**CHAPTER 1**

**INTRODUCTION**

Advertising is a means by which communication with the users of a product or service occurs. Advertising is always present, though people may not be aware of it. In present world, using every possible media advertising get’s its message through. This is done via television, print (newspapers, magazines, journals etc), radio, press, internet, events, direct selling, hoardings, posters, mailers, contests, clothes, sounds, visuals sponsorships, and even people (endorsements). However today’s advertising platforms are relatively expensive in terms of creative, production and airtime costs making it difficult for targeting your market. A professional has to be hired to design an efficient, well-crafted and effective script.

Our proposed idea helps in making the advertising more efficient and cost effective. This assists in the targeted marketing strategy.

This paper describes how the proposed idea is more effective than traditional advertising methods. The section III describes the requirements needed for the proposed idea and design of the proposed model. It is followed by the implementation of the model and the experimental results of the model.

**CHAPTER**

**TECHNOLOGY USED**

To successfully develop any application, various software technologies and platforms have to be selected and interconnected such they seamlessly work along with each other. The language used to develop an application an application plays a crucial role and should be selected according to the requirements of the application. Real time applications require procedure oriented languages such as C, while flexible object oriented languages such as java can be used for other applications. Our application, which has been developed using java language, runs on the android platform. Various other technologies can be used depending on its ease of use and requirement of the application.

**Raspberry pi**

The Raspberry Pi is a series of credit card–sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation with the intent to promote the teaching of basic computer science in schools and developing countries. It is capable of doing everything a desktop does like browsing video streaming etc. It has a Broadcom BCM2836 Arm7 Quad Core Processor powered Single Board Computer running at 900MHz,1GB RAM so you can now run bigger and more powerful applications, Micro SD slot for storing information and loading your operating systems. You can now provide up to 1.2 AMP to the USB port – enabling you to connect more power hungry USB devices directly to the Raspberry PI (This feature requires a 2Amp micro USB Power Supply) and 10/100 Ethernet Port to quickly connect the Raspberry Pi to the Internet. Most boards have between one and four USB slots, HDMI and composite video output, and a 3.5 mm phono jack for audio. Lower level output is provided by a number of GPIO pins which support common protocols like I2C. Some models have an RJ45 Ethernet port and the Pi 3 has on board WiFi 802.11n and Bluetooth.

The system on a chip (SoC) used in the first generation Raspberry Pi is somewhat equivalent to the chip used in older smartphones (such as iPhone, 3G, 3GS). The Raspberry Pi is based on the Broadcom BCM2835 SoC, which includes an 700 MHz ARM1176JZF-S processor, VideoCore IV graphics processing unit (GPU), and RAM. It has a Level 1 cache of 16 KB and a Level 2 cache of 128 KB. The Level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible.The Raspberry Pi 2 uses a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor, with 256 KB shared L2 cache.

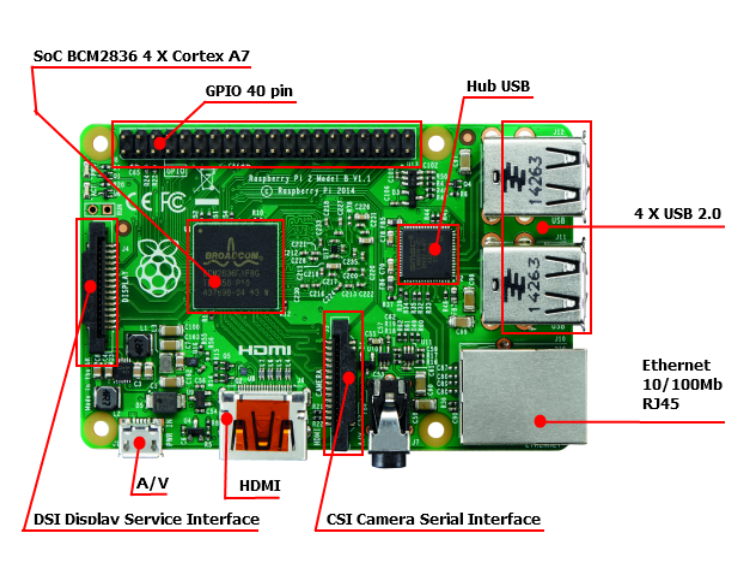


Fig 1. Architecture of raspberry pi 2

**General purpose input-output (GPIO) connector**

RPi A+, B+, 2B and Zero GPIO J8 have a 40-pin pinout.Model 3 has 40 pins as well, but someone will need to confirm that the pin layout is the same as its predecessor. Models A and B have only the first 26 pins.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **GPIO#** | **2nd func.** | **Pin#** |  | **Pin#** | **2nd func.** | **GPIO#** |
|  | +3.3 V | 1 |  | 2 | +5 V |  |
| 2 | SDA1 (I2C) | 3 |  | 4 | +5 V |  |
| 3 | SCL1 (I2C) | 5 |  | 6 | GND |  |
| 4 | GCLK | 7 |  | 8 | TXD0 (UART) | 14 |
|  | GND | 9 |  | 10 | RXD0 (UART) | 15 |
| 17 | GEN0 | 11 |  | 12 | GEN1 | 18 |
| 27 | GEN2 | 13 |  | 14 | GND |  |
| 22 | GEN3 | 15 |  | 16 | GEN4 | 23 |
|  | +3.3 V | 17 |  | 18 | GEN5 | 24 |
| 10 | MOSI (SPI) | 19 |  | 20 | GND |  |
| 9 | MISO (SPI) | 21 |  | 22 | GEN6 | 25 |
| 11 | SCLK (SPI) | 23 |  | 24 | CE0\_N (SPI) | 8 |
|  | GND | 25 |  | 26 | CE1\_N (SPI) | 7 |
| *(RPi 1 Models A and B stop here)* | | | | | | |
| EEPROM | ID\_SD | 27 |  | 28 | ID\_SC | EEPROM |
| 5 | N/A | 29 |  | 30 | GND |  |
| 6 | N/A | 31 |  | 32 |  | 12 |
| 13 | N/A | 33 |  | 34 | GND |  |
| 19 | N/A | 35 |  | 36 | N/A | 16 |
| 26 | N/A | 37 |  | 38 | Digital IN | 20 |
|  | GND | 39 |  | 40 | Digital OUT | 21 |

Model B rev. 2 also has a pad (called P5 on the board and P6 on the schematics) of 8 pins offering access to an additional 4 GPIO connections.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Function** | **2nd func.** | **Pin#** |  | **Pin#** | **2nd func.** | **Function** |
| N/A | +5 V | 1 |  | 2 | +3.3 V | N/A |
| GPIO28 | GPIO\_GEN7 | 3 |  | 4 | GPIO\_GEN8 | GPIO29 |
| GPIO30 | GPIO\_GEN9 | 5 |  | 6 | GPIO\_GEN10 | GPIO31 |
| N/A | GND | 7 |  | 8 | GND | N/A |

Models A and B provide GPIO access to the ACT status LED using GPIO 16. Models A+ and B+ provide GPIO access to the ACT status LED using GPIO 47, and the power status LED using GPIO 35.

**Operating systems**

The Raspberry Pi primarily uses Linux-kernel-based operating systems.

The ARM11 chip at the heart of the Pi (first generation models) is based on version 6 of the ARM. The primary supported operating system is Raspbian, although it is compatible with many others. The current release of Ubuntu supports the Raspberry Pi 2, while Ubuntu, and several popular versions of Linux, do not support the older Raspberry Pi 1 that runs on the ARM11. Raspberry Pi 2 can also run the Windows 10 IoT Core operating system, while no version of the Pi can run traditional Windows. The Raspberry Pi 2 currently also supports OpenELEC and RISC OS.

Raspbian – is maintained independently of the Foundation based on the Debian ARM hard-float (armhf) architecture port originally designed for ARMv7 and later processors (with Jazelle RCT/ThumbEE and VFPv3), compiled for the more limited ARMv6 instruction set of the Raspberry Pi. A minimum size of 4 GB SD card is required for the Raspbian images provided by the Raspberry Pi Foundation. There is a Pi Store for exchanging programs.

**Driver APIs**

Raspberry Pi can use a VideoCore IV GPU via a binary blob, which is loaded into the GPU at boot time from the SD-card, and additional software, that initially was closed source. This part of the driver code was later released. However, much of the actual driver work is done using the closed source GPU code. Application software use calls to closed source run-time libraries (OpenMax, OpenGL ES or OpenVG) which in turn calls an open source driver inside the Linux kernel, which then calls the closed source VideoCore IV GPU driver code. The API of the kernel driver is specific for these closed libraries. Video applications use OpenMAX, 3D applications use OpenGL ES and 2D applications use OpenVG which both in turn use EGL. OpenMAX and EGL use the open source kernel driver in turn.

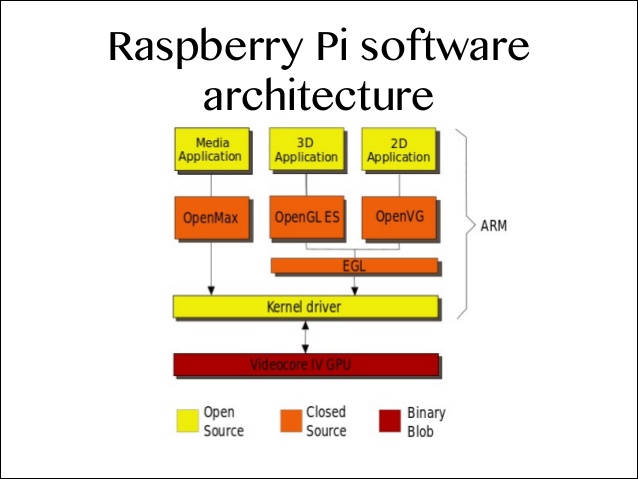


Fig 2. Raspberry pi software architecture

**Android**

Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as Smartphone and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input.

**Android SDK**

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows XP or later. As of March 2015, the SDK is not available on Android itself, but the software development is possible by using specialized Android applications.

Until around the end of 2014, the officially supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. As of 2015, Android Studio, made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others. Additionally, 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).

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

**Java standards**

Obstacles to development include the fact that Android does not use established Java standards, that is, Java SE and ME. This prevents compatibility between Java applications written for those platforms and those written for the Android platform. Android only reuses the Java language syntax and semantics, but it does not provide the full class libraries and APIs bundled with Java SE or ME. However, there are multiple tools in the market from companies such as Myriad Group and UpOnTek that provide Java ME to Android conversion services.

**Java**

Java is a set of computer software and specifications developed by Sun Microsystems, which was later acquired by the Oracle Corporation, that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. While they are less common than standalone Java applications, Java applets run in secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages.

Writing in the Java programming language is the primary way to produce code that will be deployed as byte code in a Java Virtual Machine (JVM); byte code compilers are also available for other languages, including Ada, JavaScript, Python, and Ruby. In addition, several languages have been designed to run natively on the JVM, including Scala, Clojure and Groovy. Java syntax borrows heavily from C and C++, but object-oriented features are modeled after Smalltalk and Objective-C. Java eschews certain low-level constructs such as pointers and has a very simple memory model where every object is allocated on the heap and all variables of object types are references. Memory management is handled through integrated automatic garbage collection performed by the JVM.

The Java platform is a suite of programs that facilitate developing and running programs written in the Java programming language. A Java platform will include an execution engine (called a virtual machine), a compiler and a set of libraries; there may also be additional servers and alternative libraries that depend on the requirements. Java is not specific to any processor or operating system as Java platforms have been implemented for a wide variety of hardware and operating systems with a view to enable Java programs to run identically on all of them. Different platforms target different classes of device and application domains.

**Java Virtual Machine**

The heart of the Java platform is the concept of a "virtual machine" that executes Java bytecode programs. This bytecode is the same no matter what hardware or operating system the program is running under. There is a JIT (Just In Time) compiler within the Java Virtual Machine, or JVM. The JIT compiler translates the Java bytecode into native processor instructions at run-time and caches the native code in memory during execution.

The use of bytecode as an intermediate language permits Java programs to run on any platform that has a virtual machine available. The use of a JIT compiler means that Java applications, after a short delay during loading and once they have "warmed up" by being all or mostly JIT-compiled, tend to run about as fast as native programs. Since JRE version 1.2, Sun's JVM implementation has included a just-in-time compiler instead of an interpreter.

Although Java programs are cross-platform or platform independent, the code of the Java Virtual Machines (JVM) that execute these programs is not. Every supported operating platform has its own JVM.