Sudan University of Science and Technology

College of Engineering

School of Electronics Engineering

Smart Home: Home Monitoring & control System using Simple Mobile App

المنزل الذكي: نظام التحكم والمراقبة للمنزل باستخدام تطبيق في الهاتف

A Research Submitted in Partial fulfillment for the Requirements of The Degree of Bachelor of Engineering (Electronics -Telecommunication)”

Prepared By:

1. **Ahmed Hamid Hassan**
2. **Al-fatih Yassin Awad**
3. **Ayman Hashim El-Amin**

Supervised By:

Dr. Rania Abdel-Hameed

September, 2015

بسم الله الرحمن الرحيم

استهلال

قال تعالى:

(وَلَا تَيْأَسُوا مِن رَّوْحِ اللَّهِ ۖ إِنَّهُ لَا يَيْأَسُ مِن رَّوْحِ اللَّهِ إِلَّا الْقَوْمُ الْكَافِرُونَ)

صدق الله العظيم

سورة يوسف

الآية (87)

DEDICATION

To those who helped us to get here, we couldn't do it without you!

Families

To those who prayed for us & reminded us always of the greatest God!

Parents & Beloved ones

To those who have been patience with us thought all the way!

Guiders

To those who put a smile on our faces whenever we got tired!

Friends

To ourselves & the entire human race

We dedicate this work.

AKNOWLEDGMENT

Before all we prayed grace to our god (Allah) for giving us the gift of life & all the abilities that helped us completing this work.

We extend our appreciations and gratitude to our guiders:-

Dr. Rania Abdelhameed

Eng. Mujahed Abdalrahman

for their continuous support, patience, motivation, and massive knowledge.

Lastly a lot thanks are extended to all those who directly or indirectly, have participated on this work. Thank you all.

Abstract

Smart Home: Home control & Monitoring System using Simple Mobile Application shows how to automate your lights, curtains, music, and more. It controls everything via a mobile phone from doors to curtains, including monitoring the home itself.

This thesis deals with the design and implementation of Secure Home Automation using Raspberry Pi an mobile phone to provide essential security to our homes and associated control operations.

The proposed home security solution hinges on novel integration of motion detectors. Raspberry Pi operates and controls motion detectors and remote sensing and surveillance, and finally manages operations on home appliances, such as turning ON/OFF a television or microwave.

المستخلص

المنزل الذكي : نظام التحكم والمراقبة للمنزل باستخدام تطبيق في الهاتف يوضح كيفية أتمتة الأضواء، والستائر، والموسيقى، وأكثر من ذلك. يمكن التحكم في كل شيء عبر الهاتف المحمول من أصغر الأشياء كالأبواب و الستائر حتى مراقبة المنزل نفسه.

هذه الأطروحة تقدم تصميم وتنفذ أتمته المنزل باستخدام الراسبري باي والهواتف الذكية لتوفير الأمن الضروري للمنازل وعمليات المراقبة المرتبطة بها بالإضافة لعمليات التحكم بالاجهزة عن بعد .

الحل المطروح لأتمته وتأمين المنزل الذكي يعتمد على كواشف الحركة، الراسبري باي يقوم بإدارة كواشف الحركة والاستشعار عن بعد والمراقبة، وبالتالي يمكنه في النهاية التحكم بالأجهزة المنزلية مثل فتح\غلق التلفزيون أو الميكروويف.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IoTs	Internet Of Things
TV	Television
AC	Alternating Current
ECHO	Electronic Computing Home Operator
PC	Personal computer
WiFi	Wireless Fidelity
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol
OS	operating system
WPANs	wireless personal area networks
RF	radio frequency
ISM	industrial, scientific and medical radio
HVAC	Heating ventilating and Air conditioning
RISC	Reduced instruction Set Computer
ARM	Advanced RISC Machines
CPU	Central Processing Unit
USB	Universal Serial Bus
RCA	Radio Corporation Of America
SD	Secure Digital
DVD	Digital versatile Disk
PIC	(Peripheral Interface Controller
GSM	Global system For Mobile
SMS	Short Message Service
TCP	Transmission Control Protocol

GPU	Graphic Processor Unit
SDRAM	Synchronous Dynamic Random Access Memory
UMTS	Universal Mobil Telecommunications Service
HD	High Definition
HDMI	High Definition Multimedia Interface
HTML	Hyper Text Markup Language
GUI	Graphical User Interface
E-board	embedded system board
GPIO	General Purpose Input/output
LED	Light Emitting Diode
PIR	Passive Infrared Sensor
IC	Integrated Circuit
SPST	Single pole single throw
NC	Normally Closed
NO	Normally Open
LDR	Light Dependent Resistor
Rest API	Application Program Interface
COAP	Constrained Application Protocol
ADC	Analog to digital converter
DAC	Digital to Analog Converter

CHAPTER ONE

INTRODUCTION

1.1 Introduction

The Internet of Things (IoTs) can be described as connecting everyday objects like smart-phones, Internet TVs, sensors and appliances to the Internet where the devices are intelligently linked together enabling new forms of communication between things and people, and between things themselves [1].

IoTs technology can also be applied to create a new concept and wide development space for smart homes to improve the quality of life.

A typical smart home allows homeowners to control household devices from remote Smartphone.

Homeowners by using simple app in Smartphone can set their home to adjust the thermostat, open and close blinds, lock and unlock doors and even send notification to alert them when anyone enter to home.

From lighting and climate control to home security and entertainment, homeowners can now control almost every aspect of their home with the touch of a button.

1.2 Problem statement

When people left their homes, number of problems may exist, this include:

- Saving power from wasting electricity when forgetting electrical appliances open. Also saving effort and comfort for handicaps.
- Monitoring the appliances for protection from causing fires or Electric stuns.

- Monitoring the entries of the house to provide safeness from thieves or any interlopers.

The main problem is the lack of a user's ability to control home appliances more efficiently from anywhere.

1.3 Proposed Solution

The proposed solution is to provide remote control and monitoring system for home appliances and environment. The system uses Raspberry pi embedded system with internet connectivity to communicate with Smartphone, the raspberry pi connected to household devices to perform control and monitor functions.

1.4 Aim and Objectives

The aim of this project is to develop prototype for a Smart Home System that will enable homeowners to control and monitor electric devices such as lamps, TV and other devices by using Smartphone.

Objectives:

- To design smart home based on Raspberry Pi that will run as central hub in home.
- To provide the communications mechanism between Smartphone and Raspberry pi via internet.
- To develop software for Raspberry Pi, that will enable to control and monitor household devices from Smartphone application.

1.5 Methodology

Connecting the electronic appliances to Raspberry Pi. The raspberry Pi will connect to a Smartphone by internet connection. In the internet, there will be an intermediate server between raspberry Pi and the Smartphone that will allow the exchange of control commands and the responses for it. The software for raspberry pi will be written in the C/C++ language to be able to control and monitor devices and to interact with application on Smartphone.

1.6 Thesis outline

There are five chapters in this thesis.

The Chapter 1: Will discusses briefly the overview of this project such as introduction, problem statement, proposed solution, aim &objectives, methodology and thesis outlines.

Chapter 2: contains the literature review that will be discussed about what research project was done by other people.

Chapter 3: contains methodology that will be discussed more on the project design includes the programming, the software and the hardware.

Chapter 4: contains results and discussion obtained according to the objectives of this project.

Chapter 5: contains explanations about the conclusion and recommendation that can make for future improvements.

CHAPTER TWO

LITERATURE REVIEW

2.1 Background

Concepts for home and building automation were around for decades before becoming reality and featured in the writing of the 19th century sci-fi author HG Wells, comics, and cartoons such as the Jetsons. American industrialist George Westinghouse helped to pioneer the AC (Alternating Current) electrical system – which the X10 smart home standard would later run over – and in 1966, the company that bears his name, Westinghouse Electric, employed an engineer who developed what could arguably be called the first computerized home automation system – the ECHO IV.

The Electronic Computing Home Operator (ECHO) was featured in the April 1968 edition of Popular Mechanics and had been expanded from a set of spare electronics both in the physical and literal sense, to include computing its founder Jim Sutherland's family household finances and storing their shopping lists, amongst an array of other tasks.

The ECHO never went commercial and through the '60s, hobbyists and a number of large companies such as Honeywell toyed with the idea of computerizing the home, however it was the '70s, much as with personal computing, that saw the birth of the modern era of home automation technology.

2.1.1 X10 – a standard

The beginning of modern home automation technology can be argued to be found with the introduction of the X10 technology standard. Conceived in 1975 by Pico Electronics, who later partnered with Birmingham Sound Reproducers, X10 laid out the framework for allowing remote control access of domestic appliances. The X10 standard was designed to allow transmitters and receivers to work over existing electrical wiring systems by broadcasting messages such as "turn off" and "turn on" via radio frequency bursts.

Three years later in 1978, X10 products began to make their way into stores geared towards electronics enthusiasts and shortly after, in the '80s, the CP-290 computer interface made its way into the market for the Mattel Aquarius computer.

The CP-290 unit allowed computers to communicate with X10 compatible appliances in the home. Over the years, support for Windows and Mac has been included, and gave those interested in home automation the ability to program their lighting systems, thermostats, and garage doors from their home computers.

As revolutionary as X10 has been, it unfortunately had a number of flaws. These included:

- Wiring and interference issues
- Commands getting lost in transmission
- Limited scope of products supporting X10
- Limited scope of commands available
- Slow speed of signal transmission
- Lack of encryption
- Lack of confirmation message without expensive two way devices

By the late '90s, home automation still had not penetrated the home market on a truly wide scale, however the technological advancements of the dot-com boom were providing a whole new set of tools, protocols, and standards that addressed many of the flaws that the X10 standard has been limited by.

2.1.2 The dot.com boom and open source

With the explosion of technologies that followed the birth of the web in the '90s, home computing and networking technologies were now available to the public and could easily and cheaply be installed at home. These technologies would later provide ideal candidates for pushing the boundaries of what could be achieved by home automation enthusiast, and provide the industry with the tools for building smart home appliances and systems.

It was only a small step from *PC to PC communication to appliance to PC communication.*

Home networks running on Ethernet and later WiFi provided a mechanism that could allow computers and electronic appliances to communicate with one another across a home without needing to use the existing electrical wiring. In the case of WiFi, no extra cabling was required.

As protocols such as FTP and HTTP became the norm for accessing information across the Internet, hardware developers saw the opportunity to leverage these communications technologies in open source hardware devices. Where as X10 appliances had no way of knowing if a signal had been successfully sent without the purchase of costly "two-way" devices, web technologies provide a whole framework for returning error codes and messages.

At approximately the same time as the Arduino platform we introduced earlier was being developed, the first tablet computers were beginning to be released. From 2005 until now, there has been an explosion in mobile, tablet, and Smartphone devices. This growth has been commonly referenced to as the "post-PC" era.

These devices have provided mobile computing platforms that can run complex software and be small enough to fit in the user's pocket. As a result of this, applications have been developed for the iPhone and Android that allow the user to control consumer electronics such as the TV.

Due to their size, portability, and in some cases, low cost, they have provided the perfect platform for interfacing with home appliances and devices, and provided an extension to a medium the user is familiar with.

Alongside the explosion in hardware, there was also an equivalent explosion in software. One particular product of interest that we will look at is the open source Android operating system.

The Android OS is a Linux-based operating system geared towards mobile devices. As part of the Open Handset Alliance – a consortium of 84 companies operating in the mobile sphere, Google backed and eventually purchased the Android mobile operating system.

The aim has been to create an open source operating system that can compete with companies such as Apple, and provide a robust system that can work across multiple manufacturer devices.

As a result of this, commercial manufacturers of home appliances have begun to embed the technology and software into their products, and a generation of "smart-devices" has started to appear in stores around the world.

2.1.3 ZigBee standard

ZigBee is a specification for a suite of high level communication protocols using small, low-power digital radios based on the IEEE 802.15.4-2003 standard for wireless personal area networks (WPANs), such as wireless headphones connecting with cell phones via short-range radio. The technology defined by the ZigBee specification is intended to be simpler and less expensive than other WPANs such as Bluetooth.

ZigBee is targeted at radio frequency (RF) applications that require a low data rate, long battery life, and secure networking.

ZigBee is a low-cost, low-power, wireless mesh networking standard. The low cost allows the technology to be widely deployed in wireless control and monitoring applications, the low power-usage allows longer life with smaller batteries, and the mesh networking provides high reliability and larger range.

ZigBee operates in the industrial, scientific and medical (ISM) radio bands; 868 MHz in Europe, 915 MHz in the USA and Australia, and 2.4 GHz in most jurisdictions worldwide. ZigBee chip vendors typically sell integrated radios and microcontrollers with between 60K and 128K flash memory, such as the Free scale MC13213, the Ember EM250 and the Texas Instruments CC2430. Radios are also available as stand-alone to be used with any processor or microcontroller. Generally, the chip vendors also offer the ZigBee software stack, although independent ones are also available. The ZigBee Alliance is a group of companies that maintain and publish the ZigBee standard.

- **Typical areas of application of ZigBee are**:
- **Home Entertainment and Control** — Smart lighting, advanced temperature control, safety and security, movies and music.
- **Home Awareness** — Water sensors, power sensors, smoke and fire detectors, smart appliances and access sensors.
- **Mobile Services** — m-payment, m-monitoring and control-security and access control, m-healthcare and tele-assist.
- **Commercial Building** — Energy monitoring, HVAC, lighting, access control.
- **Industrial Plant** — Process control, asset management, environmental management, energy management, industrial device control.

2.1.4 Arrival of the Raspberry Pi

A group of computer scientists led by Eben Upton at the University of Cambridge's Computer Laboratory in 2006 struck upon the idea of producing a cheap educational micro-computer geared towards the amateur computer enthusiast, budding students, and children. The aim was to help to provide the skills to future Computer Science undergraduate applicants that many of those applying in the '90s possessed, thanks to the home computers of the '80s.

However it would be another two years before the project became viable, and not until 2012 before the Raspberry Pi was being shipped out to the public.

The 2000s saw a huge growth in mobile computing technologies, a large segment of this being driven by the mobile phone industry. By 2005, ARM – a British manufacturer of CPU core components and a by-product of the '80s home computer company Acorn, had grown to where 98 percent of mobile phones were using their technology. This translated into around 1 billion CPU cores. ARM technology would later end up being featured on the Raspberry Pi with the ARM ARM1176JZF-S processor core being used.

During the same period, Ebon Upton designed several concepts for the Raspberry Pi and by 2008, thanks to a by-product of the increasing penetration of mobile phone technology, the cost of building a miniature, portable microcomputer with many of the multimedia functions that the public were accustomed to becoming viable.

Thus the Raspberry Pi foundation was formed and set about the task of developing and manufacturing the Raspberry Pi computer.

By 2011, the first Alpha models were being produced and tested, and the public finally got to see what the Raspberry Pi was capable of.

Finally in 2012, the Raspberry Pi was ready for public consumption. Two versions of the Raspberry Pi were scheduled to be manufactured, namely models A and B, with B being released first.

The model A board which will not include an Ethernet port and will consume considerably less power than the model B was given a price tag of $25.

The model B that includes an Ethernet port was given a target price of $35 USD and manufacturing in China started. This would later be moved to the UK with Sony taking over the process.

- **Raspberry Pi hardware specifications**

The Raspberry Pi is built off the back of the Broadcom BCM2835. The BCM2835 is a multimedia application processor geared towards mobile and embedded devices.

On top of this, several other components have been included to support USB, RCA, and SD card storage.

The following figure highlights some of these with a description of each provided:

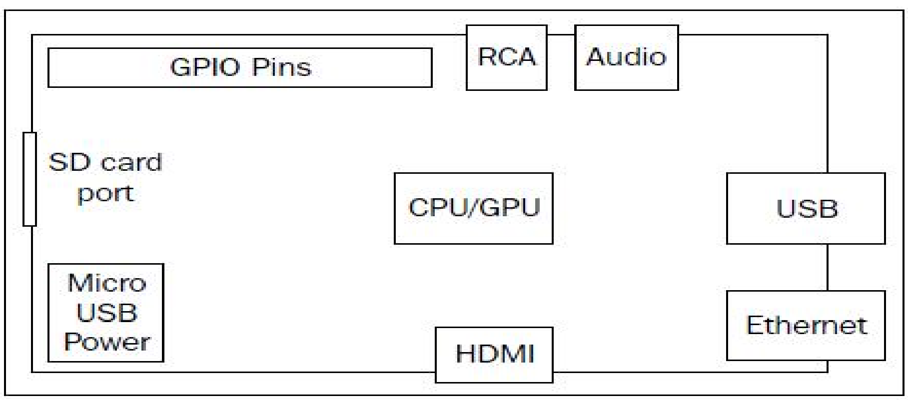

Figure 2.1 Raspberry Pi architecture

- **Dimensions**

The Raspberry Pi is a small device coming in at 85.60mm x 53.98mm x 17mm and weighing only 45g. This makes it perfect for home automation, where a small device can be placed in a case and mounted inside an electrical box, or replace an existing thermostat device on a wall.

- **3.5mm analog audio jack**

The 3.5mm analog audio jack allows you to connect headphones and speakers to the Raspberry Pi. This is especially useful for audio and media player based projects.

- **Composite RCA port**

You are probably familiar with the composite cables used to hook up your DVD player to the TV. They usually come in the red, white, and yellow plug variety. The Raspberry Pi has a port for attaching the yellow video cable from your TV to it, allowing you to use your TV as a monitor.

- **Two USB 2.0 ports plus one micro USB**

USB is one of the most common methods for connecting peripherals and storage devices to a computer. The Raspberry Pi comes equipped with two of them, allowing you to hook up a keyboard and mouse when you get started and a micro USB port for powering your device.

- **High Definition Multi-media Interface port**

The High Definition Multi-media Interface (HDMI) port allows the Raspberry Pi to be hooked up to high-definition televisions and monitors that support the technology. This provides an additional option to the composite RCA port for video and additionally supports audio.

Should you wish to stream video and audio from the web to your TV, this is the port you would want to use.

- **SD card port**

The main storage mechanism of the Raspberry Pi is via the SD card port. The SD card will be where we install our operating system and will act as our basic hard disk. Of course, this storage can be expanded upon using the USB ports.

- **256 MB/512 MB SDRAM shared with GPU**

The Raspberry Pi comes equipped with 256 MB of SDRAM on older versions of the model B and 512 MB on the newer revisions. This isn't a huge amount, and much less than you would expect on a PC, where RAM is available in gigabytes. However, for the type of applications we will be building, 256 MB or 512 MB of RAM will be more than enough.

- **central processor unit (CPU)**

Early in this chapter we touched upon ARM – the British manufacturers of central processor unit (CPU) cores. The Raspberry Pi comes equipped with a 700 MHz, ARM1176JZF-S core – part of the ARM 11 32-bit multi-processor core family.

The CPU is the main component of the Raspberry Pi, responsible for carrying out the instructions of a computer program via mathematical and logical operations.

The Raspberry Pi is in good company using the ARM 11 series and has joined the ranks of the iPhone, Amazon Kindle, and Samsung Galaxy.

- **graphics-processing unit (GPU)**

The graphics-processing unit (GPU) is a specialized chip designed to speed up the manipulation of image calculations.

In the case of our Raspberry Pi, it comes equipped with a Broadcom Video Core IV capable of hardware accelerated playback and support for OpenGL.

This is especially useful if you want to run games or video via your Raspberry Pi, or work on 3D graphics in an open source application such as Blender.

- **Ethernet port**

The Ethernet port is the Raspberry Pi's main gateway to communicating with other devices and the Internet. You will be able to use the Ethernet port to plug your Raspberry

Pi into a home router such as the one you currently use to access the Internet, or a network switch if you have one set up.

- **General Purpose Input/output (GPIO) pins**

The General Purpose Input/output (GPIO) pins on the Raspberry Pi are the main way of connecting with other electronic boards such as the Arduino.

As the name suggests, the GPIO pins can accept input and output commands and thus can be programmed on the Raspberry Pi.

With the arrival of the Raspberry Pi , a set of open source technologies now exist that combine the power of the PC, the communication and multimedia technologies of the web, the ability to interact with the environment of a microcontroller, and the portability of a mobile device.

This provides the perfect set of factors allowing us to build cheap devices for our homes that can interface with commercial devices, but can be tailored for our own needs while also providing a great tool for learning about technology.

2.2 Related Works

2.2.1 Home Automation Using GSM

Home Automation Using GSM system presents a novel, stand alone, low-cost and flexible GSM- ZigBee based home automation system. The entire system depends on a 8 bit microcontroller named PIC (Peripheral Interface Controller) in this work.

The Database equipment built around this Microcontroller and a GSM controller facilitate the heart of the system. This device is connected to a ZigBee Transceiver and it communicates with each and every node present inside our home. The GSM Controller facilitate for the data follow between user and microcontroller. The GSM Controller uses mobile phone technology to communicate. From the mobile phone, command can be send via SMS to the Controller, which in turn interprets the command and then activates the required 'switch' to control the electrical item.

As long as there is GSM mobile phone signal coverage, it is possible to control all electrical items from anywhere in the world. The system is easy to operate, and is secure in that only pre-determined mobile numbers can operate the GSM Controller. The installation of the GSM Controller is relatively simple and can be adapted for any existing home system. Control of lights and geyser are done via the electrical distribution board (circuit breakers) [3] .The block diagram of system is as follows:

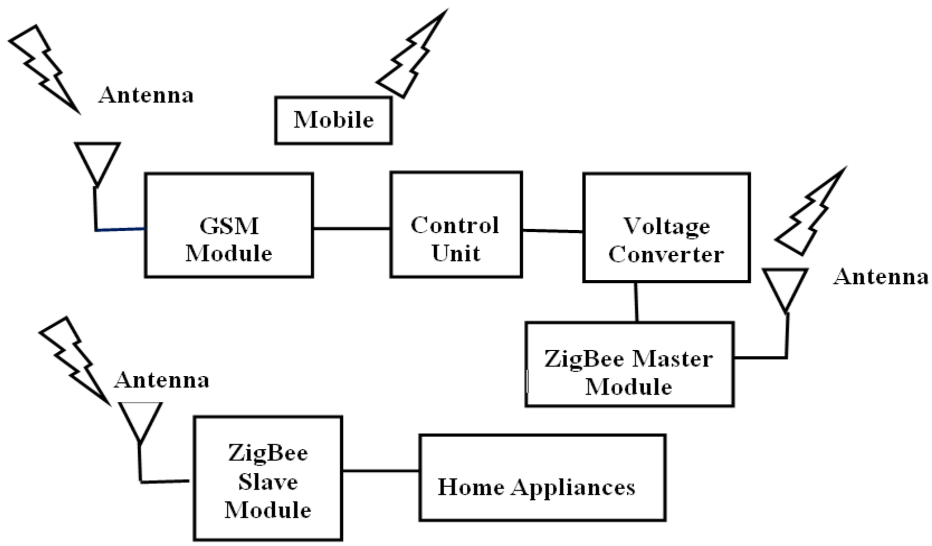

Figure 2.2 Architecture of GSM Home Automation system.

2.2.2 Enabling mobile devices for home automation using ZigBee

Home automation systems are collections of interconnected devices for controlling various functions within a house, such as light control, heating, air conditioning, etc. Mobile devices are ideal in providing a user interface in a home automation system, due to their portability and their wide range of capabilities.

They can communicate with a home automation network through an Internet gateway, but cannot directly communicate with devices in the network, as these devices usually implement low power communication protocols, such as ZigBee. There are several methods to equip an Android device with a dongle capable of ZigBee communication. The use of multiple communication channels, such as the TCP channel, that uses WiFi to connect to a gateway, and the USB channel, that can connect to a device on the home automation network through an USB dongle. Modern mobile devices have embedded modules for several wireless communication technologies, such as WiFi, UMTS and Bluetooth. The home automation system consists of various home automation devices interconnected in a wireless sensor network, a gateway at the edge of the network and one or more client devices, that can be either smart phones, tablets, or laptops.[4]

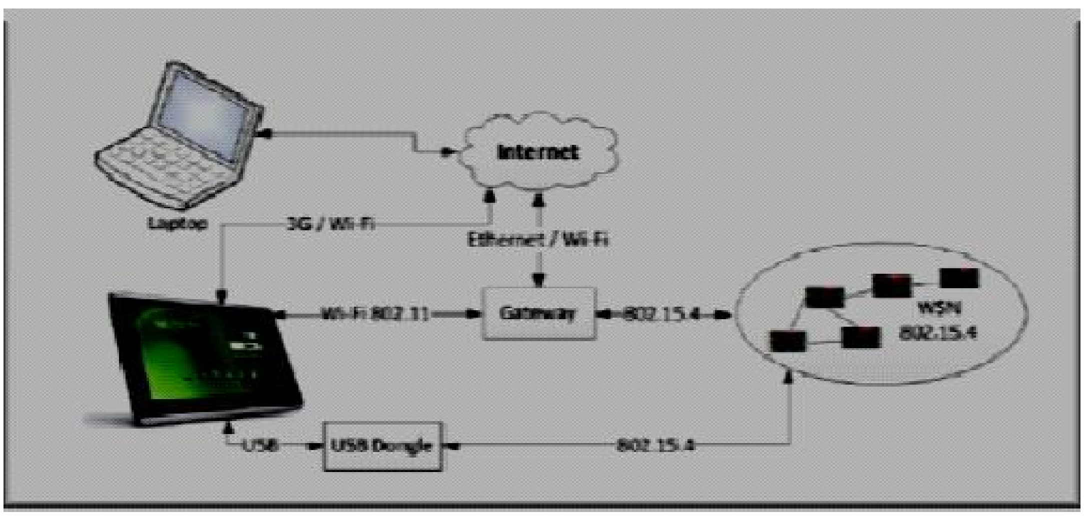

Figure 2.3: Home automation using ZigBee

2.2.3 Design and Implementation of SMS Based

Home Automation System based on SMS presents design and prototype implementation of a basic home automation system based on SMS technology. The automation system consists of two main components; the GSM modem, which is the communication interface between the home automation system and the user. GSM modem uses SMS technology to exchange data, and signaling between users and home automation system. The second module is the microcontroller, which is the core of the home automation system, and acts as the bridge between the GSM network (the user) and sensors and actuators of home automation system. Sensors and actuators are directly connected to hardware micro controller through appropriate interface. System supports a wide range of home automation devices; power management components, security, multimedia applications, and telecommunication devices. System security based on user authentication of each SMS being exchange, as each SMS contains user name and password (beside comments). User can easily configure home automation [5]

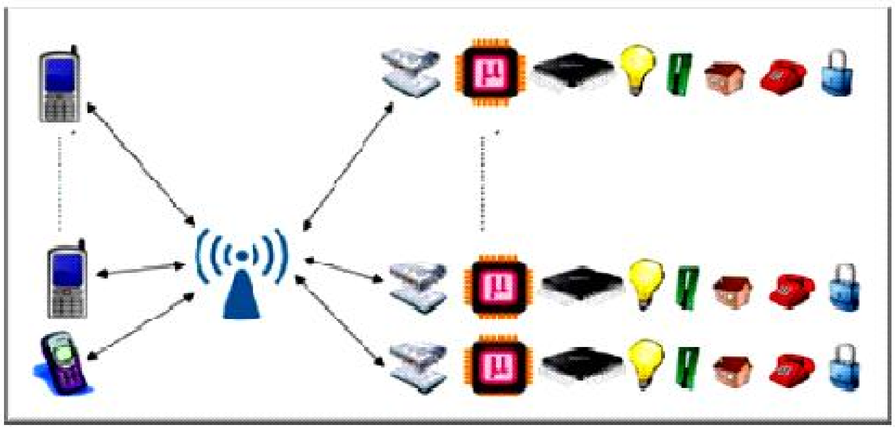

Figure 2.4 SMS Based Home Automation System.

2.2.4 Bluetooth Communication using a Touch Screen Interface with the Raspberry Pi

Bluetooth Communication with raspberry pi system brings a low cost stand-alone device which transmits data using the Raspberry Pi with Bluetooth and has a resistive touch screen display providing a user interface.

The Raspberry Pi is a low cost single-board computer which is controlled by a modified version of Debian Linux optimized for the ARM architecture.

The display contains a graphical user interface which provides various fields for data entry via an onscreen keyboard. The display is connected to the Raspberry pi via HDMI.

Background fields for displaying and entering the data has been implemented using html. Button press animations and buttons for navigating between different pages of the GUI have been implemented using JavaScript. The user enters the data at the client end in the GUI. The acknowledgement received from the server end is displayed in the GUI. If the data received at the server end matches with the sent data, the values are displayed. Cyclic Redundancy Check for achieving data integrity during the transmission [6].

2.2.5 Java Based Home Automation System

Java Based Home Automation System presents the design and implementation of a Java-based automation system that can monitor and control home appliances via the World Wide Web. The design is based on a standalone embedded system board integrated into a PC-based server at home. The home appliances are connected to the input/output ports of the embedded system board and their status are passed to the server. The monitoring and control software engine is based on the combination of Java Server Pages, JavaBeans, and Interactive C. The home appliances can be monitored and controlled locally via the embedded system board, or remotely through a web browser from anywhere in the world provided that an Internet access is available. Appliances at home are connected to an embedded system board (E-board). The control code on the E-board operates the appliances and communicates with Java-based code that resides at the server at home. The user can interact with the home automation system from anywhere at any time [7].

2.2.6 Android Based Home Automation Using Raspberry Pi

Android Based Home Automation Using Raspberry Pi system presents the design of home automation that can control Home appliances via Android device using Wi-Fi as communication protocol and Raspberry Pi as server system.

The proposed home automation system is working with very popular android phones. It is having mainly three components; the android enabled user device, a Wi-Fi router having a good scalable range, and a raspberry pi board .Here the users have provision to control the home appliances through android enabled device. This will improve the system popularity since there is no need for a wired connection, internet etc. The instructions from the user will be transmitted through the Wi-Fi network.

The raspberry pi board is configured according to the home system and it will enable the relay circuit as per user request. The relay circuit can control the home appliances also.

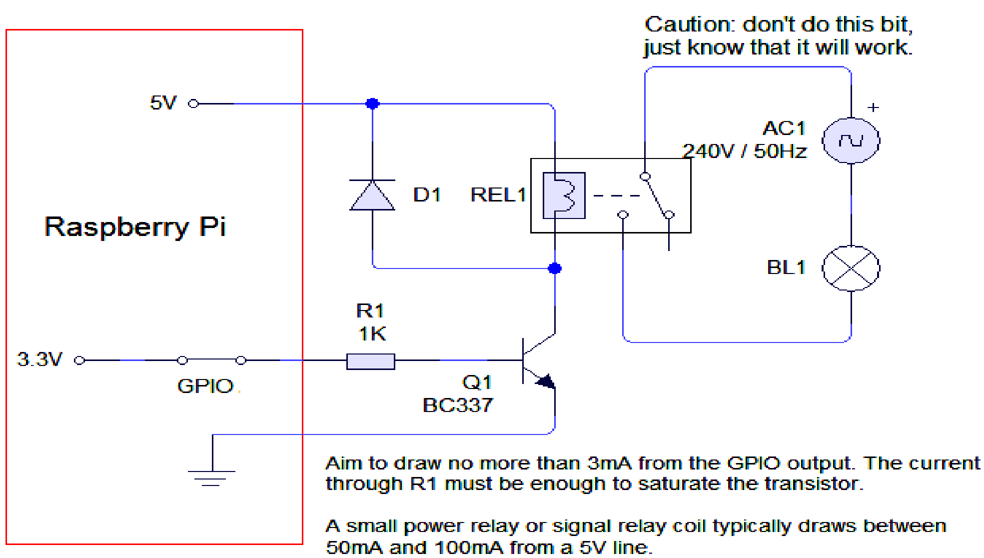

Figure.2.5 Relay connected with Raspberry-Pi.

WebIOPi is used To interface raspberry-pi with the external world.

WebIOPi is a web application which allows controlling Raspberry Pi's GPIO. It Support REST API over HTTP and CoAP .it can also handle more than 30 devices including ADC, DAC, sensors. The webIOPi interface allows better control of raspberry-pi [8].

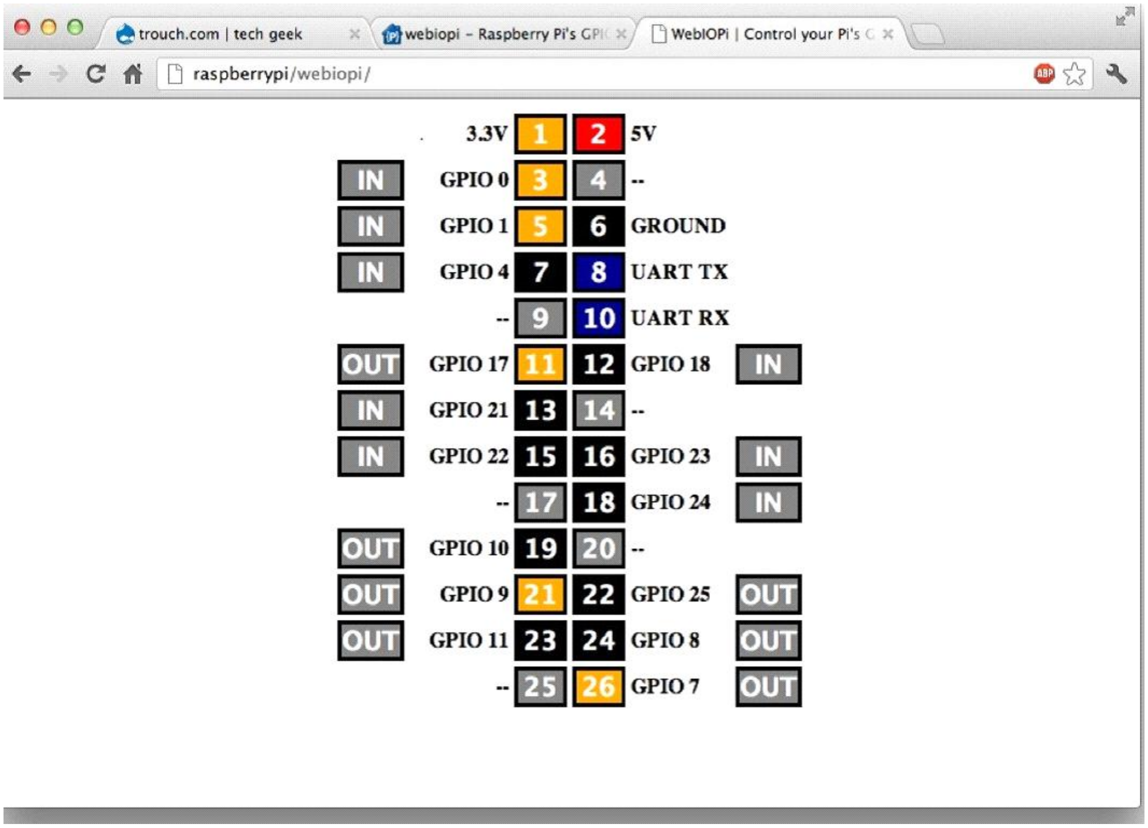

Fig.2.6 GPIO pin layout.

CHPTER THREE

SYSTEM DESIGN AND IMPLIMINTATION

3.1 Overview of the Smart Home System

The project revolves around developing a smart home system, which allows homeowners to control household devices through a Smartphone application. Currently it is focused on being able to open or close LEDs/household device, alarm when the system detects motion and enable/disable LEDs that open automatically at a dark.

This chapter describes the system and its components which might make it to the final product and the technologies used for the software part of the project.

3.2 System Description

The structure of the system consists of application installed in Smartphone that connects to Blynk server through the internet to Raspberry Pi. The Raspberry Pi interacts with Blynk server to perform controlling and monitoring function by software program written in C/C++ programming language. LEDs and household device are connected to raspberry pi via GPIO (general purpose input output) pins. Figure 3.1 below describes the structure of the system.

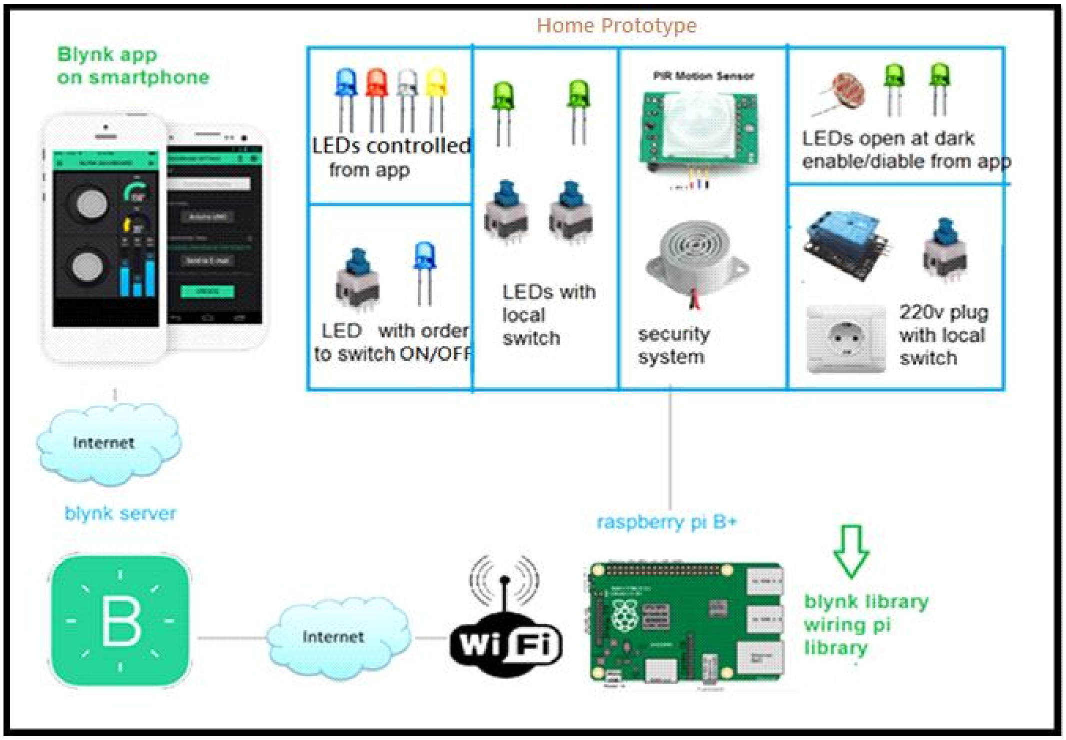

Figure 3.1: The structure of the system.

3.2.1 Types of switching

In this system there are three types of switching devices (LEDs/household device):

- The first type is switching LEDs "ON" or "OFF" directly from Smartphone application only.
- The second type of switching also from Smartphone application but users at home can send order to switching "ON" or "OFF" to application after that homeowner can response to this order.
- The third type of switching is from Smartphone application and also can switch "ON" or "OFF" local at home.

The first two types can provide privacy feature where as the last type can offer efficiency and flexibility features.

3.2.2 Security System

It consists of RIP motion detector sensor with alarm device. When someone enters home or when motion is detect by the security system, an alarm is triggered to sound which will remain "ON" until it’s triggered off by homeowner from application.

3.3 Hardware Design

3.3.1 Raspberry Pi

The Raspberry Pi is of a credit card-size, single-board computer launched in the United Kingdom by the raspberry pi foundation. The main objective of this is to encourage basic computer teaching in institutes. The Raspberry Pi has a broad com BCM2835 SoC, which comprises of an advanced RISC Machine 76JZF-S 700 MHz processor, video core IV GPU, and was originally distributed with 256 megabytes of RAM, later it is improved (Model B & Model B+) to 512 MB. It does not contain any built-in hard disk or solid-state drive, but it uses an SD card for booting and persistent storage, with the Model B+ using a Micro SD.

Figure 3.2 show the Raspberry Pi Model B+ that used in this project [9]

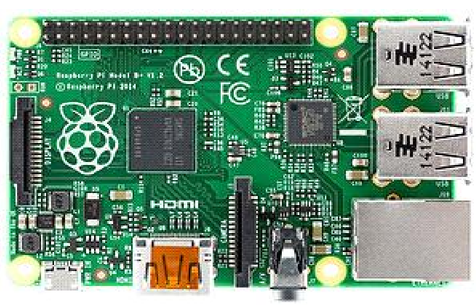

Figure 3.2 Raspberry Pi Model B+.

In order to use Raspberry Pi device it's required to start by installing an operating system onto an SD card. The Raspberry Pi operates on a LUNIX based open source operating system called Raspbian OS. This allows more control and flexibility in the software therefore making it easy to program the Pi. The Raspberry Pi communicates with the attached devices and sensors through C/C++ codes with addition library to control their functions. The Raspbian operating system was installed onto Raspberry Pi, which was obtained by downloading Raspbian-wheezy onto the SD card from the manufacturer's website [10].

General-Purpose Input/output (GPIO)

The Raspberry Pi Model B+ board contains a single 40-pin expansion header labeled as 'J8' providing access to 26 GPIO pins.

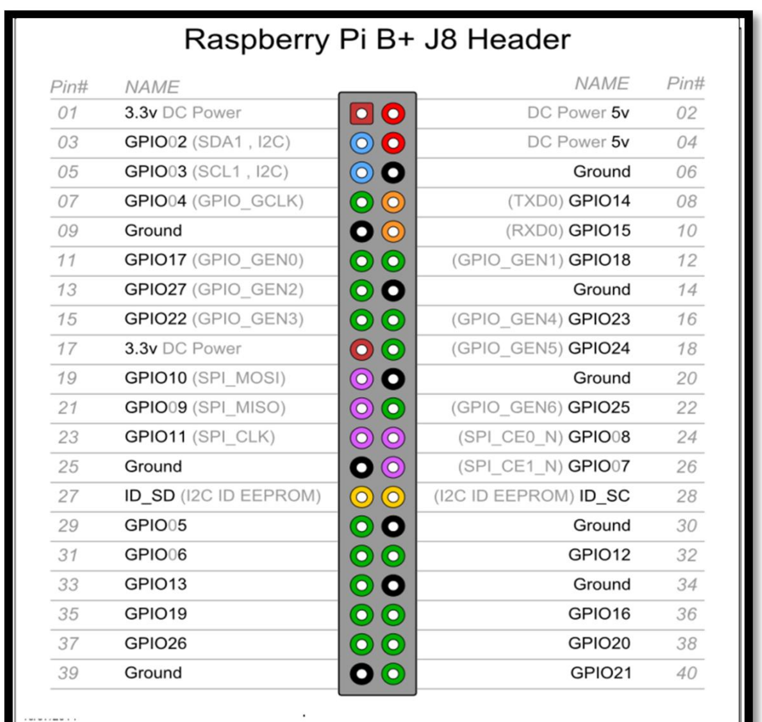

Figure 3.3 Raspberry Pi Model B+ In out pins.

3.3.2 Wi-Fi Dongle

In order to offer internet connection to Raspberry Pi in wireless mode. Will need to use USB Wi-Fi dongle connected to USB port of Raspberry Pi and configure it .The installation and configuration of Wi-Fi dongle is discussed in appendix A. Figure 3.4 shows Wi-Fi dongle module.

Figure 3.4 Wi-Fi dongle.

3.3.3 Light Emitting Diode (LED) and Switches

A light-emitting diode (LED) is a two-lead semiconductor light. It is a junction diode, which 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 [11].

Switches

Push Button Switches consist of a simple electric switch mechanism which controls some aspect of a machine or a process. Buttons are typically made out of hard material such as plastic or metal. The surface is usually shaped to accommodate the human finger or hand, so the electronic switch can be easily depressed or pushed. Also, most Push Button Switches are also known as biased switches. A biased switch, can be also considered what we call a "momentary switch" where the user will push-for "on" or push-for "off" type. This is also known as a push-to-make (SPST Momentary) or push-to break (SPST Momentary) mechanism [12].

The system uses 4 LEDs located in different rooms at home prototype connected directly to GPIO (5, 9, 10, and 11). Where other two LEDs with switch connected to GPIO via XOR circuit.

Figure 3.5 shows the input /output of XOR and switches circuit.

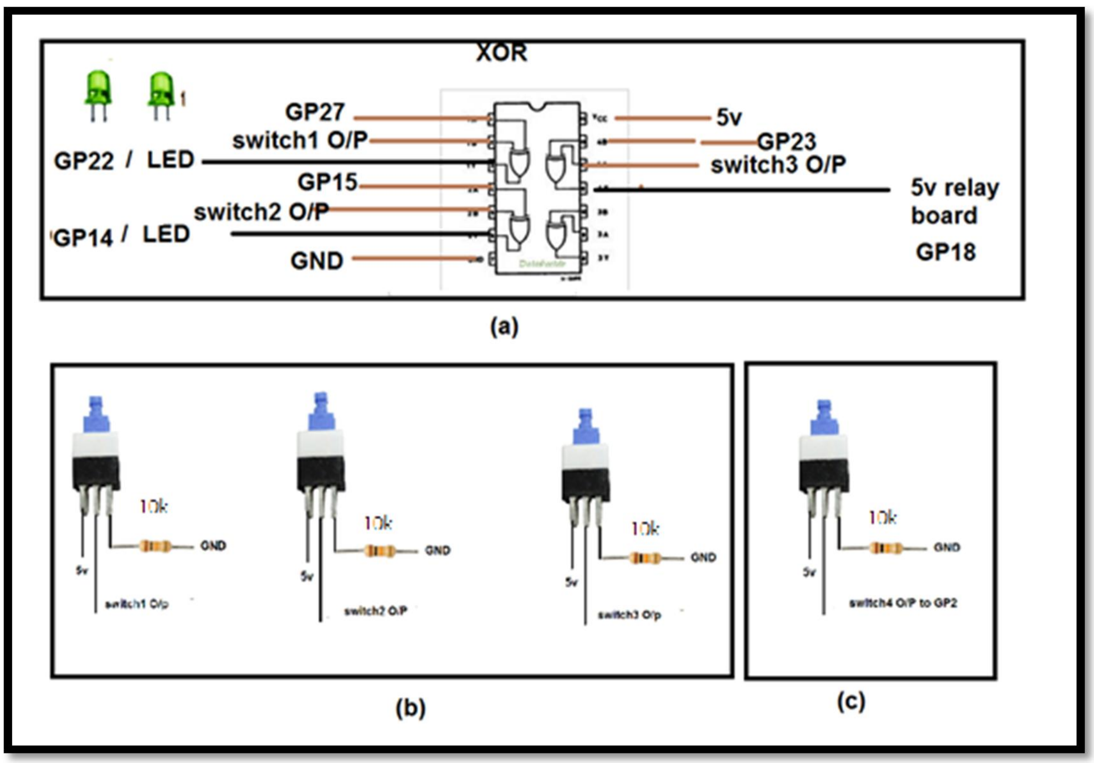

Figure 3.5 (a) XOR IC to provide switching function from both Blynk application and from local switch, the output of XOR go to LEDs/relay board to control and monitoring purpose.

Figure 3.5 (b) illustrate 3 switches circuit that used to open/close LEDs and household device.

Figure 3.5 (c) illustrate one switch that sends order to open/close LED.

3.3.4 Relay Module

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field, which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are double throw (changeover) switches.

The relay's switch connections are usually labeled COM (POLE), NC and NO:

COM/POLE= Common, NC and NO always connect to this; it is the moving part of the switch.

NC = Normally Closed, COM/POLE is connected to this when the relay coil is not magnetized.

NO = Normally Open, COM/POLE is connected to this when the relay coil is MAGNETIZED and vice versa.

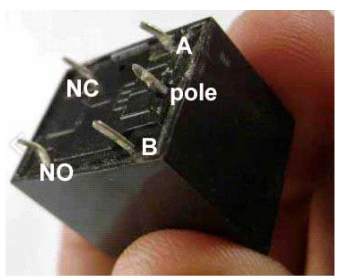

Figure 3.6 Relay.

There are 5 Pins in a relay. Two pins A and B are two ends of a coil that are kept inside the relay. The coil is wound on a small rod that gets magnetized whenever current passes through it.

COM/POLE is always connected to NC (Normally connected) pin. As current is passed through the coil A, B, the pole gets connected to NO (Normally Open) pin of the relay [13].

5V Relay that used to control high voltage device is illustrated in figure 3.7 below

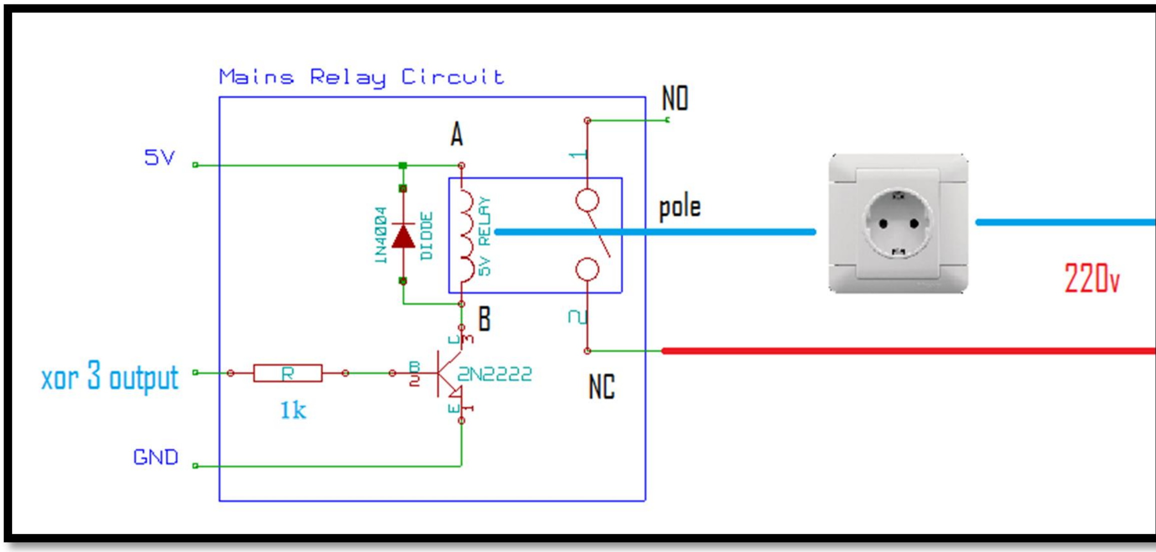

Figure 3.7 shows 5v relay circuit to switching "ON" or "OFF" 220v plug.

Transistors and ICs must be protected from the brief high voltage produced when a relay coil is switched off. Figure 3.7 shows how a signal diode 1N4001 is connected ‘backwards’ across the relay coil to provide this protection.

Current flowing through a relay coil creates a magnetic field which collapses suddenly when the current is switched off. The sudden collapse of the magnetic field induces a brief high voltage across the relay coil which is very likely to damage transistors and ICs. The protection diode allows the induced voltage to drive a brief current through the coil (and diode) so the magnetic field dies away quickly rather than instantly. This prevents the induced voltage becoming high enough to cause damage to transistors and ICs [13].

3.3.5 PIR Motion Detector

The main electronic component that used to allows picking up this detection is the PIR motion sensor. The PIR motion sensor (figure 3.8) is a sensor which detects movement through picking up infrared waves. Being that a person emits infrared radiation, the detector is able to detect this radiation and react, and according to the how the circuit is designed to react. Being that people naturally give off infrared radiation (heat energy), because of our generated body heat, the sensor can easily detect people walking and moving through a vicinity within the sensor's range. The sensor can also pick up the movement of inanimate objects as well, such a rolling ball, because as these objects move, friction acts on them, generating heat. This heat emits infrared radiation, which the PIR sensors may be able to detect if enough radiation is given off.

The motion sensor has a sensitivity range up to 20 feet (6 meters) and a 110° x 70° detection range, making it a wide lens detection sensor. This means it can measure 110° vertically (from top to bottom) and 70° horizontally (from left to right) [14].

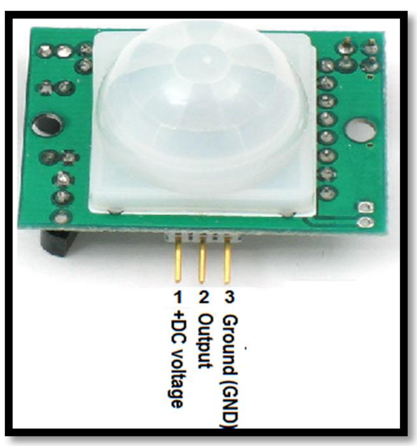

Figure 3.8 The PIR motion sensor is a 3-pin device.

In order To design the security system, the PIR motion pins are connected as follows:

Pin1 receives the positive voltage from raspberry pi 5v pin .Pin2 is the output pin of PIR motion is connected to GPIO 4 .Pin3 is a ground pin .

The alarm buzzer is connected to GPIO 17. As illustrated in figure 3.9

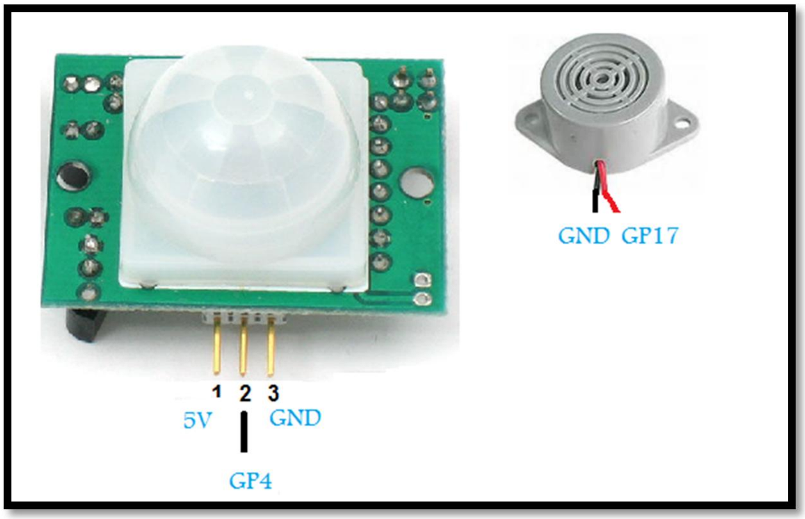

Figure 3.9 illustrate the design of the security system.

3.3.6 Light Dependent Resistor (LDR)

A photo resistor or light-dependent resistor (LDR) or photocell is a light-controlled variable resistor. The resistance of a photo resistor decreases with increasing incident light intensity; in other words, it exhibits photoconductivity. A photo resistor can be applied in light-sensitive detector circuits, and light- and dark-activated switching circuits.

This project uses LDR with LED to offer automatic opening LEDs at dark after enabled from Smartphone application. Figure 3.10 shows LDR with LED circuit.

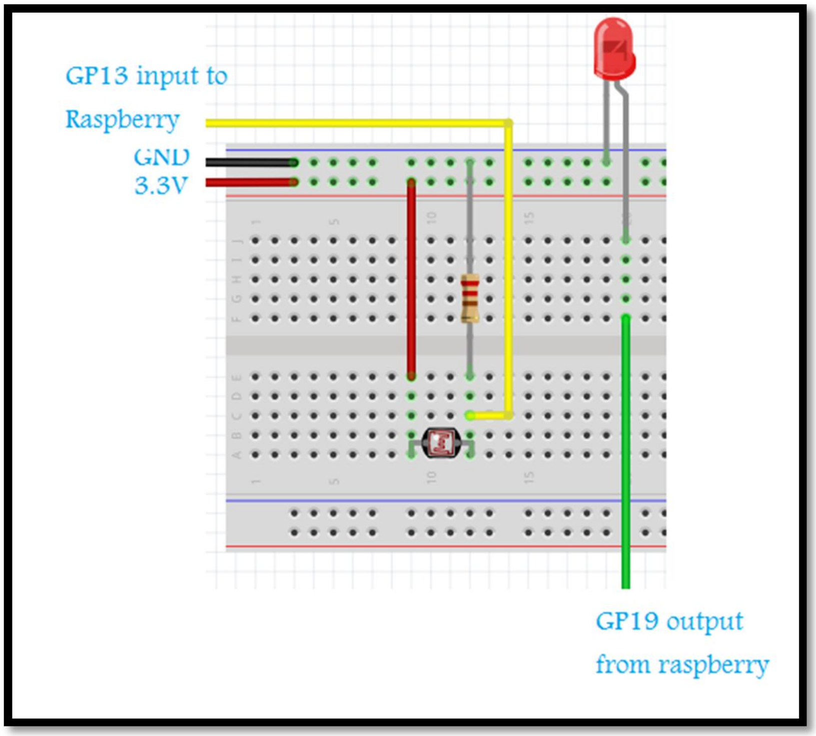

Figure 3.10 flashing LED lights circuit that depends on LDR resistor.

One of LDR terminal is connected to 3.3V and the other terminal is connected to ground via 10K ohm resistor .When homeowner enabled LDRwithLED function at Smartphone, Raspberry Pi reads the voltage value from GPIO13. When it reads low value the LEDs will be "ON”.

3.4 Software Design

Software design for develop a smart home system will be discussed in this section. The Raspberry Pi is programmed using C/C++ programming languages.

The following figures show the flow charts for functions that used to control, enable/disable and to monitor process.

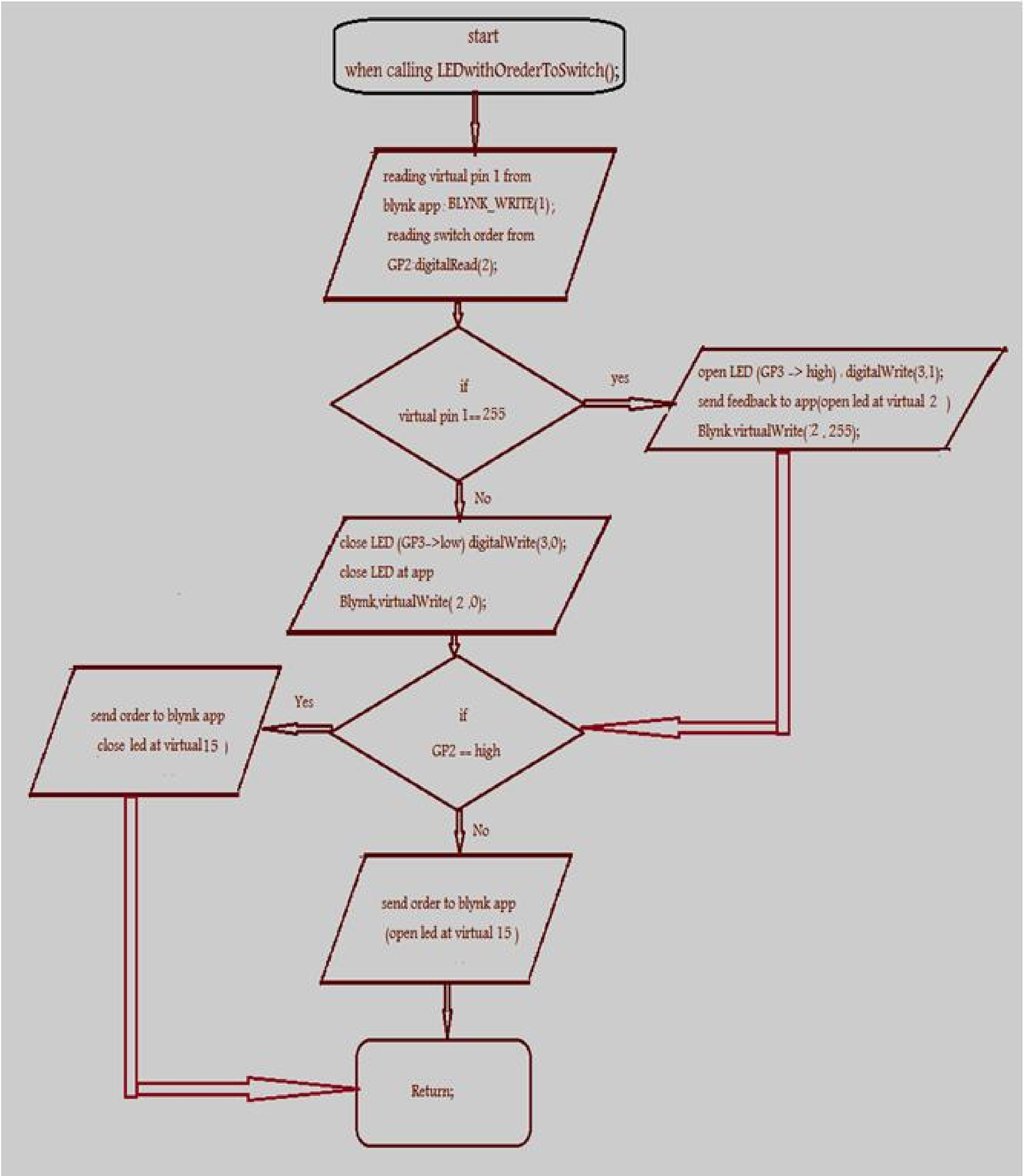

Figure 3.11 flow chart of LEDwithOrderToSwitching function.

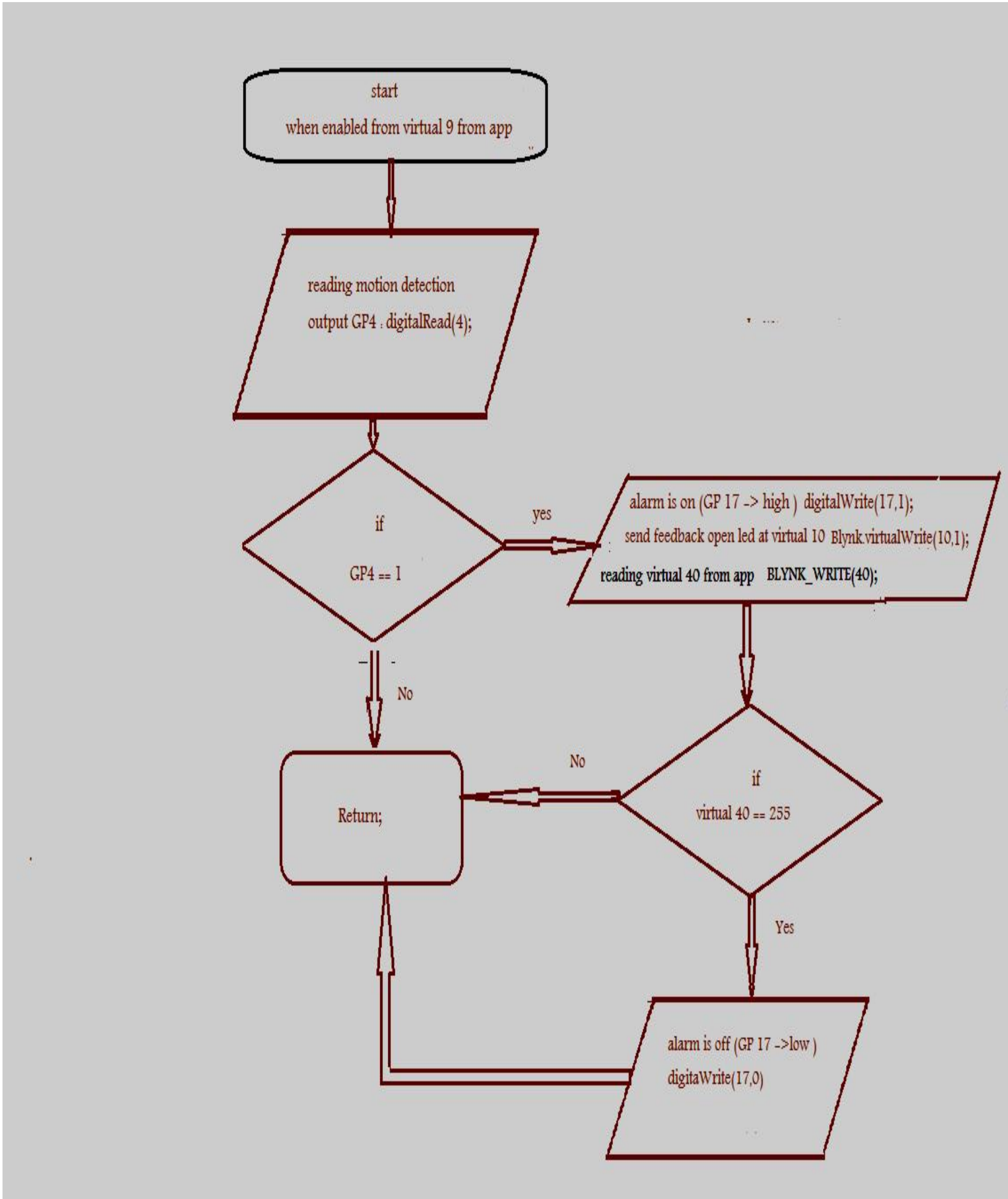

Figure 3.12 flow chart for security function.

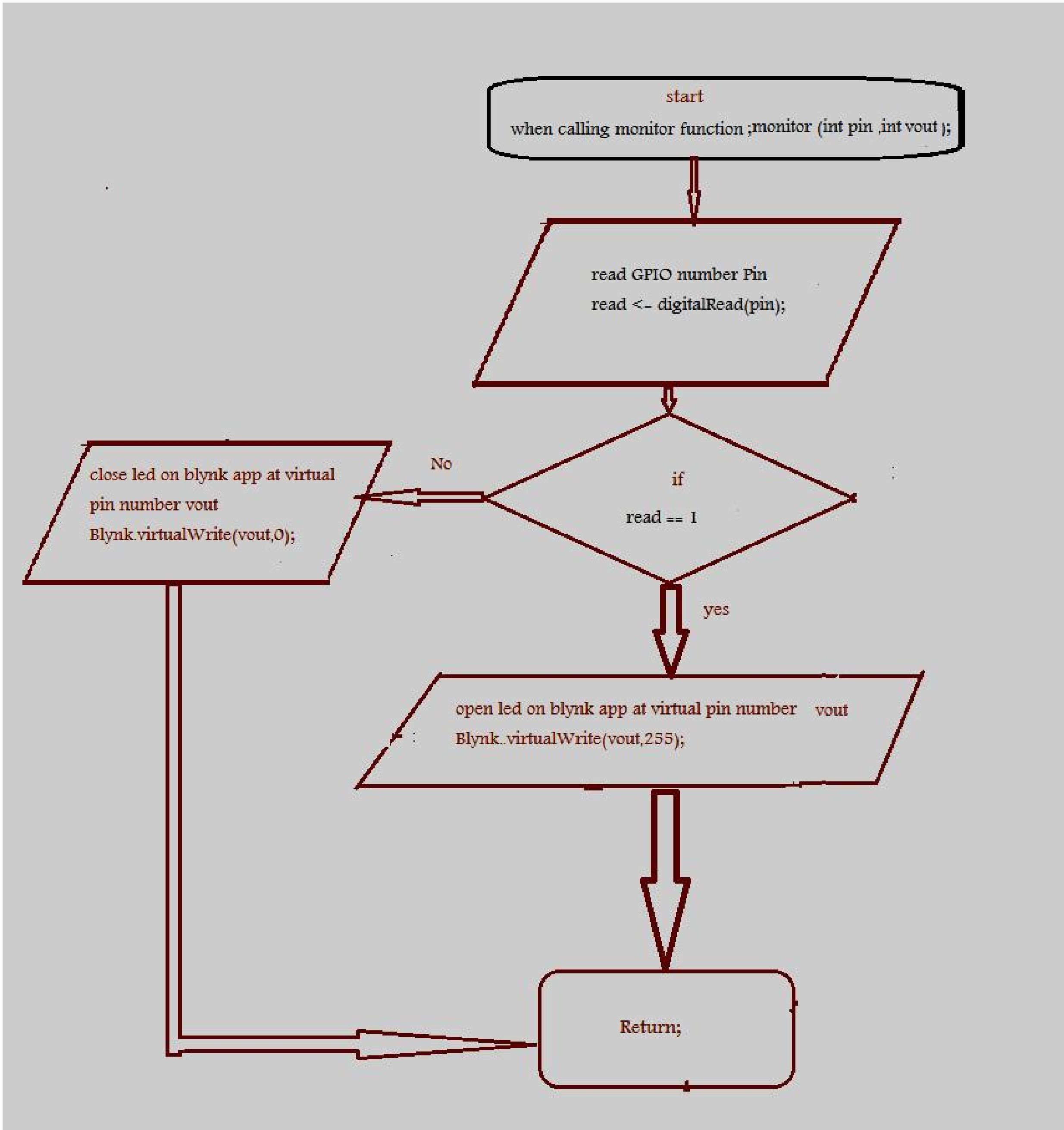

Figure 3.13 flow chart for monitoring function.

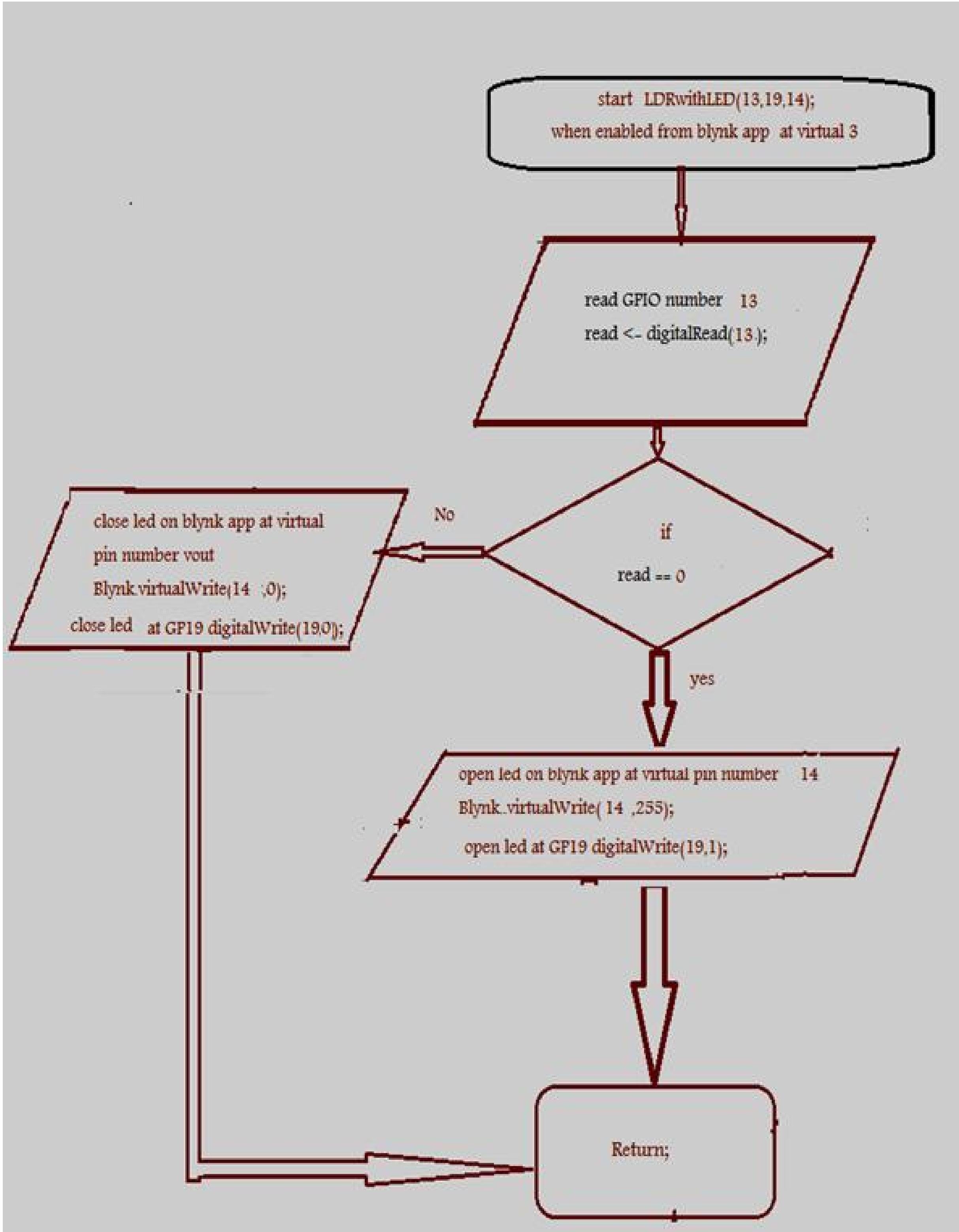

Figure 3.14 flow chart of LDRwithLED function.

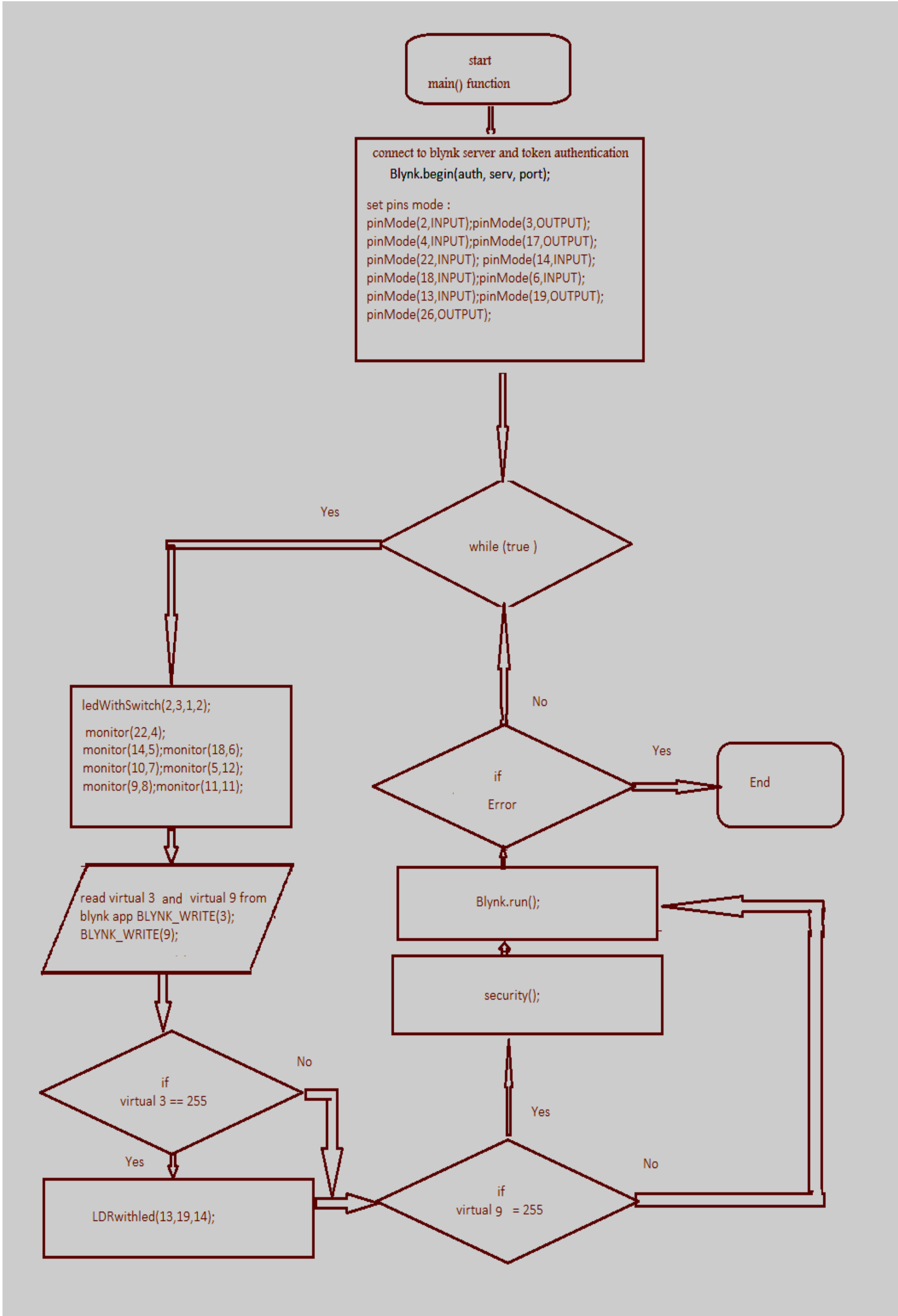

Figure 3.15 flow chart of main function in C/C++ program.

To make the source code main.cpp run automatically after boots up the Raspberry Pi by type the command (modified bash file) in Terminal as follows:

```
#create folder
mkdir ./bin
cd ./bin
#move source code to blynk library
sudo mv /home/pi/main.cpp /home/blynk-library/linux
#compile source code
cd /home/blynk-library/linux
make clean all target=raspberry
#create script
sudo nano script_auto_run
#write this script in nano editor
#!/bin/bash
# Script to start our application
echo "Doing autorun script..."
sudo /home/pi/blynk-library/blynk –authentication_token
#save the script by pressing Ctrl+O and exit by pressing Ctrl+X
#made this script executable
sudo chmod 755 script_auto_run
Setting it to be automatic run
sudo nano /etc/rc.local
#add the this script inside rc.local
/home/pi/bin/script_auto_run
#save the script by pressing Ctrl+O and exit by pressing Ctrl+X
#reboot Raspberry Pi
sudo reboot
```

3.5 Blynk Application on Smartphone

Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. Each user of application has username and password to log in to dashboard.

Digital dashboard is used to build a graphical user interface by simply dragging and dropping widgets.

Figure 3.13 shows the dashboard of Blynk application that used to control/monitor devices in smart home system.

Figure 3.16 illustrate screen shot from Smartphone for Dashboard of Blynk app.

LEDs on dashboard will receive feedback signal from device, when the button used to send control signal to Raspberry Pi is pressed.

Table 3.1 GPIO and Virtual pins for button and LEDs on dashboard

Devices	**Button** **on Dashboard**	**LEDs on Dashboard** **To show device status**
	LD GP5 LD GP9 LD GP10 LD GP11	LED9 (virtual 12) LED8 (virtual 8) LED6 (virtual 7) LED7 (virtual 11)
	LS/ORD virtual 1	ORD (virtual 15) LE10 (virtual 2)
	LS2 GP27 LS2 GP15	LED3 (virtual 4) LED4 (virtual 5)
	SECURITY virtual 9 ALARM OFF virtual 40	LED2 (virtual 10)
	LDR&LED virtual 3	LED1 (virtual 14)
	LS2 AC GP23	LED5 (virtual 6)

3.6 System Integration and Testing

The hardware and software design needed to be integrated as a complete system. Testing of the complete system is needed to ensure that system integration is functioning as required. Each part of the development of hardware and software designs should be tested before combining all parts into a complete system. Each part must pass the testing process to make sure that the system on that part is functioning correctly before moving to another part. If the part has errors, troubleshooting should be made to correct the system.

CHAPTER FOUR

RESULTS AND DISSCUSION

4.1 Results

After connecting the Raspberry Pi with LEDs and security devices as mentioned in the previous chapter and operating the Raspberry Pi, it executes the program which allows communications with the Smartphone via Blynk app.

4.1.1 LEDs Controlled from Blynk app.

When the following buttons (GP5, GP9, GP10, and GP11) are pressed from Blynk application, 4 LEDs are activated and their states changed to "ON". Shows figure.4.1.

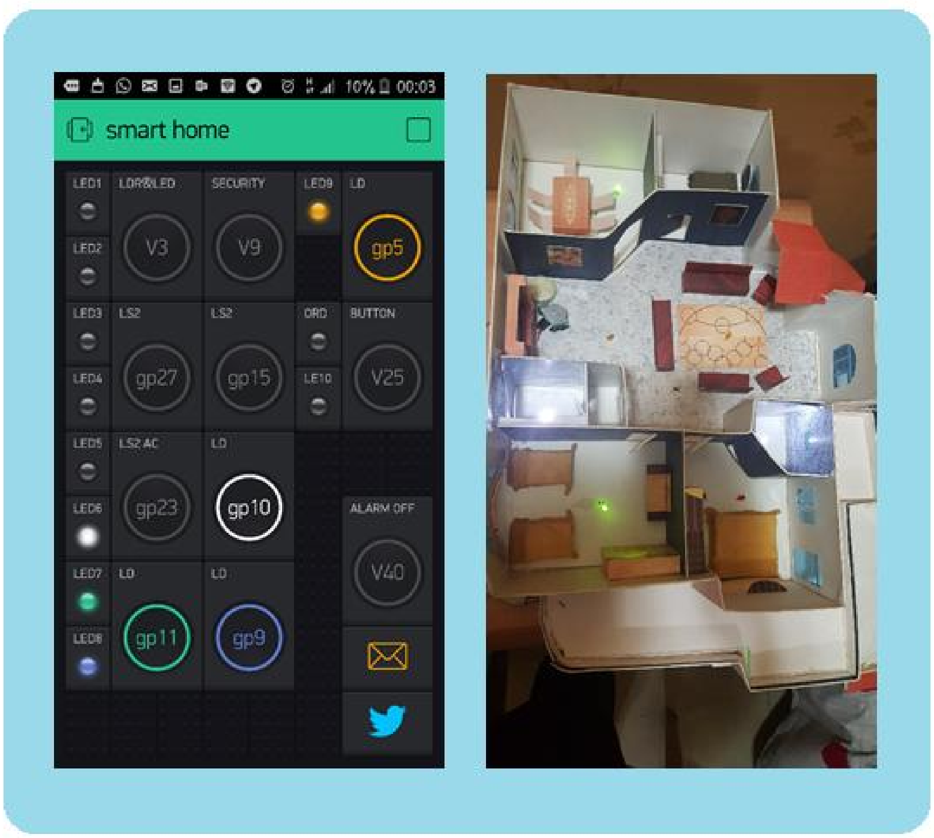

Figure.4.1 Illustrate the operation of LEDs Controlled from app.

4.1.2 LED with Order to Switch

When the switch pressed on home prototype, the state of ORD LED at the Dashboard is changed. The user cans response to the order by pressing the button V1 at the Dashboard to open the LED at home prototype. As illustrated if figure 4.2.

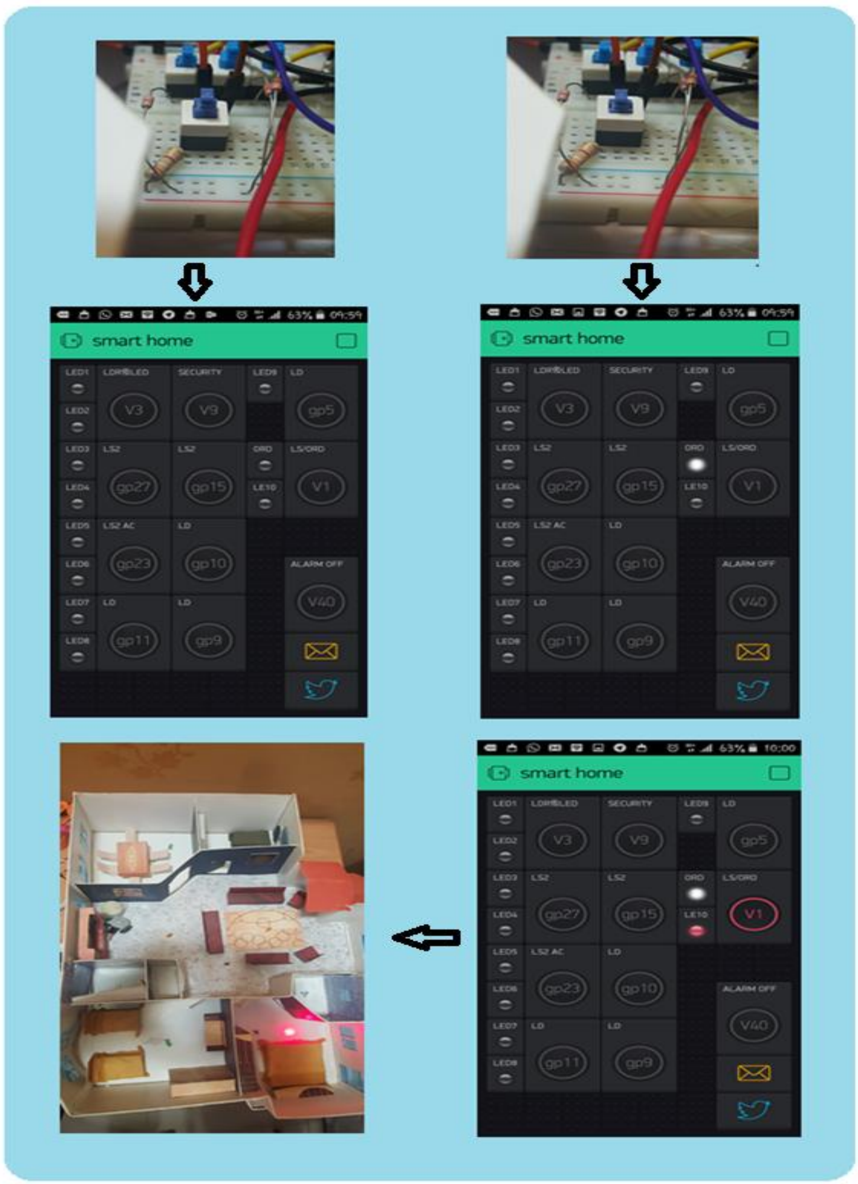

Figure 4.2 illustrate the operation of LED with Order to Switch.

4.1.3 Security System

The security system operates after it is activated from the application by pressing the button V9. When it is activated, the motion detector starts to detect motion and if it detects any motion it raised alarm and sends email to the user and sets the LED2 to ON state in the dashboard. Figure 4.3 illustrate the mechanism of its operation.

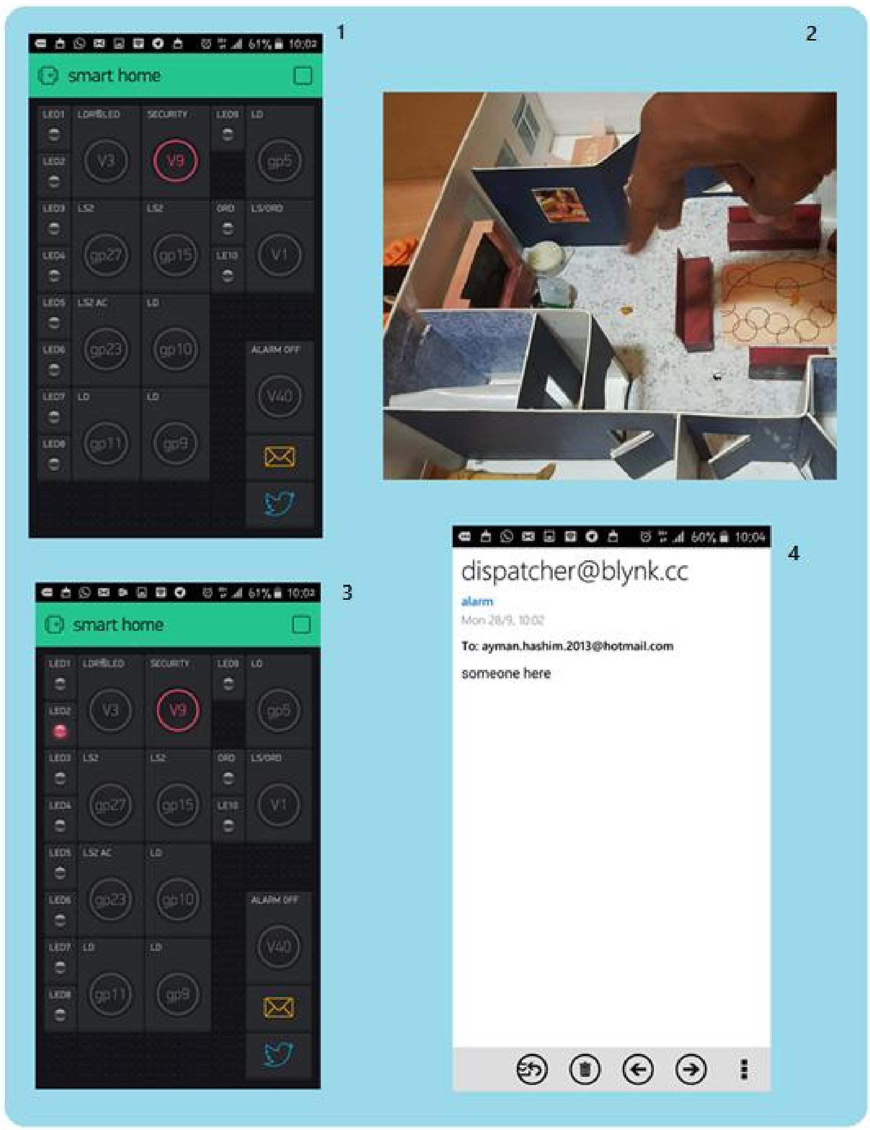

Figure 4.3 illustrate the operation of Security System.

4.1.4 LEDs with Local Switches

The operation of LEDs with local switches is illustrated in Figure 4.4.

Figure 4.4 illustrate the operation of LEDs with Local Switches.

4.1.5 LDR with LED

When LDR with LED is activated from button V3, the LDR detector starts to work and during the dark it switches the LEDs of the system ON. Figure 4.5 illustrates the mechanism of its operation.

Figure 4.5 illustrate the operation ofLDR with LED.

4.1.6 220 V plug with Local Switch

When the switch is set to ON state the 220v plug is activated and when the switch is set OFF state the 220v plug is deactivated . it is possible to activate and deactivate the 220vplug from GP23 button .shows figure 4.6

.

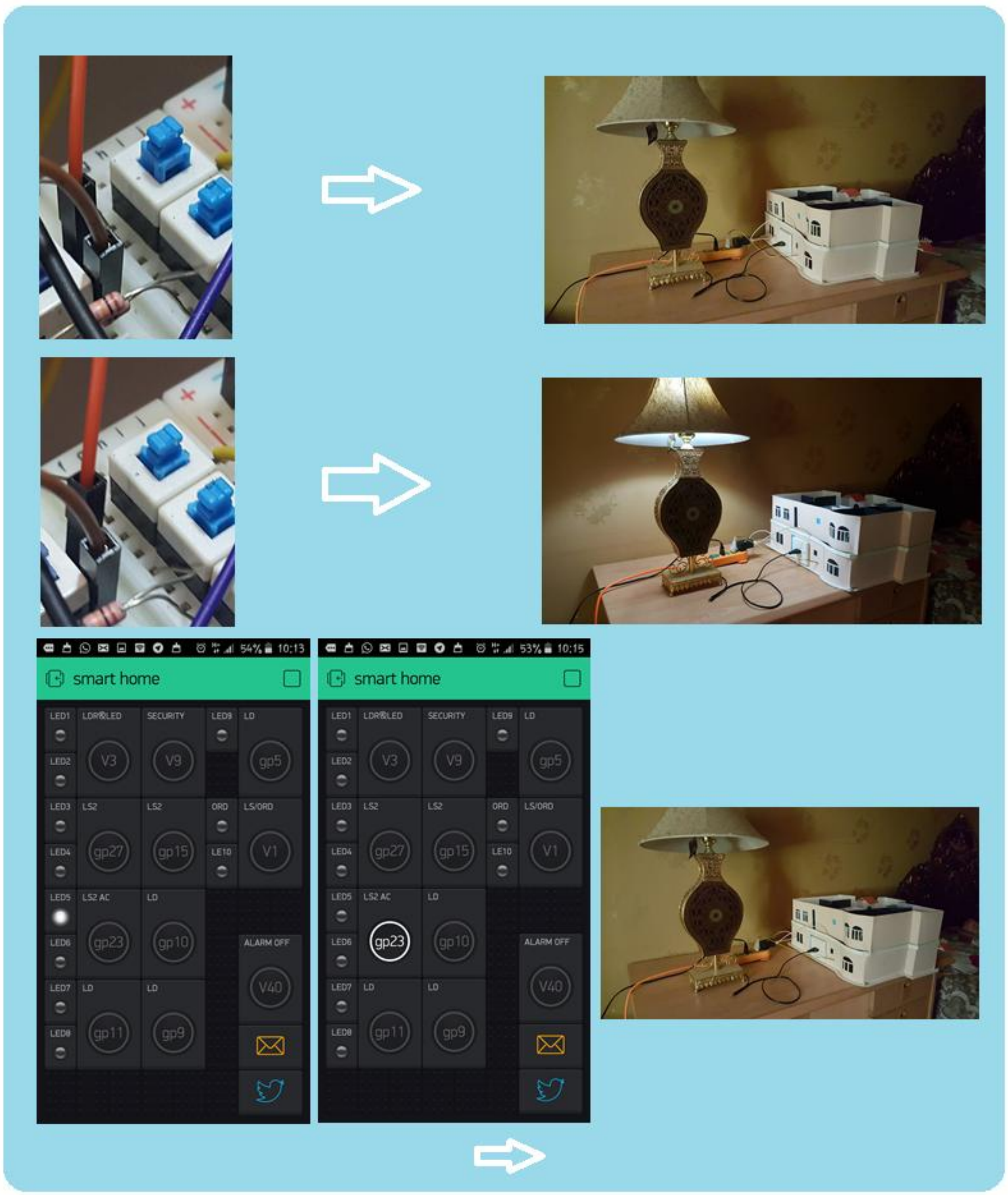

Figure 4.6 illustrate the operation220 V plug with Local Switch.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This project successfully design smart home system that enable to homeowners to control and monitor appliances remotely from Smartphone application over internet.

This application uses a graphical user interface which is easy to use.

The communication mechanism between Smartphone application and raspberry pi is achieved by Blynk intermediate server.

Finally the system is very successful at accomplishing the objectives.

5.2 Recommendations and Future Work

In future work, the following point can be taken under consider:

- Use HD camera with cloud storage to enhance the security system of smart home. HD camera take picture when detect motion and the Raspberry Pi upload picture to cloud storage such as Dropbox, Google drive and OneDrive .
- Use wireless connection without internet to control/monitor household devices when user locates at home.
- Use other feature that offered by Blynk application such as Reading analog data from sensor such as temperature sensor.

REFERENCES

[1] Kortuem, G., Kawsar, F., Fitton., D., Sundramoorthy, V.: Smart objects as building blocks for the internet of things. Internet Computing, IEEE, vol. 14, no.1, 44--51 (2010)

[2] Dennis, Andrew K. Raspberry Pi Home Automation with Arduino. Packt Publishing Ltd, 2013

[3] http://en.wikipedia.org/wiki/ZigBee. (19/9/2015)

[4] Felix, Carelin, and I. Jacob Raglend. "Home automation using GSM." Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on. IEEE, 2011.

[5] Olteanu, A-C., et al. "Enabling mobile devices for home automation using zigbee." Control Systems and Computer Science (CSCS), 2013 19th International Conference on. IEEE, 2013.

[6] Shaiju, Paul, Antony Ashlin, and B. Aswathy. "Android Based Home Automation Using Raspberry Pi." International Journal of Computing and Technology (2014).

[7] Gopinath, Shanmuga, et al. "Bluetooth Communication using a Touchscreen Interface with the Raspberry Pi." SoutheEastCon. 2013.

[8] Al-Ali, Abdul-Rahman, and Mohammad Al-Rousan. "Java-based home automation system." Consumer Electronics, IEEE Transactions on 50.2 (2004): 498-504.

[9] Prasad, Sanjana, et al. "SMART SURVEILLANCE MONITORING SYSTEM USING RASPBERRY PI AND PIR SENSOR."

[10] https://www.raspberrypi.org/ (19/9/2015).

[11] https://en.wikipedia.org/wiki/Light-emitting_diode (19/9/2015).

[12] https://www.futureelectronics.com/en/switches/push-button-switches.aspx (19/9/2015)

[13] http://electronicsclub.info/relays.htm (19/9/2015).

[14] http://www.learningaboutelectronics.com/Articles/Motion-detector-alarm-circuit.php (19/9/2015) .

```
/**
 * @file      main.cpp
 * @authors  Ayman  , Ahmed  , Al-fatih
 * @date      Mar 2015
 * @brief
 */

//#define BLYNK_DEBUG

#define BLYNK_PRINT stdout
#ifdef RASPBERRY
#include<BlynkApiWiringPi.h>
#else
#include<BlynkApiLinux.h>
#endif
#include<BlynkSocket.h>
#include<BlynkOptionsParser.h>
staticBlynkTransportSocket _blynkTransport;
BlynkSocketBlynk(_blynkTransport);

boolled_state=LOW; //default
int virtualValue,virtualValue3;
int virtualValue9,virtualValue40;
int virtualValue41;
intpid,prev=0;
inti=1;
```

```
 BLYNK_WRITE(1)
    {
virtualValue = param.asInt(); // Get the value from VPin 1 (could also be a
slider)
    }

 BLYNK_WRITE(3)
    {
      virtualValue3 = param.asInt(); // Get the value from VPin 3 (could also
be a slider)
    }


 BLYNK_WRITE(9)
    {
     virtualValue9 = param.asInt(); // Get the value from VPin 9 (could also
be a slider)
    }

 BLYNK_WRITE(40)
    {
     virtualValue40 = param.asInt(); // Get the value from VPin 40 (could
also be a slider)
    }

 BLYNK_WRITE(41)
```

```
 {
  virtualValue41= param.asInt(); // Get the value from VPin 41 (could also be a slider)
 }



voidledWithSwitch(intpin,intpout,int vin ,intvout)
 {
// Reading virtual pin 1 value
  BLYNK_WRITE(1);
// Reading order  from  pin variable
int order = digitalRead(pin)

//when swicth is "on"  send order (open ORD LED on blynk app)
if (order==1)
Blynk.virtualWrite(15,255);
else
        Blynk.virtualWrite(15,0);

//when app send controle signal on virtaul variable to open/close led
// feedback sent by vout virtual variable
if(virtualValue ==255)
{       digitalWrite(pout,1);
Blynk.virtualWrite(vout,255);}

else
{       digitalWrite(pout,0);
```

```
Blynk.virtualWrite(vout,0);}

 }



// moitor function is used to read GPIO  on pin variable  and send it to blynk
app
void monitor(int pin ,intvout)
 {
        int read = digitalRead(pin);

if(read==1) //open led in mobile phone
Blynk.virtualWrite(vout,255);
else
Blynk.virtualWrite(vout,0);

 }

// LDRwithled used to open led automaticlly at dark
//read LDR state and open/close led connected to pout GPIO
voidLDRwithled(intpin,int pout ,intvout )
 {
        int read = digitalRead(pin);
//open led in mobile phone when read low value
if(read==0)
        {
                Blynk.virtualWrite(vout,255);
```

```
        digitalWrite(pout,1);
   }

else
{  Blynk.virtualWrite(vout,0);
        digitalWrite(pout,0);
        }

 }
// alarm buzzer when detect motion
void  security()
 {
//read form motion detection
intmotSen;
//alarm when detect motion
if (motSen  = digitalRead(4))
       {digitalWrite(17,HIGH);
Blynk.virtualWrite(10,255);
                //send tweet notification
                if(i==1)
                Blynk.tweet("#smart_home_project");
                Blynk.email("ayman.hashim.2013@hotmail.com", "alarm",
"some one here");
                i==0;
                 }
        BLYNK_WRITE(40)
                //close alarm
```

```
            elseif(virtualValue40)
            { digitalWrite(17,LOW);
Blynk.virtualWrite(10,0);
            i=1;
             }
            elsereturn;
 }

int main(intargc, char* argv[])
 {
constchar *auth, *serv, *port;
parse_options(argc, argv, auth, serv, port);

Blynk.begin(auth, serv, port);
//chose pin modes  for LEDs
pinMode(2,INPUT);
pinMode(3,OUTPUT);
//pins modes for security
pinMode(4,INPUT);
pinMode(17,OUTPUT);
            //Pins Mode for others
            pinMode(22,INPUT);
            pinMode(14,INPUT);
            pinMode(18,INPUT);
            pinMode(6,INPUT);
pinMode(13,INPUT);
            pinMode(19,OUTPUT);
```

```
pinMode(26,OUTPUT);




while(true)
 {
//2 input to raspberry ,3oput , 1 virtual read from app, 2 virtual feedback to
app
ledWithSwitch(2,3,1,2);
//22 from switch to raspberry ,4 virtual , 27 select from phone
monitor(22,4);
//14 from switch to raspberry ,5 virtual , 15 select from phone
monitor(14,5);
//for ac power
//18 from switch to raspberry ,6 virtual , 23 select from phone
monitor(18,6);
//led withot switch
monitor(10,7);//10 from raspberry ,7 virtual ,10 select from phone
monitor(9,8);//9 from raspberry ,8 virtual ,9 select from phone
monitor(11,11);//11 from raspberry ,11 virtual ,11 select from phone
monitor(5,12);//5 from raspberry ,12 virtual ,5 select from phone
//enable LDR with switch by Virtual 3
  BLYNK_WRITE(3);
        if(virtualValue3)
        LDRwithled(13,19,14);
//ENABLE SECURTIY by virtual 9
```

```
  BLYNK_WRITE(9);
if(virtualValue9)
security();
//to conncet to blynk server
Blynk.run();
 }

return 0; }
```