JUNE 2018 M.Sc. in Electrical and Electronics Engineering MOHAMAD MUDDASSIR GHOORUN

UNIVERSITY OF GAZİANTEP

GRADUATE SCHOOL OF

NATURAL & APPLIED SCIENCES

INTERNET OF THINGS BASED HYBRID HOME AUTOMATION SYSTEM CONTROL VIA ANDROID APP

M.Sc. THESIS

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

BY

MOHAMAD MUDDASSIR GHOORUN

JUNE 2018

Internet of Things Based Hybrid Home Automation System Control Via Android App

M.Sc. Thesis

In

Electrical and Electronics Engineering

University of Gaziantep

Supervisor

Prof. Dr. Ergun ERÇELEBI

By

Mohamad Muddassir GHOORUN

June 2018

© 2018 [Mohamad Muddassir GHOORUN]

REPUBLIC OF TURKEY
UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Name of the thesis :Internet of things (IoT) based home automation system control via Android App

Name of the student : Mohamad Muddassir GHOORUN

Exam date : 20/06/2018

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. A. Necmeddin YAZICI

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ergun ERÇELEBİ

Head of Department

This is to certify that we have read this thesis and that in our consensus opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ergun ERÇELEBİ

Supervisor

Examining Committee Members	Signature
Prof. Dr. Ergun ERÇELEBİ	.....
Prof. Dr. İlyas EKER	.....
Associate Prof. Dr. Ahmet Mete VURAL	.....

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Mohamad Muddassir Ghoorun

ABSTRACT

INTERNET OF THINGS BASED HYBRID HOME AUTOMATION SYSTEM CONTROL VIA ANDROID APP

GHOORUN, Mohamad Muddassir
M.Sc. in Electrical and Electronics Eng.
Supervisor: Prof. Dr. Ergun ERÇELEBİ
June 2018
58 Pages

The third wave of the internet also known as the "internet of everything" is a concept whereby every part of our lives depends on an internet connection. The proliferation of internet of things (IoT) is having massive repercussions on consumer products, businesses, and our way of life. Multinational companies like Apple, Google and Samsung are all currently developing new platforms and solutions for this new boom in the industry. Nearly everyone has a smart phone and as portable as they are, they are hence ideal for controlling home appliances and viewing surveillance cameras. This thesis demonstrates how the integration of an embedded Linux system and low cost Wi-Fi chips with full TCP/IP stack with microcontroller units interacting with a custom made Android app is used to give real time data of physically connected hardware. Live streaming of camera, control of electronic appliances as well as real time measurements of atmospheric parameters can all be seen and controlled through the internet via a tailor made Android mobile application. The electronic miniaturization of microcontroller technology also allows us to propose a scalable, personalizable and cost efficient hybrid home monitoring and control system.

Key Words: raspberry pi 3, internet of things (iot), esp8266, home automation, android

ÖZET

ANDROID APP ARACILIĞIYLA NESNELERİN İNTERNETİ TABANLI HİBRİT EV OTOMASYON SİSTEMİ KONTROLÜ

GHOORUN, Mohamad Muddassir
Yüksek Lisans Tezi, Elektrik-Elektronik Müh. Bölümü
Tez Yöneticisi: Prof. Dr. Ergun ERÇELEBİ
Haziran 2018
58 sayfa

"Her şeyin interneti" olarak da bilinen internetin üçüncü dalgası, hayatımızın her parçasının internet bağlantısına bağlı olduğu bir kavramdır. Nesnelerin interneti çoğalması (IoT), tüketici ürünleri, işletmeler ve yaşam tarzımız üzerinde büyük yankı uyandırıyor. Apple, Google ve Samsung gibi çok uluslu şirketler, şu anda sektördeki bu yeni patlama için yeni platformlar ve çözümler geliştirmektedir. Neredeyse herkesin bir akıllı telefonu var ve oldukları kadar taşınabilirler, böylece ev aletlerini kontrol etmek ve gözetleme kameralarını izlemek için idealdirler. Bu tez, gömülü bir Linux sistemi ile düşük maliyetli Wi-Fi yongalarının, özel bir Android uygulamasıyla etkileşimli mikro denetleyici birimleriyle tam TCP / IP yığını ile entegrasyonunun, fiziksel olarak bağlı donanımın gerçek zamanlı verilerini vermek için nasıl kullanıldığını göstermektedir. Kameranın canlı akışı, elektronik cihazların kontrolü ve atmosferik parametrelerin gerçek zamanlı ölçümleri, kişiselleştirilmiş bir Android mobil uygulaması aracılığıyla internet üzerinden görülebilir ve kontrol edilebilir. Mikrodenetleyici teknolojisinin elektronik minyatürleştirilmesi, ölçeklenebilir, kişiselleştirilebilir ve düşük maliyetli hibrid ev izleme ve kontrol sistemi önermemize olanak sağlamıştır.

Anahtar Kelimeler: raspberry pi 3, nesnelerin interneti (ıot), esp8266, ev otomasyonu,android

To My Beloved Family

ACKNOWLEDGEMENTS

First of all, I have to thank the Almighty God for giving me health, guidance, patience, perseverance and knowledge. We always remember when we are down with no ideas and things do get better after struggling for indeed there is ease in hardship. All this work would not have been possible without his blessings and favor. On this journey towards my master's degree, I have been extremely lucky to have got Prof. Dr. Ergun Erçelebi as supervisor and there are no words to match the help and support that he has been continuously providing over the past years. Without his help and his support, this thesis would not be possible.

Along the way, family is a very important aspect of a student's life and there is no denying their support and sacrifice in supporting me both in my good times as well as in times of hardship. They were instrumental to the completion of my work.

Last but not least, I would like to thank all my lecturers and friends at Gaziantep University. They have been my family in Turkey and they all deserve the credits for the completion of this thesis.

TABLE OF CONTENTS

Page

ABSTRACT v
ÖZET vi
ACKNOWLEDGEMENTS viii
TABLE OF FIGURES xi
1. INTRODUCTION 1
1.1 Connect Both Inanimate And Living Things 2
1.2 Collection Of Data By Sensors 2
1.3 Make Changes Over An IP Address 2
2. LITERATURE REVIEW 4
2.1 Centrally Based Home Automation System 4
2.2 Home Automation Systems Using Bluetooth 5
2.3 GSM Mobile Based Home Automation 6
2.4 Home Automation System Using SMS 7
2.5 Home Automation System Using GPRS 9
2.6 Home Automation Using DTMF 11
2.7 Home Automation System Using The Internet 12
2.8 The Internet Of Things (IoT) 13
2.9 Cost Of Current Home Automation Systems 14
2.10 Summary 14
3. METHODOLOGY 16
3.1 Proposed System 16
3.2 Live Video Streaming 16
3.2.1 Raspberry Pi 16
3.2.2 Secure Shell (SSH) 20
3.2.3 Putty 21
3.2.4 Video Camera 22
3.2.5 MJPG-Streamer 24
3.2.6 Extraction Of Live Video 26
3.3 Hardware Choice For Atmospheric And Lighting Parameters 27
3.3.1 Choosing The Correct Esp8266 Wi-Fi Chip 27
3.3.2 Power Switch Tail Kit 31
3.3.3 Using Arest Dashboard To Control Power Switch Via The Cloud 35
3.4 Controlling And Monitoring Atmospheric Parameters 39
3.4.1 Linking Dweet.io to Freeboard.io 39
3.4.2 IFTTT (IF THIS THEN THAT) 41
3.4.3 Summary 44

3.5 Android Mobile App Development 45
4. RESULTS AND DATA ANALYSIS 47
4.1 Mobile App 47
4.2 Mobile App Security 48
4.3 Mobile App Camera Streaming 48
4.4 Mobile App Power Switch Control 49
4.4.1 Power Switch Security 49
4.4.2 Mobile App Lighting Control 50
4.5 Mobile App Atmospheric Parameter Monitoring 51
4.6 Summary 52
5. DISCUSSION 53
5.1 Include Motion Sensor With Raspberry Pi To Take Picture 53
5.2 Make Extensive Use Of IFTTT To Incorporate 54
5.3 Make Use Of Multiple Cameras 55
5.4 Mobile App On Iphone 55
5.5 Customized Website And Database Development/ Marketability 56
5.6 Summary 56
6. CONCLUSION 57
REFERENCES 59
APPENDICES 62
Appendix A: Camera Streaming Programming Codes 62
Appendix B: Power Switch Program 66
Appendix C: Data Logger (Dweet.Io) Program 69
Appendix D: Mobile App Development Program 72

TABLE OF FIGURES

Page

Figure 2.1. Mobile based Home automation by A. Alheraish 6
Figure 2.2. Integral parts of classic internet based smart home 12
Figure 2.3. Famous IoT companies and startups to invest 15
Figure 3.1. Proposed system 16
Figure 3.2. Raspberry Pi 3..... 17
Figure 3.3. Physical equipment needed for first time booting of raspberry pi..... 17
Figure 3.4. Operating system download..... 18
Figure 3.5. Formatting SD card..... 18
Figure 3.6. Copying Raspbian Operating system onto memory card 19
Figure 3.7. Raspberry Pi 3 booting 20
Figure 3.8. Enabling SSH..... 20
Figure 3.9. Putty configuration window 21
Figure 3.10. Putty login..... 22
Figure 3.11. V2 noir camera..... 23
Figure 3.12. Enabling camera on raspberry pi 23
Figure 3.13. MJPG streamer via browser..... 26
Figure 3.14. Extracted video from MJPG-streamer 27
Figure 3.15. The ESP-01 module 28
Figure 3.16. ESP07 on the right along with the breakout board (left) 28
Figure 3.17. Olimex ESP8266 module 29
Figure 3.18. Sparkfun 'Thing' module..... 29
Figure 3.19. Adafruit Huzzah ESP8266 breakout board..... 30
Figure 3.20. Adafruit ESP8266 feather Board 30
Figure 3.21. Micro USB connector for the Adafruit ESP8266 feather board..... 31
Figure 3.22. Power Switch tail Kit 240V 32
Figure 3.23. Lamp control via Wi-Fi 33
Figure 3.24. Cloud controlled power switch 35
Figure 3.25. Creating aREST dashboard..... 36
Figure 3.26. Creating dashboard for power switch 36
Figure 3.27. Lamp controlled via aREST dashboard 37
Figure 3.28. Power switch Tail Kit switched on 37
Figure 3.29. Lamp state after ON button is clicked 38
Figure 3.30. Lamp state after OFF button is clicked..... 38
Figure 3.31. Connecting the DHT 11 to the ESP 8266 board..... 39
Figure 3.32. Freeboard creation 40
Figure 3.33. Dashboard creation 40

Figure 3.34. Temperature and humidity display on freeboard.io 41
Figure 3.35. Webhooks settings 42
Figure 3.36. SMS settings 42
Figure 3.37. Trigger part: "this" 43
Figure 3.38. Action part: "That" 43
Figure 3.39. Content to be sent as SMS 44
Figure 3.40. Other alternatives for action part (That) 44
Figure 3.41. Preview of IOT mobile app 45
Figure 3.42. USB debugging 46
Figure 4.1. IOT home automation Android App 47
Figure 4.2. Screen lock for IOT app 48
Figure 4.3. Raspberry Pi Camera streaming from IOT mobile App 49
Figure 4.4. Power switch control security 49
Figure 4.5. Light control from home automation and control App 50
Figure 4.6. Viewing Temperature via the mobile App 51
Figure 4.7. Humidity view on mobile App 52
Figure 5.1. Perimeter detection through motion sensor 53
Figure 5.2. PIR motion sensor 54

CHAPTER 1

INTRODUCTION

The internet of things is a hot topic in the industry nowadays. Its conceptualization is not difficult to understand but it is rather a potent one. If we equip our everyday life's devices and things with tags and identifiers attached to Wi-Fi, we would be able to make them interact with each other and could thus be controlled from a main computer. IoT sounds like science fiction but with the tremendous drop in size and cost of wireless technology as well as the improved mobile data coverage with many networks offering broadband speeds, there is an expectation of billions of devices that will be connected to the network within the coming years. The internet of things will further increase the number of appliances connected to the internet with a round figure of over 50 billion according to Cisco's internet of things group (Evans, D, 2011). Wearable gadgets such as watches and glasses are also going to be part of the future trend of devices that will connect and contribute to the high rise of IoT systems on the cloud. According to Forbes magazine, General Electric even predicted that investment in the Industrial Internet of Things (IIoT) in less than two decades will rise to a figure of around $ 60 trillion (Columbus, L., 2016).

NFC, RFID, WIFI, Zigbee and Bluetooth are some of the current local area connections (LAN) that a sensor connected to an IoT system can connected to. A set of interconnected local area networks make up a wide area network (WAN) and when coupled with sensors, we are able to gain more flexibility in a system whether the WAN being GSM, 3G, LTE or even GPRS. The internet of things can:

1.1 Connect Both Inanimate And Living Things

The vision of the internet of things (IoT) is not to be restricted to industrial applications only but rather to also encompass everyday objects ranging from gas turbines, automobiles, utility meters, plants, farms, animal and people. According to a special report in the Economist titled "Augmented Business" by (Herring, M., 2010), Sparked, a Dutch company implanted sensors in the ears of cows thereby allowing the farmers to track the movement and health of their animals. Nike+ Fuel band and Fitbit fitness tracker gadgets are another example relating to how people are also connected to the IoT ecosystem. Since the internet of things can relate to everything, it has been redefined as being "Internet of everything" (IOE) by Cisco.

1.2 Collection Of Data By Sensors

The physical objects are able to connect to one or more sensors namely location, vibration, temperature, humidity and motion. An example of this is how Lufthansa made use of an IoT based system in order to get data from its operations department and making it accessible to their customers (Columbus, L., 2016). Both the actual airport and aircraft's sensor data including weather conditions are being aggregated to consistently deliver excellent customer experience. The potential realization of such projects in the future will give rise to a "hyper connected world" whereby there will be no inherent limitations on the use of technology whatever be the application.

1.3 Make Changes Over An IP Address

Sensors and objects connected to an IoT network share the information of their condition and their surroundings with people. The data can be shared in real time and can be altered if necessary by the user.

Bringing the IoT technology to the home automation sector, a study made by technology website "MakeUseOf" which focuses on bridging technology and people showed that the average pricing of a smart home in the US consisting of switching electronic devices on and off, temperature and humidity control and intelligent security costs around $1400 (Lee, 2015). Based on the goal of sustainable

development of the UN, we have to continuously push the boundaries of technology so as to be able to bridge the gap between the rich and the poor and to improve the distribution of our world's resources. In our proposed system, an embedded Linux system and low cost Wi-Fi chips with complete internet control suite stack with a microcontroller chip interacting with a custom made Android app is used to give real time data of physically connected hardware. Live streaming of camera, control of electronic appliances as well as real time measurements of atmospheric parameters can all be seen and controlled via the internet via a tailor made Android mobile application. The electronic miniaturization of microcontroller technology also allows us to propose a scalable, personalizable and cost efficient hybrid home monitoring and control system using IoT technology.

CHAPTER 2

LITERATURE REVIEW

With the advancement of technology in the field of home automation, both luxury and security concepts are gradually being redefined. The amalgamation of wireless Sensor networks and ever present computers and internet connected abled hardware has allowed researchers and engineers to propose more ingenious solutions to provide more control and security to the household of the population. People are able to control different aspects of their houses including the environment. Along the years, there have been various home automation technologies which have been developed. Here are some of the most common ones:

2.1 Centrally Based Home Automation System

A central controlled based home automation system groups several houses into one secure network whereby each node is specifically allocated to one neighborhood based on the users' tally. The control nodes are further regulated by other controls also known as "central nodes" with high performance computing so as to improve the overall security of the units connected to the system. Home Security System on Intelligent Network (HSSIN) is an example of a centrally controlled home automation system (Tsai, et al., 1998). However the central controller based security system lacks modern security parameters and also faces the following unique challenges:

- There large majority of houses in the neighborhood have to join the system for it to be cost effective.
- There will be raised eyebrows on who controls and has all the access to all the security system of the houses in the neighborhood.

This technique is not ideal in securing single homes but rather suits a large number of homes as discussed above. However, even in the case of multiple houses, the central controller will have all the intimate and private information about any house at any time. As such, the main central controller unit may itself become a target. Furthermore, we already know how the general population feels about governmental surveillance on the Internet. As such, the end defeats the purpose as there is a high chance of privacy violation in a central controller based security system.

2.2 Home Automation Systems Using Bluetooth

Bluetooth technology usage in building home automation systems has been discussed thoroughly in this thesis (Sriskanthan et al., 2002). The microcontroller based sensor and device controllers are connected to a host controller on a PC. Scientists developed a system of rules on top of Bluetooth software stack also known as Home Automation Protocol (HAP) which allows the connected devices to share or exchange information. The I2C bus is used to connect the electronic devices and the device controller. As such, more device controllers can be connected to the main controller.

Another Bluetooth home automation system remotely accessible by GPRS was proposed by another researcher (Kanma et al., 2003). A GSM modem provided Internet connectivity while a mobile phone with Bluetooth connectivity was used as the host controller. In order for the home devices to be able to communicate with the host controller, they were fitted with Bluetooth communication adapters. Update of the devices, control, and fault diagnostic as well as an electronics user manual on the phone using Bluetooth are also discussed in the thesis. However, making use of Bluetooth in the home automation sector also comes with the following issues:

- Bluetooth technology consumes a consideration amount of energy and thus considerable focus should also be given to the recharging and replacement of batteries.
- Bluetooth technology has a maximum range of 100 m which might not be sufficient for all homes.

- With the new Bluetooth Low Energy Technology (BTLE), there are still issues concerning weak encryption and eavesdropping (Ryan, 2013).

Bluetooth technology seems to be a decent technology in home automation. It is relatively easy and quick to set up. However, there is also additional cost for every Bluetooth communication adapters as we increase the number of devices in the system. There are flaws such as distance, energy and security issues related to its usage as mentioned above.

2.3 GSM Mobile Based Home Automation

Due to the ever increasing usage of smart phones, a mobile based home control and monitoring system is attractive to the researchers as it tries to link everything at the ease of the hand. 3 types of GSM ways of passing on information are notably SMS, GPRS, and Dual Tone Multi Frequency (DTMF) based home monitoring and control system. They are going to be discussed along with their shortcomings.

Figure 2.1 shows how all devices interact with the home system using GSM technology (Alheraish, 2004).A GSM module making use of a Subscriber Identity Module (SIM) is used for communication. The system works like a typical system converting atmospheric parameters like temperature, humidity, and sound through a transducer into electrical signals which are then fed to the main microcontroller. The microcontroller is then calibrated and then sends instructions which are then interpreted by the GSM unit. The unit then chooses the best way of sharing the information from the 3 mentioned above.

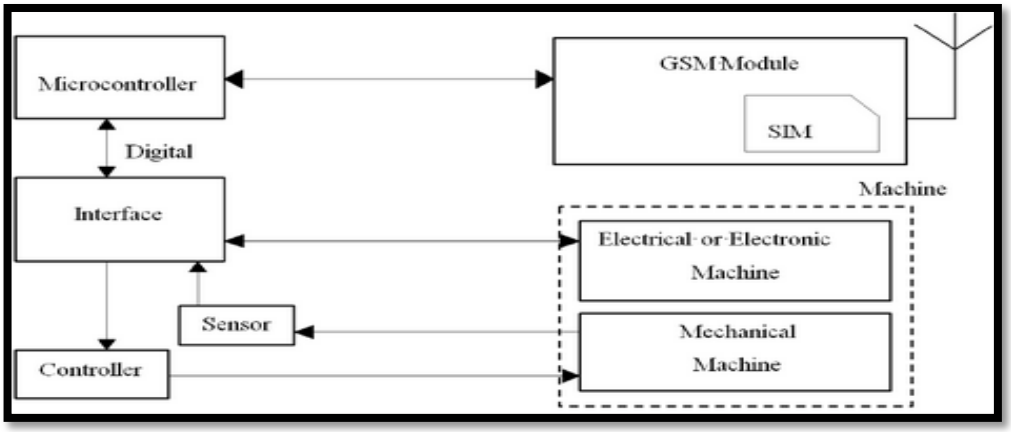

Figure 2.1. Mobile based Home automation by A. Alheraish

2.4 Home Automation System Using SMS

An SMS based home automation system is proposed by Alheraish (2004). The system allows the users to change the password for the lighting and the door. Furthermore, the system also allows the detection of intrusions by monitoring the state of the main door by incorporating the use of infrared sensors at the entrance. The password for the main door is a 4 digit number that can be set either by making use of a keypad or by using SMS from a registered mobile phone. The lights of the house can also be switched on by sending SMS from the registered phone.

Another example of a home security system depending on SMS has been presented by Khiyal et al. (2009).In this system, an SMS from an unregistered mobile phone is ignored and any incursion is tracked down and house proprietor is informed in the form of an SMS.

Another such scheme was suggested by Saeed et al. (2010) whereby the mobile phone has a Java application running on it. Users are able to log in with their username and password and can select any appropriate action they wish to. Whenever a user selects an action to accomplish, the mobile app will write the required SMS and the message is then sent to the GSM unit which acts like a decoder and interprets it to perform the required action. Facial recognition and a 4 digit password were also implemented in this work.

According to Delgado et al. (2006) an internet system has GPRS as a substitute in case something goes wrong. Here, we are seeing a gradual shift from a totally SMS based home automation system to an internet based one. Such a system provides with a fault tolerance and more durability to the system. The users will still receive SMS messages on their phone but they could also react either using SMS or by using a web interface. In this way, if one system fails, there is always a back up to compensate for the other.

Here are some of the security concerns about SMS-based home security systems:

- The work accomplished by (Alheraish, 2004; Saeed et al., 2010) relating to a 4 digit password in itself presents a security vulnerability. Since the owner enters the same key routinely, we cannot expect the person to be always on alert. This is due to the human factors' effect where the probability of one to become carefree and an attacker exploiting that opportunity.
- According to Saeed et al. (2010), different users are provided with different passwords which further enhance the vulnerability of the system due to more modern hacks.
- Most of the proposed systems are not in line with more modern sophisticated attacks. Regarding the idea proposed by Alheraish (2004), any sophisticated attacker nowadays would easily notice IR sensors at the main entrance and would usually choose to gain entrance to the house through some other ways.
- Using SMS for notifying an owner of any intrusion nowadays might not be the best solution since in this busy lifestyle people are not prone to frequently check their phone for received messages or the ringing tone might also be kept silent and thus they could easily miss such a notification.
- From the work of Mr. Saeed et al. (2010), making use of facial recognition to gain access to the system might be very detrimental as simple such systems could easily be hacked just by making use of a photograph of the required person since the system is not able to differentiate between human and picture.
- Modern hackers have the technology to clone the SIM card of any legitimate device and would gain access to the whole system in this case. Furthermore, if the user misplaces his or her phone, then the same issue would be encountered.

2.5 Home Automation System Using GPRS

Several home automation systems completely dismiss the threat that can be encountered by hackers and trespassers while only addressing the old fashioned intruder's techniques. With the modern world dynamics, mobile phone control has become the most coveted way of controlling homes.

Researchers Danaher and Nguyen (2002) proposed a different system which makes use of GPRS. 2.5 G is a technology which is 2G with added GPRS ability. It gives considerate quickness in transferring data by using the GSM system compared to 2 G which alone can only provide the system with a speed range of 20-40Kbps. The researchers used GPRS technology to view video and pictures of webcam installed in the user's home. The webcam detects movement by comparing frames for differences. However the video streaming section is not done using GSM technology by rather using a stable Internet connection.

GPRS usage along with mobile phone in the field of video camera surveillance has been described by Wu et al. (2006). When the door bell is rung or whenever there has been an intrusion, the camera unit is triggered. Infrared sensors are used to detect unlawful entrance. Whenever the buzzer key is pressed, the proprietor is called and audiovisual communication is established. Furthermore, during infraction, a picture is taken and sent to the owner's email address.

According to Yang et al. (2006), the condition of the connected devices in a system can be altered. First a mobile phone number has to be registered as a prerequisite. It is only when that number requests access to the home devices that a link is established and on the contrary any other device unregistered number is denied access. The mobile phone owner can control the devices at home in this way after going through the filtering security procedure to ensure that no threat is posed by the incoming instructions. In a situation of distress (fire), the hardware in the system establishes contact with the registered phone in the form of a call, message or even email depending on the settings used.

Another paper written by Ali et al. (2006) came up with a slightly different home automation system as to the research of Saeed et al. (2010). However the difference is that this paper deals with GPRS technology on mobile phone rather than a SMS based one. The system reaches out to the connected hardware though a server/client model realized by making use of a computer and a small Java App at home. The home computer's parallel port controls the device controller which in turn controls the home devices. Thus by gaining access to the device controller, the user is able to view and control the status of the system.

Unlike previous discussed controllers, another researcher made use of an embedded system based on a central controller (Jin et al., 2008). The system allows the control of devices and the viewing of data such as weather conditions at home by making use of a mobile phone. The researchers especially made this mobile application for China and the users are able to receive data about intrusions and emergencies like fire via the Chinese Instant Message Mobile Service.

An iOS mobile system using GPRS was developed (Das et al., 2011). The system makes use of a client/server communication model and they developed an iOS mobile App. The mobile behaves like a client requesting data and the connected hardware which are connected to the internet act as the server. Motion sensors, microphones and video cameras were installed in the house and whenever motion is detected, the camera start recording. The home owner can view the live camera on a mobile phone via GPRS or even a browser.

Security issues in GPRS home security systems:

- According to the research conducted by (Danaher et al., 2002; Das et al. 2011; Wu et al., 2006; Yang et al., 2006), the cameras were installed at the user's house. In here, we will see how streaming live videos via the net might not be a good idea. Imagine the consequences that would entail if the camera feeds were to be compromised. The vulnerability of wireless cameras has been highlighted by a BBC reporter (Kelion, L., 2014). With an eye to the reaction of the general population upon learning about governmental surveillances being carried out

without their consent, we can easily deduce what would be the consequences of such actions if they were to be perpetrated by hackers and thieves.

- If the infrared sensor based intrusion system as mentioned in the research of Wu et al. (2006) were to be compromised by a skilled attacker, the whole security of the system would be questionable.
- In a GPRS based system, there is a constant need to monitor one's phone to successfully defend against intrusion.
- Making use of email notification is not the best way to alarm someone of intrusion in his/her house as people nowadays usually put off the notification sound when they are usually busy at work.
- Researchers Das et al. (2011) made use of a web browser to access the home devices and statuses. This may lead to other types of security issues such as phishing, session hijacking, cross site scripting and cookie stealing amongst others.
- Limited security is provided by the work of Mr. Danaher and Mr. Nguyen (2002) as they do not have extra security measures and make use of only cameras.

2.6 Home Automation Using DTMF

The working on such a system is described in this thesis (Muhury and Habib, 2012). A user registers a SIM at home and then controls the connected devices by pressing on the buttons on his phone. This generates a DTMF signal which in turn is passed to the receiver. On receiver side, a DTMF decoder decodes the instructions via the GSM module. The decoded instructions are then fed to the microcontroller to implement the required actions.

Generally, DTMF based home automation is not widely used due to the presence of better communication tools with less disadvantages. DTMF based home automation systems are defenseless to attacks where random values are fed into the system (Jose and Malekian, 2015). In what is called a fuzzing attack, the hacker takes advantage of the weakness of the system's algorithms by giving it random data which causes the system to crash.

2.7 Home Automation System Using The Internet

The internet is the most popular and most easily accessible choice for home automation among researchers. The internet is widely available, easily scalable and is the most popular means of communication method in today's world. The internet provides us with low cost, high bandwidth and devices are able to network with easily via Wi-Fi. These are few of the advantages that the internet provides us with and looking at the trend, it surely seems like the trend is still going to revolve around the usage of internet for smart homes.

Looking at the issue with a consumers' point of view, using the internet is simple, and does not require a lot of research to do on how to use. Internet can also be accessed by everyday life's tools like smartphones, tablets, PCs and tablets amongst others. Figure 2.2 shows the integral parts of a classic internet based smart home.

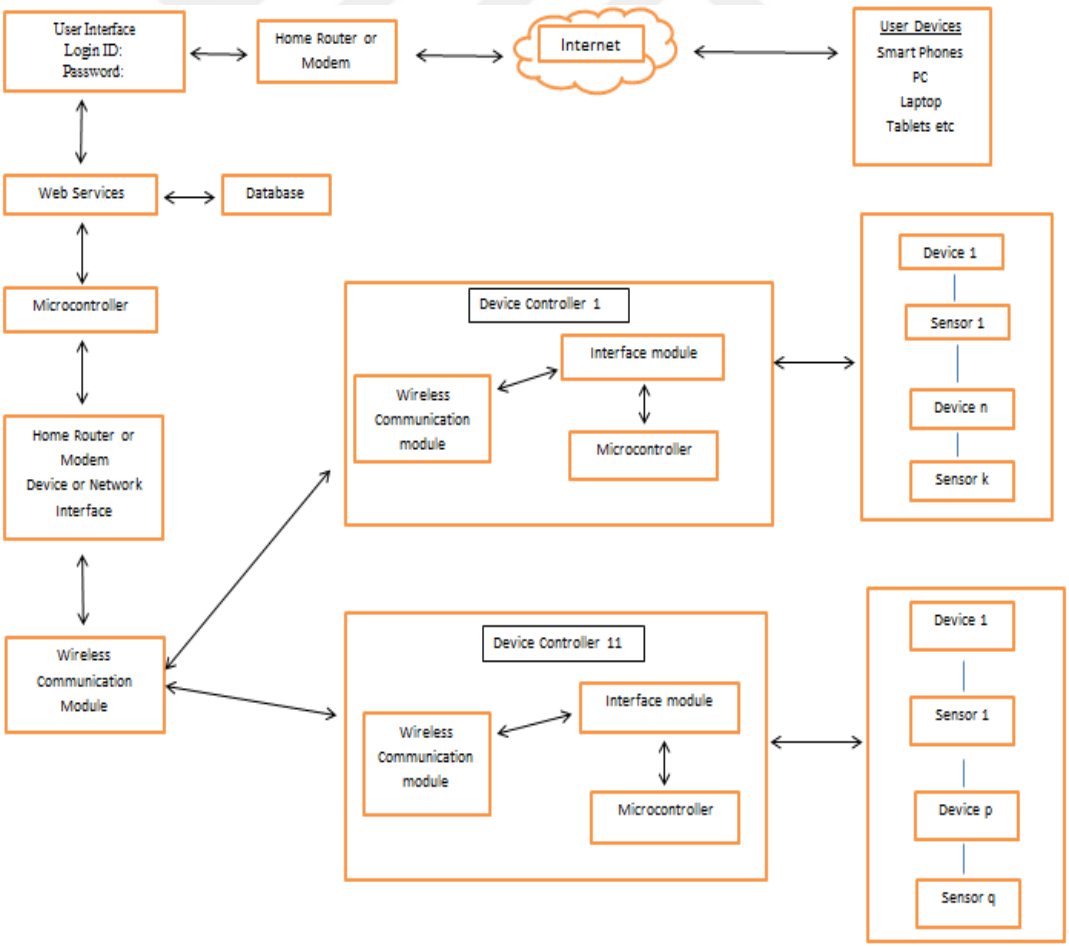

Figure 2.2. Integral parts of classic internet based smart home

A user interface can be webpages or any mobile or computer application made by anyone. The internet is then used to access an application or a web browser to gain access to the homes' devices. Smart phones' usage has also grown explicitly the last 15 years and researchers are thus developing mobile apps in order to bring the control of the homes directly at the ease of the users' finger. As for security measure, it seems to be the usual username/password combination that is most logical to make use of and it looks to be a continuing trend in the near future.

End user and their portable devices: As previously discussed, mobile phones are widely used to access the internet. However with the advancement in electronics, higher processing power, less power consumption as well as reduction in size and lower manufacturing costs, other alternatives such as laptops, tablets, and ipad amongst others are also being used to access the homes via the internet. This convergence towards the usage of the internet displays a more flexible, scalable and convenient way for users to access their homes.

2.8 The Internet Of Things (IoT)

The tendency to see computer, laptops, mobile phones and tablets everywhere on a global stage makes the Internet the best way of sharing and viewing data (Aazam et al., 2014). IoT nowadays is being discussed everywhere ranging from business magazines till conferences. Our world in the way it is is full of data and harvesting this data and to make it available for us to view and alter is what IoT is all about. Devices such as pacemakers, watches, animal collar tag, and many innumerable others can easily be linked to such a system and we are able to monitor and control them by making use of IoT. Simply put, IoT is the future of how people interact with the virtual world and that in itself, will make our world a more intertwined one. The IoT's vision is to build a connected world whereby data connected to the system could be viewed practically anywhere and anytime.

According to Islam et al. (2017), a course has been made at Central University under a program initiated by the Indian government. It was made in order to provide the students enough technical know-how to not only students but also to the researchers and teachers. They emphasized that IoT will play a massive role in curbing the

difficulties faced by the world population not only in smart homes but in everything around us such as in the transportation, IT, and energy sectors amongst many others. The goal was to bring together different scientists and engineers from different corners of the world to enhance the sharing of knowhow and dialogue which in turn would help the country reach a higher level of excellence in the field. The feedback received from the participants showed extreme usefulness of such a course.

2.9 Cost Of Current Home Automation Systems

According to the market research analysis conducted by Home Advisor website in the US, the national average to install a home automation system is $ 1209 in year 2017. The typical range is between $ 454 and $ 2014. There are 2 types of systems: Wireless systems and hardwired systems. The former is relatively cheaper easier to set up. In the case of a more complex and demanding requirement, we may choose to make use of the hardwired system. However, it should be noted that the latter is costlier and demands more space. If one wants to control or monitor his house from a distance, usually for such systems, a secure internet connection is also required.

2.10 Summary

The internet and the cloud have become inseparable nowadays. During the past 2 years, there has been plenty of cloud related announcements by several companies launching mostly new devices ranging from cars to refrigerators. Indeed they all had one thing in common: huge cloud storage. Cloud storage options of these new devices are being assumed to be incorporated by the customers.

People nowadays are expecting everything to be connected. Be it a switch, washing machine, fridge or even a car, they are all soon going to be connected or already communicate with cloud servers. We may ask the question why? The answer is very logical as companies that manufacture such devices have understood that it is illogical to have memory storage in the product itself as they must also be flexible, upgradable and scalable in such a way that they keep their worth over the long term.
Multinational companies such as Google, Microsoft, and Amazon Web Services have already started working on IoT projects relating to cloud services for their new

products. We are expected to see an ever-increasing number of cloud based devices as well as a growth in storage services and enhanced communication networks. The popularity of cloud enabled devices is another reason why cloud based system usage will keep on increasing over the coming years (Columbus, L., 2017).

IoT related projects have not only captured the attention of multinational companies but also has raised the stakes in providing new startup businesses dealing with a wide range of products namely home automation, monitoring, and other smart home devices. According to research service, by the year 2020, there will be around 34 billion devices that will connect to the internet. And from that amount, around 70% will be credited to IoT devices. The expected worth of IoT and its products will be worth trillions by 2020. Surely, multinational companies as well as new businesses have a huge market to share.

GE (GE)	AT&T (T)	Cisco (CSCO)	IBM (IBM)
Amazon (AMZN)	Skyworks (SWKS)	Apple (AAPL)	Sierra Wireless (SWIR)
Google (GOOGL)	Iridium Communications (IRDM)	Ambarella (AMBA)	ARM Holdings (ARMH)
Texas Instruments (TXN)	PTC (PTC)	Fitbit (FIT)	ORBCOMM (ORBC)
Garmin (GRMN)	Blackrock (BLK)	InvenSense (INVN)	Microsoft (MSFT)
Control4 (CTRL)	Silicon Laboratories (SLAB)	CalAmp (CAMP)	LogMeIn (LOGM)
InterDigital (IDCC)	Ruckus Wireless (RKUS)	Linear Technology (LLTC)	Red Hat (RHT)
Nimble Storage (NMBL)	Silver Spring Networks (SSNI)	Zebra Technologies (ZBRA)	Arrow Electronics (ARW)

Figure 2.3. Famous IoT companies and startups to invest

CHAPTER 3

METHODOLOGY

3.1 Proposed System

In order to accomplish the set objectives for the home automation and monitoring system, the project is going to be broken down into smaller tasks namely live camera video streaming, control and monitoring of atmospheric parameters, and development of android application. This is shown in figure 3.1 below.

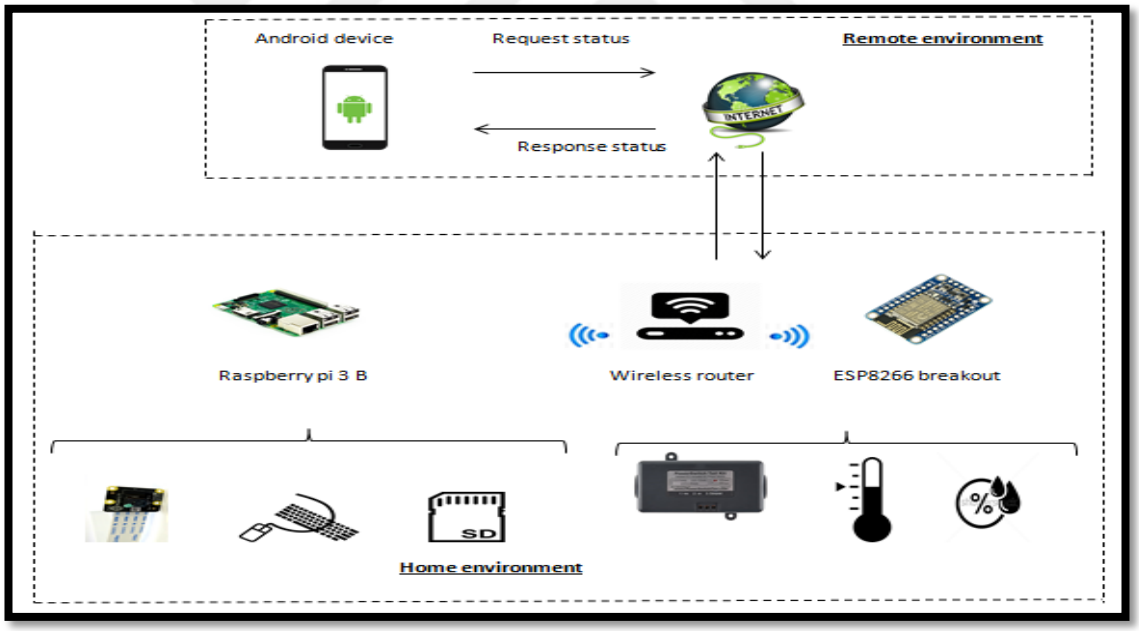

Figure 3.1. Proposed system

3.2 Live Video Streaming

3.2.1 Raspberry Pi

The Video Streaming section is done solely using the latest version of the Raspberry pi. The Raspberry Pi B has more processing power than the previous versions. It has

a 1.2 Ghz quad core processor which gives it around 10 times more than that of the pi 2. Furthermore, the Pi 3 model B is also equipped with on board 2.4 GHz Wi-Fi 802.11n (150Mbits/s) and Bluetooth 4.1(24Mbits/s) in addition to the 10/100 Ethernet port. The Pi 3's video controller is able to emit high definition (HD) and full HD pictures and recordings.

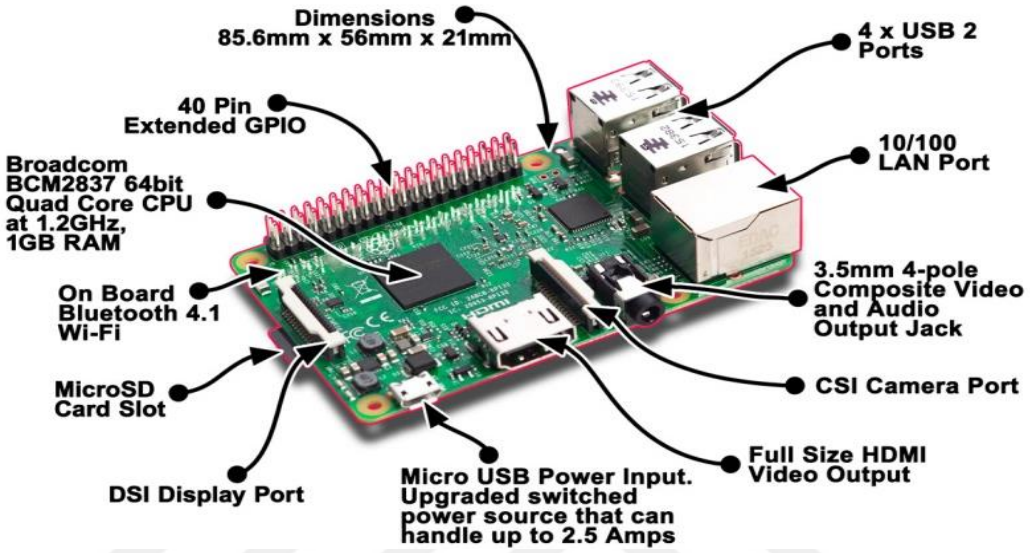

Figure 3.2. Raspberry Pi 3

The raspberry pi can be operated in either a headless setup or along with keyboard, mouse and screen. The screen uses an HDMI cable. An SD card is formatted using the SD formatter software.

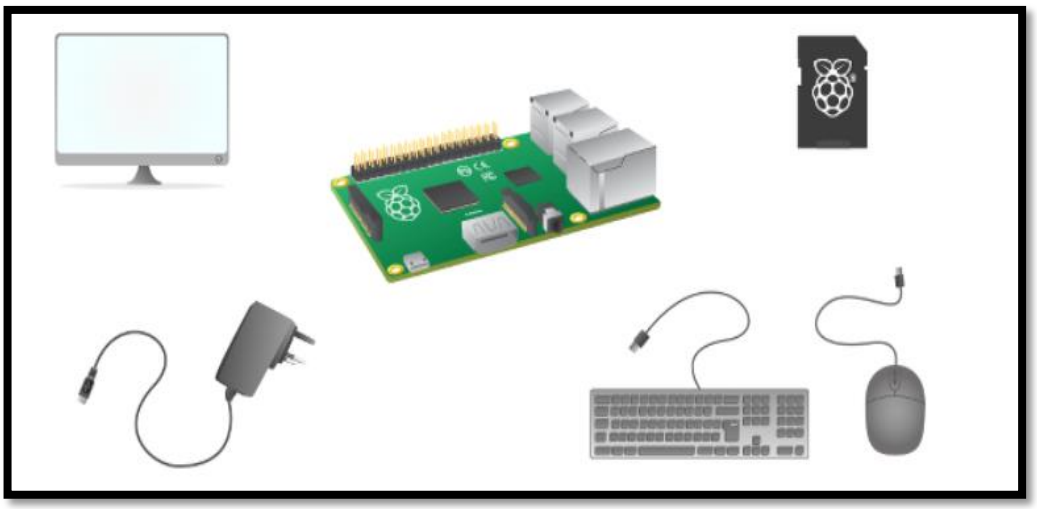

Figure 3.3. Physical equipment needed for first time booting of raspberry pi

The OS that is advocated for usage on the pi is widely known as called Raspbian. It is an adaptation of Linux that explicitly designed to work perfectly in tandem with Pi. The latest Linux operating system can be downloaded from the raspberry pi's official page as shown in figure 3.4. These operating systems are regularly updated.

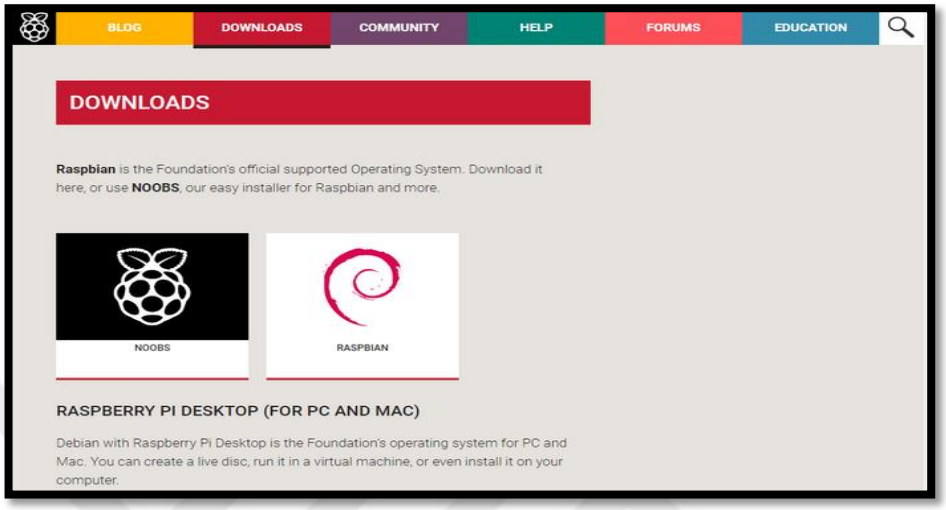

Figure 3.4. Operating system download

There are two ways of installing the operating system: One is to install raspbian using NOOBS and the second one is by directly downloading raspbian onto a computer. Before attempting to copy the downloaded operating system onto the memory card, the latter first has to be formatted. This can be done using the software SD Formatter 4.0 as shown in figure 3.5.

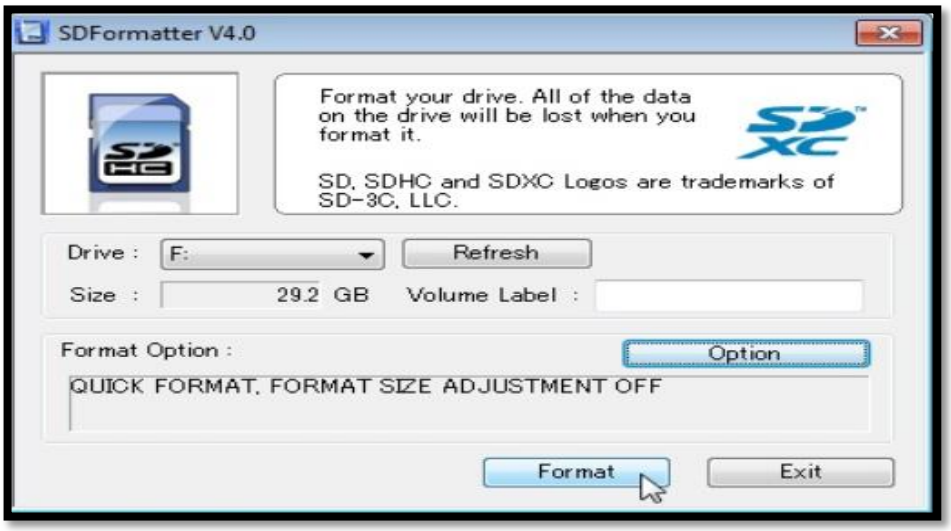

Figure 3.5. Formatting SD card

Raspbian OS was directly downloaded from the official page without the use of NOOBS and the ZIP file extracted in a folder. Win 32 disk imager can be used for copying the operating system onto the memory card. It is important not to copy the OS directly using copy paste as it will not work. Another program that can be used to copy the OS onto the memory card is Etcher SD card image utility as shown in figure 3.6. The steps below describe how to install the operating system onto the memory card.

- To download and install Etcher SD card image utility
- To run the program
- To select the correct drive
- To transfer the OS onto the drive

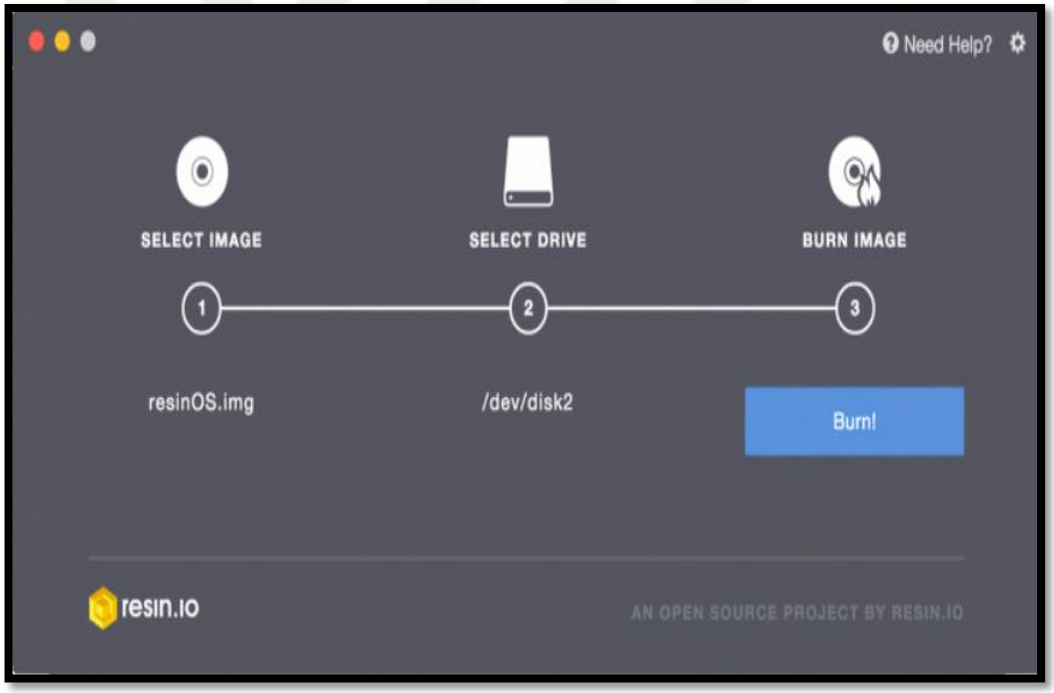

Figure 3.6. Copying Raspbian Operating system onto memory card

After successful mounting of Raspbian image onto the SD memory card, the card is tucked back into the raspberry pi 3 and the whole system is switched on. The system boots as can be seen from Figure 3.7.

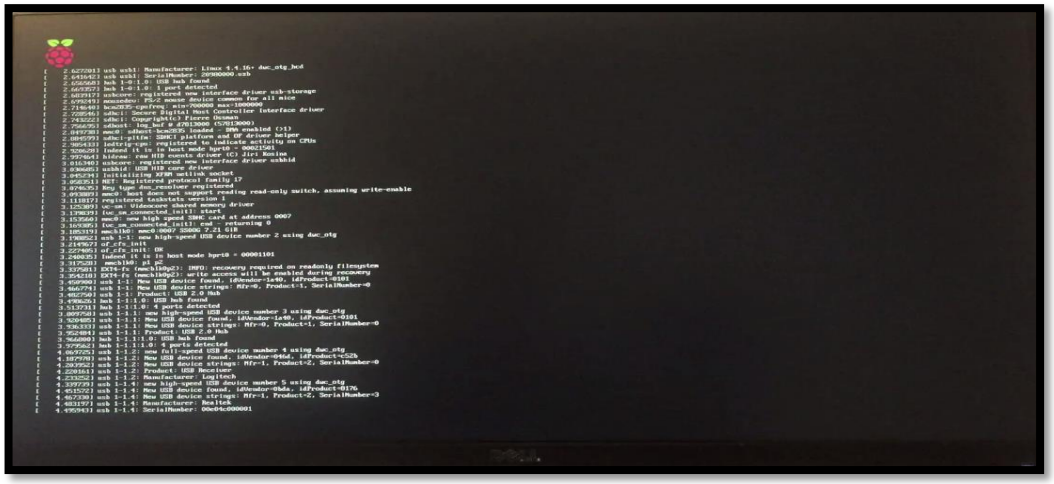

Figure 3.7. Raspberry Pi 3 booting

3.2.2 Secure Shell (SSH)

After successfully logging in for the first time, since most times it is considered a bit too bulky to always connect a screen to the raspberry pi for it to operate, we have opted to securely connect to the raspberry pi via remote access from another laptop. Secure shell is widely used by the community to remotely accessing and managing systems remotely. A raspberry Pi operating without a screen is also known as working in “headless mode”. In order for the raspberry pi to be able to run in headless mode, we have to first enable it by opening the raspberry pi terminal and typing the following: sudo raspi-config. SSH is then selected and enabled as shown in figure 3.8. However, it is important to remember that there will be access only to the raspberry pi’s command window and not the full desktop environment.

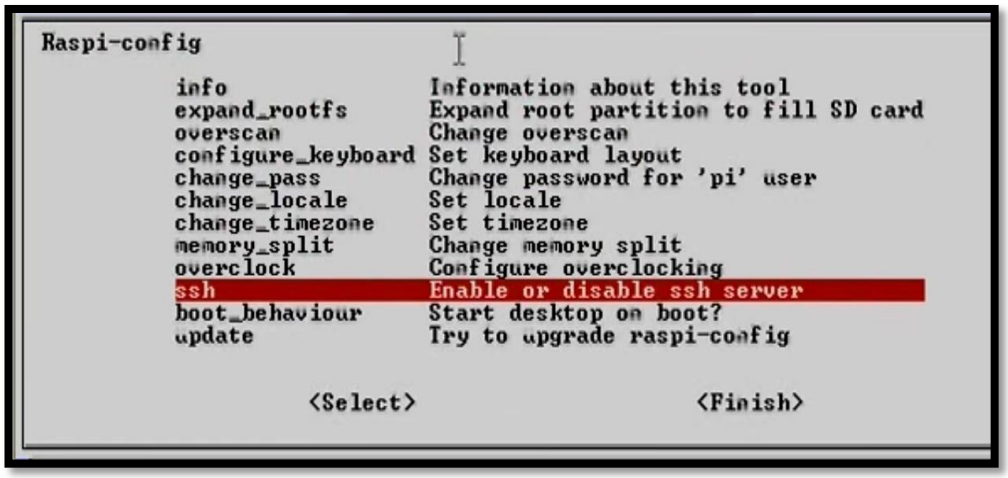

Figure 3.8. Enabling SSH

3.2.3 Putty

The raspberry pi can be terminally logged in using Putty which is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. Putty can be downloaded as a stand-alone .exe file. After running the program, the following configuration window is obtained.

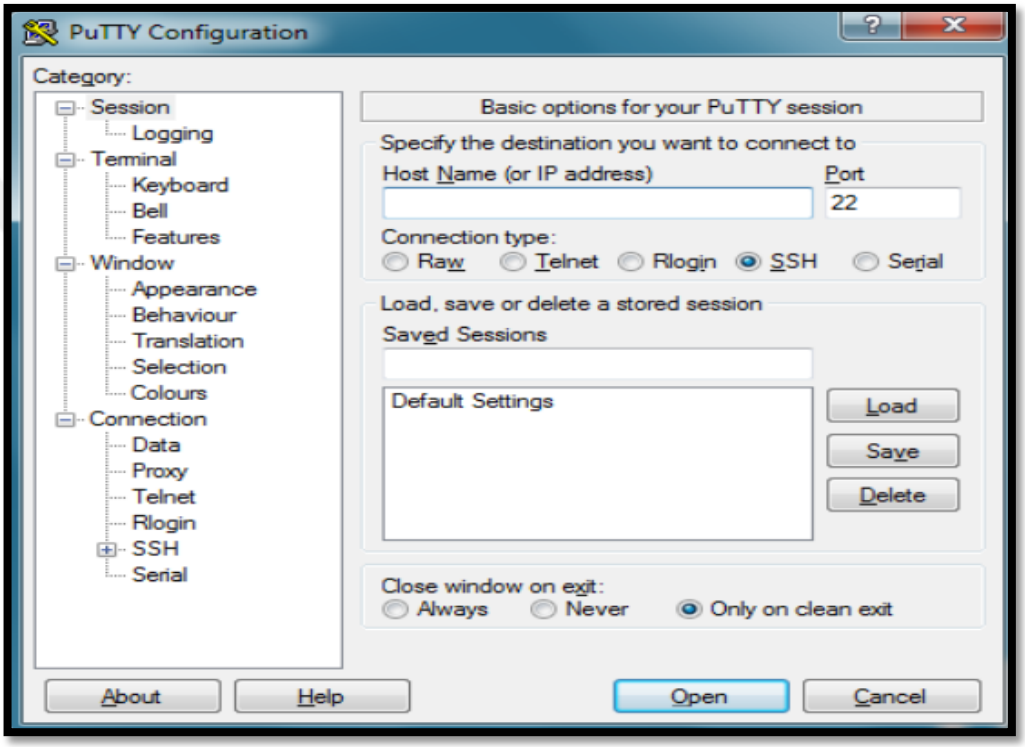

Figure 3.9. Putty configuration window

The IP address of the raspberry pi is obtained by writing the following in the raspberry pi terminal window: "`sudo ifconfig`". That address is keyed in the Host Name and the Open button is the clicked. A usual login prompt consisting of a username and a password is obtained. The default username and password respectively are "pi" and "raspberry" respectively. After confirmation, access to the pi is granted via SSH through putty as can be displayed in figure 3.10.

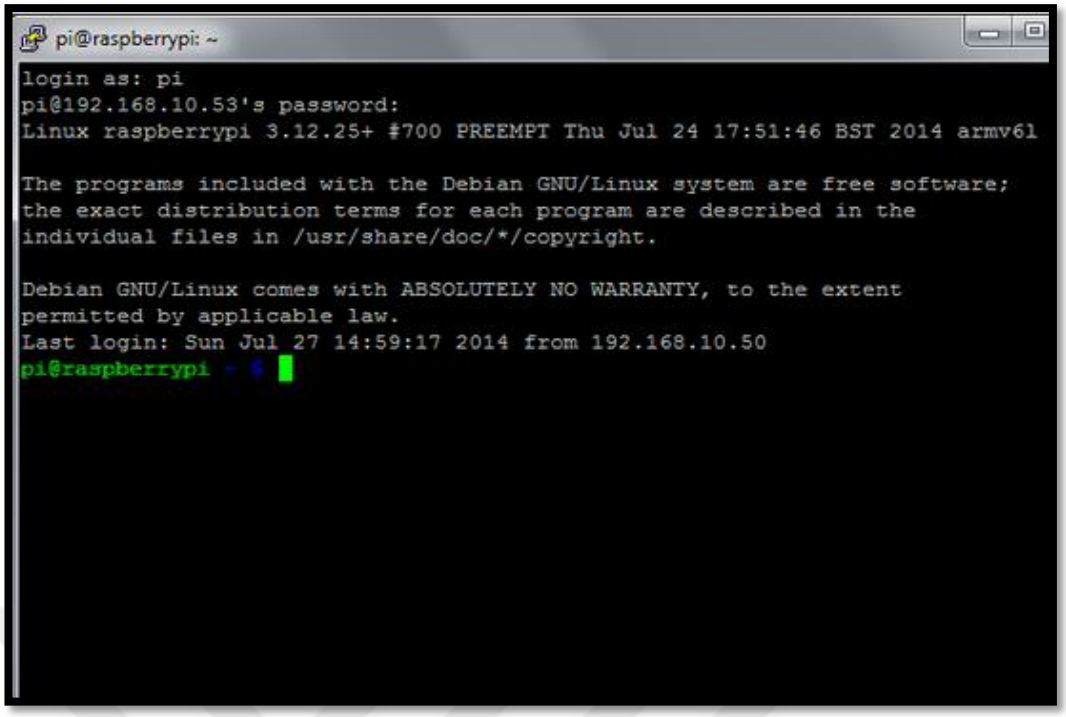

Figure 3.10. Putty login

From the above, access to the pi's command prompt is achieved and commands can be sent over the network to the pi in the form of Linux commands. RDP desk or VNC can alternatively be used to remotely access the graphical user interface (GUI) of the headless pi.

3.2.4 Video Camera

The camera pi Noir version 2 of the pi as shown in figure 3.11 is the officially phrased as the "night vision" made by the parent company of the pi. It is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for the raspberry pi. It is capable of 3280 x 2464-pixel static images and also features the following ranges of videos: 1080p30, 720p60, and 640x480p60/90 videos. The NoIR camera has no infrared filter (NoIR) which makes it even better for doing infrared photography and taking pictures in dim lights.

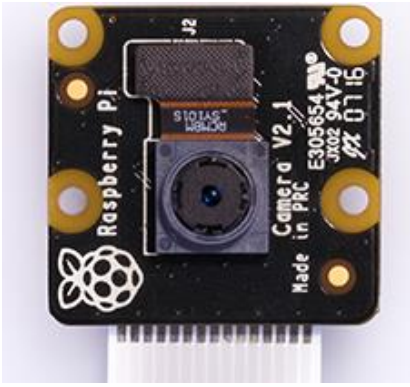

Figure 3.11. V2 noir camera

The V2 noir camera is connected to the raspberry pi 3's camera port and is then software enabled on the pi's configuration tool. First type the following in the raspberry pi terminal or via putty: "`sudo raspi-config`". The following window will open as shown in figure 3.12.

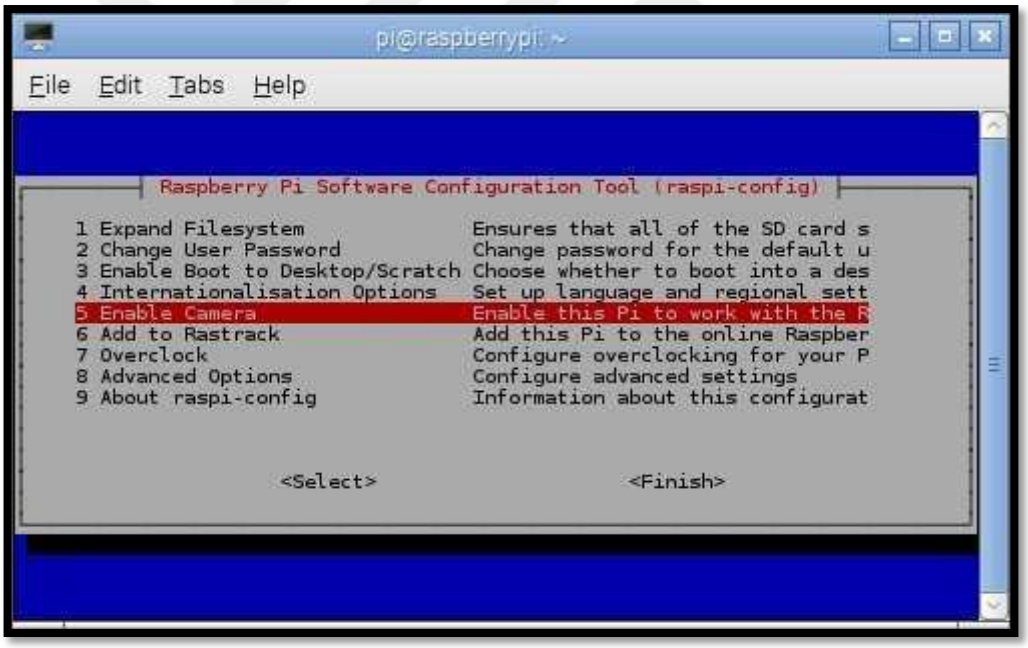

Figure 3.12. Enabling camera on raspberry pi

After successfully enabling the camera, the system will have to be rebooted for the changes to be applied. To check whether the camera is functional or not, the subsequent codes have to be keyed in: "`raspistill -v -o test.jpg`". The display should show a 5 seconds video preview along with a saved picture titled test.jpg. We should also note that a normal usb camera would also do the same job but with different video quality.

3.2.5 MJPG-Streamer

HTTP streaming makes a set of orderly arranged JPEG pictures which are sent to the clients. Whenever there is a call for a motion JPEF file, the server streams the JPEG pictures in an orderly manner via HTTP. Since raspberry pi deals with a Linux system, what is needed is a program that gets data from a video for Linux (V4L2), and sends it through a HTTP session.

MJPG-streamer is a command line application that duplicates JPEG pictures from one or multiple input plugins to multiple output plugins. From an IP based network, we are able to stream the motion of the pictures in an orderly way via most browsers as well as VLC and mplayer. MJPEG format guarantees great image quality along with great system robustness that is even if one frame is dropped, the video is not affected. The targeted frame rate is between 30 and 40. In order to install and use the MJPG streamer, the following must be done.

- Download the MJPG-streamer and initialize using the following commands:

```
git clone https://github.com/jacksonliam/mjpg-streamer
cd mjpg-streamer/mjpg-streamer-experimental
```

- Download required library (libjpeg8-dev) and key into the command prompt /putty the following: sudo apt-get install cmake libjpeg8-dev

- Building all the plugins are done by the following:

```
cd mjpg-streamer-experimental
make
sudo make install
```

- To start the streaming, use the following:

```
export LD_LIBRARY_PATH=.
```

```
./mjpg_streamer -o "output_http.so -w ./www" -i
"input_raspicam.so"
```

Instead of writing the codes all over again every time we want to start streaming, a script is going to be written and made executable so as to circumvent this minor inconvenience problem. We first create a file in a specific folder with the name camera.sh. The following command is then sent to open the file: nano camera.sh. In order to make the command camera.sh executable, we sent the following commands:

```
Chmod +x camera.sh
```

Whenever we want to launch the camera, we just have to go to the specific folder in the pi and type in the following:

```
./camera.sh
```

It is also important for us to run the camera in the background such that we are capable of sending more commands to the pi whenever needed. This is done mainly by opening another terminal from the pi or from putty and the process resumes as normal.

In order to check whether the streaming is ongoing, the IP address of the raspberry pi along with the port number 8080 (default) is inserted in the browser (chrome, Mozilla…) and we are greeted with the figure 3.13. In order to view the stream, we only have to select stream from the left hand side.

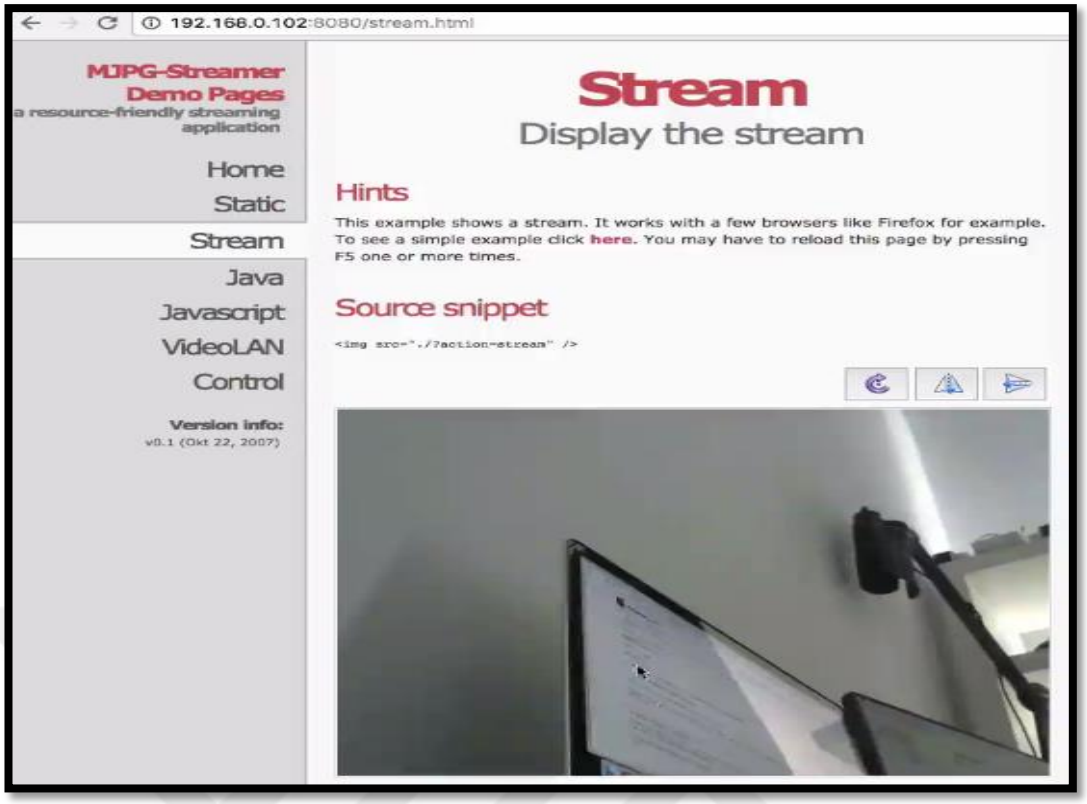

Figure 3.13. MJPG streamer via browser

3.2.6 Extraction Of Live Video

Node.js module allows us to write JavaScript applications on the server. It was made using C++ as language and has been built on the V8 JavaScript runtime. Its launch previously was made to act as a server for Apps but later reverted to make it more suitable for automation. In order to make appropriate use of the Node.js packages, they first have to be downloaded, and installed. The use of node package manager is for proper management of the packages. It also gives a useful interface to work with them. The following commands have to be typed into the terminal in order to install the required modules:

Sudo npm install express arest --unsafe-perm

We first have to make a server on another port on the raspberry pi. The server will be chosen to be on port 3000 and the video alone is going to be extracted and forwarded to the mentioned port. In order to be able to accomplish this, we have to make two files namely interface.html file for viewing the camera and interface.js file for

starting the server on port 3000. The codes for both files have been provided in appendix A. In order to start the server on page 3000, we just type the following in the terminal:

Sudo node interface.js

To check whether the extraction has worked, we can test on any browser the IP address of the raspberry pi followed by: 3000 instead of: 8080. Figure 3.14 shows the results of the successful extraction of the live video. The IP address and the port number are very important as we will be requiring them while building the mobile application.

Figure 3.14. Extracted video from MJPG-streamer

3.3 Hardware Choice For Atmospheric And Lighting Parameters

3.3.1 Choosing The Correct Esp8266 Wi-Fi Chip

The ESP8266 is great microcontroller for all smart home and IoT projects. It is less than $5, has Wi-Fi, a processor, and can be programmed using Arduino IDE. However, it is sometimes difficult to actually choose the optimum ESP8266 board. Of course there are a lot of versions of the ESP chip but which one to choose is what is going to be explored. They are all described below along with their advantages and disadvantages and the best choice is going to be selected for the second section of the project.

3.3.1.1 ESP-01 Module

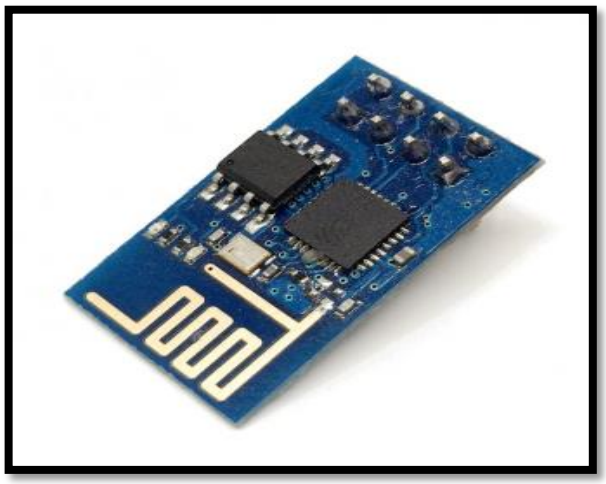

Figure 3.15. The ESP-01 module

The ESP-01 module was the first ESP microcontroller that was released. It is quite cheap and very small. It can be programmed with Arduino IDE. Nonetheless, its main disadvantage is that firstly we cannot fit it directly into a breadboard without an adapter and secondly, we don't have accessibility to all of its pins.

3.3.1.2 ESP07 Module With Breakout Board

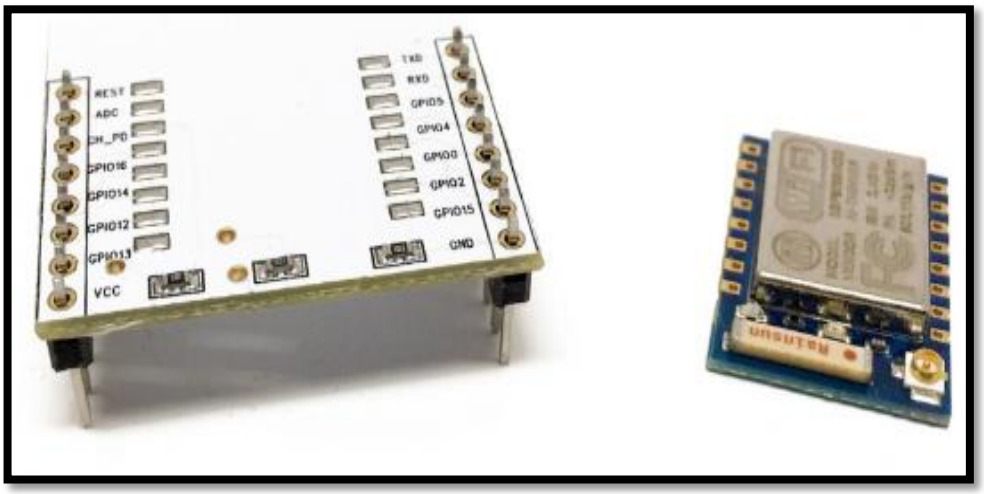

Figure 3.16. ESP07 on the right along with the breakout board (left)

The advantage of this module is that it can easily be integrated on a PCB later. However this module cannot be used as a standalone board on a breadboard as we require an adapter board to use it in the breadboard projects.

3.3.1.3 Olimex ESP8266 Module

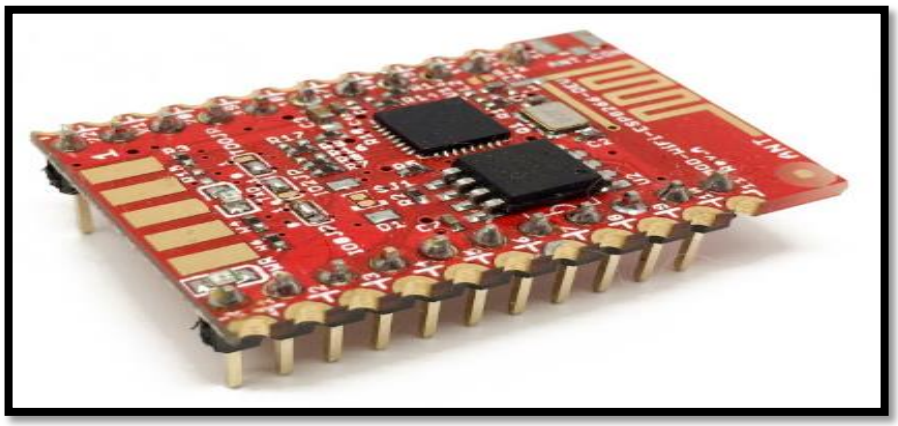

Figure 3.17. Olimex ESP8266 module

Olimex ESP8266 module is a compact breakout board. We have access to all of its general I/O pins and can be obtained at the mere price of $10. It is important to note that it comes with extra memory: 2 MB of SPI flash memory compared to the usual 512 KB. In the case of a large project with the ESP8266, this chip might be the best choice.

3.3.1.4 Sparkfun 'Thing'

Sparkfun 'Thing' is relatively similar to the Olimex module. Nonetheless, in contast to the previously discussed chips, it can be powered by a USB cable or even a lithium battery connector. It thus offers a great extent of flexibility in powering it. Furthermore, in the IoT field, it is better to make use of a Lithium connection as it does not take too much space.

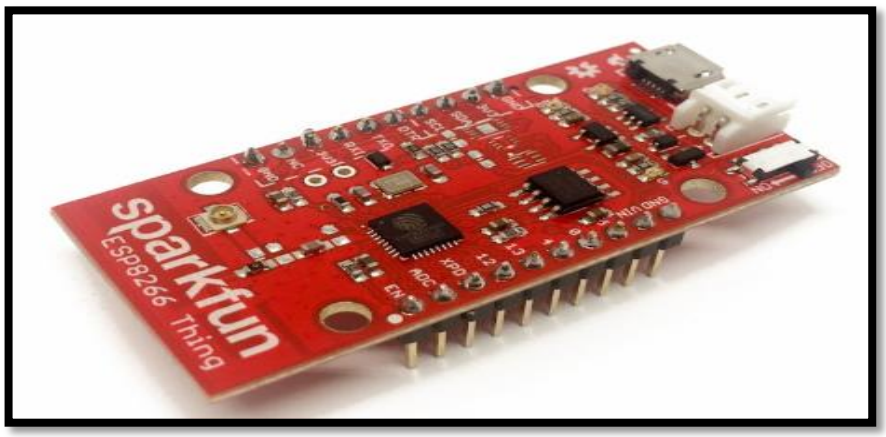

Figure 3.18. Sparkfun 'Thing' module

3.3.1.5 Adafruit Huzzah ESP8266 Breakout Board

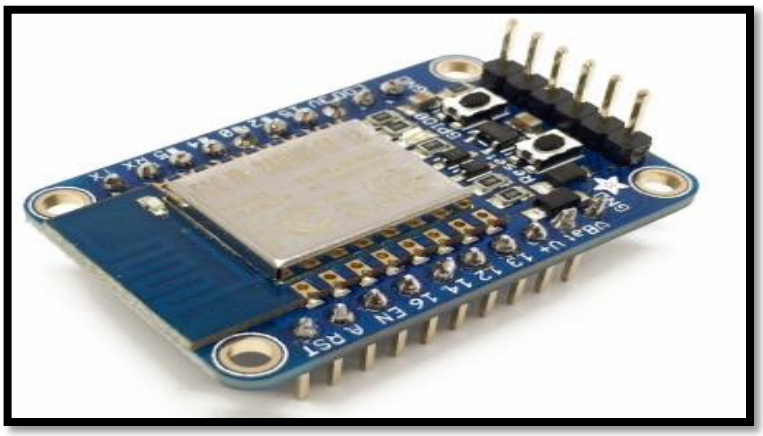

Figure 3.19. Adafruit Huzzah ESP8266 breakout board

Figure 3.19 displays the ESP8266 microcontroller made by the company Adafruit. Performance-wise, it is comparable to the Olimex microcontroller. On the other hand, it has a FTDI header on the board and is suitable while programming with an FTDI cable.

3.3.1.5.1 Adafruit Feather HUZZAH ESP8266 Board

Adafruit also has another board based on the ESP8266, which is the Adafruit ESP8266 Feather board that also comes with onboard USB. At its core, it is an ESP8266 Wi-Fi microcontroller at a frequency of 80 MHz and can be powered using 3.3 V. The board has an Wi-Fi inbuilt on the board. It can also be programmed using the Arduino IDE. In addition to that it has a speed of communication of 921600 baud which is really fast for its size. Besides that, it also comes with a reset pin and a LED light.

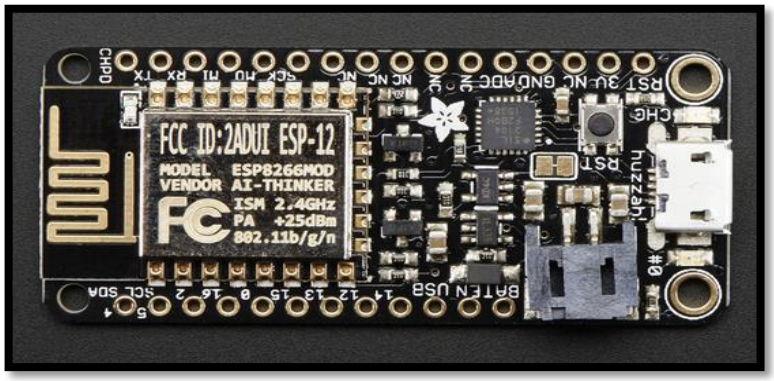

Figure 3.20. Adafruit ESP8266 feather Board

For the project built with such a microcontroller and to make it more portable, Adafruit has made a 3.7V lithium battery. In the eventuality of the project being a non-portable one, the chip will work perfectly as well if also powered from a USB cable. In case both of the power supplies are connected, the USB will recharge the battery automatically further enhancing its flexibility. The datasheet for the Adafruit ESP8266 feather board can readily be found on the internet.

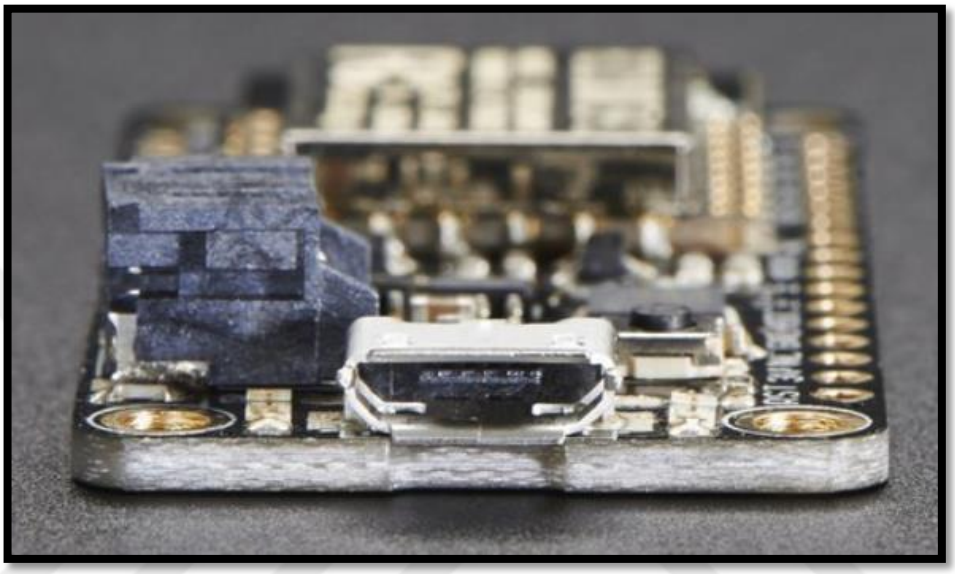

Figure 3.21. Micro USB connector for the Adafruit ESP8266 feather board

3.3.1.5.2 Conclusion

All the ESP8266 modules have their advantages and disadvantages with respect to each other. However, which one is the best depends also on how we intend to use them. In our project, we are looking more towards flexibility, portability, price and ease of programming. For these obvious reasons, we have used the ESP8266 feather board from Adafruit as it has onboard USB port and can be programmed without the use of FTDI cable along with the fact that it can also be powered by Lithium Polymer batteries.

3.3.2 Power Switch Tail Kit

The use of simple relays might not be the smartest idea while dealing with 240V AC. The effortless method to control an appliance may it be an ESP chip, Arduino, a

beagle bone or even a pi is to make use of the latest Power switch tail kit 240. The voltage for controlling the device can be tailor made by varying its resistance. The Power Switch Tail Kit has the ability to work with an input voltage ranging from 3-24 V DC. Furthermore, we are also allowed the freedom to choose the cable to use with the kit. The kit has 3 pins to be connected to the ESP8266 board: One for the control signal (3-5 V for + input), another one for the – input and one pin for the ground. As such, the Power switch tail kit gives us the ability to control any high voltage electronic appliance in the house ranging from refrigerator, lamp, air conditioning or even the garage door. The power switch can be seen in figure 3.22. Since it has been in high demand, Adafruit has even recently manufactured a controllable four output power relay module as a Power switch alternative. However, the latter can control up to 120 V AC.

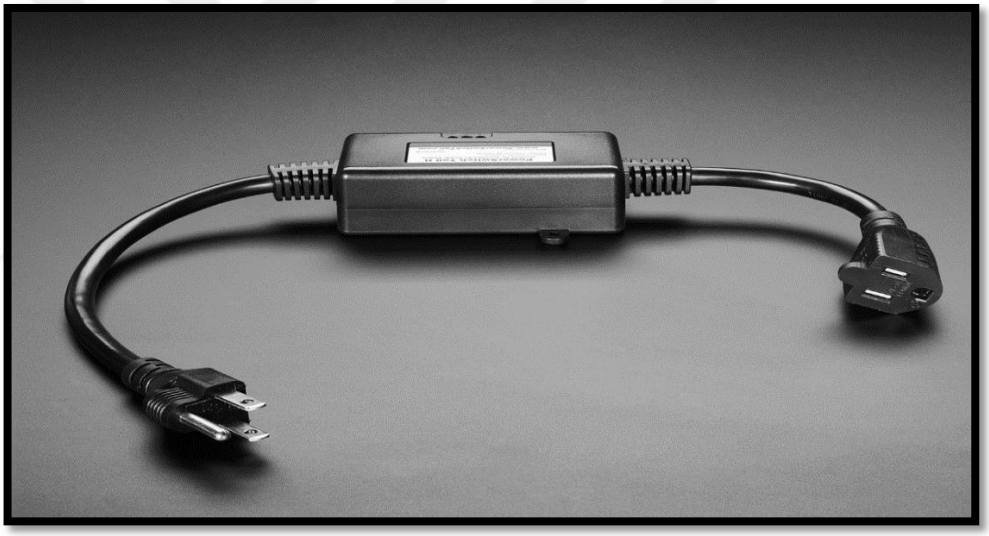

Figure 3.22. Power Switch tail Kit 240V

3.3.2.1 Building a Cloud Controlled Power Switch

There are many ways of doing the same thing. We are going to discuss a little bit about the pros and cons of each one. We could make a system like figure 3.23. However, such a system will only be available on the local network via Wi-Fi rather than the internet where it could be controlled from anywhere in the world. This is the main reason why we need IoT platforms whereby the IoT project interacts with an online server in such a way that we are able to control and monitor the related parameters via the cloud.

Figure 3.23. Lamp control via Wi-Fi

3.3.2.1.1 AREST Framework

AREST is a foundation that can be used to build powerful applications based on the Arduino and the Pi platforms. It can work with a series of communications mainly via Serial, Wi-Fi, and Ethernet amongst others. It has inbuilt libraries for each board, and also server-side code to manage contact between the server and the distant appliances connected. In a nutshell, the library permits us to send instruction to a board running aREST to make an action and to send info back in a JSON container.
The aREST framework is the open source tool that we can use to control any ESP8266, Arduino and raspberry pi boards from anywhere in the world. This is what is going to be used to control the 240 V power switch via cloud. The characteristics of an IoT platform are as follows:

- Storage for data sent by projects
- 2 way communication with the projects for remote control
- User interface: device management, data recording/visualization, dashboard creation
- API access
- Hardware libraries

There are many ready-made IoT platforms that can be used to achieve the same objectives of viewing the data via the net. The next section describes more in detail about the other different platforms, some of which are going to be used to achieve other objectives of this project.

The question we have now is where we write the codes and how to we send them to the ESP8266 board. The surprising answer to that question is: Arduino IDE. The codes written in Arduino software are referred to as sketches in the C programing language. These sketches are saved with the file type .ino. The toolbar buttons allows us to check the validity of the sketches ad to upload, modify or delete the codes as well as to open the serial monitor.

3.3.2.1.2 Other Internet Of Things Platforms

- **Dweet.io**

We all know the social media application for tweeting: Twitter. The internet of things has its own Twitter called Dweet.io which is used for messaging and alert notifications. Any machine that can link to the internet can make use dweet.io to either publish or to subscribe to the “dweeted” data. Dweet.io does not need any set up or sign up. It is a machine to machine communication for the IoT projects. The product is open source.

- **Freeboard.io**

Connecting any device to the dweet.io platform will see it assigned a name and begin sending data, or "dweeting" to the cloud. The purpose of Freeboard interface in all of this is to assemble the dweets and to introduce them as more expressive graphs, text and gauges. The product is open source.

- **Sparkfun Phant**

Sparkfun Phant has been developed by the company Sparkfun to log data coming from devices like the ESP8266. However, one of its biggest disadvantages is that it comes with libraries optimized for boards sold by Sparkfun. The product is open source.

- **ThingSpeak**

ThingSpeak is a platform used to store data measured by devices, mainly centered on detailed (Matlab) analytics. It can be used to trigger actions based on the measured data. The product is open source.

❖ **Adafruit.io**

Adafruit.io has been developed by Adafruit to control devices remotely and log data. It usually comes with libraries optimized for boards sold by Adafruit. The product is open source.

❖ **aREST**

aREST can be used to control projects from anywhere in the world via the cloud as well as logging and plotting data. It was developed to be independent of the boards used. The product is open source.

3.3.3 Using Arest Dashboard To Control Power Switch Via The Cloud

The coding for the ESP8266 board is done on Arduino IDE. It is written using the C programming language. The aREST framework is used to communicate with the cloud server and thus we make use of the aREST libraries in our coding. Furthermore, we also assign a unique ID to the power switch. This is needed for us to be able to identify the device on the cloud server. The Wi-Fi SSID and password are also inserted so as for the ESP board to be able to connect to the internet. The coding has been attached and explained in the appendix B section titled "POWER SWITCH CODING". The codes are the sent to the board via a USB cable and the board is restarted. For demonstration purposes we have made use of a 240 V lamp connected to the power switch which is in turn controlled by the control signal coming from the ESP board. The ESP board communicates connects to the cloud and communication is established via the aREST framework. The hardware setting can be seen in figure 3.24.

Figure 3.24. Cloud controlled power switch

An aREST dashboard then has to be made on the aREST website. The user has to make a username and a password and can then only start using the service for free. How is create a dashboard is shown in figure 3.25.

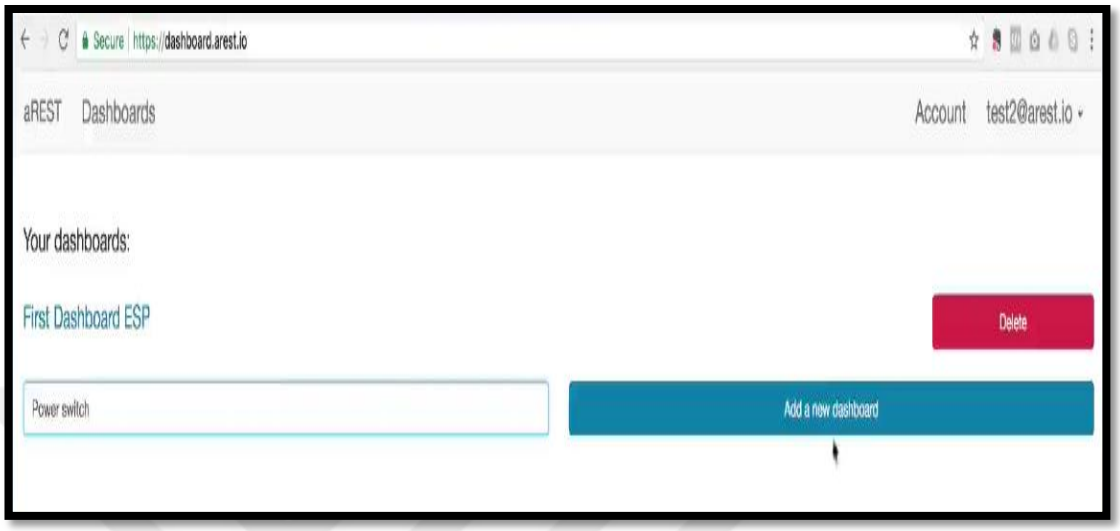

Figure 3.25. Creating aREST dashboard

The name of the project followed by the unique ID used in the Arduino sketch is then inserted. Digital switch is chosen and the pin number 5 from the ESP chip is chosen as input. We want the display to be simple on/off buttons as shown in figure 3.26.

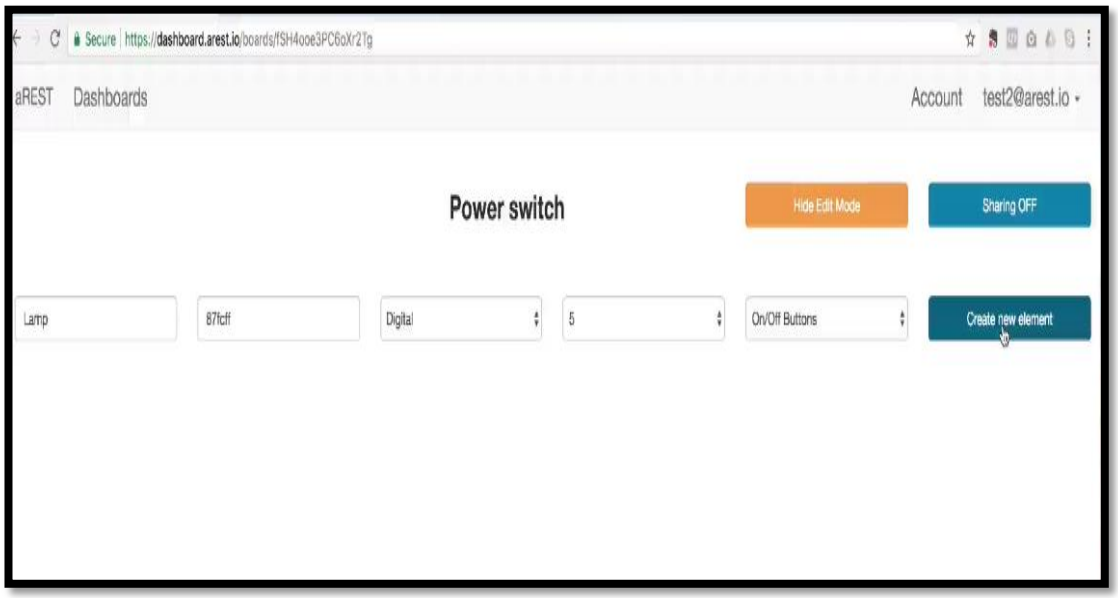

Figure 3.26. Creating dashboard for power switch

After putting clicking on create new element, we are greeted with the new window as shown in figure 3.27. The lamp can be seen to be online and the state can be seen to be low. As such, we are able to toggle the lamp on and off from anywhere in the world. It is important to note that we can make multiple dashboards consisting of multiple elements such as lamp, fridge, air conditioning and garage amongst many other uses.

Figure 3.27. Lamp controlled via aREST dashboard

3.3.3.1 Results

3.3.3.1.1 Power Switch Tail Kit Switched On And Effect On Hardware

The lamp is switched on by clicking on “ON” from the aREST dashboard.

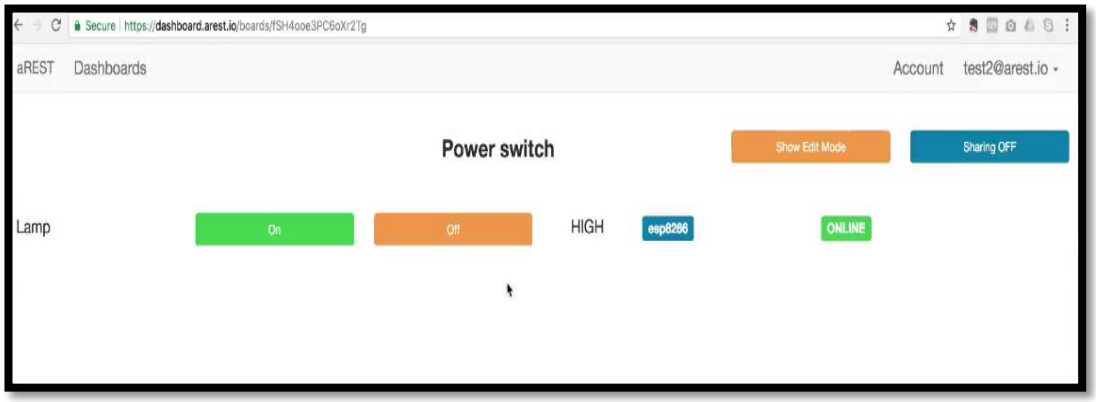

Figure 3.28. Power switch Tail Kit switched on

The lamp is switched on as a result of clicking on the “ON” button from the dashboard as shown in figure 3.29. This shows us that the power switch tail kit is being controlled via the cloud.

Figure 3.29. Lamp state after ON button is clicked

3.3.3.1.2 Power Switch Tail Kit Switched Off And Effect On Hardware

Whenever the “OFF” button is clicked on the aREST dashboard, the lamp is immediately switched off as shown in figure 3.30.

Figure 3.30. Lamp state after OFF button is clicked

3.4 Controlling And Monitoring Atmospheric Parameters

The hardware requirement for the atmospheric monitoring and control section is the DHT 11 temperature and moisture sensor working in tandem with the Adafruit ESP8266 feather board.DHT11 is an affordable digital temperature and humidity sensor.

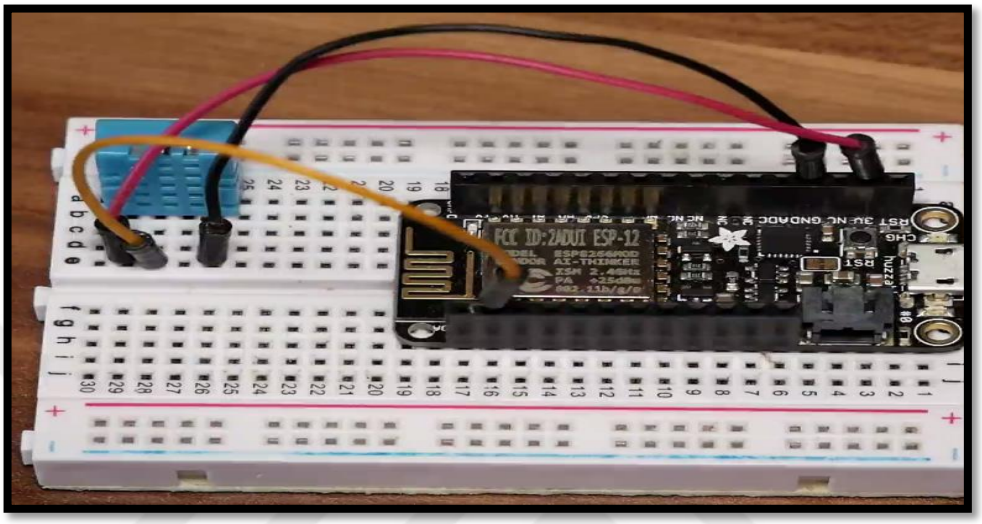

Figure 3.31. Connecting the DHT 11 to the ESP 8266 board

The coding is done using C programming on the Arduino IDE. The targeted server used is dweet.io to send the data to the cloud. There are 3 steps involved: First is the calculation of the temperature and humidity values using the libraries in the Arduino IDE. The second step is to build a request on dweet.io in order to send the data. The last step is to start the connection to the server and to view what the data that are coming back from the server. The coding for the above steps along with detailed comments has been made available in the appendix C. The program is then sent to the ESP board via a USB.

3.4.1 Linking Dweet.io to Freeboard.io

The role of dweet.io is to send the data from the ESP board to the cloud. The role of freeboard.io is to organize the data into a visual display. With freeboard.io as an open source program, we are able to visualize the internet of things by using dashboards and visualizations. The user first has to register to create and account with a valid email address as shown in the figure below.

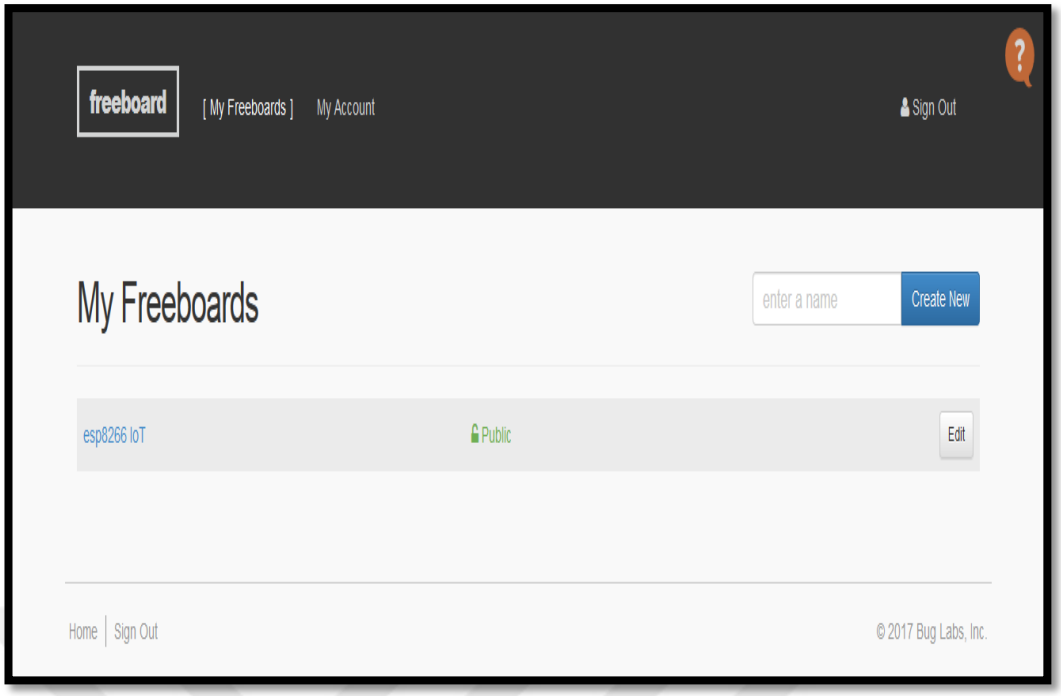

Figure 3.32. Freeboard creation

After clicking on the freeboard, we are greeted with a blank dashboard. We then click on the add data source button and choose dweet.io and mention also the name that we gave to the ESP8266 featherboard from which we want to stream the data from.

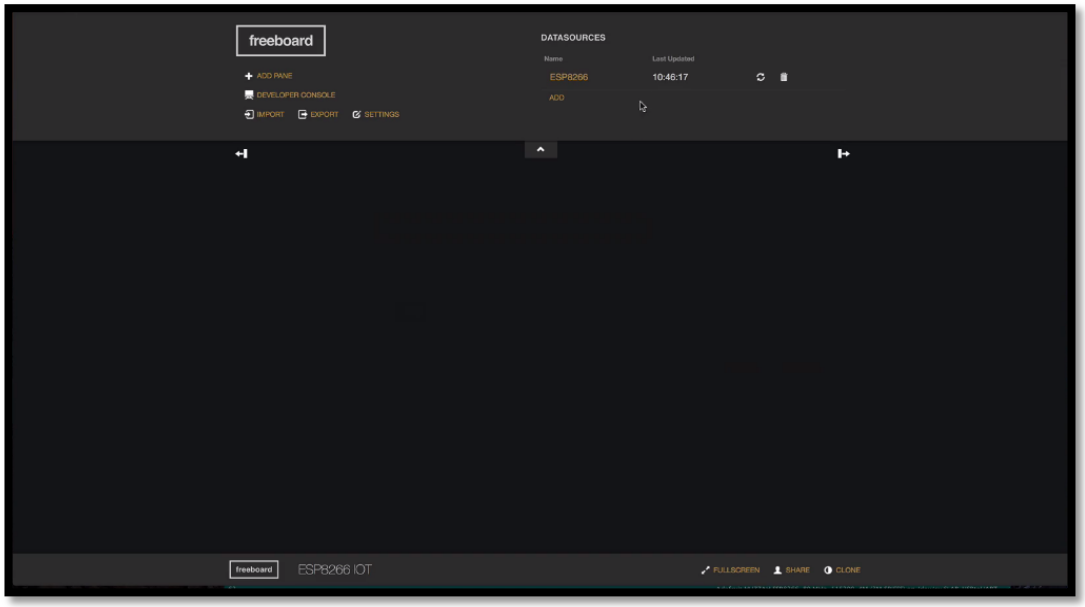

Figure 3.33. Dashboard creation

We then add a widget of type gauge for temperature and the same step is repeated for that of humidity. Then we add another widget of the type Sparkline in order to view how the data varies over time. Figure 3.34 displays how we display the data both instantaneously and over time.

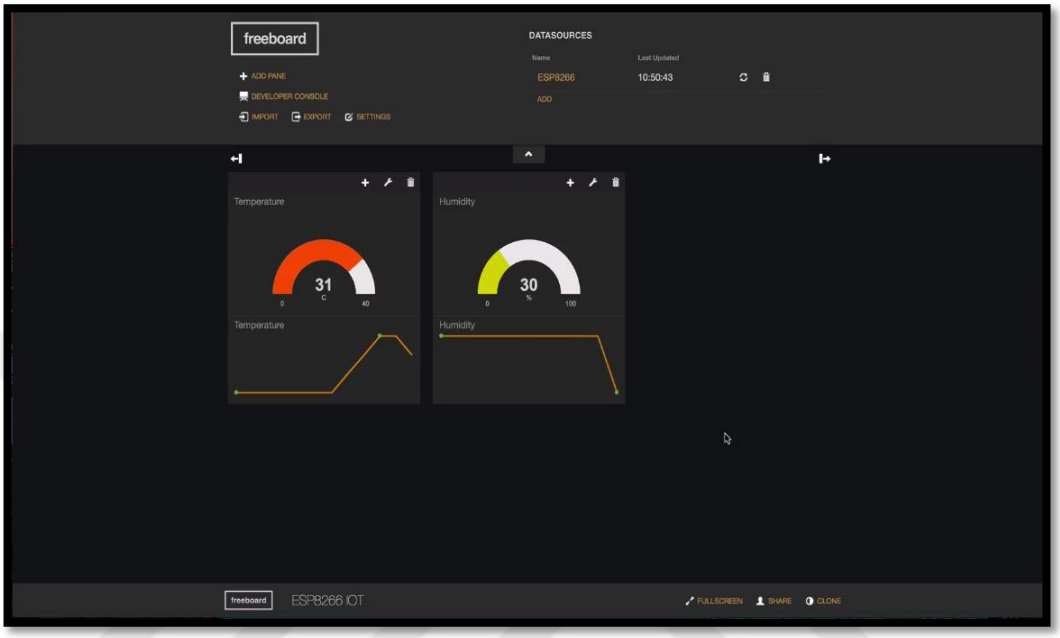

Figure 3.34. Temperature and humidity display on freeboard.io

3.4.2 IFTTT (IF THIS THEN THAT)

IFTTT is a free online service that gives the possibility to create if then conditional loops. We can make applets from this service which is very easy to use. Everything IFTTT can do is accomplished through these applets. These are tiny programs that you can create which use triggers (the “If’s”) to execute actions (the “Then That’s”). An applet can be triggered by the changes that occur within other web services such as Gmail, Facebook, instagram, and Pinterest amongst others. Besides that, it also runs on Android and iOS. According to the official website only in 2015, users created about 20 million applets every day. We can make use of IFTTT in order to monitor the temperature in the smart system and for example if the temperature is greater or equal to 35 degrees C, we can send an SMS to a phone notifying the owner that the temperature in his house is too high thereby warning him against a potential fire.

The first step in doing that is to sign up to the IFTTT service. In order to be able to make the above explained SMS alert system, we are going to make use of Webhooks and SMS from the IFTTT services. The notion of a WebHook is straightforward. A WebHook is an HTTP callback: an HTTP POST that occurs when something happens; a simple event-notification via HTTP POST. Webhooks has been used to send data from the DHT11 sensor to the IFTTT platform. In case the temperature is greater or equal to 35 degrees, an event notification will be sent to the SMS service via HTTP Post. Both Webhooks and the SMS services are going to be configured as shown in figure 3.35 and 3.36 respectively.

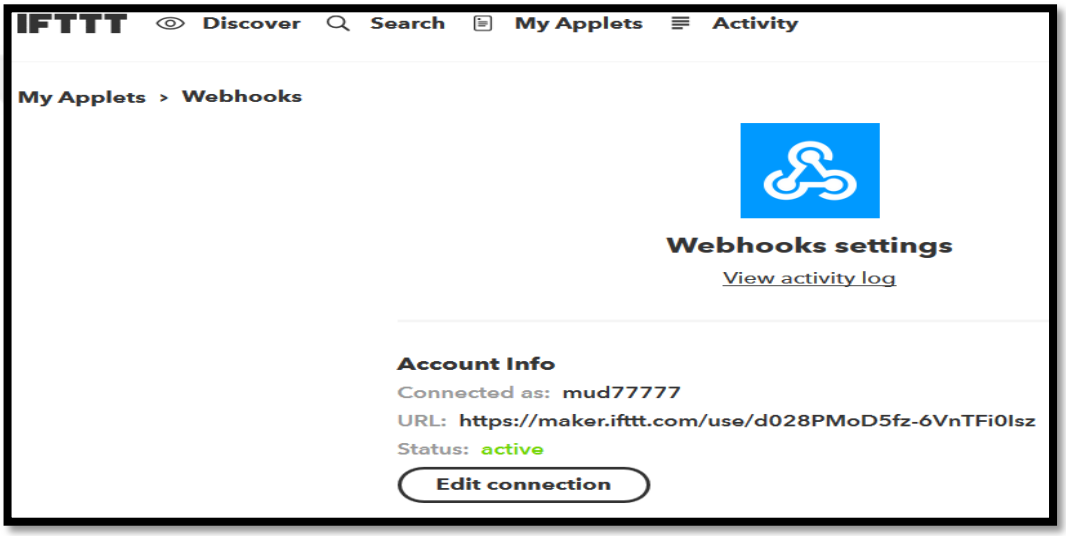

Figure 3.35. Webhooks settings

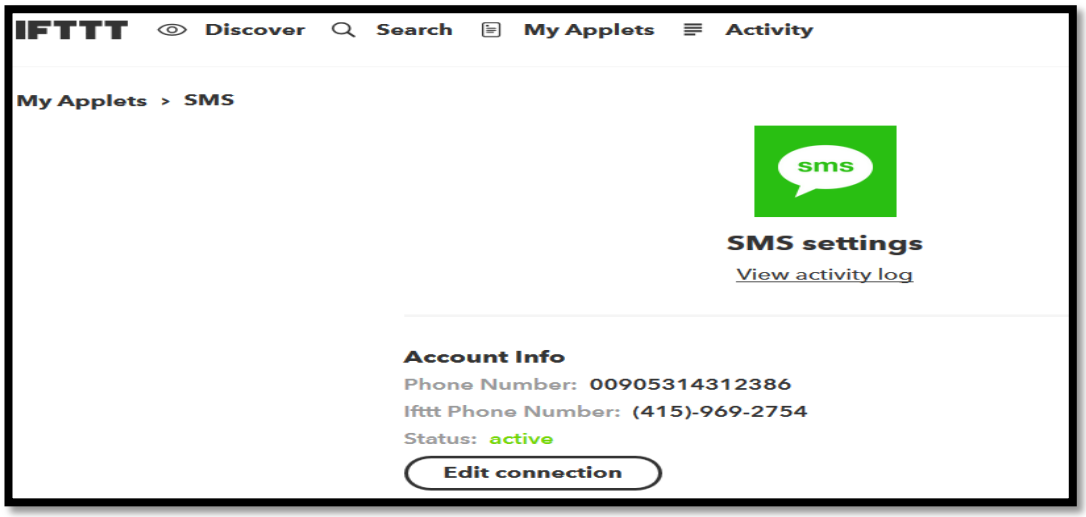

Figure 3.36. SMS settings

In order to make the both of the services work together, we have to make an "applet". We first choose our "this" part as shown in figure 3.37.

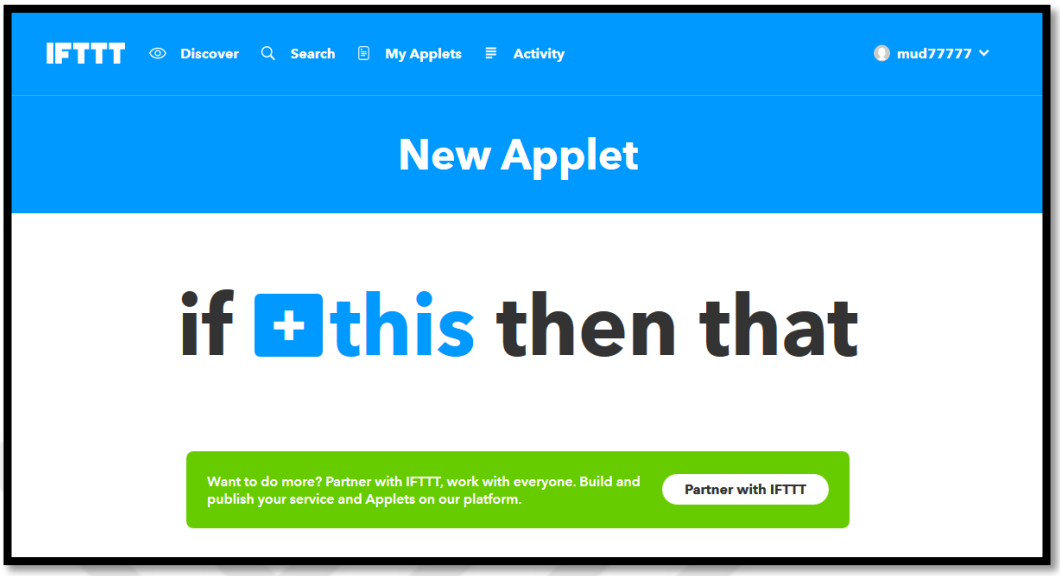

Figure 3.37. Trigger part: "this"

After linking the Webhooks service to the trigger part, we add the action service (SMS) to the part as shown in figure 3.38.

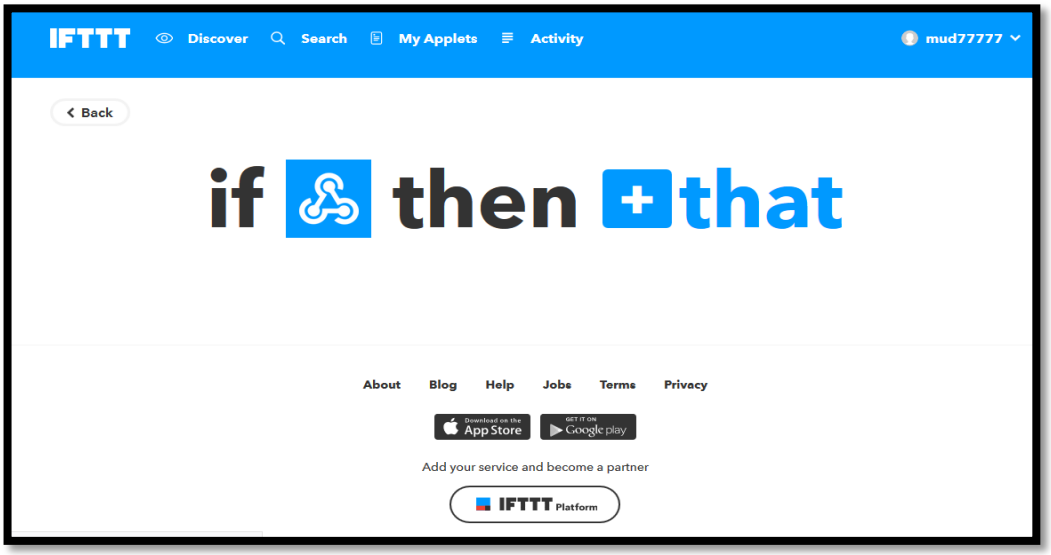

Figure 3.38. Action part: "That"

We then key in the content of the sent SMS in the case of a potential fire and we just click on finish in step 6 to complete the applet.

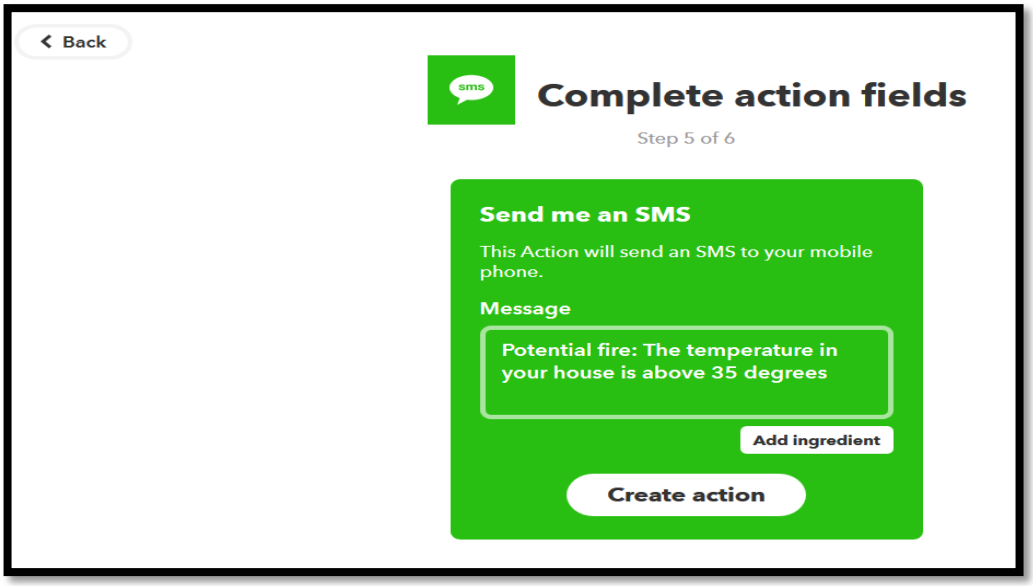

Figure 3.39. Content to be sent as SMS

3.4.3 Summary

For the trigger (this), we made use of the Webhooks service online platform to pass the temperature values to the IFTTT website. We could also make humidity become a trigger in another applet if we decide to further increase the size of the project. However, alternatively we can also choose other services available on the IFTTT services for the action part (That). The other services that can also be linked to the Webhooks are numerous; the common ones being Twitter, E-mail, phone call, Facebook, Skype, and LinkedIn amongst many other as shown in figure 3.40.

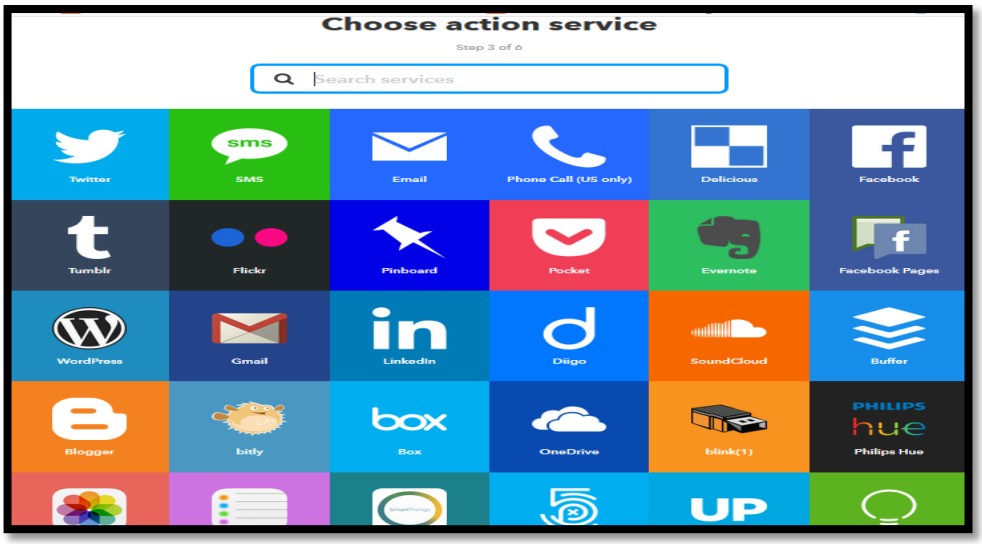

Figure 3.40. Other alternatives for action part (That)

3.5 Android Mobile App Development

The mobile App is the last section of the home automation and monitoring project. The best way to develop Android mobile application is through Android Studio. Android Studio is the software that one can use to develop android mobile Apps and is regularly updated. It is the official integrated development environment (IDE) for Google Android operating system. It is readily available from the internet. The software is always being updated and thus required continual updates. The current stable version of Android studio is the 3.0 which was released in October 2017.

The two languages that are necessary to use Android Studio are: Java and XML. XML is used as the front end while Java is used for the backend. We make use of concept of object oriented programming in Android Studio in order to make the mobile app. Section 1 and 2 of the project are linked to the mobile application via coding. The preview of how the app will look can be seen from the preview section in the Android Studio software as shown in figure 3.41.

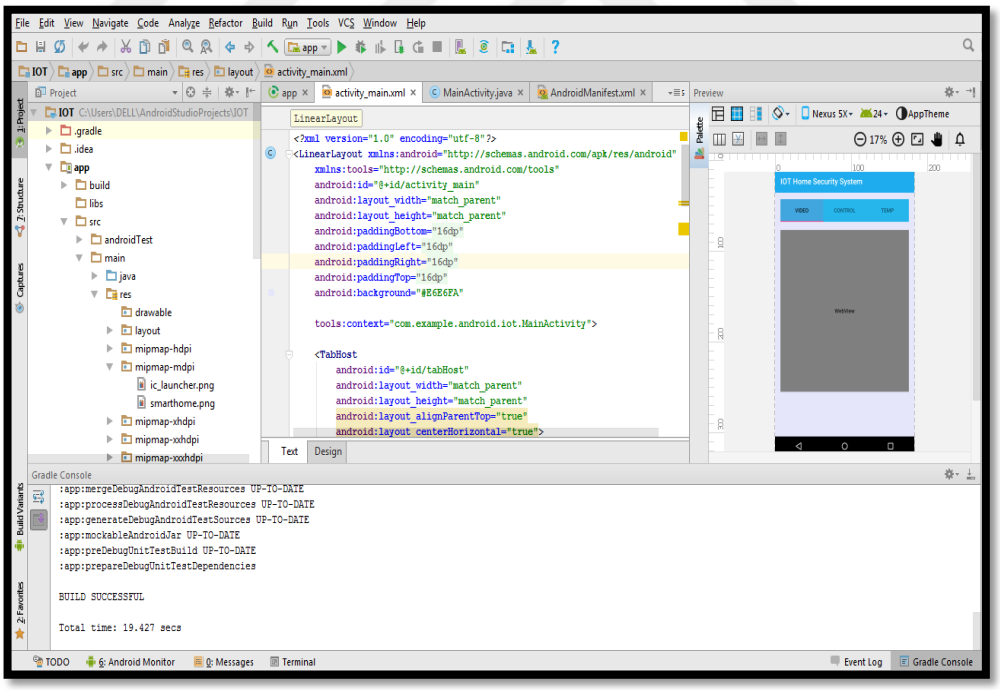

Figure 3.41. Preview of IOT mobile app

The codes from the Android studio can be sent to any other Android device via a USB cable. However, USB debugging in developer options in the connected Android device has to be on in order for installing the Android package kit (APK) as shown in figure 3.42. Furthermore, both the XML and Java coding for the IOT mobile app has been provided in appendix D. An extra bonus for the mobile app is the use of pattern in order to be able to open the programmed IOT mobile app. As such even if the mobile phone gets lost, the user can be assured that his home is still secure.

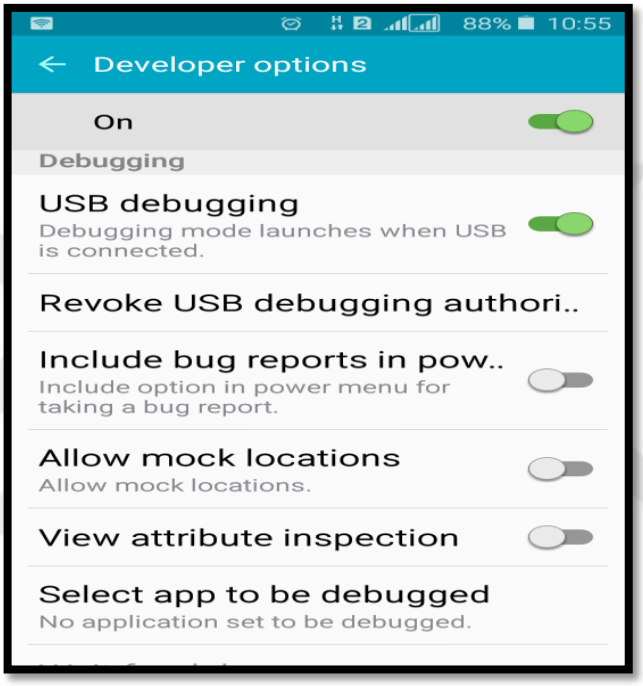

Figure 3.42. USB debugging

CHAPTER 4

RESULTS AND DATA ANALYSIS

4.1 Mobile App

After the codes have been compiled and transferred to the Android mobile device, the apk is installed and we are greeted with our custom made Home automation IOT mobile App as shown in the figure below. We have taken the liberty to beautify the App according to our requirements.

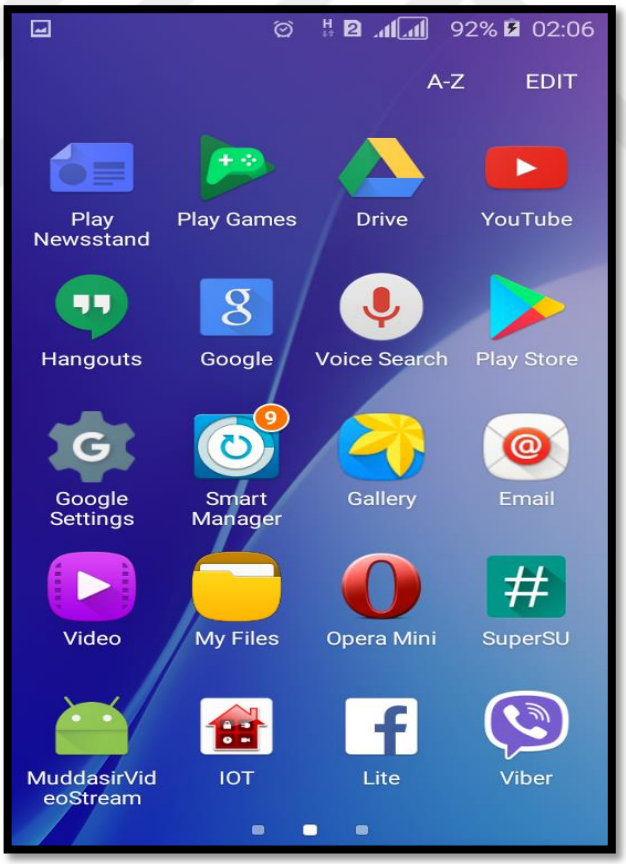

Figure 4.1. IOT home automation Android App

4.2 Mobile App Security

In order to add more security to the IOT home automation App, a security screen lock was added as shown in figure 4.2.

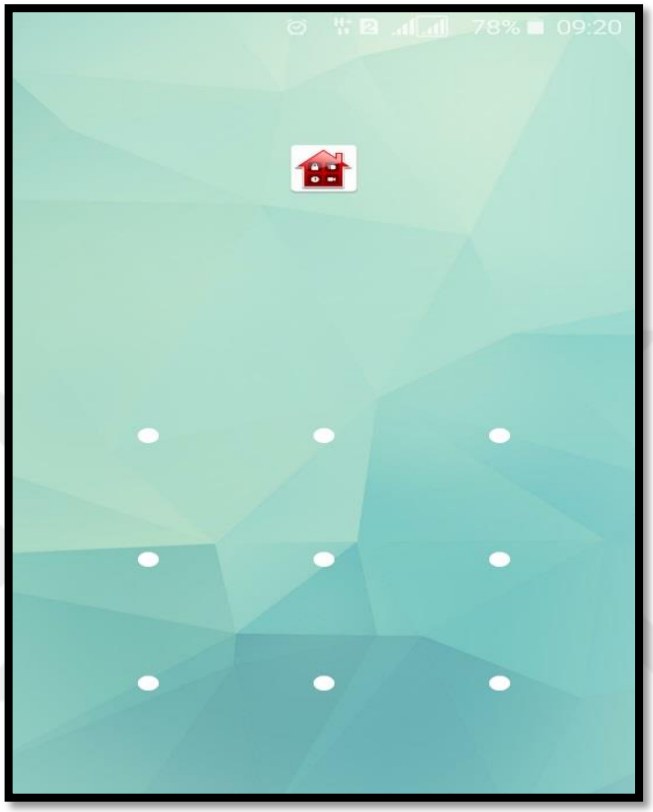

Figure 4.2. Screen lock for IOT app

4.3 Mobile App Camera Streaming

After successfully keying the correct combination for the screen lock, we are greeted with the IOT app that was developed on the Android studio software. The first tab concerns the raspberry pi camera live streaming as shown in figure 4.3 below. The stream is of very high quality with frame rate between 20-30.

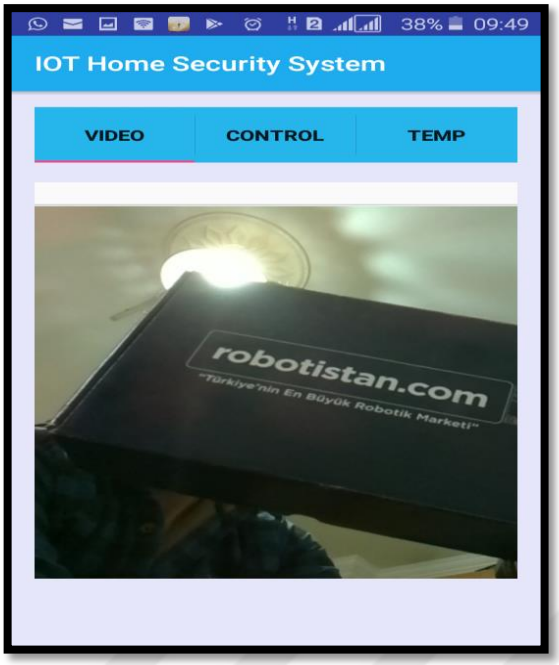

Figure 4.3. Raspberry Pi Camera streaming from IOT mobile App

4.4 Mobile App Power Switch Control

4.4.1 Power Switch Security

When we switch to the control of lighting in the home automation security app, we are first greeted with a username and password as show in figure 4.4.

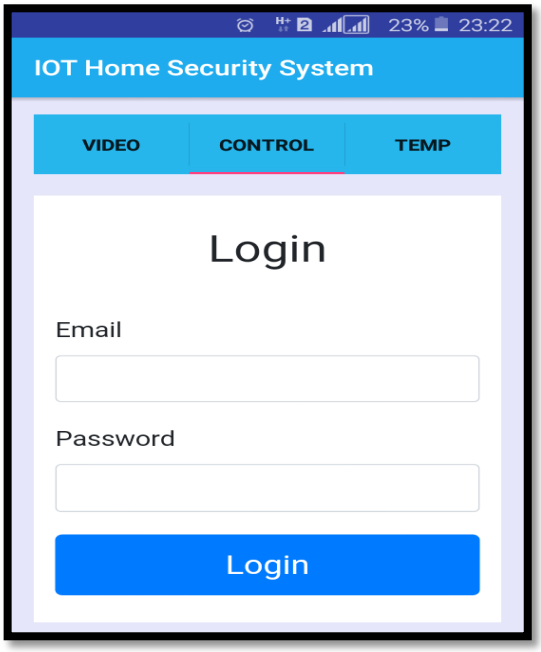

Figure 4.4. Power switch control security

4.4.2 Mobile App Lighting Control

After we input the correct registered email address and password, we are then directed to the control of the power switches as shown in figure 4.5. From there, we are able to control the lights from anywhere in the world. Furthermore, we have an option of further increasing the number of switches to be controlled as per the requirements of any project. Furthermore, the physical hardware has been provided in the last section of the chapter "methodology" with figure 3.24.

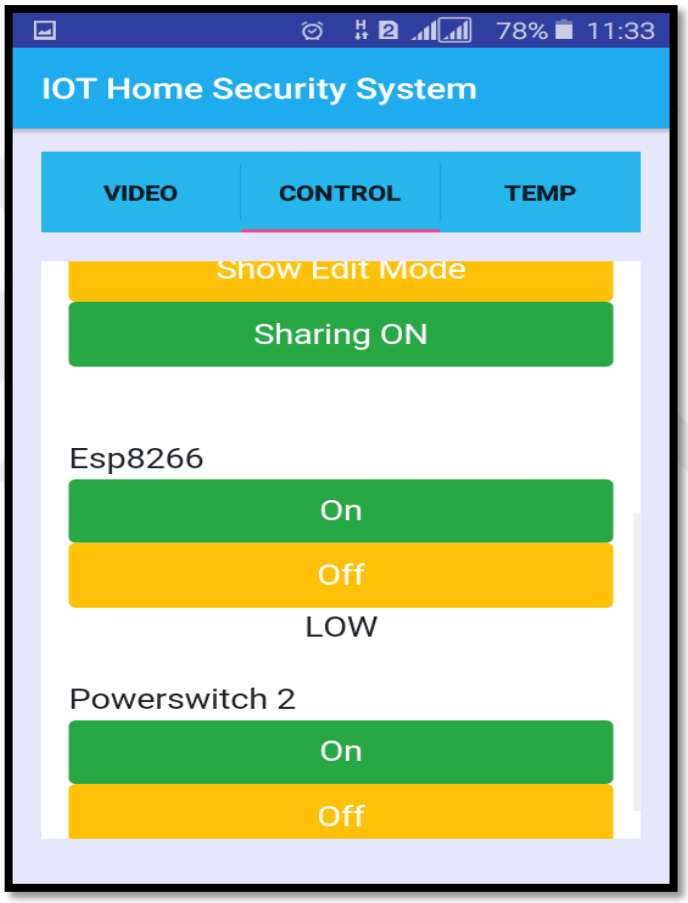

Figure 4.5. Light control from home automation and control App

4.5 Mobile App Atmospheric Parameter Monitoring

The atmospheric parameters namely temperature and humidity are constantly being sent to dweet.io which in turn is feeding the data to freeboard.io as discussed previously. What is needed is to make the webpage linking to the personalized freeboard.io available on the App. This has mainly been done using the XML and Java programming language. The required object used to accomplish this task was by making use of "WebView" layout in the Android Studio development kit. The results can be seen from the figures below.

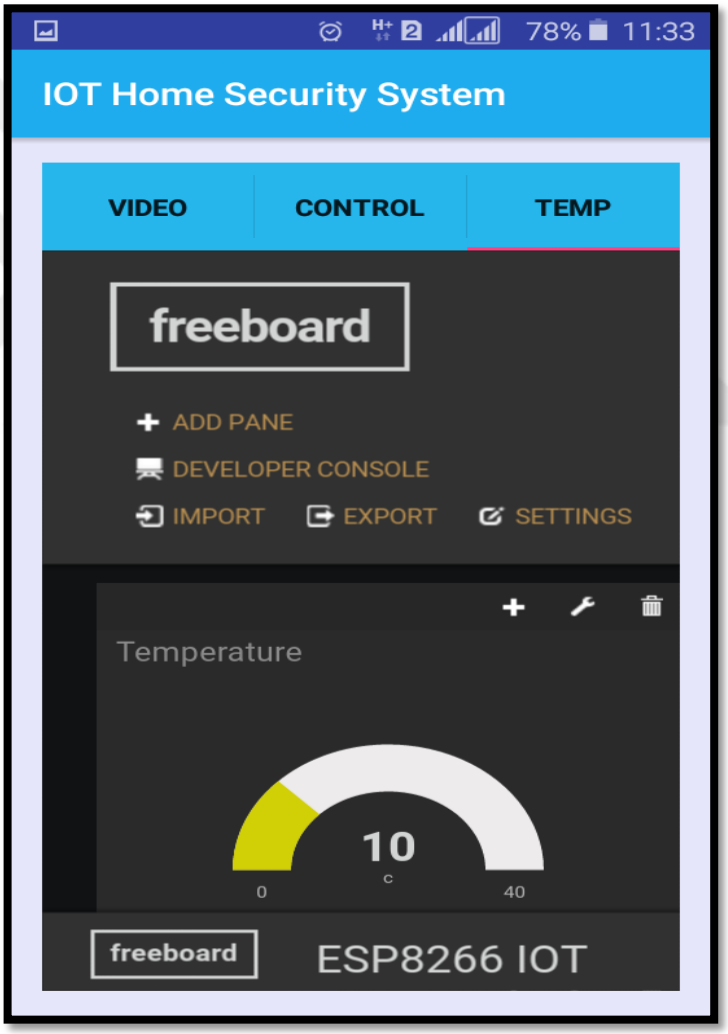

Figure 4.6. Viewing Temperature via the mobile App

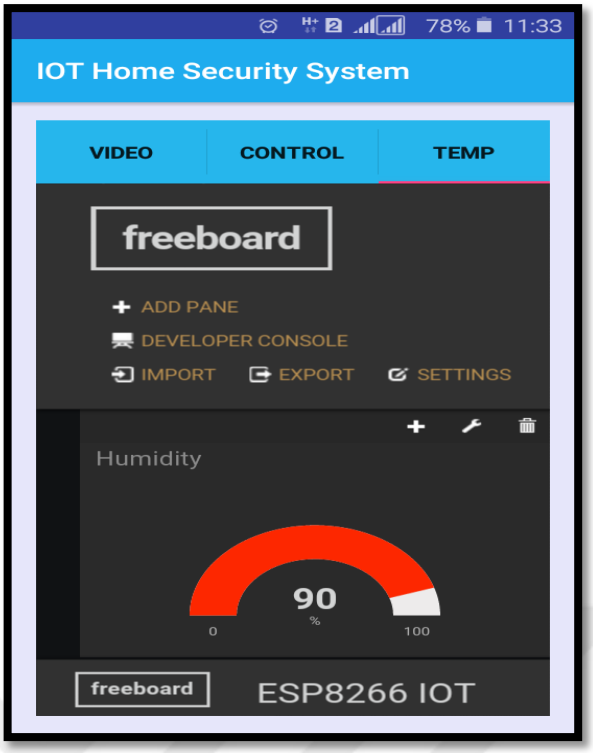

Figure 4.7. Humidity view on mobile App

4.6 Summary

As can be seen from the figures above, all the project objectives have been achieved; Camera streaming, power switch control and atmospheric parameters monitoring. The mobile App has been improved furthermore in terms of security as can be seen with the screen lock, email and password requirements to gain access to the data. Moreover, if one wants to add extra security with the system such as email notification, Skype or phone call, an optional part has been added in the methodology section titled IFTTT (if this then that) explaining in detail on how to achieve this. However, it should be noted that this home automation mobile App can further be developed and made more complex by adding other devices one wants to control. This said, we can conclude that one is allowed more space to give more flexibility to the project should one want to build on this idea.

CHAPTER 5

DISCUSSION

As previously discussed, IOT has emerged as one of the most trending technology for home automation in the very near future. And for this reason, a lot of multinational corporations have already started investing heavily in the development of smart objects that will be interacting over the web and controlled by the user. The cost of these gadgets are going to be quite pricy as per the market trend and the more gadgets a house is equipped with, the more expensive it becomes to monitor and control them. The project that we have designed is an alternative way of developing scalable and flexible home automation IOT system to reach out to the mass middle class at a reasonable price. However, there are always upgrades that can be done to further enhance the overall suitability and marketability of the project. With this in mind, we are going to discuss about possible upgrades and ideas that can be done in order to improve the overall system should another researcher try to build on the project.

5.1 Include Motion Sensor With Raspberry Pi To Take Picture

Figure 5.1. Perimeter detection through motion sensor

From figure 5.1, we can see that a motion sensor can be used as the first line of defense by the system by detecting any intruder's presence in the house. This is widely referred to as perimeter detection. The suggested motion detection device is a PIR motion sensor as can be seen in the figure below.

Figure 5.2. PIR motion sensor

Whenever the PIR sensor detects motion, a signal is sent to the raspberry pi which in turn can be used to trigger the camera to take a snapshot. Another suggestion might be to store these pictures directly on the cloud. This first suggestion will further enhance the system. However, this might also overload the system in terms of its complexity. It could however be broken into smaller parts and added to the system.

5.2 Make Extensive Use Of IFTTT To Incorporate

The motion sensor and camera snapshot might be used as a first line of defense. As explained in the methodology section in the optional part for IFTTT (If this then that), I would suggest making a new recipe for whenever motion is detected, Webhooks could be used to perform an action. The action ("That part") could be amongst the following:

- Post on twitter
- Post on facebook

- SMS notification
- Phone call (US only)
- LinkedIn notification
- Email notification
- Tumblr notification
- OneDrive notification

The list is long but the action depends on the user's choice. There is a wide range of actions beside the ones mentioned that can be done. As interesting as it may sound, a detailed explanation has been given on the IFTTT part in the section titled "IFTTT" in the methodology chapter. This idea can be used as a second line of defense for the system.

5.3 Make Use Of Multiple Cameras

There could be the use of multiple cameras located in different locations of the property that is being monitored. All the cameras could then be linked to only one server. Though the project gets even more complex, this could be achieved with a group of people each one assigned with a different task. The mobile app would also have to show the multiple streams from the server.

5.4 Mobile App On Iphone

Android has the largest market share of mobile phones users in the world but there is also a large section of people that prefer IPhone. Apple's IDE (Integrated Development Environment) for both Mac and *iOS apps* is Xcode. Xcode is the graphical interface that is used to write the App. The coding is done using Apple's new Swift programming language. As such, this would increase the overall flexibility of the system by making the App available to more people. However, there are other technical details also that have to be linked to it and that is why the last suggestion might be the best and most business oriented one as it will considerably simplify the designing of such a system.

5.5 Customized Website And Database Development/ Marketability

In order to reach the huge mass, the project has to be equipped with more centrally controlled websites and databases. Each customer can have their custom login to their database on the website. The website has to be made responsive and any mobile App could be made more easily. There are lots of IT companies which develop customized websites for their customers and getting such a website done professionally will only help in attracting the mass to make use of the IOT products. We already have that trend with the international companies but that is not a disadvantage as different countries have different rules and personally speaking, this is the beginning of innovation in a country. If one promotes their own products, then it will keep on developing in the sense that the next step will be to develop the gadgets that are associated with the system. It is also good to know that the system is tailor made that is it is totally dependent on the user to choose what to include. IOT in this sense will help to increase the number of startup businesses as well as leasing other smaller development sections to other companies thereby creating employment in a country.

5.6 Summary

A lot has been said about the massive revenues expected from IoT products as described in chapter 1. However, how to do business with such projects has been discussed above. IoT will help to boom the economies of several countries creating jobs and better living conditions for people at an affordable price. As such, new startup businesses in the IoT field will bloom.

CHAPTER 6

CONCLUSION

This thesis provided an insight on how we can make use of the Internet of things in order to make a flexible and scalable home automation and monitoring system. The latest technology and trends were used in order to come up with the idea of making an Android mobile App in order to view and control all the connected appliances via the Internet. The project was broken into 3 subsections namely live video camera streaming, control of electronic appliances and development of a user friendly Android mobile application.

As for the programming languages used, Java, XML, embedded C, Linux and HTML were all involved in the development of the project. How machines are able to "dweet" information was also reviewed. Making use of IoT platforms such as dweet.io, arest dashboard and freeboard.io showed how to make use of open source software in order to accomplish the goals and objectives of the project. Furthermore, in order to optimize the project even further, IFTTT was also used where an action could easily be linked with any mobile App such as SMS, Skype, email, and twitter amongst many more others. As such, any user can determine how he wants to customize his home automation and monitoring system hence legitimizing the name "scalable and flexible".

However in order to further provide more security to the already robust system, a screen lock and password to log in was also added so as to protect the user's security in case their mobile phones got lost or stolen. The advantage of such a system over the previous systems mentioned in the literature review is very clear: The use of the latest cutting edge technology (IoT), tweets, the use of scalable dashboards, the latest ESP8266 board combined with the raspberry pi 3 all communicating over the internet and linked to one mobile phone App shows the immensity and complexity of

the project accomplished. Research in both hardware and software were intensive and the coding process was lengthy. The importance of the ESP8266 feather board chip was also highlighted as it is able to connect to Wi-Fi without having to buy extra components and also easily reprogrammable via USB. The price and quality of the products used are the best that are available on the market at present.

Looking from a business perspective, we can easily see how the number of startups in the field of IoT and services will rise in the close future. This will be very beneficial for countries that provide loan facilities to these startups to give the best services to the public at a competitive price. As such, IoT in the field of home automation and monitoring will further increase the quality and security of lives of the people.

Startups can make use of this business opportunity opening to expand by first making their own dashboards capable of hosting about a thousand customers. This type of venture may need to lease out part of the project to software companies to make a database according to the requirements of the startup. Options to add, remove and upgrade services along with new features could also be added. This might also be a source of assistance for the governmental services like the firefighters and the police in the case of crime or incidents.

REFERENCES

[1] Aazam, M., Khan, I., Alsaffar, A. A., Huh, E. N. (2014). Cloud of Things: Integrating Internet of Things and Cloud Computing and The Issues Involved, *International Bhurban Conference on Applied Sciences & Technology Islamabad*, **11**, 414-419.

[2] Alheraish, A. (2004). Design and implementation of home automation system, *IEEE Transactions on Consumer Electronics*, **50**, 1087-1092.

[3] Ali, U., Nawaz, S. J., Jawad, N. (2006). A Real-time Control System for Home/Office appliances automation, from mobile device through GPRS network, *IEEE International Conference on Electronics, Circuits and Systems*, **13**, 854-857.

[4] Breached webcam and baby monitor site flagged by watchdogs. 2014. Available at: https://www.bbc.com/news/technology-30121159. Accessed 16.08.2018.

[5] Danaher, M., Nguyen, D. (2002). Mobile Home Security with GPRS, *Symposium for Information Science*, **8**, 377-380.

[6] Das, S. R., Chita, S., Peterson, N., Shirazi, B. A., Bhadkamkar, M. (2011). Home Automation and Security For Mobile Devices, *IEEE International Conference on Pervasive Computing and Communications Workshops*, 141-146.

[7] Delgado, A. R., Picking, R., Grout, V. (2006). Remote-Controlled Home Automation Systems with Different Network Technologies, *International Network Conference*, **6**, 357-366.

[8] Evans, D. (2011). The Internet Of Things: How The Next Evolution Of The Internet is Changing Everything, *Computer Information System Company White Paper*, **1**, 1-11.

[9] How Much Does a Smart Home Really Cost. 2015. Available at: https://www.makeuseof.com/tag/much-smart-home-really-cost/. Accessed 16.08.2018.

[10] Internet Of Things Market To Reach $267B By 2020. 2017. Available at: https://www.forbes.com/sites/louiscolumbus/2017/01/29/internet-of-things-market-to-reach-267b-by-2020/#5915137b609b. Accessed 16.08.2018.

[11] Islam, T., Mukhopadhyay, S. C., Suryadevara, N. K. (2017). Smart Sensors and Internet Of Things: A Postgraduate Paper, *IEEE Sensors Journal*, **17**, 577-584.

[12] Jin, J. S., Jin, J., Wang, Y. H., Zhao, K., Hu, J. J. (2008). Development Of Remote-Controlled Home Automation System With Wireless Sensor Network, *IEEE International Symposium*, **5**, 169-173.

[13] Jose, A. C., Malekian, R. (2015). Smart home automation security. *Smart Computing Review*, **5**, 269-285.

[14] Kanma, H., Wakabayashi, N., Kanazawa, R., Ito, H. (2003). Home appliance control system over Bluetooth with a cellular phone. *IEEE Transactions on Consumer Electronics*, **49**, 1049-1053.

[15] Khiyal, M. S., Khan, A., Shehzadi, E. (2009). SMS Based Wireless Home Appliance Control System for Automating Appliances and Security. *Issues in Informing Science & Information Technology*, **6**, 887-894.

[16] Muhury, L., Habib, A. A. (2012). Device control by using GSM network, *15th International Conference on Computer and Information Technology*, 271-274.

[17] Roundup Of Internet Of Things Forecasts And Market Estimates. 2016. Available at: https://www.forbes.com/sites/louiscolumbus/2016/11/27/roundup-of-internet-of-things-forecasts-and-market-estimates-2016/#cc8caaa292d5. Accessed 16.08.2018.

[18] Ryan, M. (2013). Bluetooth: With Low Energy Comes Low Security, *7th Workshop on Offensive Technologies*, **13**, 4-4.

[19] Saeed, U., Syed, S., Qazi, S. Z., Khan, N., Khan, A., Babar, M. (2010). Multi-Advantage and Security Based Home Automation System, *In UKSim Fourth European Modelling Symposium on Computer Modelling and Simulation*, 7-11.

[20] Sriskanthan, N. T. (2002). Bluetooth Based Home Automation System, *Microprocessors and Microsystems*, **26**, 281-289.

[21] The Economist Special Report. 2010. Available at: https://www.economist.com/special-report/2010/11/04/augmented-business. Accessed 16.08.2018.

[22] Tsai, S. M., Yang, P. C., Wu, S. S., Sun, S. S. (1998). A Service Of Home Security System On Intelligent Network, *IEEE Transactions on Consumer Electronics*, **44**, 1360-1366.

[23] Wu, B. F., Peng, H. Y., Chen, C. J. (2006). A Practical Home Security System Via Mobile Phones, *5th International Conference on Telecommunications and Informatics*, **5**, 1061-1066.

[24] Yang, L., Yang, S. H., Yao, F. (2006). Safety and Security Of Remote Monitoring and Control Of Intelligent Home Environments, *IEEE International Conference on Systems, Man and Cybernetics*, **2**, 1149-1153.

APPENDICES

Appendix A: Camera Streaming Programming Codes

```
//HTML FILE

<!DOCTYPE html>

<html>



<head>

  <script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">

  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

  <script src="js/script.js"></script>

  <link rel="stylesheet" href="css/style.css">

  <meta name="viewport" content="width=device-width, initial-scale=1">

</head>



<script type="text/javascript">

/* Copyright (C) 2007 Richard Atterer, richardÂ©atterer.net

   This program is free software; you can redistribute it and/or modify it
```

```
   under the terms of the GNU General Public License, version 2. See the file
   COPYING for details. */
var imageNr = 0; // Serial number of current image
var finished = new Array(); // References to img objects which have finished downloading
var paused = false;


function createImageLayer() {
  var img = new Image();
  img.style.position = "absolute";
  img.style.zIndex = -1;
  img.style.width = "100%";
  img.onload = imageOnload;
  img.onclick = imageOnclick;
  img.src = "http://192.168.0.105:8080/?action=snapshot&n=" + (++imageNr);
  var webcam = document.getElementById("webcam");
  webcam.insertBefore(img, webcam.firstChild);
}


// Two layers are always present (except at the very beginning), to avoid flicker
function imageOnload() {
  this.style.zIndex = imageNr; // Image finished, bring to front!
  while (1 < finished.length) {
```

```
    var del = finished.shift(); // Delete old image(s) from document
    del.parentNode.removeChild(del);
  }
  finished.push(this);
  if (!paused) createImageLayer();
}

function imageOnclick() { // Clicking on the image will pause the stream
  paused = !paused;
  if (!paused) createImageLayer();
}

</script>

<body onload="createImageLayer();">

<div id="container">

  <h3>Remote Monitoring</h3>

  <div class='voffset50'></div>

  <div class='row voffset50'>
```

```
    <div id="webcam" class='col-md-12'>
        <noscript>
          <img src="http://192.168.0.105:8080/?action=snapshot" class="img-
responsive"/>
        </noscript>
    </div>
  </div>

</div>

</body>
</html>
//==========================================================
// (Interface.js)Modules
var express = require('express');

// Express app
var app = express();

// Use public directory
app.use(express.static('public'));

// Routes
```

```
app.get('/', function (req, res) {

  res.sendfile(__dirname + '/public/interface.html');

});

// Start server
app.listen(3000, function () {
  console.log('Remote monitor system started');
});
```

Appendix B: Power Switch Program

```
// Import required libraries
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <aREST.h>

// Clients
WiFiClient espClient;
PubSubClient client(espClient);

// Create aREST instance
aREST rest = aREST(client);
```

```
// Unique ID to identify the device for cloud.arest.io
char* device_id = "98f6cs";


// WiFi parameters
const char* ssid = "Jarex_5A";
const char* password = "connect1337";


// Functions
void callback(char* topic, byte* payload, unsigned int length);


void setup(void)
{

  // Start Serial
  Serial.begin(115200);


 // Set callback
 client.setCallback(callback);


 // Give name and ID to device
 rest.set_id(device_id);
 rest.set_name("power_switch");
```

```
  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Set pin 5 to output
  pinMode(5, OUTPUT);

}

void loop() {

  // Connect to the cloud
  rest.handle(client);

}

// Handles message arrived on subscribed topic(s)
void callback(char* topic, byte* payload, unsigned int length) {
```

```
 // Handle
  rest.handle_callback(client, topic, payload, length);

}
```

Appendix C: Data Logger (Dweet.Io) Program

```
//DATA LOGGER DWEET.IO
// Import required libraries
#include "ESP8266WiFi.h"
#include "DHT.h"

// WiFi parameters
const char* ssid = "Jarex_5A";
const char* password = "connect1337";

// Pin
#define DHTPIN 5

// Use DHT11 sensor
#define DHTTYPE DHT11

// Initialize DHT sensor
DHT dht(DHTPIN, DHTTYPE, 15);
```

```
// Host
const char* host = "dweet.io";

void setup() {

  // Start Serial
  Serial.begin(115200);
  delay(10);

  // Init DHT
  dht.begin();

  // We start by connecting to a WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {

  Serial.print("Connecting to ");
  Serial.println(host);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }

  // Reading temperature and humidity
  int h = dht.readHumidity();
  // Read temperature as Celsius
  int t = dht.readTemperature();
```

```
  // This will send the request to the server

  client.print(String("GET /dweet/for/myesp8266?temperature=") + String(t) +
"&humidity=" + String(h) + " HTTP/1.1\r\n" +

              "Host: " + host + "\r\n" +

              "Connection: close\r\n\r\n");

  delay(10);


  // Read all the lines of the reply from server and print them to Serial

  while(client.available()){

    String line = client.readStringUntil('\r');

    Serial.print(line);

  }


  Serial.println();

  Serial.println("closing connection");


  // Repeat every 10 seconds

  delay(10000);


}
```

Appendix D: Mobile App Development Program

```
//XML CODES
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="#E6E6FA"

    tools:context="com.example.android.iot.MainActivity">

    <TabHost
        android:id="@+id/tabHost"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
            <TabWidget
                android:id="@android:id/tabs"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="#26b6ec"></TabWidget>

            <FrameLayout
                android:id="@android:id/tabcontent"
                android:layout_width="match_parent"
```

```
    android:layout_height="match_parent">

    <LinearLayout
        android:id="@+id/video"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">


        <WebView
            android:layout_marginTop="20dp"
            android:layout_width="match_parent"
            android:layout_height="400dp"
            android:id="@+id/webView"/>

    </LinearLayout>

    <LinearLayout
    android:id="@+id/control"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">


    <WebView
        android:layout_marginTop="20dp"
        android:layout_width="match_parent"
        android:layout_height="400dp"
        android:id="@+id/webView2"/>
</LinearLayout>
    <LinearLayout
        android:id="@+id/temp"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
```

```
            android:orientation="vertical"

            >


            <WebView
              android:layout_marginTop="0dp"
              android:layout_width="1000dp"
              android:layout_height="match_parent"
              android:id="@+id/webView3"/>
          </LinearLayout>



        </FrameLayout>
      </LinearLayout>
    </TabHost>

</LinearLayout>


//-----------------------------------------------------------------------------------------------

//JAVA CODES

package com.example.android.iot;


import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.TabHost;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        WebView webView =(WebView)findViewById(R.id.webView);
        if (webView != null) {
            webView.loadUrl("http://192.168.43.134:8080/stream.html"); // Insert
Your localhost link here.
            //
        }
        else Toast.makeText(this,"Error loading
link",Toast.LENGTH_LONG).show();
        //-----------------------------------------------------------------------
        WebView webView2 =(WebView)findViewById(R.id.webView2);

        if (webView2 != null) {

webView2.loadUrl("https://dashboard.arest.io/boards/Ddbv7AnpehjC8LGYv");
// Insert Your localhost link here.
            //
        }
        else Toast.makeText(this,"Error loading
link",Toast.LENGTH_LONG).show();

        WebSettings webSettings = webView2.getSettings();

        webSettings.setJavaScriptEnabled(true);
        //-------------------------------------------------
        WebView webView3 =(WebView)findViewById(R.id.webView3);

        if (webView3 != null) {
```

```
        webView3.loadUrl("https://freeboard.io/board/ls0taF"); // Insert Your
localhost link here.
        //
    }
    else Toast.makeText(this,"Error loading
link",Toast.LENGTH_LONG).show();
    WebSettings webSettings1 = webView3.getSettings();

    webSettings1.setJavaScriptEnabled(true);



    //-------------------------------------------------

    TabHost host = (TabHost)findViewById(R.id.tabHost);
    host.setup();
    //host = (TabHost) findViewById(R.id.tabHost); (REPETITION HERE)
    //host.setup(); (REPETITION HERE)

    //Tab 1
    TabHost.TabSpec spec = host.newTabSpec("Tab One");
    spec.setContent(R.id.video);
    spec.setIndicator("Video");
    host.addTab(spec);

    //Tab 2
    spec = host.newTabSpec("control");
    spec.setContent(R.id.control);
    spec.setIndicator("control");
    host.addTab(spec);

    //----------------------------------------
    //Tab 3
    spec = host.newTabSpec("temp");
    spec.setContent(R.id.temp);
```

```
        spec.setIndicator("temp");
        host.addTab(spec);
        //----------------------------------------

        /*
       !!!!!!   A WAY TO STREAM MPEG ONTO videoView.
        try {
            String link="http://192.168.43.39:8080/stream.html";
            VideoView videoView = (VideoView) findViewById(R.id.videoView);
            MediaController mediaController = new MediaController(this);
            mediaController.setAnchorView(videoView);
            Uri video = Uri.parse(link);
            videoView.setMediaController(mediaController);
            videoView.setVideoURI(video);
            videoView.start();
        } catch (Exception e) {
            // TODO: handle exception
            Toast.makeText(this, "Error connecting", Toast.LENGTH_SHORT).show();
        }
        */



    }
}
//try to make tab scrollable



//==============================================

//MANIFEST CODES

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.iot">
```

```
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/smarthome"
        android:label="IOT Home Security System"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

//MOBILE APP CUSTOMISED COLORS CODES

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#1eacee</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```