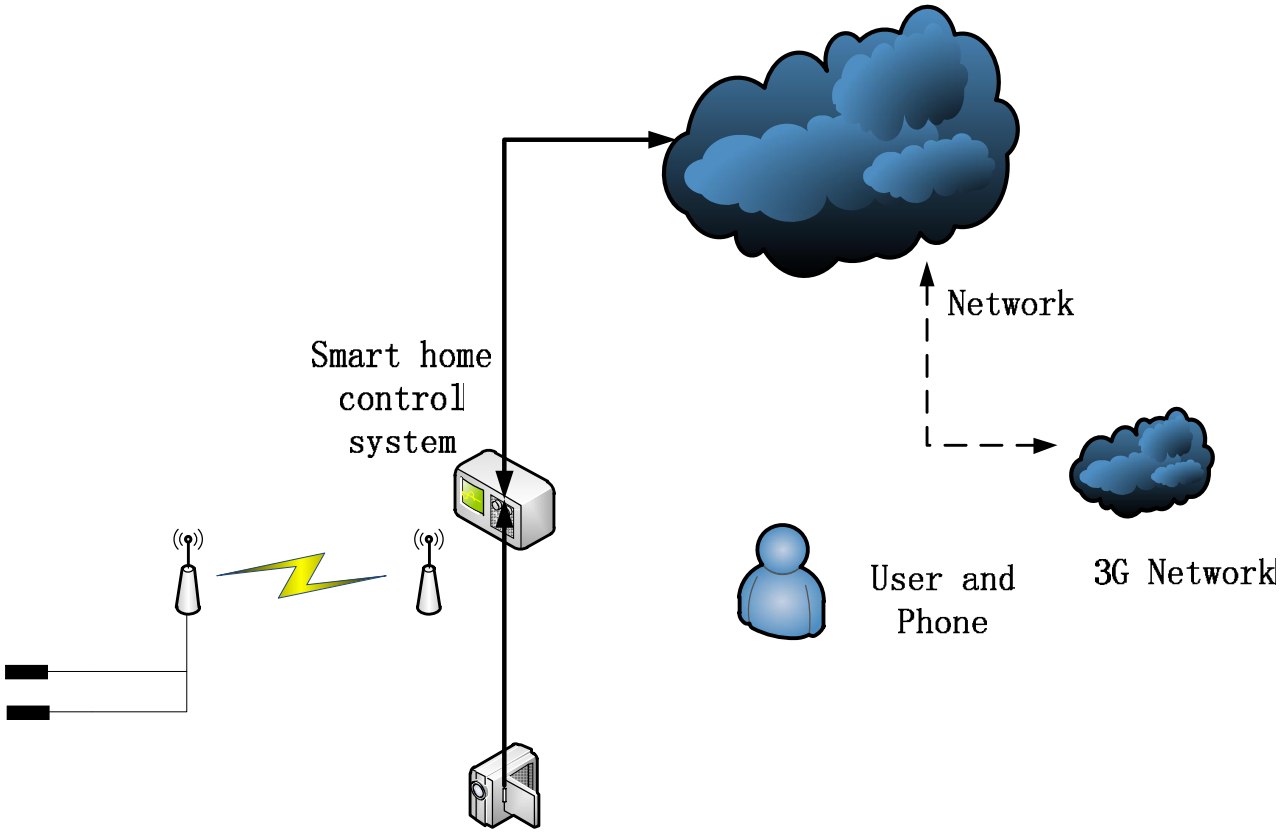


图 1 智能家居控制器工作环境描述

2.3 系统架构描述

系统架构图如下所示：

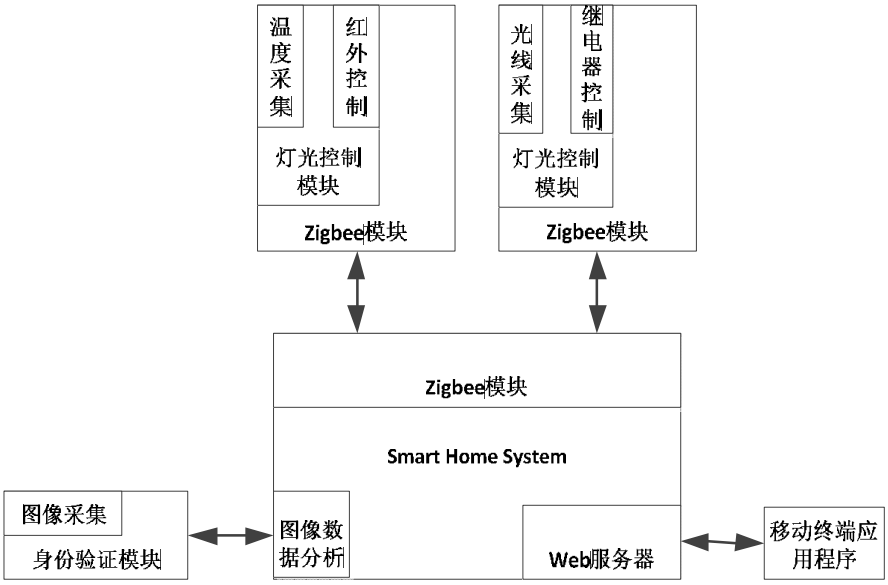


图 2 中央控制器系统架构图

2.4 软硬件组成部分

2.4.1 开发平台

本项目采用了两个开发系统，一个是项目初期采用的飞凌 OK6410，一个是 FTV210 开发系统。造成这个情况的原因是早期小组成员拥有一块飞凌 6410 的开发板，原本设想是项目进展到要构建无线传感器网络的时候，再去购买相应的模块。但是后期发现要使用这样的系统，还要考虑该模块与 OK6410 的兼容性，由于担心会产生额外的工作量及更多的困难，因此，我们采用了本身自带 Zigbee 模块的开发系统，即 FTV210 开发系统。

2.4.1.1 飞凌 6410 开发板

6410 是基于 SAMSUNG16/32 位 RISC 微处理器 S3C6410X 的一款开发平台，S3C6410X 是基于 ARM1176JZF-S 核的用于手持、移动等终端设备的通用处理器。相对于采用 ARM9 处理器的开发板而言，该开发环境速度更快，支持的功能更加齐全。其硬件参数如下所示：

1) 支持 Linux、Android 及 WinCE 操作系统；
2) 支持提供 128M 和 256M 两个版本的 RAM，本项目采用的是 256M 的 RAM；
3) 核心板支持 NOR Flash 扩展，本次采用的是 1G 的 NAND Flash；
4) OK6410-A 提供 6 个独立按键 OK6410-B 提供 8 个独立按键，并独家支持矩阵键盘，最多可额外扩展 64 个按键，可以支持开发过程中的部分测试；

5) OK6410 引出 1 个 RS232 和 3TTL 电平的串口,串口是开发过程中不可或缺的一个部分;

6) OK6410 的底板可固定 3.5、4.3、5.6 寸液晶屏，本项目中采用的是 4.3 寸的触摸屏，便于摄像头模块的开发测试。

2.4.2 操作系统的搭建

项目采用了 Linux-3.01 作为内核。选用该内核版本并没有特殊的原因，只是因为开发板自带的系统版本是该版本，因此，为了减少工作量及可能出现的其他问题，我们并没有去改用其他版本的操作系统。

2.4.3 嵌入式 Web 服务器

控制器提供远程访问的能力，本项目的做法是在控制器的运行一个 Web 服务器，以响应远程客户端的请求。

2.4.3.1 以太网接入互联网

控制器采用以太网方式接入因特网，实现与远程客户端的数据通信。本项目中采用的是有线局域网的方式。

2.4.3.2 无线传感器网络

由于需要采集的信息多样，当传感器种类较多时，传感器采用传统的有线连接方式来运行有诸多不便，因此项目采用了基于 Zigbee 自组网的传感器网络，实现的环境信息的采集与通信。由于飞凌 OK6410 没有 Zigbee 模块，因此本部分功能是在 FTV210 开发板上实现的。

2.4.4 身份验证模块

该模块主要是负责摄像头数据的采集及图像数据的处理,涉及到的技术主要包括图像采集技术 Video4Linux 及 Intel 开发的 OpenCV 图像处理函数库，摄像头采用的是 COMS OV9650 摄像头。

2.4.5 温度检测模块

该模块由传感器与 Zigbee 模块构成，温度传感器采用的是 DS18B20，模拟的温度信号经变换后转换为数字信号,并经由 Zigbee 从模块发送给 Zigbee 主模块,实现传感器网络的通信。

2.4.6 灯光远程控制模块

该模块主要的功能是实现远程客户端对于家庭电器的控制功能,项目中通过控制器上的灯光来进行模拟，具体做法是在 PC 机上或者手机客户端登陆 Web 页面，发送相应的指令，控制器端的 Web 服务器响应请求，并且对控制器上的灯发送信号。

2.5 实物展示

2.5.1 飞凌 6410

2.5.2 FTV210 实验箱

三、具体功能描述

3.1 身份验证并做出决策及相应的响应

该部分的主要功能是：

采用摄像头获取人脸数据并进行分析，实现对于主人或者访客的身份验证。

3.2 家居环境情况的检测与远程监控

该部分的主要功能是：

检测：基于 Zigbee 技术构建无线传感器网络，通过各类传感器获取信息，本项目中实现的是对环境温度信息的采集。

远程监控：基于所设计的图象处理模块、网络通讯功能及 web 界面，通过远程 web login 实现远程通讯。

3.3 家居电器的智能控制与远程控制

该部分的主要功能是：

智能控制：中央系统基于传感器所获取的数据进行分析并做出决策，发出指令控制各类家电的控制命令，本项目中实现的对灯光的控制。

远程控制：通过远程 web login，查看家居实时环境信息，并发送家电控制命令，实现控制功能。

3.4 异常情况实时报告（移动终端推送）

该部分的主要功能是：

基于中央控制器的分析结果，判读异常后，通过终端应用程序实时推送，实现异常的实时报告。该部分功能并未实现。

四、详细设计

4.1 身份验证

该部分需要完成的工作主要包括采用摄像头获取人脸数据并进行分析，实现对于主人或者访客的身份验证。该过程涉及的技术包括：

1) 视频采集过程；

2) 数字图像处理与模式识别技术；

3) 嵌入式系统图形界面。

具体流程如下：

1) 采用 Video4Linux 采集视频数据；

2) 基于 OpenCV 处理视频数据；

3) 利用嵌入式图形界面显示图像数据。

4.1.1 视频采集技术

视频采集（Video Capture）就是把模拟视频转换成数字视频，并按数字视频文件的格式保存下来。所谓视频采集就是将模拟摄像机、录像机、LD 视盘机、电视机输出的视频信号，通过专用的模拟、数字转换设备，转换为二进制数字信息的过程。

Video4Linux（简称 V4L），是 Linux 中关于视频设备的内核驱动，目前的最新版本是 Video4Linux2，已加入 Linux 内核，使用需自己下载补丁。在 Linux 中，视频设备是设备文件，可以访问普通文件一样对其进行读写。Video4Linux 目前是 Linux 系统下通用的视频开发接口。常用的视频采集流程如下：

1) 打开设备文件。；

2) 取得设备的 capability，看看设备具有什么功能，比如是否具有视频输入，或者音频输入输出等；

3) 选择视频输入，一个视频设备可以有多个视频输入；

4) 设置视频的制式和帧格式，制式包括 PAL，NTSC，帧的格式个包括宽度和高度等；

5) 向驱动申请帧缓冲，一般不超过 5 个；

6) 将申请到的帧缓冲映射到用户空间，这样就可以直接操作采集到的帧了，而不必去复制；

7) 将申请到的帧缓冲全部入队列，以便存放采集到的数据；

8) 开始视频的采集;

9) 出队列以取得已采集数据的帧缓冲，取得原始采集数据;

10) 将缓冲重新入队列尾,这样可以循环采集;

11) 停止视频的采集;

12) 关闭视频设备。

涉及到的数据结构主要包括:

1) struct v4l2_requestbuffers reqbufs;//向驱动申请帧缓冲的请求，里面包含申请的

2) 个数。

3) struct v4l2_capability cap; //这个设备的功能，比如是否是视频输入设备。

4) struct v4l2_input input; //视频输入

5) struct v4l2_standard std; //视频的制式，比如 PAL，NTSC

6) struct v4l2_format fmt; //帧的格式，比如宽度，高度等

7) struct v4l2_buffer buf; //代表驱动中的一帧

8) v4l2_std_id stdid; //视频制式，例如：V4L2_STD_PAL_B

9) struct v4l2_queryctrl query; //查询的控制

10) struct v4l2_control control; //具体控制的值

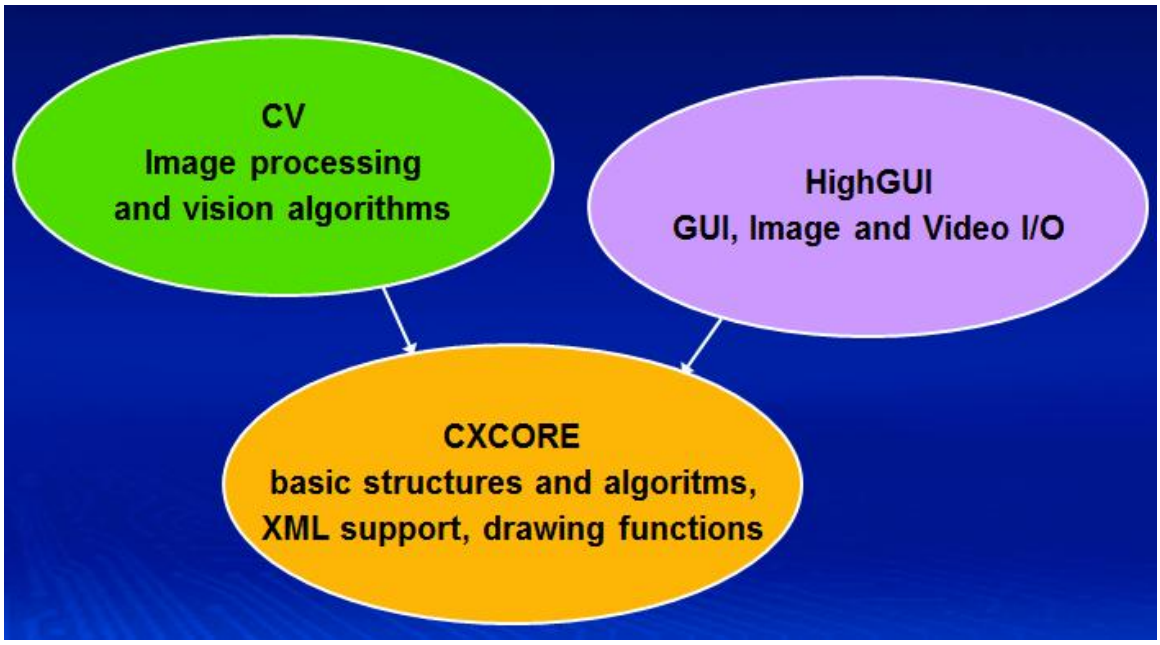
4.1.2 嵌入式系统图形界面

嵌入式 GUI 微嵌入式系统提供了一种应用于特殊场合的人机交互界面。嵌入式 GUI 要求简单、直观、可靠、占用资源小且反应快速，以适应系统硬件条件有限的条件。另外，由于嵌入式系统硬件本身的特殊性，嵌入式 GUI 应具备高度可移植性与可剪裁性，以适应不同硬件条件和使用要求。

项目初期，我们计划在控制器上采用挪威 Trolletch 公司的图形化界面开发工具 Qt 的嵌入式版本，即 Qt Embedded 来进行开发。它通过 QtAPI 与 Linux I/O 以及 Framebuffer 直接交互,拥有较高的运行效率,而且整体采用面向对象编程,拥有良好地体系架构和编程模式。但是由于项目只完成了 PC 机端的功能开发，因此该部分功能目前仍然没有完成。

4.1.3 数字图像处理与模式识别技术

数字图像处理（Digital Image Processing）通常所说的数字图像处理是指用计算机对图像信息进行的处理，因此也称为计算机图像处理（Computer Image Processing）。是利用计算机完成图像信息各种处理的基本理论和方法。其处理过程主要包括图像采集、图像预处理、图像分析和图像识别等几大部分。其精度比较高，而且还可以通过改进处理软件来优化处理效果。其内容广泛，涉及到数学、光学、信息学、摄影技术和计算机技术等学科，是一门综合性跨学科的新学科。



OpenCV 框架

该部分采用了 Intel 公司开发的 OpenCV 。

OpenCV 是 Intel 公司开发的图像处理和计算机视觉函数库，它有以下特点：

1. 开放的 C/C++源码；
2. 基于 Intel 处理器指令集开发的优化代码；
3. 统一的结构和功能定义；
4. 强大的图像和矩阵运算能力；
5. 方便灵活的用户接口；
6. 同时支持 MS-WINDOWS、LINUX 平台。

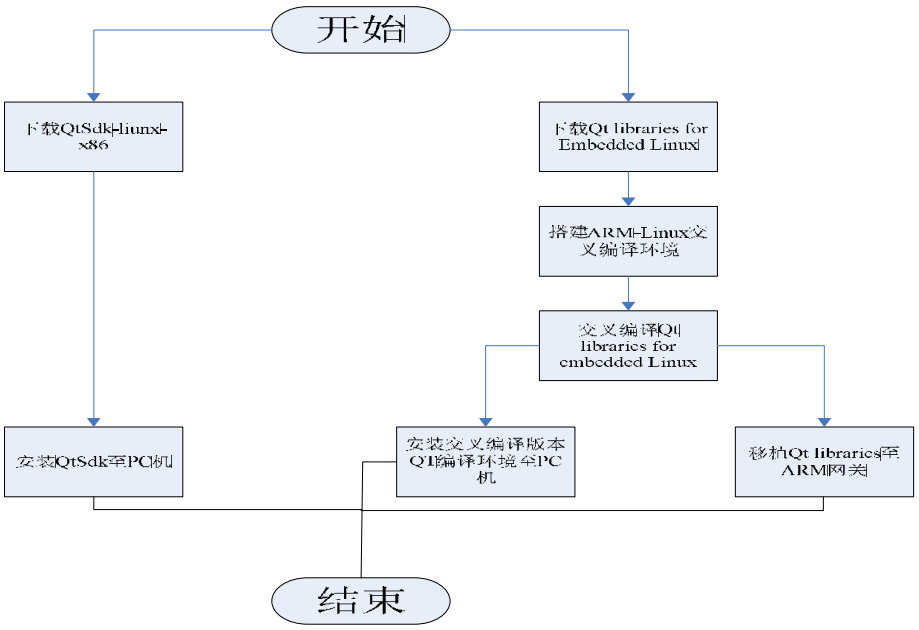
作为一个基本的计算机视觉、图像处理和模式识别的开源项目，OpenCV 可以直接应用于很多领域，是二次开发的理想工具。

4.2 嵌入式图形界面的设计

4.2.1 图形库概述

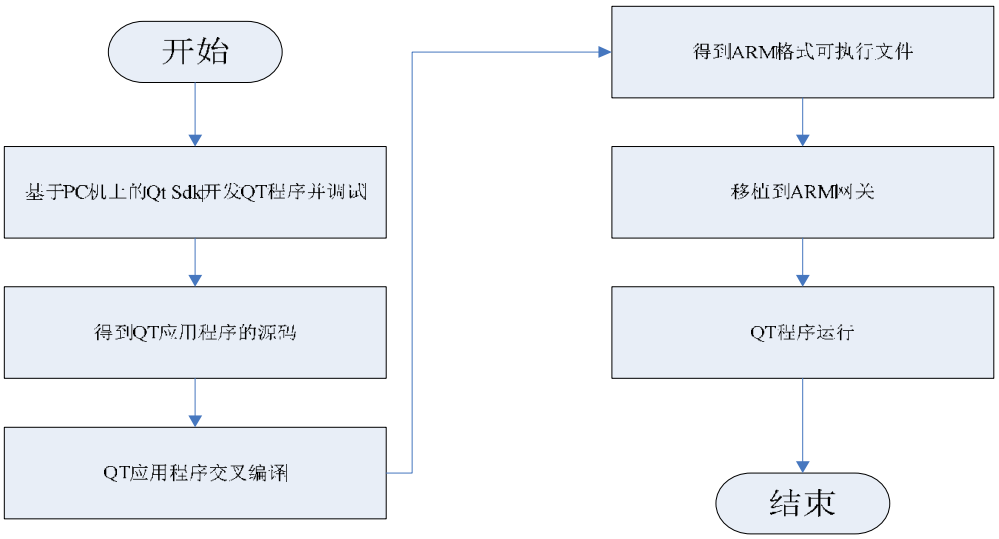
Qt 是 1991 由奇趣科技开发的跨平台 C++图形用户界面应用程序开发框架。它既可用于开发 GUI 程式，也可用于开发非 GUI 程式。Qt 是面向对象语言，易于扩展，并且允许组件编程。

4.2.2 嵌入式平台 QT 搭建



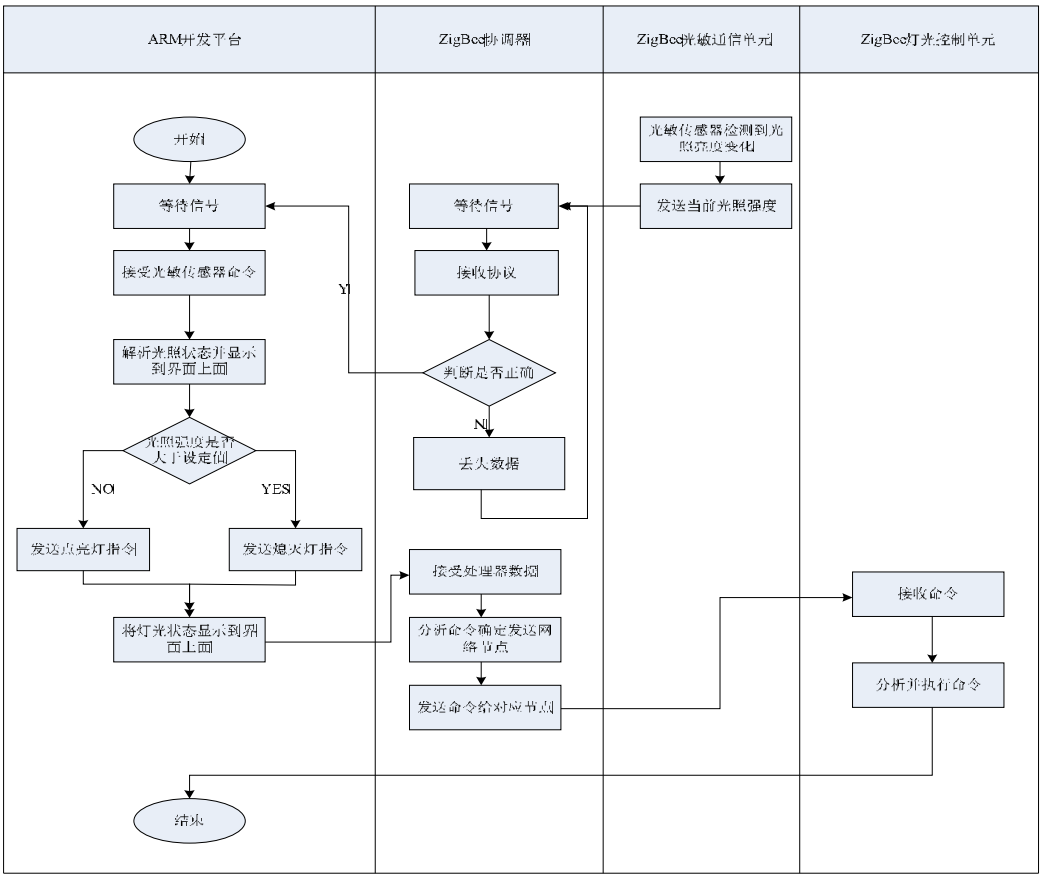
整体流程图

4.2.3 控制器端灯光控制系统



开发流程图

4.2.4 控制与通信编程



Qt 串口编程

在 Qt 中没有特定的串口控制类,而基于 Linux 系统接口自己编写串口类,涉及过于复杂,故在此次项目中采用第三方串口控制类 QextSerialPort 类。

4.2.4.1 QextSerialPort 类简介

QextSerialPort 是一个跨平台的第三方串口类，可方便的对串口进行读、写操作。本次项目中主要对应以下几个类：QextSerialBase，Posix_QextSerialPort，QextSerialPort。

QextSerialBase 类继承自 QIODevice 类，针对性的提供串口操作需要的接口，而 Posix_QextSerialPort 类继承自 QextSerialBase 类，用于 Linux 操作系统。QextSerialPort 是一个抽象类，实现跨平台性，通过宏 QextBaseType 调用不同平台上的类，实现编译。

4.2.4.2 QextSerialPort 类特性

在父类 QextSerialBase 中提供了一个枚举变量 QueryMode，可分别复制 Polling 和 EventDriven，对应两种串口读数据方式。EventDriven 使用事件驱动读取动作，当串口有数据时，立即发送 readyRead（）信号，将该信号与读串口数据的槽函数相连即可，但由于 Linux 不支持这种方式，故采用 Polling，即轮询方式读取数据，项目中定义一个定时器定时去读取

串口的数据。

4.2.4.3 QextSerialPort 类使用

1) 添加类文件

将所需的资料文件 qextserialbase.h ， qextserialbase.cpp ， Posix_QextSerialPort.h,Posix_QextSerialPort.cpp 导入 UI 编写的 smarthome 工程文件夹中，导入 Qt 工程。

2) 定义对象

3) 定义 myCom 串口对象，timer 定时器，以及一个 bool 值 isOpen 判断串 口状态；

4) 定义串口通信函数

5) 初始化函数 startinit（）：设置周期，波特率等；

6) 打开串口函数 openCom（）；

7) 关闭串口函数 closeCom（）；

8) 读串口数据函数：readMyCom（）；

9) 写串口数据函数：writeCom（）；

4.2.4.4 信号与槽实现

1) 信号与槽机制

信号和槽是一种高级接口，应用于对象之间的通信。当对象改变其状态时，信号就由该对象发射出去，信息封装，确保对象被当作一个真正的软件组件来使用，而槽用于接收信号，分为 3 种：

Public slots：任何对象都可将信号与之相连；

Protected slots：当前类及其子类可将信号与之相连；

Private slots：只有类自己可以将信号相连；

当一个信号被发射时，与其相关联的槽就会立刻执行，若一个信号与多个槽相关，则当信号发射时，这些槽会随机地全部执行，只有当所有的槽返回以后，发射函数才返回。

2) 槽函数实现

从串口读取数据后，通过信号传送给槽函数 setLightStates（bool states），槽函数收到信号发送过来的状态参数之后，经判断后，根据条件发送开关灯命令给串口设备。

3) 信号与槽函数联系

通过 connect()函数将信号 isLight(bool states)与槽函数 setLightStates(bool states)

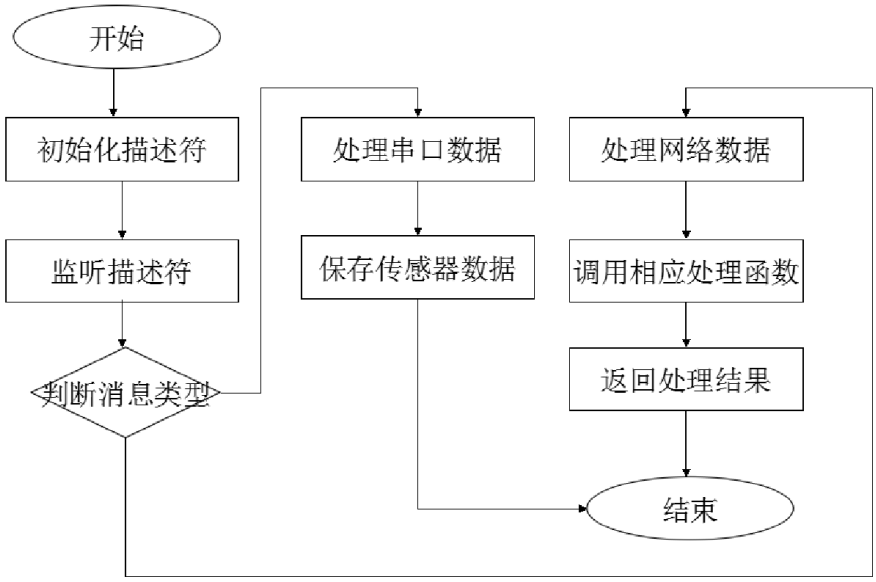
绑定，即若当前对象发送含有 bool 型的参数的信号，则由当前对象的槽函数 setLightStates 接收该信号。

4.3 服务器端程序设计

该部分需要完成的工作主要包括编写服务器端程序采集传感器数据，并完成与客户端数据的通信，具体可以划分为以下几个部分：

4.3.1 服务器端程序设计

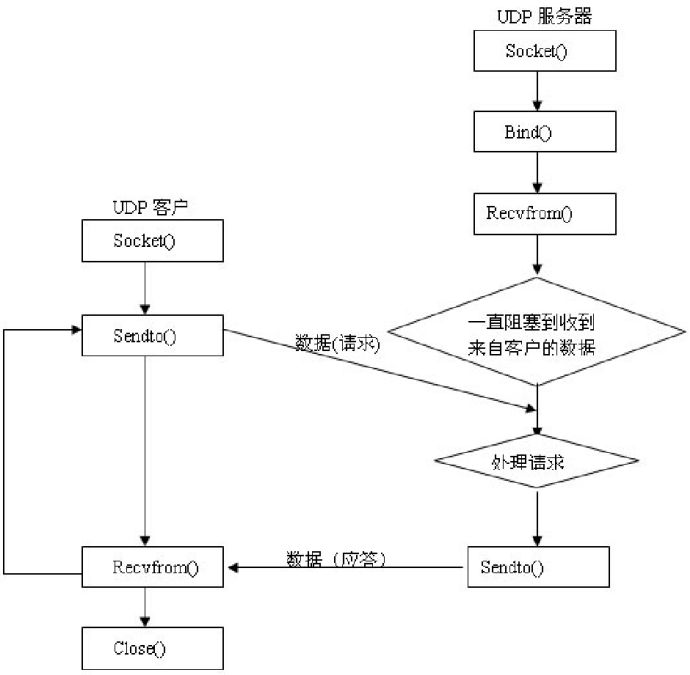
服务器端程序采用 C 语言编写，使用了 Select 系统调用实现同时对串口和网络数据的监听，首先初始化文件描述符，然后阻塞监听，直到有串口或者网络数据到达，接着进入消息处理，根据消息内容进行相关处理。



4.3.2 网络信息传输协议

UDP 协议的全称是用户数据报协议，在网络中它与 TCP 协议一样用于处理数据包，是一种无连接的协议。在 OSI 模型中，在第四层传输层，处于 IP 协议的上一层。数据发送一方（可以是客户端或服务器端）将 UDP 数据包通过源端口发送出去，而数据接收一方则通过目标端口接收数据。UDP 的特性：它不属于连接型协议，因而具有资源消耗小，处理速度快的优点,考虑到局域网不存在丢包率的问题，我们选用 UDP 协议传输数据。

UDP 协议流程图如图所示：



本系统中，手机客户端通过 UDP 协议与服务器进行数据通信，服务器端使用 Select 函数对文件描述符监听，可以处理来自手机客户端的 UDP 通信，也可以处理来自串口的传感器数据通信。服务器与客户端约定好相关数据格式，程序根据消息头部的消息类型来解析数据包的指令，然后进行相应处理。

操作 LED 消息类型可以定义为：

```
#define CS_LED_REQ          7      /* 操作 LED 信息*/
#define SC_LED_RSP          107   /* LED 返回信息*/
```

服务器端通过读取消息类型判断消息的内部结构，从而读取所需数据进行相应处理。LED 请求数据包的格式如下：

消息类型	消息长度	传输通路	数据 1	数据 2	数据 3	.....
头部			数据包实体			

4.3.3 传感器信息传输协议

目中传感器都集成了 Zigbee 模块，各终端传感器节点将数据无线发送到 ZIgbee 协调器，协调器通过串口发送给服务器端保存起来。各种传感器分别把数据封装成一定格式，定时送到 Zigbee 协调器，协调器把数据通过串口转发给服务器程序，服务器程序对数据进行解析，并

且将解析后的数据保存。

温度传感器每隔 1s 将温度数据，通过 Zigbee 协议发送给协调器，协调器接收数据并通过串口将数据发送给服务器，服务器对数据保存，示意图如图所示。

服务器端通过判断标志为断定是否为数据包，通过判断类型来确定是哪种类型传感器，从而读取所需数据进行相应处理。传感器数据包格式如下所示：

	标志	长度	目标地址	原始地址	类型	数值
温度传感器	0XFD	0X02	0X00	0X01	A	0x00(温度高位),0x00(温度低位)0c
湿度传感器	0XFD	0X02	0X00	0X02	B	0x00(湿度高位),0x00(湿度低位)%RH
光照	0XFD	0X01	0X00	0X03	C	0x01(有光),0x00(无光)
可燃气体	0XFD	0X01	0X00	0X04	D	0x01(有),0x00(无)

4.4 客户端应用的设计

4.4.1 主要功能设计

手机客户端控制软件主要用于给用户提供一个方便快捷、操作简单的一个交互平台。用户在智能手机上安装了这个软件就可以实现对与开发板互联的一些设备的控制。用户通过这个软件登陆系统后，可以查看被控制设备的工作状态，选择想要实现的功能进行相应的控制操作。

手机控制软件开发实现的功能如下：

1) 提供了一个直观方便的用户界面；

2) 可以通过局域网向服务器发送命令；

3) 可以实现对开发板上的 led 灯的开启和关闭。

4.4.2 手机客户端的功能实现模块详细设计

智能家居客户端集成多种功能，如登陆、更新、设备操作、设备控制、系统管理等，将整个软件分成各个功能模块，这样可以降低整个软件的复杂度，也可以提高软件的开发的效率。根据智能家居系统的功能需求和实现目标，整个软件的总体模块划分如下：

4.5.2.1 登录模块

主要负责处理用户的登录逻辑，通过调用网络通信模块连接服务器进行登陆操作。

鉴于本项目的重点是实现家电控制，此处未实现实际的登陆功能

登陆界面设计如图所示：

程序登录界面

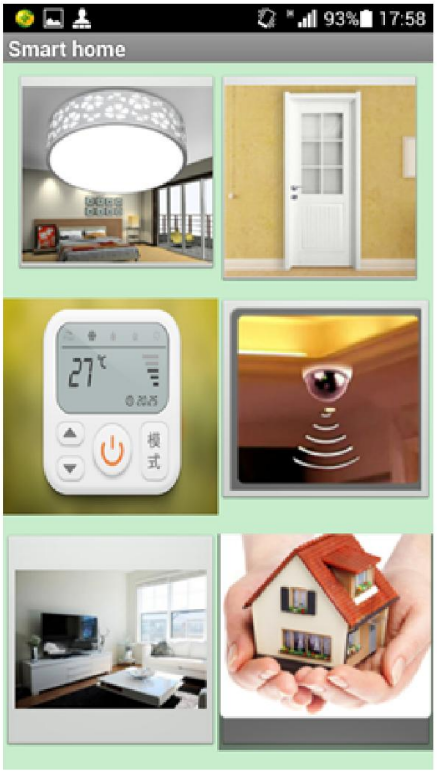
4.5.2.2 控制操作模块

用户使用客户端软件进行查看设备状态、控制设备状态等

本软件主要是通过 UDP 协议与服务器进行通信。当手机需要查看某个家庭设备的状态时，手机发送查询命令给服务器，服务器通过解析发送命令做出相应处理。智能家居系统的控制部分有查询设备、监控温度、设备管理等功能，控制器通过 UDP 协议将所查询的数据请求发送给服务器，服务器再把数据查询结果反馈给手机，手机通过解析，用文字或图片展示出来反应给客户。

将整个主界面分为六个区域，每个区域内设置一个图片按钮实现跳转到对应的功能界面，分别为：

1) 灯光控制按钮；
2) 门控制按钮；
3) 空调控制按钮；
4) 监控控制按钮；
5) 场景控制按钮；
6) 其他控制按钮。



客户端主界面

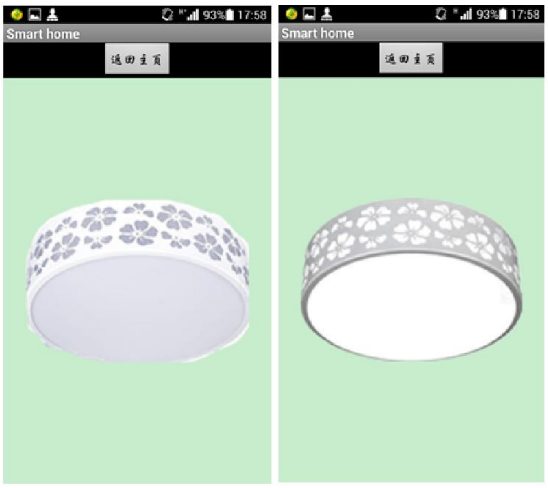
客户端主界面如上图所示。如下一一介绍该客户端的使用。

1） 灯光控制

在主界面上面点击灯光控制按钮后，跳转到灯光控制界面（下图 左）.

灯光控制主要实现对灯光的开关。返回主页按钮实现返回客户端主界面。手机上触摸控制界面上的这个灯(下图 左）以后，变成开灯界面（下图 右)，同时开发板上的灯被点亮了；再触摸一下就返回原状（下图 左)，同时开发板上的灯被关闭，可以如此反复操作。

实现开关灯的原理：路由器接入有线网，再用一根网线将路由器与开发板互联，手机连接路由器产生的 wifi。手机上点击了控制界面的灯之后，程序将启动预先编程设计好的一个线程，该线程实现发送一个 UDP 数据包到开发板的服务器，开发板上的服务器端解析数据得到控制命令后，将开启 led 灯。同理，关闭 led 灯也是如此。

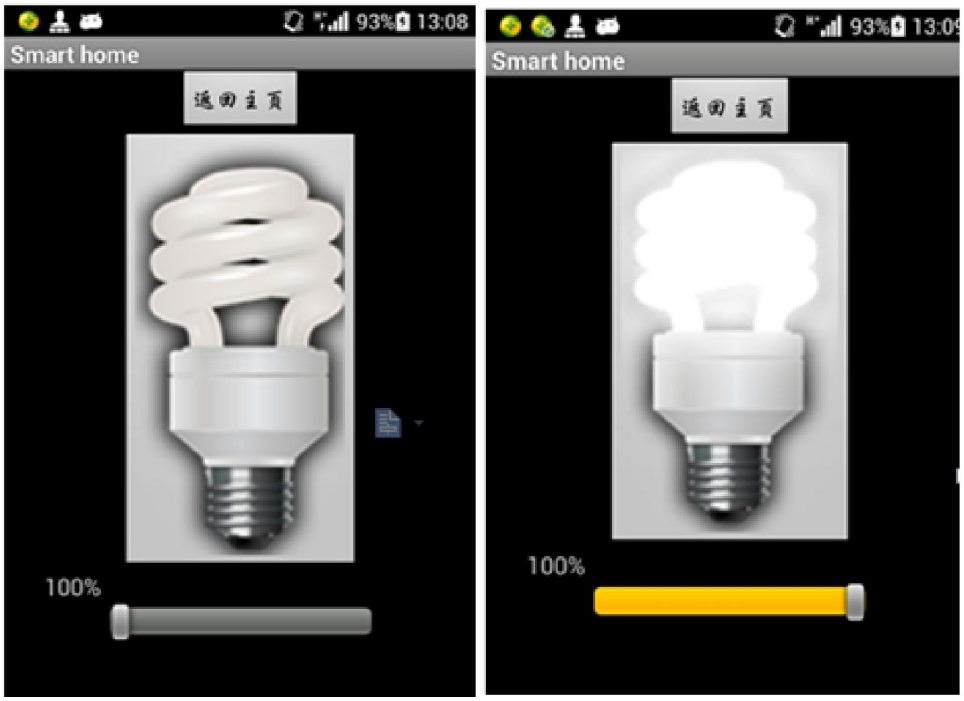


灯光的控制界面　　开灯界面

2) 情景控制

此处模拟灯光的调节，通过调节屏幕上的滚动条，达到调节灯光亮度的效果。

（仅作模拟控制界面，未实现实际控制功能）



4.5.2.3　网络编程实现数据发送、数据接收模块

1) 利用 UDP 协议将数据发送到服务器，从服务器接收反馈数据；

2) 针对 UDP 传输，Android 提供了 DatagramSocket 和 DatagramPackage 类；

3) 客户端和服务器段的通信实现；

4) 建立 Serversocket 对象，初始化服务器，等待客户端发出的连接请求；

5) 建立 socket 对象，初始化客户端，向服务器发出连接请求；

6) 服务器响应客户端并且实现服务器与客户端的连接；

7) 客户端发出命令和响应数据给服务器；

8) 服务器响应客户端的请求；

9) 服务器返回并处理从客户端所得到的结果；

10) 客户端接受服务器返回的结果；

11) 重复 4）到 7）步，直至客户端结束对话；

12) 中断连接，结束通信。

五、参考文献

无。