Expertise
and insight
for the future

Rupesh Majhi

IoT: Home Automation

Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Programme in Electronics

Bachelor’s Thesis

01 October 2019

Author Title Number of Pages Date	Rupesh Majhi IoT : Home Automation 31 pages + 2 appendices 1st October 2019
Degree	Bachelor of Engineering
Degree Programme	Electronics
Professional Major	
Instructor	Janne Mäntykoski, Senior Lecturer

Thesis target is to present a cost reduced and flexible home control and monitoring system using Raspberry Pi and Blynk mobile application with IP connectivity, accessing and controlling devices and appliances remotely using smart phone application.

To implement this project, Four-Channel Relays as a switches, a DHT11 sensor for measuring temperature and Humidity, and Blynk server working as a bridge connecting Raspberry Pi and hardware. In addition, system is also dedicated to a local server with respect to similar systems and offers high secure and encrypted communication protocol to monitor and control the home environment. Thesis contains mainly two part, Theoretical part deals with definition and history of IoT. The practical part describes the process of making system.

The designed home automation system in this project can control lights, air conditioners which are connected in the relays. Also the system monitors house surrounding with dedicated DHT11 sensor with live update of data in very short interval. The system itself is expandable and can be used for further developments by adding varieties of available sensors according to the needs.

Keywords	IoT, Blynk, Raspberry Pi

Table of Contents

Appendices

List of Abbreviations and Symbols

AVR	Advance Virtual RISC
BLE	Bluetooth Low Energy
CSI	Camera Serial Interface
DSI	Digital Serial Interface
ECHO	Electronic Computing Home Operator
GPIO	General Purpose Input Output
GSM	Global System for Mobile
HDMI	High-Definition Multimedia Interface
IEEE	Institute of Electrical & Electronic Engineers
IoT	Internet of Things
KB	Kilo Byte
LAN	Local Area Network
LPDDR	Low-Power Double-Data-Rate
NAT	Network Address Translation
OS	Operating System
OSI	Open Systems Interconnection
PIC	Peripheral Interface Controller
RF	Radio Frequency
RISC	Reduced Instructions Set Computer
SD	Secure Digital
SoC	System on Chip
USB	Universal Serial Bus
µC	Microcontroller

1 Introduction

In the modern technology, IoT and smart home system are creating numerous buzz. Day to day lives is now more comfortable and easier with the increasing smart homes and IoT devices.

Home automation has transformed the way people live. For instance switching the lights on at home before actually stepping inside. How about changing the temperature of the room with just the touch of a button from mobile device? Smart home automation has lots of potential, it has become the future of our lives.

IoT smart home automation solutions competent of changing the approach we live our lives. Ideal home automation system is believed to save our time,money and additionally improving the quality of life.

By the end of 2020, global home automation capital value will grasp over 40 billion dollars. These smart home devices range from things like coffee maker, refrigerators, air conditioners and a range of safety and security devices like alarms systems and surveillance cameras[1].

When people are not around home several problems occur which may be saving energy or monitoring house hold appliances.

To provide remotely controlled and monitoring system for home appliances and its surrounding. The system uses Raspberry pi embedded system along with Blynk mobile application to communicate with smartphone and the system. Also, in order to get maximum security, we install our Blynk server locally which restricts access to our network so nobody except us could access it. In this case all data is stored locally within our network and not send via Internet.

The goal of the project is to develop prototype for a smart home system that will enable homeowner to control and monitor electrical peripherals by simply using smartphone.
Some of the major objectives of project are:

- Home automation system designed on single-board computer, Raspberry Pi which will be a central hub in the whole system.

- To provide communication mechanism between smartphone and Raspberry Pi with Blynk app over Blynk server. Blynk app controls hardware remotely and displays sensor data, stores data and does many other things.
- More security and encryption, local Blynk server installed in our system which restricts access to our network so nobody except us could access it.

First we connect our electronics devices like sensors, relay etc to the GPIO pins of Raspberry Pi. Then Raspberry Pi will be connected to a smartphone by internet connection. Since Blynk has its own server which acts as an intermediate server between Raspberry Pi and the smartphone which allows the exchange of control commands and the response for it. We are going to discuss Blynk application and its operation in more detail in Section 2 and 3.

2 Theoretical Background

2.1 History

For a long time concept of home automation systems were just an idea but not an actual structure. For past decades, science fiction has just scrutinized the idea of home automation which featured in the writing of many 19th century science-fiction authors, comics and cartoons such as The Jetsons. Even though the concept of smart home has been around for sometime, real smart home happened in real life only a short while .
Before anyone can ever have smart devices, one has to discover the devices. Between 1901 to 1920, development in domestic appliances accelerated, initiated by invention of the vacuum cleaner powered by engine. Following this came the arrival of toaster, irons, refrigerator & washing machines, to revolutionize domestic lives which were, however, too costly for many household to afford.

After many years of development in computing and developing electronics, the first ever smart device name the ECHO IV was born. The machine was able to produce shopping list, monitor home's surrounding temperature and control appliances. A few year later the "Kitchen Computer" arrived taking cooking one level further even offering the recipes itself but was very expensive and never sold.

After the born of microprocessor in early 1970's, technology was finally taking a step in order that everyone could afford. The drastic drop in the price of electronics, finally common people were able to afford the domestic appliances. In mid-nineties electrical switch called "The Clapper" was invented, which was basically operated by a clap of a hand. The 20th century's most iconic inventions and first real move towards smart home automation.

Smart home started to create popularity in the late 1990's and early 2000's as internet technology boosted and smart home concept became a more affordable. The "Millennium House" was a British show-home launched in 1998 to demonstrate how home can be automated with computer-controlled light switches, security, doors and gardens [2].

2.1.1 X10 (Industry Standard)

Pico Electronics in Scotland were trying to build a remote system to control the basic house hold appliances. They discovered the first general purpose home automation network technology called X10 as a follow-on to nine earlier circuits-related projects. The X10 was developed to let transmitter and receiver to work by broadcasting signal via radio frequency bursts representing digital information.[3.]

In year 1978, X10 started selling into stores heading towards electronics enthusiastic. CP-290 unit allowed X10 compatible appliances to communicate with computers. Also gave authority to write their own program for lighting systems, thermostats, etc from PCs who are interested in home automation.

X10 carries undeniable technical limitation for modern home automation networks:

- Supports devices only up to 256
- Lacking Network encryption
- As the number of devices in homes increases more susceptible to electrical interference

2.1.2 Z-Wave

During late 90’s to create a standard for wireless radio frequency (RF) communication for home devices a mesh networking technology called Z-Wave was designed. Products based on Z-Wave are build using a family chips of minimum cost and minimum power, RF transceiver. In order to communicate using a common communication protocol Z-Wave product uses the same chip. [4.]

Home devices like mobile phones, internet-router typically operates at 2.4 GHz which is not the case in Z-Wave devices. The frequency used by Z-Wave works at 868.42 MHz in Europe. There is minimum chances that devices will interfere with other house hold devices.

Greater signal range is another plus side of Z-Wave devices. The Z-Wave devices range determined by various factors, and also the wall presence in certain space. Particularly, Z-Waves devices can hold about 30 meters indoors and 100 meters in the open space. Also there is possible to extend the range of these product by simply adding more Z-Wave devices to the network.

There are various application of Z-Wave devices. Z-Wave can be used as a smart hubs controlling every smart home appliances. Also can be used as Smart lightning system which can help in energy saving with efficient solution. Smart locks such as fingerprint scanners which are highly encrypted mode of access can also be designed by Z-Wave technology.

2.1.3 ZigBee

ZigBee is an IEEE 802.15.4 Standard which give up blueprint for standard personal area network using low-power, small digital radios basically ideal for home automation, data collection in medical gadgets etc. In general ZigBee is used as a bridge between the connected sensor and control system to communicate. It is basically used on short-range communication which ranges up to 100 meters. [5.]

The data transfer rate of ZigBee is about 250kbps. It operates at multiple zone of frequencies. In Europe it operates at 866 MHz, 915 MHz in United States. Since ZigBee uses low power to operates, which in other hand extend the battery life of devices significantly.

Architecture of ZigBee

In general ZigBee consists of the three parts namely coordinators, router and an end device. All devices such as sensor, actuator, etc that are in the system are end devices. The coordinators is the root of the entire network and also acts as a bridge. At least one coordinator is mandatory in the network. It performs miscellaneous data transfer operation, and also stores information. The routers assists coordinator by letting the data to pass back and forth between coordinator to end devices. The design of the system can be star, mesh or however way it is implemented .[5.]

ZigBee Nodes:

- **Coordinators** for establishment and encryption of the network.
- **Routers** is responsible for passing the signal and also extension of the network range.
- **End Devices**

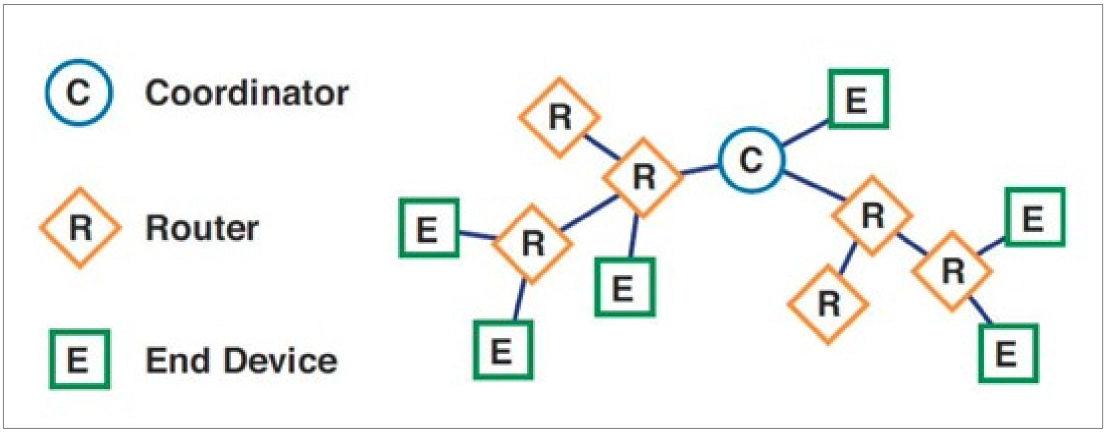


Figure 1: Figure shows ZigBee Nodes[6]

ZigBee In Home Automation

In the home automation industry ZigBee is getting low acknowledgment reason of being open-source. The protocol can be easily reshape by each manufacturer that endorse it. Consequently, devices have struggle communicating in between two different manufacturers. Result leads to the poor and sporadic performance.

However, ZigBee is becoming more prominent as it allows to control wide range of devices with less smart hubs. This can help to reduce the cost of system in other hand. Moreover, ZigBee also works with battery powered devices, which extends its efficiency.

2.2 Raspberry Pi Arrival

A sequence of credit card sized single-board computers designed in the UK by Raspberry Pi Foundation. Target of foundation is to promote programming and computing knowledge for elementary to secondary level students. The devices are also targeted for robotics and electronics hobbyists. It does not come with peripherals such as mouse and keyboard.[7.] In year 2012, Raspberry Pi get launched for public consumers. Two versions of the Raspberry Pi, namely model A and model B were lineup to be manufactured, with B being released first. Basically in model A which exclude Ethernet port and consumption of power was less than the Model B and was tagged $25 price.

2.2.1 Hardware Specification & GPIO pins

During the time period there are many Raspberry Pi Models that appeared in the market . The latest product from the company is Raspberry Pi 4 Model B. Raspberry Pi 3 Model B V1.2 used in the this project.

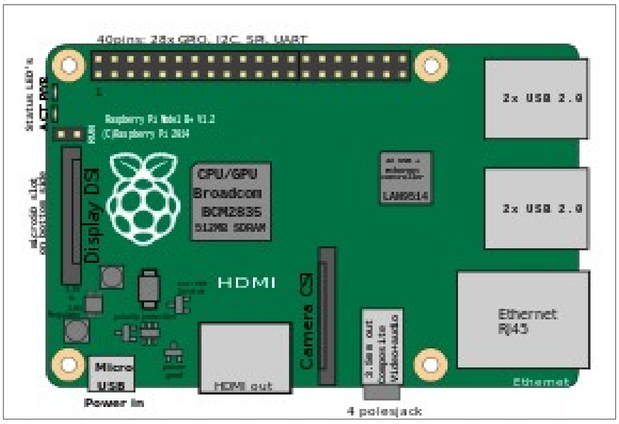


Figure 2: The figure shows Raspberry pi 3 Model B system architect [8]

V1.2 Specification

SoC: BCM2837 Broadcom

CPU: 1.2GHz ARM Cortex

GPU: VideoCore IV Broadcom

RAM: LPDDR2 (900 MHz) 1GB

Networking: 2.4GHz 802.11n wireless

Bluetooth: Bluetooth 4.1, BLE

Storage: microSD

GPIO: 40-pin header

Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI) .[7.]

2.3 Single Board Computers

After the big success of Raspberry Pi, single-board computer offers an ideal solution for anyone looking for a means to learn code, create a gaming and media platform, program robotics, experiment with IoT and more.

With rising running costs of standard PCs, single-board computes become a popular choice for startup wanting to solve their problems and consider for their next inventions.
In the following subheading there are several boards for hobbyist, developers and IT pros to consider for their next invention.

2.3.1 Arduino

Launched in year 2005, the open source Arduino board has inspired a long list of interesting creations. One project of note is Enough Already, a 2011 project that processes TV signals, muting the TV whenever an overexposed celebrity is being talked about.

As averse to Raspberry pi, which is a mini Linux-based computer, Arduino offers an integrated platform with software, hardware and development environment, with processor ranging from 8-bit AVR to 32-bit processors.
It is based on a micro-controller. There are multiple models that generally cost in between 22 Euro to 35 Euro.

2.3.2 Banana Pi

Similar to Raspberry Pi, the Banana Pi offers various single-board computer. And while Banana Pi is compatible with Raspberry Pi, this open source board can also run NetBSD, Android, Ubuntu and Debian to name a few.
A standard Banana Pi is designed with ARM Cortex-A7 Dual-core CPU and Mail400MP2 GPU, making this single-board computer an ideal host for many different types of applications and with Wi-Fi module capabilities, the Banana pi can run as a private data storage server and much more. Depending on specification prices vary in the range 23 Euro to 100 Euros.

2.3.3 BeagleBone Black

BeagleBone is another single board computer which runs an AM335x 1GHz ARM processor and can boot Linux in 10 seconds. The BeagleBone Black can run Debian, Android and Ubuntu with users able to create home security systems, robotics, 3D printers, etc.

The target users for BeagleBone Black are developers and hobbyists seeking to add actuators and other sensors for the abilities like sense, touch and temperature with support also extending to USB, Ethernet, and HDMI modules, making it ideal for IoT projects. It costs around 55 Euro.

2.3.4 Intel NUC

Intel’s NUC, or Next Unit of Computing, arrives in a 4 by 4 inch form factor and is positioned by Intel as a high-performing compact PC alternatives to Raspberry Pi.

The NUC can be found using in home theater, digital signature and shops which served as a gateway to the IoT, according to Intel anyway. The single-board PC might cost around 220 to 350 Euros which is in capsuled with Intel core processors and can runs Windows, Linux, Android and Chrome OS.

2.3.5 BBC Micro:Bit

This dinky 4 by 5 cm board is perfect for amateurs looking to get started in coding. Device is currently used in schools across UK to teach programming technique among youngsters. One can easily create games and animations using simple programming tool such as Scratch, or coding in JavaScript or MicroPython.

Device is prized 20 Euro on Amazon and also device incorporates a LED matrix display, a motion sensor, accelerometer and two buttons

2.3.6 Rock64 Media Board

The Rock64 Media Board is a piece of cake for those are looking to create a media center with just a single-board computer. It has memory four times greater than Raspberry Pi which makes it a decent candidate for anyone looking for designing media center.

The Media Board is capable of playing 4K videos if stored on the computer (lags if streamed over the internet). Moreover the board has faced sour reviews for low standard hardware and also the infamously apps like Netflix struggling to work though the built-in media player.

2.4 Blynk

Blynk is an IoT platform targeted simply to build application in mobile and web to control IoT devices. Without any big effort and struggle, the Blynk can connect many development boards like Arduino, Raspberry Pi, etc for iOS and Android in an instance.

It can easily control hardware remotely, also have capabilities to visualize sensor data, and make able to do IoT devices work in multiple way.

The three major components of Blynk are explained in briefly below:

Blynk App

With the available widgets in the app grants user to develop multiple interfaces with just drop and drag method as per required.

Blynk Server

As name suggested, pledged for all the communications between the smartphone and IoT devices. Blynk server is a bridge connecting the smartphones and IoT devices. User can even run there own private local Blynk server. Being an open-source, and have ability to handle thousands of devices and can even be launched on a Raspberry Pi. [9]

Advantages of having Local Blynk Server

- Maximum security since user will be the one who knows about the server.
- For better stability
- Lower latency
- Maximum privacy

Blynk Libraries

Blynk libraries is responsible for enabling communication with the server and available hardware. It processes all the incoming and out-coming commands. [9]

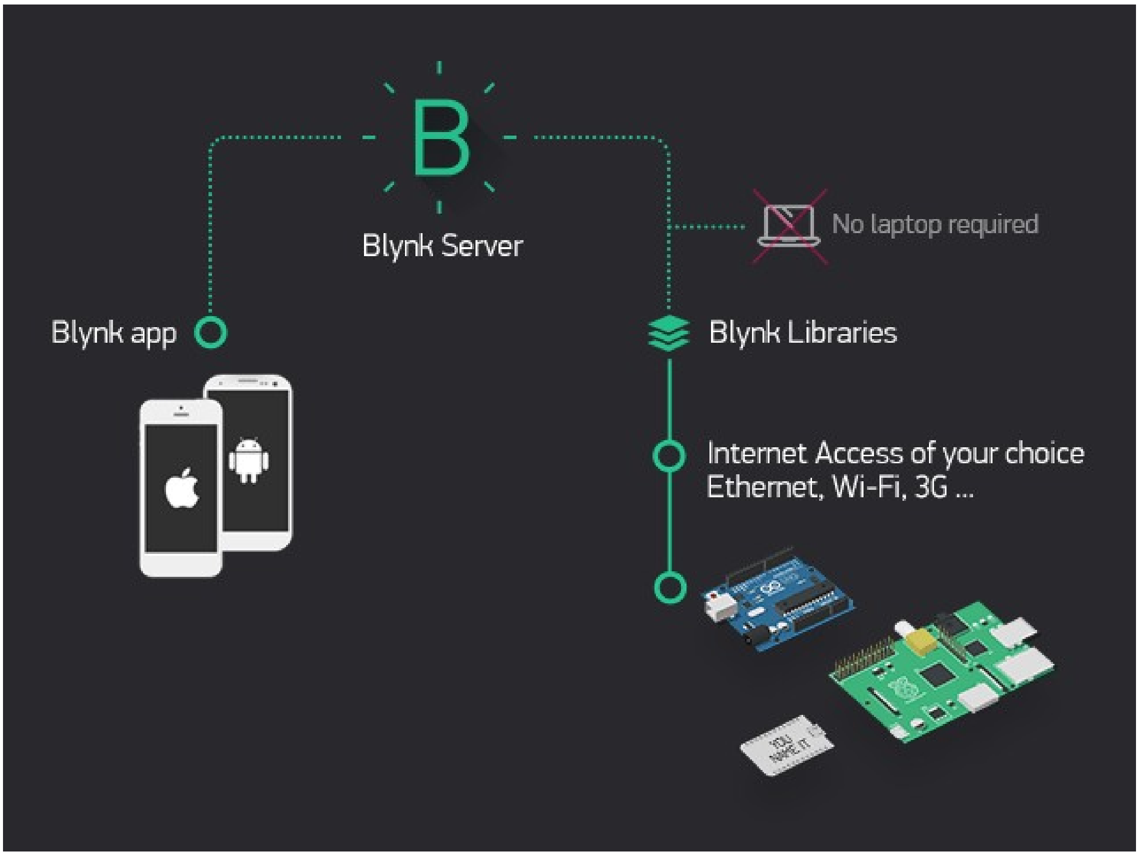


Figure 3: The figure shows the System Architecture in Blynk Platform[9]

In the above figure as we can see Blynk cloud acting as a communication Bridge. When a user press any button on the Blynk app, that message is only able to travel through Blynk cloud, where message or any commands find its way to the destination hardware.

Features

- Supported hardware & devices shares same API & UI
- Bluetooth and BLE, WiFi, Ethernet, USB(Serial), or any other communication protocol helps connecting to the clouds.
- Easy to use Widgets
- Digital pin can be manipulated without additional code.
- Effortless integration and new function can be added using virtual pins
- Easy access of both live and history data through Super Chart widget
- Updated constantly over time

2.5 Different Approaches & Comparison

2.5.1 Home Automation Using Global System for Mobile (G.S.M)

During past decade the smart home concept interest growing significantly among electronics consumers. GSM breakthrough just did not bring the consumers mobile telecommunication, but also brought a milestone for the people who wanted home automation system. Using GSM in Home Automation bought standalone, novel, reduced-cost and adjustable system. [10.]

The system entirely depends on a 8-bit µC called PIC. The Database equipment built around the µC and a GSM controller aid the brain of the system which is then connected to a ZigBee Transceiver and every node present inside the home is then able to communicates. The system requires mobile phone as a GSM controller to communicate. Mobile phone sends commands through SMS to the Controller, which then interprets the command and then activates the required ‘switch’ to control the desired device.[5]

The connected electrical items is possible to control from anywhere in the world with in the GSM mobile coverage. There might be additional roaming charge nevertheless using GSM network the security is highly encrypted and other people cannot have any access or monitor the received and sent information .[9;10]

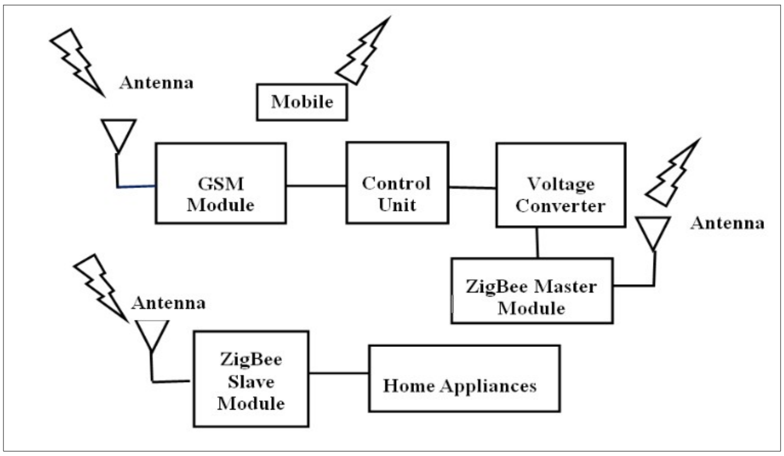


Figure 4: The above figure shows the Architecture of GSM Home automation[11]

2.5.2 Home Automation Using ZigBee

ZigBee Home Automation is not only dominant comprehensive standard helping to create smart home system but also provides security, energy saving, cost reducing and comfort for the consumers. ZigBee is allowing consumers to purchase the ZigBee certified product with confidence.

Current ZigBee V3.0, data being shot around the mesh is quiet secure which have 128 bit symmetric encryption. Nevertheless, if any one are really focused on excellent security, many consumers complaints about ZigBee susceptibility around the way it handles encryption key.

ZigBee works in different frequencies range in different part of the world. The data transfer rate is around 250kbps, which is pretty good. Since most of the devices notably Wi-Fi enabled devices uses 2.4GHz, which is pretty much everything works under that radar– that means possibility of interference.

Features

- **Easy install:** Idea for remodel or new construction
- **Internet Connection:** Device can be controlled from anywhere around the world.
- **Power Control:** Turn on/off devices remotely by monitoring used power.
- **Security:** Not best but good network security encryption

ZigBee Alliance And ZigBee Devices

Anyone can easily think the type of devices used by ZigBee, And now the manufacturer of ZigBee have already initiated an alliance. There are almost 400 members of the ZigBee Alliance, and they stacked up around 2,500 devices between them.

A half a billion ZigBee chip-sets have been sold already to date and that ZigBee Alliance technologies is targeted for 3.8 billion IEEE 802.115.4 units expected to be manufactured by 2023.[11;12]

2.5.3 Home Automation Using Java

A simple home automation concept where the devices can be controlled and monitored using World Wide Web(WWW).

A server installed in PC at home integrated by a standalone embedded system. The status of connected home appliances are passed to the server by the help of i/o ports of the automation system. The system is based on the combination of Interactive C, Java Server Pages & Beans. The Appliances can be controlled through web browser from any where around the world. Also the device can be controlled and monitored locally through the designed embedded board. [15.]

3 Components For System Design

3.1 Overview Of Smart Home System

The target of the project is to develop the smart home automation system where home-owner has a control over the household devices through a smartphone. Meanwhile, the system can monitor the room temperature and also the dedicated mechanical relays in the system where any household device can be mounted and can be turn on and off remotely with just a blink of an eye.

This section describes the system and its components where the desired final product and the technologies used for the software part of the project.

3.2 System Description

The system uses Blynk platform which consists of application that is already installed in smartphone and also connects the whole system to Blynk server through the internet to Raspberry pi. Generally pi interacts with Blynk server to perform controlling and monitoring function by software program mainly written in JavaScript and partially in python and C/C++ programming languages. DHT 11 sensor and relays are embedded to Raspberry Pi’s GPIO (General Purpose Input Output) pins.

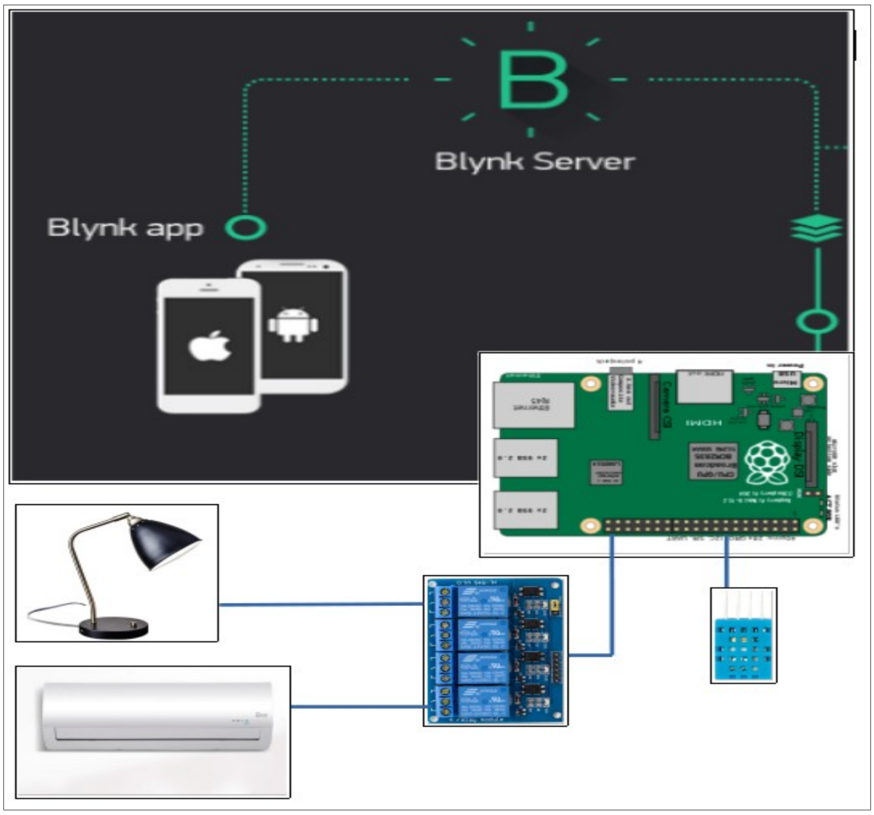


Figure 5: The figure shows peripherals connected to the Blynk System

3.3 Designed Hardware

3.3.1 Raspberry Pi

Raspberry Pi is a single-board credit-size computer with low-cost that plugs into a monitor. Mouse and keyboard can be added. In a nutshell, it is capable of performing any basic stuff in regular PCs like playing games, watching and streaming videos, programming, net surfing etc. Most of the common terms for pi are already explained in section two.

3.3.2 Relay Module

An electromagnetic switch which get activated by a relative tiny electric current enough to turn off or on larger electric current .

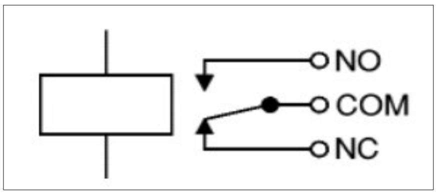


Figure 6: Physical pins of relays.

In this project, four Channel 5V relay module is used. 15-20mA driver current is supplies to every channel. Many appliances can be controlled even though the devices are equipped with large amount of current. The whole relay module can be controlled by any µC because of its standard interface.

Pin Description

Input

VCC: +Ve supply voltage

GND: Ground

IN1—IN4: Relay control port

Output:

DC 30V/10A, AC 250V/10A (load)

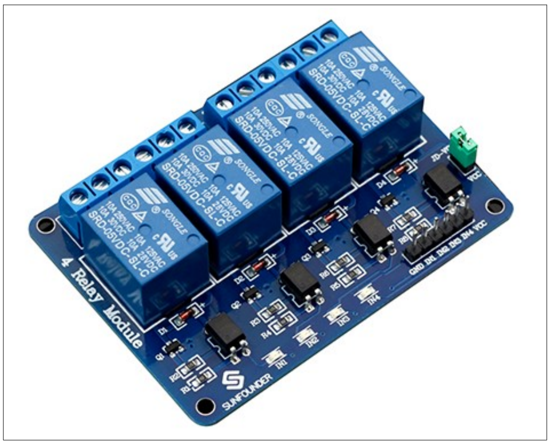


Figure 7: The figure shows 4-Channel 5V Relay Module

Features:

- Dimension(l*w*h): 75mm* 55mm * 19.3mm
- Total Wt: 61g
- For the easy installation of the board in any surface, there are four holes on each edges of rectangular PCB. The diameter of hole is measured 3.1mm.
- There is decent anti-interference also with optical coupling isolation
- When indicator on denotes at low level(Closed state), indicator off denotes released at high level.
- Power source of relay is JD_VCC where VCC itself is a source of power
- Limited output of the relay: AC 250V/10A DC 30V/10A. [16.]

Connection of Relay in Raspberry pi

IN1 – GpiO12
IN2 – GpiO21
IN3 – GpiO20
IN4 – GpiO16

JD-VCC – 5V
VCC – 3.3V
GND – GND

3.3.3 DHT11 Sensor

A DHT11 is simple, cost-reduced, and handy sensor used for measuring temperature and humidity. Powered by capacitive humidity and also to measure the surrounding, thermistor is used. A very easy to use but needs a bit of timing to grab the data. The backside of the sensor is only it able to get new data in every two second, which makes sensor reading uo to two second old when using sensor libraries.

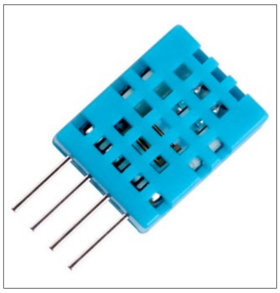


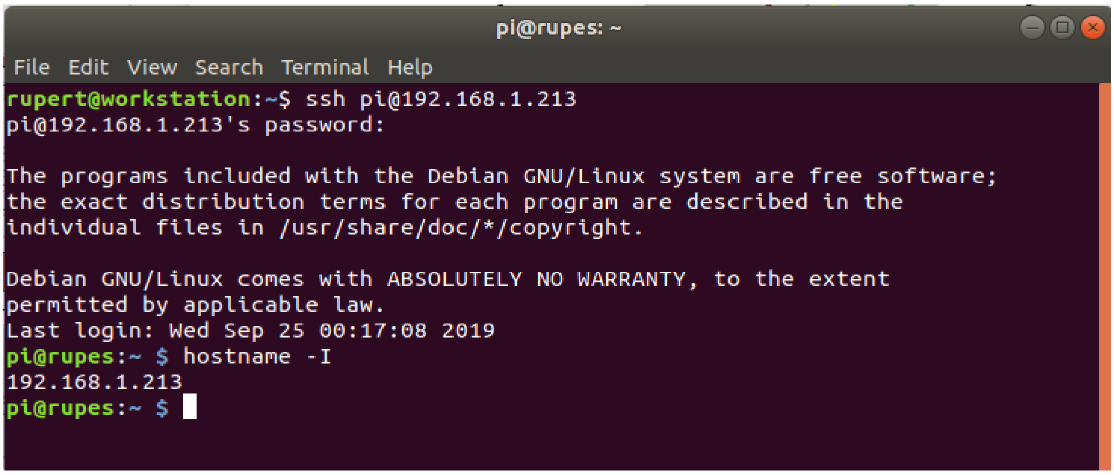
Figure 8: The figure 8 shows DHT 11 sensor

Technical Details

- Affordable cost
- 3 to 5V operable range
- During conversion Max current 2.5mA
- 5% accuracy, Good for 20-80% humidity
- Ideal for reading temperature range in between 0-50°C ±2°C accuracy.
- Limited to 1 Hz sampling rate
- Dimensions: (15.5* 12 * 5.5) mm
- 4 pins with 0.1” spacing [17]

3.4 Software Designed

Raspberry Pi being the main hub of the system is mainly programmed with JavaScript with NodeJs build engine. For testing some sensors, and in addition understanding Pi in more detail, many other programming languages such as Python, C/C++, etc were used.
Raspberry Pi is connected to local computer through ssh connection. Since Debian is installed on Raspberry Pi and the local host computer itself has a Ubuntu, getting things work was fun and easier. No any external peripherals such as monitor, mouse and keyboard were connected to the Raspberry Pi.



```
rupert@workstation:~$ ssh pi@192.168.1.213
pi@192.168.1.213's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Sep 25 00:17:08 2019
pi@rupes:~ $ hostname -I
192.168.1.213
pi@rupes:~ $ 
```

Figure 9: Above figure shows raspberry pi connected to local computer through SSH connection

3.4.1 Blynk Application On Smartphone (over Blynk Server)

Blynk is a simple software for developer that support Android and iOS to control the development board like node MCU, Raspberry Pi, Arduino to name a few. User have to create the account and each user application has username and password to log in to dashboard.

As discussed earlier in section 2.4 Blynk have three fundamental parts.

- Blynk App
- Blynk Server
- Blynk Libraries

Blynk server is an open-source Netty based Java server which is responsible for forwarding messages between various µC and mobile applications or any other development board.
Blynk libraries can perform basic IO(input-output) operation out of the box.[9]

Network Address Translation (NAT)

In NAT, the source and destination IP addresses are changed. NAT translate a set of IP Addresses to another set of IP addresses. The process is usually done by router or firewalls. NAT helps preserve the limited amount of IPv4 public IP addresses because there are only 4,294,967,296 public IPv4 addresses available. That amount is not enough for every device in the world

Working principle of NAT:

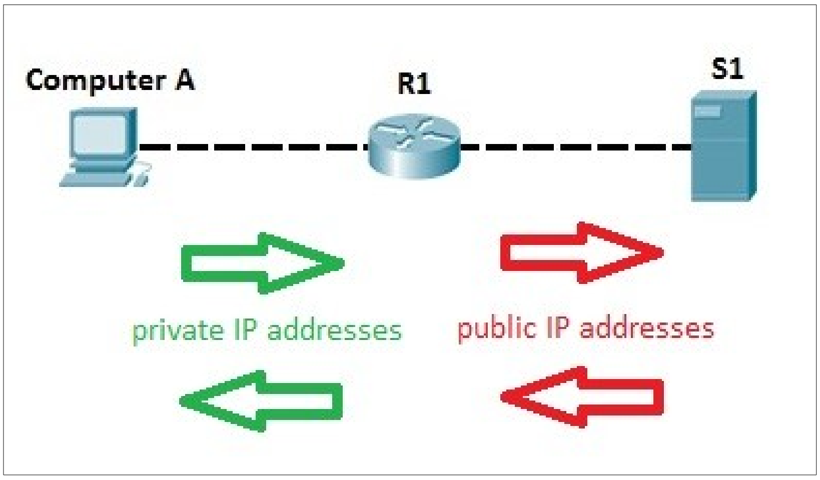


Figure 10: The figure shows the Working principle of NAT

If Computer A wants to request any web pages from server S1. In general, the Computer in home network always used Private IP addresses. Also to have access over the internet the Computer must have public IP. So the router changes the Private IP to the Public IP address and sends the packets to server S1. After Server S1 receives the packets and replies to router R1. Again router receives the packets from Server 1 and now again the router translate the public IP to the private IP address of Computer A and sends the requested packets from the server to the destination.[18.]

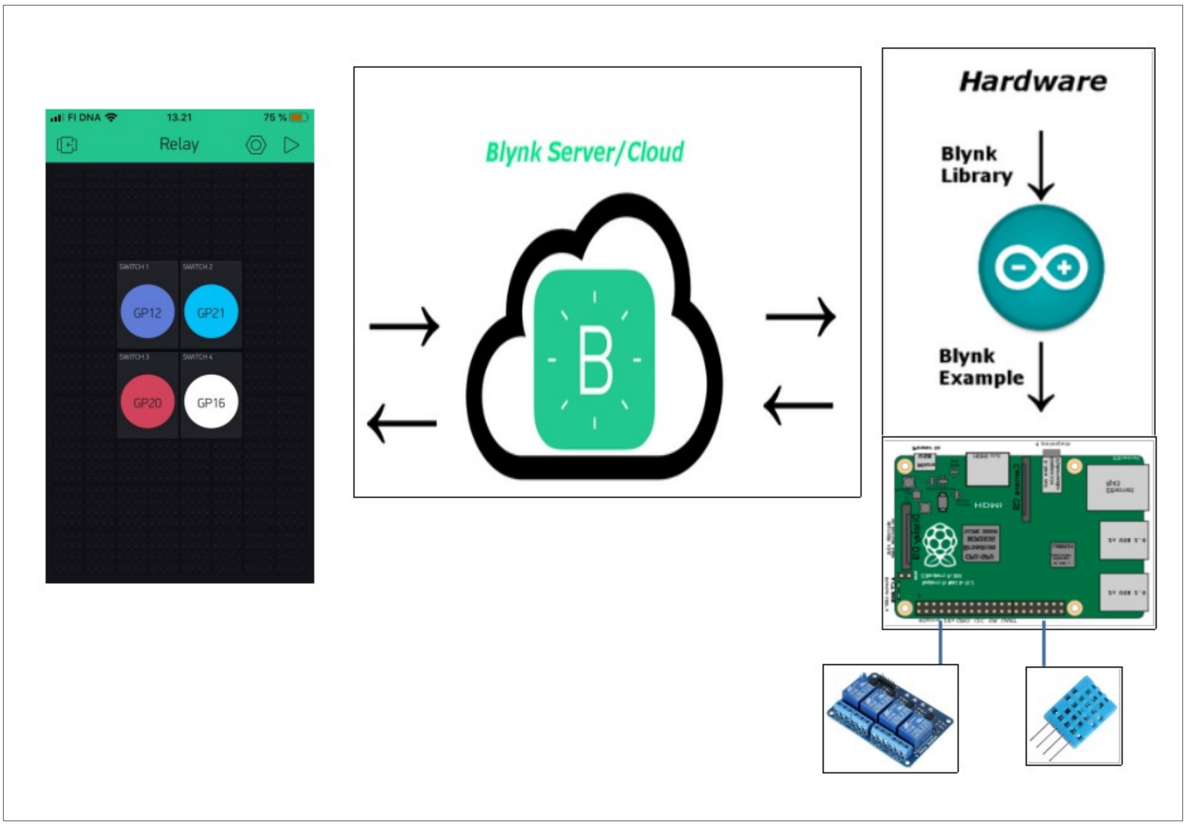


Figure 11: Above figure shows a Blynk System Architecture

Blynk Review

The main task of Blynk Cloud is to exchange messages between the connected hardware and application in the system. Blynk has its own Binary Protocol for exchanging communication between hardware and apps.

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

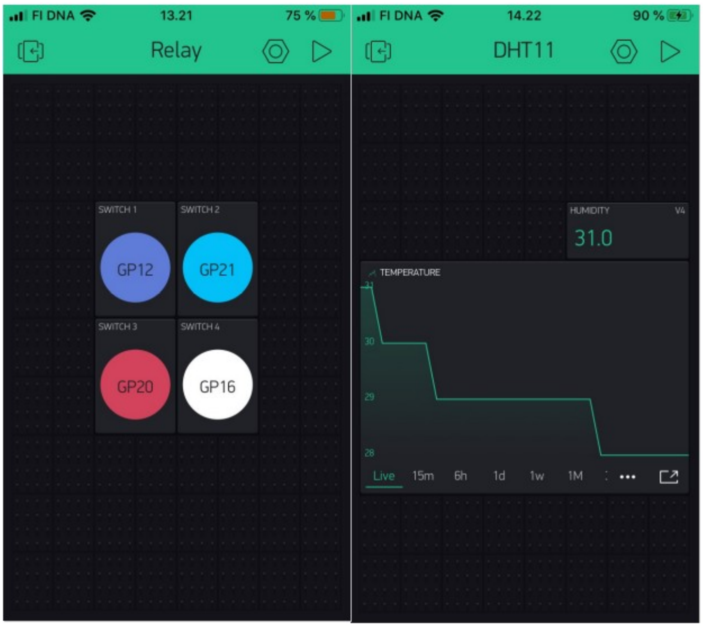


Figure 12: The dashboard of Blynk application used to control/monitor devices in Home Automation

Blynk system is smart enough to take all the complicated parts of the building a slick interface on a phone out of the equation. Also it handles entire internet portion of IoT device, meaning no need to figure out any networking or communication stuff.

In general, the Blynk project (i.e drag and drop widgets) which are designed on phone has a unique identifier assigned to it. When the internet connected device (pi in our case) the Blynk program running on the device connects to the Blynk server using the same unique identifier. The server let's the app on the phone know it is connected, and creates the connection.

4 Implementation Of Home Automation System

4.1 Overview

In this section, there are some instruction for implementing the home automation system. Firstly, installing the required software which is essential to build the system such as Blynk Libraries, Blynk server which is intermediary for communicating Blynk Clouds and hardware etc. Secondly, installing hardware components and required sensor libraries to the board itself. After all these steps, the simple home automation system will be ready to demonstrate with just a start of code. Finally, local Blynk server is created for more security and encryption.

4.2 Installing Required Software

4.2.1 Node Package Manager (npm) installation

Before starting anything, first it is needed to install npm, stands for Node Package Manager on our Board Pi. Basically npm is an online repository and command-line utility for interacting with repository that aids in dependency management, version management and package installation.[19]

Here are the list of commands performed step wise in the terminal of our Raspberry Pi.

At first Node.js need to be installed, before updating Node.js any old version needs to be removed:

```
sudo apt purge node nodejs node.js -y
sudo apt autoremove
```

Automatics Node.js installation

Adding repositories:

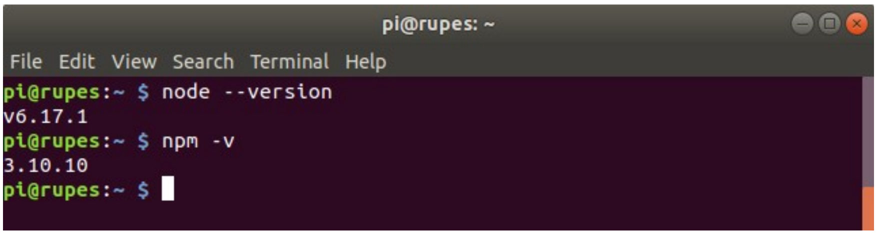
```
curl -sL https:/deb.nodesource.com/setup_6.x | sudo -E bash -
```

Install Node.js

```
sudo apt update && sudo apt upgrade
sudo apt install build-essential nodejs -y [9]
```

Manually install can also be performed if somethings would have gone wrong with automatic installation. In our case auto installation worked without any errors.

Checking Node.js and npm installation



```
pi@rupes:~ $ node --version
v6.17.1
pi@rupes:~ $ npm -v
3.10.10
pi@rupes:~ $ 
```

Figure 13: Figure shows installed Node versions

Installing Blynk Libraries

```
sudo npm install blynk-library -g
sudo npm install onoff -g
```

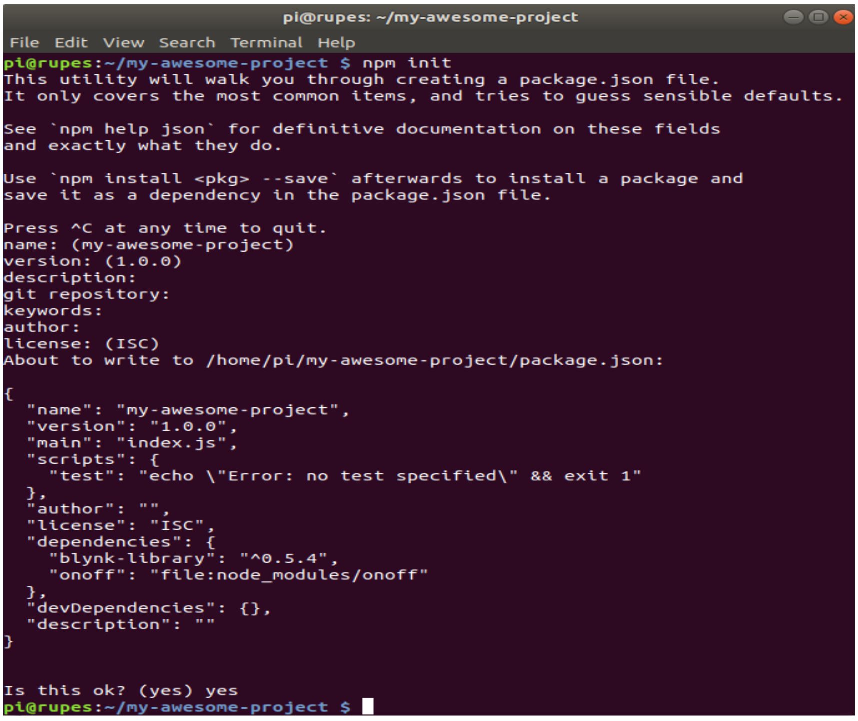
Running default Blynk client (replacing authentic code)

```
export PATH=$PATH:/opt/nodejs/bin/
unset NODE_PATH
blynk-client “OurAuthToken”
```

Creating a new Node.js project with Blynk

```
mkdir my-awesome-project
cd my-awesome-project
npm init
```

This prompts for general information about the project and will create a package.json file which is project description.


pi@rupes: ~/my-awesome-project
File Edit View Search Terminal Help

```
pi@rupes:~/my-awesome-project $ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (my-awesome-project)
version: (1.0.0)
description:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/pi/my-awesome-project/package.json:

{
  "name": "my-awesome-project",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "blynk-library": "^0.5.4",
    "onoff": "file:node_modules/onoff"
  },
  "devDependencies": {},
  "description": ""
}


Is this ok? (yes) yes
pi@rupes:~/my-awesome-project $ 
```

Figure 14: The figure shows the project description of package.json file

Controlling Relay and Sending Sensor Data To The Dashboard.

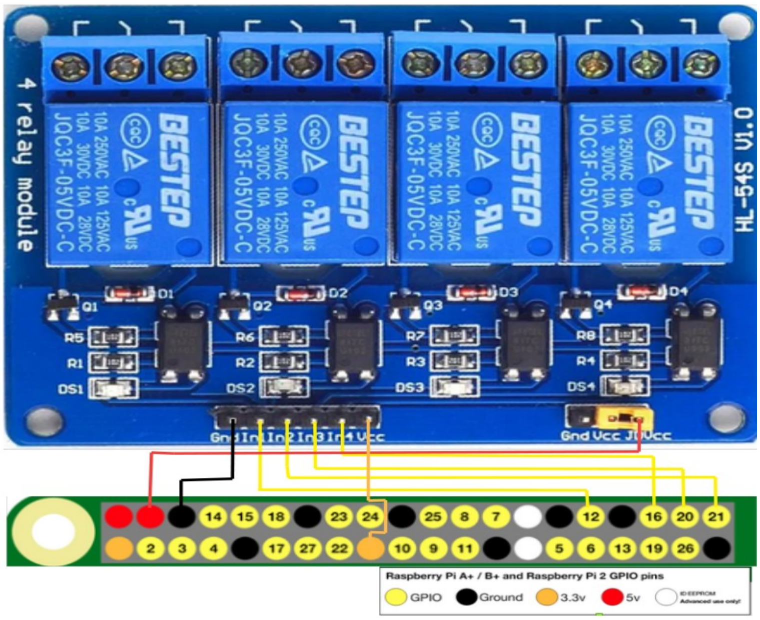
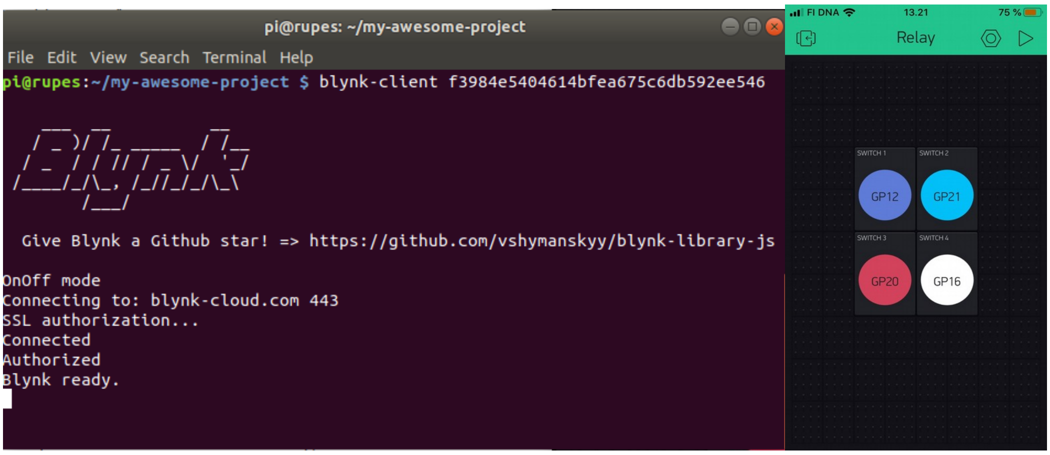


Figure 15 : The figure shows GPIO connection of 4-Channel Relay to Pi

Relay is controlled in the system with digital wire in Blynk app. Advantage of using digital wire over virtual wire in Blynk app is, that no extra code required to write in order to control the pins. However, “onoff” module should be installed on the board. The following step carried out to create widgets in Blynk app to control our 4-Channel relay.

1. New dashboard was created of device type Raspberry Pi 3 B and “AuthToken” was generated for the project.
2. Four buttons were added and named Switch 1, 2, 3 and 4 respectively.
3. Switches 1,2,3 and 4 were connected to digital pins of Pi whose GPIO are 12,21,20 and 16 respectively.
4. Finally buttons were enabled to switch mode.



```
pi@rupes:~/my-awesome-project $ blynk-client f3984e5404614bfea675c6db592ee546

    ___  __          __
   / _ )/ /_ _____  / /__
  / _  / / // / _ \/  '_/
 /____/_/\_, /_//_/_/\_\
        /___/

  Give Blynk a Github star! => https://github.com/vshymanskyy/blynk-library-js

OnOff mode
Connecting to: blynk-cloud.com 443
SSL authorization...
Connected
Authorized
Blynk ready.
```

Figure 16.1: Blynk app controlling 4-channel relay in the Home automation system

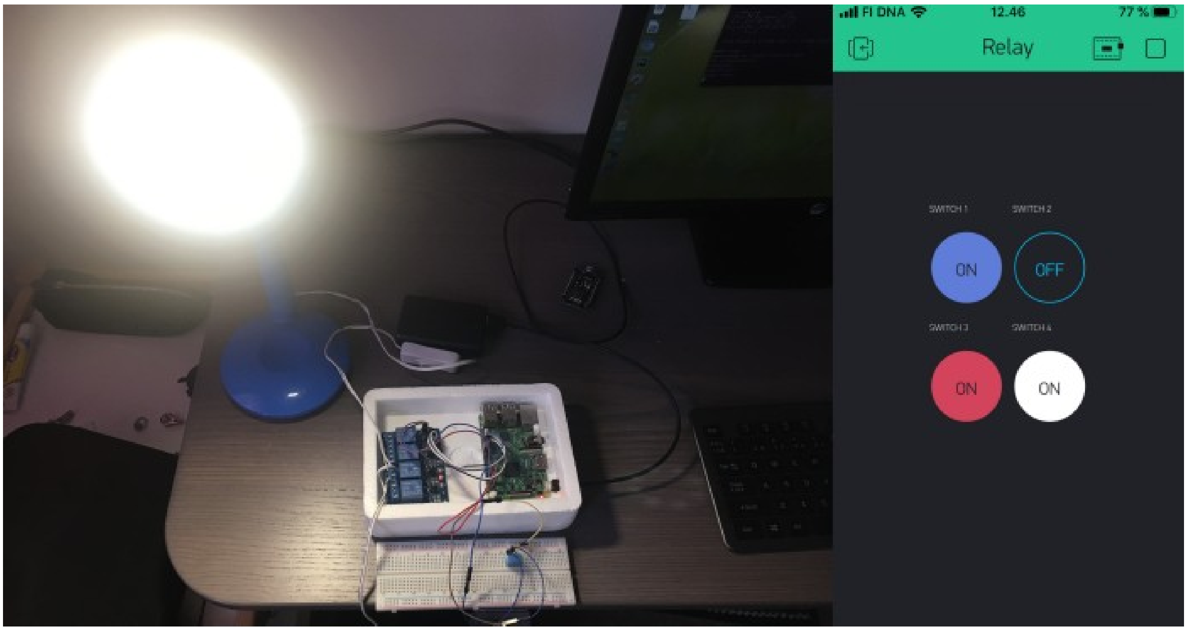


Figure 16.2: Blynk app controlling 4-channel relay in the Home automation system

After successfully controlling the 4-Channel relay in the system the DHT 11 sensor is connected then to the GPIO pins of the Pi to send data to the dashboard of Blynk app. Some libraries need to be installed beforehand otherwise board will unable to recognize the sensor and which was exactly the same in this project.

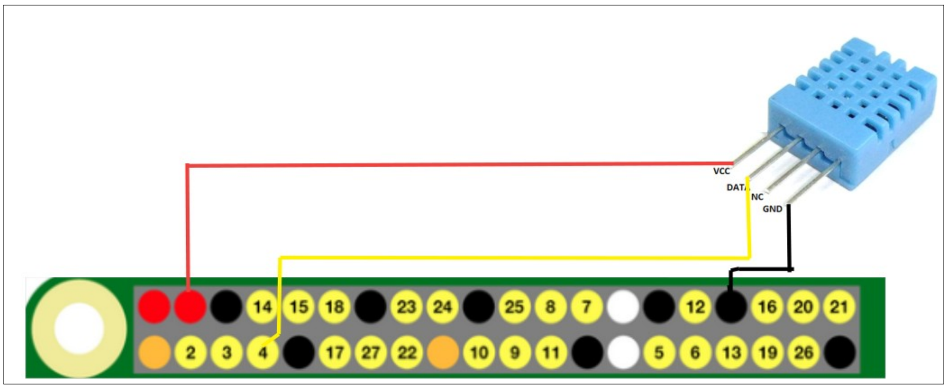


Figure 17: GPIO connections of DHT 11 sensor to the Pi.

Installing DTH sensor libraries

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.46.tar.gz
tar zxvf bcm2835-1.46.tar.gz
cd bcm2835-1.46
```

```
./configure
make
sudo make check
sudo make install
sudo npm install -g node-dth-sensor
```

In order to build the module standard node-gyp used. To be certain for node and node -gyp as well as the Broadcom libraries installed or not to build the projects. Otherwise DHT11 sensor will not respond the command which was in our case.

The following commands performed step wise to get work the module and sensor libraries together

1. Firstly, node-gyp installed

```
sudo npm install -g node -gyp
```

2. The generated configuration files

```
node -gyp configure
```

3. The built component

```
node -gyp build
```

4. Tracing & Debugging verbose output the module enabled by specifing the –dth_verbose=true

```
npm install node-dht-sensor --dht_verbose=true
```

Now the dth-sensor libraries finally got installed in the board then following steps were performed to create widgets on Blynk mobile app to display our sensor values on dashboard of Blynk app.

1. New dashboard created of type Generic and “Auth Token” was send
2. Super Chart widget added and binned it to V3

   On Y-scaling set to Auto which is for temperature
3. Display value widget added which is for Humidity and binned it to V4

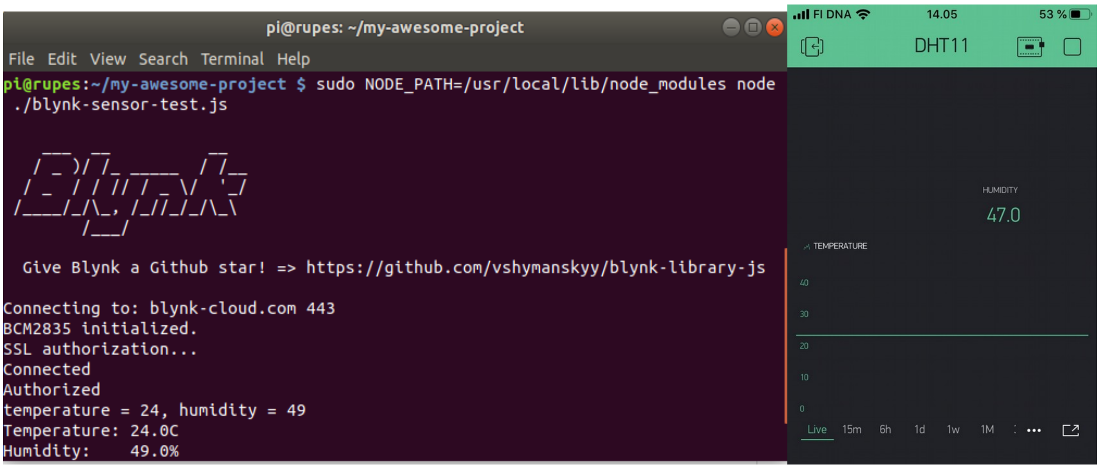
A file named blynk-sensor-test.js is created which is actually a code for displaying sensors values in the dashboard of the Blynk app.

Before running the code following lines of code was performed

```
sudo NODE_PATH=/usr/local/lib/node_modules node ./blynk-sensor-test.js
```

Node_PATH=… is set in order to locate global modules otherwise Node.js may fail.

DHT 11 sensor read the data and send it back to Blynk app. When reading the data Humidity was 47% and temperature was 24 deg Celsius.



pi@rupes: ~/my-awesome-project

```
pi@rupes:~/my-awesome-project $ sudo NODE_PATH=/usr/local/lib/node_modules node
 ./blynk-sensor-test.js

    ___  __          __
   / _ )/ /_ _____  / /__
  / _  / / // / _ \/  '_/
 /____/_/\_, /_//_/_/\_\
        /___/

  Give Blynk a Github star! => https://github.com/vshymanskyy/blynk-library-js

Connecting to: blynk-cloud.com 443
BCM2835 initialized.
SSL authorization...
Connected
Authorized
temperature = 24, humidity = 49
Temperature: 24.0C
Humidity:    49.0%
```

Figure 18: Displaying Humidity and temperature read by sensor in Blynk app

The source code for the DHT11 sensor for displaying temperature and humidity can be found in Appendix 1.

4.2.2 Building Local Server For Better Security and Encryption

The Blynk local server is ready to run out of the box, as soon as the installation complete. Some of the configuration were done creating following files:

`server.properties:` which allows to configure all server settings

`mail.properties:` which allows to configure the server to send emails

Configuring the server(server.properties file)

At first, the server.properties file is created. This file helps to modify the communication port as per required. This is especially important when installed a home automation server (or other software) that uses the same ports as the local Blynk server. For example by default the port http (and web socket) is 8080.

Blynk has a management interface accessible from a web browser. By default this interface is accessible only from the computer the raspberry pi on which the server is installed. To be able to access it from any computer on the local network, the **allowed.admisintrator.ips** parameter is changed. At the end the file like this

```
allowed.administrator.ips = 0.0.0.0/0
```

All the files are saved in the same folder as the server.

Configuring the email account(mail.properties file)

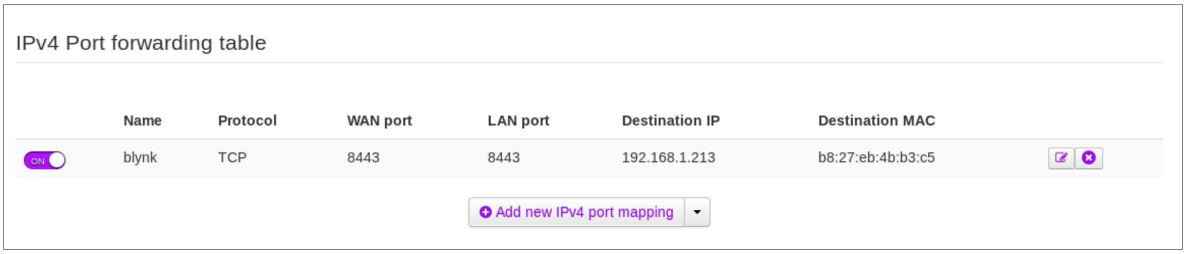
Blynk always sends an email containing the token named “AuthToken” as soon as new Project is created. It is also possible to return a Token at any time by email from mobile application. A new configuration file named mail.properties was created and saved in the same folder as the server.

Following command executed to run the server

```
java -jar server-0.23.0.jar -dataFolder /home/pi/Blynk
```

Making the Blynk server accessible from the internet

For accessing our private Blynk server, we configured internet box to make the Raspberry pi available from the internet. Since our IP is private, NAT rule is added in NAT setting box to return calls from the external port 8443 to port 8443 of the Raspberry pi. NAT is explained briefly in section 3.4.1.



IPv4 Port forwarding table

	Name	Protocol	WAN port	LAN port	Destination IP	Destination MAC	
ON	blynk	TCP	8443	8443	192.168.1.213	b8:27:eb:4b:b3:c5	

Figure 19: NAT rule performed by adding port to the router

5 Conclusion

The target of the project was to design small and smart home automation system which can control the peripherals of home such as Lamp, Air-Conditioner and many other with just a simple mobile app called Blynk, and to read the humidity and temperature from DHT 11 sensor and sending the live data to Blynk app. These data are sent to Raspberry pi through Blynk server. Finally, our own local Blynk server was developed for more security and data encryption.

This projects mainly targeted on the application of IoT with sensor, Raspberry pi and Blynk application. The projects achieved desired results and can be said to be done successfully. In addition the project covers the main aspect of so called Ideal Smart Home Automation, for instance controlling light, measuring surrounding temperature, etc. The powerful relay even let us to perform with high voltage but with strong precaution and safety.

Nevertheless, this was an introduction to the smart home system, additional components need to make even more sophisticated and dedicated Smart Home. Already thousands of Smart Home product already launched including Water Boiler to Laundry Machine. It is probably be possible to connect these products to the system which would be an application.

6 References

[1] S. Arjun, “IoT Smart Home Automation,” 2019. [Online]. Available: https://dzone.com/articles/iot-smart-home-automation-and-its-future-predictio. [Accessed: 11-May-2019].

[2] D. Hendricks, “The History Of Smart Homes,” *M2M Features News*, 2014. [Online]. Available: https://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm. [Accessed: 10-May-2019].

[3] Wikipedia, “X10(industry standard),” 2019. [Online]. Available: https://en.wikipedia.org/wiki/X10_(industry_standard). [Accessed: 23-Jun-2019].

[4] Rajiv, “Application of Z-Wave technology,” 2018. [Online]. Available: https://www.rfpage.com/applications-of-z-wave-technology/. [Accessed: 16-May-2019].

[5] Iplocation, “How ZigBee Work.” [Online]. Available: https://www.iplocation.net/zigbee. [Accessed: 25-May-2019].

[6] doxygen, “ZigBee,” 2015. [Online]. Available: http://thomarmax.github.io/QtXBee/doc/pre_alpha/xbee_zigbee_basic_concepts.html. [Accessed: 15-May-2019].

[7] V. S. ; K. L. G. ; S. N. ; A. A. ; L. S. Kumar, “Globally accessible machine automation using Raspberry pi based on IoT,” *IEEE*, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7275764. [Accessed: 17-Jun-2019].

[8] Wikipedia, “Hardware Architecture.” Wikipedia.

[9] Blynk, “Blynk,” 2019. [Online]. Available: http://docs.blynk.cc/. [Accessed: 23-May-2019].

[10] M. . B. and U. . A. B.I Ahmad, F.Yakubu, “Remote Home Management,” *WCSIT*, pp. 144–147, 2011.

[11] M.S Khandare and A. Mahajan, “Mobile Monitoring System for Smart Home,” in *International ConferenceEmerging tends in Engineering and Technology (ICETET)*, 2010, pp. 848–852.

[12] G. G. and G. Peng, “No Title,” in *The Survey of GSM wireless Communication System*, 2010, pp. 121–124.

[13] J. Stables, “Zigbee explained,” *the ambient*, 2018. [Online]. Available: https://www.the-ambient.com/guides/zigbee-devices-complete-guide-277. [Accessed: 23-May-2019].

[14] Z. Alliance, “Zigbee Home Automation,” 2019. [Online]. Available: https://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbeehomeautomation/. [Accessed: 23-May-2019].

[15] A.-A. & M. AL-Rousan, *IEEE Transaction on Consumer Electronics*. 2004.

[16] S. Founder, “No Title.” [Online]. Available: http://wiki.sunfounder.cc/index.php?title=4_Channel_5V_Relay_Module&oldid=399. [Accessed: 15-May-2019].

[17] DigiKey, “Components 101.” [Online]. Available: https://components101.com/dht11-temperature-sensor. [Accessed: 16-May-2019].

[18] Study-ccna, “NAT,” 2019. [Online]. Available: https://study-ccna.com/what-is-nat/. [Accessed: 21-May-2019].

[19] N. j. Foundation, “What is npm?” [Online]. Available: https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/. [Accessed: 06-Jun-2019].

Appendices

Appendix 1: Source Code (DHT 11)

```
var blynkLib = require('blynk-library');
var sensorLib = require('node-dht-sensor');

var AUTH = '***********************************';

// Setup Blynk
var blynk = new blynkLib.Blynk(AUTH);

// Setup sensor, exit if failed
var sensorType = 11; // 11 for DHT11, 22 for DHT22 and AM2302
var sensorpin  = 4;  // The GpiO pin number for sensor signal
if (!sensorLib.initialize(sensorType, sensorpin)) {
    console.warn('Failed to initialize sensor');
    process.exit(1);
}

// Automatically update sensor value every 2 seconds
setInterval(function() {
    var readout = sensorLib.read();
    blynk.virtualWrite(3, readout.temperature.toFixed(1));
    blynk.virtualWrite(4, readout.humidity.toFixed(1));

    console.log('Temperature:', readout.temperature.toFixed(1) + 'C');
    console.log('Humidity:    ', readout.humidity.toFixed(1)    + '%');
}, 2000);
```

Appendix 2: Enabling Server auto restart on Unix-like System

To enable server auto restart find /etc/rc.local file and add:

```
java -jar /home/pi/server-0.41.11-java8.jar -dataFolder /home/pi/Blynk &
```

Or if approach doesn’t work, executed

```
crontab -e
```

And adding following line

```
@reboot java -jar /home/pi/server-0.41.11-java8.jar -dataFolder
/home/pi/Blynk &
```