

The Design of Electronic Commerce Terminal Based on Android

Hanmi Lin¹, Yan Qian²

¹hangzhou normal university, Hangzhou, China

²hangzhou normal university, Hangzhou, China

Email: ccnupolaris@163.com grpwork@163.com

Abstract: This document presented the property of the android operation system, and the requirements and the hardware design of electronic commerce terminal, described the operation system porting method, and how to develop the driver program and the application program.

Keywords: android; electronic commerce; operation system; porting

基于 android 的电子商务终端设计

林菡密¹, 钱言²

¹杭州师范大学钱江学院, 杭州, 中国, 310012

²杭州师范大学钱江学院, 杭州, 中国, 310012

Email: ccnupolaris@163.com, grpwork@163.com

摘要: 本文主要讲述了 android 操作系统的特点, 电子商务终端的功能需求, 详细阐述了基于 android 操作统的电子商务终端的硬件结构设计, 操作系统移植方法, 主要的驱动程序及应用程序的开发, 完整介绍了电子商务终端的开发过程。

关键词: android; 电子商务; 操作系统; 移植;

1 引言

电子商务已经深入到我们的日常生活中, 让更多的人有更多的机会接触互联网成为电子商务的参与者是电子商务普及的重要一环。当前电子产业发展迅速, 微型处理器的性能已经能很好的满足日常使用需求, 另外各种嵌入式操作系统也为发展便携可移动产品提供良好的软件平台, 采用微型处理器和嵌入式操作系统制作通用的电子商务终端对电子商务的发展有重要的促进意义。移动电子商务终端要求良好的用户交互界面, 快速方便的网络接入, 较长的待机时间, 同时用户可以方便的升级程序, 安装新的引用, 综合以上需求, 我们采用了 ARM cortex-A8 处理器和开源的 android 操作系统对终端设计进行了深入的研究和实践。

2 系统设计框架

2.1 硬件框架

目前在移动设备市场上, ARM 处理器占据了绝对的优势, ARM 处理器使用精简指令集, 功耗低, 嵌入

式操作系统支持完善, 丰富的技术开发资源, 所以我们采用了基于 ARM cortex-A8 核的 OMAP3430 处理器; 为了更好的功耗表现, 采用了 TWL5030 电源管理芯片, TWL5030 同时具有键盘扩展功能, 电池充电功能; 在输入输出设备上我们采用了 5.7 寸 TFT 液晶显示屏及 I2C 接口的电阻式触摸屏, 并外扩了 64 个输入键盘; 电源采用锂电池, 由 TWL5030 负责电池电量及充放电管理; 网络连接采用 WIFI 接入, 通过 SDIO 接口和 OMAP 相连。

模块图如下:

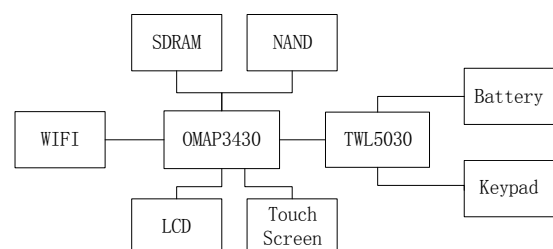


图 1 硬件结构图

2.1 软件框架

软件操作系统采用了 google 公司推出的 android 操作系统，android 系统基于 linux 内核，完全开放源代码，具有开放性，应用程序开发方便，具有兼容的浏览器，能快速的访问互联网^[1]。Android 体系结构如下图 2 所示。



图 2 android 系统结构图

应用层是用 Java 语言编写的运行在虚拟机上的程序；应用框架层是 Google 发布的核心应用时所使用的 API 框架，开发人员同样可以使用这些框架来开发自己的应用，这样便简化了程序开发的架构设计；系统运行库层是 C/C++ 库用来支持应用框架层。

Android 的操作系统使用的是 Dalvik 虚拟机，每个 Java 程序都运行在该虚拟机之上，Dalvik 虚拟机不需要很快的处理器速度和大量的内存空间，非常适合在移动终端上使用。

Linux 内核层是基于 linux2.6 内核，如安全性，内存管理，进程管理，网络协议栈和驱动模型都依赖于该内核。Linux 内核同时也作为硬件和软件栈之间的抽象层。

3 操作系统的移植

从上面 android 系统的框架结构可以看出，移植 android 操作系统需要对应硬件平台做出修改的是 linux 内核层，我们使用的移植方法是将 android 源代码下载，同时根据已有的 OMAP3430 的 linux 操作系统代码对 android 中的 linux 内核层进行修改，这一步骤的主要目标是使 android 可以在平台上运行起来，外部驱动设备可以没有使能。

在这一过程中主要需要修改的是 OMAP3430 的中断文件，定时器文件，串口输出文件，及启动时需要匹配的处理器型号和电路板型号。

在将 linux 内核层修改完成后，使用 android 自带

的编译工具链或者下载 ARM 的 gcc 编译工具链进行编译，最终在 out 目录下生成所需的 android 的文件系统。另外 android 文件系统也可以从模拟器中通过 adb 工具来获取。

Android 的启动 bootloader 代码可以使用 uboot，在 uboot 中对 OMAP3430 进行必要的初始化和 linux 内核的搬运，将启动参数和匹配的处理器的型号传递到内核，启动内核。

内核会执行文件系统中的初始化脚本，参考模拟器我们可以发现，对应于一个电路板会有一个相应的初始化脚本，所以我们根据自己的情况对初始化脚本做必要的修改，如可以启动某些应用程序，配置网络 IP 地址等，在初次启动时会创建 data 目录，所以要花费较多的时间，data 目录存放了启动过程中必要的中间数据，如应用程序解包后的文件。最终启动整个 android 系统。

在移植该系统时会遇到的一些问题包括启动参数不能正确传递，处理器型号不匹配，data 目录不能创建，在启动过程中会重新启动等问题，处理这些问题主要要对启动的机理进行详细的分析，再进行代码的逐步分析。

4 主要驱动程序的实现

在操作系统能运行之后，需要做的工作就是让电路板上的外围器件都能正常运转起来，这就是驱动程序完成的工作。

4.1 LCD 驱动程序

LCD 驱动程序是开发 android 的最基础也是最重要的一步。LCD 的驱动决定了 android 的输出色彩，也是调试其他 android 模块的一个前提。

LCD 的驱动主要是通过通过对液晶屏的文档计算出场信号和位信号的频率，通过计算所得的频率配置处理器的 LCD 模块寄存器，并为 LCD 开辟显示缓冲区。

Android 使用 LCD 是通过 mmap 系统调用的方式将 LCD 显示缓冲区直接映射到用户空间，用户空间对该地址直接操作；同时 Android 使用的两个缓冲区来进行绘图的加速，即一个显示缓冲区进行图形绘制，一个显示缓冲区进行显示，当绘制完成后，将该缓冲区进行显示，再对另外一个缓冲区进行绘制，所以在驱动中需要实现两倍大小的显示缓冲区并实现显示缓冲区切换的接口。

Android 通过系统调用获取 LCD 的详细信息包括像素点多少, 像素点位宽, 通过获取的这些信息, android 会进行显示界面的重定位。

在实际开发中遇到的问题主要是液晶屏闪烁, 色彩不正确, android 的图形界面不能显示等。

对应于液晶屏闪烁、色彩不正确主要是液晶控制器的时间相关参数没有正确配置, 这需要仔细确认液晶屏需要的每一个正确的参数。

对于 android 图形界面不能显示主要问题是 android 不能正确获取 LCD 驱动的信息, 或者没有正确实现 mmap 函数接口, 这需要对 LCD 驱动的各个接口函数做详细的分析, 可以在驱动中适当的增加调试信息确认问题的原因。

4.2 触摸屏驱动程序

触摸屏驱动程序是用户输入的重要途径, 我们采用的是 I2C 接口的触摸屏控制芯片, 四线式电阻式触摸屏, 当触摸屏被触摸, 触摸屏控制芯片会将中断发送引脚拉低, 该引脚连接到处理器的中断源输入引脚上, 处理器在接收到中断后会通过 I2C 接口读取具体的触摸点位置信息。

在 linux 内核中触摸屏的使用是 input 子系统的一部分, input 子系统会通过 event 的方式将获取的点位置信息报告给用户层。

在触摸屏驱动程序中需要对拖动的处理, 这需要利用定时器来对一次拖动进行持续的位置点跟踪, 直至没有触摸。

触摸屏因为需要和屏幕的显示对应起来, 所以需要做些必要的测试, 确认触摸屏的最小点和最大点位置信息, 并且使触摸屏的最小点和液晶屏的显示起始点对应。

4.3 键盘驱动程序

本系统中键盘是通过 TWL5030 来管理的, TWL5030 可以管理最多 8 行 8 列键盘, 支持多按键的探测, 在有按键按下时, TWL5030 会以中断方式报告处理器, TWL5030 可以主动扫描记录下按键信息, 或

者由软件控制进行扫描来获取按键。

键盘驱动在 linux 中也是 input 子系统的一部分, 在驱动获取到按键值后, 将键值以 event 的方式通知用户程序。

在 android 中使用键盘是一个比较复杂的过程, 这主要是功能键的切换, 如按下大写键或者按下 SHIFT 键后需要切换输入的具体键值, android 使用了 kcm 和 kl 文件来完成这个功能, 当用户空间接收到驱动程序报上的键值后会到 kl 文件中进行匹配, 如果不是功能键则查找到匹配的键值后传递给具体的应用程序, 如果是功能键则会记录该功能键, 当再有按键到来时则根据 kcm 文件和记录的功能键来查找当前的键值。

通过修改文件系统中的 kl 和 kcm 文件用户可以方便的修改具体的按键映射而不必修改内核驱动程序。

5 应用程序的实现

Android 应用程序使用 JAVA 编写, 通过 Eclipse 开发工具可以创建一个应用开发工程, 编译生成 apk 包通过 adb 可以安装进 android 系统。

一个应用程序一般由以下几个部分组成^[2]。

Activity 是应用程序的主体结构; Intent receiver 是处理某个任务的主体; Service 是没有用户接口的服务线程; Content provider 是一个基础的框架用来处理和保存数据。

在具体的应用中 android 为了安全性需要注意签名的问题。

在应用程序的实现上可以根据具体的应用编写具体的程序, 这部分代码不依赖于具体的平台, 也使得我们的硬件平台和系统软件平台可以作为不同应用环境的电子商务终端

References (参考文献)

- [1] Yang Fengsheng, Anroid Unleashed[M], Beijing, China Machine Press, 2010, P3-P7 (Ch).
杨丰盛, Android 应用开发解密[M], 北京, 机械工业出版社, 2010, P3-P7
- [2] Chris Haseman, Android Essential[M], Berkeley, apress, 2008, P15.