



# 智能家居原型系统

---

系统设计说明书

## Software Design Report

开发人员：张源峰 黄若海 李钟鑫

开发技术：android + android + websocket

开发日期：2014.10.18-2014.10.31

# 智能家居原型开发

---

## 目标：

---

- 1、实现基于ARM与Android的互联网智能家居系统.
- 2、将大学学到的技术串连起来.
- 3、完成毕业设计，积累项目经验,为找工作打基础。

## 应用场合：

- 1、夏日炎炎，在外工作一天，准备回家时，通过手机，提前将家里的空调启动，一回到家就能享受惬意的清凉。
- 2、在公司上班，亲友想用自己车库里的车，去某某地方，且钥匙还在我这，肿么办？

## 涉及技术与平台：

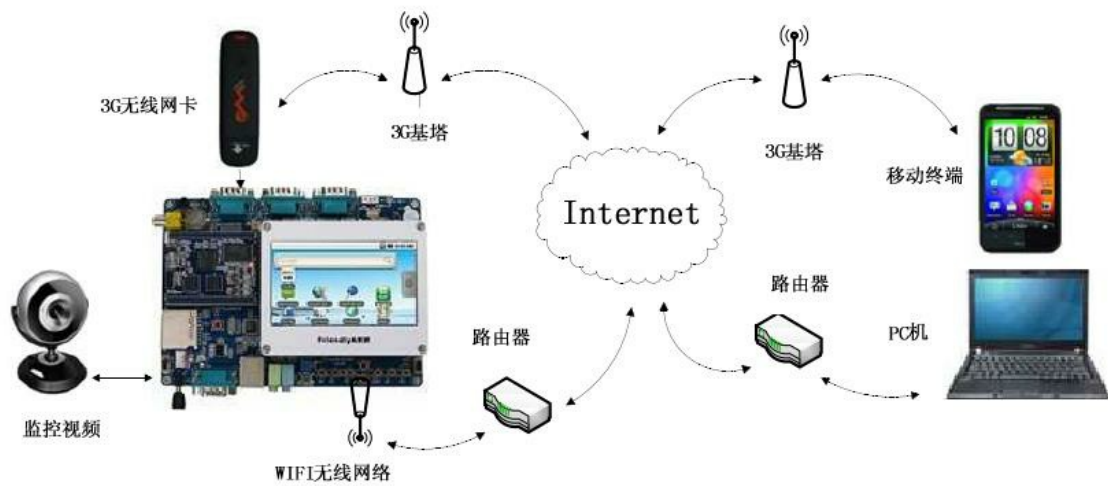
- 1、android客户端应用开发技术（应用客户端app + 遥控器客户端app）
- 2、MVC框架（抽象工厂 + 观察者模式）+ SSH / Django
- 3、网络通信(WebSocket + WebRTC + Webservice)
- 4、服务器推送技术（JPUSH）
- 5、ARM涉及模块：WIFI模块、网络通信模块、（传感器模块）
- 6、代码控制管理：git + jenkins（maven / Gradle）

## 扩展技术：

- 1、语音技术
- 2、高并发
- 3、联网安全
- 4、实时数据传输
- （5、数据挖掘）

## 原型结构图：

本系统应用模式如图 3-1 所示。



## 系统设计：

### 1\_1.系统设计与核心难点分析

### 1\_2.系统详细设计与实现

### 1\_3.介绍网站数据库设计

## 参考文献：

[基于ARM与Android的智能家居系统设计与实现.pdf](#)

[基于嵌入式Webservice的远程数据采集监控系统的设计.pdf](#)

[基于Android系统的便携式智能家居控制终端技术研究\\_应山.caj](#)

[基于Android的移动视频监控系统的设计与实现\\_余荣发.caj](#)

[基于Android的智能家居控制系统的设计与实现\\_鹿曼.caj](#)

[基于Android的智能家居系统的研究与实现\\_王朝华.caj](#)

遥控客户端功能需求：

0. 用户注册

1. 用户登录

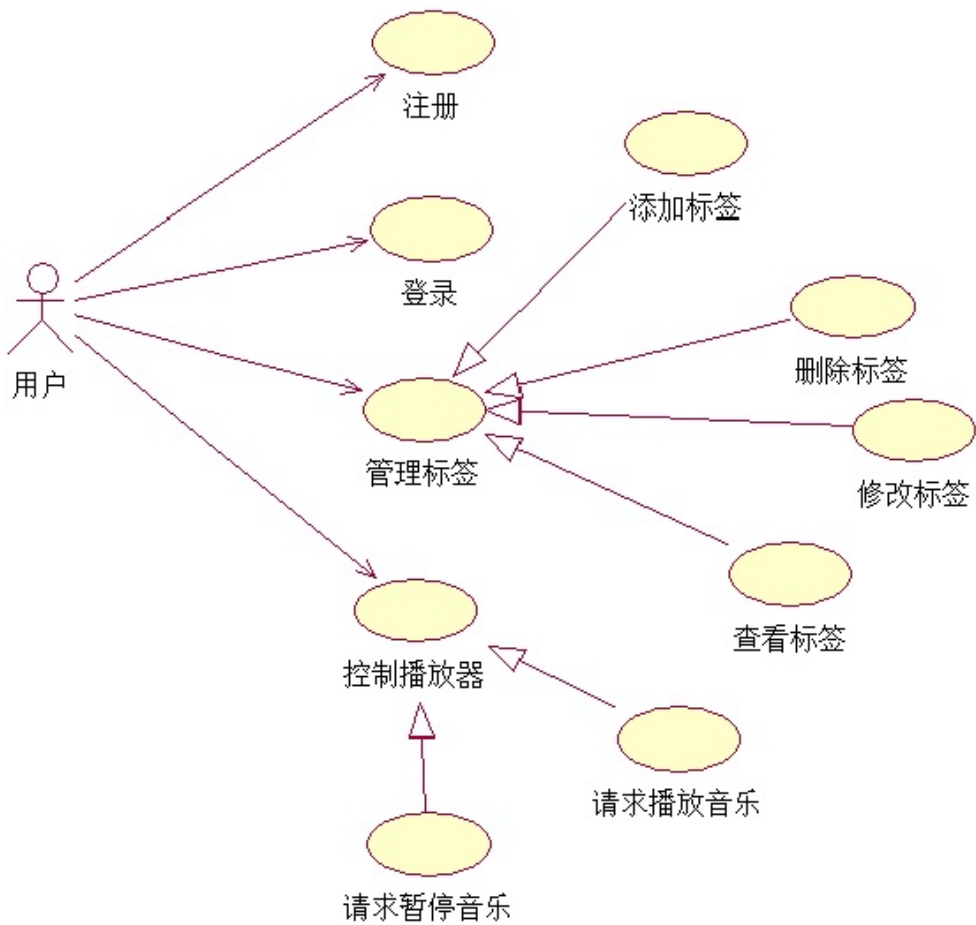
2. 管理标签

- 添加标签
- 删除标签
- 修改标签
- 查看标签

3. 控制音乐播放器

- 请求播放音乐
- 请求暂停播放音乐

遥控客户端需求用例图：



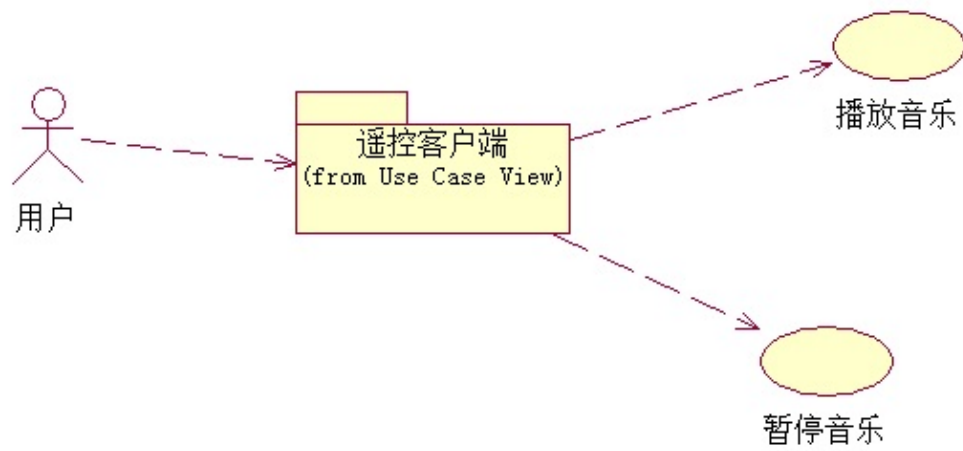
受控客户端功能需求：

## 0. 播放音乐

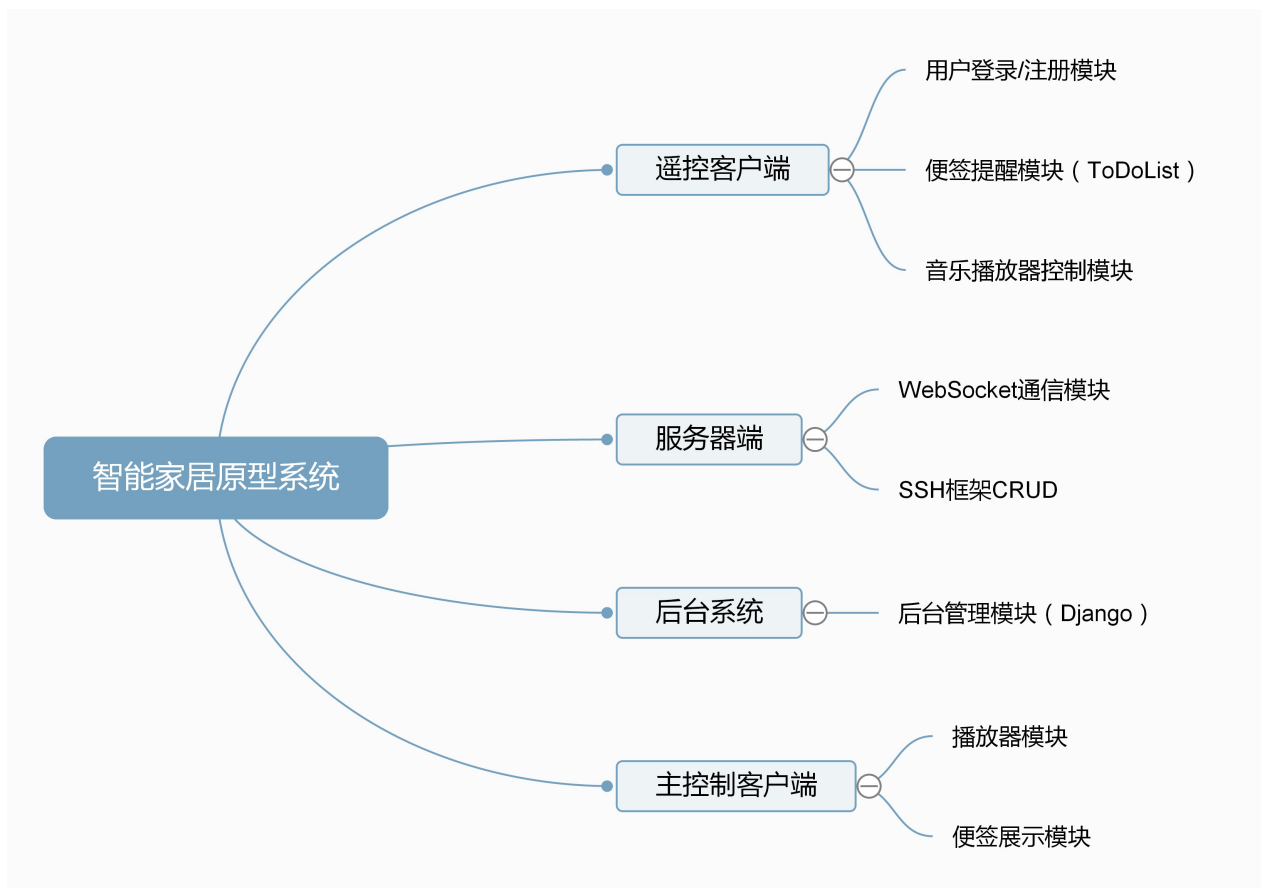
### 1. 暂停播放音乐

#### 受控客户端需求用例图

---



# 系统设计与核心难点分析



## 遥控器客户端模块任务：

1. 熟悉Android基础
2. 实现webservice请求
3. 实现json/xml的解析
4. 接受服务器传来的流媒体数据，并进行解码，播放

## 通信服务器端模块：

1. 熟悉Java基础，搭建Java环境
2. 建立JAVA的ssh框架，搭建智能家居原型网站
3. 实现webservice服务
4. 实现xmpp或者websocket服务
5. 实现推送服务

## 服务器后台管理模块：

1. 数据库设计
2. 熟悉python，建立Django框架
3. 快速实现CRUD界面化

## 智能家居控制客户端模块：

1. android实现一个音乐播放器
2. android实现打开手机摄像头，获取拍摄数据，压缩转码，再通过rtsp传回服务器，再传到客户端

## 系统详细设计与实现任务：

---

### 遥控器客户端模块任务：

---

1. 熟悉Android基础
2. 实现webservice请求
3. 实现json/xml的解析
4. 接受服务器传来的流媒体数据，并进行解码，播放

### 通信服务器端模块：

---

1. 熟悉Java基础，搭建Java环境
2. 建立JAVA的ssh框架，搭建智能家居原型网站
3. 实现webservice服务
4. 实现xmpp或者websocket服务
5. 实现推送服务

### 服务器后台管理模块：

---

1. 数据库设计
2. 熟悉python，建立Django框架
3. 快速实现CRUD界面化

### 智能家居控制客户端模块：

---

1. android实现一个音乐播放器
2. android实现打开手机摄像头，获取拍摄数据，压缩转码，再通过rtsp传回服务器，再传到客户端

# 数据库设计

---

--

## -- 表的结构 **admin**

---

```
CREATE TABLE IF NOT EXISTS admin ( admin_id int(11) NOT NULL AUTO_INCREMENT, admin_name varchar(50) NOT NULL, admin_pwd varchar(50) NOT NULL, admin_img varchar(150) DEFAULT NULL, admin_type_id int(11) DEFAULT NULL, PRIMARY KEY ( admin_id ), KEY FK_Reference_8 ( admin_type_id ) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;
```

```
INSERT INTO admin ( admin_id , admin_name , admin_pwd , admin_img , admin_type_id ) VALUES (2, 'admin', 'admin', NULL, 2);
```

--

## -- 表的结构 **admintype**

---

```
CREATE TABLE IF NOT EXISTS admintype ( admin_type_id int(11) NOT NULL AUTO_INCREMENT, admin_type_name varchar(50) DEFAULT NULL, PRIMARY KEY ( admin_type_id ) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=4 ;
```

--

## -- 转存表中的数据 **admintype**

---

```
INSERT INTO admintype ( admin_type_id , admin_type_name ) VALUES (2, '超级管理员'), (3, '管理员');
```

--

## -- 表的结构 **info**

---

```
CREATE TABLE IF NOT EXISTS info ( info_id int(11) NOT NULL AUTO_INCREMENT, admin_id int(11) DEFAULT NULL, info_type_id int(11) DEFAULT NULL, info_con text, info_title varchar(70) DEFAULT NULL, info_date datetime DEFAULT NULL, info_hits int(11) DEFAULT NULL, PRIMARY KEY ( info_id ), KEY FK_Reference_4 ( admin_id ), KEY FK_Reference_5 ( info_type_id ) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=22 ;
```

--

## -- 表的结构 **infoimg**

---

```
CREATE TABLE IF NOT EXISTS infoimg ( info_img_id int(11) NOT NULL AUTO_INCREMENT, info_id int(11) DEFAULT NULL, info_img_name varchar(100) DEFAULT NULL, info_img_path varchar(150) DEFAULT NULL, PRIMARY KEY ( info_img_id ), KEY FK_Reference_7 ( info_id ) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=43 ;
```

--

## -- 表的结构 **infotype**

---

```
CREATE TABLE IF NOT EXISTS infotype ( info_type_id int(11) NOT NULL AUTO_INCREMENT, info_type_name varchar(50) DEFAULT NULL, PRIMARY KEY ( info_type_id ) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
```



--

## -- 转存表中的数据 infotype

---

```
INSERT INTO infotype ( info_type_id , info_type_name ) VALUES (1, '简介'), (2, '系统设计'), (3, '系统实现'), (4, '数据库设计');
```

--

## -- 表的结构 user

---

```
CREATE TABLE IF NOT EXISTS user ( user_id int(11) NOT NULL AUTO_INCREMENT, user_name varchar(50)
DEFAULT NULL, user_pwd varchar(50) DEFAULT NULL, weibo_key int(11) DEFAULT NULL, qq_key int(11) DEFAULT
NULL, user_img varchar(200) DEFAULT NULL, PRIMARY KEY ( user_id ) ) ENGINE=InnoDB DEFAULT CHARSET=utf8
AUTO_INCREMENT=1 ;
```

--

## -- 表的结构 device

---

```
CREATE TABLE IF NOT EXISTS device ( device_id int(11) NOT NULL AUTO_INCREMENT, device_name varchar(50)
DEFAULT NULL, user_id int(50) DEFAULT NULL, device_key int(11) DEFAULT NULL, is_online bool DEFAULT NULL,
PRIMARY KEY ( device_id ) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

--

## -- 表的结构 project

---

```
CREATE TABLE IF NOT EXISTS project ( pro_id int(11) NOT NULL AUTO_INCREMENT, pro_name varchar(50)
DEFAULT NULL, pro_intro varchar(50) DEFAULT NULL, PRIMARY KEY ( pro_id ) ) ENGINE=InnoDB DEFAULT
CHARSET=utf8 AUTO_INCREMENT=1 ;
```

--

## -- 表的结构 task

---

```
CREATE TABLE IF NOT EXISTS task ( task_id int(11) NOT NULL AUTO_INCREMENT, task_name varchar(50)
DEFAULT NULL, task_type varchar(50) DEFAULT NULL, task_from_pro varchar(50) DEFAULT NULL, task_from_task
varchar(50) DEFAULT NULL, task_date varchar(50) DEFAULT NULL, task_tomatoes_count varchar(50) DEFAULT NULL,
PRIMARY KEY ( task_id ) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

--

## -- 表的结构 task\_detail

---

```
CREATE TABLE IF NOT EXISTS task_detail ( task_detail_id int(11) NOT NULL AUTO_INCREMENT, task_detail_from_task
varchar(50) DEFAULT NULL, task_detail_con varchar(50) DEFAULT NULL, PRIMARY KEY ( task_detail_id ) )
ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

--

## -- 表的结构 pro\_detail

---

```
CREATE TABLE IF NOT EXISTS pro_detail ( pro_detail_id int(11) NOT NULL AUTO_INCREMENT, pro_detail_from_task  
varchar(50) DEFAULT NULL, pro_detail_con varchar(50) DEFAULT NULL, PRIMARY KEY ( pro_detail_id ) )  
ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

Table of Contents

智能家居原型系统	1
系统简介	2
系统需求分析与设计	4
系统设计与核心难点分析	6
系统详细设计与实现	7
数据库设计	8