

基于 Android 的智能家居终端设计与实现^{*}

周时伟^{1,2}, 谢维波^{1,2}

(1. 华侨大学 计算机科学与技术学院, 福建 厦门 361021;

2. 华侨大学 厦门软件园嵌入式技术开发实验室, 福建 厦门 361008)

摘要: 在深入分析 Android 的系统架构及其应用程序组成的基础上, 设计和实现了一款基于 Android 的智能家居终端。终端主要包括主控制模块、信号处理模块、网络管理模块、媒体流管理模块及信息存取模块等功能模块。较好地融合了 Android 与智能家居终端, 具有较强的实用价值和开发指导意义。

关键词: 智能家居; Android 系统; 家居安防; 可视对讲; 智能终端

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2012)14-0010-04

Design and implementation of smart home terminal based on Android

Zhou Shiwei^{1,2}, Xie Weibo^{1,2}

(1. College of Computer Science & Technology, Huaqiao University, Xiamen, 361021, China;

2. Open Laboratory of Embedded Technology, Xiamen Software Park, Huaqiao University, Xiamen 361008, China)

Abstract: This paper on the basis of in-depth analysis of the Android system architecture and its application, designs and realizes an Android-based smart home terminal. The terminal includes a main control module, signal processing module, network management module, management module of the media stream, and information access module. The system better integrates the Android smart home terminal, with a strong practical value and the development of guiding significance.

Key words: smart home; Android system; home security; video intercom; intelligent terminal

自 2007 年 Google 推出开源手机操作系统 Android 以来, Android 的应用开发一时间风靡全球, 各高科技领域应用都在向 Android 进军。Android 的盛行取决于其独特的优势。首先, Android 是一款基于 Linux 内核的开源系统, 它具有操作系统、中间件和关键应用程序。对第三方应用开发完全开放、人员具有较大的自由度^[1]。其次, Android 应用程序的开发采用 Java 语言, 通过应用框架层提供的 API, 开发简单、周期短。第三, Android 具有很强的开放性和灵活性。虽然现在市场上已经出现各种各样的智能家居终端, 但 Android 的出现及其广泛应用, 使得很多智能家居厂商正投入很大的人力物力进行研究开发基于 Android 的智能家居终端。

1 Android 系统

1.1 Android 架构

Android 是一个包括操作系统、中间件和关键应用程序的移动设备软件堆。其作为一个开发式的系统平台, 包含了众多的功能和庞大的基于 Linux 的代码, 在用户

空间又分为本地代码(C/C++)和 Java 代码。Android 系统可以分为以下各层次:

自底向上依次为 Linux 操作系统及其驱动层、本地代码框架和 Java 虚拟机层、Java 框架层和 Java 应用程序层^[2]。在各层之间存在相应的分界线接口。如图 1 所示。

作为开放式的系统, Android 提供了丰富的组件和视图控件。大部分的开发只需要调用 Android 应用框架层



图 1 Android 软件系统架构

* 基金项目: 华侨大学高层次人才科研启动费项目(11BS120)

表 1 Android 各层次功能

Linux 内核及驱动层	基于 Linux2.6 内核, 主要包括安全性、内存管理、进程管理、网络协议和驱动模型。Linux 内核也同时作为硬件和软件系统之间的抽象层 ^[3] 。
Android 本地库和 Java 运行环境层	本地库包含一些 C/C++ 库, 这些库可以被 Android 系统中不同的组件使用, 它们通过应用程序框架为开发者提供服务; Java 运行环境提供了 Java 编程语言的核心库的大多数功能, 由专门为 Android 设计的 Dalvik 虚拟机和基础 Java 类库组成。
Android 应用程序框架层	包含了四种基本组件 (Activity、Service、Broadcast Receiver、Content Provider)、丰富的视图控件、资源管理器、通知管理等。
Android 应用程序层	这里的程序一般是由 Java 编写而成, 其中核心应用程序与系统一起发布, 包括常用的电话、浏览器、联系人、桌面等。

提供的 API 就可以方便地完成, 并且在 Android 的 SDK 中提供了丰富的实例供参考学习, 使得应用开发简单易行。Android 各层次的功能如表 1 所示。

1.2 Android 应用程序的组成

一个程序可以调用另外一个程序的程序片段。这是 Android 程序框架的基本设计理念。例如, 在 Windows 操作系统中, 资源管理器可以浏览图片文件, 当浏览到这个图片文件时, 双击该图片文件, 便启动图片浏览软件。Android 的程序理念便是如此, 所不同的是, 在 Windows 中, 打开图片会启动图片浏览软件的全部功能, 虽然实际上只是需要查看图片; Android 中是把一个程序的功能分为各种不同的程序片段, 每个程序片段都能单独启动。

Android 应用程序由 Activity、Service、Broadcast Receiver 和 Content Provider 四大组件构成。这四大组件也称为 Android 的四种类型的程序片段, 且每种类型都可以独立运行。

Activity 在四大组件中最复杂。从视觉上, 一个 Activity 占据当前的窗口, 响应所有窗口事件, 具有控件、菜单等界面元素, 为用户提供一个与界面交互的平台。从内部逻辑来看, Activity 需要为了保持各个界面状态, 需要做很多持久化的工作, 还需要妥善管理生命周期和一些转跳逻辑。

Service 就是剥离了界面的 Activity, 都是封装有一个完整的功能逻辑实现, 只不过 Service 不抛头露面, 默默提供后台服务。举例, 当用户处理其他问题时, 服务可以播放背景音乐, 或者可以从网上获取数据、计算、提高, 提供给活动所需要的结果。

Broadcast Receiver 只能接收广播, 一个广播可以有多个广播接收器对它认为重要的通知作出反应。Android 系统提供了一个 sendBroadcast() 方法, Application Framework 内部调用该方法发送广播, 比如有新短信息、有来电等广播消息, Broadcast Receiver 会响应这些系统消息的独立执行单元^[4]。

Content Provider 提供了一套特殊的可供其他应用使用的数据库。这些数据可以存储在文件系统中、SQLite 数据库中或者其他有意义的形式。实现了一套标准的允许

其他用户检索、储存数据的方法。通过使用 Content-Provider 的实例, 调用它的方法。

Intent 是 Android 中一个特殊的类, 组件之间的通讯, 由 Intent 协助完成。Intent 负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述, Android 则根据此 Intent 的描述, 负责找到对应的组件, 将 Intent 传递给调用的组件, 并完成组件的调用。Intent 在这里起着实现调用者与被调用者之间的解耦作用, 如图 2 所示。

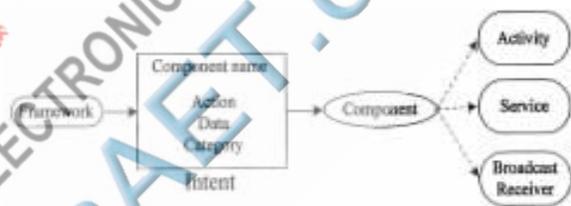


图 2 Intent 的解耦作用

2 智能家居终端

2.1 终端主要功能

智能家居终端的主要功能如下^[5]:

(1) 可视对讲。室内机与门口机进行可视对讲; 室内机可以监视门口情况; 访客留影; 两个或多个室内机进行视频对话。

(2) 家居安防。防区设定: 对室内和室外安装的监视器进行独立的设定; 报警: 监视器侦测到异常发出报警; 家居安防模式设定: 外出、在家、就寝等。

(3) 家居控制。灯光控制; 窗帘控制; 空调控制; 电源控制; 燃气阀控制; 电梯控制。

(4) 生活资讯。天气预报; 万年历; 常用电话; 物业服务。

(5) 系统管理。常用设置; 使用说明; 安防设置; 家居设置。

系统结构功能如图 3 所示。

2.2 软件系统架构

整个软件系统架构分为四个层次, 每个层次的功能分别如下:

(1) 操作系统层。对下为各种硬件设备提供外围设备驱动, 控制硬件工作, 如音视频的采集播放设备、各类

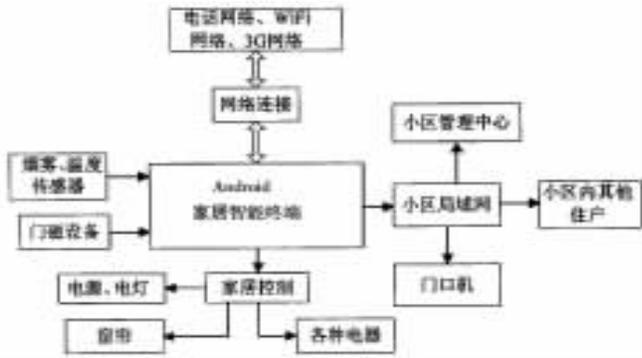


图3 智能家居系统结构



图4 主程序中的布局

传感器、摄像头等;对上为库函数提供各种设备的驱动接口,以使得上层可以调用设备。

(2)库和接口层。该主要提供给应用框架层一些基本的类库和接口,包括 Android 的 C/C++库、RTP/RTCP 库和多媒体的处理接口。

(3)应用框架层。该层采用模块化开发方法,通过使用其下层的函数库提供的接口,将实现相同或相似功能的封装成为一个模块进而供应用层来开发使用。

(4)应用层。通过与应用框架层提供的模块相互交互、资源分配等,基于 Android 的四大组件及其他相关视图控件完成界面设计,为用户提供一个可视的界面平台。在该软件架构下,涉及到的功能模块有信号处理模块、网络管理模块、媒体流管理模块以及信息存取模块。

2.3 功能模块的实现

2.3.1 主控制模块

最上层为主控制模块,是系统的主要模块,通过使用 Android 的各种组件和控件,在应用层实现系统的主要功能。一方面,通过 Android 的资源文件如布局、菜单、视图等提供的各种类和方法来完成系统的 GUI 设计,提供给用户一个统一的图形化接口给用户。另一方面,通过使用应用框架中的信号处理、网络管理、媒体流管理、信息存储等功能模块接口,分别实现门磁控制、家居安防、家居控制、生活资讯、可视对讲及系统管理的功能。在此模块中的门磁控制可以直接控制房门的开关^[6]。

所有的功能都聚集在主模块中,这就需要有一个合理的布局才能使系统使用起来比较协调。由于每个 Activity 中包含一个 Window,每个 Window 中包含一个 View Group,而每个 View Group 中包含更多的 View 和 View Group。在主模块中的布局如图 4 所示。

2.3.2 信号处理模块

信号处理模块是为实现家居安防和家居控制来提供服务的。主要是接收一些传感器信息,然后针对各种传感器做出相应的处理,或者是将内部命令发送给各个外围传感器。在 Android 操作系统中内置了很多传感器,在部分软件中可以自动识别屏幕的横屏、竖屏方向来改变屏幕显示布局。下面是 Android 支持的几种传感器:

SENSOR_TYPE_LIGHT:亮度传感器;
SENSOR_TYPE_PRESSURE:压力传感器;
SENSOR_TYPE_TEMPERATURE:温度传感器;
SENSOR_TYPE_ORIENTATION:方向传感器。

这些传感器都为应用程序提供了 API,通过 API 的调用就可以方便地实现各种需求,如温度超过某一值时自动报警等。对于烟雾传感器等外围设备送来的信号,在系统中把 Message 交付给相应的 Handler 处理,输出最终结果。

如果需要使用一些外围设备如关闭电视、空调等家用电器,在系统可以触发一个信号,经由 Handler 处理之后发送到相应的智能设备。消息处理过程如图 5 所示。

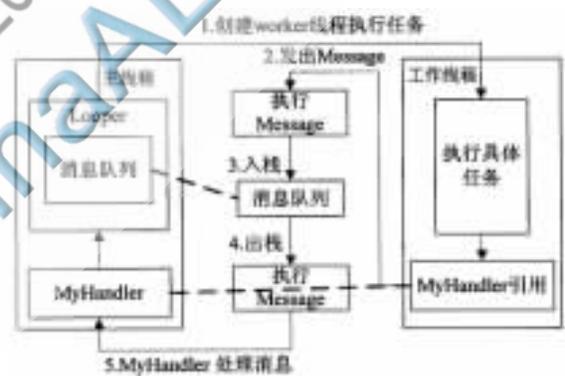


图5 消息处理过程

2.3.3 网络管理模块

网络管理模块主要为生活资讯提供服务。此模块提供服务的分为两种。一种为局域网服务,如物业服务。这里实现起来比较简单,用户根据系统所提供的设备名称向中心管理机报备需要服务的设备。例如:点击某个设备,终端向中心管理机发送一条“智能设备需要服务”指令。物业收到该指令后,会主动联系该用户。另一种为 Internet 服务,需要提供各种信息的 Web 服务,如天气。

2.3.4 媒体流管理模块

媒体流管理模块是为可视对讲提供服务的。在系统中通过调用 Android 的 OpenCORE 接口来实现音视频的编解码。OpenCORE 是 Android 多媒体框架的核心,所有 Android 平台的音频、视频的采集以及播放等操作都是

通过它来实现的。它主要包含了两个大方面的内容:

PVPlayer:提供媒体播放器的功能,完成各种音频、视频流的重放功能。

PVAuthor:提供媒体流记录的功能,完成各种音频、视频流以及静态图像的捕获功能。

当门口机需与室内机请求对讲时,室内机只需要对门口机传输过来的音视频进行解码播放,即 Android 调用 **MediaPlayer** 类用来播放音频、视频。

当两个室内机进行视频对讲时,双方机器要同时进行音视频的编码解码。这时就要同时调用 **MediaPlayer** 类和 **MediaRecorder** 类。**MediaRecorder** 类用来进行媒体采样,包括音频和视频。

2.3.5 信息存取模块

信息存取模块主要为系统管理提供开发接口以实现系统的管理。如智能家居系统的终端系统管理模块主要提供本机设置、安防设置、家居设置、使用帮助等功能。本机设置包括本机的网络设置、楼栋单元号房间号设置、密码设置等;安防设置可以对室内的各个安防设备进行布防和撤防设置;家居设置则可以调整家用电器的断电或启动时间;使用帮助则是关于本机的使用说明。设置的一般过程为先通过相应的 **get** 方法获得要设置的状态,然后通过 **set** 方法保存设置。

2.4 模块封装

本文开发是按照 Android 开发规范进行,应用层使用 Java 语言,在底层的一些本地库和接口开发过程中,考虑到代码的使用效率及开发的简易性,采用了 JNI,是由 C/C++ 语言实现的。如图 6 为实现效果图。

整个系统充分利用了 Android 内部资源,极大地提高了开发效率。测试结果表明系统的稳定性极高,具有

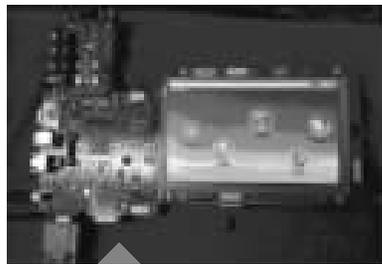


图 6 智能家居界面效果图

较强的实用性。

参考文献

- [1] 邵长彬,张重阳,郑世宝.基于 Android 的可视 IP 电话多媒体终端设计与实现[J].电视技术,2011,3(6):62-65.
- [2] 柯元旦,宋锐.Android 程序设计[M].北京:北京航空航天大学出版社,2010:12-35.
- [3] 尹浩,陈美鑫,黎飘.基于 FPSLIC 的网络家居终端系统的设计与实现[J].计算机工程与设计,2006,27(8):1442-1445.
- [4] 杨丰盛.Android 应用开发揭秘[M].北京:机械工业出版社,2009:188-219.
- [5] 金鑫.基于 ARM-Linux 的智能家居网络终端的设计与实现[J].电脑知识与技术,2010,6(7):1612-1618.
- [6] 宋小青,周东升.基于 Android 平台的应用开发研究[J].软件导刊,2011,10(2):104-106.

(收稿日期:2012-03-29)

作者简介:

周时伟,男,1985 年生,硕士研究生,主要研究方向:嵌入式技术。

谢维波,男,1964 年生,教授,博士,硕士生导师,主要研究方向:嵌入式技术、数字信号处理。