# A REVIEW ON ANDROID APPLICATION USING IN EMBEDDED SYSTEMS

**Mrs.G.Sumalatha[1], Mr.S.Bharathiraja[2]**

*[1]Information Technology,V.R.S College of Engg. & Tech., Anna University Chennai,India,*
*[2] Information Technology,V.R.S College of Engg. & Tech., Anna University Chennai,India*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract—** *Android's growth is phenomenal. In a very short time span, it has succeeded in becoming one of the top mobile platforms in the market. Clearly, the unique combination of open source licensing, aggressive go to-market and trendy interface is bearing fruit form Google's Android team. Needless to say, the massive user uptake generated by Android has not gone unnoticed by handset manufacturers, mobile network operators, silicon manufacturers, and app developers. Products, apps, and devices "for," "compatible with, "or "based on" Android seem to be coming out ever so fast. Beyond its mobile success, however, Android is also attracting the attention of yet another, unintended crowd embedded systems developers. While a large number of embedded devices have little to no human interface, a substantial number of devices that would traditionally be considered "embedded" do have user interfaces. For a goodly number of modern machines, in addition to pure technical functionality, developers creating user-facing devices must also contend with human-computer interaction (HCI) factors. The Android mobile platform is being adopted by a broad range of embedded devices that span multiple industries and segments. Android-enabled custom solutions, such as the kindle Fire by Amazon, have proven to be game-changing in the industry. According to Amazon, the kindle Fire captured 22 percent of the US tablet market, and increased the company's e-books sales by 175 percent in 2014.*

***Keywords/Index Term—Android, DVM, HCI, Embedded system***

## 1. INTRODUCTION

Android is an open source platform built by Google that includes an operating system, middleware, and applications for the development of devices employing wireless communications. This article takes a look at the design of Android, how it works, and how it might be deployed to accelerate the development of a connected device. Along with basic guidelines to getting started with Android, the Android SDK, available tools and resources are reviewed and some consideration is given to applications for Android. Beyond conventional mobile handsets such as medical devices, consumer electronics, and military/aerospace systems, with a particular emphasis on the application of Android in the home.

## 2. WHAT IS ANDROID?

It is easy to think of Android as being yet another operating system for high-end mobile phones. It's really a software platform rather than just an OS – with the potential to be utilized in a much wider range of devices. In practical terms, Android is an application framework on top of Linux, which facilitates its rapid deployment in many domains. A key to its likely success is licensing.

**Fig.1. Android**

Android is open source and a majority of the source is licensed under Apache2, allowing adopters to add additional proprietary value in the Android source without source distribution requirements.

Another way to appreciate the significance of Android is to take a historical perspective. In the early days of PCs, the operating system was DOS. This presented some interesting challenges to application developers, as DOS provided a minimal number of services.

The result was that every application needed a complete framework toprovide the full functionality that was required. For example, a word processing program would need to have a driver for every imaginable printer. This was a major headache for developers and a serious ongoing maintenance problem. The solution came in the early 1990s with the release of Windows, or rather the development of Windows 3.0. Although we think of Windows as being primarily a GUI, it's really much more than that. Nowadays, a word processor just talks to a logical printer.

In some respects, a similar situation exists today when developers want to deploy Linux for embedded applications. Android is the enabler for a broad application developer base, a complete stack on top of the Linux kernel.

## 3. ARCHITECTURE

An Android system is a stack of software components (Figure 1). At the bottom of the stack is Linux – Linux 2.6 with approximately 115 patches. This provides basic system functionality like process and memory management and security.

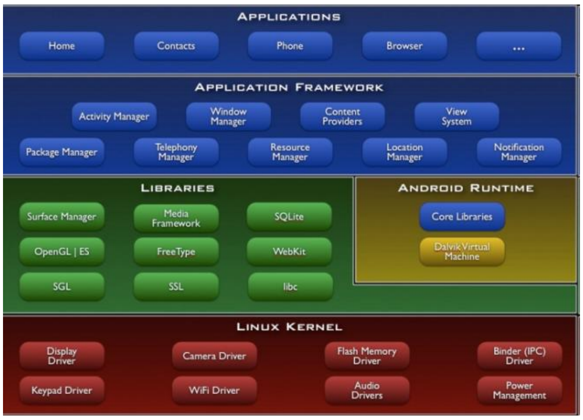

**Fig.2. Architecture**

Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

• Libraries On top of Linux is a set of libraries including bionic, media support for audio, video, and graphics along with a lightweight database that serves as a useful repository for storage and sharing of application data.

• Android Runtime A key component of an Android system is the runtime – the Dalvik Virtual Machine (VM). This is not strictly a Java virtual machine. It was designed specifically for Android and is optimized in two key ways. First, it is designed to be instantiated multiple times – each application has its own private copy running in a Linux process. And second, it was designed to be very memory efficient, being register based (instead of being stacked based like most Java VMs) and using its own byte code implementation. The Dalvik VM makes full use of Linux for memory management and multi-threading, which is intrinsic in the Java language.

It is important to appreciate that Android is not a Java virtual machine, but does use the Java language.

• Application Framework the Application Framework provides many higher level services to applications in the form of Java classes. This will vary in its facilities from one implementation to another. A key Android capability is the sharing of functionality.

• Applications At the top of the Android software stack are the Applications layer. There are a number of supplied standard applications. As mentioned, each application may also expose some of its functionality for use by another application. For example, the message sending capability of the SMS application can be used by another application to send text messages. The supplied applications are not particularly "special" – all Android applications have the same status in a given system.

Although there are other options, Android applications are commonly implemented in Java utilizing the Dalvik VM. Not only is Dalvik highly efficient, but it also accommodates interoperability which results in application portability.

While all of these attributes are attractive, many developers will also want their C/C++ applications to run on an Android-based device.

## 4. ANDROID DEVELOPMENT

Development Environment The standard Android development environment from Google is, as you might expect, Eclipse-based, using a plug-in to provide the necessary facilities. You need to define your target configuration by specifying an Android Virtual Device. You can then execute code on either the host-based emulator or a real device, which is normally connected via USB. This environment only supports Android development on ARM-based target devices.

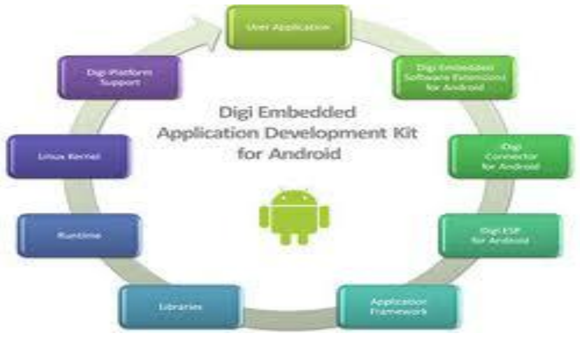
User Application
Digi Embedded Software Extensions for Android
Digi Connector for Android
Digi ESP for Android
Application Framework
Libraries
Runtime
Linux Kernel
Digi Platform Support
Digi Embedded Application Development Kit for Android

**Fig.no.3.Android development**

ProgrammingModel an Android application consists of a number of resources, which are bundled into an archive called an android package. Programs are generally written in Java, built using the standard Java tools, and then the output file is processed to generate specific code for the Dalvik VM. Each application runs in its own Linux process - an instantiation of the Dalvik VM – which protects its code and data from other applications. Of course, there are mechanisms for applications to transfer, exchange, and share data. An application is a set of components which are instantiated and run as required.

There are four types of application components: activities, services, broadcast receivers, and content providers.

• An Activity in a functional unit of the application, which may be invoked by another activity or application. It has a user interface of some form. An application may incorporate a number of activities. One activity may be nominated as the default which means it may be directly executed by the user.

• A Service is similar to an activity, except it runs in the background without a UI. An example of a service might be a media player that plays music while the user performs other tasks.

• Broadcast Receivers simply respond to broadcast messages from other applications or from the system. For example, it may be useful for the application to know when a picture has been taken. This is the kind of event that may result in a broadcast message.

• A Content Provider supplies data from one application to other applications on request. Such requests are handled by the methods of the Content Resolver class. The data may be stored in the file system, the database, or somewhere else entirely.

When you develop an Android application, you'll need to describe it to the system and this is achieved by means of a manifest file. This is an XML file called AndroidManifest.xml, which is stored in the root folder of the application's file system.

## 5. ANDROID UI

With so many Android-based devices under development, one question needs to be asked: How can one product are properly differentiated from another Android-based device? A highly effective way of achieving this is through user interface (UI) differentiation. One method is to customize the home screen as this allows standard third-party Android applications to run without modification.

It is possible to implement these UI changes through hard coding; however, the UI effects that can be achieved will be limited (without detailed knowledge and a lot of hard work). Fortunately, there are a small number of tools available that allow changes to be made to the look and feel of the UI without touching the code. This means that product variants or new products can be achieved with very little engineering effort.

One such tool is Inflexion™ UI solution from Mentor Graphics, which allows visually rich 2D & 3D UIs to be created quickly and easily whilst retaining Android compatibility (i.e. applications can still be downloaded from the Google Apps store and run on the device).

The output of the tool runs on Mentor Graphics' Inflexion graphics engine which has built in rendering for 2D and 3D effects. If hardware graphics acceleration is available (supporting Open GL ® ES) the Inflexion engine will offload the graphics processing to this accelerator and allow true 3D (for example ARM's Mali™ graphics processor).

**Fig.no.4.Android user interface**

## 6. EXTENDED ANDROID BEYOND MOBILE

Until very recently, Android deployment has been focused on mobile handsets. This was Google's target market and the available software IP and development tools are designed and configured with this in mind. The potential for Android is enormous in other market areas – anywhere that sophisticated software, including connectivity and a user interface, encapsulates the functionality of a device. Consumer, telecom, automotive, medical, and home applications are all attractive candidates for the deployment of

Android. However, there are challenges in moving away from mobile handsets.

To expand into other markets, investment is required in order to optimize and tune the Dalvik VM and the libraries to the selected SoC, develop or integrate drivers and libraries for industry specific peripheral devices, and to customize the UI for the required market. In addition, it is not uncommon for a large amount of legacy native C/C++ code to exist which needs to be ported to Android.

**Fig.no.5. Example**

This code needs to be integrated within the Android environment and potentially interface to the Dalvik VM so that developers can make use of this functionality in their Java applications. At the same time it is imperative that modules and their licenses are tracked and managed and that open source compliance isachieved.

This is best achieved through using an experienced Android partner who has the required experience, knowledge, and toolset to ensure that the Android development is successful and that the consider - able benefits of using Android are fully realized.

### 7. THE RISE OF MODERN EMBEDDED SYSTEMS

As we know, embedded systems control many devices that are in common use today. They range from portable devices such as digital watches and MP3 players, to large stationary installations like ATM and vending machines. However, embedded systems have changed.

Dramatically in recent years today, they are largely media-rich, connected and highly integrated. Many include graphical user interfaces with high revolution 2D and 3D graphics. Additionally, nearly all embedded systems include IP networking stacks, and links connectivity via a combination of wired and wireless network interfaces. The core feature sets often rely upon connectivity. And last of all, or reasons such as power efficiency, size and performance, size and performance, chipsets for embedded systems are designed to be highly integrated.

**Fig.no.6.MODERN embedded systems**

This drastic change in the characteristics of modern embedded systems has given rise to advanced functionality and user experience needs, and Android helps address these needs.

### 7.1. DALVIK VIRTUAL MACHINE

Android’s dalvik virtual machine is specifically designed to support a diverse set of devices, where the applications must be sand-boxed for security, performance and reliability. Also, it works very well with limited processor speeds and RAM.

### 7.2. HARDWARE PLATORMS SUPPORT

Android supports a variety of hardware platforms. Apart from ARM-based Android development phones, it also supports hardware platforms or prototyping and benchmarking Android systems, such as table and automotive soC.

### 7.3. NATIVE DEVELOPMENT KIT

Android supports a native development kit embedscomponents that make use of native C/C++ code in Android applications. The NDK helps address the needs of performance and graphics-sensitive applications.

### 7.4. OPTIMISED GRAPHICS AND MEDIA SUPPORT

Android provides support for a wide range of media formats through stage fright, its custom media framework. It also provides its own 2D graphics library, but relies on openGLES for its 3D capabilities. This feature makes it feasible to create small –sized embedded systems with high-end audio and video capabilities.

### 7.5. TELEPHONY SUPPORT

Android supports telephony, which is dependent on hardware. For this feature to work, device manufactures need to create a HAL module to interface with their hardware, which is integrated as part of the Android build systems.

### 7.6. WIRELESS CONNECTIVITY

Android supports most wireless connectivity options, like Bluetooth, EDGE, 3G and Wi-Fi. This enables the embedded device to have a wide range of connectivity options to any third-party systems.

### 7.7. LICENSING

Android licensing terms are pretty friendly for both commercial and free open source applications. This is because all its core packages are open sourced under the terms of the Apache 2.0 license.

### 7.8. CUSTOMISABILITY

Android has well defined interfacing between the framework ants its components. This helps to enhance or replace components as per the desired functionality. For example, the default launcher application can be enhanced or replaced with a different launcher application code base. The platform can be enhanced to support additional features and hardware, as desired.

### 7.9. A WELL ACCEPTED AND RECOGNISED UX

Android user experience (UX) designs such as the frog's feel UX, HTC sense and many others have raised UX standards to quite a high level. This means that your system remains can enjoy the high end UX capabilities that Android supports .with this edge, the learning curve for end users of your embedded system remains no longer a big concern.

### 7.10. FORWARD-COMPATIBLE APPS

All APIs provided in the application frame work are meant to be forward compatible. Hence, apps developed for one version would continueworking in future Android versions. However, platform-level changes would require to be ported when planning to upgrade the Android version of your custom solution.

### 7.11. WIDE CHOICE OF HARDWARE CONFIGURATION

Android has wide support from the OEM and soC community. Though primarily supporting ARM-based soCs, Android now also supports x86-based soCs. This provides for a wide range of hardware configurations to choose from, depending on what fits your budget and system requirements.

### 7.12. HASSLE-FREE SUPPLY CHAIN COMMODITISATION

No doubt-Android, with its solid foundations on Linux, sound ecosystem and developer community, together with the cutting edge advantages, has truly proven itself as the platform of choice for any modern embedded system.

## 8. APPLICATIONS

### 8.1 Personal computer remote

PC remote is the coolest way to control your computer. With a simple and beautiful design we provide maximum ease and enjoyment of use. Using the PC remote application you can control your computer remotely via your Wi-Fi network or hotspot. PC remote includes simulation of Touchpad, Gyroscope control, File Explorer, and simulation of remote control for various popular video players (BSplayer, VLC player, WMP, GOM, KMP, UMP...) and Power Point. PC remote includes wireless file sharing! Saving your pictures and music has never been easier.

### 8.2 Remote personal computer

Remote PC is an application which allows you to remotely control your PC over Bluetooth or Wi-Fi. Get access to a computer's desktop and use your favorite programs from distance. You can control mouse, keyboard (with functional keys and keyboard shortcuts), multimedia, presentation or power system commands.

### 8.2 NXT remote control

Control your Lego Mind storms NXT robot from your phone via Bluetooth. Connect the wheel motors to outputs B and C and optionally the action motor to output A. Several control schemes to choose from (switch from menu). Send me an email if you encounter any problems or have any comments or suggestions.

### 8.4 Remote desktop

Remote Desktop Client for Android enables you to connect to your Windows computers across the Internet from a mobile device powered by Google Android platform. Once connected, it gives you mouse and keyboard control over your computer while showing you everything that's happening on the screen. You can leave your computer without losing access to your files, applications, and e-mail. Remote Desktop Client can connect using Microsoft Remote Desktop Protocol (RDP) or VNC protocol. For RDP protocol all the necessary server components are already part of most versions of the Windows operating system (except Home varieties).

### 8.5 Microsoft remote control:

With the Microsoft Remote Desktop app, you can connect to a remote PC and your work resources from almost anywhere. Experience the power of Windows with Remote FX in a Remote Desktop client designed to help you get your work done. **Getting Started** Windows Professional and Windows Server editions are supported. To view the full list of supported Windows versions and successfully connect to a Windows PC.

**Features:**
Access to remote resources through your Remote desktop Gateway. Rich multi-touch experience with remote desktop protocol (RDP) and Remote FX supporting Windows gestures. Secure connection to your data and applications with breakthrough Network Layer Authentication (NLA) technology. Simple management of all remote connections from the connection center.
High quality video and sound streaming with improved compression and bandwidth usage.

### 8.6 Remote control collection:

The Remote Control Collection is a compilation of remotes, which you can use to wirelessly control your Windows PC! The unified remotes enable you to control the PC beyond mouse and keyboard!
Included remote controls are:

-Mouse
-Keyboard
-Live Screen (Pro)
-Speech recognition
-Media Player (Pro)
-Slideshows (Pro)
-Scroll
-Shortcuts

### Mouse Remote

Imagine your PC’s touchpad right on your Android device. Multi touch gestures like scrolling and zooming are supported. You can toggle the keyboard to send keys at the same time.

## 9. ANDROID APPLICATION IN OUR PERSONAL COMPUTER

Intel is now interested in pushing “Dual OS” PCs devices with both Windows 8 and Android on them. Android apps and even the Android operating system on your current PC.This allows you to use Android’s ecosystem of touch-based apps on touch-enabled Windows laptops and tablets, so it does make some sense. Of course, the process is clunker than just using Windows 8 apps.

### Android-x86

Android-x86 is a community project to port Android to the x86-platform so it can run natively on Intel and AMD processors, allowing you to install Android on a laptop or tablet just as you’d install Windows or Linux. This project was originally noteworthy for providing a way to run Android on low-power netbooks, giving those old netbooks some additional life.

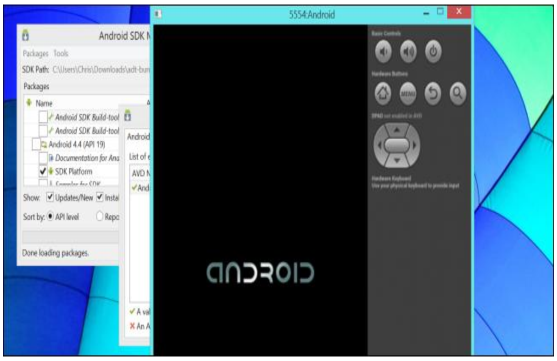

Fig.no.7.Android app in pc

### Android on Intel Architecture

Intel develops their own distribution of Android for new Intel-based PCs with UEFI firmware. It’s named Android on Intel Architecture, or Android -IA. Intel even provides an installer, which you can use to install Android on your Windows 8 device. The installer will ask if you want to preserve Windows in a dual-boot scenario, so this is a way to dual boot Android and Windows on a new laptop or tablet.

## 10. CONCLUSION

Thus we have surveyed about embedded android that is being adopted by a broad range of embedded devices that span multiple industries and segments. Android-enabled custom solutions, such as the kindle Fire by Amazon, have proven to be game-changing in the industry.

## REFERENCES

[1]. http://www.android.com
[2]. http://www.dalvikvm.com
[3]. Embedded android - **Karim Yaghmour.**
[4]. Android-x86, “Android-x86 project,” Jan. 2010.
[6]. G. Macario, M. Torchiano, and M. Violante, “An in-vehicle infotainment software architecture based on Google android.
[7]. RTMACH, “Linux/rk,” Mar. 2010.

## BIOGRAPHIES

**Mrs.G.Sumalatha** received the B.E in CSE from Anna University Chennai, in 2005 and received the M.E. degree in computer science and engineering from Anna University, Chennai, in 2009. Her research interests include Networking, Data ware housing & mining

**Mr.S.Bharathiraja** received the B.E in CSE from Anna University Chennai, in 2005 and received the M.E. degree in computer science and engineering from Anna University, Chennai, in 2007. His research interests include compiler design, Networking.