A Prototype of Home Automation System using Android App

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

by

Sumiya Sayeed

CSE 052 06619

&

Md Kamrul Hasan

CSE 052 06577

Supervised by: Tamanna Haque Nipa

Assistant Professor

Department of Computer Science and Engineering

STAMFORD UNIVERSITY BANGLADESH

November 2017

Abstract

Home automation refers to the control of home appliances and domestic features by local networking or by remote control. Artificial Intelligence provides us the framework to go real-time decision and automation for Internet of Things (IoT). The home automation becomes important, because it gives the user the comfortable and easily for using the home devices. The implementation and design of wireless home automation control uses two methods, WLAN technology and RF remote control handheld to control of the selective home devices with integral security and protected system. Heterogenerous home automation systems and technologies considered in review with central controller based (Arduino or Raspberry pi), web based, email based, Bluetooth-based, mobile-based, SMS based, ZigBee based, Dual Tone Multi Frequency-based, cloud-based and the Internet with performance. In this project, we built a prototype for smart home automation that does the simple deeds like switching on/off of lights and fans.

Approval

The project report "A Prototype of Home Automation System using Android App" submitted by Sumiya Sayeed ID: CSE 052 06619, Md Kamrul Hasan ID: CSE 052 06577, to the Department of Computer Science & Engineering, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (B.Sc.) in Computer Science & Engineering and as to its style and contents.

Board of Examiner's Name, Signature and Date:

……………………………...	**……………………………..**	**……………………………..**
Tanveer Ahmed	**Mahfida Amjad Dipa**	**Maliha Mahbub**
(Board Member 1)	**(Board Member 2)**	**(Board Member 3)**
Date:	Date:	Date:

Supervisor's Signature and Date:

……………………………...

Tamanna Haque Nipa

Date:

Declaration

We, hereby, declare that the work presented in this Project is the outcome of the investigation performed by us under the supervision of Tamanna Haque Nipa, Assistant Professor, Department of Computer Science & Engineering, Stamford University Bangladesh. We also declare that no part of this Project and thereof has been or is being submitted elsewhere for the award of any degree or Diploma.

Signature and Date:

……………………………...

Student Name: Md Kamrul Hasan

Date:

……………………………...

Student Name: Sumiya Sayeed

Date:

Dedicated to our respected parents

Acknowledgements

First and foremost, we are grateful to Allah, the Almighty, the Merciful without whose blessing, this project would not have been successful. Allah gave us confidence, courage and determination to overcome the obstacles we faced during this journey.

We would like to express our sense of gratitude to Tamanna Haque Nipa, Assistant Professor of the Department of Computer Science & Engineering, Stamford University Bangladesh, for her unconditional guidance, insights, immense patience, unlimited encouragement and inspirations throughout this whole project. She is the one who inspired us to take this project and supported us every step of the way in every possible way. Without her, this project would not have come to fruition. She provided the valuable technical advice and pointed us in the right directions which were much required for a project like this. We are truly grateful for having a teacher like her as our supervisor.

We specially owe thanks to all the teachers of the faculty of Computer Science & Engineering, Stamford University Bangladesh for their help, valuable suggestions and discussions.
Moreover, we would like to thank all our friends who have supported and helped us throughout this project.

We would like to give our special thanks to Mohammad Manzurul Islam Sir.

Last of all we would also like to thank our parents who have walked along with us in every step of our life and always encouraged and inspired us to do greater things in our life.

Table of Contents

List of Figures

List of Table

1. Introduction

The Internet of things (IoT) is the inter-networking of physical devices, vehicles (also referred to as connected devices and smart devices), buildings and other items embedded with electronics, software, sensors, actuators and network connectivity which enable these objects to collect and exchange data. It represents a general purpose for the capability of network devices to sense and collect data from the world around us and then share that data across Internet where it can be processed and utilized for various interesting purposes.

1.1 Idea of IoT (Internet of Things)

The Internet of Things is the interconnected sphere of physical devices with the Internet and other networks through uniquely identifiable IP addresses, whereby data is gathered and communicated through embedded sensors, electronics and software. [1]

In other word, the Internet of Things is the interconnection of endpoints (devices and things) which can be uniquely addressed and identified with an IP (Internet Protocol) address. With the Internet of Things, devices can be connected to the Internet, sense, gather, receive and send data and communicate with each other and applications via IP technologies, platforms and connectivity solutions.

1.1.1 Common elements in Internet of Things definitions

We can define IoT (Internet of Things) in seven characteristics. These are as follows:

1.1.1.1 Internet of Things Connectivity

A dimension of networks and connectedness needs to be present in any decent IoT definition.

1.1.1.2 The things in the Internet of Things

Devices, physical objects, sensors, the physical world, appliances, endpoints these are essential parts of a network of things as shown in figure 1.1.

1.1.1.3 The Internet of Things and Data

Data is a crucial part of IoT (Internet of Things). Although it is just not enough for Internet of Things but there is not Internet of Things without (big) data. Here data brings information shown in figure 1.1.

1.1.1.4 Communication in the Internet of Things

The data gathered and sensed by IoT devices needs to be communicated in order to start turning it into actionable information. [1]

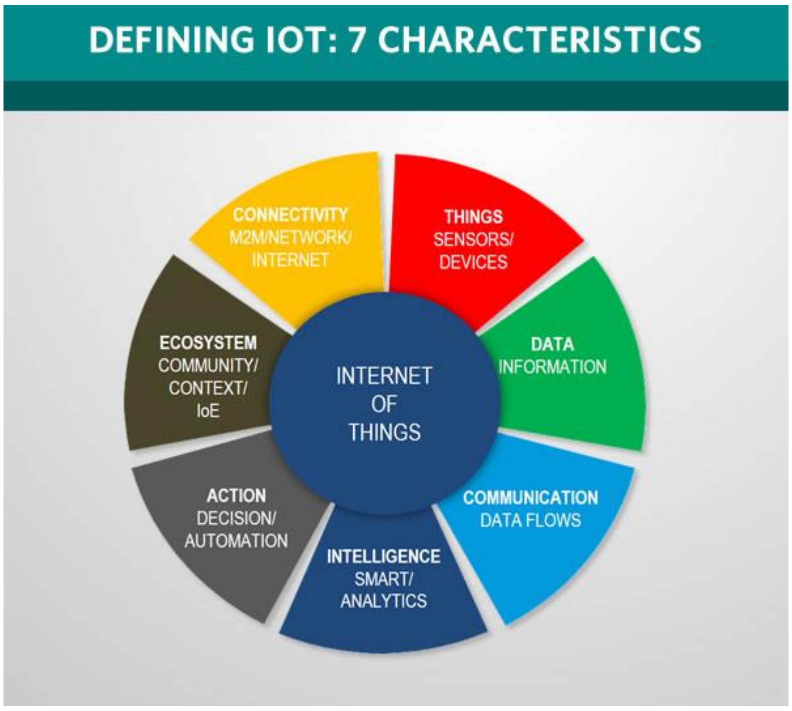

DEFINING IOT: 7 CHARACTERISTICS
CONNECTIVITY M2M/NETWORK/ INTERNET
THINGS SENSORS/ DEVICES
DATA INFORMATION
COMMUNICATION DATA FLOWS
INTELLIGENCE SMART/ ANALYTICS
ACTION DECISION/ AUTOMATION
ECOSYSTEM COMMUNITY/ CONTEXT/ IoE
INTERNET OF THINGS

Fig 1.1: Defining the Internet of Things using 7 characteristics.

1.1.1.5 Internet of Things, Intelligence and action

While connected devices have a capacity of action, the real intelligence and action sits in the analysis of the data and the smart usage of this data to solve a challenge, create a competitive benefit, to automate a process or to improve something. There is no Internet of Things without (big) data, there is no useful Internet of Things deployment without understanding meaning, intelligence, (big) data analytics, cognitive and AI and so on.

1.1.1.6 Automation

Most IoT applications are essentially all about automation. Industrial automation, business process automation or the automatic updating of software play a role, depending on the context.

1.1.1.7 Ecosystem

We can say that this isn't strictly about the Internet of Things but more about the Internet of Everything or the Internet of Things ecosystem. [1]

1.1.2 Understanding the Internet of Things

First IoT devices and embedded intelligence should be connected. Then the sensors and devices sense and captures data from different contexts. Then they communicate transport the data over a network via Internet or cloud. The whole system is shown in figure 1.2.

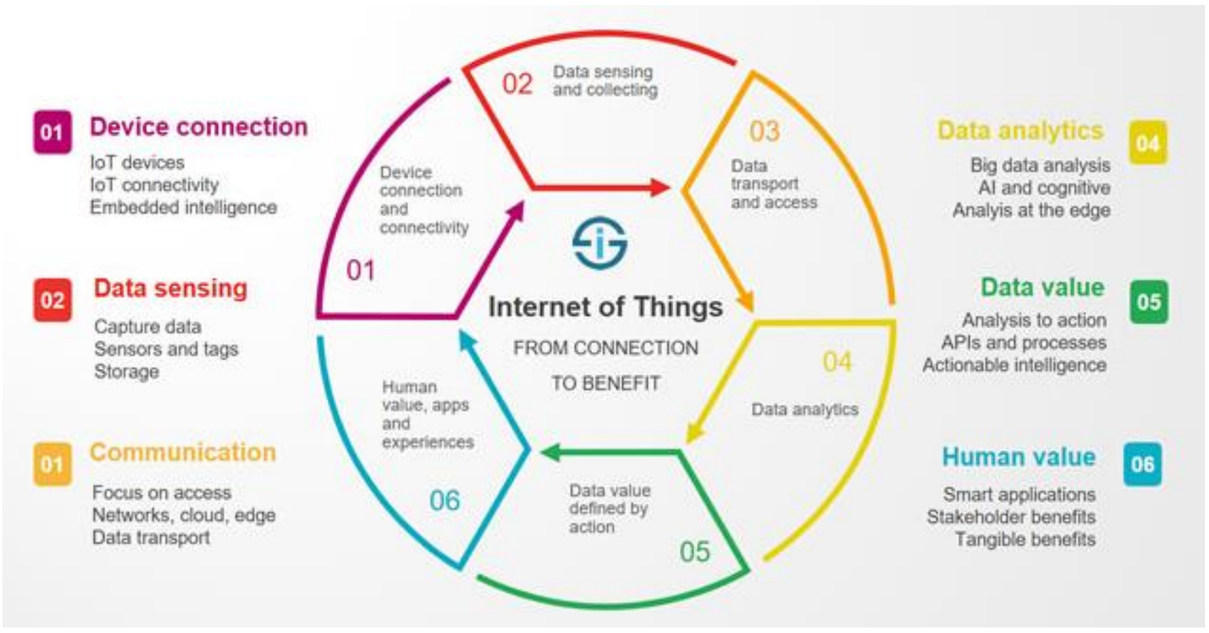


Fig 1.2: The Internet of Things from connecting devices to human value.

In combination with big data and IoT analytics, actuators, data hubs, artificial intelligence, connectivity and networks, the cloud, information processes, business process optimization, people, smart goals, increasingly robotics, IoT platforms/middleware and new ecosystems of value, the Internet of Things enables an unseen new wave of innovation and optimization. [1]

1.1.3 Business Opportunity

The wave of connectivity is going beyond laptops and smartphones, it is going towards connected cars, smart homes, connected wearables, smart cities and connected healthcare, basically a connected life.

Table 1.1: HP survey result in estimating the connected devices.

Year	Number of Connected Devices
1990	0.3 million
1999	90 million
2010	5 billion
2013	9 billion
2025	1 trillion

HP did a small survey in which they estimated the rise of connected devices over the years and the results are surprising. Eventually we are moving towards a fully automated world. These devices will bridge the gap between physical and digital world to improve the quality and productivity of life, society and industries.

A survey conducted by KRC research in UK, US, Japan and Germany the early adopters of IOT has revealed which devices are the customers more likely to use in the coming years. Smart Appliances like thermostat, smart refrigerator to name a few are most liked by the customers and are seem to change the way we operate.

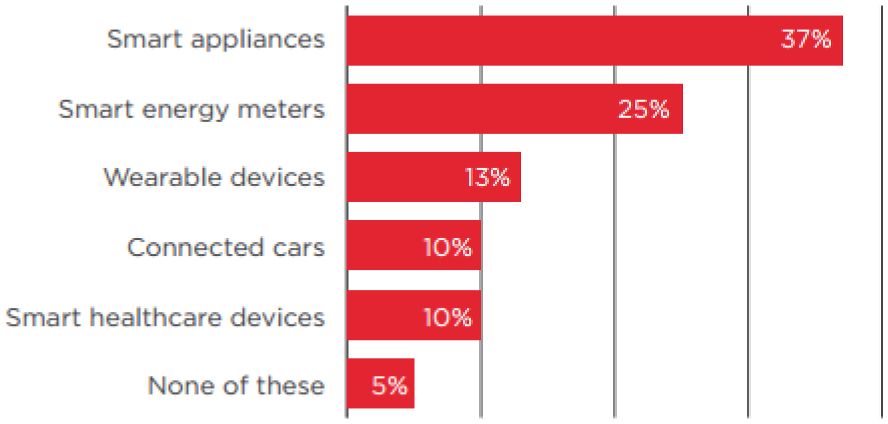


Fig 1.3: A suvey result of IoT appliances.

According to the Cisco report IoT will generate $14.4 trillion in value across all industries in the next decade. Surprisingly, IoT will bring a wave which nobody can foresee. [2]

1.1.4 Real World Applications of IoT

- **Smart Home**

With IoT creating the buzz, 'Smart Home' is the most searched IoT associated feature on Google. Smart Home has become the revolutionary ladder of success in the residential spaces and it is predicted Smart homes will become as common as smartphones.

- **Wearables**

Wearables have experienced an explosive demand in markets all over the world. Google, Samsung have invested heavily in building such devices. Wearable devices are installed with sensors and softwares which collect data and information about the users. This data is later pre-processed to extract essential insights about user.

These devices broadly cover fitness, health and entertainment requirements. The pre-requisite from internet of things technology for wearable applications is to be highly energy efficient or ultra-low power and small sized.

- **Connected Cars**

A connected car is a vehicle which is able to optimize its own operation, maintenance as well as comfort of passengers using onboard sensors and internet connectivity.

Most large auto makers as well as some brave startups are working on connected car solutions. Major brands like Tesla, BMW, Apple, Google are working on bringing the next revolution in automobiles.

- **Industrial Internet**

Industrial Internet is the new buzz in the industrial sector, also termed as Industrial Internet of Things (IIoT). IIoT holds great potential for quality control and sustainability. Applications for tracking goods, real time information exchange about inventory among suppliers and retailers and automated delivery will increase the supply chain efficiency.

- **Smart Cities**

Smart city is another powerful application of IoT generating curiosity among world's population. Smart surveillance, automated transportation, smarter energy management systems, water distribution, urban security and environmental monitoring all are examples of internet of things applications for smart cities. IoT will solve major problems faced by the people living in cities like pollution, traffic congestion and shortage of energy supplies etc.

- **IoT in agriculture**

With the continuous increase in world's population, demand for food supply is extremely raised. Smart farming is one of the fastest growing field in IoT. Farmers are using meaningful insights from the data to yield better return on investment. Sensing for soil moisture and nutrients, controlling water usage for plant growth and determining custom fertilizer are some simple uses of IoT.

- **Smart Retail**

The potential of IoT in the retail sector is enormous. IoT provides an opportunity to retailers to connect with the customers to enhance the in-store experience.

- **IoT in Healthcare**

Connected healthcare yet remains the sleeping giant of the Internet of Things applications. Research shows IoT in healthcare will be massive in coming years. IoT in healthcare is aimed at empowering people to live healthier life by wearing connected devices. [2]

1.2 Objective

Smart home is the place where everything is connected & controlled. Simply where electronics talk & live as neighbors. We have developed a home automation system in which we have used a Raspberry pi, an Android Application with UI design, APACHE 2 server, a Wi-Fi connection, a buzzer, some lights and a fan.

Home automation or smart home is one of the most recently used and interesting IoT projects of time. The objective of this project is to develop a Home automation system in which client can be able to control the light or fan by switching on or off and to press a buzzer from the mobile application via Wi-Fi.

1.3 Scope:

This project can play a vital role for make life easier. This project has been implemented with a low cost whereas anyone can use it, specially disabled person for make their everyday life easier. This system can be used as controlling over the electro device like on/off light, fan or other electronics accessories via a mobile application.

In future, Face recognition, Biometric Door Lock, Computer Vision will be added for better security. When intruder tries to break the door, the vibration is sensed by sensor which makes an alarm. This will inform the neighbors or security about intruders and this will help to take further action to prevent intruder from entering. Or, if the face recognition system is unable to recognize the stranger face, it will notify the user.

1.4 Overview:

In this chapter, we explore the concept of IoT, its elaboration in detail, business opportunity and real world applications, home automation systems and its prospects in daily life with its huge benefits such as safety for babies and physically disabled people.

In next chapter we discuss about the literature review of the home automation system, its characteristics, applications etc as well as we discuss about some existing home automation devices. In 3rd chapter, we introduce about the hardware and software tools that we use in this project. In 4th chapter, we elaborate the hardware and software design of our system and in 5th chapter, we explore our project in detail.

And finally in 6th chapter, we conclude with some limitations of our system, future plans and possibities in our country.

2. Literature Review

In this chapter we will discuss about the home automation system. We will also compare with some existing project which is almost similar to our project.

2.1 Home automation:

2.1.1 Definition:

Home automation, also known as “domotics” which is a contraction of the Latin word for a home (domus) and the word robotics, gives access to control devices in home from a mobile device anywhere in the world. Home automation describes homes in which nearly everything like lights, appliances, electrical outlets, heating and cooling systems are hooked up to a remotely controllable network. From a home security perspective, this also includes alarm system, and all of the doors, windows, locks, smoke detectors, surveillance cameras and any other sensors that are linked to it. [3]

2.1.2 Characteristic of home automation:

There are two main characteristics of home automation. These are as follows:

1. Automation: Automation refers to the ability to program and schedule events for the devices on the network. The programming may include time-related commands, such as having lights turn on or off at specific times each day. It can also include non-scheduled events, such as turning on all the lights in home when the security system alarm is triggered.

2. Remote Control: The other main characteristic of cutting-edge home automation is remote monitoring and access. While a limited amount of one-way remote monitoring has been possible for some time, it's only since the rise in smartphones and tablets that the ability to truly connect to home networks while client is away. With the right home automation system, client can use any Internet-connected device to view and control the system itself and any attached devices. [3]

2.1.3 History of home automation:

Early home automation began with labor-saving machines. Self-contained electric or gas powered home appliances became viable in the 1900s with the introduction of electric power distribution and led to the introduction of washing machines (1904), water heaters (1889), refrigerators, sewing machines, dishwashers, and clothes dryers.

In 1975, the first general purpose home automation network technology, X10, was developed. It is a communication protocol for electronic devices. It primarily uses electric power transmission wiring for signaling and control, where the signals involve brief radio frequency bursts of digital data, and remains the most widely available. By 1978, X10 products included a 16 channel command console, a lamp module, and an appliance module. Soon after came the wall switch module and the first X10 timer. [3]

2.1.4 Generations of home automation:

Home automation is generally divided into three generations:

i. Wireless technology (First generation)

ii. Artificial intelligence (Second generation)

iii. Robot buddy (Third generation)

2.1.5 Protocols:

The home automation devices can be wired or wireless, depends on the technology we choose, actually the protocols. The widely used protocols are Z Wave, Zigbee, EnOcean, KNX, X 10 etc. Each protocol has specific media, data transmission & reception rate and application. [4]

Table 2.1: Specific media, data transmission & reception rate and application of the protocols.

Protocol	Communication	Rate
Zigbee	Radio Frequency	20~250 Kbps
Z Wave	Radio Frequency	100 Kbps
EnOcean	Radio Frequency / PLC	9600 bps
X 10	Radio Frequency / PLC	20 bps
Universal Power Bus(UPB)	PLC (Power Line Communication)	480 bps
KNX	RF, PLC, Twisted Pair, Infrared Ethernet	9600 bps

2.1.6 Applications:

The devices controlled using smart home technology are many but not limited to

- Heating, ventilation and air conditioning (HVAC).
- Lighting control system
- Security system
- Access control
- Intrusion detection and alarm system
- Gas detector
- Home automation for the elderly and disabled people
- Swimming pool alarm system
- Audion and video system and so on.

2.2 Existing Home Automation Systems

There are tons of home automation systems running in the current market. Some of them are as follows:

2.2.1 Nest Cam Indoor:

With 24/7 live streaming, a versatile magnetic stand, person alerts with Nest Aware and one app for Nest products, Nest Cam Indoor helps to keep an eye on what matters from anywhere. Some cameras can't stay on all the time because they rely on batteries. So they only start recording when they sense motion.

5089220

Figure 2.1: Nest Cam.

With a Nest Aware subscription, Nest Cam continuously records 24/7 and saves up to 30 days of footage securely in the cloud. Thus Nest Cam secure home from crime and intruders. [5]

2.2.2 Alexa:

Amazon echo is one of the famous home automation device in recent days. It can do many household chores like

- Streaming music and podcasts from Amazon Music Unlimited, Spotify, Pandora and iHeartRadio.
- Adding items to your to-do list and shopping list.
- Setting kitchen timers and recurring alarms.

Figure 2.2: Alexa (Amazon Echo).

- Looking up facts and unit conversions.
- Playing a curated "flash briefing" of news headlines from the sources and topics of your choice.
- Controlling compatible smart home gadgets, including lights, locks and thermostats. [6]

2.2.3 Jarvis:

Today’s most successful home automation system is Jarvis. It is an AI system that Mark Zuckerberg, CEO of Facebook has built to control his home and perform basic tasks, such as-

- Turning the lights off or on
- Controlling a particular room’s temperature
- Playing music
- Opening doors
- Recognizing user’s voice and the context of the command
- Telling the schedule for the day

- Teaching languages
- Informing the user what others in the home doing
- Setting up video conferences
- Playing movies
- Even cracking a few jokes
- Offering to make a toast when user looks to whip up some breakfast and so on. [7]

3. Tools

In this chapter we will discuss about all the hardware and software which is require to build up this project.

3.1 Hardware

The hardware tools we have used for this project are as follows

3.1.1 Raspberry Pi 3 Model B

The Raspberry Pi is a very small and low-cost computer which has the size of a credit card and can be plugged into any computer monitor or TV. It uses a standard keyboard and mouse. It has all the capability of a desktop like browsing the Internet, playing high definition video, making spreadsheets, word processing and playing games. What is unique to Raspberry Pi is that it allows the user to interact with the outside world and is currently used in many digital projects. Users can learn to write programs by using languages like Scratch and Python. [9]

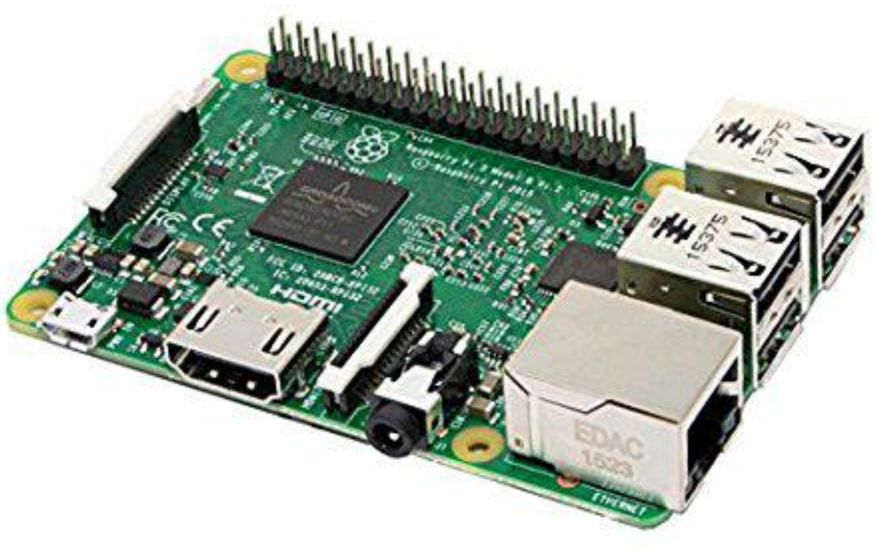


Figure 3.1: Raspberry Pi 3 model B

The Multipurpose Surveillance Robot has utilized the Raspberry Pi 3 Model B which is the second generation Raspberry Pi which replaced the original Raspberry Pi 3 Model B+.
It has:

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- USB ports

- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5 mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- Video Core IV 3D graphics core

3.1.2 Raspberry Pi 3 Model B Required Accessories

3.1.2.1 Micro SD Card

An 8 GB class 4 micro SD card with Raspbian jessie pre-installed and another 8 GB class 4 micro SD card with Android Things pre-installed, is recommended. The minimum recommended capacity of an SD card is 8 GB. 4 GB is recommended for image installation. Even smaller cards can be used for some distributions like OpenElec8 and Arch. The SD card class determines the write speed a card can sustain. A class 4 SD card can achieve 4 MB/s write speed whereas for a class 10 card 10 MB/s is attainable.

3.1.2.2 Display and Connectivity Cable

Any HDMI/DVI monitor and any TV works as display for pi. But one with an HDMI input is recommended.

3.1.2.3 Keyboard and Mouse

Raspberry pi works with any standard keyboard and mouse. Wireless keyboard and mouse will work if already paired with Pi. Keyboard layout can be configured through raspberry -config.

3.1.2.4 Power Supply

The Pi is powered by a USB micro power supply like most standard mobile phone chargers. A good-quality power supply is required that can supply at least 700mA at 5V.Power supplies with less than 700 mA current should work for basic usage but the Pi might reboot if too much power is drawn.

3.1.2.5 Ethernet (Network) Cable

An Ethernet cable is necessary to connect Pi to a local network and the internet. It is necessary when built-in WiFi or Wi-Fi module is not available for connection.

3.1.2.6 USB Wireless Dongle

Pi can also be connected to a wireless network through use of a USB wireless dongle which will need to be configured.

3.1.2.7 Audio Lead

A standard 3.5mm audio jack is used to audio through speakers and headphones. An audio is required in the absence of an HDMI cable. If the Pi is connected to the monitor through an HDMI cable no separate audio lead is necessary as audio can be played directly from the display. But if playing audio through speakers is preferable, it will have to be configured.

3.1.3 BCM2837

The Raspberry Pi 3 Model B uses the Broadcom processor BCM2837. The underlying architecture in BCM2837 is identical to BCM2836. It has an ARMv8 CPU.

3.1.4 Power

The device is powered by a 5V micro USB supply. Exactly how much current (mA) the Raspberry Pi requires is dependent on what is connected to it. A 1.2A (1200 mA) power supply from a reputable retailer will provide apple power to run a Pi.

Typically, the model B uses between 700-1000mA depending on what peripherals are connected whereas the model A can use as little as 500mA with no peripherals attached. The maximum power the Raspberry Pi can use is 1 Amp. If someone needs to connect a USB device that will take the power requirements above 1 Amp, then they must connect it to an externally powered USB hub. The power requirements of the Raspberry Pi increase as it makes use of the various interfaces on the Raspberry Pi. The GPIO pins can draw 50mA safely distributed across all the pins. An individual GPIO pin can only safely draw 16mA.

The HDMI port uses 50mA, the camera module requires 250 mA, and the keyboards and mice can take as little as 100mA or over 1000mA. Back powering occurs when USB hubs do not provide a diode to stop the hub from powering against the host computer. Some hubs will back feed the Raspberry Pi. This means that the hubs will power the Raspberry Pi through its USB input cable, without the need for a separate micro-USB power cable, and bypass the voltage protection. If the Raspberry Pi is connected to a hub that back feeds to it and the hub experiences power surge, the Raspberry Pi could potentially be damaged.

3.1.5 GPIO Pins

One powerful feature of the Raspberry Pi is the row of GPIO (General Purpose Input/Output) pins. These pins are a physical interface between the Pi and the outside world. The GPIO pins can be programmed to interact in amazing ways with the real world. Inputs do not have to come from a physical switch. It could be input from a sensor or a signal from another computer or device. The output can also do anything. In addition to the familiar USB, Ethernet and HDMI ports, the Raspberry Pi offers the ability to connect directly to a variety of electronic devices. These include:

- Digital outputs turn lights, motors, or other devices on or off.
- Digital inputs read an on or off state from a button, switch, or other sensor.

3.1.5.1 Communication with Chips or Modules Using Low-Level Protocols

SPI, IC, or serial UART. Connections are made using GPIO (”General Purpose Input/Output”) pins. Unlike USB, etc., these interfaces are not “plug and play” and require care to avoid miswiring.

The Raspberry PI GPIOs use 3.3V logic levels, and can be damaged if connected directly to 5V levels (as found in many older digital systems) without level-conversion circuitry. No analogue input or output is available. However, add-on boards such as the Rpi Gert board provide this capability. The Raspberry Pi Model A+ and B+ boards, and the Pi 3 Model B, have a 40-pin header marked J8, arranged as 2x20 pins. The first 26 pins are the same as P1 on the A/B boards, with the remaining 14 pins providing additional GPIO and ground pins, and an EEPROM ID feature for auto-configuration with add-on “HAT” boards. GPIO voltage levels are 3.3 V and are not 5 V tolerant. There is no over-voltage protection on the board - the intention is that people interested in serious interfacing will use an external board with buffers, level conversion and analog I/O rather than soldering directly onto the main board.

All the GPIO pins can be reconfigured to provide alternate functions, SPI, PWM, IC and so. At reset only pins GPIO 14 15 are assigned to the alternate function UART, these two can be switched back to GPIO to provide a total of 17 GPIO pins. Each of their functions and full details of how to access are detailed in the chipset datasheet. [9]

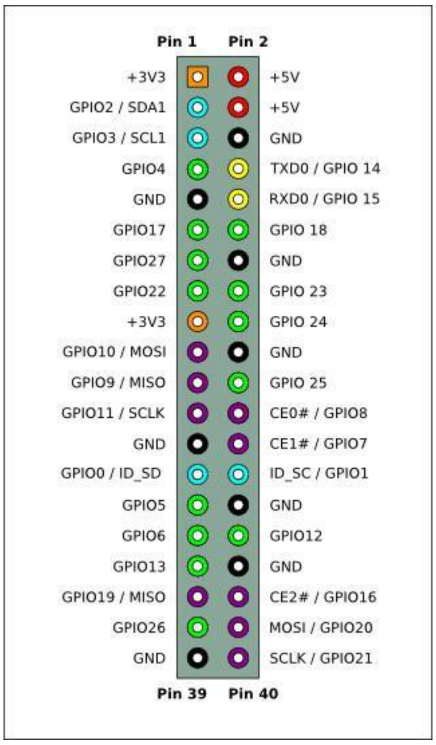


Figure 3.2: Raspberry Pi 3 pin diagram

Each GPIO can interrupt, high/low/rise/fall/change. There is currently no support for GPIO interrupts in the official kernel, however a patch exists, and requiring compilation of modified source tree. The Raspbian “wheezy” version that is currently recommended for starters already includes GPIO interrupts. GPIO input hysteresis (Schmitt trigger) can be on or off, output slew rate can be fast or limited, and source and sink current is configurable from 2 mA up to 16 mA. - GPIO Pads Control. Particular attention should be applied to the note regarding SSO (Simultaneous Switching Outputs): to avoid interference, driving currents should be kept as low as possible. The available alternative functions and their corresponding pins are detailed in the above figyre. These numbers are in reference to the chipset documentation and may not match the numbers exposed in Linux. Only fully usable functions are detailed, for some alternative functions not all the necessary pins are available for the functionality to be actually used. [9]

3.1.6 USB

The Raspberry Pi 3 Model B is equipped with four USB 2.0 ports. These are connected to LAN9512 combo hub/Ethernet chip IC3, which is itself a USB device connected to the single upstream USB port on BCM2837. The USB ports enable the attachment of peripherals such as keyboard, mice, webcams that provide the Pi with additional functionality.

There are some differences between the USB hardware on the Raspberry Pi and the USB hardware on desktop computers or laptop/tablet devices. The USB host port inside the
PI is an On-The-Go (OTG) host as the application processor powering the PI, BCM2836, was originally intended to be used in the mobile market: i.e. as the single USB port on a phone for connection to a PC, or to a single device, In essence, the OTG hardware is simpler than the equivalent hardware on PC. OTG in general supports communication to all types of USB device, but to provide an adequate level of functionality for the most of the USB devices that one might plug into a Pi, the system software has to do more work.

In general, every device supported by Linux is possible to use with the Pi, subject to a few caveats detailed further down. Linux has probably the most comprehensive driver database for legacy hardware of any operating system. The OTG hardware on Raspberry Pi has a similar level of support for certain devices, which may present a higher software processing overhead. The Raspberry Pi also has only one root USB port: all traffic from all connected devices is funneled down this bus, which operates at a maximum speed of 480mbps. The OTG hardware on Raspberry Pi has a simpler level of support for certain devices which may present a higher software processing overhead.

The USB specification defines three device speeds - Low, Full and High. Most mice and keyboards are Low-speed, most USB sound devices are Full-speed and most video devices (webcams or video capture) are High-speed. Generally, there are no issues with connecting multiple High-speed USB devices to a Pi. The software overhead incurred when talking to

Low- and Full-speed devices means that there are soft limitations on the number of simultaneously active Lowand Full-speed devices. Small numbers of these types of devices connected to a Pi will cause no issues. USB devices have defined power requirements, in units of 100mA from100mA to 500mA. The device advertises its own power requirements to the USB host when it is first connected.

In theory, the actual power consumed by the device should not exceed its stated requirement. The USB ports on a Raspberry Pi have a design loading of 100mA each sufficient to drive "low-power" devices such as mice and keyboards. Devices such as WiFi adapters, USB hard drives, USB pen drives all consume much more current and should be powered from an external hub with its own power supply. While it is possible to plug a 500mA device into a Pi and have it work with a sufficiently powerful supply, reliable operation is not guaranteed. In addition, hot plugging high-power devices into the Pi's USB ports may cause a brownout which can cause the Pi to reset. [23]

3.1.7 Piezo Buzzer

Piezo buzzers are used for making beeps, tones and alerts. [10]

Figure 3.3: Piezo buzzer

This one is petite but loud! Drive it with 3-30V peak-to-peak square wave. To use, connect one pin to ground (either one) and the other pin to a square wave out from a timer or micro controller. For the loudest tones, stay around 4 KHz, but works quite well from 2 KHz to 10KHz. For extra loudness, you can connect both pins to a micro controller and swap which pin is high or low (differential drive) for double the volume.

3.1.7.1 Applications

- Judging panels
- Educational purposes
- Annunciator panels
- Electronic metronomes
- Game show lock-out device
- Microwave ovens and other household appliances
- Sporting events such as basketball games
- Electrical alarms
- Klaxon [10]

3.1.8 LED

3.1.8.1 Definition

A **light-emitting diode** (**LED**) is a two-lead semiconductor light source. It is a p–n junction diode that emits light when activated. When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor. LEDs are typically small (less than 1 mm^2) and integrated optical components may be used to shape the radiation pattern. [11]

Figure 3.4: LED (Light Emitting Diode).

3.1.8.2 Efficiency and Operational Parameters

Typical indicator LEDs are designed to operate with no more than 30–60 milliwatts (mW) of electrical power. Around 1999, Philips Lumileds introduced power LEDs capable of continuous use at one watt. These LEDs used much larger semiconductor die sizes to handle the large power inputs. Also, the semiconductor dies were mounted onto metal slugs to allow for heat removal from the LED die.

One of the key advantages of LED-based lighting sources is high luminous efficacy. White LEDs quickly matched and overtook the efficacy of standard incandescent lighting systems. In 2002, Lumileds made five-watt LEDs available with luminous efficacy of 18–22 lumens per watt (lm/W). For comparison, a conventional incandescent light bulb of 60–100 watts emits around 15 lm/W, and standard fluorescent lights emit up to 100 lm/W.

As of 2012, Philips had achieved the following efficacies for each color. The efficiency values show the physics – light power out per electrical power in. The lumen-per-watt efficacy value includes characteristics of the human eye and is derived using the luminosity function. [13]

Table 3.1: Efficacy and efficiency of different colors of LEDs.

Color	Wavelength Range (nm)	Typical Efficiency Coefficient	Typical Efficacy(lm/W)
Red	$620 < \lambda < 645$	0.39	72
Red-orange	$610 < \lambda < 620$	0.29	98
Green	$520 < \lambda < 550$	0.15	93
Cyan	$490 < \lambda < 520$	0.26	75
Blue	$460 < \lambda < 490$	0.35	37

In September 2003, a new type of blue LED was demonstrated by Cree that consumes 24 mW at 20 milliamperes (mA). This produced a commercially packaged white light giving 65 lm/W at 20 mA, becoming the brightest white LED commercially available at the time, and more than four times as efficient as standard incandescents. In 2006, they demonstrated a prototype with a record white LED luminous efficacy of 131 lm/W at 20 mA. Nichia Corporation has developed a white LED with luminous efficacy of 150 lm/W at a forward current of 20 mA. Cree's XLamp XM-L LEDs, commercially available in 2011, produce 100 lm/W at their full power of 10 W, and up to 160 lm/W at around 2 W input power. In 2012, Cree announced a white LED giving 254 lm/W, and 303 lm/W in March 2014. Practical general lighting needs high-power LEDs, of one watt or more. Typical operating currents for such devices begin at 350 mA.

These efficiencies are for the light-emitting diode only, held at low temperature in a lab. Since LEDs installed in real fixtures operate at higher temperature and with driver losses, real-world efficiencies are much lower. United States Department of Energy (DOE) testing of commercial LED lamps designed to replace incandescent lamps or CFLs showed that average efficacy was still about 46 lm/W in 2009 where red color has 72 lm/W efficacy, green has 93 lm/W and so on as given in table 3.1 (tested performance ranged from 17 lm/W to 79 lm/W). [12]

3.1.8.3 Advantages

- **Efficiency:** LEDs emit more lumens per watt than incandescent light bulbs. The efficiency of LED lighting fixtures is not affected by shape and size, unlike fluorescent light bulbs or tubes.
- **Color:** LEDs can emit light of an intended color without using any color filters as traditional lighting methods need. This is more efficient and can lower initial costs.
- **Size:** LEDs can be very small (smaller than 2 mm^2) and are easily attached to printed circuit boards.
- **Warmup time:** LEDs light up very quickly. A typical red indicator LED achieves full brightness in under a microsecond. LEDs used in communications devices can have even faster response times.
- **Cycling:** LEDs are ideal for uses subject to frequent on-off cycling, unlike incandescent and fluorescent lamps that fail faster when cycled often, or high-intensity discharge lamps (HID lamps) that require a long time before restarting.
- **Dimming:** LEDs can very easily be dimmed either by pulse-width modulation or lowering the forward current. This pulse-width modulation is why LED lights, particularly headlights on cars, when viewed on camera or by some people, appear to be flashing or flickering. This is a type of stroboscopic effect.
- **Cool light:** In contrast to most light sources, LEDs radiate very little heat in the form of IR that can cause damage to sensitive objects or fabrics. Wasted energy is dispersed as heat through the base of the LED.

- **Slow failure:** LEDs mostly fail by dimming over time, rather than the abrupt failure of incandescent bulbs.
- **Lifetime:** LEDs can have a relatively long useful life. One report estimates 35,000 to 50,000 hours of useful life, though time to complete failure may be longer. Fluorescent tubes typically are rated at about 10,000 to 15,000 hours, depending partly on the conditions of use, and incandescent light bulbs at 1,000 to 2,000 hours. Several DOE demonstrations have shown that reduced maintenance costs from this extended lifetime, rather than energy savings, is the primary factor in determining the payback period for an LED product.
- **Shock resistance:** LEDs, being solid-state components, are difficult to damage with external shock, unlike fluorescent and incandescent bulbs, which are fragile.
- **Focus:** The solid package of the LED can be designed to focus its light. Incandescent and fluorescent sources often require an external reflector to collect light and direct it in a usable manner. For larger LED packages total internal reflection (TIR) lenses are often used to the same effect. However, when large quantities of light are needed many light sources are usually deployed, which are difficult to focus or collimate towards the same target. [13]

3.1.8.4 Disadvantages

- **Initial price:** LEDs are currently slightly more expensive (price per lumen) on an initial capital cost basis, than other lighting technologies. As of March 2014, at least one manufacturer claims to have reached $1 per kilolumen. The additional expense partially stems from the relatively low lumen output and the drive circuitry and power supplies needed.
- **Temperature dependence:** LED performance largely depends on the ambient temperature of the operating environment – or thermal management properties. Overdriving an LED in high ambient temperatures may result in overheating the LED package, eventually leading to device failure. An adequate heat sink is needed to maintain long life. This is especially important in automotive, medical, and military uses where devices must operate over a wide range of temperatures, which require low failure rates. Toshiba has produced LEDs with an operating temperature range of −40 to 100 °C, which

suits the LEDs for both indoor and outdoor use in applications such as lamps, ceiling lighting, street lights, and floodlights.

- **Voltage sensitivity:** LEDs must be supplied with a voltage above their threshold voltage and a current below their rating. Current and lifetime change greatly with a small change in applied voltage. They thus require a current-regulated supply (usually just a series resistor for indicator LEDs).
- **Color rendition:** Most cool-white LEDs have spectra that differ significantly from a black body radiator like the sun or an incandescent light. The spike at 460 nm and dip at 500 nm can cause the color of objects to be perceived differently under cool-white LED illumination than sunlight or incandescent sources, due to metamerism red surfaces being rendered particularly poorly by typical phosphor-based cool-white LEDs.
- **Area light source:** Single LEDs do not approximate a point source of light giving a spherical light distribution, but rather a lambertian distribution. So LEDs are difficult to apply to uses needing a spherical light field; however, different fields of light can be manipulated by the application of different optics or "lenses". LEDs cannot provide divergence below a few degrees. In contrast, lasers can emit beams with divergences of 0.2 degrees or less.
- **Electrical polarity:** Unlike incandescent light bulbs, which illuminate regardless of the electrical polarity, LEDs only light with correct electrical polarity. To automatically match source polarity to LED devices, rectifiers can be used.
- **Blue hazard:** There is a concern that blue LEDs and cool-white LEDs are now capable of exceeding safe limits of the so-called blue-light hazard as defined in eye safety specifications such as ANSI/IESNA RP-27.1–05: Recommended Practice for photo biological safety for Lamp and Lamp Systems.
- **Light pollution:** Because white LEDs, especially those with high color temperature, emit much more short wavelength light than conventional outdoor light sources such as high-pressure sodium vapor lamps, the increased blue and green sensitivity of scotopic vision means that white LEDs used in outdoor lighting cause substantially more sky glow. The American Medical Association warned on the use of high blue content white LEDs in street lighting, due to their higher impact on human health and environment, compared to low blue content light sources (e.g. High-Pressure Sodium, PC amber LEDs, and low CCT LEDs).

- **Efficiency droop:** The efficiency of LEDs decreases as the electric current increases. Heating also increases with higher currents, which compromises LED lifetime. These effects put practical limits on the current through an LED in high power applications.
- **Impact on insects:** LEDs are much more attractive to insects than sodium-vapor lights, so much so that there has been speculative concern about the possibility of disruption to food webs.
- **Use in winter conditions:** Since they do not give off much heat in comparison to incandescent lights, LED lights used for traffic control can have snow obscuring them, leading to accidents. [13]

3.1.8.5 Applications

LED uses fall into four major categories:

- Visual signals where light goes more or less directly from the source to the human eye, to convey a message or meaning
- Illumination where light is reflected from objects to give visual response of these objects
- Measuring and interacting with processes involving no human vision.
- Narrow band light sensors where LEDs operate in a reverse-bias mode and respond to incident light, instead of emitting light. [14]

3.1.9 MANHATTAN 703307 Case/Power Supply Fan

MANHATTAN Case/Power Supply Fans are equipped with high-quality sleeve or ball bearings to help deliver efficient, reliable cooling and quiet performance. Easily installed and offered in range of sizes, MANHATTAN Case/Power Supply Fans help circulate cool air through the case to dissipate heat away from processors and components to reduce heat damage. [15]

Figure 3.5: MANHATTAN 703307 Case/Power Supply Fan

3.1.9.1 Features

- Three-pin power connector for easy installation
- Ideal for new systems, upgrades and replacements
- Ball bearing delivers quiet, reliable operation
- Rated voltage: 12 V DC
- Input current: 0.25 A (maximum)
- Air flow: 66 cfm (maximum)
- Fan speed: 1,800 rpm
- Noise level: 30 dBA (maximum)
- Dimensions: 120 x 120 x 25 mm

3.1.10 DC motor

3.1.10.1 Definition

A DC motor is an electric motor that runs on direct current power. In any electric motor, operation is dependent upon simple electromagnetism. A current carrying conductor generates a magnetic field, when this is then placed in an external magnetic field, it will encounter a force proportional to the current in the conductor and to the strength of the external magnetic field.It is a device which converts electrical energy to mechanical energy.

Figure 3.6: DC motor

It works on the fact that a current carrying conductor placed in a magnetic field experiences a force which causes it to rotate with respect to its original position.

3.1.10.2 Types of DC motor

There are 4 main types of DC motors:

3.1.10.2.1 Permanent Magnet DC Motors

The permanent magnet motor uses a permanent magnet to create field flux. This type of DC motor provides great starting torque and has good speed regulation, but torque is limited so they are typically found on low horsepower applications.

3.1.10.2.2 Series DC Motors

In a series DC motor, the field is wound with a few turns of a large wire carrying the full armature current. Typically, series DC motors create a large amount of starting torque, but cannot regulate speed and can even be damaged by running with no load. These limitations mean that they are not a good option for variable speed drive applications.

3.1.10.2.3 Shunt DC Motors

In shunt DC motors the field is connected in parallel (shunt) with the armature windings. These motors offer great speed regulation due to the fact that the shunt field can be excited separately from the armature windings, which also offers simplified reversing controls.

3.1.10.2.4 Compound DC Motors

Compound DC motors, like shunt DC motors, have a separately excited shunt field. Compound DC motors have good starting torque but may experience control problems in variable speed drive applications. [16]

3.1.10.3 Advantages

- Provide excellent speed control for acceleration and deceleration
- Easy to understand design
- Simple, cheap drive design

3.1.10.4 Disadvantages

- Speed regulation in the series motor is quite poor. With the increase in the load speed of the machine decreases.
- If speed decreases torque will decrease.
- DC series motor should always require to be loaded before starting the motor, hence not suitable for all type of applications.

3.1.10.5 Applications

- Electric traction
- Cranes
- Elevators
- Air compressor
- Vacuum cleaner
- Hair drier
- Sewing, Spinning and weaving machines
- Lathes
- Drills
- Boring mills
- Shapers
- PressesShears
- Reciprocating machine[16]

3.2 Raspberry Pi 3 Model B Operating System

There are different ways to install operating system into raspberry pi. The main two systems are discussed below:

3.2.1 Raspbian

Raspbian jessie is the default operating system for regular use on Raspberry Pi. Raspbian jessie is a free operating system based on Debian and optimized for Raspberry Pi hardware. Raspbian jessie comes with over 35000 packages; precompiled software bundled in a nice format for easy installation on Raspberry Pi. Raspbian jessie is a community project under active development, with the emphasis on developing stability and performance of as many Debian packages as possible [17].

3.2.2 Android Things

Android Things (codenamed Brillo) is an Android-based embedded operating system platform by Google, announced at Google I/O 2015. It is aimed to be used with lowpower and memory constrained Internet of Things (IoT) devices, which are usually built from different MCU platforms. [18]

3.3 Network Server & Software

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 CM43438 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. [19]

3.3.1 Access Over Internet

Raspberry Pi can be connected to another computer or a mobile device. One method is to set-up port forwarding. To set-up port forwarding one must change the configuration of their router to forward all inbound traffic from the Internet to a specific port to the local IP address of their

Raspberry Pi. One disadvantage of doing this is that it exposes a network port on a private LAN to the public network. This is security vulnerability and must be managed carefully. One secure alternative to port forwarding is the Weaved service. Weaved is software that needs to be install on the Raspberry Pi and it will allow Pi to connect to any device over the Internet.

3.3.2 VNC

VNC is a graphical desktop sharing system that allows someone to remotely control the desktop interface of one computer from another. It transmits the keyboard and mouse events form the controller, and receive updates to the screen over the network from the remote host. This enables a user to use the desktop of the Pi inside a window on their computer. They will be able to control it as though they were working on Raspberry Pi itself.

VNC is used widely across every industry sector by individuals and organizations for different use cases which include providing IT desktop support to colleagues and friends and also accessing system and services on the move.

3.3.3 SSH (Secure Shell)

Secure Shell, or SSH is a network protocol which operates at Application layer of OSI model to allow encrypted remote login and other services for secure operations over an unsecured network. SSH provides a secure channel in an unsecure network through a client-server architecture. SSH is capable of securing any network service but typical applications include remote command-line login and remote command execution. The command line of Raspberry Pi can be accessed from another computer in the same network using SSH. SSH server is enabled on Raspberry Pi by default. It can be disabled as well. SSH is built into Linux distributions and Mac OS, and a third-party SSH client is available for Windows.

3.4 Programming & Scripting Languages

3.4.1 XML

In addition to Java code, Android projects (and their developers) have the ability to utilize XML to perform many standard tasks. Some XML usage is required, such as the definition of the projects and its components. Much of XML's usage is optional, making many common tasks easier. The Android XML schema is highly flexible and may be used in combination with code, exclusively, or not at all. Below is a list of common usage of XML:

- **Manifest** - Definition of the project.
- **Layout** - Creation of partial or complete layouts for Activities, Dialogs, and Widgets.
- **Colors** - Constant color values used throughout the project.
- **Style & Themes** - Application of custom standardized looks of Views.
- **Animations** - Standardized animations that may be applied to Views.
- **Drawables** - Some specialized icons and graphics that may not be created wholly with an image editor. (StateDrawables, TranstionDrawables, Shapes, and Vector- Graphics).
- **Menu** - A resource used to aid in standardized menus for an Activity Integers, Strings, and Arrays - Constants used by the Application as resources.

3.4.2 Material Design

Material design is a comprehensive guide for visual, motion, and interaction design across platforms and devices. Material Design makes more liberal use of grid-based layouts, responsive animations and transitions, padding, and depth effects such as lighting and shadows. Google announced Material Design on June 25, 2014, at the 2014 Google I/O conference. Android 5.0 Lollipop includes support for material design application. Polymer and Angular Material projects also provide official implementations.

3.4.3 Java

Java is a general-purpose computer programming language that is concurrent, class based, object oriented and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers “write once, run anywhere” (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems’ Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses.

As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (byte code compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plug in for applets). The latest version is Java 8, which is the only version currently supported for free by Oracle, although earlier versions are supported both by Oracle and other companies on a commercial basis. [20]

3.5 Software Tools

The softwar tools we used in this project are as follows

3.5.1 Android Studio IDE

Android Studio was on May 16, 2013. Google's product manager Ellie Powers on the developer conference Google I/O announced. Shortly after this time, Google periodically new test versions. After a development time of two years, Google published on 8 December 2014Version1.0 for Windows, OS X and Linux. Since the alpha version 1.2 Preview 1, which was published on 10 March 2015 is based on Android Studio IntelliJ IDEA 14.

With the preview version 1.3 of 28 May 2015 the SDK Manager has been fully integrated into Android Studio, further is now support for the Android NDK (Native Development Kit) available. In the final version1.3 also the Android Memory Viewer and Allocation Tracker was integrated. From this version, it is also possible to in line annotations for the new application permissions of Android M Data Binding to use as well feature is Instant Run available, which allows developers to modify the amended code and Since the Android Studio preview version 2.0, the resources directly on the device within the current app. [21]

3.5.2 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 7 (previously 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) Plug-in, though IntelliJ IDEA IDE (all editions) fully supports Android out of the box and NetBeans IDE also supports Android development via a plug-in. 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 download able 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 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.

3.5.3 Gradle

Gradle is an open source build automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the XML form used by Apache Maven of declaring the project configuration.
Gradle uses a directed acyclic graph (DAG) to determine the order in which tasks can be run. Gradle was designed for multi-project builds which can grow to be quite large and supports incremental builds by intelligently determining which parts of the build tree are up-to-date, so that any task dependent upon those parts will not need to be re-executed.
The initial plug-in are primarily focused around Java, Groovy and Scale development and deployment, but more languages and project work flows are on the road map.

3.5.4 FFmpeg

FFmpeg is a free software project that produces libraries and programs for handling multimedia data. FFmpeg includes libavcodec, an audio/video codec library used by several other projects, libavformat (Lavf), an audio/video container mux and demux library, and the ffmpeg command line program for transcoding multimedia files. FFmpeg is published under the GNU Lesser General Public License 2.1+ or GNU General Public License 2+ (depending on which options are enabled). [22]

3.6 Server

We use server as via for communication between the app and raspberry Pi that interacts with Pi and app through a script file.

3.6.1 Apache Server

The Apache HTTP server is a software (or program) that runs in the background under an appropriate operating system, which supports multi-tasking, and provides services to other applications that connect to it, such as client web browsers. It was first developed to work with Linux/Unix operating systems, but was later adapted to work under other systems, including Windows and Mac. The Apache binary running under UNIX is called HTTPd(short for HTTP daemon), and under win32 is called Apache.exe.

Installing Apache on Linux does require a bit of programming skills (though it is not too difficult). Installing it on a Windows platform is straight forward, as you can run it through a graphical user interface.

Apache's original core is fairly basic and contains a limited number of features. Its power rather comes from added functionality introduced through many modules that are written by programmers and can be installed to extend the server's capabilities. To add a new module, all you need to do is install it and restart the Apache server. Functionality that you don't need or want can easily be removed which is actually considered a good practice as it keeps the server small and light, starts faster, consumes less system resources and memory, and makes the server less prone to security holes. The Apache server also supports third party modules, some of which have been added to Apache 2 as permanent features. The Apache server very easily integrates with other open source applications, such as PHP and MySQL, making it even more powerful than it already is.

A web server in its simplest form is a computer with special software, and an internet connection that allows it to connect to other devices. Every device connected to a network has an IP address through which others connect to and communicate with it. This IP address is sort of like a regular address that you need in real life to call or visit any contact of yours. If they didn't have an address, you wouldn't know how to call or reach them. IP addresses serve the exact same purpose. If a device didn't have one, the other machines on the same network wouldn't know how to reach it.

The Apache server offers a number of services that clients might make use of. These services are offered using various protocols through different ports, and include: hypertext transfer protocol (HTTP), typically through port 80, simple mail transfer protocol (SMTP), typically through port 25, domain name service (DNS) for mapping domain names to their corresponding IP addresses, genearlly through port 53, and file transfer protocol (FTP) for uploading and downloading files, usually through port 21. [25]

3.6.1.1 How Does It Work?

Apache's main role is all about communication over networks, and it uses the TCP/IP protocol (Transmission Control Protocol/Internet Protocol which allows devices with IP addresses within the same network to communicate with one another).

The TCP/IP protocol is a set of rules that define how clients make requests and how servers respond, and determine how data is transmitted, delivered, received, and acknowledged.

The Apache server is set up to run through configuration files, in which directives are added to control its behavior. In its idle state, Apache listens to the IP addresses identified in its config file (HTTPd.conf). Whenever it receives a request, it analyzes the headers, applies the rules specified for it in the Config file, and takes action.

But one server can host many websites, not just one - though, to the outside world, they seem separate from one another. To achieve this, every one of those websites has to be assigned a different name, even if those all map eventually to the same machine. This is accomplished by using what is known as virtual hosts.

Since IP addresses are difficult to remember, we, as visitors to specific sites, usually type in their respective domain names into the URL address box on our browsers. The browser then connects to a DNS server, which translates the domain names to their IP addresses. The browser then takes the returned IP address and connects to it. The browser also sends a Host header with the request so that, if the server is hosting multiple sites, it will know which one to serve back.

HTTP is a request / response stateless protocol. It's a set of rules that govern communication between a client and the server. The client (usually but not necessarily a web browser) makes a request, the server sends back a response, and communication stops. The server doesn't look forward for more communication as is the case with other protocols that stay at a waiting state after the request is over.

If the request is successful, the server returns a 200 status code (which means that the page is found), response headers, along with the requested data. [25]

4. System Design

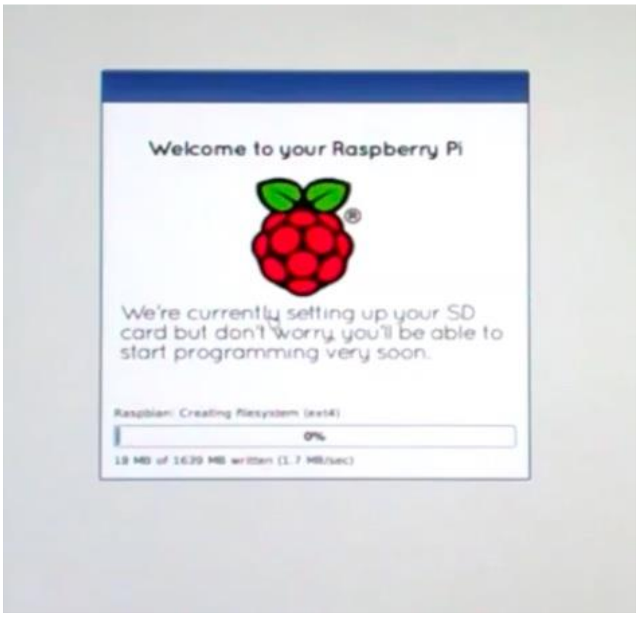
This System design is consist of two part. One of the part Hardware Design and another one is Software Design.

4.1 Hardware Design & Development:

Hardware design and development of the project is shown below

4.1.1 Raspbian -Jessie Installing on Raspberry pi 3

Raspbian-Jessie Software is an easy operating system install manager for the Raspberry Pi. The most convenient way to get Raspbian-Jessie is to purchase a micro SD card with Raspbian-Jessie pre-installed. It can also be downloaded from the Raspberry Pi website. After downloading the Raspbian-Jessie zip file, the contents of the file will have to be copied to a formatted SD card on a computer.



Welcome to your Raspberry Pi

We're currently setting up your SD card but don't worry you'll be able to start programming very soon.

Raspbian: Creating filesystem (ext4)

0%

18 MB of 1639 MB written (1.7 MB/sec)

Figure 4.1: Raspbian-Jessie installation

4.1.1.1 Installing Raspbian-Jessie on an SD Card

Format an SD that is 4 GB or larger as FAT. Download and extract files from the Raspbian-Jessie zip file. Copy the extracted file onto the SD card immediately after formatting so that the files are in the root directory of the

SD card. The files might be copied into a single folder. In that case, the files inside the folder must be copied across rather than the folder itself. The RECOVERY fat partition will be automatically resized to a minimum and a list of available to install OS will be displayed.

4.1.1.1.1 Windows

The SD Associations Formatting Tool is recommended for Windows users which can be downloaded from sdcard.org. It is necessary to set FORMAT SIZE ADJUSTMENT option ON in the Options menu so that the entire SD card volume is formatted instead of a single partition.

4.1.1.1.2 Mac OS

The SD Associations Formatting Tool is also available for Mac users. The default OSX Disc utility can also be used to format the disc. In order to do that, the SD card has to be selected and choose Erase with MS-DOS format.

4.1.1.1.3 Linux

For Linux users parted (or the command line version parted) is recommended.
Included in Raspbian-Jessie. Only Raspbian is installed in Raspbian-Jessie by default.
The others can be installed with a network connection. Raspbian-Jessie and Raspbian-Jessie Lite.

4.1.2 Installing Apache:

Apache is a popular web server application that can install on the Raspberry Pi to allow it to serve web pages. On its own, Apache can serve HTML files over HTTP, and with additional modules can serve dynamic web pages using scripting languages such as PHP.

4.1.2.1 Install Apache:

First install the apache2 package by typing the following command in to the Terminal:

```
sudo apt-get install apache2 –y
```

4.1.2.2 Test The Web Server:

By default, Apache puts a test HTML file in the web folder. This default web page is served when you browse to http://localhost/ on the Pi itself, or http://192.168.1.10 (whatever the Pi's IP address is) from another computer on the network. To find the Pi's IP address, type hostname -I at the command line (or read more about finding the IP address). Browse to the default web page either on the Pi or from another computer on the network.

4.1.2.3 Changing the Default Web Page:

This default web page is just a HTML file on the filesystem. It is located at

/var/www/html/index.html.

Navigate to this directory in a terminal window and have a look at what's inside:

```
cd /var/www/html

ls -al
```

This will show:

```
total 12

drwxr-xr-x 2 root root 4096 Jan 8 01:29 .

drwxr-xr-x 12 root root 4096 Jan 8 01:28 ..

-rw-r--r-- 1 root root 177 Jan 8 01:29 index.html
```

This shows that by default there is one file in /var/www/html/ called index.html and it is owned by the root user (as is the enclosing folder). In order to edit the file, you need to change its ownership to your own username. Change the owner of the file (the default pi user is assumed here) using

```
sudo chown pi: index.html.
```

Now it is ready to try editing this file and then refreshing the browser to see the web page change.

ADDITIONAL - INSTALL PHP

To allow Apache server to process PHP files, there needs to install PHP5 and the PHP5 module for Apache. The following command to install these:

```
sudo apt-get install php5 libapache2-mod-php5 -y
```

Now remove the index.html file:

```
sudo rm index.html
```

and create the file index.php:

```
sudo leafpad index.php
```

Note: Leafpad is a graphical editor. Alternatively, user can use nano if the user is restricted to the command line

Put some PHP content in it:

```
<?php echo "hello world"; ?>
```

Now save and refresh browser. User should see "hello world". This is not dynamic but still served by PHP. Try something dynamic:

```
<?php echo date('Y-m-d H:i:s'); ?>
```

or show PHP info:

```
<?php phpinfo(); ?>
```

4.1.3 Device Interfacing:

The Smart Home System consists of one Raspberry pi 3 model B, four LED, a DC motor,a fan, a Piezo Buzzer, a Router, an Android application and Android phone.

Raspberry pi connected with router via wifi to Internet and pass information to the server. When android application connect to the local network the server passes the information about the the current state of LEDs, buzzer and fan.

4.2 Software Design & Development

4.2.1 Program Flow

The system architecture is as shown in Figure 3. When the mobile is connected to WiFi the user can control electronics devices and other usage services such as Smart light and fan on/off system, doorbell notification through the mobile App.

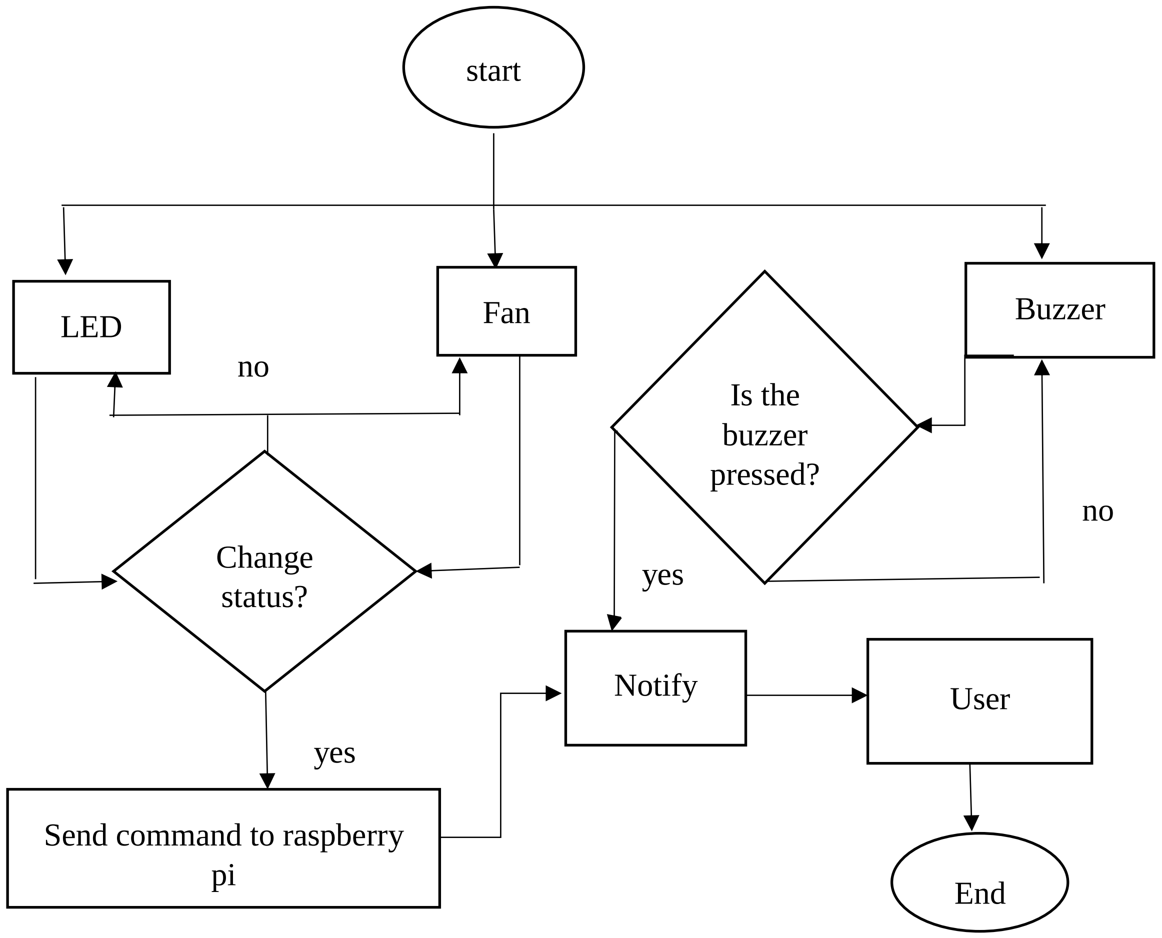


Figure 4.2: Program flow chart of the system.

5 Implementation:

5.1 Frame:

At first we measured a space for our entire project that how much will space it to take.

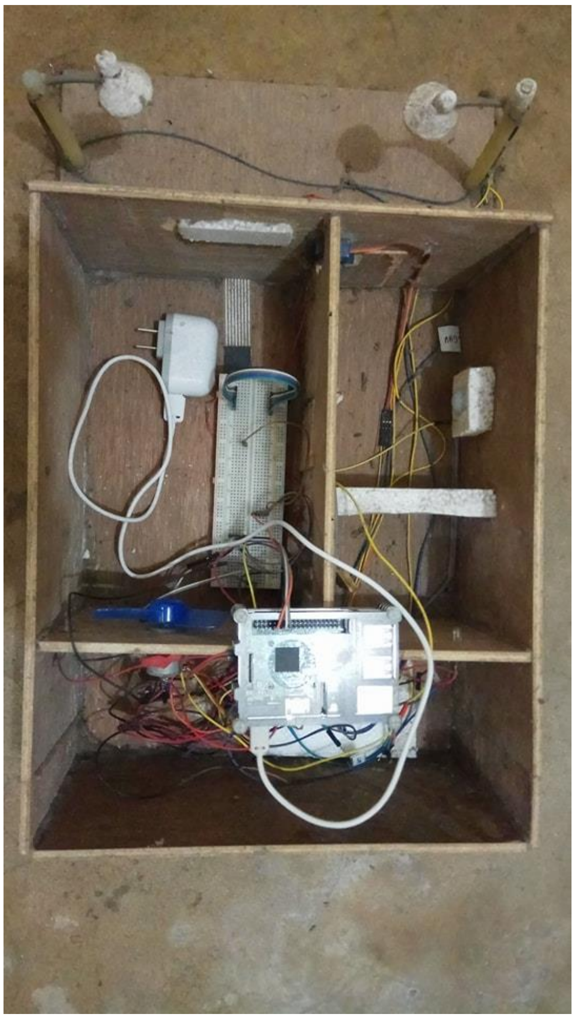
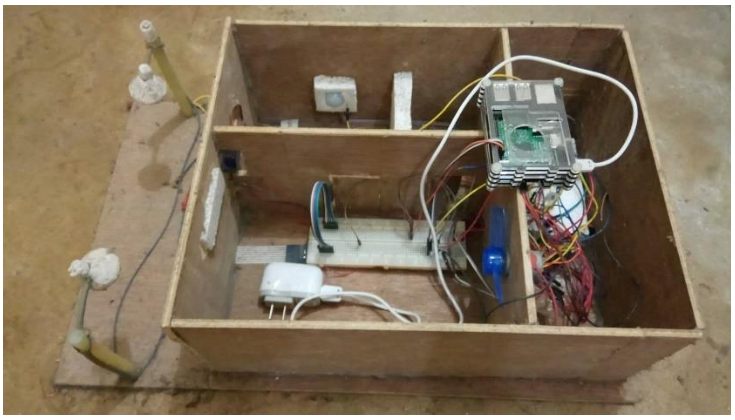



Figure 5.1: Home Design

We shape the house and design it. Then we put fan, LED lights, raspberry pi, and buzzer in this demo home.

5.2 Android Application:

The application name is eco Home. The application consists of some pages. The flash page is as follows:

Figure 5.2: The splash page.

From the home page the user can choose the room for further activities.

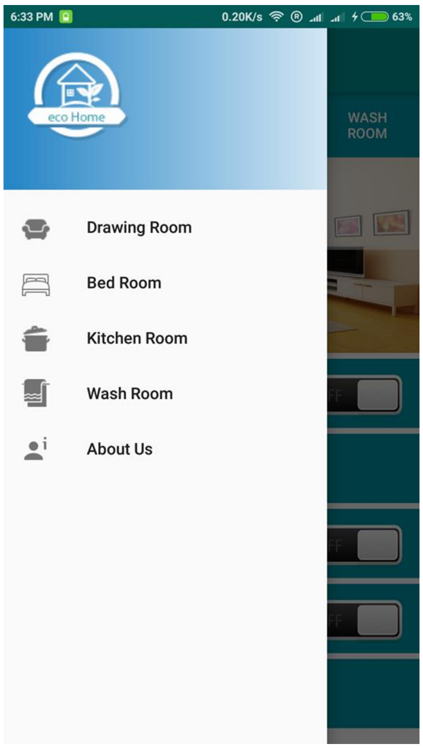


Figure 5.3: The home page.

The application have four different rooms and user can select any of those from the Home page.

- Living room
- Bedroom
- Kitchen
- Washroom

Whenever the user chooses any specific room the user can do the following tasks for each room remotely:

- Lights on/ off
- Fan on/ off
- Press the buzzer

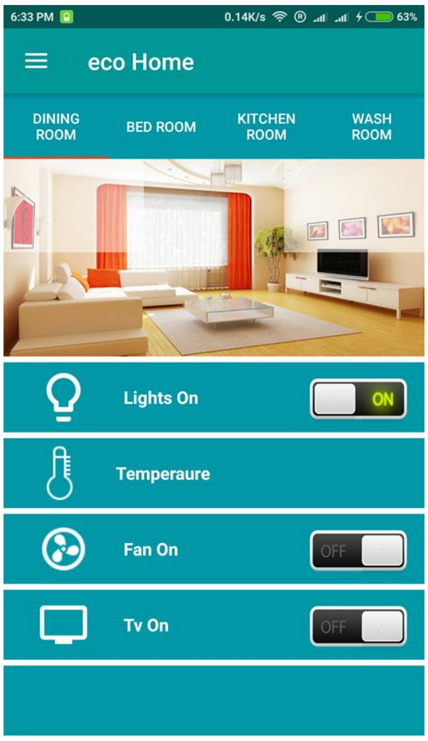


Figure 5.4: Living room

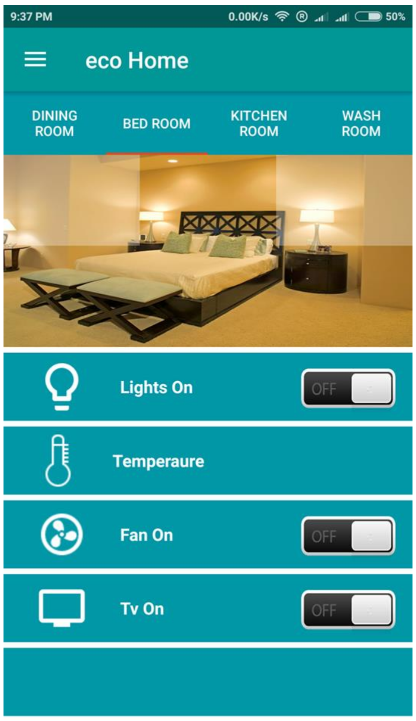


Figure 5.5: Bedroom

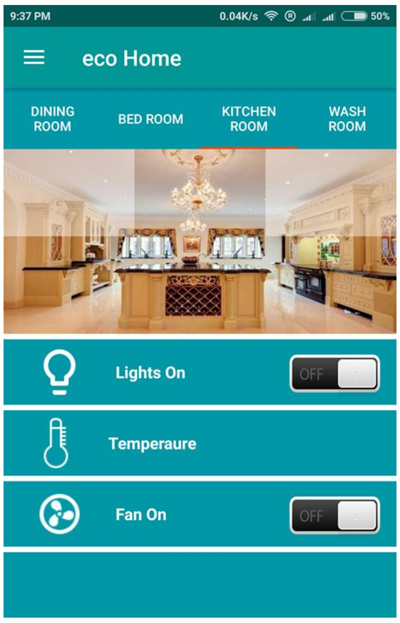


Figure 5.6: Kitchen

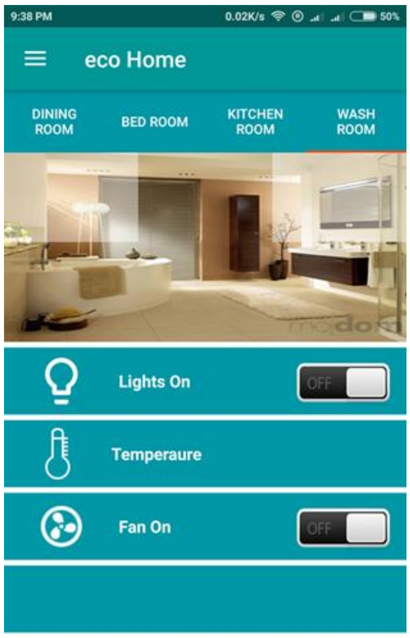


Figure 5.7: Washroom.

When a user gives any command from android app, then the pi checks the state of switch which is written in a server containing the script file over a local network. It is noted that the app and pi have to be in the same network or under the same router as we used a server over a local network.

6. Conclusion

Day by day IoT (Internet of Things) is taking the control of our daily life. Now-a-days, every device is going to connect with Internet so that every device can connect to each other. The demand of home automation system is also increasing day by day. But in abroad, these kind of devices are way too expensive. Adobe Home Security Starter Kit costs $299.00 which finds the sweet spot between a basic self-monitored DIY security system and a professionally installed and monitored solution. The Nest Learning Thermostat costs $249.60 which has built-in Wi-Fi so the user can remotely control the temperature from phone, tablet, or PC. That is why we have taken this project so that we can give people of our country in a cheap range. Our project costs only 4000 BDT. We have finished our project which includes controlling lights, fans and buzzer. We will add many more functionalities in this project and will provide in cheap range.

6.1 Limitations

In this project, there are some limitations.

- There is no security or safety alarms in this project.
- We have included buzzer but we could not open the door remotely.
- There is no sensor work in this project.

6.2 Future Works

We keep the scalability to the project so that we can include other functionalities in near future. Such as-

- Security will be the first concern in our project.
- Face Recognition System makes a system more secure. We will add it in our project.
- Fingerprint door lock system will add to make the system safe.

- By adding Video Streaming we can ensure the security of a home from far away. We also will add it.
- We will add Cloud Vision API to our System.

References:

[1] “The Internet of Things - essential Internet of Things business guide,” i-SCOOP. [Online]. Available: https://www.i-scoop.eu/internet-of-things-guide/. [Accessed: 11-Oct-2017].

[2] S. Kashyap, G. Blog, F. Shaikh, and A. Gupta, “10 Real World Applications of Internet of Things (IoT) - Explained in Videos,” *Analytics Vidhya,* 25-Aug-2016. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/08/10-youtube-videos-explaining-the-real-world-applications-of-internet-of-things-iot/. [Accessed: 10-Oct-2017].

[3] What Is Home Automation and How Does it Work? [Online]. Available: https://www.safewise.com/home-security-faq/how-does-home-automation-work. [Accessed: 11-Oct-2017].

[4] N. U. Mushtaq, “Smart Home CCTV Surveillance Smart-homes Home Automation,” CCTV Institute, 29-Apr-2017. [Online]. Available: http://cctvinstitute.co.uk/smart-home/. [Accessed: 11-Oct-2017].

[5] “Meet the Nest Cam Indoor security camera,” Nest. [Online]. Available: https://nest.com/ca/camera/meet-nest-cam/. [Accessed: 11-Oct-2017].

[6] R. Crist and D. Carnoy, “Amazon Echo: The speaker of the house,” CNET, 19-Jul-2017. [Online]. Available: https://www.cnet.com/au/products/amazon-echo-review/. [Accessed: 11-Oct-2017].

[7] R. Sharma, “Mark Zuckerberg's Jarvis AI: What It Is, What It Can Do, and More,” NDTV Gadgets360.com, 22-Dec-2016. [Online]. Available: http://gadgets.ndtv.com/apps/features/mark-zuckerbergs-jarvis-ai-what-it-is-what-it-can-do-and-more-1640658. [Accessed: 11-Oct-2017].

[8] R. Pi, “Raspberry pi,” Raspberry Pi, vol. 1, p. 1, 2013.

[9] raspberrypi.org, “An introduction to gpio and physical computing on raspberry pi,”

[Online]. Available: https://www.raspberrypi.org/documentation/usage/gpio/.

[10] ——, “Piezo buzzer,” [Online]. Available: https://www.adafruit.com/product/160.

[11] “What is light-emitting diode (LED)? - Definition from WhatIs.com,” WhatIs.com. [Online]. Available: http://whatis.techtarget.com/definition/light-emitting-diode-LED. [Accessed: 11-Oct-2017].

[12] “How Light Emitting Diodes Work,” How Light Emitting Diodes Work | HowStuffWorks, 31-Jan-2002. [Online]. Available: http://electronics.howstuffworks.com/led.htm. [Accessed: 11-Oct-2017].

[13] “Nichia Unveils White LED with 150 lm/W Luminous Efficiency,” Nikkei Technology Online. [Online]. Available: http://techon.nikkeibp.co.jp/english/NEWS_EN/20061221/125713/. [Accessed: 11-Oct-2017].

[14] W. D. C. A. F. C. B. C. C. A. Y. Q. Q. E.-mail:yuzhiguo1985@163.com Website:http://www.yuzhiguo.com, “Advantages and Disadvantages of LED,” LEDKE Technology Co., Ltd. [Online]. Available: http://www.ledke.com/news/Advantages-Disadvantages-LED.html. [Accessed: 13-Oct-2017].

[15] “Case/Power Supply Fan,” Manhattan Products - Case/Power Supply Fan (703307). [Online]. Available: http://www.manhattan-products.com/en-us/products/7859. [Accessed: 13-Oct-2017].

[16] “Electric DC Motors - Direct Current Motor Basics,Types and Application,” ElProCus - Electronic Projects for Engineering Students, 28-Mar-2016. [Online]. Available: https://www.elprocus.com/dc-motor-basics-types-application/. [Accessed: 11-Oct-2017].

[17] raspberrypi.org, “Raspbian jessie with desktop,” [Online]. Available: https://www.raspberrypi.org/downloads/raspbian/.

[18] android.com, “Android things,” [Online]. Available: https://developer.android.com/things/preview/download.html.

[19] seemoo-lab, “seemoo-lab/nexmon,” GitHub, 05-Oct-2017. [Online]. Available: https://github.com/seemoo-lab/nexmon. [Accessed: 11-Oct-2017].

[20] Tutorialspoint, “Java (programming language),” [Online]. Available:

https://www.tutorialspoint.com/java/java overview.htm, Accessed on 2017-01-01.

[21] R. Rogers, J. Lombardo, Z. Mednieks, and B. Meike, *Android application development:*

Programming with the Google SDK. O'Reilly Media, Inc., 2009.

[22] "FFmpeg Resource | Learn About, Share and Discuss FFmpeg at like2do.com," like2do.com. [Online]. Available: http://www.like2do.com/learn?s=FFmpeg. [Accessed: 13-Oct-2017].

[23] "USB." USB - Raspberry Pi Documentation, www.raspberrypi.org/documentation/hardware/raspberrypi/usb/README.md.

A Appendix

A.1 Logic & Code:

For this system we use Raspberry pi 3 model b as a micro controller. We implemented our logic and code in Raspberry pi 3. So all the code are given below.

A.1.1 pHp Code:

We use pHp code for Apache server. Those codes are as follows:

```
<?php
exec("gpio mode 0 out");
exec("gpio mode 2 out");
exec("gpio mode 3 out");
if (isset($_GET['led1'])) {
        if($_GET['led1'] == 1) {
                exec("gpio write 0 1");// pin 0 in wiring pi is gpio 17
        } else {
                exec("gpio write 0 0");
        }
}
if(isset($_GET['led2'])) {
        if($_GET['led2'] == 1) {
                exec("gpio write 2 1");// pin 2 in wiring pi is gpio 27
        } else {
                exec("gpio write 2 0");
        }
}
if(isset($_GET['led3'])) {
        if($_GET['led3'] == 1) {
                exec("gpio write 3 1");// pin 3 in wiring pi is gpio 22
        } else {
                exec("gpio write 3 0");
        }
}
```

?>

A1.2 Mobile Application Design Code:

Key code of mobile application design code as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.shakil.demohome">
<uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/eco_logo"
        android:label="eco Home"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category

   ..................................................
..................................................................................
   ..................................................

        </activity>
    </application>
</manifest>

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.shakil.demohome.About">


    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

     .......................................................
........................................................................................
     .......................................................

    </android.support.design.widget.AppBarLayout>

    <LinearLayout
        android:layout_marginLeft="10dp"
        android:orientation="vertical"
        android:layout_marginRight="10dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    <TextView
        android:textSize="16dp"
        android:layout_marginTop="15dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:textSize="16dp"
        android:layout_marginTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    </LinearLayout>
</LinearLayout>
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:app=http://schemas.android.com/apk/res-auto

   ………………………………………
………………………………………………………………
   ………………………………………

    <include
        layout="@layout/app_bar_home"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_home"
        app:menu="@menu/activity_home_drawer" />

</android.support.v4.widget.DrawerLayout>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background"
    android:orientation="horizontal"
    tools:context="com.example.shakil.demohome.MainActivity">

   ………………………………………
………………………………………………………………
   ………………………………………

<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.shakil.demohome.HomeActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_home" />


</android.support.design.widget.CoordinatorLayout>



<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app=http://schemas.android.com/apk/res-auto

   ……………………………………………
……………………………………………………………………………
   ……………………………………………

    <android.support.design.widget.TabLayout
        android:id="@+id/tabLayout"
        app:tabGravity="fill"
        app:tabMode="fixed"
        android:background="#0097a7"
        app:tabTextColor="@android:color/white"
        app:tabSelectedTextColor="@android:color/white"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize">


    </android.support.design.widget.TabLayout>


        <android.support.v4.view.ViewPager
            android:id="@+id/viewPager"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

        </android.support.v4.view.ViewPager>


</LinearLayout>
```

   ....

....

   ....

```
   <FrameLayout
       android:id="@+id/secondFrame"
       android:layout_width="match_parent"
       android:layout_height="match_parent">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"

    <TextView
        android:layout_marginTop="40dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
```

```
</FrameLayout>
</RelativeLayout>

   ……………………………………………
……………………………………………………………………………
   ……………………………………………

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="@drawable/bed">

        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="80dp"
            android:layout_weight="1"
            android:background="#50FFFFFF"></LinearLayout>
        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_weight="1"></LinearLayout>
        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="80dp"
            android:layout_weight="1"
            android:background="#50FFFFFF"></LinearLayout>
    </LinearLayout>

   ……………………………………………
……………………………………………………………………………
   ……………………………………………

                android:layout_height="match_parent">
                <LinearLayout
                    android:orientation="horizontal"
                    android:layout_width="match_parent"
                    android:layout_height="60dp"
                    android:background="#0097a7">

                    <LinearLayout
                        android:layout_width="0dp"
                        android:layout_height="match_parent"
                        android:layout_weight="1"
                        android:background="#0097a7">
                        <ImageView
                            android:id="@+id/lightBulb"
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:src="@drawable/ic_lightbulb"
                            android:padding="5dp"/>
                    </LinearLayout>
                    <LinearLayout
                        android:layout_width="0dp"
```

……………………………………………

…………………………………………………………………………

……………………………………………

```
android:layout_gravity="center_vertical"/>
                    </LinearLayout>
                    <LinearLayout
                        android:layout_width="0dp"
                        android:layout_height="match_parent"
                        android:layout_weight="1"
                        android:padding="5dp"
                        android:background="#0097a7">

                        <ToggleButton
                            android:id="@+id/toggleButtonLightBed"
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:layout_gravity="center_vertical"
                            android:layout_marginRight="10dp"
                            android:layout_marginTop="6dp"

android:background="@drawable/toggle_button"
                            android:text="ToggleButton"
                            android:textOff=""
                            android:textOn="" />
                        </LinearLayout>
                    </LinearLayout>
                    <LinearLayout
                        android:orientation="horizontal"
                        android:layout_width="match_parent"
                        android:layout_height="60dp"
                        android:layout_marginTop="5dp"
```

   ……………………………………………

……………………………………………………………………

   ……………………………………………

```
                            android:layout_gravity="center"
                            android:padding="7dp"/>
                        </LinearLayout>
                        <LinearLayout
                            android:layout_width="0dp"
                            android:layout_height="match_parent"
                            android:layout_weight="1.75">
                            <TextView
                                android:id="@+id/textViewTemperature"
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:text="Temperaure"
                                android:textSize="15dp"
                                android:textColor="@android:color/white"
                                android:textAllCaps="false"
                                android:textStyle="bold"
```

```
android:layout_gravity="center_vertical"/>
                    </LinearLayout>
                    <LinearLayout
                        android:layout_width="0dp"
                        android:layout_height="match_parent"
                        android:layout_weight="1"
                        android:background="#0097a7">

                    </LinearLayout>
                </LinearLayout>
                <LinearLayout
                    android:orientation="horizontal"
```

   ....

....

   ....

```
                        android:background="#0097a7"
                        android:weightSum="1">
                        <ImageView
                            android:id="@+id/imageViewFan"
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:src="@drawable/fan"
                            android:padding="8dp"

android:layout_gravity="center_horizontal"
                            android:layout_weight="1" />
                    </LinearLayout>
                    <LinearLayout
                        android:layout_width="0dp"
                        android:layout_height="match_parent"
                        android:layout_weight="1.50">
                        <TextView
                            android:id="@+id/textViewFan"
                            android:layout_width="wrap_content"
```

```
                                    android:layout_height="wrap_content"
                                    android:text="Fan On"
   ……………………………………………
…………………………………………………………………………
   ……………………………………………
                                android:background="#0097a7">

                                <ToggleButton
                                    android:id="@+id/toggleButtonFanBed"
                                    android:layout_width="match_parent"
                                    android:layout_height="wrap_content"
                                    android:layout_gravity="center_vertical"
                                    android:layout_marginRight="10dp"
                                    android:layout_marginTop="6dp"

android:background="@drawable/toggle_button"
                                    android:text="ToggleButton"
                                    android:textOff=""
                                    android:textOn="" />
                            </LinearLayout>
                        </LinearLayout>
                        <LinearLayout


   ……………………………………………
…………………………………………………………………………
   ……………………………………………


                                android:background="#0097a7">

                                <ImageView
                                    android:id="@+id/imageViewTv"
                                    android:layout_width="wrap_content"
                                    android:layout_height="wrap_content"
                                    android:layout_gravity="center_vertical"
                                    android:padding="8dp"
                                    android:src="@drawable/ic_tv" />
                            </LinearLayout>
```

```
                    <LinearLayout
                        android:layout_width="0dp"
                        android:layout_height="match_parent"
                        android:layout_weight="1.50">
                        <TextView

   ……………………………………………
…………………………………………………………………………
   ……………………………………………

android:layout_gravity="center_vertical"/>
                    </LinearLayout>
                    <LinearLayout
                        android:layout_width="0dp"
                        android:layout_height="match_parent"


android:background="@drawable/toggle_button"
                            android:textOn=""
                            android:textOff=""
                            android:layout_marginRight="10dp"
                            android:layout_marginTop="6dp"

   ……………………………………………
…………………………………………………………………………
   ……………………………………………

            </LinearLayout>
        </ScrollView>
    </LinearLayout>
</LinearLayout>
                <LinearLayout


   ……………………………………………
…………………………………………………………………………
   ……………………………………………
```

```
                        <ImageView
                            android:id="@+id/imageViewTv"
                            android:layout_width="wrap_content"
                            android:layout_height="wrap_content"
                            android:layout_gravity="center_vertical"
                            android:padding="8dp"
                            android:src="@drawable/ic_tv" />
                    </LinearLayout>
                    <LinearLayout


   ……………………………………………
……………………………………………………………………………
   ……………………………………………


android:layout_gravity="center_vertical"/>
                    </LinearLayout>
                    <LinearLayout
                        android:layout_width="0dp"
                        android:layout_height="match_parent"
                        android:layout_weight="1"
                        android:padding="5dp"
                        android:background="#0097a7">
                        <ToggleButton
                            android:id="@+id/toggleButtonTvDining"
                            android:layout_width="match_parent"
                            android:layout_height="wrap_content"
                            android:text="ToggleButton"

android:background="@drawable/toggle_button"
                            android:textOn=""
                            android:textOff=""
                            android:layout_marginRight="10dp"
                            android:layout_marginTop="6dp"

android:layout_gravity="center_vertical"/>
                    </LinearLayout>
```

```
                </LinearLayout>

   ……………………………………………
……………………………………………………………………
   ……………………………………………

        </ScrollView>
    </LinearLayout>
</LinearLayout>



<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"

   ……………………………………………
……………………………………………………………………
   ……………………………………………

    <ImageView
        android:id="@+id/imageView"
        android:layout_marginBottom="25dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/header_logo" />

</LinearLayout>
```