CUSTOMISED POWER EFFICIENT ANDROID ROM

SOFTWARE DESIGN DOCUMENT

Done By :

Ajith K. M ETAKECS004

Anjana Sasikumar ETAKECS007

K. Haridas ETAKECS026

Guided By :

BISNA N.D

(Asst. Professor)

On

8th February 2013

Contents

Chapter 1

Introduction

1.1 Purpose

A software design is a meaningful engineering representation of some software product that is to be built. During the design process the software specifications are transformed into design models that describe the details of the data structures, system architecture, interface, and components. Each design product is reviewed for quality before moving to the next phase of software development. At the end of the design process a design specification document is produced. This document, called the Software Design Document is composed of the design models that describe the data,architecture, interfaces and components.

A software design document (SDD) is a written description of a software product, that a software designer writes in order to give a software development team an overall guidance of the architecture of the software project. Practically, a design document is required to coordinate a large team under a single vision.

The proposed system aims to improve the power efficiency of the android device by building a customized Android ROM(similar to an operating system) which would make the installation of the stock applications optional. Power efficiency will also be improved by making changes to the kernel to bring about controlling the processor speeds to obtain maximum power efficiency. In building the ROM from the source code additional features like support for native languages like Malayalam can be brought about.

1.2 Scope

The Scope of this document is to cover all the aspects involved in the software design procedure in building a customized android ROM.It includes details about the various modules included in the customized ROM.It also shows how the user interacts with the ROM and what kind of output it produces through that interaction through data flow diagrams.The deatailed description of Android Architecture is also specified.

1.3 References

- http://developer.android.com/index.html
- http://forum.xda-developers.com/
- https://developers.google.com/
- www.droidforums.net

Chapter 2

Definitions, Acronyms & Abbreviations

2.1 Defnitions

JDK :- The **Java Development Ki**t (JDK) is an Oracle Corporation product aimed at Java developers. Since the introduction of Java, it has been by far the most widely used Java SDK. On 17 November 2006. Sun contributed the source code to the OpenJDK.

Android SDK:-The **Android software development kit** (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator , documentation, samplecode, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.4.9 or later, Windows XP or later. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though developers may use any text editor to edit Java and XML files then use command line tools (Java Development Kit and Apache Ant are required) to create,build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

AVD:- AVD is the **Android Virtual Device** , which is helpful in running the application in a computer.AVD is a virtually creates a mobile device,which helps in running the application in a computer.The program running in avd seems to be like running the application in android mobile phone.

Ambiguity between SDK and JDK:- The JDK forms an extended subset of a software development kit (SDK). In the descriptions which accompany its recent releases for Java SE, EE, and ME, Sun acknowledges that under its terminology, the JDK forms the subset of the SDK which has the responsibility for the writing and running of Java programs. The remainder of the SDK comprises extra software, such as application servers, debuggers, and documentation.

2.2 Abbreviations

- SDK : Software Development Kit
- NDK : Native development kit
- AVD : Android Virtual Device
- GUI : Graphical User interface

Chapter 3

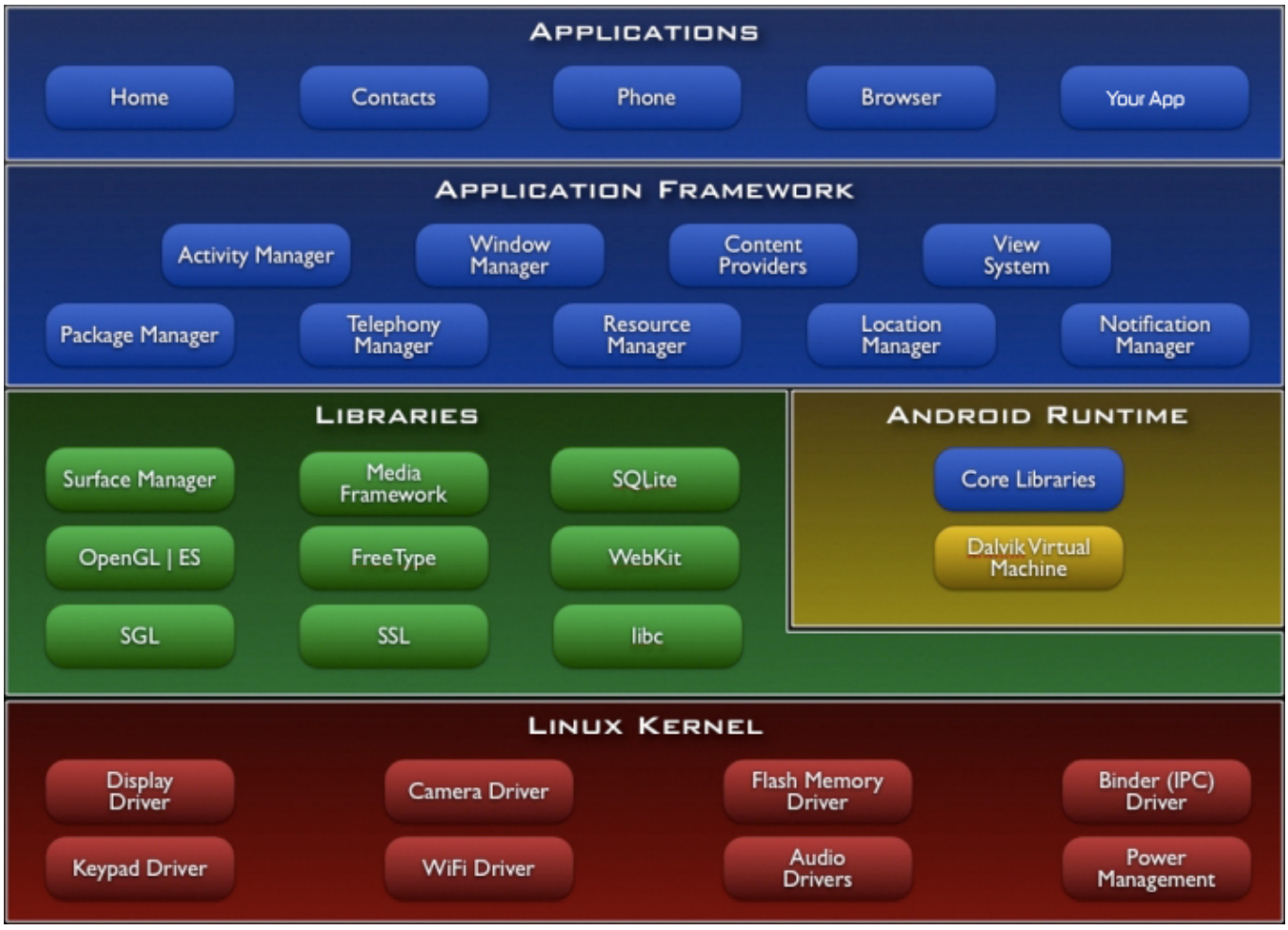
System Overview

The main components in building a rom includes a device and a computer running Linux with a few necessary packages installed like Java Development Kit 6,Android Virtual Device, Android SDK .The AVD is a android device emulator which works as an android phone. The prepared rom can run on this virtual device. The ROM will be developed by compiling from the android source code provided by Google. After the primary compilation different modifications are made to the ROM to make it power efficient.
The main development environment for the Rom is Android SDK.It contains all the packages necessary for compiling the source code of the Android OS and AVD's which can be used for installing and testing the developed ROM. The example of an Android SDK is shown below. The device is HTC Wildfire for which the proposed ROM is developed.

Chapter 4

Architectural Design

1.1 Introduction



The above figure shows the diagram of Android Architecture. The Android OS can be referred to as a software stack of different layers, where each layer is a group of several program components. Together it includes operating system, middleware and important applications. Each layer in the architecture provides different services to the layer just above it. We will examine the features of each layer in detail.

Linux Kernel

The basic layer is the Linux kernel. The whole Android OS is built on top of the Linux 2.6 Kernel with some further architectural changes made by Google. It is this Linux that interacts with the hardware and contains all the essential hardware drivers. Drivers are programs that control and communicate with the hardware. For example, consider the Bluetooth function. All devices has a Bluetooth hardware in it. Therefore the kernel must include a Bluetooth driver to communicate with the Bluetooth hardware. The Linux kernel also acts as an abstraction layer between the hardware and other software layers. Android uses the Linux for all its core functionality such as Memory management, process management, networking, security settings etc. As the Android is built on a most popular and proven foundation, it made the porting of Android to variety of hardware, a relatively painless task.

Libraries

The next layer is the Android's native libraries. It is this layer that enables the device to handle different types of data. These libraries are written in c or c++ language and are specific for a particular hardware.

Some of the important native libraries include the following:

Surface Manager: It is used for compositing window manager with off-screen buffering. Off-screen buffering means you can't directly draw into the screen, but your drawings go to the off-screen buffer. There it is combined with other drawings and form the final screen the user will see. This off screen buffer is the reason behind the transparency of windows.

Media framework: Media framework provides different media codecs allowing the recording and playback of different media formats

SQLite: SQLite is the database engine used in android for data storage purposes

WebKit: It is the browser engine used to display HTML content

OpenGL: Used to render 2D or 3D graphics content to the screen

Android Runtime

Android Runtime consists of Dalvik Virtual machine and Core Java libraries.

Dalvik Virtual Machine

It is a type of JVM used in android devices to run apps and is optimized for low processing power and low memory environments. Unlike the JVM, the Dalvik Virtual Machine doesn't run .class files, instead it runs .dex files. .dex files are built from .class file at the time of compilation and provides higher efficiency in low resource environments. The Dalvik VM allows multiple instance of Virtual machine to be created simultaneously providing security, isolation, memory management and threading support. It is developed by Dan Bornstein of Google.

Core Java Libraries

These are different from Java SE and Java ME libraries. However these libraries provides most of the functionalities defined in the Java SE libraries.

Application Framework

These are the blocks that our applications directly interacts with. These programs manage the basic functions of phone like resource management, voice call management etc. As a developer, you just consider these are some basic tools with which we are building our applications.

Important blocks of Application framework are:

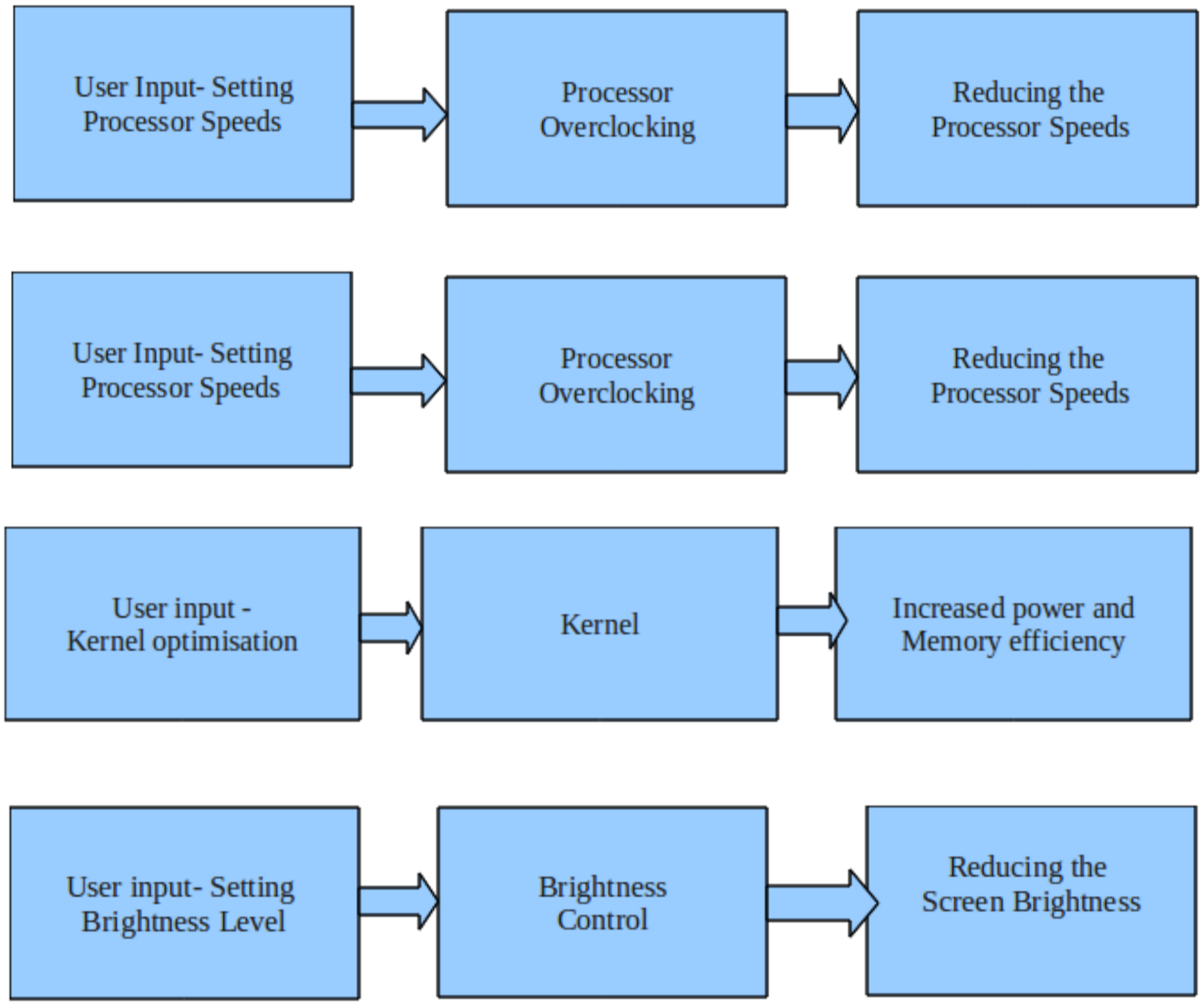
Activity Manager: Manages the activity life cycle of applications
Content Providers: Manage the data sharing between applications
Telephony Manager: Manages all voice calls. We use telephony manager if we want to access voice calls in our application.
Location Manager: Location management, using GPS or cell tower
Resource Manager: Manage the various types of resources we use in our Application

Applications

Applications are the top layer in the Android architecture and this is where our applications are gonna fit. Several standard applications comes pre-installed with every device, such as:

- SMS client app
- Dialer
- Web browser
- Contact manager

4.2 Data Flow Diagram



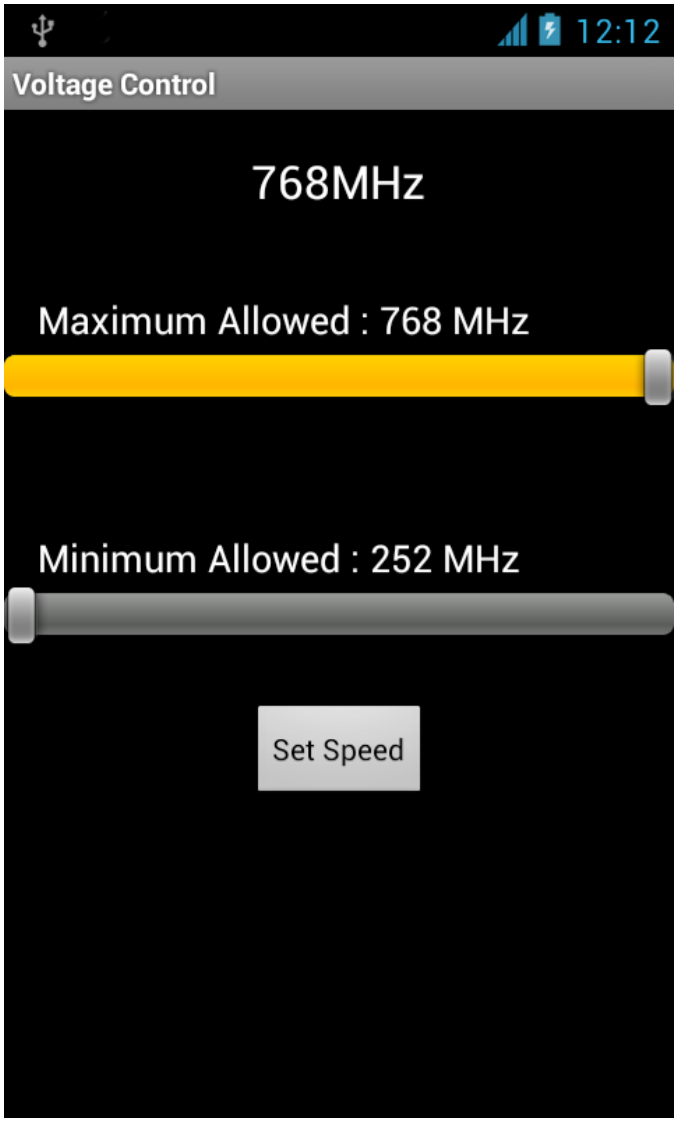
The above figures break down the flow of working of the rom in different modules. Bloatwares are one of the main reasons why the newer versions of the android phones do not give proper battery drain. These bloatwares run in the background utilising the battery charge. The unwanted bloatwares are removed in the rom and the incorporation of inverted google applications is done. The google applications include gmail,gtalk,android play store, etc. These are tweaked to perform with a black interface so as to reduce the battery drain. This is first done by the user input optimising application which in turn removes the bloatwares and adds the inverted google applications. The next method of maximising the

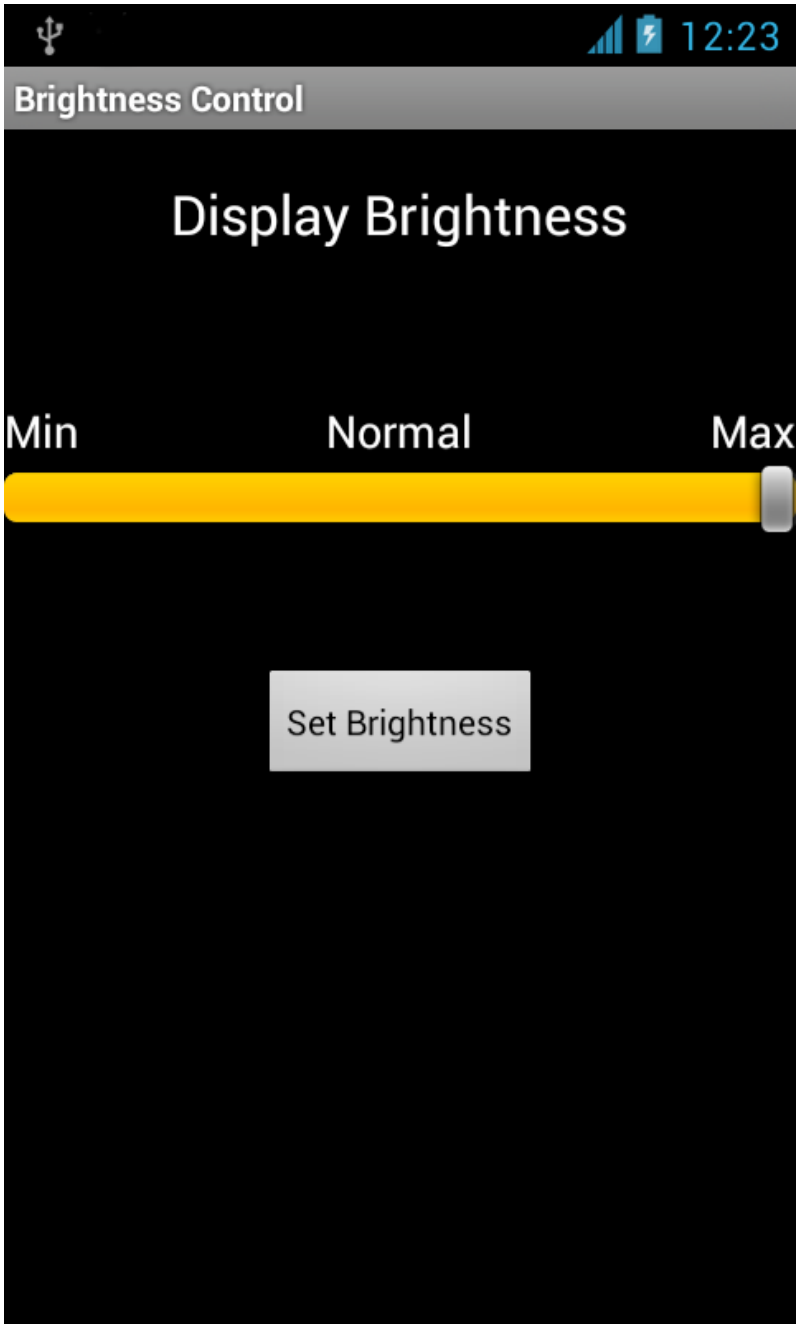
power efficiency is by optimising the brightness level of the device. By making it automatic, the device senses the change in the light falling on it and adjusts the brightness according to it. Reduced brightness levels reduce the battery drain considerably. The kernel can be modified to perform better than how it works in the stock device. In the rom, the kernel used has many different modifications done to it in order to perform better and work longer. In the utilisation of the fourth module, the device can be made to work at a lower voltage. This is done by tweaking the kernel so as to optimise the working voltage of the device. The user is given the option to select the voltage at which the device works.this is made aware to the kernel and the necessary adjustments are made.

4.3 GUI Diagrams [User Interface Design]

The GUI is the Graphical user interface of the ROM.It Shows the user interfaces available with the device. In the ROM there will be options to choose the different modules for optimising the power efficiency.

The Display interfaces may be different with different diagrams. It is possible to have several views with same details. Its is an optional requirement.Examples of some of the output graphical user interfaces . The Android Virtual Device view will be different from the real device view.In our ROM the options for setting the processor voltage and display brightness will be as shown in the diagrams.The other options for kernel tweakings will be provided in the source code and as such does not require any special GUI.Similarly the removal of bloatwares and installation of inverted gapps are done at the time of installation of the ROM and therefor as such does not validate the need for a seperate GUI.


12:12
Voltage Control
768MHz
Maximum Allowed : 768 MHz
Minimum Allowed : 252 MHz
Set Speed


12:23
Brightness Control
Display Brightness
Min
Normal
Max
Set Brightness

Chapter 5

Component design

Component design is a description of various components of the system.it includes the front end and back end.

Front end

The front end is the software used to create the user interface which is in our case is the android sdk. The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug ROMs for Android.These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, etc

Back end

The back end of the ROM is the source code provided by google.the source code is compiled in order to build the android ROM.The source code has the necessary definitions for the working of all the drivers of the device.The android source code is written in Java.

Chapter 6

Conclusion

This project aims to implement an Android ROM .Detailed knowledge of the building environment and the Android system is a prerequisite for building the ROM. Intricate working of the Android system can be learnt by doing the project. The ROM can be ported to other Android devices with minor modifications depending on the hardware differences.The design of the proposed ROm is completed in this procedure.The different modules involved in the project have been identified and has been structured in the necessary manner so as to implement it effectively in the device.