Software Architecture

for Low Cost Mobile Handsets

A complete solution dervied from Open Source Software

Rajiv Poddar

Tantra Tele

Table of Contents

1 Introduction

This document describes the architecture of a fully integrated software solution for Low Cost Mobile Phones using Open Source Software. The solution consists of an Operating System (OS), Wireless Protocol Stack, Man Machine Interface (MMI) Framework, Device Driver Framework, Integrated Development Environment (IDE) and Software Development Kit (SDK). Various processor architectures including IA32, ARM, PowerPC and MIPS are supported. All the software components are derived from open sources which are available publically.

This document is intended for clients, developers, testers and end users. Section 2 provides an overview of the system, Section 3 covers the details of all the components and Section 4 explores some licensing consideration. Prior knowledge of embedded systems, mobile phone architecture and wireless telephony systems is assumed.

2 Overview

The Architecture provides a simplified view of the software components.

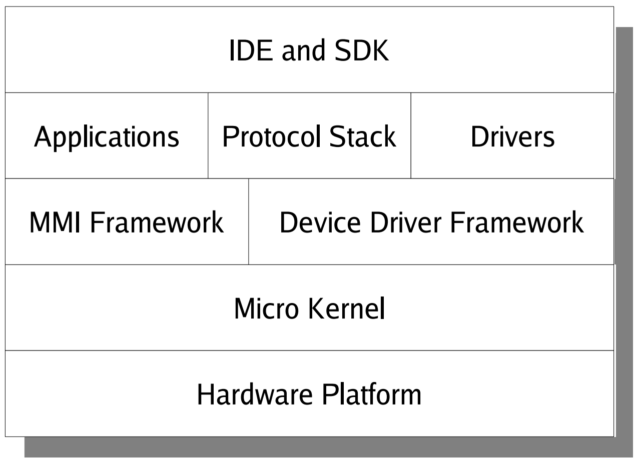

Illustration 1: Architecture

A real time microkernel forms the core of the system. The kernel has a very small footprint which can be used on a restricted hardware platform. All device drivers are seperate tasks which are managed by the kernel. The MMI Framework is ncurses based and can be used to create menus and panels. The DD Framework provides support for peripherals such as keypad, display, USIM etc. A suite of applicatons is also provided which includes a text based Web Browser, Games and Personal Information Manager (PIM). The Protocol Stack provides the telephony related functions and consists of all the upper layers of the stack as per the standards. An ecplise based IDE and tools chains are plugins are provided for development of applications. The SDK consists of utility libraries which are ported the platform.

2.1 Hardware Platform

The illustration of the platform is shown in Hardware Platform.

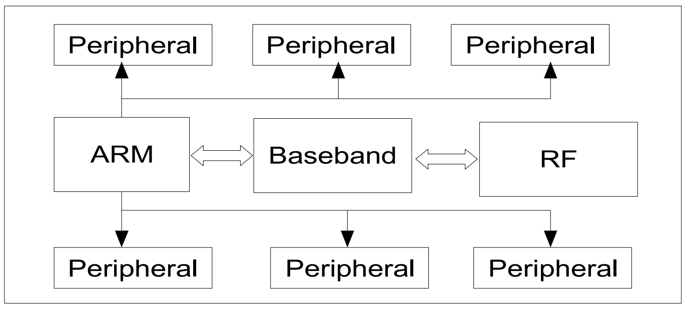

Illustration 2: Hardware Platform

It consists of a ARM based host processor which is connected to a Baseband and RF susbsytem via the system bus. External memmory and other peripherals are connected to the host processor via the system bus or other interfaces. The hardware platform depicted is a generic embedded system platform. The software solution resides on the host processor and controls the baseband and peripherals through specialized device drivers.

2.2 Microkernel

Microkernels are minimal operating systems which only provide scheduling, address-space management and inter process communication (IPC). All the other functionalities are provided by user-space programs called servers. The device drivers are also treated as normal user-space programs and interact with the kernel to provide I/O services. Networking and PCMCIA services are also provided as user-space servers. This approach is a fundamental departure from the monolithic kernel based systems which provide all the above mentioned services within the kernel (eg. Linux).

Since the kernel only provides the essential services, the footprint is usually in kilo bytes. This makes them ideal for embedded systems which have low memory and slower CPU's. Sicne all the device drivers are implemented as user-space programs, they can be easily managed. All these leads to a highly modular and configureable system which is suited for embedded platforms.

2.3 MMI Framework

The MMI framework refers to the user interface provided on the mobile handsets. It generally consists of menus, windows and panels. The user uses navigation keys to choose a particular option which can either launch an application or lead to a sub menu.

The ncurses library is used in this solution to provide a simple lightweight MMI Framework. The library is adapted to embedded system usage and provides an API's for creating menus, sub menus, panels, windows and launching applications. The framework is completely text based and does not support any graphics.

The display is treated as a VT 100 terminal with a specific numbers of rows and columns. These can be configured during the build. Special capabilities of VT 100 based terminals are also supported. UTF-8 characters are also supported.

2.4 Device Driver Framework

The device driver framework is a library which provides the minimal set of device drivers for the embedded system. It consists of a watchdog timer, serial port driver and boot rom drivers. The watchdog timer monitors the CPU and resets the CPU if any task hogs the CPU time. The serial port driver is used for debugging over the serial port and provides an alternate means to load the software and firmware.The boot rom is a Flash beased memory which is used during the initial boot up. The Das U Boot is loaded on to the boot flash by default and it loads the OS from the Non Volatile memory. It also supports TFTP and network boot for easier development.

The DD Framework is also provides certain essential services to the drivers.

2.5 Applications

A suite of applications are provided along with the solution. It includes a text based browser, a simple Personal Information Manager (PIM) and few games. They are all derived from Open Source Software soultions and are relatively simple. They have small foot print and are text based.

2.6 Protocol Stack

The Protocol Stack provides the essential telephony services. Only the upper layers of the protocol stack are implemented. It interacts with the baseband using the system bus and DMA. The protocol stack is also dervied from Open Source Software solutions and ported to the microkernel based OS. It is amongst one of the task which is managed by the OS.

Both GSM and CDMA families of protocol stack are supported.

2.7 Device Drivers

The Device Drivers are used to control the peripherals connected to the system. They include the SIM Driver, the Keypad Driver, the Display driver, the Sound Driver and the Baseband Driver. All these drivers depend upon the type of peripheral connected to the system. Generally the device drivers are supplied by the manufacturer of the peripheral. These are ported to the microkernel OS so that they can be used in the system.

All the device drivers also appear as normal tasks running on top of the OS.

2.8 IDE and SDK

An IDE and SDK is provided for development and debugging. The IDE is Eclipse based and plugins for the OS tool chains are provided. The SDK includes the libraries which are needed for development of applications over the OS. All the essential libraries are provided.

3 Licensing Considerations

All the software used in this solution are dervied from Open Sources. They follow various licensing methods and the licenses need to ahered for commercial use. The GNU Public License (GPL) which restricts mixing of any proprietary code with it (copyleft clause). Further GPL'ed code cannot be mixed with BSD style licenses since GNU forbids addition of more caluses. The Library GPL (LGPL) licenses are used for the libraries and it does not have the copyleft clause.

In the software soultion, the kernel is licensed under BSD Style and the ncurses libraray is provided under the LGPL library. All the device drivers are either proprietary in nature or GPL'ed. The applications provided are mostly GPL.