SMART UNIVERSAL REMOTE USING MOBILE FOR HOME AUTOMATION

By

Chong Chin Mun

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2021

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: SMART UNIVERSAL REMOTE USING MOBILE HOME AUTOMATION

Academic Session: JAN 2020

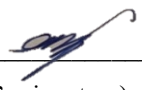
I CHONG CHIN MUN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)

TEOH

(Supervisor's signature)

Address:
NO5, Jalan Sungai Long 11/1,
Bandar Sungai Long 43000
Selangor

Teoh Shen Khang

Supervisor's name

Date: 16 April 2021

16 April 2021

Date:

Smart Universal Remote Using Mobile for Home Automation

By

Chong Chin Mun

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2021

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Smart Universal Remote Using Mobile for Home Automation**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : CHONG CHIN MUN

Date : 16/4/2021

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Mr Teoh Shen Khang who has given me this bright opportunity to engage in project of **Smart Universal Remote Using Mobile for Home Automation**. It is my first step to establish a career in my future. A million thanks to you.

Also, thanks a million, Ts Ooi Joo On for moderating my project and giving some suggestion in my projects. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

This project is a Smart Universal Remote design project for academic purposes. It will provide students with the methodology, concept, and design of hardware implementation. In this project, Raspberry Pi 3 Model B+ is a small CPU as a central part of the project with Linux Infrared Remote Control (LIRC). LIRC is an open-source library that allows users to receive and send IR signals with a Linux-based computer system, Raspberry Pi used in this project. IR receiver and IR LEDs must be implemented in the hardware circuit to receive an IR signal sent by IR remote control. Those remote control can be found around the home quickly such as light, fan, TV, home theatre speaker system, and more home devices. After recorded the IR signals as the configure files. Python GUI will use to control the home devices that they have recorded earlier. Then, Ubidots, an IoT platform, is used to make the user control the home device. Using MQTT within Ubidots so that the device can publish the data to the connected and subscribed topic. Then, Telegram Bot used as an additional option to control the IR home devices with a smartphone.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

°C	Degree Celsius

LIST OF ABBREVIATIONS

IoT	Internet of Things
RPi	Raspberry Pi
URC	Universal Remote Control
IR	Infrared radiation
GUI	Graphical User Interface
LIRC	Linux Infrared Remote Control
LED	Light-Emitting Diode
A/C	Air Conditioner
TV	Television
PCM	Pulse-Code Modulation
VNC	Virtual Network Computing
VCC	Common Collector Supply Voltage
GND	Ground
GPIO	General-purpose input/output
OS	Operating System
IP	Internet Protocol
MQTT	MQ Telemetry Transport

Chapter 1: Introduction

1.1 Problem Statement

The project title of this project is “Smart Universal Remote using mobile for home automation”, this project is consisting the element of which in terms of the Internet of Things (IoT) and Smart Home. Hence, the problem statement of this project has been told by the project title already. This project looks for Smart Universal Remote for home automation to control such as light, fan, TV, home theatre speaker system, and more home devices. From this example, we know that there are many types of home devices in a house nowadays controlled by a remote controller. However, many types of home devices require various remote controls. Therefore, what if we try to construct a Universal Remote Control to make the remote 3 in 1 or many in 1 to allow people to control their Universal Remote Control, for example, our smartphone.

1.2 Background

In the 21st century, the Internet of Things, commonly abbreviated as IoT, has become high-profile and one of the most critical technologies over the past few years. With the concept of the Internet of Things, cars, home appliances (television, coffee machine, washing machines) and even baby monitor as long as connected and shared data to the Internet with embedded devices. Today, we can easily find out that electronic and electrical appliances have Wi-Fi capability because there was embedded Wi-Fi. Besides offering smart devices, IoT is crucial to businesses. For example, when we went to dining, we noticed that the waiter recorded the order from customers by an online app to avoid a mistake that worsened customer satisfaction. Also, cashless payment becomes a trend now. Therefore, IoT is essential because it helps people improve their daily lives to become more innovative and help companies operate their business more effectively. One of the common ideas that lead from IoT is Smart Home.

According to (Rouse.M 2018), Smart Home is a house that uses internet-connected devices to enable management and monitoring of home appliances also referred to as Home Automation. A smart home gives the user control of internet-connected home

appliances such as smart television, washing machines, rice cookers, etc., from a mobile device or other networked devices. For example, we can tell a simple command to Google Home speaker with "Hey Google, turn on all the switches.". Then, the bulbs that in Wi-Fi-connected will lights up after the Google Home decoded the voice command. We can also use Google Home apps to control our home appliances connected with Wi-Fi by using smartphones.

For example, home automation devices are popular nowadays, for example, electrical home devices and HVAC systems that abbreviate as heating, ventilation, air-conditioning, and thermostats. According to (cooking hacks, n.d.), the main objective of HVAC systems is helping to maintain better indoor air quality through adequate ventilation with filtration and provide better thermal comfort. We can find any one of the home devices that are HVAC systems such as A/C, dehumidifier, air filter, or even fan. Therefore, this is very general that many home devices can control by IR remote control.

According to (Woofford 2019), IR remote control uses electromagnetic waves to operate in infrared radiation. However, human eyes cannot detect IR but when can see it by phone camera. When the remote-control button presses, it will generate a systematic series of 1/0 infrared pulses, which a binary code. For example, the TV remote sends ‘11100011’ as a signal to allow the TV on and off while sending ‘10011001’ to allow TV to change the channel. Also, different manufacturers might have other codes of the same key. For instance, the power key of Toshiba TV is different from Samsung TV.

1.3 Motivation

If there are many remote-control types in a home, there will be troublesome because those remote controls might occupy home spaces, and it is difficult to find when the remote control had hidden or missing somewhere else. Especially, there is no IR universal remote which can control all the home device by a phone. Therefore, to solve this problem, there is enough motivation to solve and combine all the IR remote control found in the home, commonly like a fan, TV, and A/C remote, into a prototype for testing. Then, use the IoT platform to send those IR signals from the prototype to control home devices, which can control by phone.

1.4 Objectives

The main objective and sub-objectives of the project are shown below:

1. To design a Universal Remote Control that can be able to control by phone to communicate with IR-controllable home appliances.
    - To build a hardware prototype which able to record and send IR signals
    - To receive the IR signal from the IR remote control from the IR remote.
    - To learn and store the new IR signal bit pattern from the IR remote control
    - To transmit the pre-recorded IR signal from the Universal Remote Control to IR device to control it
    - To control at least three IR home appliances like AC, TV and Fan
    - To control the Smart Universal Remote by smartphone via IoT Platform
    - To control the Smart Universal Remote by smartphone via phone app

Therefore, IR signals sent by remote controls are our main keys to undergo this project. If without recording the IR signals of each remote control for controlling the home devices, this project cannot continue smoothly. Also, this project targets to control the home devices without using any IR remote controls.

1.5 Proposed approach/study

Throughout the project, to solve the problem statement which has mentioned above. This project needs to design a hardware prototype with an IR receiver and IR transmitter that can record and save the IR signal data into a file with a library. Then, send and test the IR signal to the IR home devices by executing the recorded IR signal data. This project also involves using an IoT platform which able to control by a smartphone via the Internet. The diagram below is shown a simple system flowchart for the project of the proposed approach.

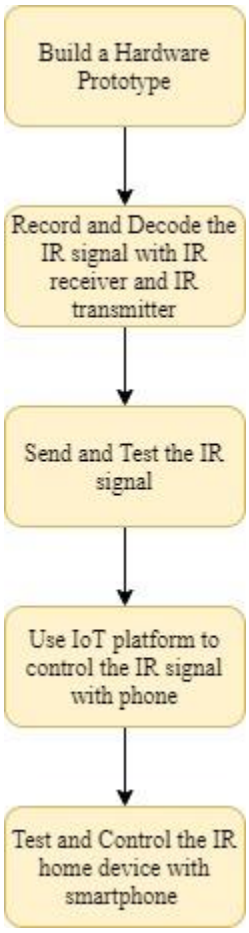


Figure 1.1 Simple System Flowchart

To make the scope narrow and sharply, we target IR-controllable home appliances and then target the IR remote control on those common home appliances at least three examples such as Air Conditional, TV, and Fan. We only target those common and frequent being use button keys such as the power key, fan speed key. Of course, the scope would expand to challenge them further.

1.6 Highlight of what have been achieved

This project is going to benefit the reader who desired to build a Smart Universal Remote for IR home devices by themselves.

A universal remote control is useful because it can be a combination of infrared remote control among several home appliances. Otherwise, it is painful for the user to find a particular remote control to control the appliances. Also, universal remote control able to save some costs the user must purchase again the replacement original remote control which costs highly. Hence, a universal remote control has the inventive principle of merging that has the value to study to let user ease to operate with one universal remote control. However, this project possible to add more functions, for example, the hardware prototype can connect a temperature sensor to detect the surrounding temperature when high temperature, the sensor will be interrupted and send the IR signal to the A/C for cooling down.

However, many remote controls are still in analog form since IR is a wave of electromagnetic waves. Thus, this project proposed to digitize the signal of IR which we get from IR remote control so that it can enable and widen the application.

1.7 Report organization

The rest of this report has organized as follows:

- Chapter 1: Explain the Smart Universal Remote's introduction and background Using Mobile for Home Automation project.
- Chapter 2: Review and compare some literature the previous works which similar to this project.
- Chapter 3: Describes details of how the project is developed with necessary information.
- Chapter 4: Describes and list the methodology, tools, requirement, specification.
- Chapter 5: Describe the implementation and testing of this project.
- Chapter 6: Conclude this project what has done and achieved related to objectives, discuss the problems encountered and unique idea, and further improve this project.

Chapter 2: Literature Review

Chapter 2.1: IOT based Smart IR Device using CC3200

There are different methods used to propose a home automation system.

They investigated the issue of IoT-based Smart IR System using the CC3200 device, according to Chaitanya, Sekhar, and Ramesh (2016, p2). The CC3200 interface is a wireless microcontroller unit (MCU) with integrated Wi-Fi networking, according to Texas Instruments (2000, p.4), and it is suitable for use in IoT architecture. Their concept is to create a Smart IR system with a CC3200 as the core component, which is connected to IR transmitter LEDs and an IR receiver module. The IR receiver's job is to collect IR bit patterns from different home appliance remote controls. Their concept is to create a Smart IR system with a CC3200 as the core component, which is connected to IR transmitter LEDs and an IR receiver module. The IR receiver's job is to collect IR bit patterns from different home appliance remote controls. As a result, one of the goals is to create a system that can learn and store some kind of remote bit pattern from an IR remote control. The concept of the IR receiver circuit architecture is to receive IR remote bit patterns from the IR remote, as seen in this diagram. (Omprakash cited in Chaitanya, Sekhar and Ramesh, 2016, p.3)

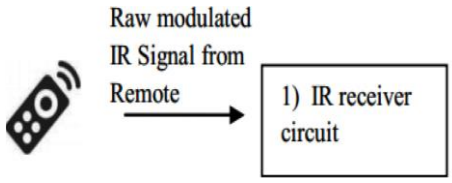


Figure 2.1 IR receiver circuit design to receive to IR remote bit patterns from IR remote.

After the IR receiver has learned the button key and stored the basic remoter bit pattern from the IR system in the smart IR device, the user must attach it to their home's Wi-Fi router or a 3G or 4G data link with a smartphone to monitor the corresponding IR device. Then, on the mobile application that has been saved in the SFLASH of the CC3200, open it and click the corresponding button. It retrieved the corresponding button code from SFLASH intermediate and sent it to home appliances including the television and air conditioner via IR transmitter LEDs.

The benefit of this Smart IR interface solution is that it can learn and store remote bit patterns from an IR remote control in SFLASH of the CC3200. However, due to increasing advancements in electronic devices from different manufactures, creating a single IR-based universal remote that can control all IR-based electronic home appliances is challenging. Furthermore, each home gadget, such as a home radio, air conditioner, and television, has various functions, making it difficult to control.

Chapter 2.2: Design of Smart Universal Remote using Mobile for Home Automation

Next, (Roy J and Roy J.K 2014) researched Design of Smart Universal Remote Using Mobile for Home Automation with two goals in mind: turning the smartphone into a true universal remote that can learn new devices and IR protocols, and turning the smartphone into a true universal remote that can learn new devices and IR protocols. They found out no or bad self-learning element in many projects they hear about from other people's work. They suggested methodologies are vulnerable to or fixed to a certain collection of IR protocols. The key goals of this paper are then listed in the table below.

Main Objective	Explanation
Universal Remote Control (URC) able to learn new device faster.	Different remotes have different bit patterns. Power on button bit pattern will be unlike due to different manufacturers such as Toshiba and Sony. Thus, the URC have the goal of learning new bit pattern easily, and it can be used and learn with any new devices one brings in the home.
Self-adaptive to underlying IR protocol	The format used in various IR remote is different such as there are many available IR formats like Panasonic command, Sony Code SIRCS, Sharp Data format etc. Even bit logic for these protocols are unlike, a true URC must not de sensitive and must be able to control a TV from different manufactures following any protocol. Also, URC must self-adaptive to that change if someday a new protocol comes up.

Table 2.1.1 Main objective of on Design of Smart Universal Remote using Mobile for Home Automation

The instructional module part and the operational module part are the two functional parts of this article. To begin, URC had to learn a new device and its new IR bit patterns as part of the teaching module. When we want to implement a new TV remote feature, we need to teach URC to learn the IR bit code of the corresponding function and send it to the IR receive circuit. This research is close to the previous one. The IR receiver circuit can receive an IR signal from the new or any remote, decode it into a bit pattern, and send it as a microphone input for recording in .war files with PCM. In short, WAV

is an audio file format for solid audio bitstreams, while PCM encoding is a technique for digitally representing sampled analogue signals. Second, the operating module is played back on the cell phone via the headphone audio out when a user needs to monitor some pre-learnt unit from URC for that particular purpose. 38KHz frequency is being used in order to avoid all those sources remote control pulsates its IR signal since there may have hundreds of tiny IR sources in the room.

They used a Microcontroller MSP430G2231 to modulate the IR Bit patterns with a carrier frequency of 38KHz before sending them out to the IR transmitter in this mission. The microcontroller then takes the demodulated IR waves as feedback and produces an IR wave with a modulation of 38KHz. Finally, the intensified modulated IR wave is sent out via the IR transmitter. When the IR Receiver decodes the IR signal sent from the IR transmitter, the computer can perform the procedure if the pre-stored bit pattern fits the bit pattern. The concept of how the functional module and working module operate is seen in Figure 2.2, which is abstracted. (Roy J and Roy J.K, 2014, p77)

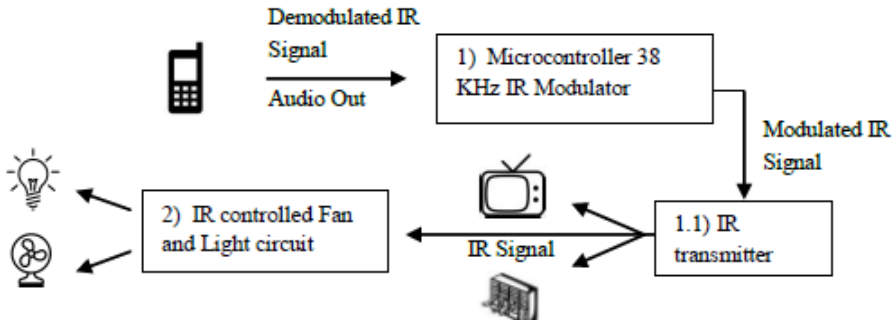


Figure 2.2 Idea on functional module and operating module.

Chapter 2.3 Critical Remarks of previous works

The research article has use of CC3200 and MSP430G2231 to control IR device as a central device, then this project is also needed to design a Smart Universal Remote to control the IR device by phone for home automation. We proposed the same method of articles previously. The IR signals will be recorded by the microcontroller which connects to the hardware circuit.

We can try to use Raspberry Pi which also a high-performance microcontroller to use this project. This is due to Arduino and Raspberry Pi is a high-performance microcontroller because we have more familiar with RPi. Also, Python GUI is used in this program with RPi have a program called Geany to run the Python code. However, we will use the LIRC library (Linux Infrared Remote Control) to do this project since it a powerful library to record the IR signal

Chapter 3: System Design

Figure 3.1 A flowchart of the System Design

The flowchart above shows there are approximate five step to complete of this project. The explanation will introduce detailly on Chapter 5.

3.1 Top-Down System Diagram

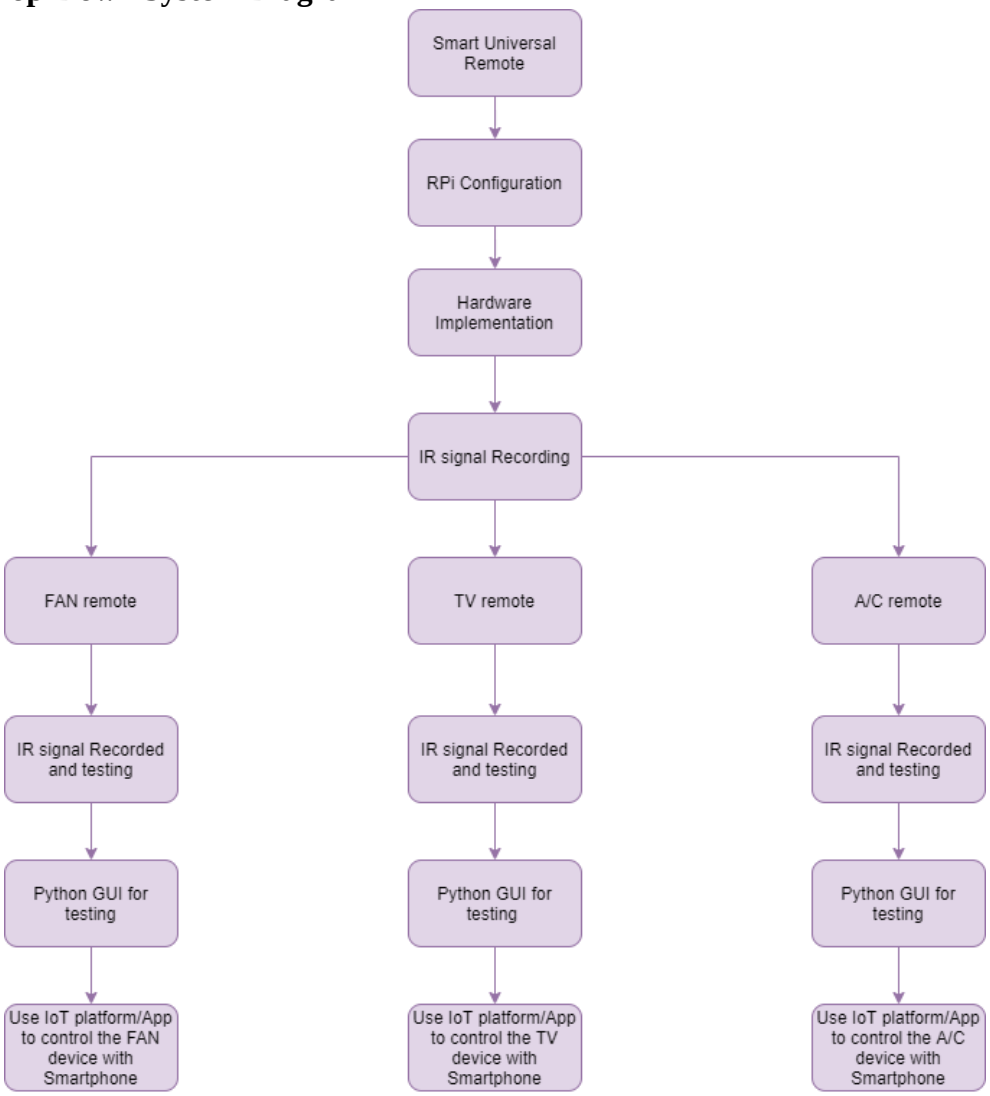


Figure 3.2 Top-Down System Diagram

As Figure 3.2 shown above, this is a top-down system diagram of the Smart Universal Remote Control. From the RPi configuration to IR signal Recording, it will be introduced in Chapter 5. Thus, there are 3 IR remote parts for the IR signal Recording: Fan, TV, and A/C, respectively. Each device needs to undergo IR signal testing after

using Python GUI to stimulate the remote-control interface. After doing the Python GUI testing and familiar control, we implemented an IoT platform to control each device with a smartphone from the IoT dashboard.

3.2 Flowchart of the Fan Control

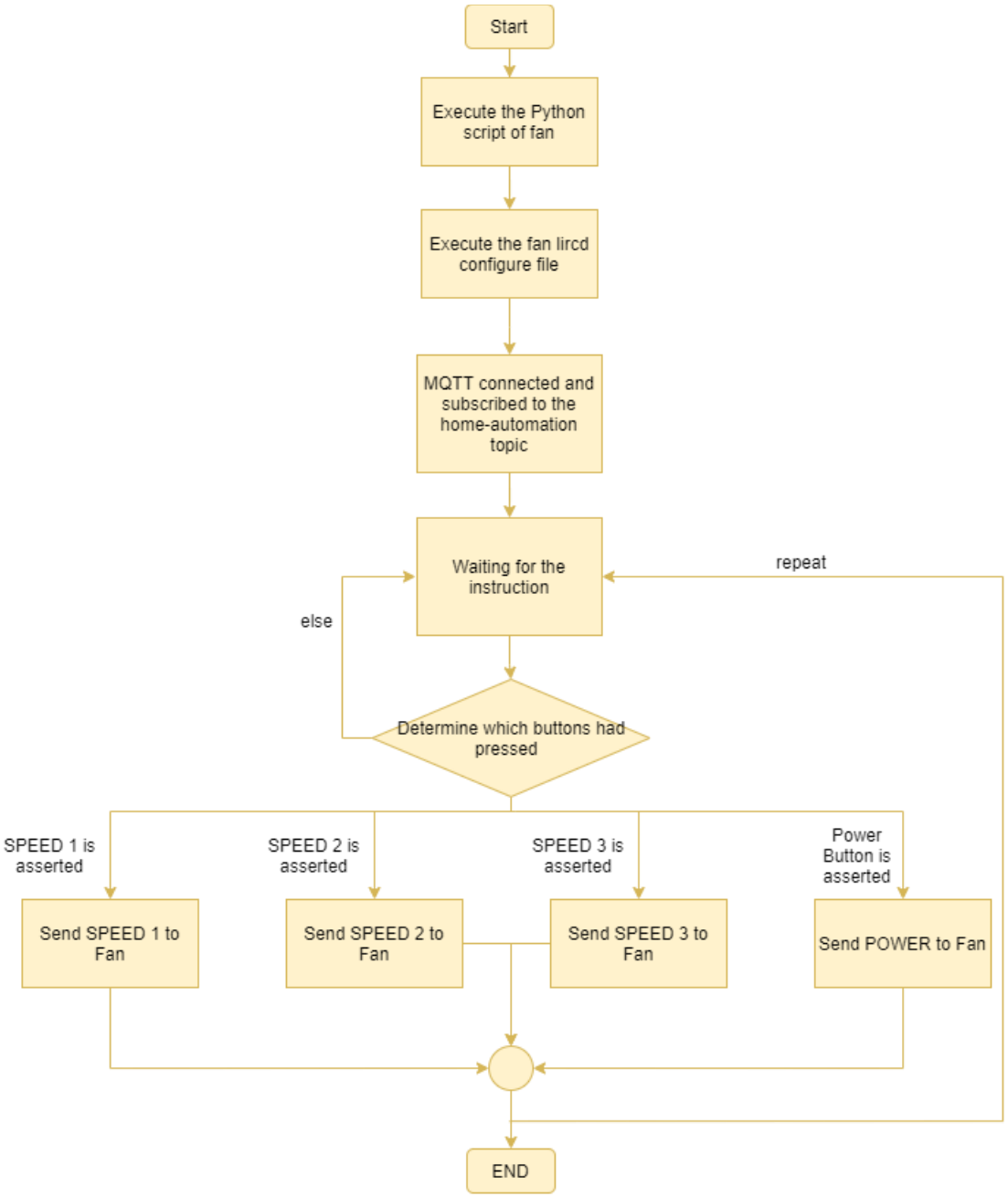


Figure 3.3 Flowchart of the Fan Control

Figure 3.3 has shown the flowchart of the Fan Control. After the Python script program has run and the LIRC remote has configured as a fan remote. Then, the MQTT topic also connected and subscribed. When one of the IoT platform buttons have pressed, the data will be published, and then the SPEED 1 command will execute. Lastly, the fan

device will receive the signal and do the operation based on the signal sent. For example, the prototype sent the POWER signal from the Universal Remote Control (URC) to the fan device. The fan will stop rotating until it stops. This operation can be repeated until the program exit.

3.3 Flowchart of the TV Control

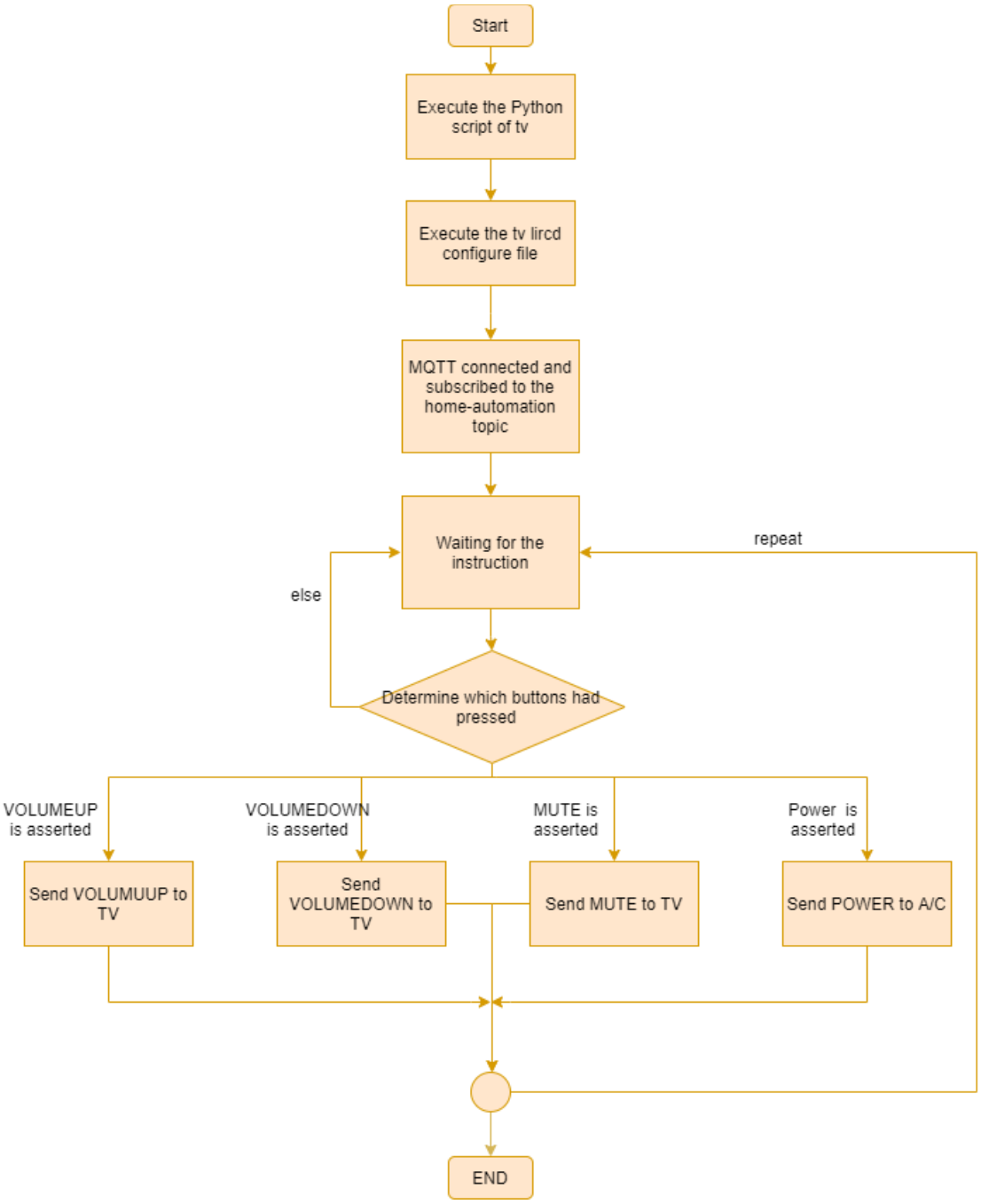


Figure 3.4 Flowchart of the TV Control

Figure 3.4 has shown the flowchart of the TV Control. After the Python script program has run and the LIRC remote has configured as a TV remote. Then, the MQTT topic also connected and subscribed. When one of the buttons has pressed the IoT platform,

the data will be published, and then the VOLUMEUP command will execute. Lastly, the TV device will receive the signal and do the operation based on the signal sent. For example, Universal Remote Control sent the POWER signal to the TV device. The prototype will turn the TV off. This operation can be repeated until the program exit.

3.4 Flowchart of the A/C Control

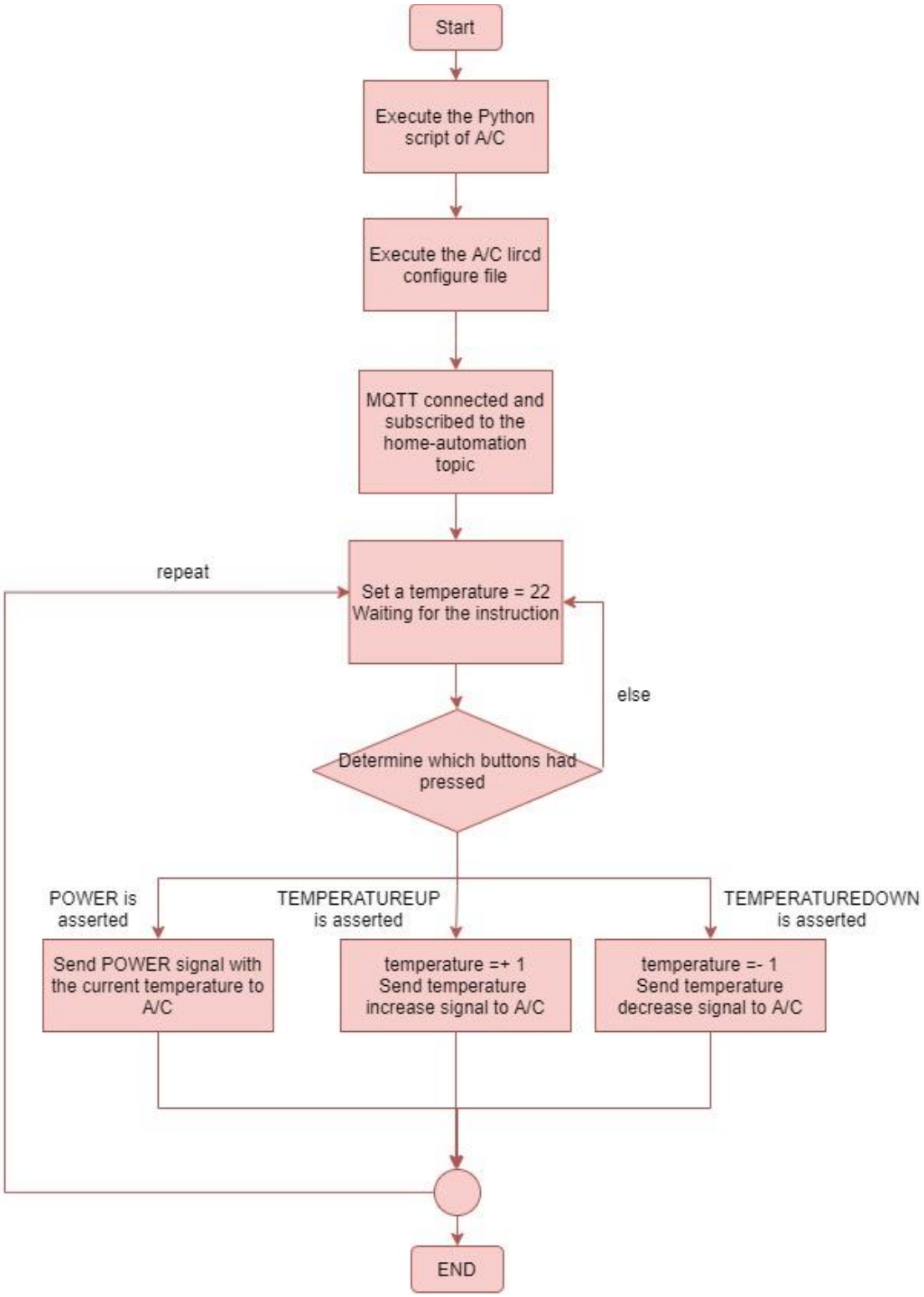


Figure 3.5 Flowchart of the A/C Control

Figure 3.5 has shown the flowchart of the A/C Control. After the Python script program has run and the LIRC remote has configured as an A/C remote. Then, the MQTT topic also connected and subscribed. When one of the buttons has pressed from the IoT platform, the IoT platform will publish the data, and the POWER command will execute. However, the A/C Control is slightly different to Fan and TV control. The temperature needs to declare so that the program able to know what the temperature is currently. Unlike TV, the volume level can observe from the screen but not in A/C. Lastly, the A/C device will receive the signal and do the operation based on the signal sent. For example, Universal Remote Control sent the TEMPERATUREUP signal to the A/C device. The A/C will increase by 1°C for its temperature. This operation can be repeated until the program exit.

Chapter 4 Methodology, Tools, Requirement and Specifications

4.1 Methodology

As a Smart Universal Remote Using Mobile for Home Automation, the most important objectives are to set up the hardware implementation of the circuit with RPi and software of the RPi. Secondly, the most time-consuming process has used the circuit to decode IR signals from Remote Controls for home appliances. Although the IR remote control can easily send the IR signal to the specific home appliances, the most vital part is to decode a correct signal by the circuit. After the completed decode of the IR signals, the IR signals will save into lircd.config files to communicate with RPi between those decoded IR home appliances. The diagram shows below is a general idea to understand through the project.

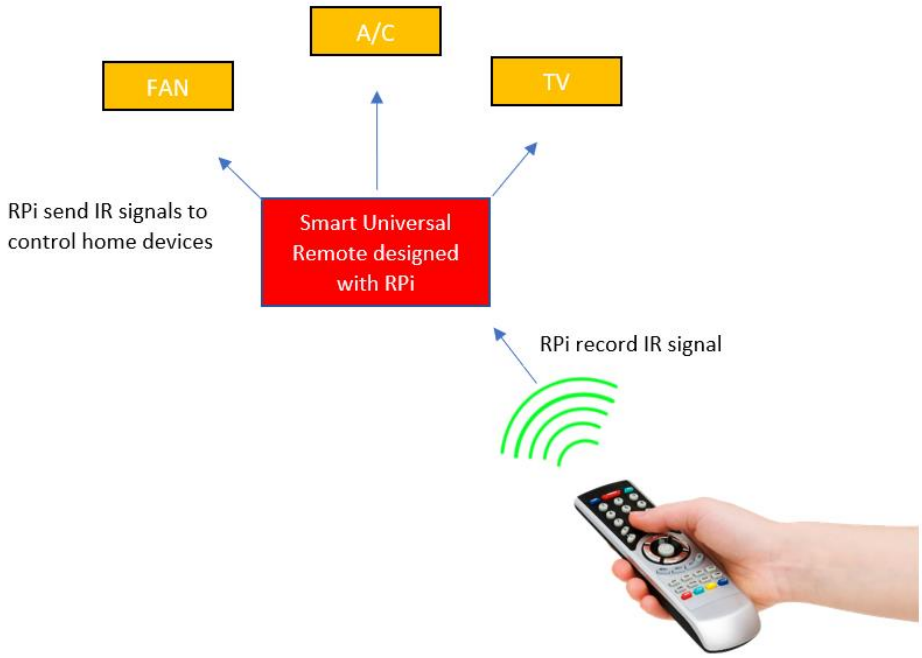


Figure 4.1 A general idea of Smart Universal Remote

After decoding and recording the IR signals of those home devices such as fan, TV and A/V. We tried to use Ubidots, which is an IoT platform to control those home devices. If those home devices can communicate by smartphone via Ubidots Cloud, this project has completed the project objective. As Figure 4.2 below is used to introduce smartphone communication between those home devices while RPi acted as the central part.

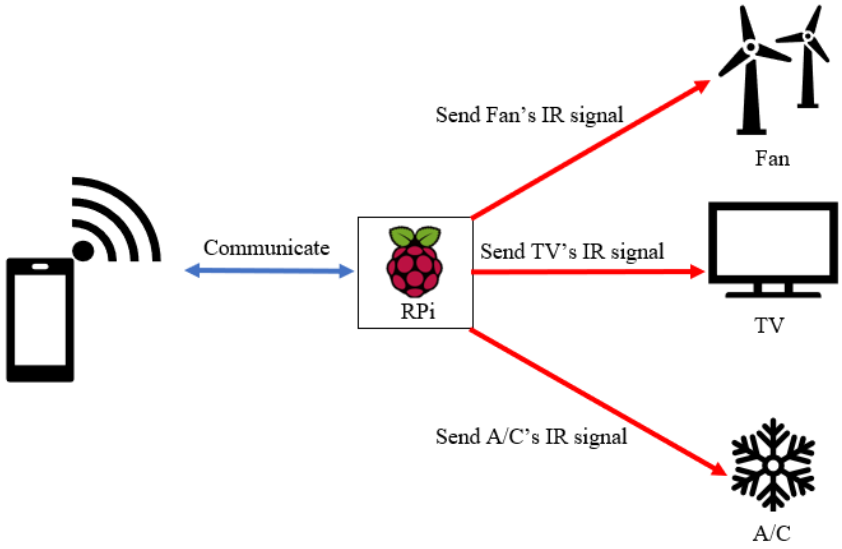


Figure 4.2 Communicate between Smartphone and Home Devices

4.2 Tool to use

4.2.1 Hardware

- Raspberry Pi 3 Model B+

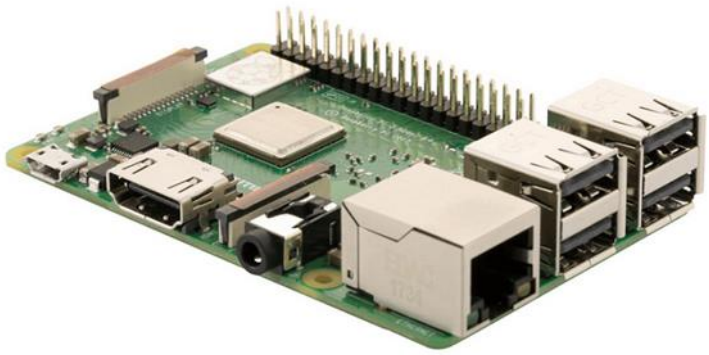


Figure 4.3 RPi 3 Model B+

Raspberry Pi 3 B+ is used in this project as central part of the design. An ARM based single-board computer which is running on Linux Operating System. It can be used to surf the internet and programming. The general specification of RPi are including boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, PoE capability via a separate PoE HAT and Micro SD format card support. (raspberrpi, n.d. p.2)

- IR remote controls
    - Fan remote (Fanco DF-100),
    - TV remote (LG AKB74915311)
    - AC remote (YORK ECGS01).
- IR Receiver Diode VS1838B
- IR LEDs
- Resistor 0.2W
- NPN Transistor 2N333
- Female jumper Wire
- Desktop/Laptop
- Smartphone
- Breadboard

4.2.1 Software

- Python 3 (Geany)

Python is an easy-to-learn programming language. There are some python strengths which are portable on various platform, increase productivity since no compilation step, object-oriented, and GUI programming. Raspberry Pi allow the user to code in Python. The version we use in this project is Python 3. Python has a simple syntax, and it is had increased the readability which similar to the English language compared to other programming languages.

- LIRC libraries

LIRC is an abbreviation of Linux Infrared Remote Control. According to (Bartelmus 2016), 'A package that allows user to decode and send infra-red signals of many commonly used remote controls. Recent Linux kernels make it possible to use some IR remote controls as regular input devices. Using LIRC on Raspberry Pi is quite popular these days. However, LIRC offers more flexibility and functionality and is still the right tool in a lot of scenarios. The most important part of LIRC is the lircd daemon which decodes IR signals received by the device drivers and provides the information on a socket. It also accepts commands for IR signals to be sent if the hardware supports this. The user space applications allow you to control your computer with your remote control.'

- VNC Viewer

  According to Raspberry Pi (n.d.), VNC Connect is included with RPi OS. It consists of both VNC Server which able the remote access the RPi with a laptop. To connect with the laptop, the laptop required to download the VNC Viewer then prompt in the IP address to process the remote access.

Figure 4.4 VNC logo

- Ubidots

  According to (Teoh,2019), Ubidots is an IoT platform specialised in connected hardware and firmware solutions to control remotely, monitor, and automate processes. It provides powerful real-time dashboards that can be used to control devices and analyse data. Ubidots API supported sending and retrieving data from RPi using HTTP, MQTT, TCP, UDP etc. Hence, MQTT is used as communication protocols in this project.

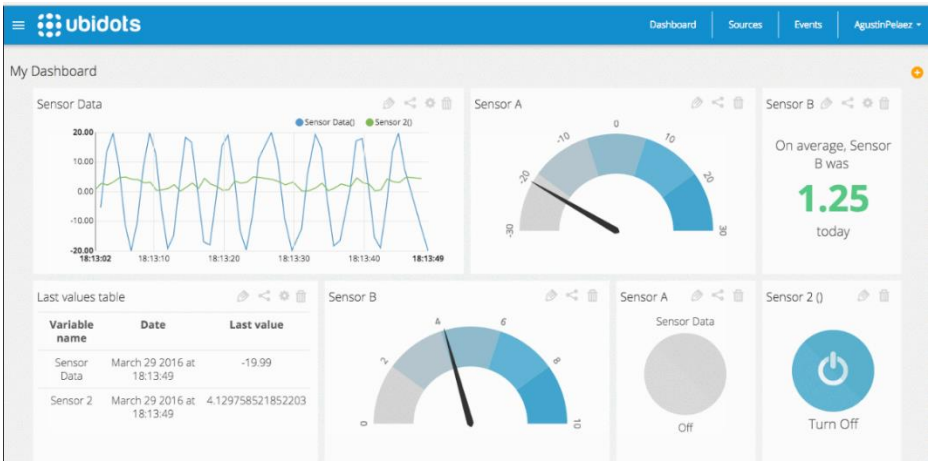


Figure 4.5 An example of Ubidots dashboard

- MQTT

  According to (Sepúlveda 2021), MQTT is beneficial for delivering data to your computers. It is designed for machine-to-machine communication between two devices, which means we can control the A/C from smartphone. Likewise, A/C is telling us what the temperature is setting currently. The computer will "listen" to the cloud using MQTT and only be alerted when the variable changes.

  Then, MQTT is a broker of publishing and subscribe to message transport protocol. This broker can communicate to those devices that have the same topics. For example, as in the diagram below, smartphone is subscribing to the Home Topic to check the light bulb condition whether turn on or off, while the watering sensor is used to check the soil moisture if it indicates watered means that the soil still has watered.

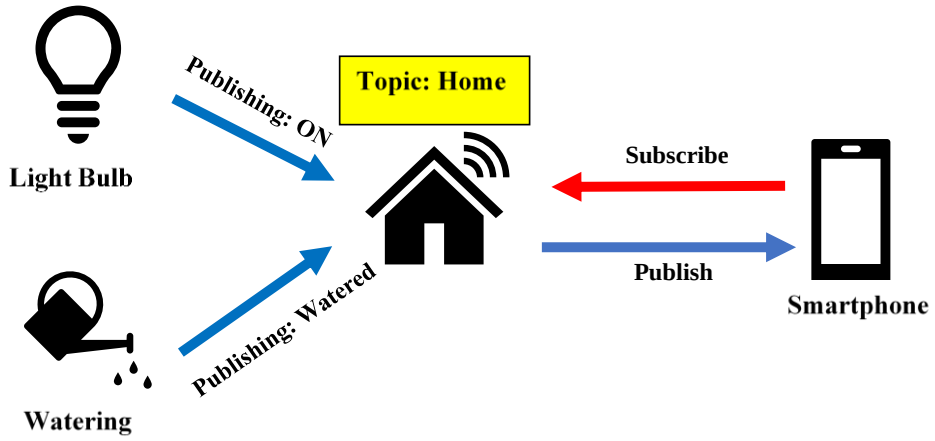


Figure 4.6 A communication of the device between one topic

The communication between the computer and the cloud is kept open in this manner, but data only moves when it is needed, saving battery life, network bandwidth, and increasing real-time performance.

- Telegram

  As we all know, Telegram is well-known freeware of instant messaging software and service to the user which allows communication between their users. However, in this project, we can use Telegram to control our RPi and control our home device. This is because Telegram has provided a Bot API to allow the user to create their bot themselves. Thus, BotFather is used to create a bot in Telegram, allowing the user to control by using their provided token. According to (Faris 2017), he has introduced a method to use a Telegram bot with RPi. Therefore, we can set up a simple bot to communicate with RPi while running a Python script.

Figure 4.7 Telegram Bot

4.3 Requirements

Project Requirement

	Descriptions
Functionally	This system is designed for home use or small area use. It can be used in a large area but needs to move the infrared LED by pointing at the IR devices to control. This can be used a smartphone to control remotely. That means it is convenient for the user to control the IR home devices with their smartphone.
User Interface	This system is designed in a well-looking and well-understand interface. The interface, even in window-view or phone-view, is well-looking. Thus, users can easily find out which button controls which device and what the button does.
Performance	Accuracy • The accuracy in this program would be ten out of ten unless the IR LEDs are not pointing at IR detector of the home devices. Timing • The timing is aimed to send the IR signals within 2 to 3 second to make a fast respond.
Cost	Approximately RM240 for the RPi Total cost of this project is RM280 approximately.
Physical Size and Weight	The size of RPi is 67×56×11.5 mm, so it is light and easy to move.
Power consumption	15W

Table 4.3.1 Project Requirement

Minimum Requirement of PC/laptop

Description	Minimum Requirements
Processor	Intel i5 or higher with frequency of 3GHz
Operating System	32-bits Window 10
Resolution	1920 x 1080
RAM	8GB or higher
Storage	256GB or more

Table 4.3.2 A minimum requirement of the PC/laptop to process the project

4.4 Specification

4.4.1 Verification Plan

Test No.	Test Case	Expected Results	Results
1	VNC Viewer Remote Access Configuration	Able to remote access with RPi	Pass
2	IR Receiver test	IR Receiver able to detect IR remote controls signal by using the command of **mode2 -d /dev/lirc1**	Pass
3	IR LEDs test	IR LEDs can blink by watching the phone camera when executing python program code.	Pass
4	LIRCD status check test	To check the availabilities of LIRC configuration files of remote and errors.	Fan: Pass TV: Pass A/C: Pass
5	IR Remote recording test	Repeating the recording process once the remote key can be decoded by LIRC as a configuration file (lircd.conf)	Fan: Pass TV: Pass A/C: Pass
6	IR Controlling Test with RPi	To test the decoded lircd.conf with the home device	Fan: Pass TV: Pass A/C: Pass
7	IR Controlling Test with RPi GUI	To build a Python GUI program with Tkinter that able to test recorded lircd.conf to test with home devices.	Fan: Pass TV: Pass A/C: Pass
8	IR Controlling Test on Ubidots with smartphone	To control the home devices with mobile phone on Ubidots Cloud remotely	Fan: Pass TV: Pass A/C: Pass

9	IR Controlling Test on Telegram with smartphone	To control the home devices with mobile phone on Telegram remotely	Fan: Pass
			TV: Pass
			A/C: Pass

Table 4.4.1 A table of Verification Plan

Chapter 5: Implementation and Testing

5.1 Implementation

5.1.1 RPi Configuration Setup

First and foremost, connect the RPi with the monitor by using an HDMI cable and start the initial simple setup. After setting up the time zone and enable the Wi-Fi connection, run the command **sudo raspi-config**, go to **Interfacing Options > VNC** and select **Yes**. Navigate the VNC Server, which is located at the right-hand side menu bar and copy the IP address. After that, download the VNC Viewer on the laptop and finish those installation instructions. Lastly, open the VNC Viewer application and paste the IP address to connect.

Note: The username and password would be **pi** and **raspberry** respectively by default.

5.1.2 Hardware Implementation

Next, we need to build a simple and easy hardware circuit with RPi. According to (Stanton 2017), the website has introduced the way of designing a Universal Remote with RPi by using LIRC. We followed and modified the circuit that they had mentioned, and this is a hardware circuit as in the diagram below we built.

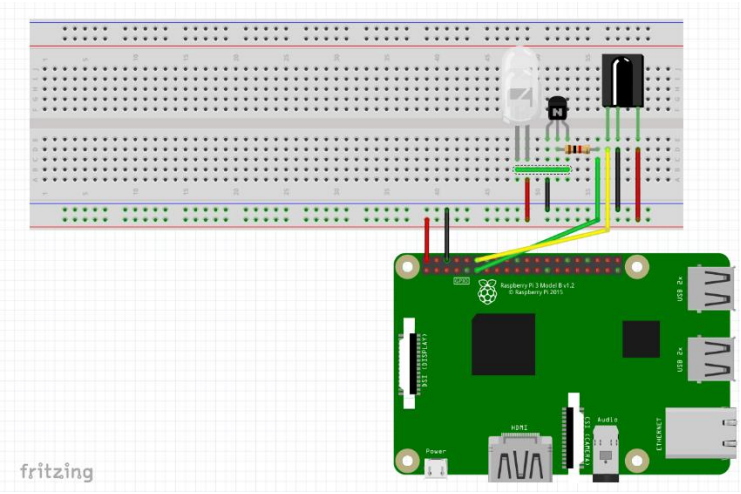


Figure 5.1 A hardware drawing of Smart Universal Remote

There are two parts of the hardware implementation as an IR receiver part and IR transmitter parts. Firstly, IR Receiver Diode VS1838B is used to record the IR signals. We connected the PIN OUT to GPIO Pin18 from RPi, PIN GND connected to ground, and VCC pin connected to 3.3V pin from RPi. (Note: Please refer to the appendix behind)

Secondly, for the IR transmitter part, we installed three IR LEDs on the breadboard. The reason we need to install more IR LEDs used to strengthen IR signal transmission between IR home devices. Then, NPN Transistor 2N2222 is used to help drive IR LED by amplifying the current output. Lastly, a resistor is used to connect between the RPi GPIO Pin 18 and 2N2222 transistor. (Note: Please refer to the appendix behind)

5.1.3 IR signals recording and sending with terminal

Referring to Prasanth (n.d.), and Rich (n.d.), the following instructions below are the methods and procedures to record and control the home devices with RPi

1. After finished the RPi start-up setup, open terminal window to update the RPi and install LIRC.
   $ sudo apt-get update
   $ sudo apt-get install lirc
2. Type **$ sudo nano /etc/lirc/lirc_options.conf** and edit the file of /etc/lirc/lirc_options.conf as the following code below.
   .
   .
   driver = default
   device = /dev/lirc0
   device = /dev/lirc1
   .
   .
   NOTE: According to Leon,2020, device **/dev/lirc1** is used by the IR receiver and device **/dev/lirc0** is the IR LEDs. Initially, **/dev/lirc1** is used to scan and record a remote control.
3. Uncomment the following lines in /boot/confi.txt to allow RPi 'listening' on GPIO Pin 18 and send the signal on GPIO Pin 17
   dtoverlay=gpio-ir,gpio_pin=18
   dtoverlay=gpio-ir-tx,gpio_pin=17
   by typing **$ sudo nano /boot/config.txt**

4. Restart LIRC by doing and check whether have errors
   $ sudo /etc/init.d/lircd stop
   $ sudo /etc/init.d/lircd start
   $ sudo /etc/init.d/lircd status

5. Reboot the system by tying
   $ reboot
6. Check the LIRC driver and IR Receiver by typing the command.

$ sudo /etc/init.d/lircd stop

$ mode2 -d /dev/lirc1

7. Record only one key button of the IR remote control and follow the instructions until lircd.conf file will be generated.

   $ sudo irrecord -f -d /dev/lirc0 ~/lircd.conf

   Note: The reason for the recommended record one key first due to it is depending on luck and environment (e.g. lights) sometimes. It might try several times until it successfully is written config file. After decoded one key then try another key button until it is done, copy the second key button raw code into the first key button configure file.

8. Backup the original lircd.conf

   $ sudo mv /etc/lirc/lircd.conf /etc/lirc/lircd_original.conf

   $ sudo cp ~/lircd.conf /etc/lirc/lircd.conf

   $ sudo /etc/init.d/lircd restart

   Note: The name of the lircd.conf file will be different, thus before executing the command, the name of the target file and destination files need to change.

9. Check the data of signals that recorded in the configure file. E.g:

   $ irsend SEND_ONCE <remote_name> KEY_POWER

According to (Aufranc 2017), he introduced how to decode the IR signals of A/C is slightly different but similar to the instruction above the way of fan and TV. We are still using step 6 above, which is **$ mode2 -m -d /dev/lirc1**. When the IR signal from A/C remote is sent to the IR receiver from RPi, we will observe many raw code lines. The raw code copy and paste into the lircd.config file which able to control the A/C when the lircd.config file is executing.

Note: Ignore the first “large number” from the raw code; otherwise, the lircd.config file would not work.

5.1.4 Python GUI

After completed recording and testing on the terminal window, we used Geany to program a GUI program to control with Python instead of terminal command. Tkinter library is the Python interface to the Tk GUI toolkit shipped with Python and used to create GUI applications. It is easy to use with Python as we only import the Tkinter module. Also, a button widget is used to simulate and modify as button keys of the IR remote controls. Next, import the os module and time module when using Python code to send the IR signal using LIRC and give some second to suspend the execution. For example, as the code below

```
os.system('sudo /etc/init.d/lircd restart')
time.sleep(1)
os.system('irsend SEND_ONCE <remote_name> <power command>')
time.sleep(1)
```

5.1.5 Phone Controls

5.1.5.1 Ubidots Implementation

After testing the code, we must register an educational or personal use of Ubidots account at https://ubidots.com/stem/. This is a free account to register and test with our project. Ubidots is an IoT platform that able to connect hardware and firmware solutions to control remotely.

First, create devices and variables. According to Maria (2021), a Device in Ubidots is a virtual representation of a data source. For example, Home Automation is a device name we used in this project. As the diagrams below have shown, the way to create a device by **clicking the "+" icon** and then **chose the blank device** to proceed.

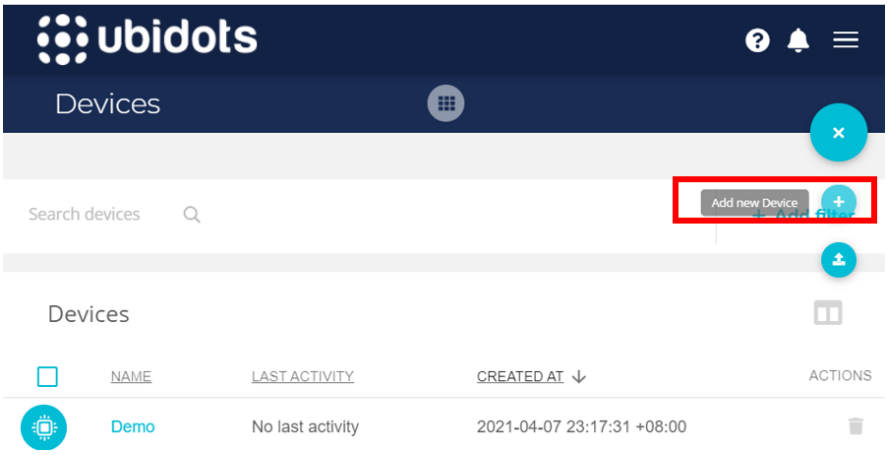


Figure 5.2 Add a new Device (RED circle)

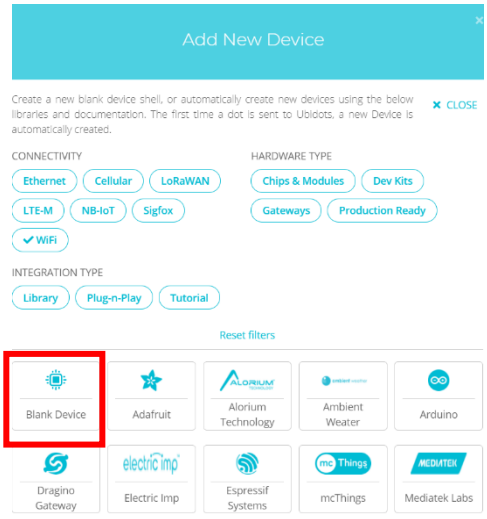


Figure 5.3 Start from a Blank Device (RED circle)

Variable in Ubidots is created when a device is created and receiving the data from our hardware data-source, the data can be represented ad raw or synthetic. we have always used raw data to represent our remote button. The diagram has shown below the step to create a variable by clicking the "**add variable**" and then "**raw**".

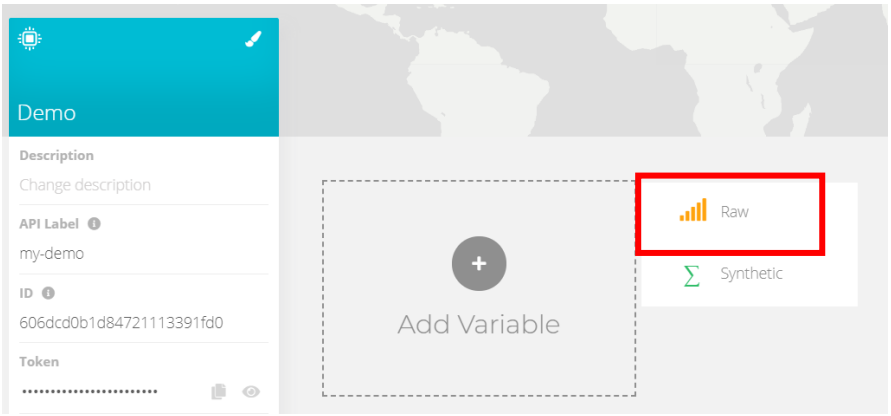


Figure 5.4 Choose a Raw Variable

After that, to insert or add a switch variable in the dashboard, which is shown the diagram below with Red Circle, go to:

Data > Dashboards > "+" icon > Choose Switch > Click and select the variables that needed > "✓" icon

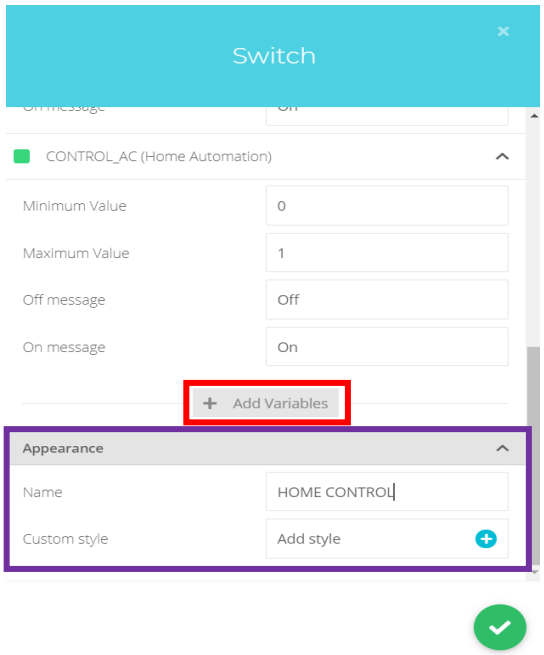


Figure 5.5 Add Variable (RED circle)

Additional: Ubidots provide a function to custom their dashboards and widgets by clicking **Add styles from Appearances** which has shown in the diagram above with a

Purple circle**.** For example, change the colour of the header and font, change the size and weight of the font, etc. This is depending on the creativity of the user to make a well-looking dashboard and widgets.

Install the Ubidots Python API Client library by typing the statement below in the terminal.

$ sudo pip3 install ubidots == 1.6.6

Next, the token is a must because it is a unique key that is used to control the Python program with Ubidots. Thus, go to the **API credentials > copy the default token.** Insert the code as below in the Python script to connect Ubidots.

```
from ubidots import ApiClient
api = ApiClient(token='your_token') #copy and paste the token
```

Also, we need to assign variable as a button widget in Ubidots,

```
switch_button = api.get_variable('variable_id') #copy and paste the variable ID from the variable
```

If we need to get the value of the variable to **swith_button,** get a single item in the get_values method as below,

```
s0 = switch_button.get.values(1)
```

If the if-else condition of the value is equal to 1(high-asserted of the switch button), execute the remaining code.

```
if s0[0]['value'] == 1:
    #excute the remaining code statement.
    …
```

5.1.5.2 MQTT Implementation

Install Mosquitto and its command line clients by typing the statement in the terminal below,

$ sudo apt-get install mosquitto mosquitto-clients

$ sudo pip3 install paho-mqtt

5.1.5.3 Publish message with Python

To explain how to use publish messages, the diagram is shown below that of the sample code of the project. **Note:** This is just for explanation of the code.

```
1  import paho.mqtt.client as mqtt
2  BROKER_ENDPOINT = "things.ubidots.com"
3  MQTT_USERNAME = "YOUR_TOKEN"  # Put here your TOKEN
4  TOPIC = "" #name your topic
5
6  def on_connect(mqttc, obj, flags, rc):
7      print("[INFO] Connected!")
8
9  def on_publish(mqttc, obj, mid):
10     print("[INFO] Published!")
11
12 # Setup MQTT client
13 mqttc = mqtt.Client()
14 mqttc.on_connect = on_connect
15 mqttc.on_publish = on_publish
16
17 mqttc.connect(BROKER_ENDPOINT, PORT, 60)
18 while(1):
19
20         mqttc.publish(topic,0)
```

Figure 5.6 Publish message sample code

- First, from line 1 to line 4, importing the paho.mqtt.client as mqtt, and define a broker to **things.ubidots.com** (educational account). Next, assign the Ubidots token as a username of MQTT Credentials and name the topic.
- Secondly, for line 13 and line 18, we are telling the MQTT client to connect to the broker which is **things.ubidots.com**. If the client has connected to the broker, the console will indicate **[INFO] Connected!** (from line 7)
- Lastly, to publish the message, we write a line of code (line 20) to publish the message in the console.

5.1.5.4 Subscribe Topic with Python

To explain how to use to subscribe to topic, the diagram is shown below that of the extracted code of the project. Note: This is just for explanation of the code.

```
1  import paho.mqtt.client as mqtt
2  BROKER_ENDPOINT = "things.ubidots.com"
3  MQTT_USERNAME = "YOUR_TOKEN"  # Put here your TOKEN
4  TOPIC = ""#name your topic
5
6  def on_connect(mqttc, obj, flags, rc):
7      print("[INFO] Connected!")
8
9  def on_message(mqttc, obj, msg):
10     print("[INFO] value received: {}".format(float(msg.payload)))
11
12
13 def on_publish(mqttc, obj, mid):
14     print("[INFO] Published!")
15
16
17 def on_subscribe(mqttc, obj, mid, granted_qos):
18     print("[INFO] Subscribed!")
19
20 # Setup MQTT client
21 mqttc = mqtt.Client()
22 mqttc.on_message = on_message
23 mqttc.on_connect = on_connect
24 mqttc.on_publish = on_publish
25 mqttc.on_subscribe = on_subscribe
26
27 mqttc.loop_start()
28 mqttc.connect(BROKER_ENDPOINT, PORT, 60)
29
30 mqttc.subscribe(topic, 0)
31 mqttc.loop_stop()
```

Figure 5.7 Subscribe topic sample code

- Firstly, the code structure of publish a message and subscribing are similar for instance, from line 1 to line 4 is the same, as well as line 28 creates a client connection.
- Thus, the difference of the subscribing codes from line 22 to line 25, these lines of code are used to set up the MQTT client. When line 30 is executed, it will go to the on_subscribe callback function to print the **[INFO] Subscribe** statement to indicate the topic has been subscribed.

5.1.6 Telegram Implementation

5.1.6.1 BotFather

1. Install and Open Telegram app
2. Go **Contacts** > **Search** BotFather

Figure 5.8 BotFather

3. Type /start to start the BotFather

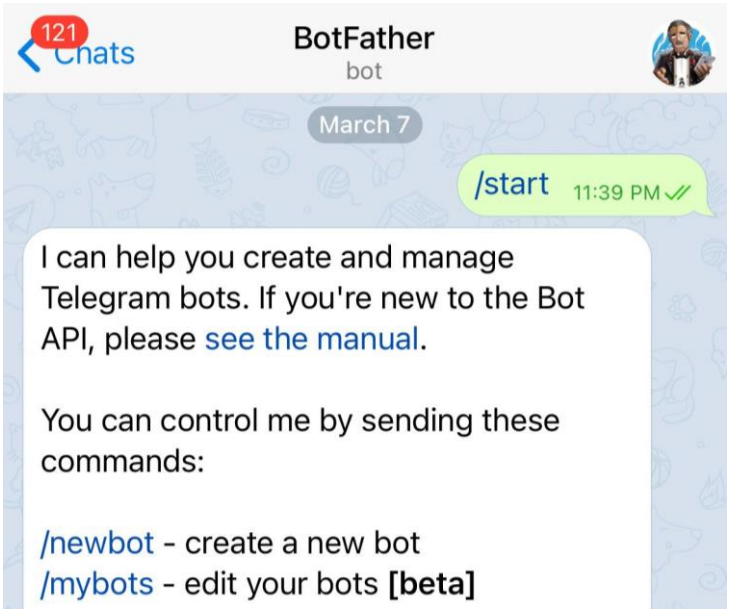


Figure 5.9 Start BotFather

5. Type **/newbot** to create a new bot by naming the bot and the username and obtained the token. **Note:** 3 bots have created for represent to FAN, TV, A/C respectively.

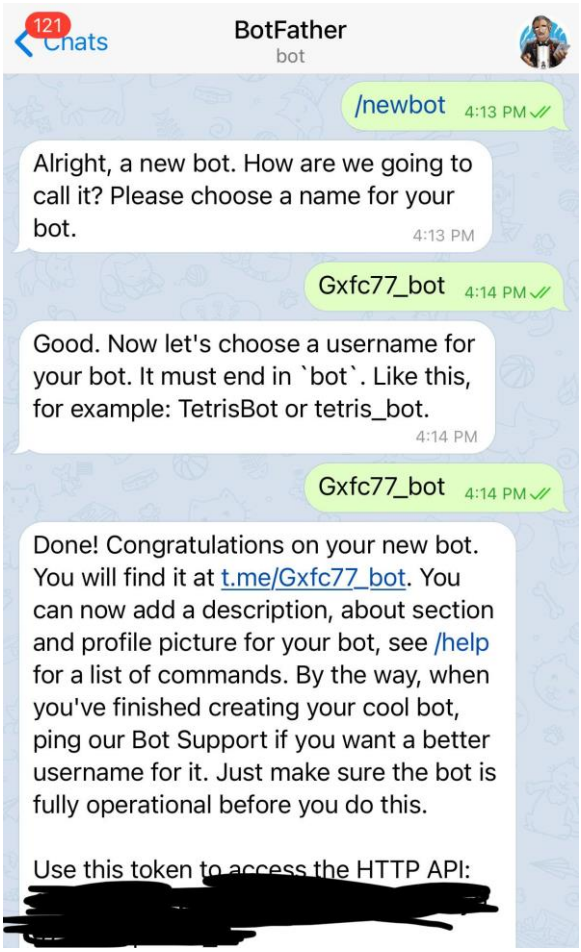


Figure 5.10 Create new bot and Obtain a Token

5.1.6.2 Telegram Bot with Raspberry Pi

1. Install telepot in RPi by typing the command in terminal.

   $ sudo pip install telepot

2. Import telepot in Python script

   import telepot

3. Copy and paste to assign the telepot token as bot

   bot = telepot.Bot('Your_Token')

4. Receiving the message form telegram in the handle function

   chat_id = msg['chat']['id']

5. Getting text from the user in the handle function

   command = msg['text']

6. Compare the command with the if-else condition statement, for example as below, if the user type ‘1’ to the bot, the RPi will receive and execute the command to send the instruction to fan devices.

   if command == ‘1’

   bot.sendMessage(chat_id,”SENDING SPEED 1”)

   …

   os.system(‘irsend SEND_ONCE fan KEY_1’) #send SPEED 1 signal

   …

7. The command shown as below will loop the handle function of the bot while the program is exit.

   bot.message_loop(handle)

5.2 Testing

1) VNC Viewer Remote Access Configuration

Navigate the VNC Server which is located at the right-hand side menu bar and copy the IP address.

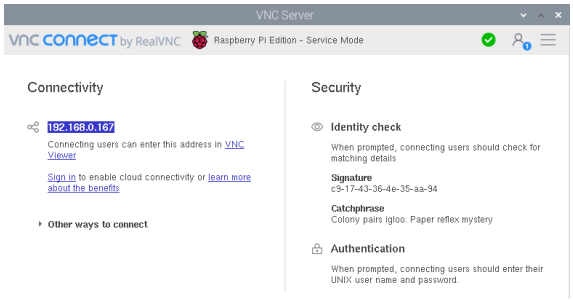


Figure 5.11 A VNC Server window that show the IP address

Open the VNC Viewer application and paste the IP address to connect. After login, RPi can get control by laptop.

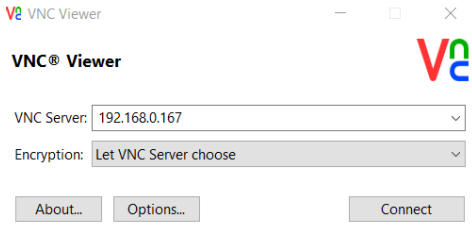


Figure 5.12 A login window of VNC Viewer by entering the IP address

2) Hardware Implementation

This a simple circuit to record and send the IR signal from the RPi that has connected with RPi GPIO pins.

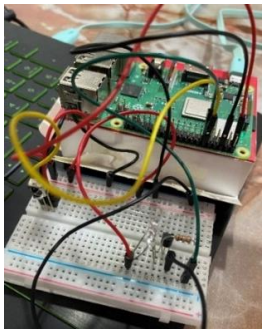


Figure 5.13 A hardware circuit which connected with RPi

3） IR Receiver test

After setting up the LIRC configure file, type command of **$ mode2 -d /dev/lirc1** to testing the IR receiver. The sample output is showing in the image below.

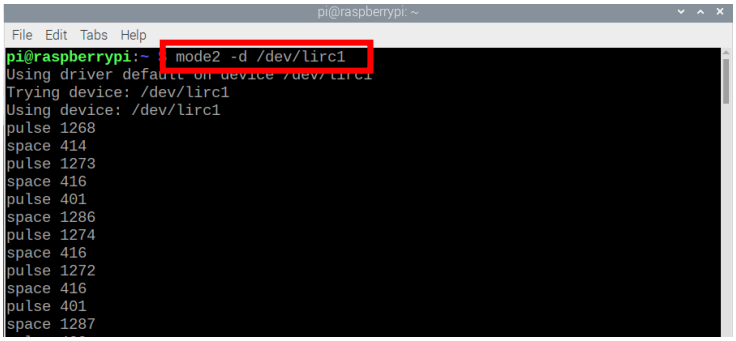


Figure 5.14 A terminal window with the command for testing the IR Receiver

4) IR LEDs test

Write a simple code to test IR LEDs whether it will be blinking. NOTE: Try to watch it by a phone camera because bare eyes cannot see IR ray.

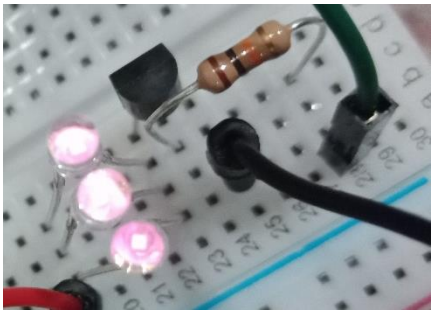


Figure 5.15 LEDs is blinking when execute the blinking program code

5) LIRCD status check test

The result shown as below, when execute the following command:

$ sudo mv /etc/lirc/lircd.conf /etc/lirc/lircd_original.conf

$ sudo cp ~/<filename>.lircd.conf /etc/lirc/lircd.conf

$ sudo /etc/init.d/lircd status

This is used to check the status of LIRC which lircd.config.file it are running.

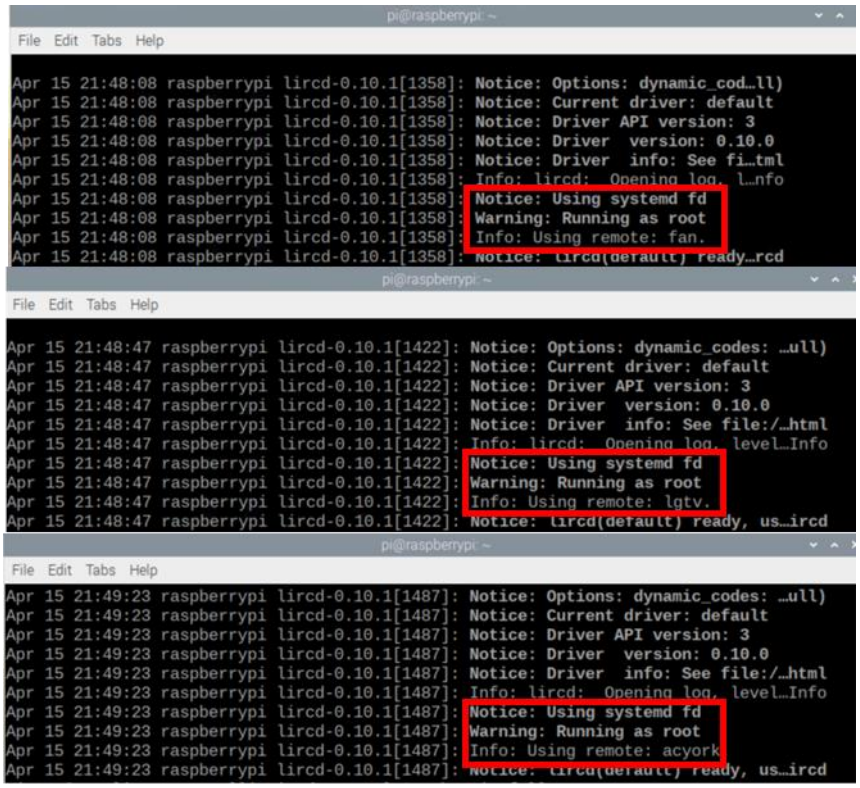


Figure 5.16 LIRCD status check for Fan, TV, A/C

6) IR Remote Recoding Test

These are the raw code results of recorded IR device remote lircd.config.file, by using the command below:

$ sudo irrecord -f -d /dev/lirc0 ~/lircd.conf

Note: Listed on sub chapter 5.1.3 at point 7.

fan.lircd.conf - Mousepad

File Edit Search View Document Help

```
begin remote

 name  fan
 flags RAW_CODES|CONST_LENGTH
 eps            30
 aeps          100

 gap          54035

     begin raw_codes

         name KEY_POWER
            1322     356    1305     384     460    1228
            1305     384    1307     382     459    1228
             461    1226     462    1225     488    1201
             462    1226    1303     385     460    7985
            1307     382    1304     385     461    1226
            1305     384    1304     384     460    1229
             460    1226     487    1202     462    1226
             462    1226    1306     383     461

         name KEY_3
            1243     437    1242     443     400    1302
            1234     441    1247     442     427    1260
             426    1292     399    1260     399    1291
            1271     417     425    1262     427    8031
            1268     411    1275     412     400    1296
            1236     444    1274     416     399    1292
             421    1262     426    1264     425    1266
            1240     442     401    1290     456
```

Figure 5.17 Fan lircd config file

lgtv_test.lircd.conf - Mousepad

File Edit Search View Document Help

```
begin remote

  name  lgtv
  flags RAW_CODES|CONST_LENGTH
  eps            30
  aeps          100

  gap          108088

      begin raw_codes

          name KEY_POWER
             8999    4521     501     618     503     619
              503    1751     493     617     503     618
              503     618     503     618     503     618
              504    1741     503    1747     495     618
              502    1740     503    1740     503    1745
              498    1747     496    1740     503     618
              504     618     502     618     505    1737
              505     617     503     625     497     619
              502     618     502    1740     503    1740
              503    1739     504     618     507    1738
              503    1738     504    1739     503    1741
              501

          name KEY_MUTE
             8966    4526     491     617     504     620
              501    1738     514     621     490     617
              504     616     504     617     504     617
              504    1745     498    1739     502     623
              500    1736     504    1739     503    1739
              504    1738     503    1742     500    1741
              501     620     501     616     505    1738
```

Figure 5.18 TV lircd config file

Note: There is another method to record the raw code of the A/C while have mentioned on sub chapter 5.1.3 last paragraph. Remember to delete the first “large number” from the raw code if the value too large; otherwise, the lircd.config file would not work.

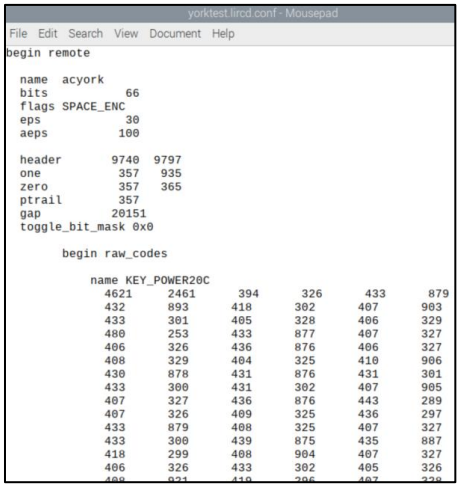


Figure 5.19 A/C lircd config file

As the Figure 5.20 shown below, this is command to record the IR signal from the A/C remote with at state of 22°C, AUTO fan speed, AUTO swing and COOLING mode.

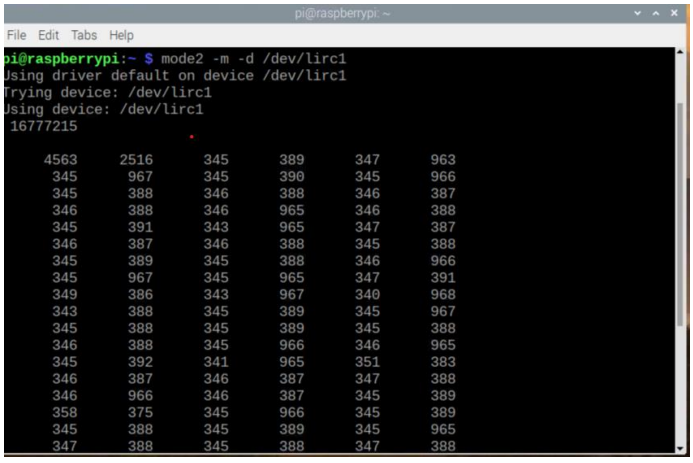


Figure 5.20 The command of mode2 -m -d /dev/lirc1

6.1) IR Remote recording test

There is an output when has executed the command of **$ sudo irrecord -f -d /dev/lirc0.** Do not be worry when saw the output like the diagram below. Lots of Patients is needed to record the IR signal until its success.

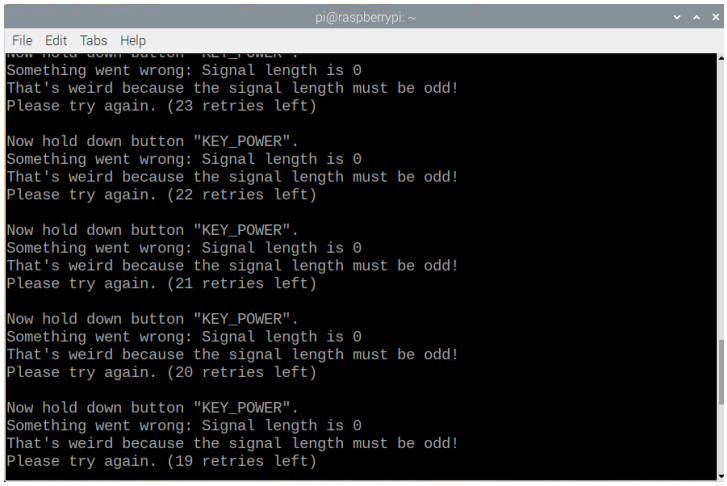


Figure 5.21 A terminal shows errors when recording the IR signals from remote control

7) IR Controlling Test with RPi

Now, test the POWER COMMAND of those devices. Type the following command in the terminal:

$ irsend SEND_ONCE <remote_name> KEY_POWER

8) IR Controlling Test with RPi GUI

As the diagrams have shown below, the GUI windows are named FAN REMOTE, A/C YORK REMOTE. The fan can be remote control pointed by the IR LED of the circuit when the button on GUI when pressed. Other GUI windows also able to send the IR signal based on the button has pressed.

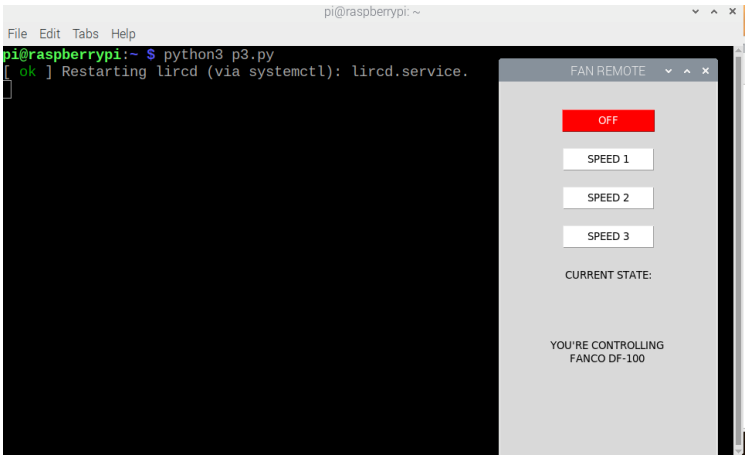


Figure 5.22 FAN REMOTE is working when running the Python program

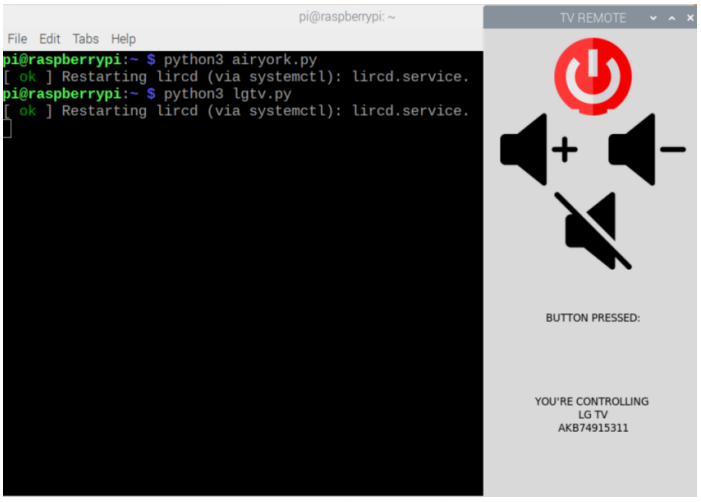


Figure 5.23 TV REMOTE is working when running the Python program

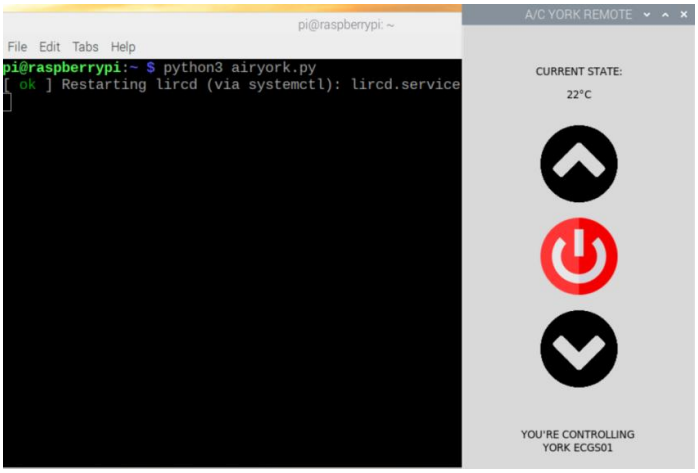


Figure 5.24 A/C REMOTE is working when running the Python program

9) IR Controlling Test on Ubidots with smartphone

As the diagram shown below, this is the dashboard from Ubidots. There are 4 four different colour parts which are FAN, TV, A/C and HOME CONTROL. HOME CONTROL is used to change which IR home device need to control.

NOTE: Login into Ubidots account with smartphone to control this dashboard.

Figure 5.25 Dashboard from Ubidots (Web-View)

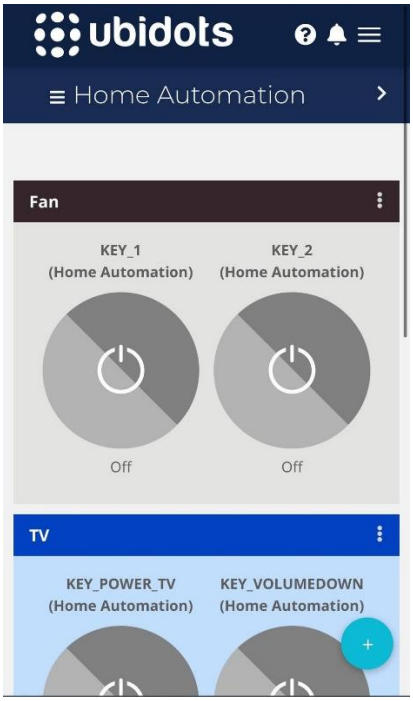


Figure 5.26 Dashboard from Ubidots (Phone View)

10) IR Controlling Test on Telegram with smartphone

As the diagrams have shown below, theses are the bots of the Telegram which are MyHomeFan, MyHomeTV and MyHomeAC. These have shown the way to control those IR home devices by using Telegram. For example, type '0' in the chat box, the prototype will send SPEED1 IR signal to the Fan device: type '1' will send the SPEED 1 command.

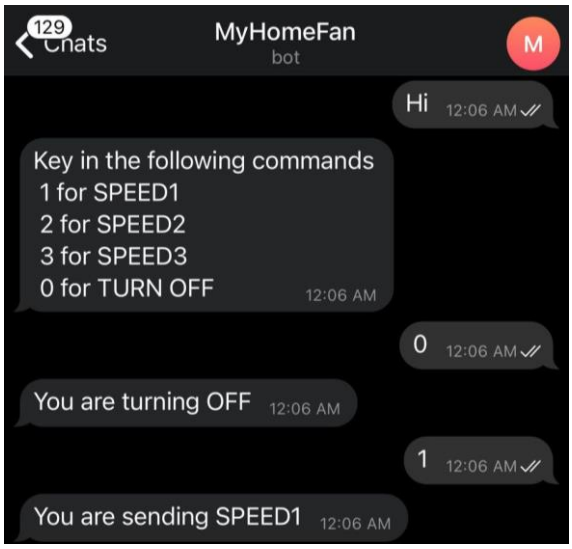


Figure 5.27 Telegram Bot of MyHomeFan

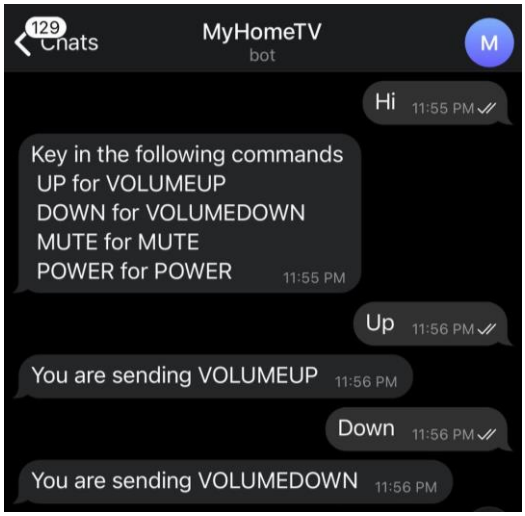


Figure 5.28 Telegram Bot of MyHomeTV

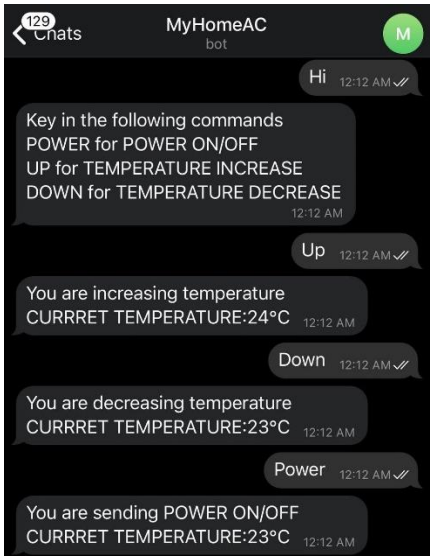


Figure 5.29 Telegram Bot of MyHomeAC

Chapter 6: Conclusion

6.1 Project Review and Discussion

From this IoT project, a smart universal remote has designed to control the IR home devices with smartphones by using Ubidots and Telegram. The last process is done in Project 1 which where the RPi configuration setup has done. It can connect with a laptop by using VNC Viewer to remote access. The LIRC package installation has been done successfully and can use for receiving and sending after setting up. The hardware circuit also runs steadily which IR Receiver can receive the IR signal from IR remote control. From the past FYP1, we only able to decode and record the IR signal from the fan remote control into a fan.lircd.config file. After recording successfully, the file can be executed by sending the raw codes of any fan remote control buttons to control the fan without any help from IR remote controls. Now, TV and A/C also recorded successfully into lircd.config file. Thus, the RPi can use this lircd.file to send the recorded button's raw codes signal to Fan, TV, and A/C. In this process, patience is vital to record, debug and test. This is a long path to lead to success.

After testing the IR signal using the command in the terminal window on RPi, the Tkinter library used for creating GUI applications to simulate and build a similar look to the virtual remote control to control the home device. In the past FYP 1, Fan GUI is the only one that has designed so far. However, TV and A/C has designed successfully to control with those IR home devices.

In FYP2, we tried to implement the control on the IoT platform, which is Ubidots. First, we also tried to implement the fan part as a trial. Hence, the fan part has a lesser IR signal to control than the TV and A/C. MQTT is used in this project with Ubidots so that it can push data to the RPi. For instance, when the python script implemented in Ubidots API is running, the device RPi will be able to listen to the data as the button was turned on from Ubidots. Then it will publish values to the broker, which means send a PUBLISH message to the subscribed topic. After the implementation of Ubidots on the fan have successful, then TV and A/C can start to process. This also a long path to go on debugging and testing.
Also, we edited the Udidots dashboard into eys-cathing so that user easy to classify which buttons are used to control which devices.

After that, we tried to implement Telegram to control the IR home devices. This is an additional method to control the IR home devices with the smartphone. By using the Telegram Bot API, we can create a bot to handle the chatbox. When we typed a command in the chatbox, the bot used it as a command, then executed the instruction to send the IR signal from RPi to the IR home device. We need to create three bots to handle the different home device. Otherwise, the Telegram would have an error to continue.

Furthermore, we can control each home device by using Ubidots and Telegram. We tried to combine those home device into one to make the universal remote control become “universal”. Without integrate the Python script, we need to change to run the Python script manually. We designed a Python script (urc.py) that acted as Universal Controller that can run another file. For example, when we are running urc.py and decided to control A/C, we can assert a button from Ubidots which from urc.py. Then, the program would go to execute another Python script which is ac.py to control by smartphone. Thus, this is more convenient to use.

6.2 Novelties and Contributions the project has achieved

Again, this project will benefit the reader who desired to build a Smart Universal Remote for IR home devices by themselves. The proposed solution has a successful design to control the fan, TV and A/C IR control signal. With the IoT platform and Telegram bot, it has not required to build a mobile app to make as URC. Therefore, the reader can extend this proposed solution to control the IR home devices automatically by setting their time for operating.

6.3 Future Work

This project can provide many IoT solution for further development. However, due to the time constraint in 14 weeks, many works still cannot implement it. The proposed solution can insert several sensors in the projects for making automated universal remote control. For example, the temperature sensor can detect the surrounding temperature to increase the surrounding temperature to optimum temperature so that power consumption can be conserved. Thus, A/C, while the temperature reaches too cold for the surrounding, RPi can send the IR signal for adjusting. A motion sensor can implement to control those IR LED bulbs with the same concept for the A/C when detected by the motion or movement of something.

Next, the signal of the A/C remote control can decode one by one. Since this project only focused on the increase, decrease temperature and power on/off of A/C. These functionalities are dull because we know there are still have much other functionality to control the A/C, such as changing the fan speed of the A/C, swing, or even changing the A/C mode. Thus, we narrowed the project's scope, and we just targeted to adjust the temperature and power on/off A/C only. Due to the A/C IR signal containing a lot of information (dynamic), there is another method to decode the signal. Therefore, to fully “reverse-engineer” for decoding and analysing the IR signal of A/C, it required a lot of research and testing.

Bibliography

Bartelmus C 2016, What is LIRC? [online] Available from: https://www.lirc.org/ [Accessed 15 April 2020]

Chaitanya, Sekhar and Ramesh, 2016, IOT based Smart IR Device using CC3200 [online] Available from: https://www.researchgate.net/profile/B_Sai_Chaitanya/publication/303377186_IOT_based_smart_IR_device_using_CC3200/links/575e8f7b08aed884621b4596/IOT-based-smart-IR-device-using-CC3200.pdf [Accessed 14 April 2020]

CNX Software - Embedded Systems News. 2021. *How to Control Your Air Conditioner with Raspberry Pi Board and ANAVI Infrared pHAT - CNX Software*. [online] Available at: <https://www.cnx-software.com/2017/03/12/how-to-control-your-air-conditioner-with-raspberry-pi-board-and-anavi-infrared-phat/> [Accessed 16 April 2021].

Cooking-hacks.com. 2020. *Control Your HVAC Infrared Devices From The Internet With IR Remote (Arduino UNO / Raspberry Pi Compatible)*. [online] Available at: https://www.cooking-hacks.com/documentation/tutorials/control-hvac-infrared-devices-from-the-internet-with-ir-remote/index.html [Accessed 4 December 2020].
Deshmukh, S,A Deshmukh

Deshmukh, S,A Deshmukh, S.A and Prof.Bansode (2018) Raspberry Pi Based Interactive Home Automation Gadget through the Net of Factors: A Survey

Faris, S. and Pi, T., 2021. *Telegram Bot With Raspberry Pi*. [online] Hackster.io. Available at: <https://www.hackster.io/Salmanfarisvp/telegram-bot-with-raspberry-pi-f373da> [Accessed 16 April 2021].

Gist. 2020. *Getting Lirc To Work With Raspberry Pi 3 (Raspbian Stretch)*. [online] Available at: <https://gist.github.com/prasanthj/c15a5298eb682bde34961c322c95378b> [Accessed 4 December 2020].

Medium. 2021. *MQTT Beginners Guide*. [online] Available at: <https://medium.com/python-point/mqtt-basics-with-python-examples-7c758e605d4> [Accessed 16 April 2021].

Meola A 2018, What is the Internet of Things? What IoT means and how it works Available from: https://www.businessinsider.com/internet-of-things-definition?IR=T. [15 April 2020].

python, What is Python? Executive Summary [online] Available from: https://www.python.org/doc/essays/blurb/ [Accessed 14 April 2020]

Raspberry, Raspberry Pi 3 Model B+ [online] Available from: https://static.raspberrypi.org/files/product briefs/200206+Raspberry+Pi+3+Model+B+plus+Product+Brief+PRINT&DIGITAL.pdf [Accessed 14 April 2020]

Raspberrypi.org. 2020. *[Stretch/Buster] Using LIRC With Kernel 4.19.X And Gpio-Ir - Raspberry Pi Forums*. [online] Available at: <https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=235256> [Accessed 4 December 2020].

RealVNC Help Center. 2020. *VNC Connect And Raspberry Pi*. [online] Available at: <https://help.realvnc.com/hc/en-us/articles/360002249917-VNC-Connect-and-Raspberry-Pi> [Accessed 4 December 2020].

Rouse,M 2018, smart home or building (home automation or domotics) Available from: https://internetofthingsagenda.techtarget.com/definition/smart-home-or-building. [15 April 2020].

Roy, J and Roy, J.K (2014, p73) Design of Smart Universal Remote using Mobile for Home Automation

Teoh, SK, 2019, MQTT as IoT Protocol, IoT Platform, lecture notes distributed in UCCE2513 Mini Project at The Universiti Tunku Abdul Rahman, Kampar Campus on 20 October 2019

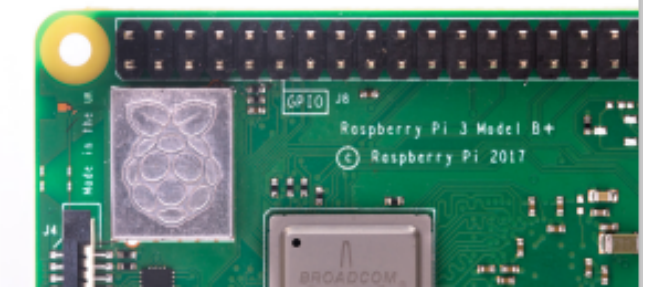
Texas Instruments, CC3200 SimpleLink™ Wi-Fi® and Internet of Things Solution with MCU LaunchPad™ Hardware [online] Available from: http://www.ti.com/lit/ug/swru372c/swru372c.pdf [Accessed 14 April 2020]

Appendices

Specifications of Raspberry Pi 3 Model B+

Specifications

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	■ 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE ■ Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) ■ 4 × USB 2.0 ports
Access:	Extended 40-pin GPIO header
Video & sound:	■ 1 × full size HDMI ■ MIPI DSI display port ■ MIPI CSI camera port ■ 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	■ 5V/2.5A DC via micro USB connector ■ 5V DC via GPIO header ■ Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50°C
Compliance:	For a full list of local and regional product approvals, please visit www.raspberrypi.org/products/raspberry-pi-3-model-b+
Production lifetime:	The Raspberry Pi 3 Model B+ will remain in production until at least January 2023.



raspberrypi.org

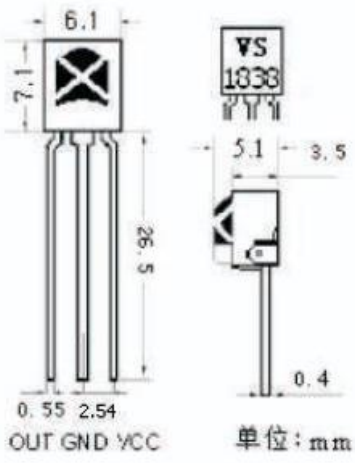
VS1838B

Infrared Receiver Module 红外线接收器

型号: VS1838B

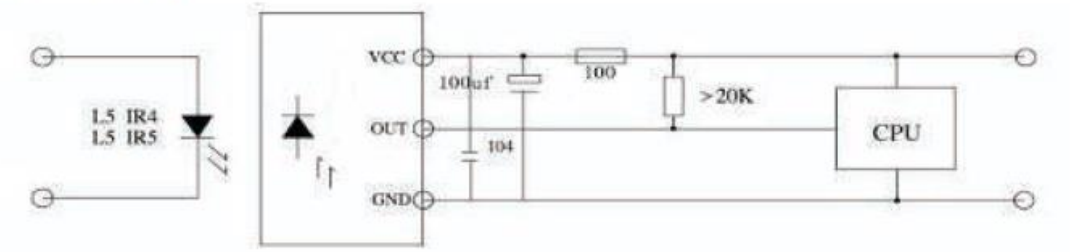
1. 特性
   - ●小型设计;
   - ●内置专用 IC;
   - ●宽角度及长距离接收;
   - ●抗干扰能力强;
   - ●能抵御环境光线干扰;
   - ●低电压工作;

外型尺寸及引脚排列图



2. 应用:
   - ■视听器材(音响,电视,录影机,碟机)
   - ■家用电器 (冷器机,电风扇,电灯)
   - ■其它无线电器遥控产品;

3. 应用电路图:



2N2222 NPN Transistor

Philips Semiconductors **Product specification**

NPN switching transistors **2N2222; 2N2222A**

FEATURES

- High current (max. 800 mA)
- Low voltage (max. 40 V).

APPLICATIONS

- Linear amplification and switching.

DESCRIPTION

NPN switching transistor in a TO-18 metal package.
PNP complement: 2N2907A.

PINNING

PIN	DESCRIPTION
1	emitter
2	base
3	collector, connected to case

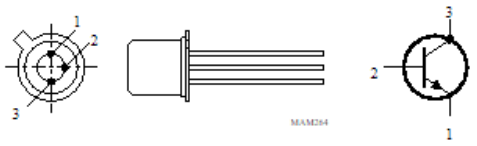


Fig.1 Simplified outline (TO-18) and symbol.

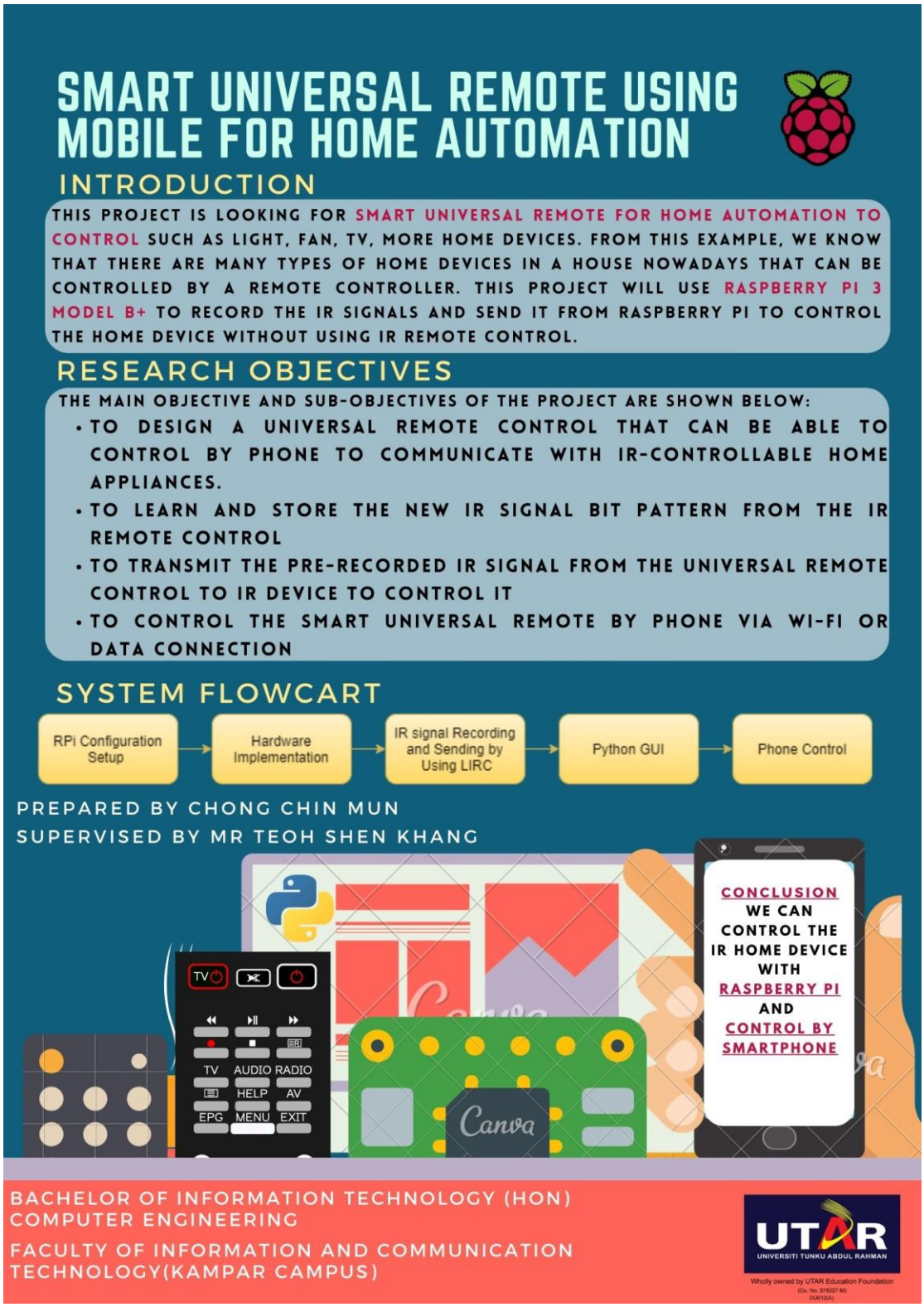
QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CBO}	collector-base voltage	open emitter			
	2N2222		–	60	V
	2N2222A		–	75	V
V_{CEO}	collector-emitter voltage	open base			
	2N2222		–	30	V
	2N2222A		–	40	V
I_C	collector current (DC)		–	800	mA
P_{tot}	total power dissipation	$T_{amb} \leq 25$ °C	–	500	mW
h_{FE}	DC current gain	$I_C = 10$ mA; $V_{CE} = 10$V	75	–	
f_t	transition frequency	$I_C = 20$ mA; $V_{CE} = 20$ V; $f = 100$ MHz			
	2N2222		250	–	MHz
	2N2222A		300	–	MHz
t_{off}	turn-off time	$I_{Con} = 150$ mA; $I_{Bon} = 15$ mA; $I_{Boff} = -15$ mA	–	250	ns

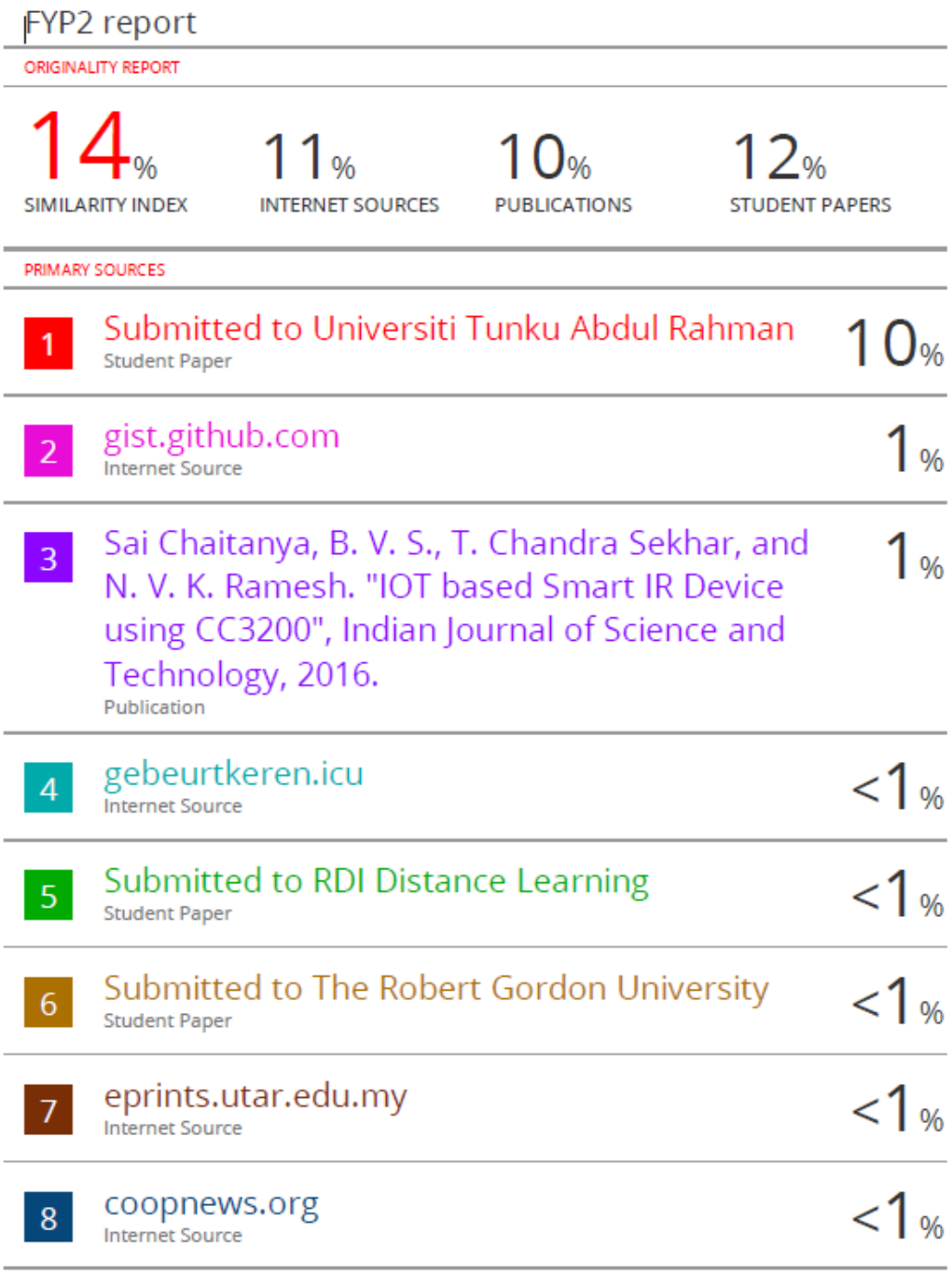
Drawings of Equipment

- Fritzing
- Drawio

Poster


SMART UNIVERSAL REMOTE USING MOBILE FOR HOME AUTOMATION
INTRODUCTION
THIS PROJECT IS LOOKING FOR SMART UNIVERSAL REMOTE FOR HOME AUTOMATION TO CONTROL SUCH AS LIGHT, FAN, TV, MORE HOME DEVICES. FROM THIS EXAMPLE, WE KNOW THAT THERE ARE MANY TYPES OF HOME DEVICES IN A HOUSE NOWADAYS THAT CAN BE CONTROLLED BY A REMOTE CONTROLLER. THIS PROJECT WILL USE RASPBERRY PI 3 MODEL B+ TO RECORD THE IR SIGNALS AND SEND IT FROM RASPBERRY PI TO CONTROL THE HOME DEVICE WITHOUT USING IR REMOTE CONTROL.
RESEARCH OBJECTIVES
THE MAIN OBJECTIVE AND SUB-OBJECTIVES OF THE PROJECT ARE SHOWN BELOW:
• TO DESIGN A UNIVERSAL REMOTE CONTROL THAT CAN BE ABLE TO CONTROL BY PHONE TO COMMUNICATE WITH IR-CONTROLLABLE HOME APPLIANCES.
• TO LEARN AND STORE THE NEW IR SIGNAL BIT PATTERN FROM THE IR REMOTE CONTROL
• TO TRANSMIT THE PRE-RECORDED IR SIGNAL FROM THE UNIVERSAL REMOTE CONTROL TO IR DEVICE TO CONTROL IT
• TO CONTROL THE SMART UNIVERSAL REMOTE BY PHONE VIA WI-FI OR DATA CONNECTION
SYSTEM FLOWCART
RPi Configuration Setup
Hardware Implementation
IR signal Recording and Sending by Using LIRC
Python GUI
Phone Control
PREPARED BY CHONG CHIN MUN
SUPERVISED BY MR TEOH SHEN KHANG
CONCLUSION
WE CAN CONTROL THE IR HOME DEVICE WITH RASPBERRY PI AND CONTROL BY SMARTPHONE
TV
TV AUDIO RADIO
HELP AV
EPG MENU EXIT
Canva
BACHELOR OF INFORMATION TECHNOLOGY (HON) COMPUTER ENGINEERING
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY(KAMPAR CAMPUS)
UTAR
UNIVERSITI TUNKU ABDUL RAHMAN

Plagiarism Check Result



FYP2 report

ORIGINALITY REPORT

14%	11%	10%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	10%
2	gist.github.com Internet Source	1%
3	Sai Chaitanya, B. V. S., T. Chandra Sekhar, and N. V. K. Ramesh. "IOT based Smart IR Device using CC3200", Indian Journal of Science and Technology, 2016. Publication	1%
4	gebeurtkeren.icu Internet Source	<1%
5	Submitted to RDI Distance Learning Student Paper	<1%
6	Submitted to The Robert Gordon University Student Paper	<1%
7	eprints.utar.edu.my Internet Source	<1%
8	coopnews.org Internet Source	<1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date:	Page No.: 1of 1

FACULTY OF Faculty of Information and Communication Technology

Full Name(s) of Candidate(s)	CHONG CHIN MUN
ID Number(s)	17ACB02573
Programme / Course	CT
Title of Final Year Project	Smart Universal Remote Using Mobile for Home Automation

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 14 % **Similarity by source** Internet Sources: 11 % Publications: 10 % Student Papers: 12 %	
Number of individual sources listed of more than 3% similarity: 1	10% is due to footer appeared in the database.
Parameters of originality required and limits approved by UTAR are as follows: (i) Overall similarity index is 20% and below, and **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to

Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

TEOH

Signature of Supervisor
Name: Teoh Shen Khang
Date: 16 April 2021

Signature of Co-Supervisor
Name:
Date:

FYP 2 Checklist

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	17ACB02573
Student Name	CHONG CHIN MUN
Supervisor Name	Mr Teoh Shen Khang

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. _____ (Signature of Student) Date:16/4/2021	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. TEOH _____ (Signature of Supervisor) Date: 16 April 2021