IOT BASES HOME AUTOMATION USING RASPBERRI PI AND ANDROID

Joshiga santh, Department of Electronics and Communication Engineering
Email: joshigasanth94@gmail.com

Abstract-*In this busy and comfortable lifestyle of people, communication technology has evolved in such a way that any information can be accesses from anywhere at any time by anyone using INTERNET. Internet is a globally connected network system that uses TCP/IP to transmit/Receive data via various types of media. Home applications like LIGHT, CLAMATIZATION , ETC could be controlled by using one of the internet media.. This report represents and shows the application of IOT BASED HOME AUTOMATION USING RASPBERRY PI AND ANDROID which includes a Raspberry Pi 3 as a processing unit for data which is extracted from server computer which is created in Linux based operating system. In this project, we used three applications named app1, app2 and app3. These three are controlled by using android application which is in customer side. This android application was developed by using android studio. This used to control the home applications in remote locations.*

Keywords: *Internet, IOT, server computer, Linux, Android application, JSON*

INTRODUCTION:

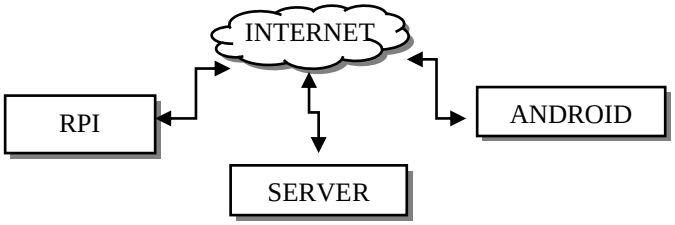

Fig 1.1

We designed a system which will control the electrical applications in home or industries by using android application from remote location. There are three main modules in this system they are RASPBERRY PI 3, LINUX SYSTEM with SERVER installed and ANDROID MOBILE with developed application called IOT. This application allows users to controlled the electrical applications in home through internet.

PROPOSED SYSTEM:

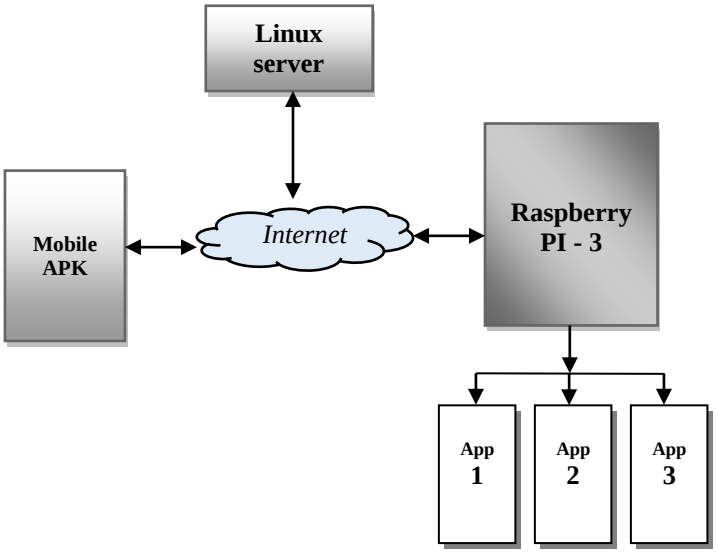

Fig 1.2

There are three systems which are run separately in remote location by connecting through internet.

- ✓ Android mobile are handled by customer side
- ✓ Raspberry Pi 3 (computing system) are connected in application area where the electrical applications need to controlled.
- ✓ Server system are connected to internet in server area.

All the applications are controlled through relay circuit relay circuit is not shown in this paper.

ANDROID APK:

Fig 1.3

A mobile app is a computer program designed to run on a mobile device such as a phone/tablet or watch.

Mobile applications often stand in contrast to desktop applications that run on desktop computers, and with web

applications which run in mobile web browsers rather than directly on the mobile device.

Android Package (APK) is the package file format used by the Android operating system for distribution and installation of mobile apps and middleware.

Apk used in this project were developed by using android studio.

Development:

Developing apps for mobile devices requires considering the constraints and features of these devices. Mobile devices run on battery and have less powerful processors than personal computers and also have more features such as location detection and cameras. Developers also have to consider a wide array of screen sizes, hardware specifications and configurations because of intense competition in mobile software and changes within each of the platforms.

Mobile application development requires use of specialized integrated development environments. Mobile apps are first tested within the development environment using emulators and later subjected to field testing. Emulators provide an inexpensive way to test applications on mobile phones to which developers may not have physical access.

ANDROID STUDIO:

Fig 1.4

Android Studio is the official integrated development environment for Google's android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for android development.

System Requirements

Android application development may be performed on any of the following system types:

- Windows 2003 (32-bit or 64-bit)
- Windows Vista (32-bit or 64-bit)
- Windows 7 (32-bit or 64-bit)
- Windows 8 / Windows 8.1/ Windows 10
- Mac OS X 10.8.5 or later (Intel based systems only)
- Linux systems with version 2.11 or later of GNU C Library (glibc)
- Minimum of 2GB of RAM (4GB is preferred)
- 1.5GB of available disk space

LINUX:

Fig 1.5

Ubuntu is an open source operating system for computers. It is a Linux distribution based on the Debian architecture. It is usually run on personal computers, and is also popular on network servers, usually running the Ubuntu Server variant, with enterprise-class features. Ubuntu runs on the most popular architectures, including Intel, AMD, and ARM-based machines. Ubuntu is also available for tablets and smartphones, with the Ubuntu Touch edition.

Data from android mobile and data from Raspberry Pi 3 were saved in database which were created in Linux operating system with configuration of AMD processor,4GB RAM and 500gb hard disk.

RASPBERRY PI 3:

Fig 1.6

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

Processor:

The Broadcom BCM2835 SoC used in the first generation Raspberry Pi is somewhat equivalent to the chip used in first modern generation smartphones, which includes a 700 MHz ARM1176JZF-S processor, VideoCoreIV graphics processing unit (GPU), and RAM. It has a level 1 (L1) cache of 16 KB and a level 2 (L2) 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 earlier V1.1 model of the Raspberry Pi 2 used a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor, with 256 KB shared L2 cache. The Raspberry Pi 2 V1.2 was upgraded to a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, the same SoC which is used on the Raspberry Pi 3, but underclocked (by default) to the same 900

MHz CPU clock speed as the V1.1. The BCM2836 SoC is no longer in production (as of late 2016).

The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

Networking:

The Model A, A+ and Pi Zero have no Ethernet circuitry and are commonly connected to a network using an external user-supplied USB Ethernet or Wi-Fi adapter. On the Model B and B+ the Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip. The Raspberry Pi 3 and Pi Zero W (wireless) are equipped with 2.4 GHz WiFi 802.11n (150 Mbit/s) and Bluetooth 4.1 (24 Mbit/s) based on Broadcom BCM43438 Full MAC chip with no official support for Monitor mode but implemented through unofficial firmware patching and the Pi 3 also has a 10/100 Ethernet port.

Memory:

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. On the first 256 MB release Model B (and Model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding, or for simple 3D, but probably not for both together. 224 MB was for Linux only, with only a 1080p framebuffer, and was likely to fail for any video or 3D. 128 MB was for heavy 3D, possibly also with video decoding (e.g. XBMC). Comparatively the Nokia 701 uses 128 MB for the Broadcom VideoCore IV.

For the later Model B with 512 MB RAM initially there were new standard memory split files released (arm256_start.elf, arm384_start.elf, arm496_start.elf) for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM). But a week or so later the RPF released a new version of start. elf that could read a new entry in config.txt (gpu_mem = xx) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single start.elf worked the same for 256 and 512 MB Raspberry Pis.

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero and Zero W have 512 MB of RAM.

Specifications

- **SoC:** Broadcom BCM2837
- **CPU:** 4× ARM Cortex-A53, 1.2GHz
- **GPU:** Broadcom VideoCore IV
- **RAM:** 1GB LPDDR2 (900 MHz)
- **Networking:** 10/100 Ethernet, 2.4GHz 802.11n wireless
- **Bluetooth:** Bluetooth 4.1 Classic, Bluetooth Low Energy
- **Storage:** microSD
- **GPIO:** 40-pin header, populated
- **Ports:** HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera
- Serial Interface (CSI), Display Serial Interface (DSI)

APPLICATIONS:

Applications would be anything like Fan, Light, Television, Refrigerator ETC.

OUTPUT AND RESULT:

1.Output running in python shell

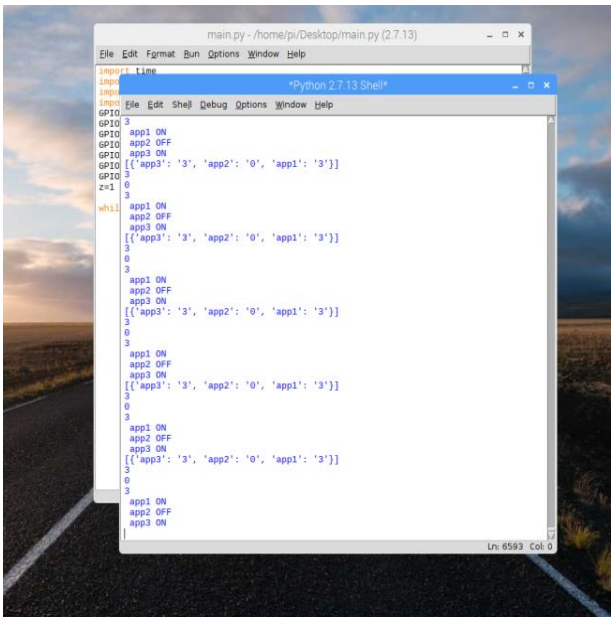

Fig 1.7

Fig 1.7 shows ON and OFF states of individual application through python shell.

2.Output through Relay

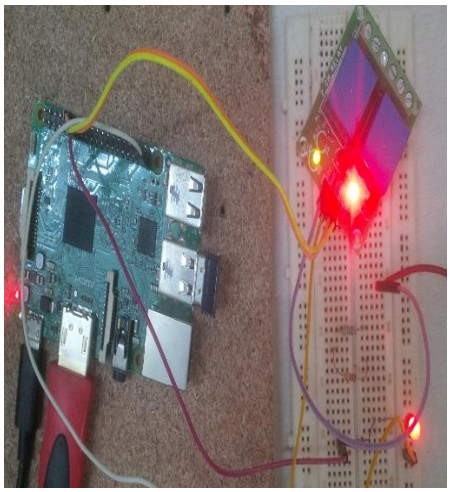

Fig 1.8

3.During server in OFF state

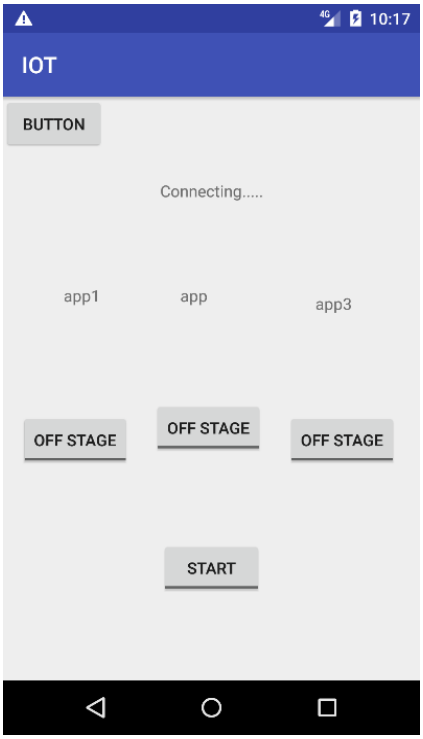

Fig 1.9

4.During Server ON state

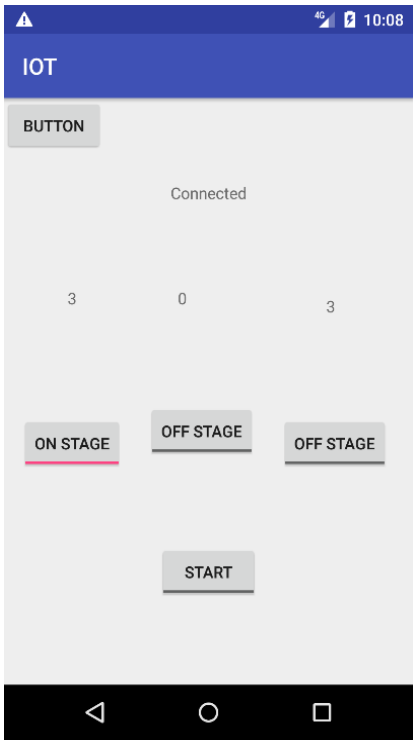

Fig 1.10

CONCLUSION & FURTHER DEVELOPMENT:

As mentioned, in **Output and result** section, interfacing android application to raspberry pi using internet were interfaced successfully. The data which were sending and receiving from and to between android mobile and raspberry pi will be **encrypted** to protect the data from other external interferences.

By adding **Image processing**, **RFID** authentication for automobile would improve the security for the smart system.

REFRENCES:

[1]: https://www.android-examples.com/set-setoncheckedchangelistener-to-switch-button-in-android/
[2]: https://www.androidhive.info/2011/11/android-sqlite-database-tutorial/
[3]: https://www.w3schools.com/js/js_json_intro.asp