Global Journal of Computer Science and Technology: E Network, Web & Security
Volume 20 Issue 3 Version 1.0 Year 2020
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 ISSN Print 0975-4350

Smart Air Conditioner using Internet of Things

By Khaloud Bati AL-Sa'idi & Dr. Vladimir Dyo
University of Bedfordshire

Abstract- The local remote control is the traditional mechanism in which the end user controls the air conditioner. In the absence of this mechanism, the user loses the control. This thesis aimed to design and implement a smart air conditioner using Internet of Things (IoT) technology. Recent Literatures were reviewed to select the most optimal platform to design and implement the project. The design of the project was then developed based on the selected platform. The project was then implemented and tested successfully. In order to validate the project, a questionnaire was carried out by potential users who tested the product on their SANYO air conditioner. All potential users were able to control their air conditioner remotely over the internet from anywhere. The smart air conditioner has absolutely no inference against real remote control. The product is cost effective, energy efficient and achieves the required automation functionality.

GJCST-E Classification: C.2.6

Strictly as per the compliance and regulations of:

Smart Air Conditioner using Internet of Things

Khaloud Bati AL-Sa'idi [α] & Dr. Vladimir Dyo [σ]

Abstract- The local remote control is the traditional mechanism in which the end user controls the air conditioner. In the absence of this mechanism, the user loses the control. This thesis aimed to design and implement a smart air conditioner using Internet of Things (IoT) technology. Recent literatures were reviewed to select the most optimal platform to design and implement the project. The design of the project was then developed based on the selected platform. The project was then implemented and tested successfully. In order to validate the project, a questionnaire was carried out by potential users who tested the product on their SANYO air conditioner. All potential users were able to control their air conditioner remotely over the internet from anywhere. The smart air conditioner has absolutely no inference against real remote control. The product is cost effective, energy efficient and achieves the required automation functionality.

I. Introducton

As it is known, the usual and traditional mechanism in which the end user controls the air conditioner is through local remote control. However, what if the local remote control is lost, broken, out of batteries or no longer available due to whatever faulty?

On the other hand, what if the air conditioner is forgotten on due to human nature and no one is available to turn it off?

How about controlling the temperature degree of your air conditioner while you are actually away?

How about having a smart air conditioner that would be able to turn off by itself when people are not present and save energy?

Reaching this point, Automation feature seems the best logical solution to handle and control the air conditioner remotely.

Nowadays, Internet of Things (IoT); an emerging technology has risen in the digital realm. The original idea of Internet of Things was proposed at the end of 1990's. IoT is much more related to the wireless sensors networks, mobile communications networks and Internet. IoT can be defined as a network that connects every existing physical object in the world to a unique address in order to provide quick and smart services. In contrast to traditional Internet which interconnects intelligent physical objects only, IoT interconnects both intelligent and non intelligent physical objects due to the availability of object sensing layer (Ma, 2011).

Hence, with Internet of Things, you may control everything using internet service. More specifically, through Internet of Things technology, you will be able to remotely control your air conditioner which will be connected to the internet from anywhere.

Internet of Things technology uses cost effective, powerful and small size device that is considered to be a small size single board computer called Raspberry Pi. Raspberry Pi was developed by Raspberry Pi Foundation. There are four different types of Raspberry Pi such as: Original Raspberry Pi, Raspberry Pi, Raspberry Pi 2 and Raspberry Pi 3. The four different types come with both Model A and Model B flavors. Different platforms can be used as an Operating System for the Raspberry Pi such as: RISC OS, Arch Linux, Pidora, Raspbian and Microsoft Windows 10 IoT core (Harrington, 2015).

The aim of this project is to design and implement a smart air conditioner using Internet of Things technology using Raspberry Pi 3 Model B device.

The aim of this project is accomplished through fulfilling pre-defined objectives. Starting with reviewing related home automation system literatures. Moving to selecting the most suitable platform (Raspbian: the most popular platform used with Raspberry Pi or Windows 10 IoT core: the new platform developed by Microsoft) to design and implement the smart air conditioner. Testing, validating and exploring the gained features of the product are the final step towards accomplishing the project aim.

The features of the designed smart air conditioner were decided through testing the implemented product by potential users. The smart air conditioner has absolutely no inference against real remote control. The product is cost effective, energy efficient and achieves automation functionality indeed.

Each implemented project must involve intellectual challenges. Apparently, there are implemented air conditioner projects using Internet of Things raspberry Pi with different web and mobile enabled applications. However, in this project the web application is developed using PHP web language and MySQL database engine which are not used by any of the developed projects.

Internet of Things means any physical object is connected to the internet. In this system, a smart air conditioner which can be controlled remotely through a web application is to be implemented. However, in order to control the air conditioner remotely, it must be connected to the internet in the first place. Obviously, the air conditioner does not have any internet connection port. Hence, it will be connected to the raspberry pi 3 that has the required internet connection

Author α σ: MSc. (Computer Networking), University of Bedfordshire, 2016. e-mail: khaloud.al-saidi@study.beds.ac.uk

port. The connection between the air conditioner and the raspberry pi 3 is through the Infra-Red (IR) transmitter which is a Lite Emitting Diode (LED) emitting Infra-Red lights (connected to the raspberry pi 3) and Infra-Red (IR) receiver. Then, the internet connection port in raspberry pi 3 will be connected to Wi-Fi hot spot in order to get internet service. On the other hand, any physical device such as: a desktop, a laptop, a PAD and a smart phone which has a web browser in order to use the implemented web application is connected to the internet from anywhere to control the air conditioner remotely. Furthermore, the implemented air conditioner would be able to turn off by itself when people are not present.

The following figure illustrates the above mentioned project specifications.

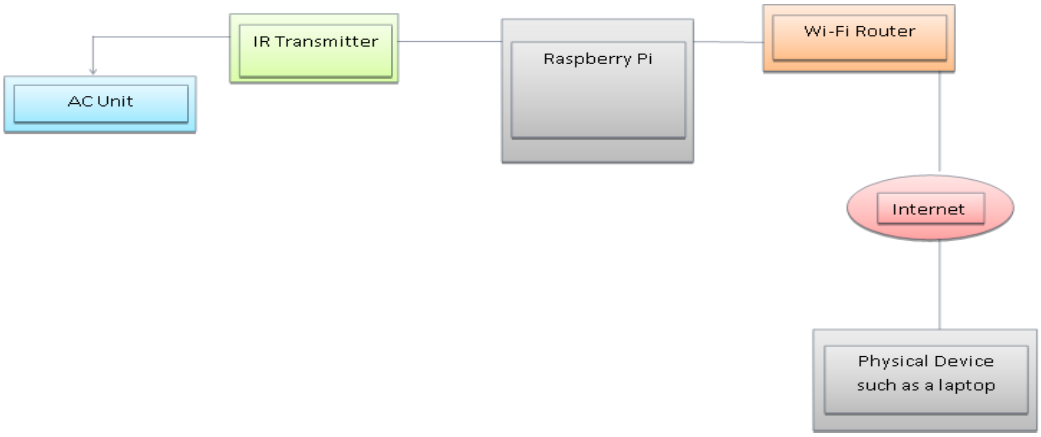


Figure 1: Smart Air Conditioner Using Internet of Things System Specifications

II. Literature Review

Home automation systems have been a successful revolution in the technology world. Extensive researches were conducted on home automation systems. Indeed, home automation systems still receive an inordinate attention from academic organizations and institutions.

Different technologies have been employed to build home automation systems, Al-Ali & Al- Rousan (2004) built a java-based home automation system where all home automated appliances were connected physically to an embedded board with integration to a Personal Computer acts as a web server which provided a remote access to the built system.

Java-based home automation system provides a secure solution due to the built-in security feature handed over by java technology. However, the system is not cost effective due to the need of high quality specifications of the Personal Computer as well as expensive installed wired environment. Furthermore, since the system depends mainly on wired communication, installation's intrusiveness is increased.

Sriskanthan et al. (2002) developed a Bluetooth based home automation system where each home physical appliance is connected to a local Bluetooth sub controller through wired connections. Each appliance communicates with the local Bluetooth sub controller connected to and then all local Bluetooth sub controllers communicate with a primary controller through wireless connection.

In contrast to Java-based home automation system, Bluetooth based home automation system architecture reduces intrusiveness of wired installation due to the use of wireless technology. Nevertheless, some intrusiveness is still there due to the wired communication between home appliances and Bluetooth sub controllers. Moreover, this system connects one Bluetooth sub controller to many home physical appliances due to the cost of Bluetooth technology where it is appropriate to have a dedicated Bluetooth sub controller for each appliance. Never forget to mention that the use of one Bluetooth sub controller shared between many home appliances actually causes access delay.

Zhu et al. (2010) proposed a Voice Control System for Zig Bee-based Home Automation system. Automatic speech recognition module is used to translate voice commands and send them to the actuator of the designed system via ZigBee network. Each automated home appliance is connected to a dedicated ZigBee module in order to operate and control it remotely.

Similar to Java-based home automation system, developing a Voice Control System for ZigBee based Home Automation system is not cost effective due to the expense of ZigBee module; hence the system is becoming uneconomical as the number of physical home appliances to be automated is enlarged. Furthermore, the speech recognition system must have a module which unfortunately causes errors due to the noise.

When reviewing academic researches on home automation systems, it is apparently that, the developed systems with different existing technologies such as Bluetooth, Java and ZigBee are facing some

imperfections and limitations such as the cost, access delay, wired installation intrusiveness and power consumption.

In this project, I selected different and innovative but existing technology which is Internet of Things (IoT). The reason behind using this technology in this project can be justified from my following findings on IoT technology.

Internet of Things is considered to be the revolutionary technology in the future of the internet (Gubbi et al., 2013).

Ma (2011) declared that the original idea of Internet of Things was proposed at the end of 1990's through MIT Auto-ID Labs. IoT is much more related to the wireless sensors networks, mobile communications networks and Internet. IoT can be defined as a network that connects every existing physical object in the world to a unique address in order to provide quick and smart Services (Ma, 2011).

IoT is a complicated technology and it consists of four layers; application service layer, information integration layer, data exchange layer and object sensing layer. Application service layer offers satisfied services to different users. Information integration layer integrates unclear information into usable knowledge, recombines and cleans unclear information attained from networks. Data transmission transparency is handled by data exchange layer. Sensing objects and obtaining data are handled by the forth layer which is object sensing layer. Never forget to mention that sensing feature is not supported by traditional Internet and accordingly it only interconnects intelligent physical objects. On the other hand, IoT interconnects both intelligent and non-intelligent physical objects due to the availability of object sensing layer (Ma, 2011). Table 1 summarizes limitations of different existing home automation technologies and IoT addressed solutions.

Table 1: IoT Addressed Solutions over Existing Technologies

Existing Technology	Short	IoT Addressed Solutions
Java-based home automation system	- High quality specifications of the Personal Computer acts as a Web Server - Expensive installed wired environment - High intrusiveness - High power consumption - Not cost effective	- Cloud storage - Wireless connection - Intrusiveness free - Energy effective - Cost effective
Bluetooth based home automation system	- Low intrusiveness - Access delay	- Wireless connection - Intrusiveness free
	- High power consumption - Not cost effective	- Access delay free - Energy effective - Cost effective
ZigBee-based Home Automation system	- ZigBee module is expensive - Noise caused by Speech recognition module - High power consumption - Not cost effective	- Noise free - Energy effective - Cost effective

Existing recent studies and conducted researches on controlling home appliances remotely mainly focuses on the use of IoT devices such as Raspberry Pi developed by Raspberry Pi Foundation. Raspberry Pi is cost effective, powerful and small size device that is considered to be a small size single board computer. Raspberry Pi may operate using different platforms; RISC OS, Arch Linux, Pidora, Raspbian and Microsoft Windows 10 IoT core (Harrington, 2015).

In this thesis, I will discuss two different platforms which are Raspbian, the most popular Operating System used for Raspberry Pi and Microsoft Windows 10 IoT core, the new raised Operating System developed by Microsoft.

Raspbian is an open source Linux based Operating System. It is a modified platform from Debian Operating System. Raspbian Operating System was developed exclusively for Raspberry Pi and hence it is

called Raspbian. Raspbian inherits almost all Debian features including above 35,000 free software packages. Beginners with Raspberry Pi are strongly recommended to start with Raspbian since it is designed for an easy use with different software packages (Harrington, 2015).

Windows 10 IoT core is an innovative version of Windows 10 and is targeting the small and embedded devices with or without display screens Raspberry Pi 2, Raspberry Pi3, Minnow Board MAX and Dragon Board 410c (Teixeira , 2015; Microsoft, 2016). Windows 10 IoT core intended to have a low barrier to access; hence making it easy to build professional devices. Windows 10 IoT core is compatible with different open source languages and works efficiently with Visual Studio platform as well (Teixeira, 2015).

Window 10 IoT core brings all powerful feature of Windows into your devices such as online storage, automatic Windows update through internet, user interface, security, Universal Windows Platform (UWP) APIs; the rich platform to easy control designed applications and cloud-based services (Microsoft, 2016; Anders, 2016).

Celebre et al. (2015) used Siri enabled mobile devices for remotely control home appliances, which are air conditioner unit, television, window blinds and lights using raspberry pi with Raspbian Operating System. In this system, the home appliances are connected to the raspberry pi through a relay and a motor driver. Both raspberry pi and Siri enabled mobile device are connected to the same local network (Celebre et al., 2015).

Rieger (2016) used raspberry pi, IR Diode, IR receiver and stepper motor to build a web interface to remotely control blind opener and air conditioner. The user accessed the web interface which transmits issued commands to a controller script. This system used raspberry pi with Raspbian Operating System.

Ivancreations.com (2016) built a mobile application and used Google voice recognition to remotely control Daikin air conditioner unit using raspberry pi and LED. In this system, the air conditioner unit is connected to the raspberry pi through Infra-Red transmitter LED. Both raspberry pi and the application based mobile are connected to the same local network through home Wi-Fi router (Ivancreations.com, 2016). The system is implemented using raspberry pi with Raspbian Operating System.

Vasanwala (2015) developed Home Automation using Raspberry Pi2 and Windows 10 IoT system. Lights, fans and wall sockets are connected to an Arduino – Internet of Things microcontroller device -. Each room must have its own Arduino connected to home appliances in that room, one Passive Infra-Red module, one temperature sensor that senses human presence and one LDR which detects light intensity. All Arduino microcontrollers are then connected to the Raspberry Pi through I2C Bus. Basically, Arduino controls all home appliances and reads data from sensors and periodically sends those collected data to Raspberry Pi. Raspberry Pi sends data collected from Arduino microcontrollers to a wire frame application. You may control connected home appliances using wire frame application as well.

Low-cost Home Automation with Voice Control system is built by Gillett (2015). The system used Node.js server to control different existing hardware in a room such as: lights, door and LED Strip using Raspberry Pi. A web application is built based on voice control in order to control room's hardware remotely. When the user clicks on microphone button in the application, Raspberry Pi starts recording voice audio. The recorded audio is then sent to a natural language API called Wit.ai in order to analyze it and extracts the meanings. The extracted meanings are then sent back to Raspberry Pi in order to perform the action. The system is built using Raspberry Pi with Microsoft Windows 10 IoT core Operating System Platform.

Ganesan (2015) built WARAN – Home Automation system. WARAN is a modular system stands for Windows IoT, Azure, Raspberry Pi, Arduino, NRF24L01+ wireless solution. WARAN consists of one Hub acts as a control server and many modules such as: temperature sensor module and humidity sensor module which are connected to an Arduino. The basic functionality of the system is that the added modules read data and sends them to the Arduino through NRF24L01+. Arduino is then sends collected data from the sensors to the control server in Raspberry Pi through I2C Bus. Collected data from sensors such as: warnings and alerts in any module is also posted in a Windows phone application.

Through reviewing the above existing recent studies and conducted researches on controlling home appliances remotely, it is obvious that air conditioner appliance is successfully controlled remotely via Raspbian Platform using Linux Infrared Remote Control (LIRC). LIRC is an open source library that allows a user to record, decode and send Infra-Red signals of many standard remote controls (Bartelmus, 2016). On the other hand, there isn't any published system that controlled air conditioner remotely using Microsoft Windows 10 IoT core Platform. No one till now could implement any home appliance operates using Infra-Red signals using Windows 10 IoT core because it does not have any Infra-Red library. There were researches which attempted to automate home air conditioners using Win LIRC but they all failed. Win LIRC is Windows equivalent of LIRC which enables users receive and transmit Infra-Red signal of standard remote controls (Baily et al., n.d.).

III. System Deisghn

Through reviewing different recent literatures demonstrated in CHAPTER 2, apparently there is a serious limitation with Windows 10 IoT core platform in reference to the lack of Infra-Red library. As a result, the most optimal platform to implement the smart air conditioner using Internet of Things is going to be Raspbian platform since it supports LIRC library. Before implementing the project, an overall design is built.

a) Web Application Wire Frame

The Wire Frame Design for the system is shown in Figure2. When the user who has a right access logs in the website, he will be able to see Control AC tab. The Control AC tab includes controlling AC power and the temperature of the air conditioner.

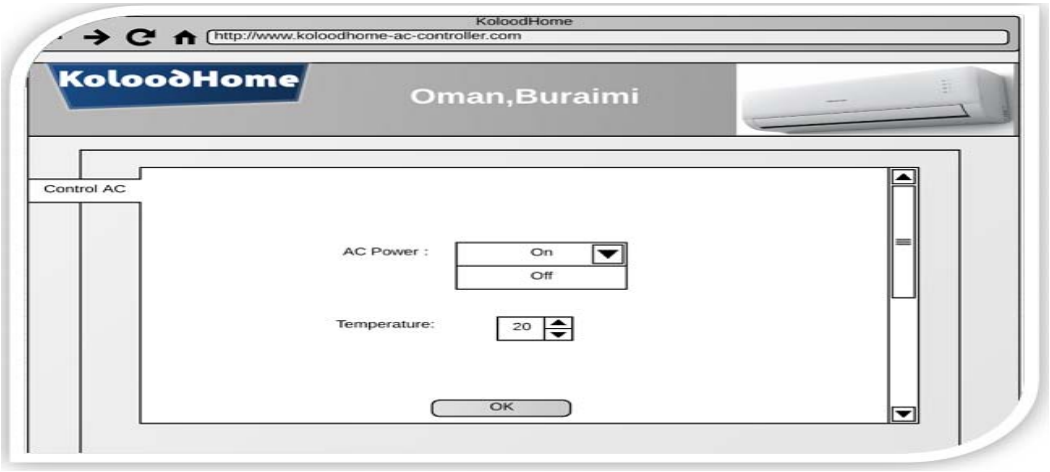


Figure 2: Web Application Wire Frame for Smart Air Conditioner Controller

b) Architecture Diagram

The system architecture includes the infrastructure and network design of the system. The web application will be designed using PHP web language and Raspbian will be the business logic for the design. The system architecture of the system is shown in Figure 3. My SQL server is used as web server to store data for web console and database server will be hosted in LAMP in Raspberry Pi 3. VPN will be used between the LAMP and the Internet for security issue. PHP will be used as the front End and SQL database as the backend of the web application. The user can access the web application by writing the address in the web browser. On the other hand, the air conditioner is connected to the Wi-Fi at the same time.

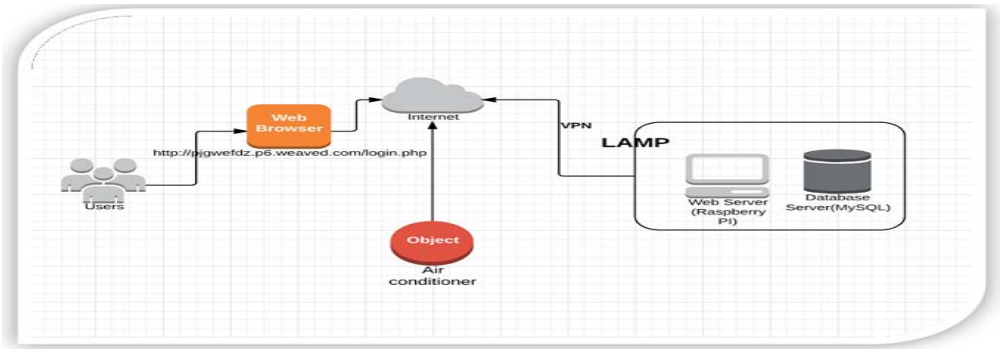


Figure 3: Architecture Diagram for Smart Air Conditioner Controller

c) Circuit Design

As mentioned earlier, air conditioner does not have an internet connection port. Hence, it will be connected to the raspberry pi 3 that has the required internet connection port. The connection between the air conditioner and the raspberry pi 3 is through the Infra-Red (IR) transmitter which is a Lite Emitting Diode (LED) emitting Infra-Red lights (connected to the raspberry pi 3) and IR receiver. Always remember that the circuit needs resistors in order to control current flow and maintain the raspberry pi from damage. Figure 4 illustrates circuit design used in the project. IR LED is responsible of emitting infra-red signals and IR receiver is responsible of receiving infrared signals and modulating them. Basically this circuit design is going to be used to read SANYO air conditioner remote control and interpret its codes.

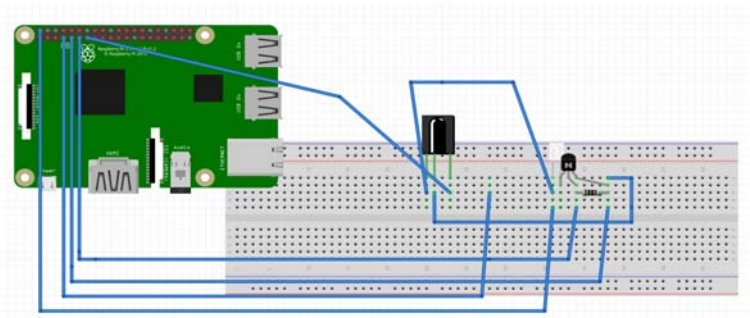


Figure 4: Smart Air Conditioner Circuit Design

IV. Implementation

a) Hardware

In order to implement the smart air conditioner, the following hardware is required:

- Raspberry Pi 3 Model B.

Figure 5: Raspberry Pi 3 Model B

- PIR (Passive Infra-Red) Motion Sensor

Figure 6: Passive Infra-Red Sensor

- SD Card (Minimum 8 GB).

Figure 7: SD Card

- SD Card Reader.
- IR Receiver.

Figure 8: IR Receiver

- IR Transmitter

Figure 9: IR Transmitter

- 10K ohm Resistor

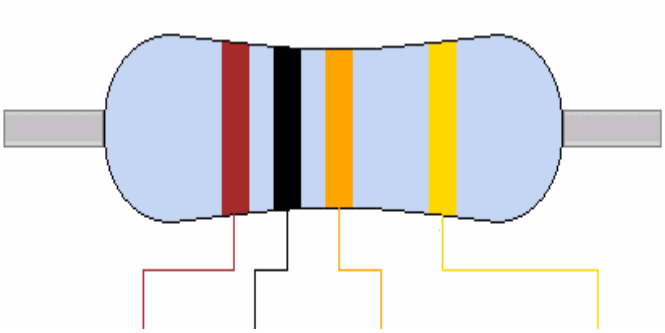


Figure 10: 10k ohm Resistor

- PN2222 Transistor

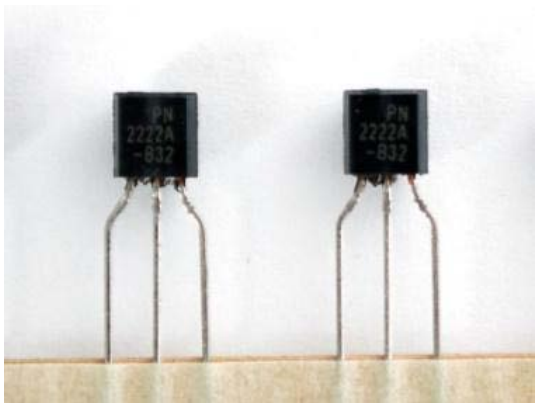


Figure 11: PN2222

- Wi-Fi Hot Spot.
- Monitor.
- USB Keyboard.
- USB Mouse.
- HDMI to VGA Cable.

Figure 12: HDMI to VGA Cable

- Solder less Bread Board

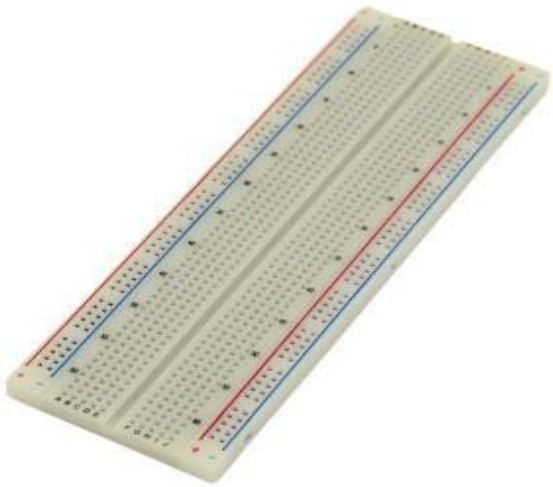


Figure 13: Solder less Bread Board

- Jumper Cables

Figure 14: Jumper Cables

b) Software

The following software packages are needed to implement the smart air conditioner:

- Raspbian Jessie Operating System (The full desktop image based on Debian Jessie).
- Wiring Pi.
- Linux Infrared Remote Control (LIRC).
- LAMP (Linux, Apache, MySQL, PHP) Web Development Platform.
- Python (The programming language that is pre-installed in Raspbian Jessie Operating System).
- Win32 Disk Imager.

c) Raspbian Jessie Operating System Setup

1. Download Raspbian Jessie Operating System image from the official site of Raspberry Pi: https://www.raspberrypi.org/downloads/raspbian/
2. Place 8 GB SD card into your SD card reader. In this project, I used the built in SD card reader in my laptop.

Figure: screenshot of Raspbian Jessie OS image mager

3. Mount the downloaded Raspbian Jessie Operating System image into your SD card using Win32 Disk Imager (if you are using Windows Operating System).

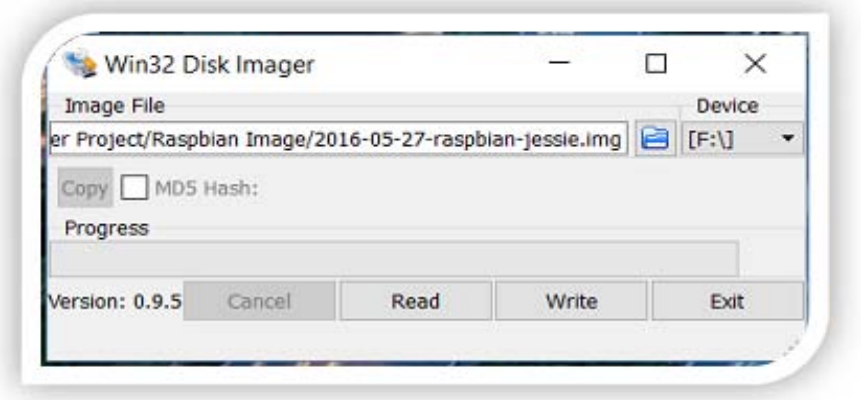


Figure 15: Win32 Disk

4. Eject your SD card from the SD card reader and mount your SD card into your Raspberry Pi 3.

Figure 16: SD CARD on Raspberry Pi 3 Model

5. Connect your Raspberry Pi 3 to your monitor using HDMI to VGA cable.
6. Connect your USB keyboard and mouse.
7. Connect the power cable of the Raspberry Pi 3.
8. Power on your Raspberry Pi 3.

Figure 17: Screenshot of startup Raspbian Operating System

9. Test the raspberry Pi 3 by opening terminal window and type any command for example: hostname

Figure 18: Screenshot of testing terminal window in Raspbian

d) Wiring Pi Setup

In order to control GPIO pins in your Raspberry pi 3 such as: read the pins, write the pins and control the pins from shell scripts, you have to setup Wiring Pi in your Raspberry Pi 3.

➢ Definition

Wiring Pi: is a GPIO access library used in Raspberry Pi and is written in C language (Wiring Pi, 2016).

The followings steps describe installation, setup and test of Wiring Pi in your Raspberry Pi 3:,k

1) Connect your Raspberry Pi 3 to your wireless hot-spot using built in Wi-Fi dongle in your Raspberry Pi.
2) Open command terminal in your raspberry Pi 3.
3) Type the following command to setup GIT. Actually, GIT is maintained for Wiring Pi so that the user can easily track changes:

```
sudo apt-get install git-core
```

4) Type the following command to obtain Wiring Pi through GIT:

```
sudogitclonegit://git.drogon.net/wiringPi
```

5.

5) Type the following command to access WiringPi directory:

```
sudo cd wiringPi
```

6) Type the following command to install and build WiringPi library:

```
./build
```

7) Test Wiring Pi through typing the following command in terminal window:

```
gpio –vgpio readall
```

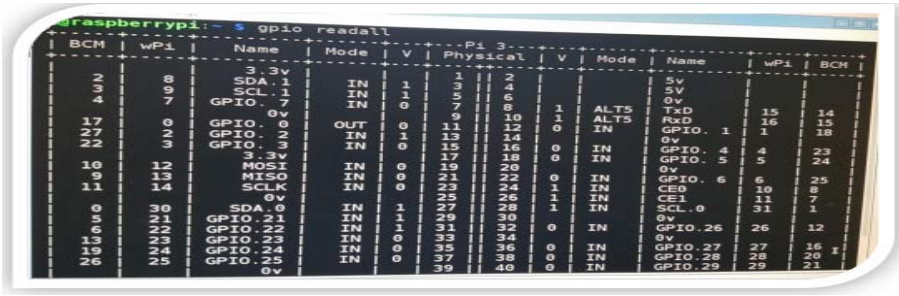


Figure 19: Screenshot of Testing Wiring Pi in terminal

e) LIRC (Linux Infrared Remote Control) Installation and Configuration

In order to be able to record your air conditioner remote control Infra-Red codes, you have to install and configure LIRC in your Raspberry Pi 3.

➢ Definition

LIRC: is an open source library that allows a user to record, decode and send Infra-Red signals of many – not all- remote controls (Bartelmus, 2016).

The followings steps describe installation and configuration LIRC in your Raspberry Pi 3:

1. Open command terminal in your raspberry Pi 3.
2. Type the following command to install LIRC library:

```
sudo apt-get install lirc
```

f) LAMP (Linux, Apache, MySQL, PHP) Installation and Configuration

In order to be able to record your air conditioner remote control Infra-Red codes, you have to install and configure LIRC in your Raspberry Pi 3.

➢ Definition

LAMP: is a web development platform used in Linux Operating System. It is the equivalent of WAMP (Windows, Apache, MySQL, and PHP) web development platform used in Windows Operating System.

The followings steps illustrate installation and configuration LAMP in your Raspberry Pi 3:

First: Apache Sever Setup

➢ Definition

Apache: is one of the most popular web server applications. It is installed in Raspberry Pi 3 to serve developed web pages by the user (raspberrypi.org, 2016).

Follow the following steps in order to install and configure Apache server:

1. Open command terminal in your raspberry Pi 3.
2. Type the following command to install Apache2 package:

```
sudo apt-get install apache2 –y3.
```

3. Open a web browser in your Raspberry Pi 3 and type: http://localhost/ in order to test that Apache 2 server has been installed successfully.
4. Note that the default webpage index.html is stored in /var/www/html directory in your Raspberry Pi 3.

Second: PHP Setup:

➢ Definition

PHP: is one of the most popular web languages. It is a preprocessor that runs any received requests from a web page, process the requested page and sends it back to the web browser (raspberrypi.org, 2016).

Follow the following steps in order to install and configure PHP:

1. Open command terminal in your raspberry Pi 3.
2. Type the following command to install PHP 5 and Apache packages:

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

3. Test PHP using steps listed below:

1. Create test.php file using the following command in terminal:

```
sudo leafpad test.php
```

2. Type the following PHP code inside test.php and save the file:

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

3. Open a web browser in your Raspberry Pi 3 and type: http://localhost/test.php

Third: MySQL Setup

➢ Definition

MySQL: is one of the most popular database engines (raspberrypi.org, 2016).

Follow the following steps in order to install and configure MySQL:

1. Open command terminal in your raspberry Pi 3.
2. Type the following command to install MySQL server and PHP- MYSQL packages:

```
sudo apt-get install mysql-server
php5-mysql -y
```

3. While installing MySQL server, it will ask you to enter a root password in order to protect your server. Insert the password and remember it very well or you will lose your root access into MySQL server in your Raspberry Pi 3.
4. Type the following command to restart Apache server:

```
sudo service apache2 restart
```

g) Implementation and Testing

i. *LIRC Setup in Raspberry Pi 3:*

Prior implementing the project, you must setup LIRC in your Raspberry Pi 3. The following steps describe implementation of IR Receiver circuit:

1. Open command terminal in your raspberry Pi 3.
2. Type the following command to check GPIO output and input pins in Raspberry Pi 3 and select the appropriate for you:

```
gpio readall
```

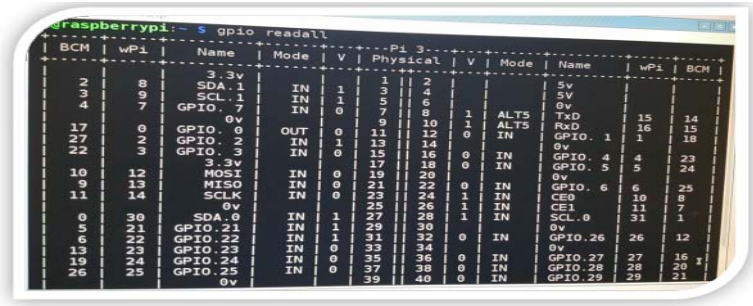


Figure 20: Screenshot of gpio readable command on Rasbpian Operating

3. Type the following command to edit *modules* file:

```
sudo leafpad /etc/modules
```

4. Add the following two lines into *modules* file:

```
lirc_dev lirc_rpi
gpio_in_pin=17
gpio_out_pin=27
```

❖ *Note:* The added two lines specify that the GPIO input pin in Raspberry Pi 3 is 17 for IR Transmitter and GPIO output pin is 27 for IR Receiver.

5. Type the following command to edit *hardware*. conf file:

```
sudoleafpad/etc/lirc/hardware.conf
```

6. Change *hardware. conf* file exactly as the following file

```
############################################################
# /etc/lirc/hardware.conf
#
# Arguments which will be used when launching lircd
LIRCD_ARGS="--uinput"
# Don't start lircmd even if there seems to be a good config
file
# START_LIRCMD=false
# Don't start irexec, even if a good config file seems to exist.
# START_IREXEC=false
# Try to load appropriate kernel modules
LOAD_MODULES=true
# Run "lircd --driver=help" for a list of supported drivers.
DRIVER="default"
# usually /dev/lirc0 is the correct setting for systems using
udev
DEVICE="/dev/lirc0"
MODULES="lirc_rpi"
# Default configuration files for your hardware if any
LIRCD_CONF=""
LIRCMD_CONF=""
############################################################
```

7. Type the following commands to stop and start *lircd* service so that the above made changes take effect successfully:

```
sudo /etc/init.d/lirc stop
sudo /etc/init.d/lirc start
```

8. Type the following command to edit *config.txt* file:

```
sudo leafpad /boot/config.txt
```

9. Add the following line into *config.txt* file:

```
dtoverlay=lircrpi,gpio_in_pin=17,
          gpio_out_pin=7
```

10. Type the following command to reboot your Raspberry Pi 3 in order to save changes made above:

```
sudo reboot
```

h) IR Receiver

i. *Wiring up IR Receiver*

1. Place IR Receiver in your solder less bread board.
2. The data pin is connected to GPIO pin 27 as per the configuration we made earlier in LIRC.
3. The ground pin is connected to GPIO ground pin in your Raspberry Pi 3.
4. The +5v pin is connected to GPIO 5v pin (DC Power) in your Raspberry Pi 3 in order to power on your IR Receiver.

❖ *Note:* Data pin and +5v in your IR Receiver vary from type to type. You are recommended to review data sheet of the IR Receiver you decide to use. In this project, I used the data sheet of the IR Receiver I used and listed in LIST of FIGURES Section (Page 7).

ii. *Testing the IR Receiver Circuit*

1. Power on your Raspberry Pi 3.
2. Open the terminal in your Raspberry Pi 3 and type the following command to stop *lircd* service:

```
sudo /etc/init.d/lirc stop
```

3. Type the following command to start outputting raw data received from IR Receiver:

```
mode2 -d /dev/lirc0
```

4. Point your air conditioner remote control to the IR receiver you wired up earlier and start pressing the buttons. If your output in the terminal looks as the following:

```
space 16300
pulse 95
space 28794
pulse 80
space 19395
pulse 83
space 402351
pulse 135
space 7085
pulse 85
space 2903
```

It means your IR Receiver circuit is implemented and configured properly.

i) IR Transmitter

i. *Wiring up IR Transmitter*

1. Place IR Transmitter in your solder less bread board.
2. Place NP2222 transistor in series with the IR Transmitter (short pin of the IR Transmitter).

❖ *Note:* The main function of the NP2222 transistor is amplifying or/and switching electronic signals and electrical power. In other words, any applied current or voltage to one pair of the transistor may be changed before reaching the other pair of the transistor as needed.

3. Place 10k ohm resistor in series with the NP2222 transistor, one pin in series with the base and one pin in series with the collector.

❖ *Note:* The main function of the 10k ohm resistor is to control the current flows across the IR Transmitter and keep it safe from burning.

4. The long pin of the IR Transmitter is connected to GPIO 5v pin (DC Power) in your Raspberry Pi 3 in order to power on your IR Transmitter.
5. The first pin of 10k ohm resistor that is connected in series with NP2222 transistor is connected to GPIO pin 17 as per the configuration we made earlier in LIRC.
6. The second pin of 10k ohm resistor is connected to GPIO ground pin in your raspberry Pi 3.

ii. *Testing the IR Transmitter Circuit*

1. Power on your Raspberry Pi 3.
2. Open the terminal in your Raspberry Pi 3 and type the following command to stop *lircd* service:

```
sudo /etc/init.d/lirc stop
```

Figure 21: Screenshot of lircd command on terminal window

Create a new configuration file for your air conditioner remote control called *ir1.conf* using the following command:

```
irrecord -d /dev/lirc0 –f
ir1.conf
```

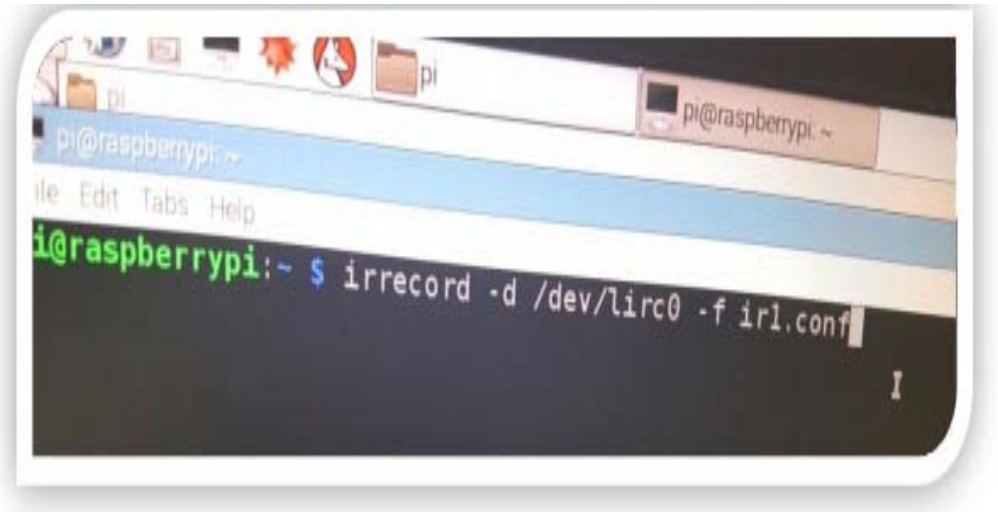


Figure 22: Screenshot of creating new configuration file on Raspbian

Then you are supposed to get message as shown in figures below to start recording your air conditioner remote control codes:

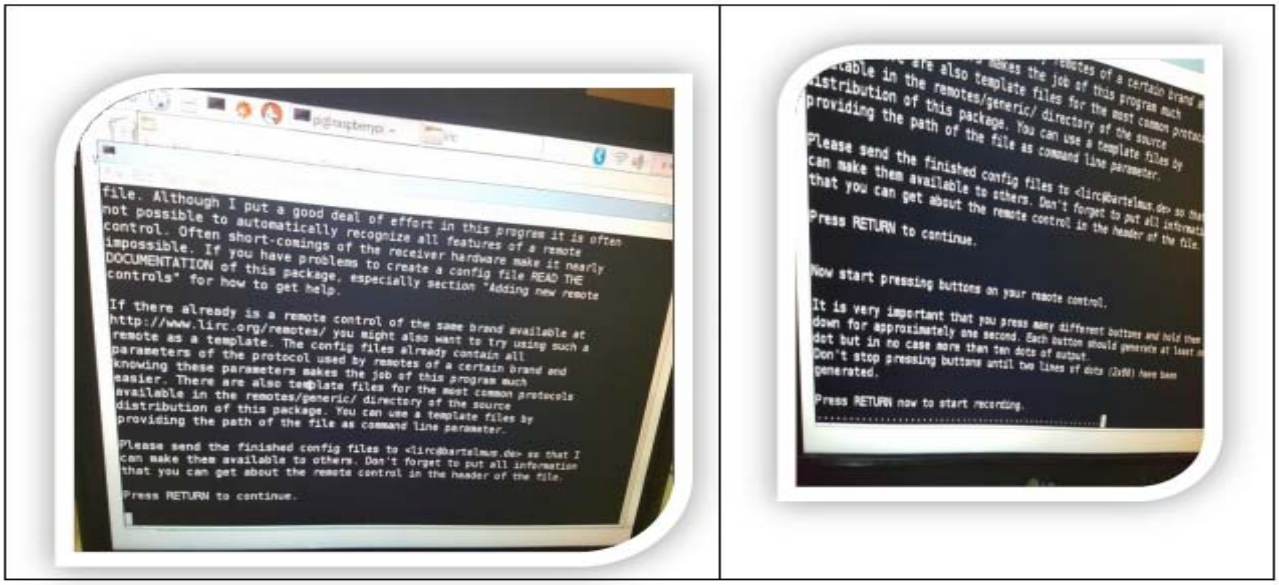


Figure 23: Screenshots of irrecod instructions

You have to follow the instruction and record codes to turn on, turn off, increase the volume of the temperature and decrease the volume of the temperature for the air conditioner.

Then you will be prompt to enter a name for each button you press. Kindly, note that you cannot enter any random names for your recorded keys. LIRC has its own buttons' names. In order to check these valid names, type the following command:

```
irrecord --list-namespace
```

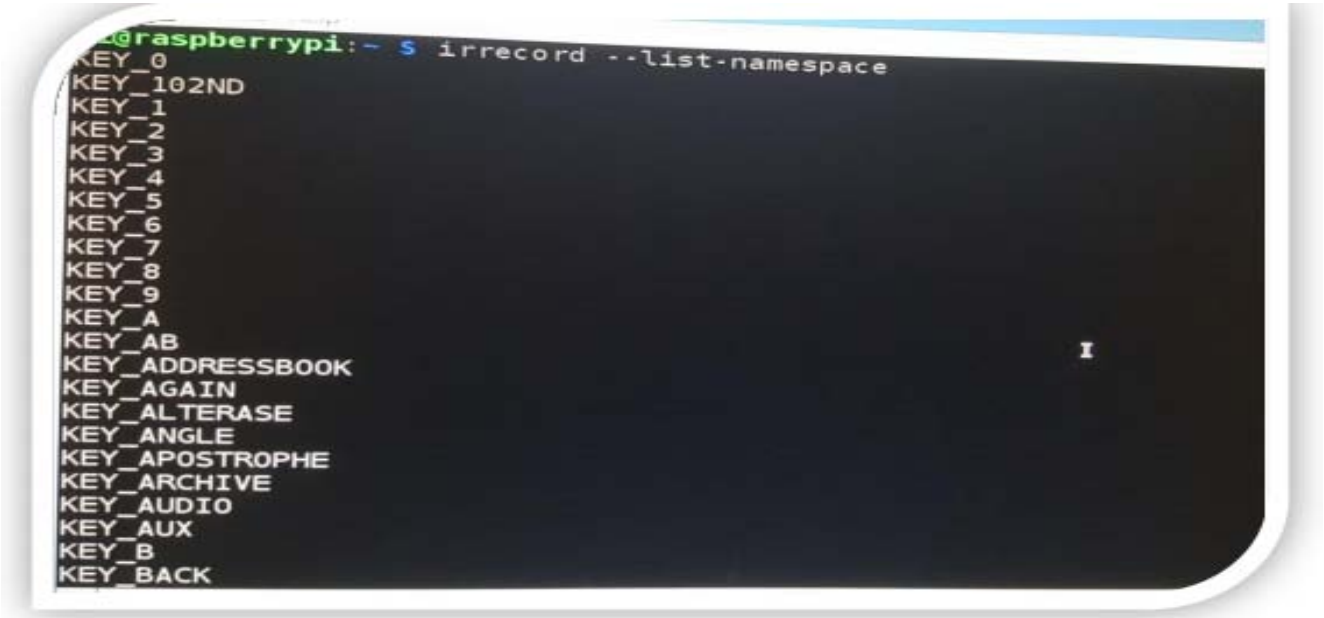


Figure 24: Screenshots of valid names assigned to recorded buttons in LIRC

In this test I picked up the following keys

- KEY_POWER to turn on the air conditioner.
- KEY_POWER2 to turn off the air conditioner.
- KEY_UP to increase the volume of the temperature.
- KEY_DOWN to decrease the volume of the temperature.

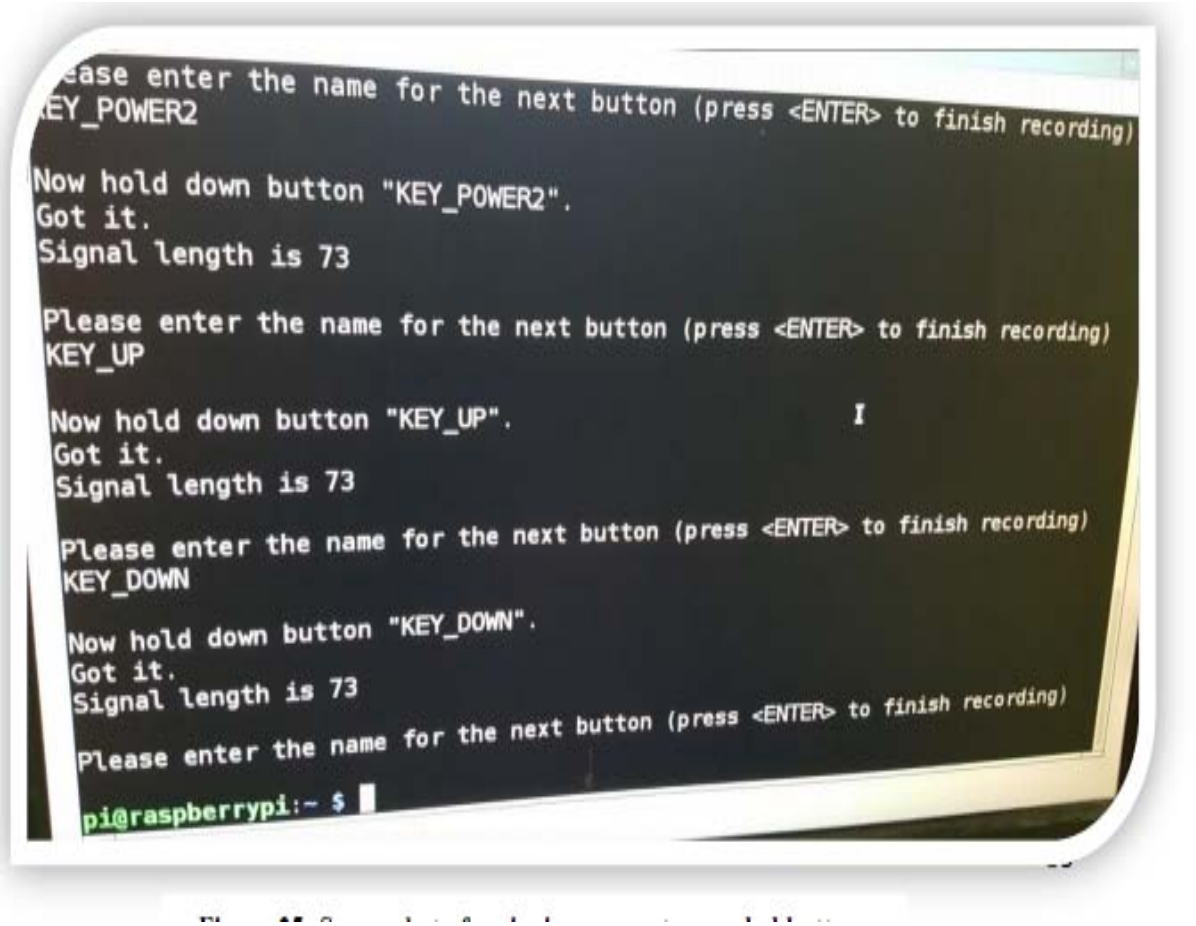


Figure 25: Screenshot of assigning names to recoded buttons

3. Copy the new created *ir1.conf* file into the original *lircd.conf* file created by LIRC using the command:

```
sudo cp ir1.conf
/etc/lirc/lircd.conf
```

4. Type the following command to *start lircd* service:

```
sudo /etc/init.d/lirc start
```

5. In order to send the recorded signal for turning on the air conditioner that is saved in *lircd.conf*, type the following command:

```
irsend SEND_ONCE ir1.conf
KEY_POWER
```

Figure 26: Screenshot of switching sir conditioner in command window

6. In order to send the recorded signal for turning off the air conditioner that is saved in *lircd.conf*, type the following command:

```
irsend SEND_ONCE ir1.conf
KEY_POWER2
```

7. In order to send the recorded signal for increasing the volume of the temperature, type the following command:

```
irsend SEND_ONCE ir1.conf KEY_UP
```

8. In order to send the recorded signal for decreasing the volume of the temperature , type the following command:

```
irsend SEND_ONCE ir1.conf
KEY_DOWN
```

If your air conditioner turns on and off, increases and decreases the volume of the temperature successfully, it means your IR Transmitter circuit is implemented and configured properly.

j) PIR Sensor

i. *Wiring up PIR Sensor*

1. Place PIR Sensor in your solder less bread board.
2. The input pin is connected to GPIO pin 04 in your Raspberry Pi 3.
3. The output pin is connected to GPIO pin 27 in your Raspberry Pi 3.
4. The +5v pin is connected to GPIO 5v pin (DC Power) in your Raspberry Pi 3 in order to power on your PIR Sensor.

ii. *Programming PIR Sensor*

❖ *Note:* Python programming language will be used to program PIR Sensor.

1. Type the following command to create *pir.py* file:

```
sudo nano pir.py
```

2. Write the following Python code inside *pir.py* file:

See APPENDIX B: IMPLEMENTATION SOURCE CODE, Section I: PIR Sensor Source Code

iii. *Testing the PIR Sensor Circuit*

❖ Note: For testing purposes, I have changed time period to sense any motion before switching off the air conditioner into 10 seconds. In my real project, I set time period to 1800 seconds.

1. Power on your Raspberry Pi 3.
2. Type the following command to start *lircd* service:

```
sudo /etc/init.d/lirc start
```

3. Type the following command to start your PIR Sensor:

```
sudo python /home/pi/pir.py
```

Now you will see that PIR Sensor is working, if you move your hand in front of the PIR Sensor circuit, you will see number 1 displays in your terminal.

Number 1 means that PIR Sensor detects a motion. Now remove your hand and stay stable for 10 seconds, you will notice number 0 displays in the terminal. Number 0 means that PIR Sensor does not detect any motion.

Wait for 10 seconds and if your air conditioner turned off then, your PIR Sensor circuit is implemented and configured successfully.

k) Web Application Development

i. *Developing Web Application*

❖ *Note 1:* PHP web language is used to create the web page.

MySQL database engine is used to create the database.

PHP web language is used as connection agent between the web page and the database.

❖ *Note 2:* All files related to the web application development will be found the following path:

```
/var/www/html
```

ii. *Creating the Web Page*

1. Open the terminal in your Raspberry Pi 3.
2. Type the following command to access `/var/www/html` directory:

```
cd /var/www/html
```

3. Type the following command to create *iotAC.php* file:

```
sudo touch iotAC.php
```

4. Type the following command to give full permission to iotAC.php file while editing:

```
sudo chmod 777 iotAC.php
```

5. Go to /var/www/html directory and double click on iotAC.php you have created in step 3.
6. Add the following PHP code and save the file:

See APPENDIX B: IMPLEMENTATION SOURCE CODE, Section II: iotAC.php Source Code

iii. *Building up the Database*

1. Open the terminal in your Raspberry Pi 3.
2. Type the following command to access MySQL server:

```
mysql –u root –p
```

It will prompt you to enter the password you have created when you setup MySQL server earlier in this Chapter. In my project, my password is: mysql

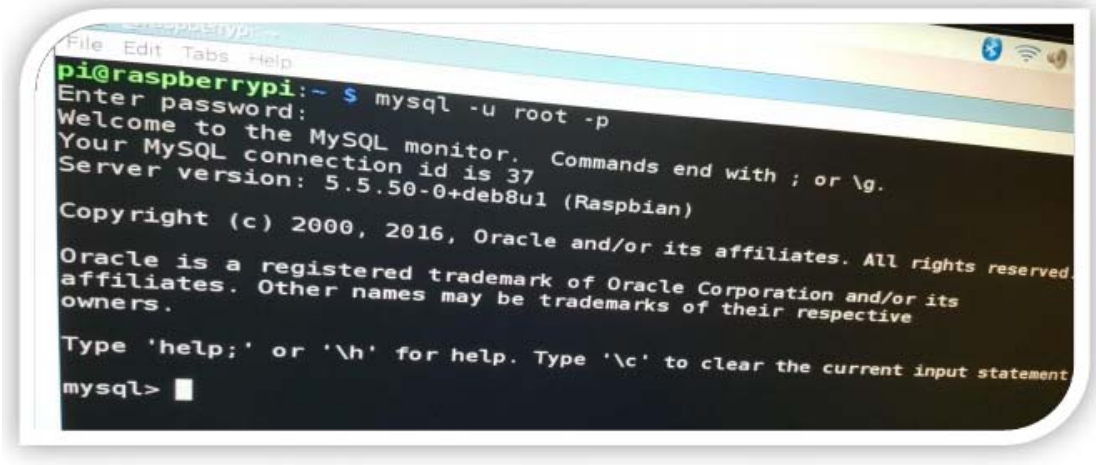


Figure 27: Screenshot of establishing MySQL server connection

3. Type the following command to show the databases you have in your server

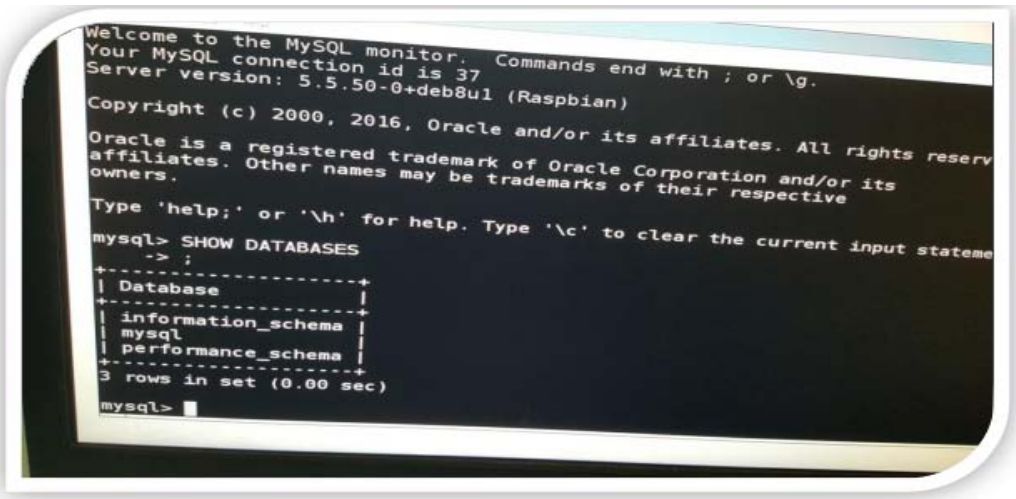


Figure 28: Screenshot of displaying databases available in MySQL

4. Type the following command to create a new database called *ac_control:*

```
CREATE DATABASE ac_control
```

7. Type the following command to show the databases you have in your server and make sure that *ac_control* database you have created in step 4 is among them:

```
SHOW DATABASES
```

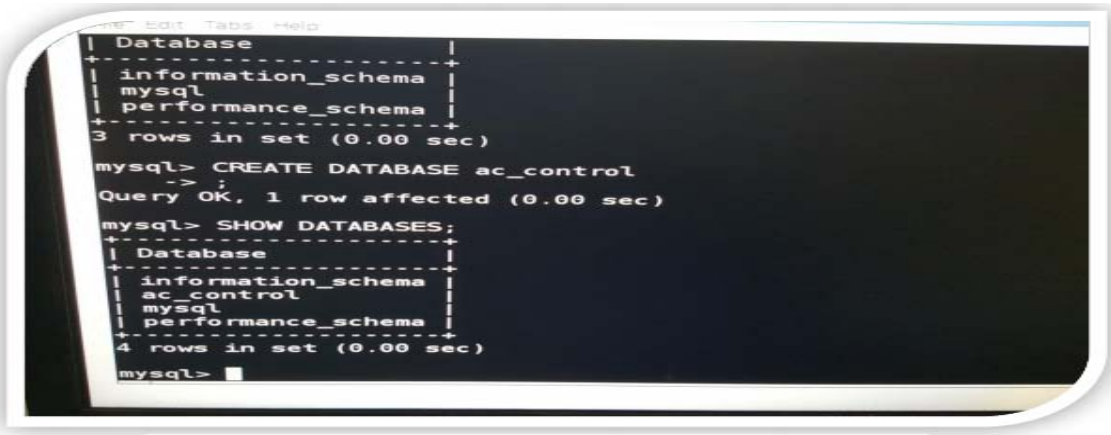


Figure 29: Screenshot of creating a new database in MySQL

8. Type the following command to access *ac_control* database:

```
USE ac_control;
```

9. Type the following command to create a table inside ac_control database called *login* which have username and password parameters:

```
CREATE TABLE `login` (`username
VARCHAR (255), `password`
VARCHAR(255))
```

10. Type the following command to add master username with master-ac@321 password into login table inside ac_control database:

```
INSERT INTO login values
(`master`,`master-ac@321`)
```

11. Type the following command to check if the username and password you have added in step 8 exist in login table:

```
SELECT * FROM login
```

12. Type the following command to quit MySQL server:

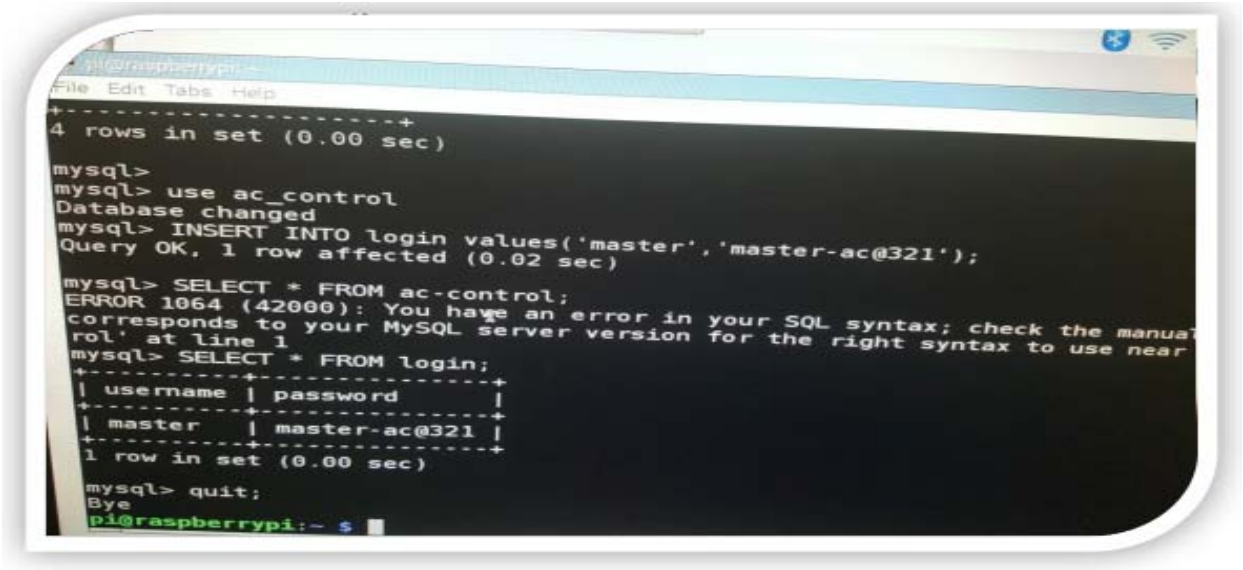


Figure 30: Screenshot of different quires to ac_control database in MySQL

l) Login Web Page

❖ *Note:* In order to make my web application secure, I have created a login web page.

1. Open the terminal in your Raspberry Pi 3.
2. Type the following command to access /var/www/html directory:

```
cd /var/www/html
```

3. Type the following command to create *login.php file:*

```
sudo touch login.php
```

4. Type the following command to give full permission to login.php file while editing:

```
sudo chmod 777 login.php
```

5. Go to /var/www/html directory and double click on login.php you have created in step 3.
6. Add the following PHP code and save the file:

See APPENDIX B: IMPLEMENTATION SOURCE CODE, Section II: login.php Source Code

m) Connection between Login Web Page and ac_control Database

❖ *Note1:* In order to connect between login web page and ac_control database, I have created a PHP file called *submit.*

1. Open the terminal in your Raspberry Pi 3.
2. Type the following command to access `/var/www/html` directory:

```
cd /var/www/html
```

3. Type the following command to create *submit.php* file:

```
sudo touch header.php
```

4. Type the following command to give full permission *to submit.php* file while editing:

```
sudo chmod 777 header.php
```

5. Go to /var/www/html directory and double click on submit.php you have created in step 3.
6. Add the following PHP code and save the file:

See APPENDIX B: IMPLEMENTATION SOURCE CODE, Section IV: submit.php Source Code

n) Creating PHP file that contains Required Credentials to Connect to MySQL Server:

❖ *Note:* In order for the submit.php file to connect to MySQL server, header.php file must be created to contain all required credentials.

1. Open the terminal in your Raspberry Pi 3.
2. Type the following command to access `/var/www/html` directory:

```
sudo /etc/init.d/lirc start
```

3. Type the following command to create *header.php* file:

```
sudo touch header.php
```

4. Type the following command to give full permission to *header.php* file while editing:

```
sudo chmod 777 header.php
```

5. Go to /var/www/html directory and double click on header.php you have created in step 3.
6. Add the following PHP code and save the file:

See APPENDIX B: IMPLEMENTATION SOURCE CODE, Section V: header.php Source Code

o) Testing Web Application

First: Testing Web Application when the Raspberry Pi 3 and Web Browser Device belong to the same network:

1. Power on your Raspberry Pi 3.
2. Connect your Raspberry Pi 3 to Wi-Fi through wireless dongle.
3. Open the terminal in Raspberry Pi 3 and type the following command to start *lircd* service:

```
sudo /etc/init.d/lirc start
```

4. Type the following command to start your PIR Sensor:

```
sudo python /home/pi/pir.py
```

5. Open the browser in your Raspberry Pi 3 and type the following:

```
http://the IP address of your Raspberry Pi3/login.php
```

In order to check the IP address of the Raspberry Pi, type the following command:

```
ifconfig
```

In my project, the IP address of my Raspberry Pi 3 is: 192.168.43.181, so it will be something like:

```
http://192.168.43.181/login.php
```

6. Insert the username: master and the password: master-ac@321 and click *Submit* button. You should be directed to *iotAC.php* page. Now click ON to turn on the air conditioner, OFF to turn off the air conditioner, + to increase temperature volume and – to decrease temperature volume. If all buttons operate as it is supposed, then your application is developed successfully.

Figure 31: Login Page of air conditioner website

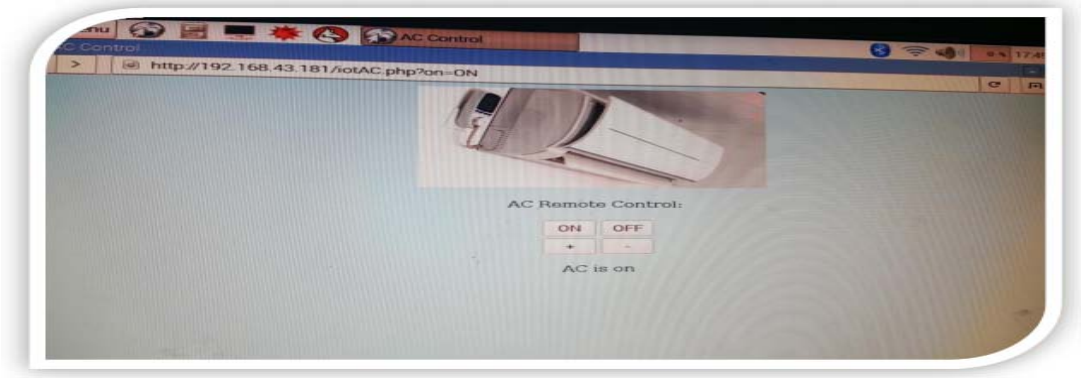


Figure 32: Air conditioner webpage

Note 1: If the user forgot to insert the username, the password or both of them, an error message is displayed.

7. Repeat steps 5 and 6 but with any device (desktop, laptop, PDA and smart phone) connected to the same local network as the Raspberry Pi 3.

❖ *Note 2:* It is not practical to run *lircd* service and *pir.py* service each time. As a result, I will configure them to start automatically when Raspberry Pi starts up.

1. Open terminal in Raspberry Pi 3.
2. Type the following command to open crontab editor:

```
sudo crontab –e
```

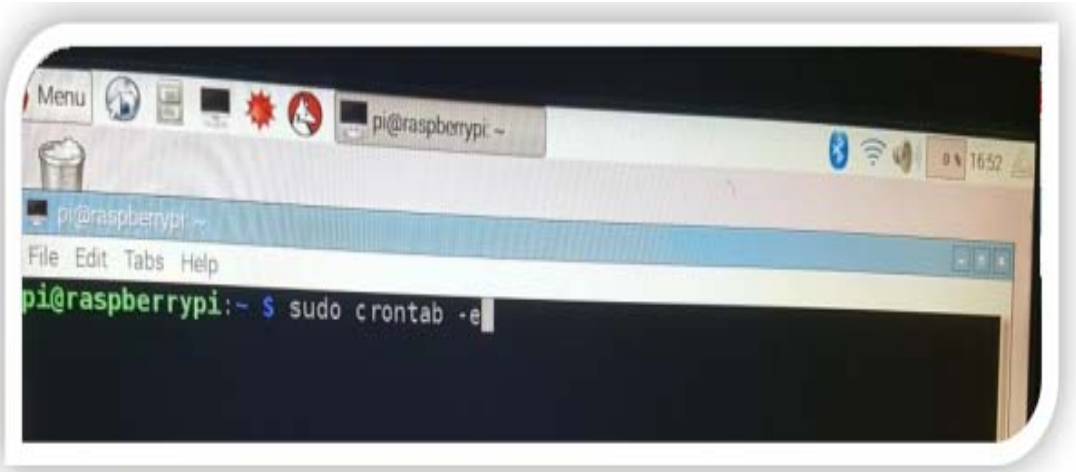


Figure 33: Screenshot of executing crontab editor

3. Add the following lines and exit the editor:

@reboot sudo /etc/init.d/lirc start &
@reboot sudo python /home/pi/pir.py &

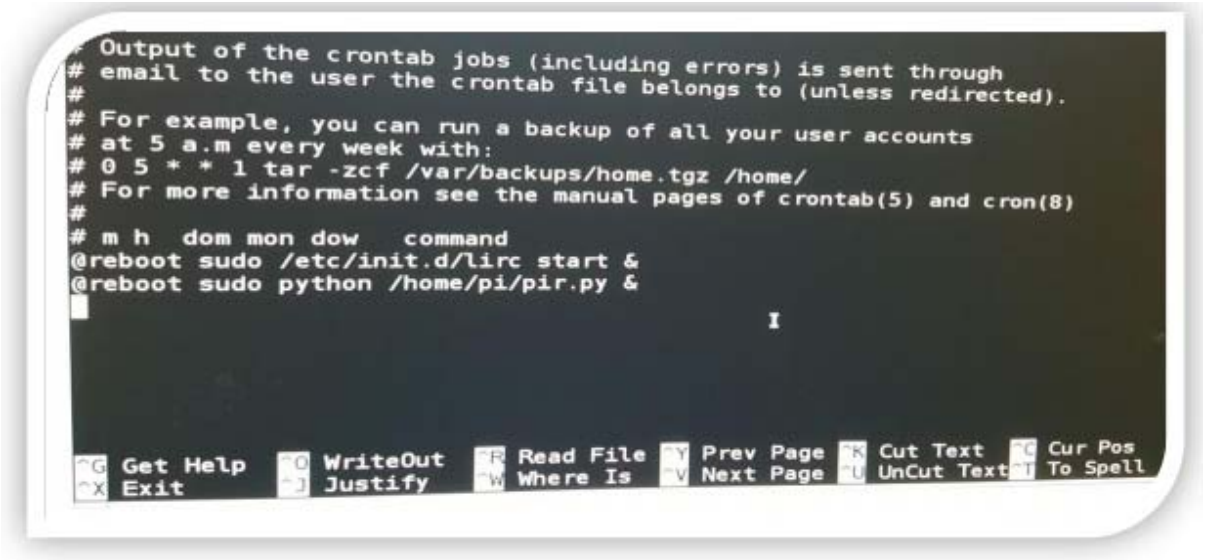


Figure 34: Screenshot of crontab editor

4. Press ctrl + x to exit the editor and press Y when it prompts you to save changes using CTRL + X

❖ *Note 3:* It is not practical for the Raspberry Pi IP address to keep changes automatically.

As a result, I will configure both Wireless and Ethernet interfaces to have a static IP address in my Raspberry Pi 3.

1. Open terminal in Raspberry Pi 3.
2. Type the following command to edit *dhcpd.conf* file:

```
sudo nano /etc/dhcpcd.conf
```

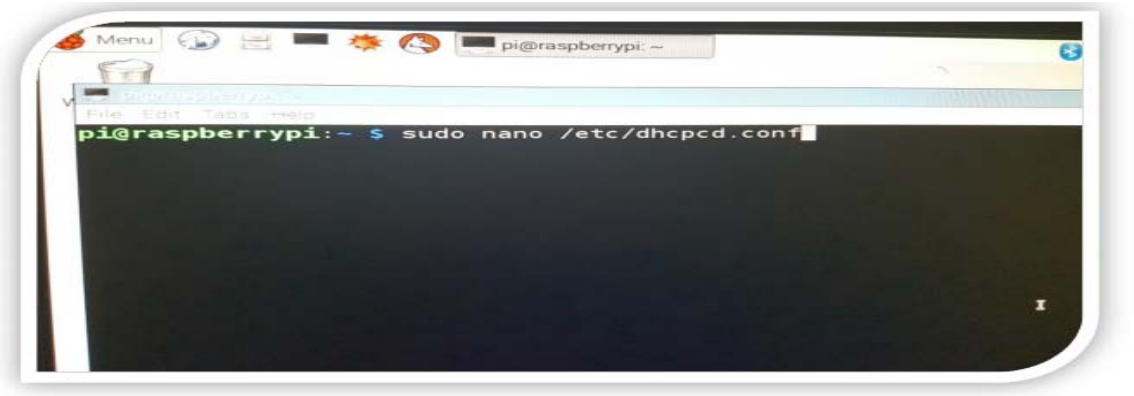


Figure 35: Screenshot of editing dhcpcd. conf file using nano text editor

3. Add the following parameters under interface eth0:

```
staticip_address=192.168.43.181/24
static routers=192.168.43.1
static domain_name_servers=192.168.43.1
```

4. Add the following parameters under *wlan0:*

```
static ip_address=192.168.43.181/24
static routers=192.168.43.1
static domain_name_servers=192.168.43.1
```

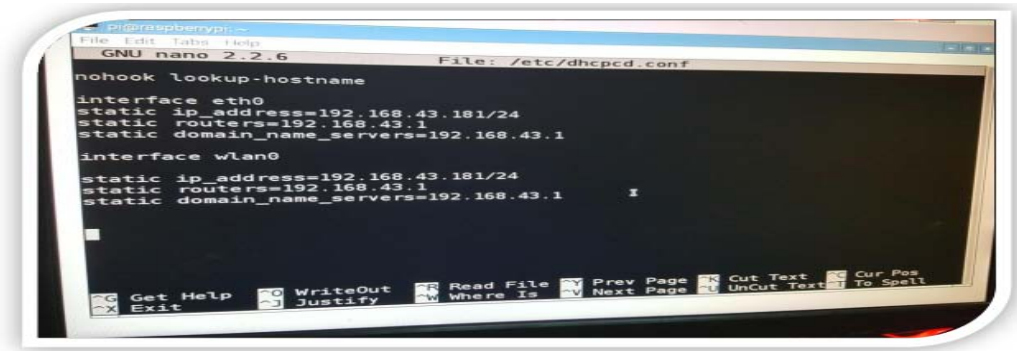


Figure 36: Screenshot of dhcpcd.conf

5. Press ctrl + x to exit the editor and press Y when it prompts you to save changes using CTRL + X
6. Reboot your Raspberry Pi 3 in order to take effect of made changes using the command:

```
sudo reboot
```

7. Check the IP address of the Raspberry Pi by type the following command:

```
Ifconfig
```

p) Accessing Developed Web Application from Anywhere Over the Internet

- *Note 1:* The developed web application is implanted and tested successfully as it discussed above. However, any user can control the air conditioner remotely within the local network only. In order to be able to control the air conditioner from the internet outside your local network, you need a public IP address for your Raspberry Pi 3. In this project, I used Weaved services.

Weaved is a free software to be installed in Raspberry Pi and enables the user to connect to this Raspberry Pi and access its hosted web pages over internet from anywhere. In fact, Weaved provides Internet of Things (IOT) Kit to be used in Raspberry Pi. Weaved offers many services such as: SSH on port 22, Web (HTTP) on port 80, VNC on port 5901 and custom TCP connection (Sangesari, 2015).

q) Setting Up Weaved Software in Raspberry Pi 3:

1. Create a free account in Weaved website:
   https://developer.weaved.com/portal/index.php
2. Power on your Raspberry Pi 3.
3. Connect your Raspberry Pi 3 to Wi-Fi.
4. Open terminal in your Raspberry Pi 3.
5. Type the following command to download Weaved software package using *wget* utility:

```
wget
https://github.com/weaved/installer/raw/master/binarie
s/weaved-nixinstaller_1.2.13.bin
```

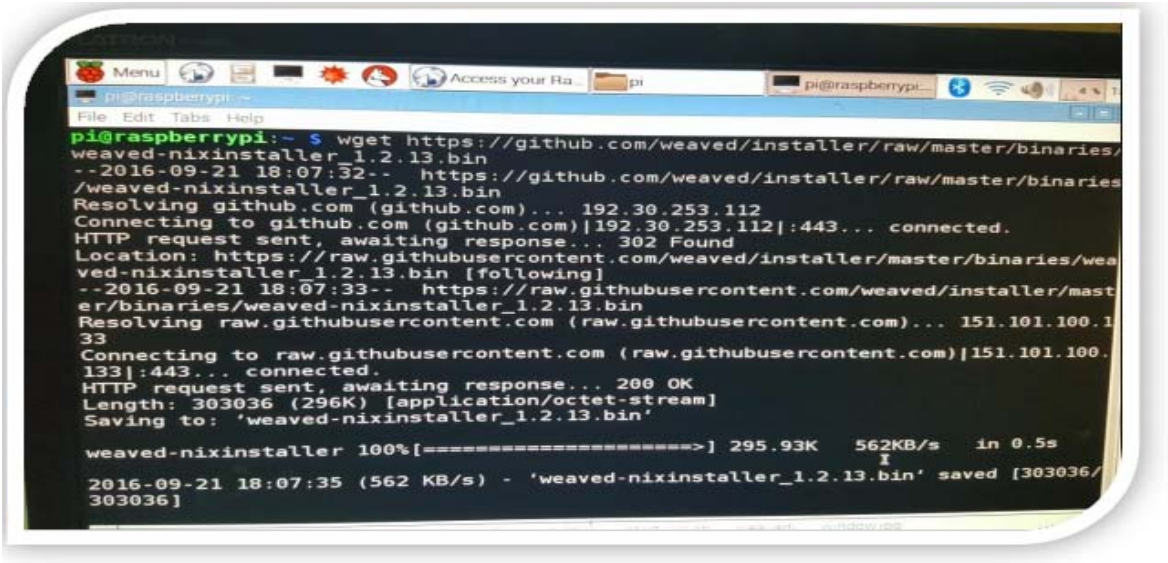


Figure 37: Screenshot of downloading weaved installer using wget utility

6. Type the following command to make the installer executable:

```
chmod +x weaved-
nixinstaller_1.2.13.bin
```

7. Type the following command to launch the executable installer:

```
./weaved-nixinstaller_1.2.13.bin
```

8. Select the service you want from the listed services. In my project I select:

Web (HTTP) on default port 80

❖ *Note 1:* You will be asked in you want to continue with the default assigned port which is 80. If you decide to keep it the same as default, type y. If you decide to change the port, type n and follow the instructions. In my project, I chose to keep the default port 80.

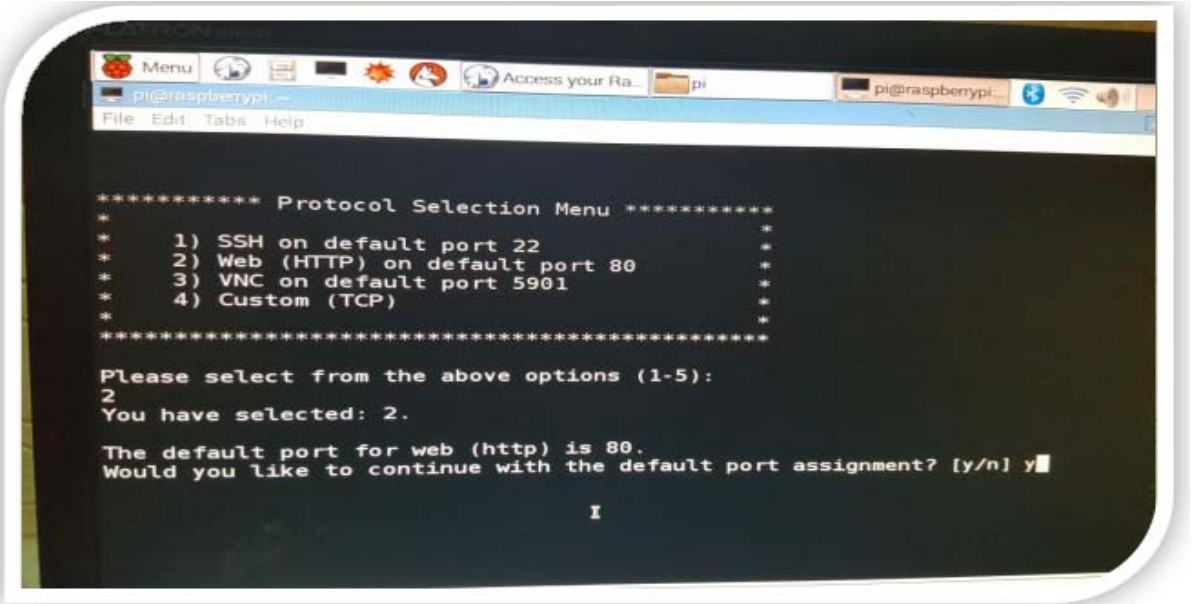


Figure 38: Select the required weaved service

9. Enter your username which is the email address you have created in step 1 above.
10. Enter your password for your username.

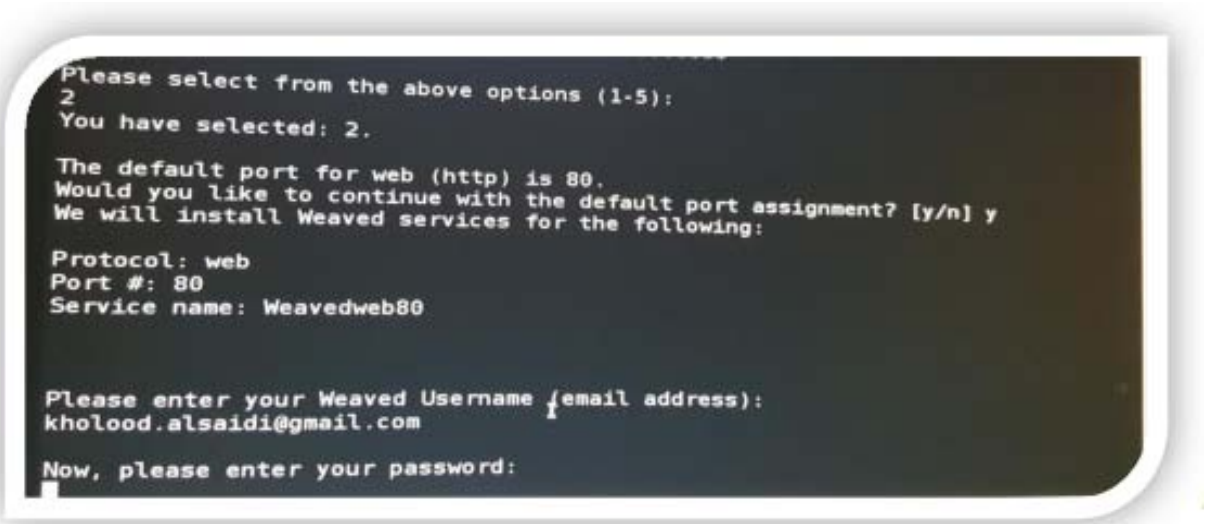


Figure 39: Login details for waved web services

❖ *Note 2:* You will be asked to enter an alias for your device, type the name you admire. In my project, I typed*: iotac*

Figure 40: Register the device with Weaved backend service

11. Wait until installation is done.

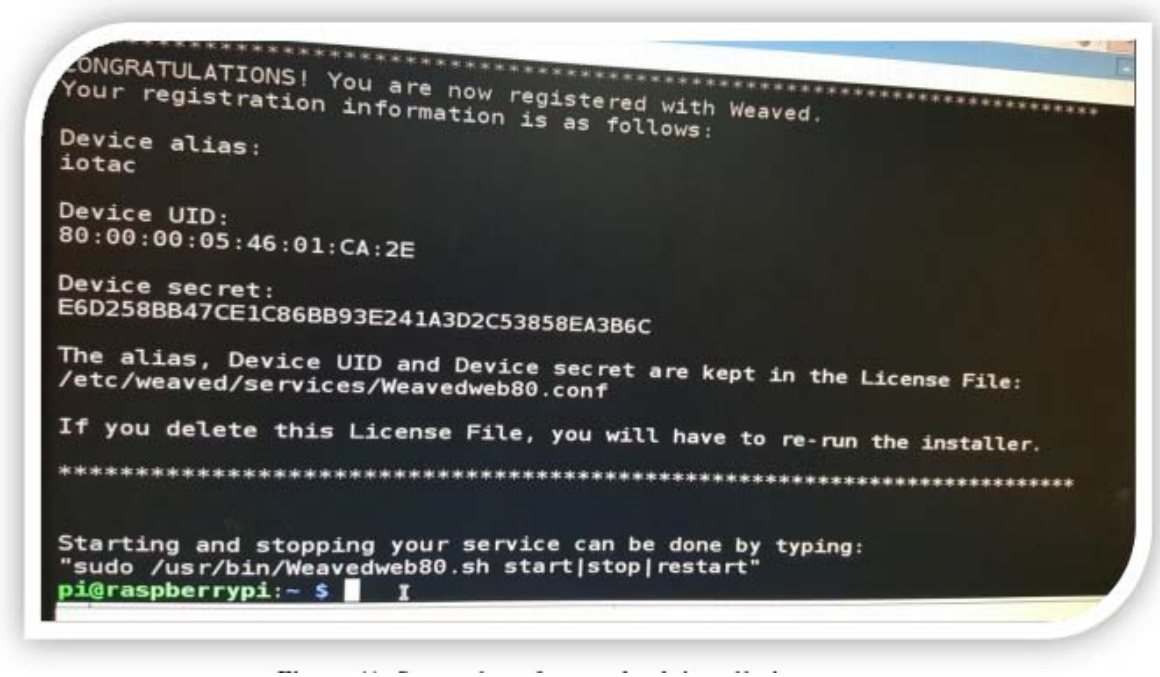


Figure 41: Screenshot of weaved web installation

12. Sign into your Weaved service in the following link:
https://developer.weaved.com/portal/login.php?error=NoSession

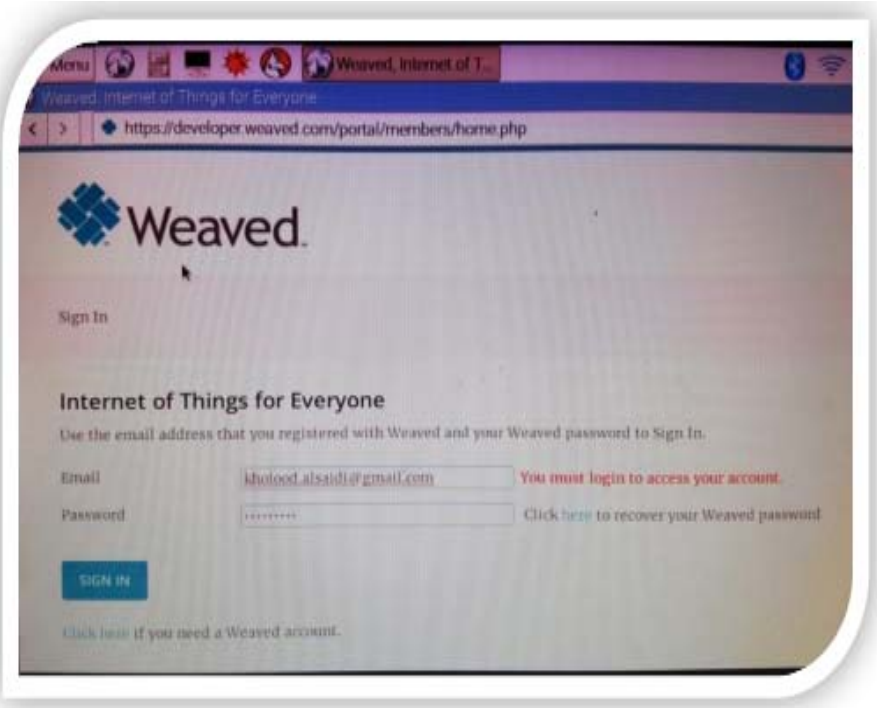


Figure 42: Login Page of Weaved service

13. Navigate to Your current list of services, you must be able to see that your selected service is listed as shown

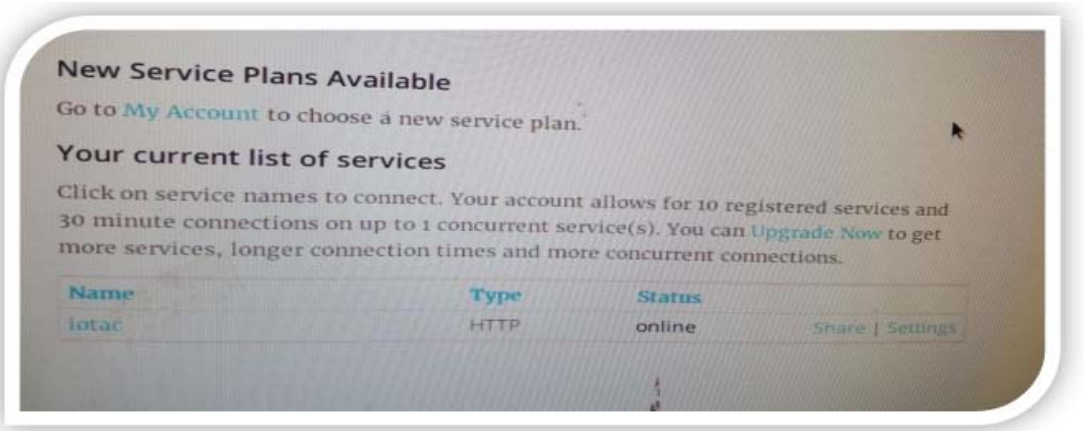


Figure 43: Weaved service page

14. Click on the *alias for your services listed under Name column* in order to connect your service. Then your service settings will be displayed.

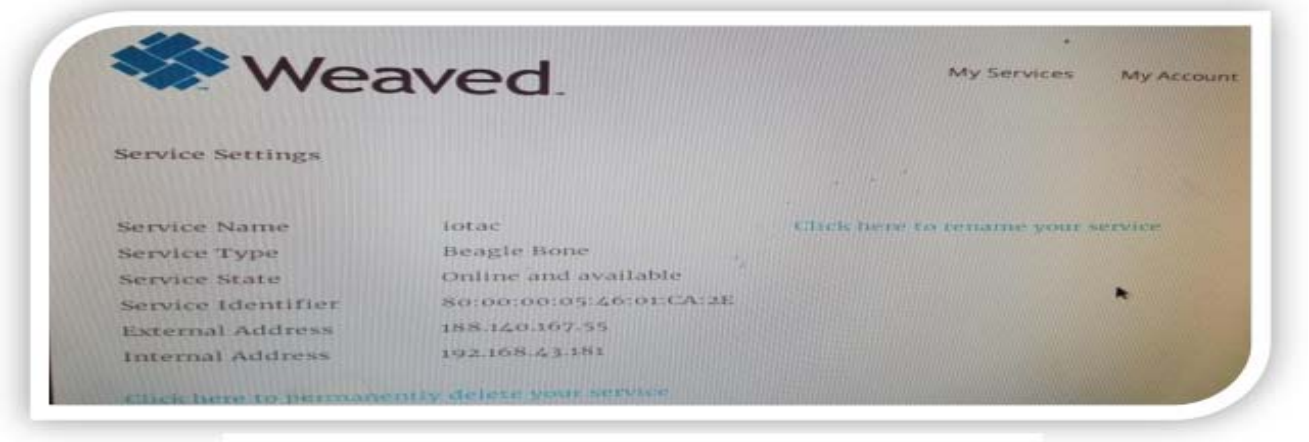


Figure 44: IP address to connect to service

15. When your service is connected it will display apache server web page says It works! It means your service is up now. Take the address after http:// and paste it in your web browser with the name of your web page in your Raspberry Pi3. In my project: `pjgwefdz.p6.weaved.com` is the address of after http://

Figure 45: Apache2 Debian Default Page

16. I opened my web browser and typed the following in the address bar:

http:// pjgwefdz.p6.weaved.com/login.php

❖ *Note 3:* Note that each time you connect to your service; the address is going to be different. It is not practical from one side but from the other side you may consider this as a security in case of any hacker eaves-dropped your address.

❖ *Note 4:* It is not practical to run Weaved service each time. As a result, I will configure it to start automatically when Raspberry Pi starts up.

1. Open terminal in Raspberry Pi 3.
2. Type the following command to open crontab editor:

```
sudo crontab –e
```

3. Add the following lines and exit the editor:

```
@reboot sudo
/usr/bin/Weavedweb80.sh start &
```

4. Press ctrl + x to exit the editor and press Y when it prompts you to save changes using CTRL + X
5. Reboot your Raspberry Pi 3 in order to take effect of made changes using the command:

```
sudo reboot
```

V. Integration and Testing

a) Integration

Now implementing and configuring each circuit was done successfully. The second stage of implementation is integration. Integration means combining all individual circuits in one solderless bread board in order to create one complete circuit. The integrated circuit is illustrated in Figure11.

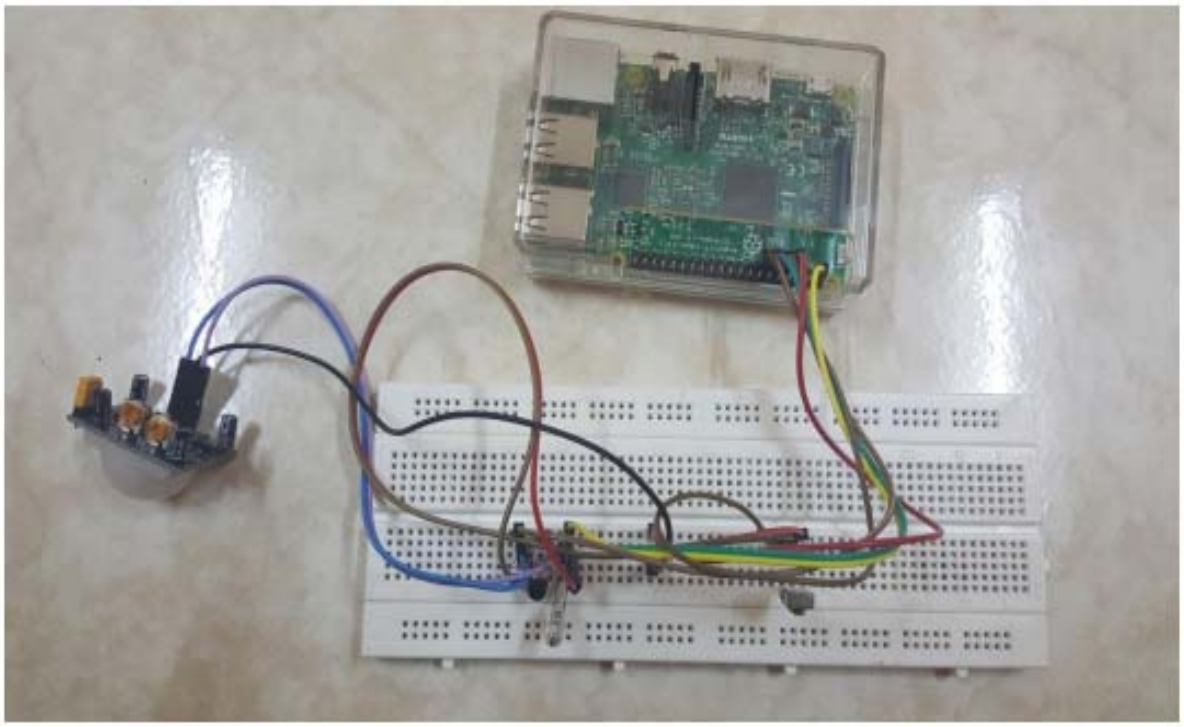


Figure 46: Smart Air Conditioner Using Internet of Things Integrated Circuit

Testing:

Same testing techniques used in Section 4.3 Implementation and Testing can be applied in the integrated circuit. If same results are obtained, then your integration is done properly.

Results are discussed and illustrated in details in 4.5 Results Section.

b) Results

After implementing the smart air conditioner project successfully, the following figures illustrate the results of the project tested and carried out by a smart phone which does not belong to the same network the Raspberry Pi 3 is connected to. It means, the smart phone is somewhere away from the Raspberry Pi 3 network and they are connected over the internet using Weaved service.

Figure 47: login.php Web Page

Figure 48: Login Credentials

Figure 49: iotAC.php Web Page

Figure 50: The Air Conditioner is Turned Off

VI. Conclusion

a) Evaluation

In order to validate the implemented smart air conditioner, a questionnaire was conducted by 14 potential users. The collected results were analyzed using IBM SPSS Statistics software. IBM SPSS Statistics is a famous data analysis software package. It helps the user to address his/her analytical process starting from planning and collecting data, moving to analyzing, reporting and deploying data (IBM, 2016).

The followings are the obtained statistics per each question raised in the questionnaire.

1. I am interested on home automation services. Statistics

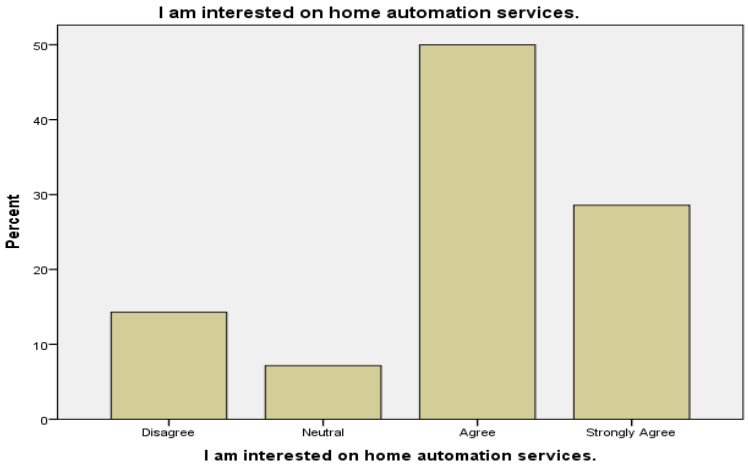
I am interested on home automation services.

N	Valid	14
	Missing	0
Mean		3.9286
Median		4.0909[a]
Maximum		5.00

a. Calculated from grouped data.

I am interested on home automation services.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Disagree	2	14.3	14.3	14.3
	Neutral	1	7.1	7.1	21.4
	Agree	7	50.0	50.0	71.4
	Strongly Agree	4	28.6	28.6	100.0
	Total	14	100.0	100.0	



2. The system helped me to control my air conditioner unit remotely from anywhere using any device with a web browser.

Statistics

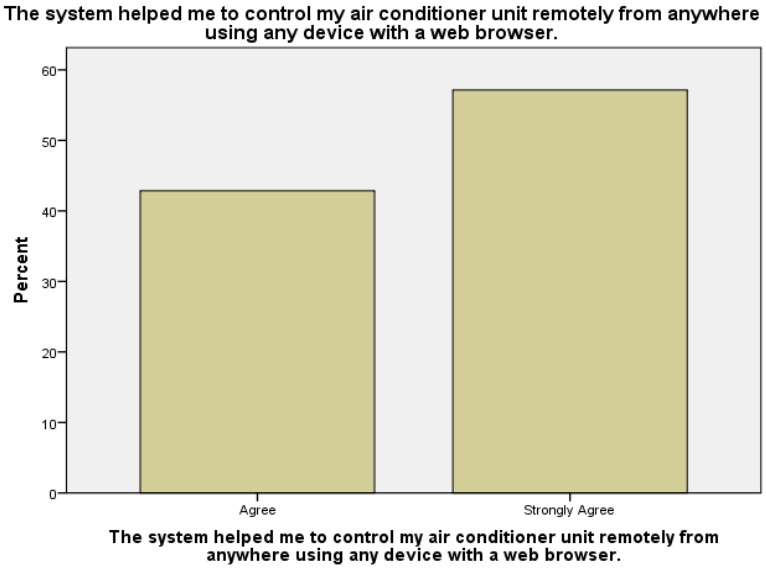
The system helped me to control my air conditioner unit remotely from anywhere using any device with a web browser.

N	Valid	14
	Missing	0
Mean		4.5714
Median		4.5714[a]
Maximum		5.00

a. Calculated from grouped data.

The system helped me to control my air conditioner unit remotely from anywhere using any device with a web browser.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Strongly	6	42.9	42.9	42.9
		8	57.1	57.1	100.0
	Total	14	100.0	100.0	



3. The system is easy to use.

Statistics

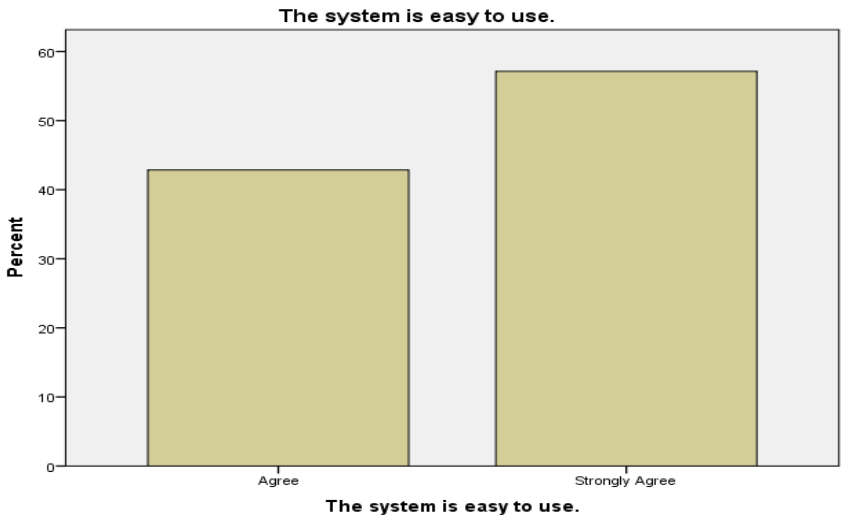
The system is easy to use.

N	Valid	14
	Missing	0
Mean		4.5714
Median		4.5714[a]
Maximum		5.00

a. Calculated from grouped data.

The system is easy to use.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Agree	6	42.9	42.9	42.9
	Strongly Agree	8	57.1	57.1	100.0
	Total	14	100.0	100.0	



4. I feel much more comfortable to use this system than the local remote control.

Statistics

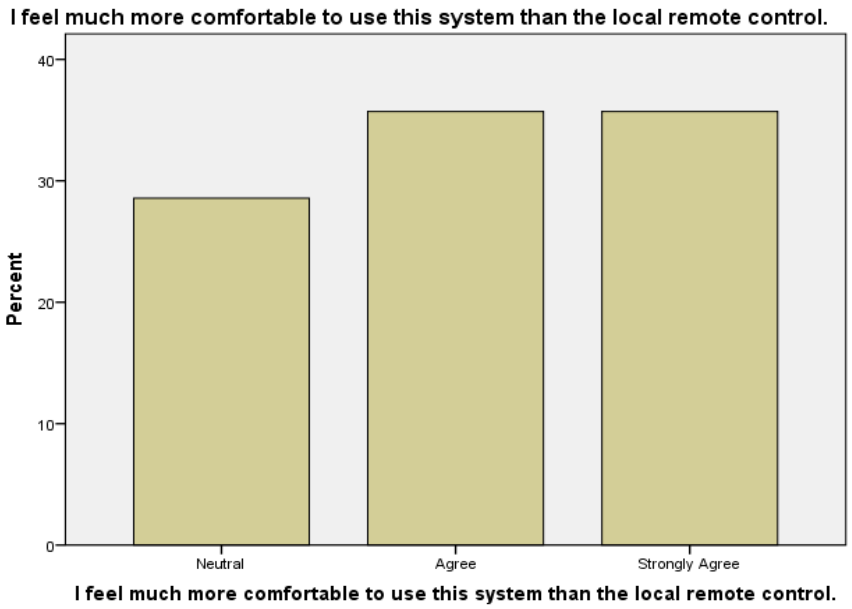
I feel much more comfortable to use this system than the local remote control

N	Valid	14
	Missing	0
Mean		4.0714
Median		4.1000[a]
Maximum		5.00

a. Calculated from grouped data.

I feel much more comfortable to use this system than the local remote control.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Neutral	4	28.6	28.6	28.6
	Agree	5	35.7	35.7	64.3
	Strongly Agree	5	35.7	35.7	100.0
	Total	14	100.0	100.0	



5. I recommend using this system as a product in technology market.

Statistics

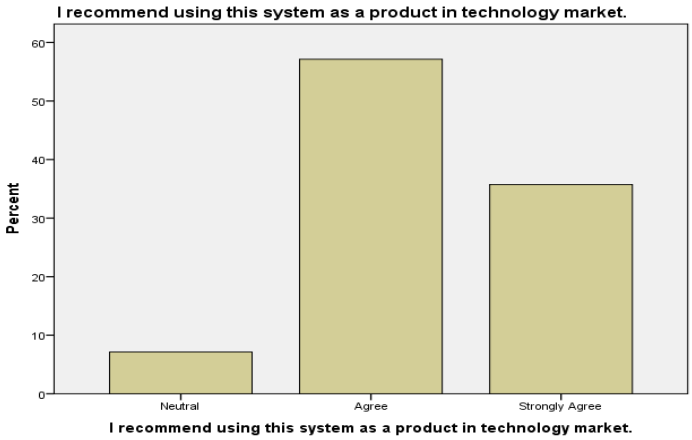
I recommend using this system as a product in technology market.

N	Valid	14
	Missing	0
Mean		4.2857
Median		4.3077[a]
Maximum		5.00

a. Calculated from grouped data.

I recommend using this system as a product in technology market.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Neutral	1	7.1	7.1	7.1
	Agree	8	57.1	57.1	64.3
	Strongly Agree	5	35.7	35.7	
	Total	14	100.0	100.0	100.0



Through reviewing the above obtained statistics, the implemented smart air conditioner product is obviously gaining trust of the potential users and accordingly the gained features from the implemented product are: The smart air conditioner has absolutely no inference against real remote control. The product is cost effective, energy efficient and achieves automation functionality indeed.

b) Summary

The local remote control is the traditional mechanism in which the end user controls the air conditioner. In the absence of this mechanism, the user loses the control. However, there is another mechanism in which the user may remotely control the air conditioner through Internet of Things (IoT) technology. A smart air conditioner using IoT was designed and implemented using Raspberry Pi 3 Model B device. Validity of the project was achieved through testing the implemented product by 14 potential users who own SANYO air conditioner. All potential users were able to control their air conditioner remotely over the internet from anywhere. The smart air conditioner has absolutely no inference against real remote control. The product is cost effective, energy efficient and achieves the required automation functionality.

c) Future Work

In future, I would like to expand this project to contain almost all controllable home appliances. A smart home automation system will absolutely help people control their home appliances remotely over the internet from anywhere.

Acknolodgments

I would like to seize the opportunity to openly give my special thanks to the following people who granted me their support and assistance during my Master's degree course. Dr. Vladimir Dyo for his precious supervision, assistance and comments during the course; Dr.Haider AL-Khateeb for his continuous directions and workshops over the course. My extreme thanks to people who took time completing my questionnaire. I would also like to offer my gratefulness to my family for their full encouragement and support to complete my Master's degree. Finally, my utmost thanks to my best friend for her continuous assistance over the past two years.

References Références Referencias

1. Adafruit, (no date), *NPN Bipolar Transistors (PN2222) - 10 pack* [image]. Available at: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved-=0ahUKEwiph7GQwJ7PAhXIfhoKHSscDAwQjRwIBw&url=https%3A%2F%2Fwww.adafruit.com%2Fproduct%2F756&bvm=bv.133387755,d.d2s&psig=AFQjCNF2Tqfao5I260UJsswEZwpaekoMNQ&ust=1474147981711608 [Accessed 6 September 2016].
2. Al-Ali, A.R. and Al-Rousan, M., 2004.Java-based home automation system. *IEEE Transactions on Consumer Electronics*, *50*(2), pp.498-504.
3. AliExpress, (2010), *wholesale ir receiver sensor* [image]. Available at: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiCtKPjtZ7PAhWCxxQKHfJ_CqcQjRwIBw&url=http%3A%2F%2Fwww.aliexpress.com%2Fw%2Fwholesale-ir-receiversensor.html&bvm=bv.133387755,bs.2,d.d2s&psig=AFQjCNFQSU4yWI7bHpwiD2b6bP1YC1v_gg&u=1471474477202999505 [Accessed 6 September 2016].
4. Anders, B. 2016. *Windows 10 IoT core support for raspberry pi 3*. [ONLINE] Available at:https://blogs.windows.com/buildingapps/2016/02/29/windows10-iot-core-support-for-raspberry-pi-3/.[Accessed 12 April 2016].
5. Azure Austin. (2015). *Windows 10 IoT on Raspberry Pi with Visual Studio and C# Universal Apps*. [Online Video]. 29 November 2015. Available from: https://www.youtube.com/watch?v=jZpW6EkLZ9U. [Accessed: 17 May 2016].
6. Baily, S., Paris, J. and Curtis, I. n.d. *WinLIRC*. [ONLINE] Available at:http://winlirc.sourceforge.net/. [Accessed 25 May 2016].
7. Bain, A. 2013. *Setting up LIRC on the RaspberryPi - alexba.in*. [ONLINE] Available at: http://alexba.in/blog/2013/01/06/setting-up-lirc-on-the-raspber rypi/. [Accessed 4 September 2016].
8. Bartelmus, C. 2016. *LIRC - Linux infrared remote control*. [ONLINE] Available at: http://www.lirc.org/. [Accessed 5 September 2016].
9. Buyapi, (2016), *HDMI TO VGA ADAPTER* [image]. Available at: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiE6lipwJ7PAhWLnRoKHZfFAkAQjRwIBw&url=https%3A%2F%2Fwww.buyapi.ca%2Fproduct%2Fhdmi-to-vgaadapter%2F&bvm=bv.133387755,bs.2,d.d2s&psig=AFQjCNHRqQT7GnYkvXB9xqoloZpKvQGKDg&ust=1474480035052921 [Accessed 6 September 2016].
10. Carretero, J. and García, J.D., 2014. The Internet of Things: connecting the world. *Personal and Ubiquitous Computing*, *18*(2), pp.445-447.
11. Celebre, A.M.D., Dubouzet, A.Z.D., Medina, I.B.A., Surposa, A.N.M. and Gustilo, R.C., 2015, December. Home automation using raspberry Pi through Siri enabled mobile devices. In *Humanoid. Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2015 International Conference on* (pp. 1-6).IEEE.
12. CES, (2016), *PIR motion sensor module* [image]. Availabl.at:https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjdlt2ytJ7PAhULWRoKHUlMAzAQjRwIBw&url=http%3A%2F%2Fwww.ceseshop.com%2Fdir%2Findex.php%3Froute%3Dproduct%2Fproduct%26product_id%3D691&bvm=bv.133387755,d.d2s&psig=AFQjCNEGvpCnTSxFrIwqDnZekI7WmMFHXQ&ust=1474476830366270 [Accessed 6 September 2016].
13. Circuit Specialists, (2016), *Solder less Breadboards* [image]. Available at: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiToDDwJ7PAhUGExoKHftUDhoQjRwIBw&url=https%3A%2F%2Fwww.circuitspecialists.com%2Fsolderlessbreadboards&bvm=bv.133387755,d.d2s&psig=AFQjCNFhZvVC-7Ez-p2-sTwv9bYI0FB3_g&ust=1474480075353879 [Accessed 6 September 2016].
14. Gadgetar, (2016), *Raspberry pi 3 model B $30* [image]. Available at: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwilILnVtJ7PAhVHWRQKHRnNB4UQjRwIBw&url=http%3A%2F%2Fwww.gadgetar.com%2Fraspberry-pi-3-modelb%2F&psig=AFQjCNH0c28fbs0_m01DowQzmdbYZW95Q&ust=1474476903101223&cad=rjt [Accessed 6 September 2016].
15. Ganesan, A. 2015. *WARAN - Home Automation*. [ONLINE] Available at: https://www.hackster.io/arjun/waran-home-automatione84a26?ref=challenge&ref_id=15&offset=11.[Accessed 25 May 2016].
16. Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M., 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), pp.1645-1660.
17. Harrington, W., 2015. *Learning Raspbian*. Packt Publishing Ltd.
18. IBM. 2016. *SPSS Statistics*. [ONLINE] Available at: http://www-03.ibm.com/software/products/en/spsstatistics . [Accessed 9 September 2016]. Ivancreations.com. 2016. *Control Air Conditioner with RaspberryPI and a LED*. [ONLINE] Available at: http://www.ivancreations.com/2015/04/control-air-conditioner-with.html. [Accessed 3 May 2016].
19. Kopetz, H., 2011. Internet of things. In *Real-time systems* (pp. 307-323).Springer US. Kortuem, G., Kawsar, F., Fitton, D. and Sundramoorthy, V., 2010.Smart objects as building blocks for the internet of things. *Internet Computing, IEEE*, *14*(1), pp.44-51.

20. Ma, H.D., 2011. Internet of things: Objectives and scientific challenges. *Journal of Computer science and Technology*, *26*(6), pp.919-924.
21. Meetup. 2015. *Building Internet of Things using C#*. [ONLINE] Available at: http://www.meetup.com/NET-Developers-Association/events/220140826/. [Accessed 21 May 2016].
22. MODMYPI. 2016. *How to give your Raspberry Pi a Static IP Address - UPDATE*. [ONLINE] Available at: https://www.modmypi.com/blog/how-to-give-your-rspberry-pi-a-static-ip-address-update. [Accessed 4 September 2016].
23. Gillett, M. 2015. Low-cost Home Automation with Voice Control. [ONLINE] Available at: https://www.hackster.io/michael-gillett/dorm-automation-9fed01?ref=challenge&ref_id=15&offset=20. [Accessed 25 May 2016].
24. Microsoft. 2016. *'Hello, World!' Sample*. [ONLINE] Available at: https://developer.microsoft.com/enus/windows/iot/win10/samples/helloworld. [Accessed 14 May 2016].
25. Microsoft. 2016. *Learn about windows 10 IoT core*. [ONLINE] Available at: https://developer.microsoft.com/en-us/windows/iot/explore/iotcore. [Accessed 12 April 2016].
26. Microsoft. 2015. *Microsoft Projects*. [ONLINE] Available at: https://developer.microsoft.com/enus/windows/iot. [Accessed 12 April 2016].
27. Microsoft. 2016. *The Internet of your things*. [ONLINE] Available at: https://developer.microsoft.com/enus/ windows/iot. [Accessed 10 April 2016].
28. Miorandi, D., Sicari, S., De Pellegrini, F. and Chlamtac, I., 2012. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), pp.1497-1516. net-tutorials. 2006. *The complete C# Tutorial*. [ONLINE] Available at: http://csharp.net-tutorials.com/. [Accessed 22 May 2016].
29. Microsoft. 2016. *Overview of Visual Studio 2015 Products*. [ONLINE] Available at: https://www.visualstudio.com/en-us/products/vs-2015-product-editions.aspx. [Accessed 15 April 2016]. raspberrypi.org. 2016. *Build a LAMP Web Server with Word Press*. [ONLINE] Available at: https://www.raspberrypi.org/learning/lamp-web-server-with-wordpress/worksheet/. [Accessed 5 September 2016].
30. Rieger, C. 2016. *Raspberry Pi Blind & AC Controller*. [ONLINE] Available at: http://chrisrieger.com/projects/roomautomation. [Accessed 24 May 2016].
31. SanDisk , (n.d), *8GB Sandisk MicroSDHC Memory Card with SD Adapter* [image]. Available at: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwirocbHtZ7PAhVBnBQKHXNIB5QQjRwIBw&url=https%3A%2F%2Fwww.amazon.com%2FSandisk-MicroSDHC-Memory-Card-Adapter%2Fdp%2FB000WH6H1M&bvm=bv.133387755,bs.2,d.d2s&psig=AFQjCNGikpcggSgXwgyDN4ZphEhLxO3lyA&ust=1474477100739052 [Accessed 6 September2016].
32. Sangesari, R. 2015. *Access your Raspberry Pi over the Internet*. [ONLINE] Available at: https://www.hackster.io/idreams/access-your-raspberry-pi-over-the-internet-157ad1. [Accessed 9 September 2016].
33. Solarbotics, (n.d), *40 pin Female-to-Female Jumper Cable 45072* [image]. Available at: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiXvsjhwJ7PAhXHExoKHeLkDGwQjRwIBw&url=https%3A%2F%2Fsolarbotics.com%2Fproduct%2F45072%2F&bvm=bv.133387755,d.d2s&psig=AFQjCNEm2gdG6H2KoqoqGovHCVtqsQFcA&ust=1474480144244078 [Accessed 6 September 2016].
34. Sriskanthan, N., Tan, F. and Karande, A., 2002. Bluetooth based home automation system. *Microprocessors and Microsystems*, *26*(6), pp.281-289.
35. Steve Teixeira. 2015. *Hello, Windows 10 IoT Core*. [ONLINE] Available at: https://blogs.windows.com/buildingapps/2015/08/10/hello-windows-10-iot-core/. [Accessed 12 April 2016].
36. Tao, F., Zuo, Y., Da Xu, L. and Zhang, L., 2014.IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing. *Industrial Informatics, IEEE Transactions on*, *10*(2), pp.1547-1557.
37. TechshopBD, (2012), *IR Transmitter - White* [image]. Available at: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiD9M39tZ7PAhXGXRQKHeOzDNQQjRwIBw&url=https%3A%2F%2Fwww.techshopbd.com%2Fproductcategories%2Flight%2F743%2Fir-transmitter-white-techshopbangladesh&bvm=bv.133387755,bs.2,d.d2s&psig=AFQjCNF4vTn5HtAl0ADxrLqbKsBQEL9MQ&ust=1474477258911023 [Accessed 6 September 2016].
38. Vasanwala, A.S. 2015. Home Automation using Raspberry Pi 2 and Windows 10 IoT. [ONLINE] Available at: https://www.hackster.io/AnuragVasanwala/home-automation0dcefc?ref=challenge&ref_id=15&offset=2. [Accessed 25 May 2016].
39. Vis, P.J., (n.d), *10k / 10k ohm Resistor Colour Code* [image]. Available at: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwiVh_bTvZ7PAhXBlxoKHUcDCewQjRwIBw&url=http%3A%2F%2Fwww.petervis.com%2Felectronics%2FStandard_Resistor_Values%2F10K.html&bvm=bv.133387755,d.d2s&psig=AFQjCNHvwYc-WVntd_DyEVBHB8z7TNG7w&ust=1474479311654384&cad=rjt [Accessed 6 September 2016].
40. WiringPi. 2016. WiringPi. [ONLINE] Available at: http://wiringpi.com/. [Accessed 4 September 2016].

Wortmann, F. and Flüchter, K., 2015.Internet of Things.

41. Xia, F., Yang, L.T., Wang, L. and Vinel, A., 2012.Internet of things. *International Journal of Communication Systems*, *25*(9), p.1101.
42. Zhu, J., Gao, X., Yang, Y., Li, H., Ai, Z. and Cui, X., 2010, September. Developing a voice control system for zigbee-based home automation networks. In *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content* (pp. 737-741).IEEE.

Appendix a: User Satisfaction Questionnaire

User Satisfaction Questionnaire

Dear SANYO split air conditioner users; Smart Air Conditioner Using Internet of Things Product helps you to remotely control your air conditioner through a web application from any windows physical device such as a desktop, a laptop, a PAD and a smart phone you possess. The implemented smart air conditioner would be able to turn off by itself when people are not present and save energy. Results of the survey will be treated with full confidentiality and it will be stored in a secure place. Likewise, the obtained results will be used to monitoring your satisfaction as a customer and to identifying improvements to the product.

I am looking forward to your cooperation to conduct the questionnaire.

Gender: Age: .....

o Male o Female

---

1. I am interested on home automation services.

o Strongly Disagree
o Disagree
o Neutral
o Agree
o Strongly Agree

2. The system helped me to control my air conditioner unit remotely from anywhere using any device with a web browser.

o Strongly Disagree
o Disagree
o Neutral
o Agree
o Strongly Agree

3. The system is easy to use.

o Strongly Disagree
o Disagree
o Neutral
o Agree
o Strongly Agree

4. I feel much more comfortable to use this system than the local remote control.

o Strongly Disagree
o Disagree
o Neutral
o Agree
o Strongly Agree

5. I recommend using this system as a product in technology market.

o Strongly Disagree
o Disagree
o Neutral
o Agree
o Strongly Agree

Thank you for your valuable time…

Appendix B: Implementation Source Code

Section I: PIR Sensor Source Code:

```
import time
import os
import RPi.GPIO as gpio
pir = 4 // define PIN 4 in Raspberry PI 3 GPIO as a variable called pir
gpio.setmode(gpio.BCM)
gpio.setup(pir, gpio.IN) // define PIN 4 in Raspberry PI 3 GPIO as an INPUT PIN
n = 0
while True: // While loop
ir = gpio.input(pir)
print ir
if ir == 0: // when output from PIR Sensor is LOW
n = n+1
time.sleep(1) // set the time delay equals to 1 second
if n > 1800:
n = 2000
else:
n = 0
time.sleep(1) // set the time delay equals to 1 second
/* if PIR Sensor does not sense any presence or motion, it will ask Raspberry Pi 3 to turn off the ac
automatically */
if n == 1800:
os.system("irsend SEND_ONCE ir1.conf KEY_POWER2")
time.sleep(10)
```

Section II: iotAC.php Source Code:

```
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title>AC Control</title> // set the name of the web page to AC Control
</head>
<body>
<?php
echo "<body style='background-color: PowderBlue'>"; // set the background color of the web page
echo'
<html>
/* Insert image into the web page */
<img src="ac.jpg" width="330" height="190" title="AC Control" alt="AC Control" vspace="none"
/>
</html>
';
?>
<br><br> // create a break = line space
AC Remote Control:
<br><br><form method="get" action="iotAC.php">
<input type="submit" value="ON" name="on" style="width: 50px; hight: 50px;">
<input type="submit" value="OFF" name="off" style="width: 50px; hight: 50px;"><br>
<input type="submit" value="+" name="p" style="width: 50px; hight: 50px;">
<input type="submit" value="-" name="n" style="width: 50px; hight: 50px;">
</form>
<?php
/* If user click on ON button, Raspberry Pi will execute a shell command to turn on the AC */
if(isset($_GET['on'])){
$gpio_on = shell_exec("irsend SEND_ONCE ir1.conf KEY_POWER");
```

```
echo "AC is on"; // display 'AC is on' message
}
/* If user click on OFF button, Raspberry Pi will execute a shell command to turn off the AC */
else if(isset($_GET['off'])){
$gpio_off = shell_exec("irsend SEND_ONCE ir1.conf KEY_POWER2");
echo "AC is off"; // display 'AC is off' message
}
/* If user click on + button, Raspberry Pi will execute a shell command to increase temperature volume of
the AC */
else if(isset($_GET['p'])){
$gpio_p = shell_exec("irsend SEND_ONCE ir1.conf KEY_UP");
echo "Temperature is increased"; // display 'Temperature is increased' message
}
/* If user click on - button, Raspberry Pi will execute a shell command to decrease temperature volume of
the AC */
else if(isset($_GET['n'])){
$gpio_n = shell_exec("irsend SEND_ONCE ir1.conf KEY_DOWN");
echo "Temperature is decreased"; // display 'Temperature is decreased' message
}
?>
</body>
</html>
```

Section II: login.php Source Code:

```
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title>AC Control</title> // set the name of the web page to AC Control
</head>
<body>
<div align="center"> // make content of the web page in the center
<?php
echo "<body style='background-color:PowderBlue'>"; // set the background color of the web page
?>
<form id='login' action='submit.php' method='post' accept-charset='UTF-8'>
<fieldset >
<legend>Login</legend>
<input type='hidden' name='submitted' id='submitted' value='1'/>
<label for='username' >UserName</label> // create a button on the web page named UserName
/* specify the type of UserName button is a text and set the length of the entered username equals to 100
*/
<input type='text' name='username' id='username' maxlength="100" />
<label for='password' >Password</label> // create a button on the web page named Password
/* specify the type of Password button is a text and set the length of the entered password equals to 100 */
<input type='password' name='password' id='password' maxlength="100" />
/* create a button on the web page named Submit and specify the type of it as submit */
<input type='submit' name='Submit' value='Submit' />
</fieldset>
</form>
</body>
</html>
```

Section IV: submit.php Source Code:

```
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title>AC Control</title> // set the name of the web page to AC Control
</head>
```

```
<body>
<div align="center">
<?php
echo "<body style='background-color:PowderBlue'>"; // set the background color of the web page
session_start();
/* Declare Variables */
$field1 = $_POST['username']; // this is the username variable posted from login.php web page
$field2 = $_POST['password']; // this is the password variable posted from login.php web page
/* If the user missed enter the username and password in login.php page */
if (($_POST['username'] == "") || ($_POST['password'] == "")){
/* display sorry.. There are some missing required information message */
echo"sorry.. There are some missing required information ";
include("login.php"); // direct the user to login.php web page
echo'<br>'; // create a break = line space
echo'<br>';
echo'<br>';
echo'<br>';
die(mysql_error());
exit;
}
include("header.php"); // refer to header.php web page for required information to connect to MySQL
server
/* connect to MySQL server using host, username and password declared in header.php */
mysql_connect($host,$username,$password );
/* select the database declared in header.php */
mysql_select_db($database);
/* query the database to select all records in login table in ac_control database */
$query= "SELECT * FROM login";
/* save data received from performing the above query in a variable called result */
$result=mysql_query($query);
/* If the query failed to perform the action */
if (!$result) {
/* display 'Could not run query: 'message and exit MySQL Server */
echo 'Could not run query: ' . mysql_error();
exit;
}
/* declare a variable called num that saves the number of rows in login table */
$num=mysql_numrows($result);
$i=0;
// While loop: as long as the variable i is less that the number of rows in login table, fetch the row of the table and
check if row[0] which is username in login table equals to the username posted from login.php and row[1] which is
password in login table equals the password posted from login.php, direct the user to iotAC.php web page. If not,
display 'error' message */
while($i< $num){
$row= mysql_fetch_row($result);
if (($row[0]==$field1) && ($row[1]==$field2))
{
include("iotAC.php");
}
else
{
echo 'error';
}
$i++;
}
?>
```

Section V: header.php Source Code:

```
<?php
/* set the 'root' username used to access MySQL Server in a variable called username */
$username="root";
/* set 'mysql' password used to access MySQL Server in a variable called password */
$password="mysql";
/* set ac_control database in a variable called database */
$database="ac_control";
/* set localhost (the location where the database is stored) in a variable called host */
$host="localhost";
?>
```

Appendix C:

Smart Air Conditioner using Internet of Things Poster