

# 基于 Android 的电话拨号功能<sup>①</sup>

杜江, 周渊平

(四川大学 电子信息学院, 成都 610065)

**摘要:** Android 是一个开放性的平台, 它具有非常丰富应用. 将 Android 平台和电话联合起来研究, 设计一个在 Android 平台上拨打和接听电话的应用, 使 Android 平台具有电话拨号的功能. Android 平台和电话通过 CD4052 模拟开关组成的控制电路进行连接. 通过在 Android 的 Linux Kernel 中编写驱动、在 HAL 和 JNI 中生成动态库、以及在应用层创建 Application, 实现 Android 系统电话拨号功能. 经测试, 设计的拨号应用能成功拨通号码.

**关键词:** Android; 电话; 拨号; Linux Kernel; HAL; JNI

## Telephone Dialing Function Based on Android

DU Jiang, ZHOU Yuan-Ping

(College of Electronic Information, Sichuan University, Chengdu 610065, China)

**Abstract:** Android is an open platform. It has very rich applications. This paper combines Android platform and telephone to study and design an application of making and receiving calls on the Android platform to make it has the function of telephone dialing. Android platform and telephone are connected by the control circuit with analog switches CD4052. The telephone dial-up function in Android system can be implemented by writing a driver in Linux kernel, generating dynamic libraries in JNI and HAL and creating application in the application layer. After testing, the design of the dial-up application can successfully dial the number.

**Key words:** Android; telephone; dialing; Linux Kernel; Hal; JNI

Android 是由 Google 公司和开放手机联盟领导及开发的一种基于 Linux 的自由及开放源代码的操作系统, 在 Google 发布了 Android 的源代码后, Android 的应用逐渐从移动设备扩展到诸多领域.

TQ210 开发板搭载了 Android4.0 系统, 它使用三星公司的 S5PV210 芯片作为处理器<sup>[1]</sup>, 该芯片使用的是 Cortex-A8 的架构, 性能高功耗低, 能够满足绝大多数设备的需求, 并已在通信系统、医疗电子、工业控制等场合广泛应用.

电话已成了绝大多数企业、政府机关和公共事业机构所必需的办公工具. 它除了信号稳定、抗干扰能力强、话音清晰优点外, 相比于移动设备, 它辐射更小. 在如今移动设备功能越来越多的背景下, 将 Android 与电话有效结合起来研究, 实现 Android 电话拨号功

能, 将 Android 上实用的应用赋予电话, 丰富了电话的应用功能.

## 1 硬件系统结构设计

### 1.1 硬件框架

硬件系统主要由 Android 平台、控制电路、电话模块三部分组成, 如图 1 所示. Android 平台采用 TQ210 开发板, 它采用了底板加核心板的结构. 核心板采用了 ARM Cortex-A8 内核的 S5pv210 微处理器, 主频可达 1GHZ, 同时配置了 1G 的 DDR2 和 1G 的 SLC Flash, 具有强大的应用功能. 底板上扩展出了串行接口、100M 网络接口、USBHOST 接口、LCD、CAMERA 接口、SPI、SD 卡等外围电路<sup>[2]</sup>, 功能强大, 实用. Android 平台 GPIO 接口连接控制电路,

①收稿时间:2014-04-22;收到修改稿时间:2014-05-12

控制电路连接电话模块。

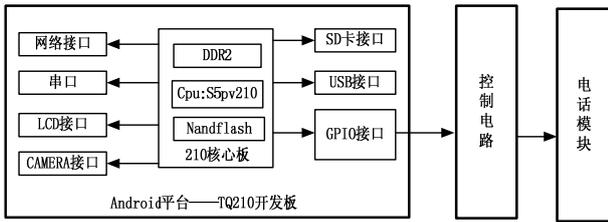


图 1 系统框架图

### 1.2 电话模块

电话的拨号电路采用 HM9102D 音频/脉冲可转换的拨号器, 引脚排列如图 2 所示, 它有上次号码重拨 (LNB) 功能. 在挂机状态下, 消耗保持电流很小. R1-R4、C1-C4 为键盘行、列输出引脚, 提供键盘扫描和拨号功能, 只能单键键入, 同时按下两个或两个以上的键将无效, 为避免键盘接点颤动而出错, 芯片内部设有防颤电路, 防颤时间为 20ms.

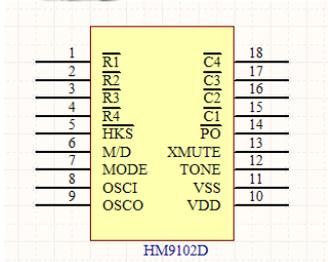


图 2 HM9102D 引脚排列图

### 1.3 控制电路设计

控制电路是由四片模拟开关 CD4052 及两个接口 H1、H2 构成. CD4052 是一个差分 4 通道数字控制模拟开关, 有 S0、S1 两个二进制控制输入端和 INH 输入. H1 为电话拨号器 HM9102D 的 C1-C4、R1-R4, 8 个引脚接口. H2 即是 TQ210 开发板的 GPIO 接口, 控制 CD4052 的 INH、S0、S1. 原理图如图 3 所示.

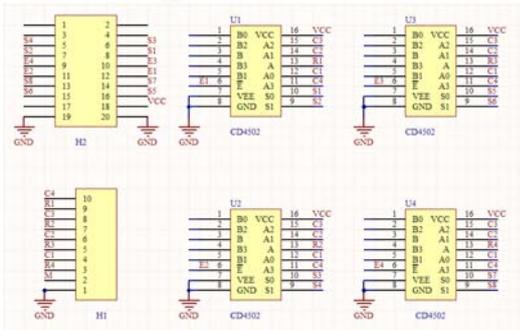


图 3 控制模块原理图

## 2 软件部分

### 2.1 编译 Android 系统所需的软件平台

由于编译 Android 文件系统需要很大的内存和较高的 cpu 资源, 为了能顺利的在短时间内编译出所需的文件系统, 计算机配置了 inter 第三代酷睿 i3 处理器, 另外配置了 8G 的内存, 操作系统使用了 64 位的 ubuntu 11.10, 在 ubuntu 系统中安装 arm-linux-gcc 交叉编译器、java、Eclipse、samba 服务器等软件<sup>[3]</sup>.

### 2.2 Android 的系统架构

Android 的系统采用了分层架构, 整个架构共有四层, 从上层到下层分别是 Application、Application Framework、Libraries and Android Runtime 和 Linux Kernel<sup>[3,4]</sup>. Android 系统架构如图 4 所示. 为了实现硬件设计所达到的功能, 需要分别在 Linux Kernel 中编写驱动文件、在中间层生成动态库文件以及在顶层编写应用程序.

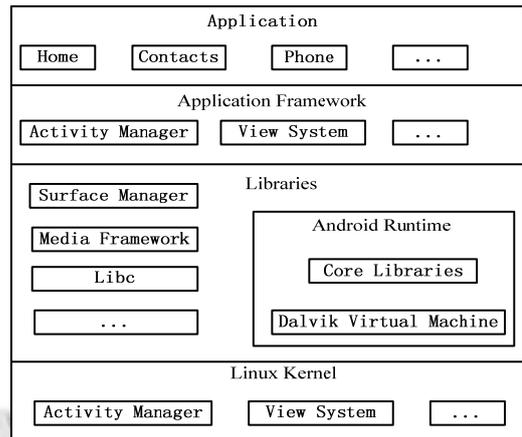


图 4 Android 系统框架图

### 2.3 Linux Kernel 驱动程序

在 Linux 系统中加载驱动程序有两种方式: 静态加载和动态加载. 前者编译生成.o 文件, 直接编译进内核中; 后者编译生成.ko 文件, 可以在系统启动后通过命令对驱动模块进行加载或卸载<sup>[5]</sup>. 由于系统是对 GPIO 口进行操作, 采用 Linux 字符设备驱动程序<sup>[5]</sup>, 将编写好的驱动程序 wiretelephone.c 放入/driver/char 目录下, 并修改该目录下的 Kconfig 文件和 Makefile 文件<sup>[6]</sup>. 在内核文件的根目录下, 使用 menuconfig 命令, 在相应的路径下即可看到新写的驱动. 通过命令: make zImage, 即可制作内核镜像文件 zimage.bin.

### 2.4 JNI--JAVA 本地接口

Java Native Interface (JNI)标准是 java 平台的一部分,它允许 Java 代码和其他语言写的代码进行交互,使得在 Java 虚拟机 (VM) 内部运行的 Java 代码能够与用其它编程语言编写的应用程序和库进行交互操作,起到承上启下的作用. 编译 JNI 层库文件具体操作方法如下:

1)在 Android 文件系统下的 packages/apps 下创建一个名为: libWiretelephone 的文件夹,再在该文件夹中创建 jni 文件夹,在 jni 文件夹中创建 WiretelephoneService.cpp 和 Android.mk 文件. 在 Wiretelephone Service.cpp 文件中通过 JNIInvokeInterface 来完成 Java 和 C 函数的映射. 使用 register\_android\_server\_WiretelephoneService (JNIEnv \*env)完成设备注册等.

2) 在文件系统的主目录下运行命令: source build/envsetup.sh 建立环境, export TARGET\_PRODUCT=full\_tq210 设置平台, 在这里只需要在文件系统的主目录下运行命令: mmm packages/apps/ Wiretelephone/jni/ 在指定的输出目录下生成 libwiretelephone.so 动态共享库文件, 并将该文件拷贝到 out/target/product/tq210/rootfs\_dir/system/lib 目录下.

### 2.5 HAL--硬件虚拟层

Android 的 HAL 是为了避开 linux 的 GPL 束缚,保护一些硬件提供商的知识产权而提出的,思路是把控制硬件的动作都放到了 Android 的 HAL 中,而 linux driver 仅仅完成一些简单的数据交互作用,甚至把硬件寄存器空间直接映射到 user space. 目前 HAL 存在两种构架,位于 libhardware\_legacy 目录下的“旧 HAL 架构”和位于 hardware 的 libhardware 目录下的“新 HAL 架构”,TQ210 使用“新 HAL 架构”,如图 5 所示.

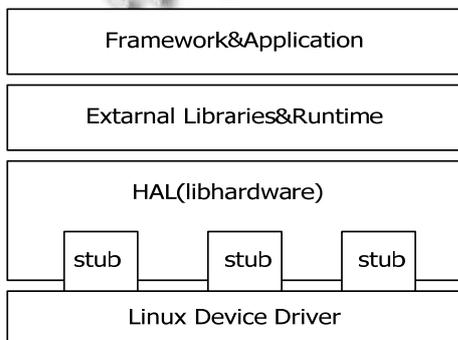


图 5 新 HAL 架构

编译 HAL 层库文件具体操作方法如下:

1)在 hardware/libhardware/include 目录下创建 wiretelephone.h 头文件.

2) 在 device/embedsky/tq210 目录下创建 libwiretelephone 文件夹, 在该文件夹中添加 wiretelephone.c、Android.mk 两个文件.

3) 由于在 JNI 层已经设置了环境和平台, 使用命令: mmm device/embedsky/ tq210 /lib wiretelephone 在输出目录下生成 wiretelephon e.tq210.so 动态共享库文件, 并将该文件拷贝到 out/target/product/tq210/rootfs\_dir/system/lib/hw/目录下.

Android 的 HAL 架构中程序的系统结构如图 6 所示: 其中 Wiretelephone App 是我们编写的拨号应用程序. App 中的 java 代码不能直接操作硬件, 将硬件操作工作交给本地 module 库 Wiretelephone\_runtime.so, 它从当前系统中查找 Wiretelephone Stub, 查找到之后, Wiretelephone Stub 将硬件驱动操作返回给 module, Wiretelephone App 操作硬件时, 通过保存在 module 中的操作接口间接访问底层硬件.

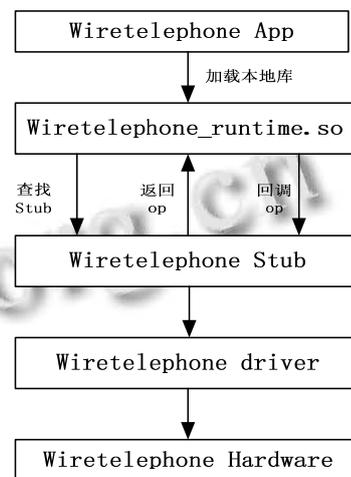


图 6 HAL 架构中程序的系统结构图

### 2.6 制作 Android 系统镜像文件

由于编译文件系统耗时较长, 在搭建软件平台时, 已经成功的编译出了 Android4.0 的系统文件 Rootfs\_dir, 这里不需要再次编译, 在检查 libwiretelephone.so 和 wiretelephone.tq210.so 两个库文件已经顺利的拷贝到 Rootfs\_dir 目录的相应位置后, 在 out/target/product/tq210/ 目录下运行命令: mkyaffs2image\_for\_TQ210 rootfs\_dir/ root\_android.

bin,将在该目录下生成 root\_android.bin 镜像文件。

### 2.7 使用 SD 卡烧写内核和文件系统

用工具将 SD 卡制作成启动卡,并将做好的内核镜像文件 zImage.bin 和文件系统镜像文件 root\_android.bin 拷贝到 SD 卡,将开发板设置成从 SD 卡启动,上电即可看到系统和内核的加载。在文件系统加载完成后,关掉电源,将开发板改成从 Nandflash 启动,重新上电即可,在烧写时连接开发板串口 COM1 与 PC,通过串口软件可查看烧写信息,便于调试和分析错误。

### 2.8 APP 层 UI 界面设计和源码实现

在 Ubuntu 的 Eclipse 中创建一个工程名为 Wiretelephone 的 Android 工程<sup>[7]</sup>。在该工程中,使用 layout 文件夹下的 mian.xml 文件控制 UI 界面,布局方式选择了表格布局<sup>[7]</sup>,表格布局与常见的表格类似,以行、列的形式来管理放入 UI 组件,如图 7 所示。

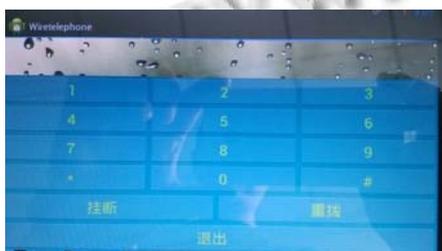


图 7 UI 界面效果图

在 java 源码中主要分为三个步骤:加载库文件 lib wiretelephone.so、定义本地接口函数、设置事件监听器,对按钮点击事情进行处理。完成 UI 界面设计和源码实现后,在 Eclipse 中选择运行 Run As Android Application,在 bin 目录下就会得到一个 Wiretelephone.apk 的应用程序<sup>[6]</sup>。将其安装到 Android 平台即完成 APP 层的开发工作。

## 3 测试系统

测试系统能否正常工作分为以下几步:

1)将 Android 平台、控制模块、电话通过接口连接起来,向有线电话拨号。经测试能顺利拨通。

2)点击桌面的 Wiretelephone 图标,即可出现拨号界面,通过点击按钮进行拨号,经测试能顺利拨通,且语音质量良好,在通话完成后点击按钮“挂断”即可结束通话。

3)点击“重拨”按钮,能够顺利的对最近一次拨出

的号码进行重拨。

经上述测试,该系统能实现拨号功能,拨号测试如图 8 所示,手机显示的来电,正是通过设计的拨号应用程序拨出的。



图 8 拨号测试

## 4 结语

经过上述测试,论证了该设计方案是可行的,通话时语音信号稳定可靠、无延迟。该方案即有电话辐射小、抗干扰能力强、实时、准确的通信的特点,又具有 Android 平台开放性、丰富的硬件选择、软件资源丰富等特点。从而使单一的打电话的功能,扩展到很多方面。比如通讯录、备忘录、时钟、记事本、天气预报、室内温度、Google 地图定位等与办公、生活先关的软件都可以融入到系统中,丰富有线电话的实用功能。

### 参考文献

- 1 陈文震,卞卫锋,张亚伟.Android 系统下的 LCD 驱动移植.仪表技术,2012,(4):16-19.
- 2 熊积健,王琪.基于 S5PV210 平板电脑的设计.江西通信科技,2012,(1):12-15.
- 3 姚昱旻,刘卫国.Android 的架构与应用开发研究.计算机系统应用,2008,(11):110-112.
- 4 宋小倩,周东升.基于 Android 平台的应用开发研究.软件导刊,2011,(2):104-106.
- 5 宋宝华.Linux 设备驱动开发详解.第 2 版.北京:人民邮电出版社,2010.
- 6 付兴武,张军.基于 SPI 总线协议的字符设备驱动程序.计算机系统应用,2013,(2):146-150.
- 7 明日科技.Android 从入门到精通.北京:清华大学出版社,2012.