

INSTITUTE *of*  
TECHNOLOGY  

---

CARLOW

Institiúid Teicneolaíochta Cheatharlach

**Bachelor of Science (Honours)  
Software Development**

**Home Automation System  
Final Report**

**Course Code: CW238  
Student ID: C00212235  
Student Name: Xiaohui Ling  
Supervisor: Paul Barry**

Submission Date: 20/04/2020

## Abstract

This document is a final report to conclude my major project studying during this academic year. The project I studied is the Home Automation field to explain the system I developed working principle and the output of results during this project.

## Contents

1. Introduction .....	6
2. Project Description.....	6
3. Programming Language .....	6
3.1. Dart .....	7
3.2. Python .....	7
4. Development Environment.....	7
4.1. Linux/Debian .....	7
4.2. Flutter Framework .....	7
4.3. RPi.GPIO .....	8
5. Mid-ware Introduction.....	8
5.1. OpenWRT Firmware.....	8
5.2. AWS ActiveMQ.....	8
5.3. MQTT.....	8
5.4. Mjpeg-streamer .....	9
6. System Environment.....	9
7. Networks Communication .....	10
7.1. Wi-Fi .....	10
7.2. Internet .....	10
8. Communication Process.....	10
8.1. Local Network Communication Process .....	10
8.2. Wide Network Communication Process .....	12
9. Achieved.....	14
9.1. Hardware Assets .....	14
9.1.1. Home Appliances Control .....	14
9.1.2. Security Monitoring .....	20
9.1.3. Sensor Control.....	25
9.1.4. Hardware Integration.....	28
9.2. Software Asset .....	28
9.2.1. Flutter App .....	28
9.2.2. Raspberry Pi Python Scripts .....	36
10. Not Yet Achieved.....	37
10.1. Unit Test.....	37
10.2. Entertainment Module Implementation .....	37
11. Deviation .....	37
12. Process Methodology .....	39

13.	Milestone .....	41
14.	Problems Encountered .....	42
15.	Further Work.....	42
15.1.	Device Configuration Subsystem .....	42
15.2.	Data Statistical Analysis Subsystem .....	43
15.3.	The Router as a Bridge .....	45
16.	Challenges .....	45
	Acknowledgement .....	46
	Plagiarism Declaration .....	47
	Declaration.....	47

## Table of Figures

Figure 1 local network communication process 1 .....	11
Figure 2 local network communication 2 .....	12
Figure 3 wide network communication process 1.....	13
Figure 4 wide network communication process 2.....	14
Figure 5 light hardware design .....	15
Figure 6 light hardware implementation.....	16
Figure 7 fan hardware design .....	17
Figure 8 fan hardware implementation.....	17
Figure 9 shutter widget 1.....	18
Figure 10 shutter widget 2.....	18
Figure 11 shutter widget 3.....	18
Figure 12 shutter hardware design.....	19
Figure 13 shutter hardware implementation .....	19
Figure 14 cctv hardware design .....	20
Figure 15 cctv hardware implementation.....	21
Figure 16 door access hardware design.....	22
Figure 17 door access hardware implementation front.....	23
Figure 18 door access hardware implementation back.....	24
Figure 19 temperature & humidity sensor hardware design .....	26
Figure 20 temperature & humidity sensor hardware implementation.....	27
Figure 21 whole project devices integration .....	28
Figure 22 home screen.....	29
Figure 23 devices control panel screen .....	29
Figure 24 network options.....	30
Figure 25 light control panel 1 .....	30
Figure 26 light control panel 2 .....	31
Figure 27 fan control panel 1.....	31
Figure 28 fan control panel 2.....	32
Figure 29 shutter control panel .....	32
Figure 30 cctv control panel 1.....	33
Figure 31 cctv control panel 2.....	33
Figure 32 door access control panel 1 .....	34
Figure 33 door access control panel 2 .....	34
Figure 34 indoor atmosphere panel .....	35
Figure 35 outdoor atmosphere panel.....	35
Figure 36 game option .....	36
Figure 37 original design.....	38
Figure 38 a real implementation.....	39
Figure 39 user story wall.....	40
Figure 40 milestone .....	41
Figure 41 visualization 1.....	43
Figure 42 visualization 2.....	44
Figure 43 visualization 3.....	44
Figure 44 visualization 4.....	45

## 1. Introduction

This Home Automation System is a real application created from scratch. Through this project, understanding the working principle of the underlying hardware, to introduce how to create my own automation home rather than using an open-source home automation system to build.

## 2. Project Description

The project aims to research the Home Automation field. According to this aim the system I designed that mainly involved aspects included home appliances remotely control, security monitoring, sensor control and home entertainment. Each of those can be controlled and managed through the mobile app.

Further on the home appliances remotely control, there are 3 types home appliance were implemented in this Home Automation System that are respectively table light, fan and shutter. They can be controlled via Wi-Fi in which home local network, or across Internet. The light also can be controlled with the voice of the householder. The fan and shutter are models I created by my hands.

The security monitoring consists of the CCTV camera and door access subsystem. The CCTV camera can capture the live view that is able to show on the screen of a smart mobile phone in real-time. The door access subsystem is a sample model created by my hands that implemented a system that maintains a doorbell, doorbell button, door camera, and door lock, put them on works together. Through manual tests, that system can emulate a scenario that a visit process of a guest from standing at the door to enter the house.

The sensor control takes advantage of two sensors to collect the temperature and humidity measurement from indoor and outdoor. When the householder would like to know the measurement of the current staying environment, the householder can use the mobile app to view the measurement in real-time.

Finally, the entertainment module is designed for the householder who playing the games. I planned to integrate last academic year projects in college that respectively are Bomb Man and Word Game to this Home Automation System. Due to does not have enough time to implement it, so that module should be done later.

With the gradual popularization of home automation, our lives may change dramatically in the next 5 to 10 years. Before the shock that the home automation will bring, through this project, I can dive to understand the working principle of home automation.

## 3. Programming Language

In this project, mainly used the Dart and Python programming languages. Next section will describe them.

### 3.1. Dart

The home automation system takes advantage of Dart programming language to implement the system features that mobile app to manage and maintain remotely the home equipment. Dart was found by Google in 2011. It is an open-source general-purpose object-oriented programming language. Its coding style seems extremely like Java and JavaScript that means it is easier to learn and use. Dart is cross-platform and supports all major operating systems.

Meanwhile, it is not only designed to web front-end but also can be used to develop a mobile app supports Android OS and iOS both mobile phones.

### 3.2. Python

Python is an object-oriented high-level programming language that is a great language for the beginner who never touches the coding before. It also supports functional and structured programming patterns. Python widely applies to big data analysis, machine learning and execution script. In this project, as an execution script programming language that is used to implement the home equipment controls by Raspberry Pi.

Currently, the version of python used is 3.7 in this project.

## 4. Development Environment

### 4.1. Linux/Debian

Linux is an open-source high-performance small size kernel operating system that usually applied to cloud virtual servers and tiny smart devices. In recent years, It has also been popular as a desktop operating system such as Ubuntu which bases on a Debian Linux kernel. This project using the Raspberry Pi that was designed to run the Debian Linux kernel operating system due to a limitation of CPU and memory. The python programming language can integrate into Linux very well as execution scripts that easily maintain tasks where run on the computer.

In this project, the python just also as execution scripts in Linux to maintain the status of the home equipment.

### 4.2. Flutter Framework

Flutter is an open-source mobile app development framework that was developed by Google found in 2017. It supports both platforms Android and iOS used to develop an app runs on mobile phones. This framework is implemented using Dart programming language that also means the framework is used to develop a mobile app easier. The flutter framework comes with two UI styles that are Material and Cupertino in which respectively corresponding to Android and iOS. It provides a Scaffold component (or called widget in Flutter) that is very useful to quickly create a beautiful layout of a screen.

Google official claims that the mobile app is developed by the Flutter framework would smoother rather than others else which not developed by Flutter.

### 4.3. RPi.GPIO

The RPi.GPIO is a module implemented with python that used to send high-level or low-level to the GPIO pins on a Raspberry Pi through analog signal and digital signal transforms thus maintains the status of the terminal device. In recent years, the RPi.GPIO was widely applied to massive micro-electronic mechanic system (MEMS).

In this project, when the Raspberry Pi received the message sent by the mobile app, the RPi.GPIO would be invoked to change the status of the home equipment.

## 5. Mid-ware Introduction

### 5.1. OpenWRT Firmware

The OpenWRT firmware is an open-source router embedded operating system based on a Linux kernel that offers a network traffic route feature. Due to this project need an information exchange centre between clients which such as among mobile phone, Raspberry Pi, cloud service, etc. Because of security reasons, the Internet Service Providers (ISP) do not allow the householder to custom the features of the route offered by them. So, the OpenWRT is required in this case. The OpenWRT allows you to login to it through an SSH client-side program and maintains its operating system such as installing software that you want or removing software you will not use in the future.

In this project, an MQTT implementation version called Mosquitto MQTT broker is installed into the router which flashed OpenWRT firmware. Meanwhile, a router that has flashed the OpenWRT firmware that is easier to deploy an application which as a bridge to transfer the messages between AWS ActiveMQ and Raspberry Pi.

### 5.2. AWS ActiveMQ

The ActiveMQ is an open-source message queue mid-ware provided by the Apache foundation that is integrated into AWS as a cloud service. It supports multiple networking communication protocols such as WSS, MQTT, STOMP, AMQP, OpenWire. Meanwhile, the AWS ActiveMQ offers a broker configuration and management features on AWS console and it also provides an independent ActiveMQ console to monitoring the traffic of the message for each topic.

In this project, the AWS ActiveMQ is used to offer a cloud broker ability that as a bridge between the home local router and mobile phone when the mobile app communicates with the home router through the Internet.

### 5.3. MQTT

MQTT is a lightweight message transport protocol that provides the publish and subscribe method to exchange information through wireless between the mobile phone and the devices. It was widely applied to the IoT field that is able to communicate at low latency and high performance. In the MQTT protocol, It used the classical C/S architecture, the MQTT Client that either a publisher or subscriber, even both publisher and subscriber at the same time. The MQTT broker as a server between clients to transfer the information.



In this project, the Raspberry Pi takes advantage of a version called Paho-MQTT as its client to publish/subscribe message to/from MQTT broker within the router that installed a Mosquitto MQTT version. The mobile app used the MQTT-Client lib provided by the Flutter framework. As a cloud service, AWS ActiveMQ that is also a MQTT broker the other clients can communicate with. The desktop computer you work on, it also can install a Mosquitto MQTT client that such is convenient to debug when they interact with each other.

#### 5.4. Mjpeg-streamer

The Mjpeg-streamer as a command-line program that is installed on the Linux of the Raspberry Pi to control the USB camera and Raspberry Pi camera of which type as an input. When the Mjpeg-streamer turns on the camera by issue a command on the command line, the client who looks forward to receiving the live stream from camera could have multiple ways such as reading a file or through the HTTP protocol, etc.

In this project, the Raspberry Pi using a python script that issue a command to turn on the USB camera and Raspberry Pi camera.

## 6. System Environment

I would like to declare system environment versions which this project used that is listed below for each of the those I used.

Dart

Dart 3.9.1

Python

Python3.7.3

Raspberry Pi Linux

Raspberry Pi 4.19.75-v7l+ #1270

Flutter

Channel dev, v1.15.18

OpenWRT

openwrt-mt300n-v2-3.102

MQTT Broker in Router

Version 3.0.1-1

Mjpeg-stremer

Version: 501f6362c5afddcfb41055f97ae484252c85c912

## 7. Networks Communication

### 7.1. Wi-Fi

In this project, it uses the WiFi protocol to provide the local network service between the mobile phone and Raspberry Pi to exchange information. The Raspberry Pi and mobile phone both have already had the WiFi feature that means it is easier to implement the project. The disadvantages that not secure enough and the number of nodes supported is small. However, in this stage of the project is not too matter.

### 7.2. Internet

Due to the system requires that the home appliances should be able to control across the Internet, So this project will take advantage of AWS cloud to provide a capability which is able to control the devices between Raspberry Pi and mobile phone across the Internet. The AWS service called ActiveMQ that is a message queue mid-ware offer the publish and subscribe method to exchange information between networks. It is also compatible with the MQTT protocol.

## 8. Communication Process

There are diagrams described process of communication via networks between mobile app and terminal devices.

### 8.1. Local Network Communication Process

The mobile app actively publishes the message to the MQTT broker then the device control center subscribe to the message to trigger a modification event that updates the status of the device.

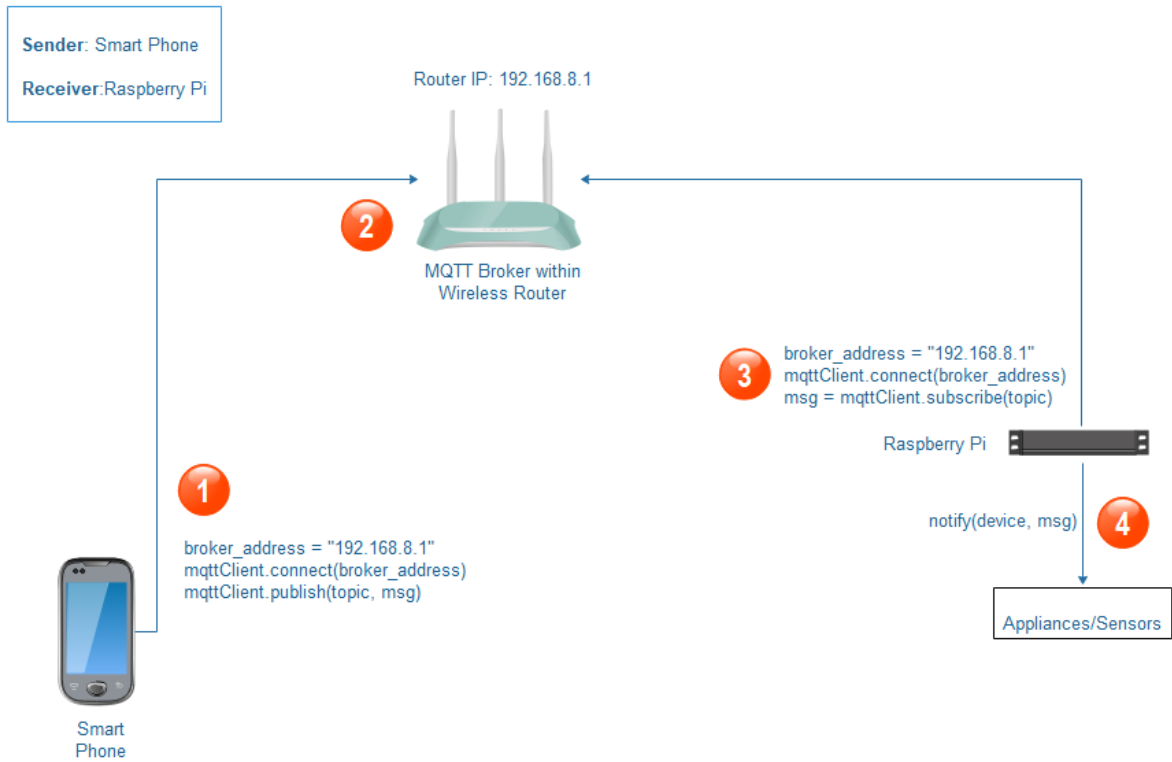


Figure 1 local network communication process 1

The terminal device actively publishes the message to the MQTT broker via Raspberry Pi then mobile app subscribes to the message to trigger a modification event that updates the status of the UI component.

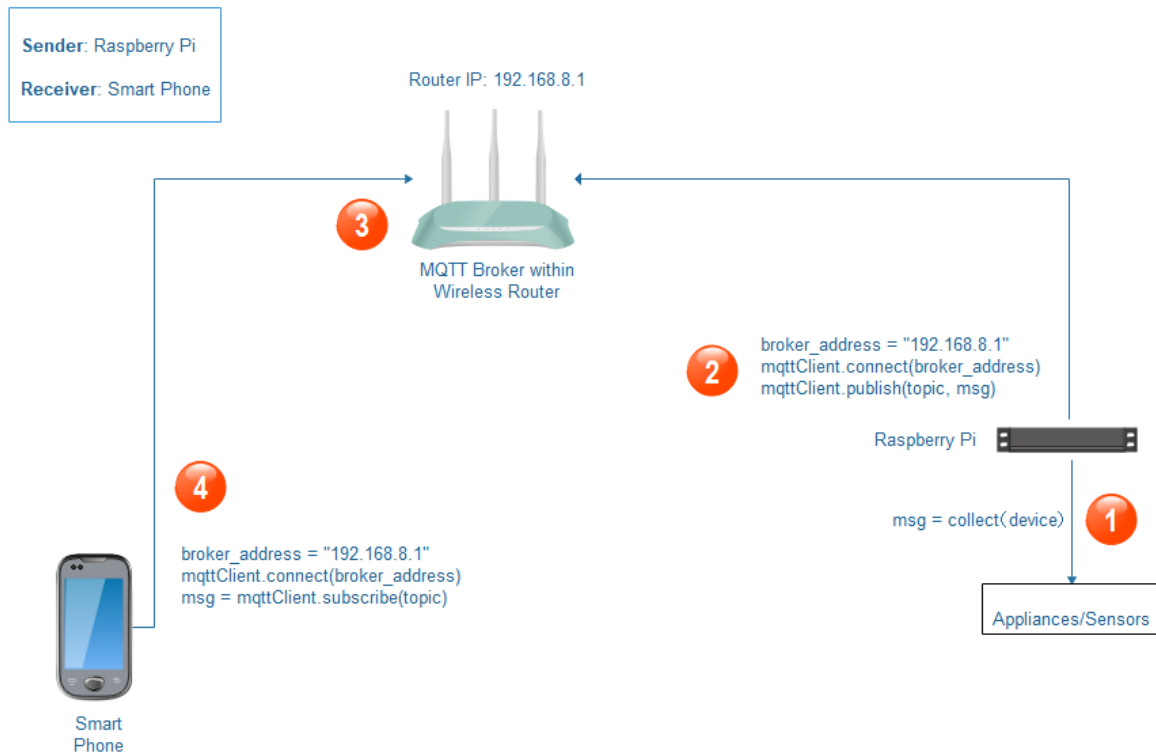


Figure 2 local network communication 2

## 8.2. Wide Network Communication Process

The mobile app actively publishes the message to the MQTT broker then the device control center subscribe to the message to trigger a modification event that updates the status of the device.

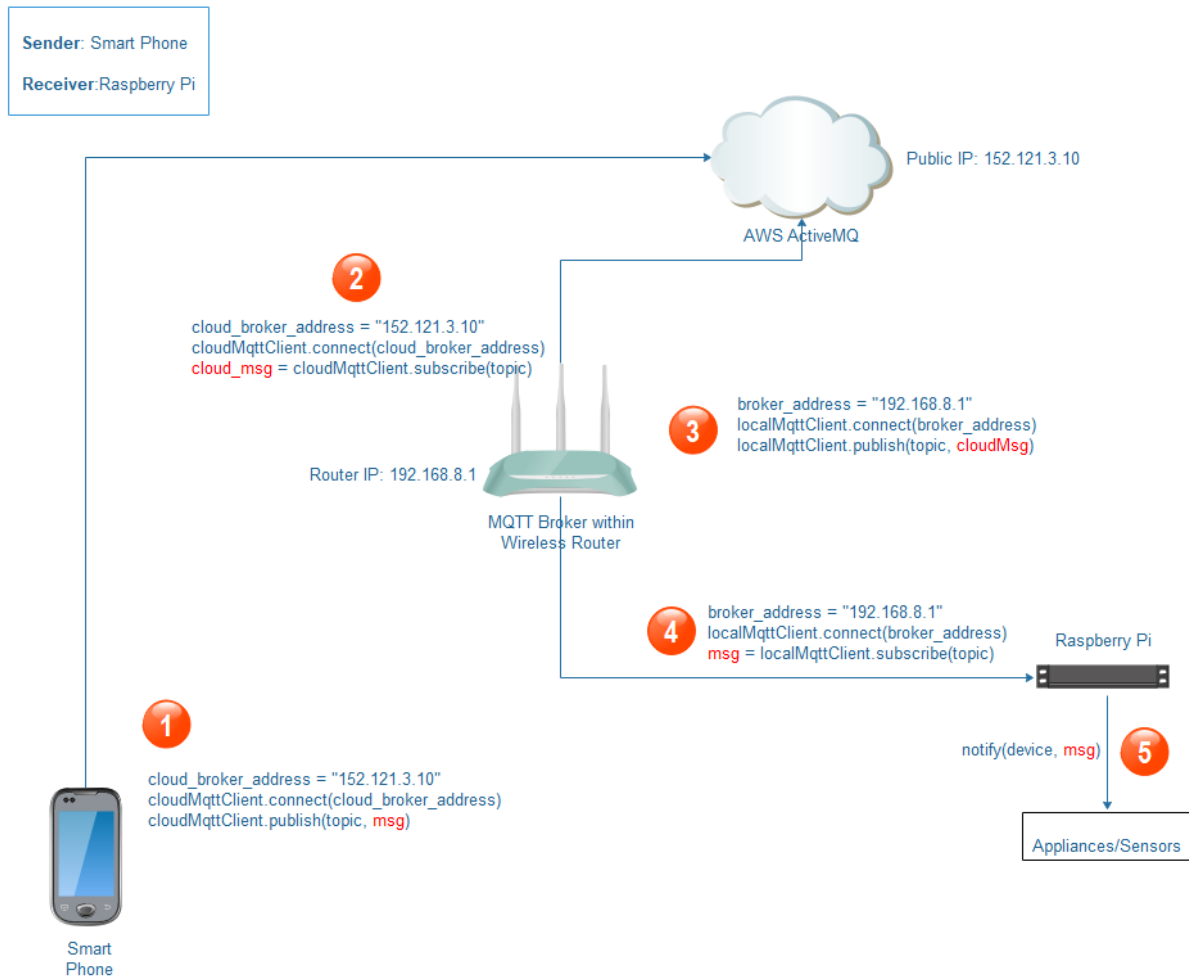


Figure 3 wide network communication process 1

The terminal device actively publishes the message to the MQTT broker via Raspberry Pi then mobile app subscribes to the message to trigger a modification event that updates the status of the UI component.

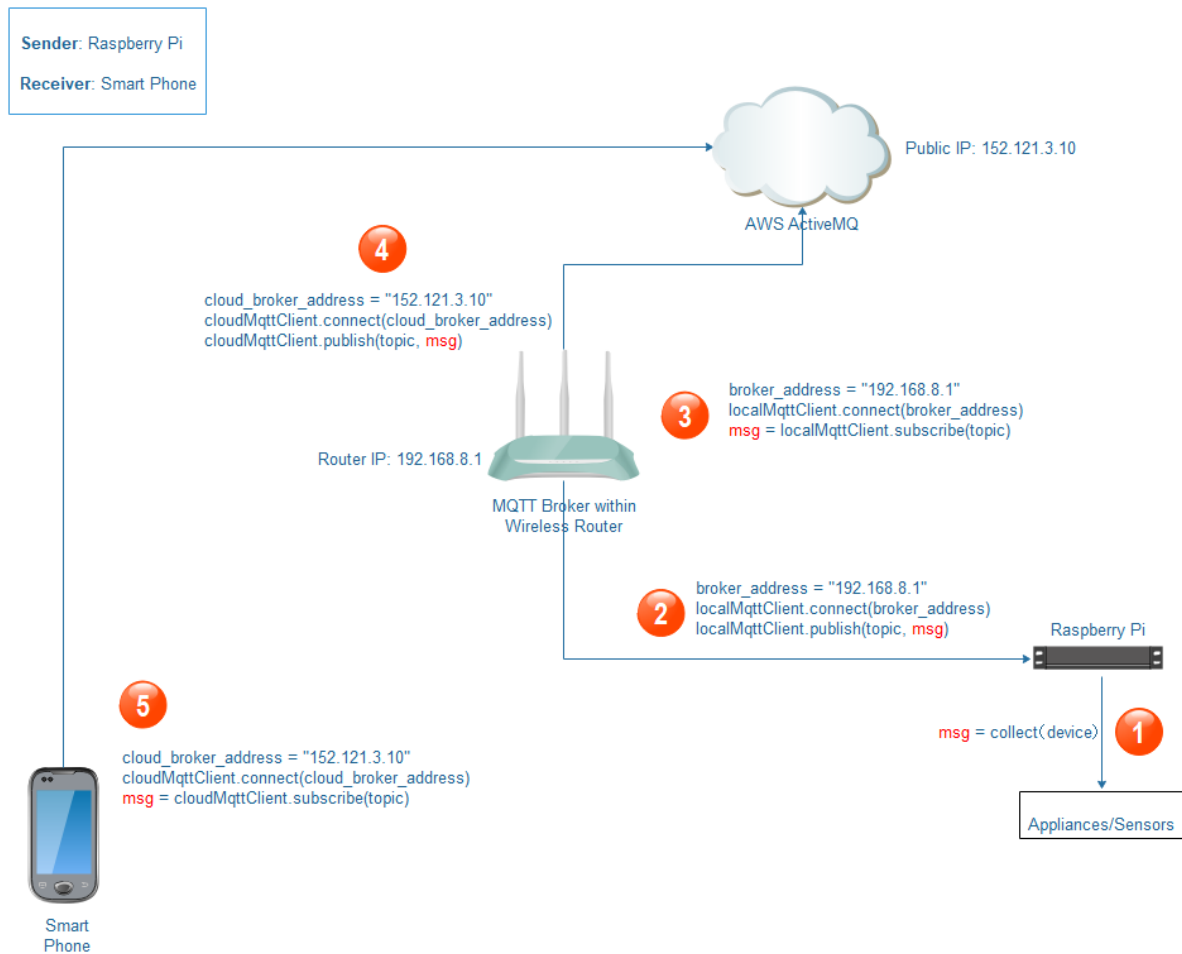


Figure 4 wide network communication process 2

## 9. Achieved

At the end of the Home Automation System of this stage of the development, there are two parts asset I achieved that are hardware assets and software assets, they are listed below.

### 9.1. Hardware Assets

In this part of the Assets, I conform the hardware design planning in which is completed in the system design phase. That mainly involved home appliances control, security monitoring, and sensors control.

#### 9.1.1. Home Appliances Control

There are 3 types of home appliance that are controlled in the Home Automation System to be implemented.

### 9.1.1.1. Light

The table light as a home appliance that was designed to connect to a Raspberry Pi. Like a switch the relay module is placed between the Raspberry Pi and the Light. The Raspberry Pi maintains a switch circuit of the relay module to keep the power line of the light connecting or disconnecting.

- Hardware Design

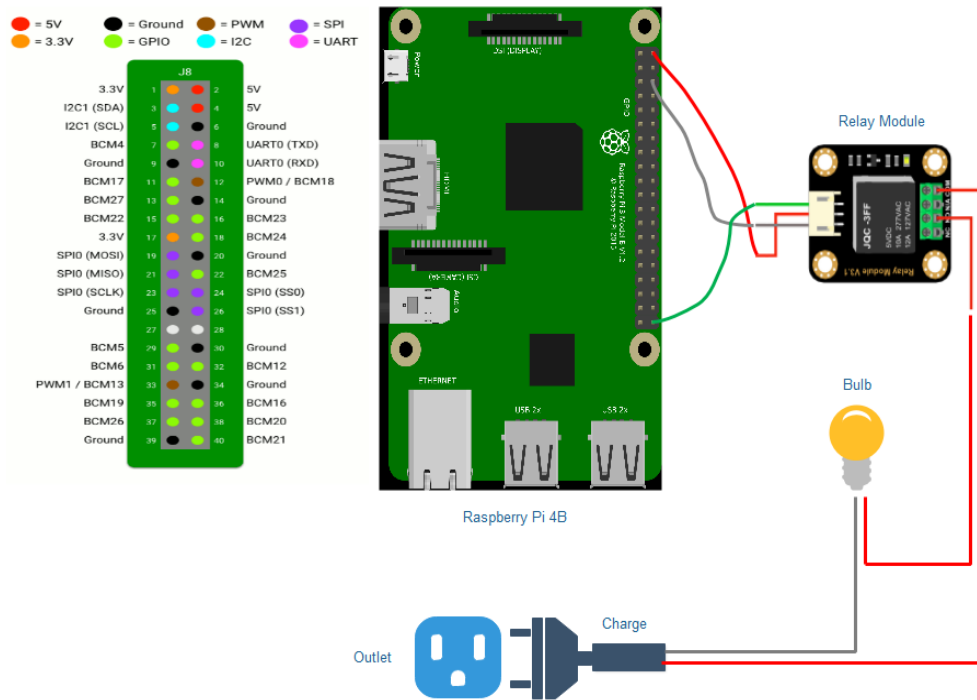


Figure 5 light hardware design

- Hardware Implementation

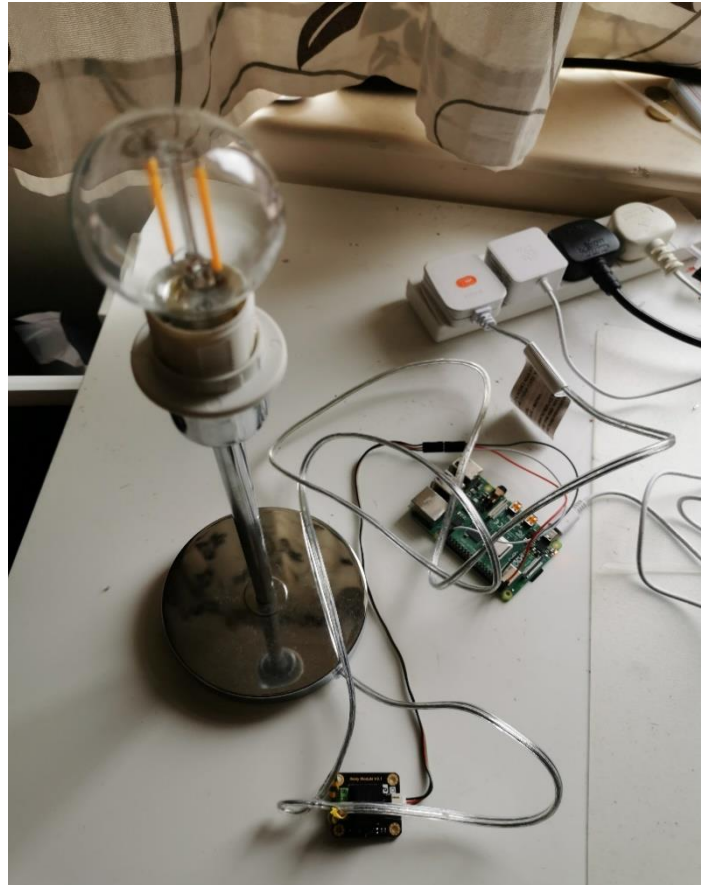


Figure 6 light hardware implementation

- Key Hardware Parameters

1. Raspberry Pi and Relay Module

Raspberry Pi	Relay Module	Wire Colour
5v Pin	Positive Pin	Red
GND Pin	Negative Pin	Grey
BCM 21	I/O Pin	Green

2. Relay Module and Light

Relay Module	Light	Charge of Light	Wire Colour
Normal Open Pin	Positive of Charge	<i>Not Used</i>	Red
COM Pin	<i>Not Used</i>	Positive of Charge	Red

### 9.1.1.2. Fan

The fan also as a home appliance that was designed to be managed by the Raspberry Pi. In this case, there is a motor driver module is placed between the fan and the Raspberry Pi, which means, the Raspberry Pi can turn on or off the fan through the motor driver module, even adjust the speed of the fan. The motor driver module powered by the Raspberry Pi, but the fan powered by motor driver module, that means, the Raspberry Pi easily to power on or off the motor driver module, thus, further the fan is shutdown.

In this stage, I simply create a model of the fan with a motor that connects to the motor driver module. In this case, a PWM pin where on Raspberry Pi board should be used to connect the input pin of the motor driver module, therefore, the motor driver is able to adjust the speed of the fan.



- Hardware Design

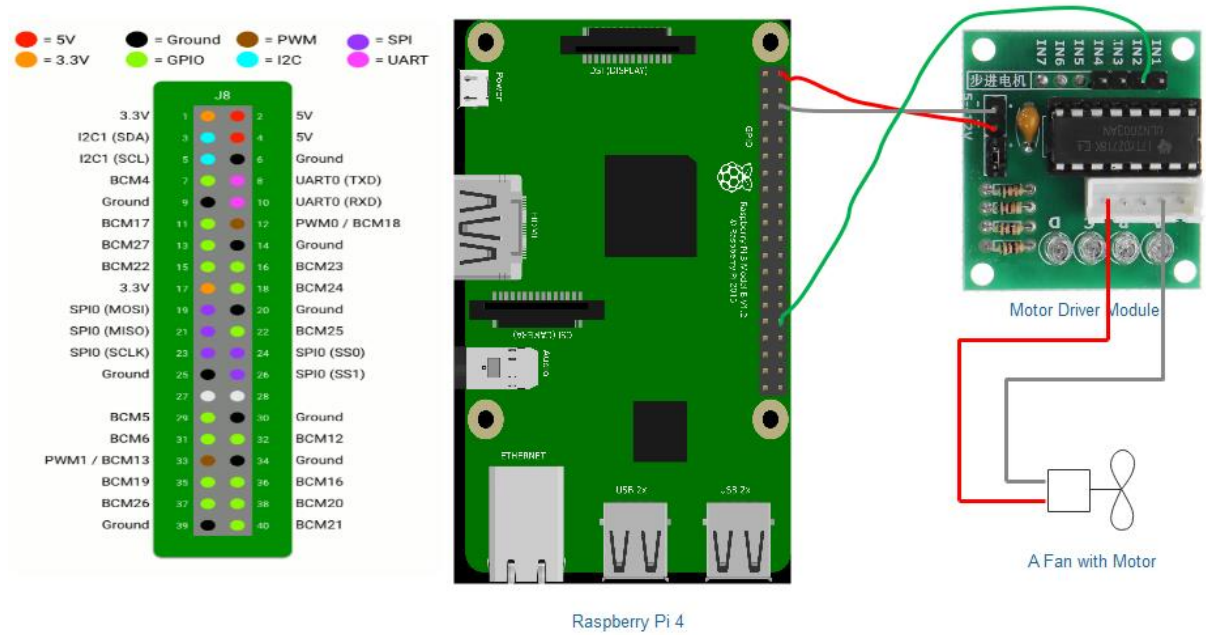


Figure 7 fan hardware design

- Hardware Implementation



Figure 8 fan hardware implementation

- Key Hardware Parameters

1. Raspberry Pi and Motor Driver Module

Raspberry Pi	Motor Driver Module	Wire Colour
--------------	---------------------	-------------

5v Pin	Positive Pin	Red
GND Pin	Negative Pin	Grey
BCM 12	In2 Pin	Green

## 2. Motor Driver Module and Fan Motor

Motor Driver Module	Fan Motor	Wire Colour
First Pin in Slot	Positive	Red
Fourth Pin Slot	<i>Negative</i>	Grey

### 9.1.1.3. Shutter

The shutter is a sample model I created that also can be controlled by the Raspberry Pi.

This sample model just takes advantage of a mechanical system that consist of the motor, motor driver module, nuts and rebar. Further on the part of nuts and rebar, that structure is shown below.



Figure 9 shutter widget 1



Figure 10 shutter widget 2



Figure 11 shutter widget 3

Once the motor is driven by the motor driver module, the rebar will be rotating. Since the rebar was fixed, the nut is stuck with wood and cannot rotate with the rebar, so the nut must move in the direction of the rebar's rotation. So, if the motor changed the direction of rotation, the nut must move reverse.

The motor driver module works between the Raspberry Pi and the motor that principle like the Fan which mentioned before.

- Hardware Design

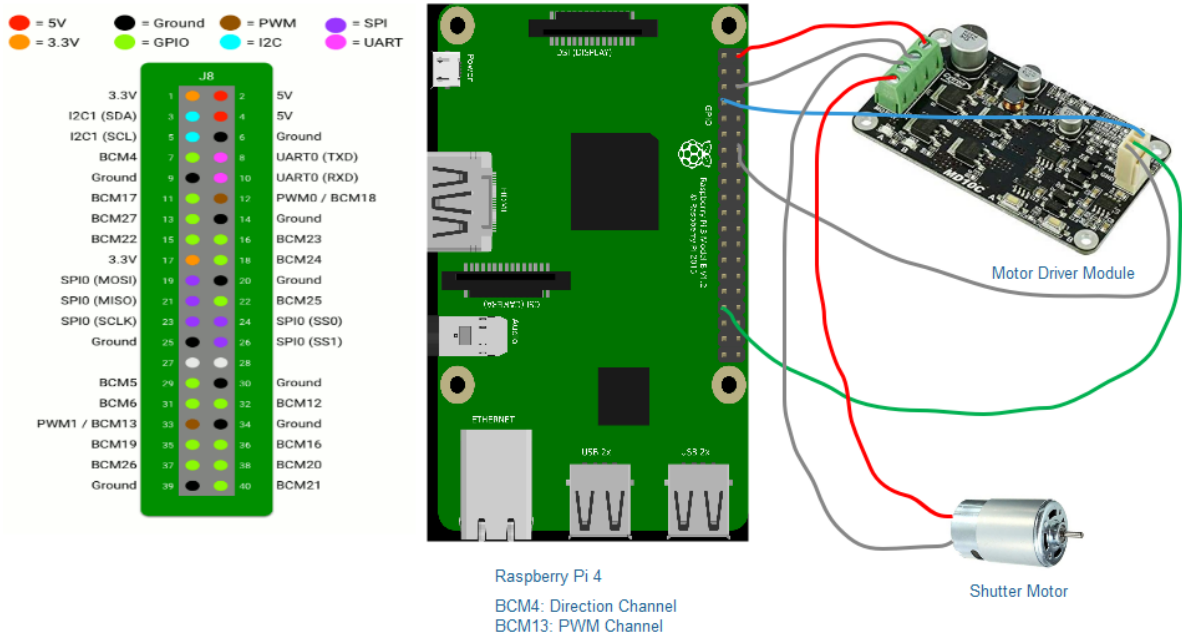


Figure 12 shutter hardware design

- Hardware Implementation

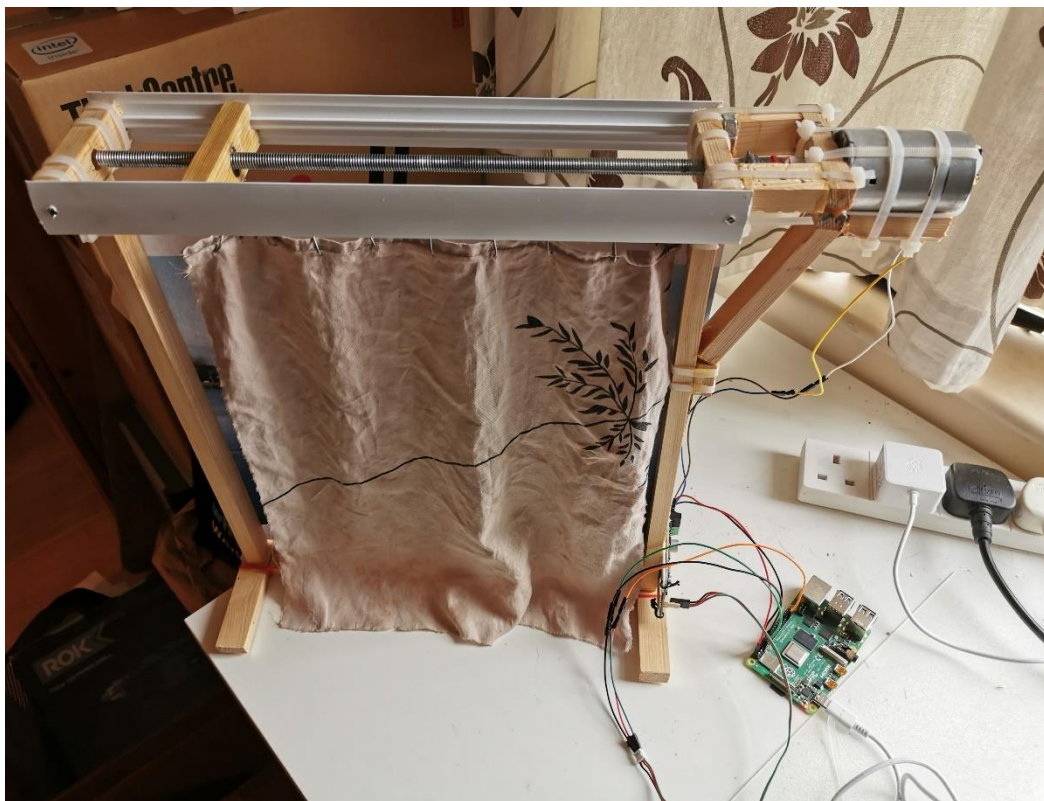


Figure 13 shutter hardware implementation

- Key Hardware Parameters

- Raspberry Pi and Motor Driver Module

Raspberry Pi	Motor Driver Module	Wire Colour	Description
5v Pin	Positive Pin	Red	
GND Pin	Negative Pin	Grey	
BCM 4	DIR Pin	Blue	Control motor direction
BCM 13	PWM Pin	Green	Control motor speed
GND Pin	GND Pin	Grey	

- Motor Driver Module and Fan Motor

Motor Driver Module	Shutter Motor	Wire Colour
A Pin	Positive	Red
B Pin	Negative	Grey

### 9.1.2. Security Monitoring

There are 2 types of home security monitoring that are controlled in the Home Automation System to be implemented.

#### 9.1.2.1. CCTV

The CCTV is a USB camera that connects to a USB interface of Raspberry Pi with a cable. Meanwhile, the Raspberry Pi has already installed the Mjpeg-streamer program in which mentioned above that can drive the USB camera to works.

- Hardware Design

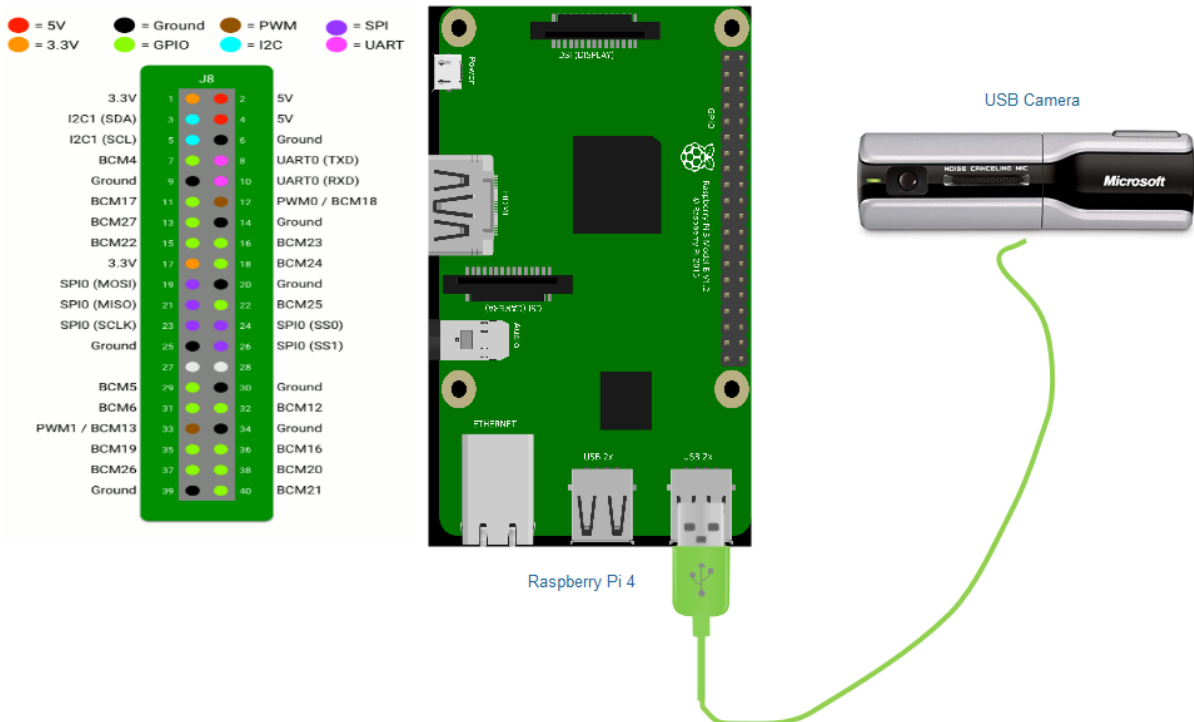


Figure 14 cctv hardware design

- Hardware Implementation

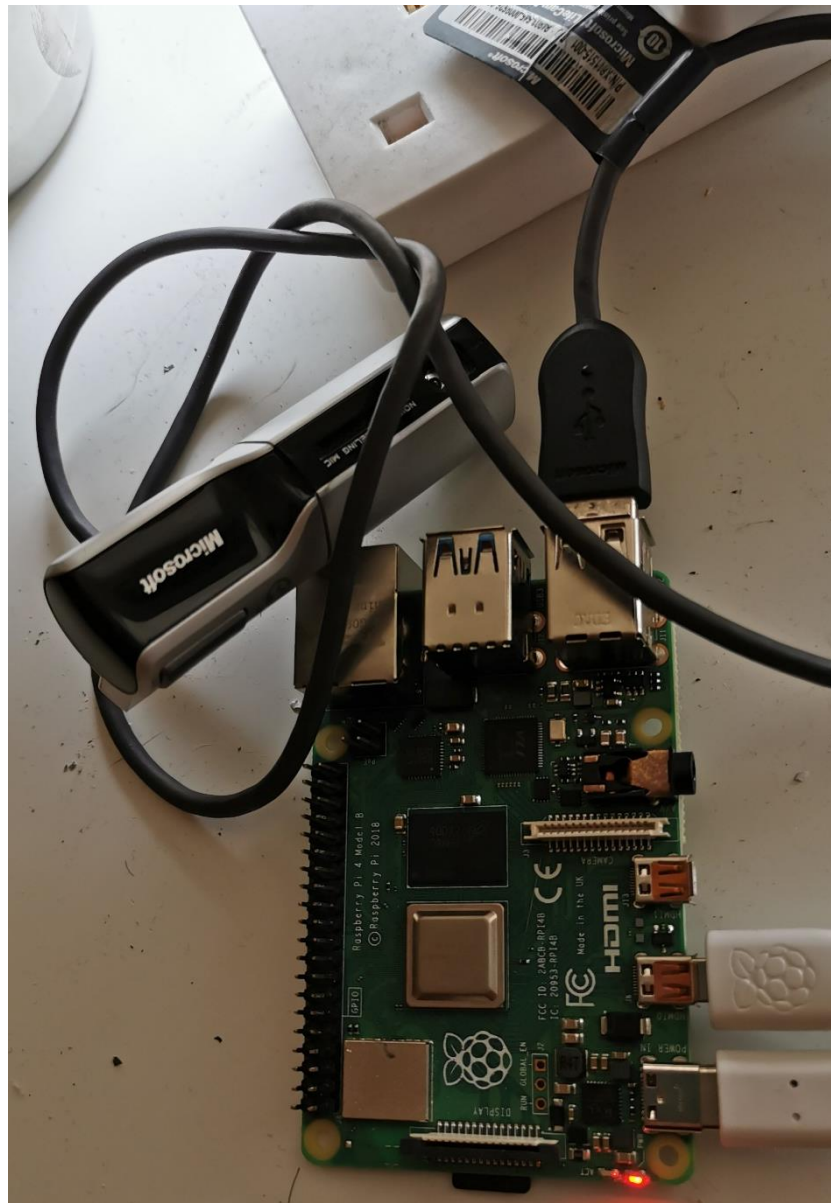


Figure 15 cctv hardware implementation

- Key Hardware Parameters
  1. Raspberry Pi and USB Camera

Raspberry Pi	USB Camera	Wire Colour
USB Interface	USB Cable	Green

#### 9.1.2.2. Door Access

In fact, the door access as a subsystem in the home automation system that puts on the multiple devices works together. The devices in the door access subsystem connect to the Raspberry Pi, and they work through the high-level or low-level signal sent by the Raspberry Pi.

- Hardware Design

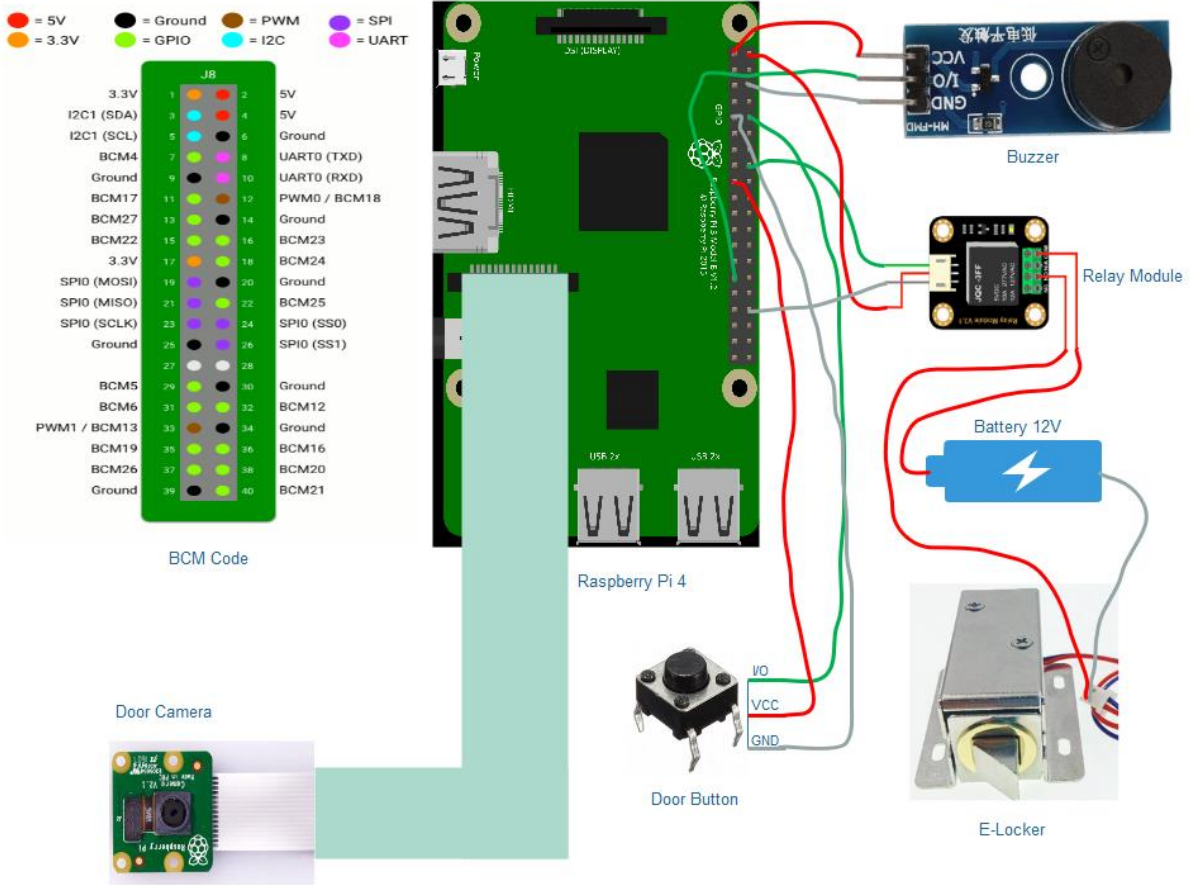


Figure 16 door access hardware design

- Hardware Implementation



Figure 17 door access hardware implementation front

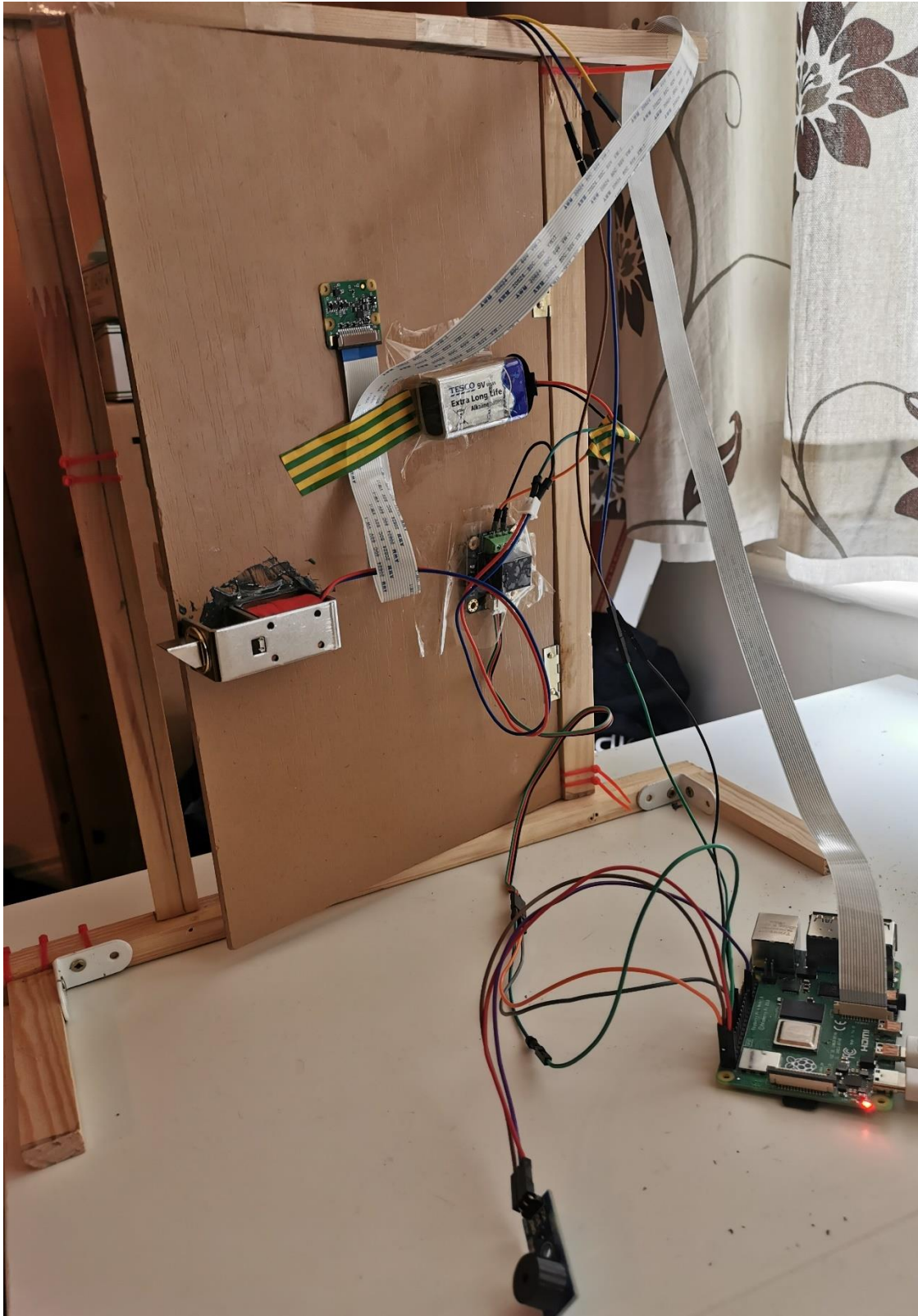


Figure 18 door access hardware implementation back

- Key Hardware Parameters
  1. Raspberry Pi and Buzzer (Doorbell)

Raspberry Pi	Buzzer (Doorbell)	Wire Colour
3.3v Pin	VCC Pin	Red



GND Pin	GND Pin	Grey
BCM 5	I/O Pin	Green

2. Raspberry Pi and Doorbell Button

Raspberry Pi	Doorbell Button	Wire Colour
3.3v Pin	VCC Pin	Red
GND Pin	GND Pin	Grey
BCM 15 (RXD)	I/O Pin	Green

3. Raspberry Pi and Relay Module

Raspberry Pi	Relay Module	Wire Colour
5v Pin	Positive Pin	Red
GND Pin	Negative Pin	Grey
BCM 23	I/O Pin	Green

4. Relay Module, Lock and Battery

Relay Module	Lock	Battery	Wire Colour
Normal Open Pin	Positive of Charge	<i>Not Used</i>	Red
COM Pin	<i>Not Used</i>	Positive	Red

5. Raspberry Pi and Raspberry Camera

Raspberry Pi	Raspberry Camera	Wire Colour
Raspberry Camera Interface	Camera Cable	Green

### 9.1.3. Sensor Control

As an indispensable part of the Home Automation System, there are two sensors are placed in this project that collects the temperature and humidity measurement and sent the data to the mobile app is shown in real-time.

#### 9.1.3.1. Temperature & Humidity Sensor

Temperature & Humidity sensor is a device used to collect the environment's data of which outdoor and indoor in this Home Automation System. It has high-sensitivity induction to sense with staying environment. So in this part, the very important thing is that should be a real-time interaction between sensors and mobile app.

- Hardware Design

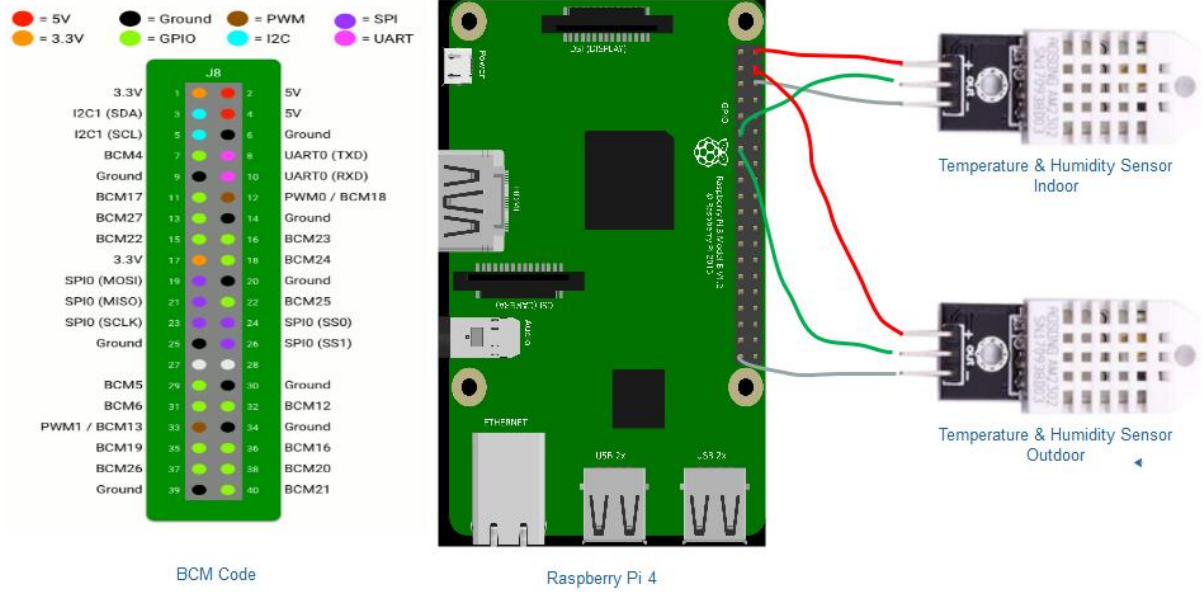


Figure 19 temperature & humidity sensor hardware design

- Hardware Implementation

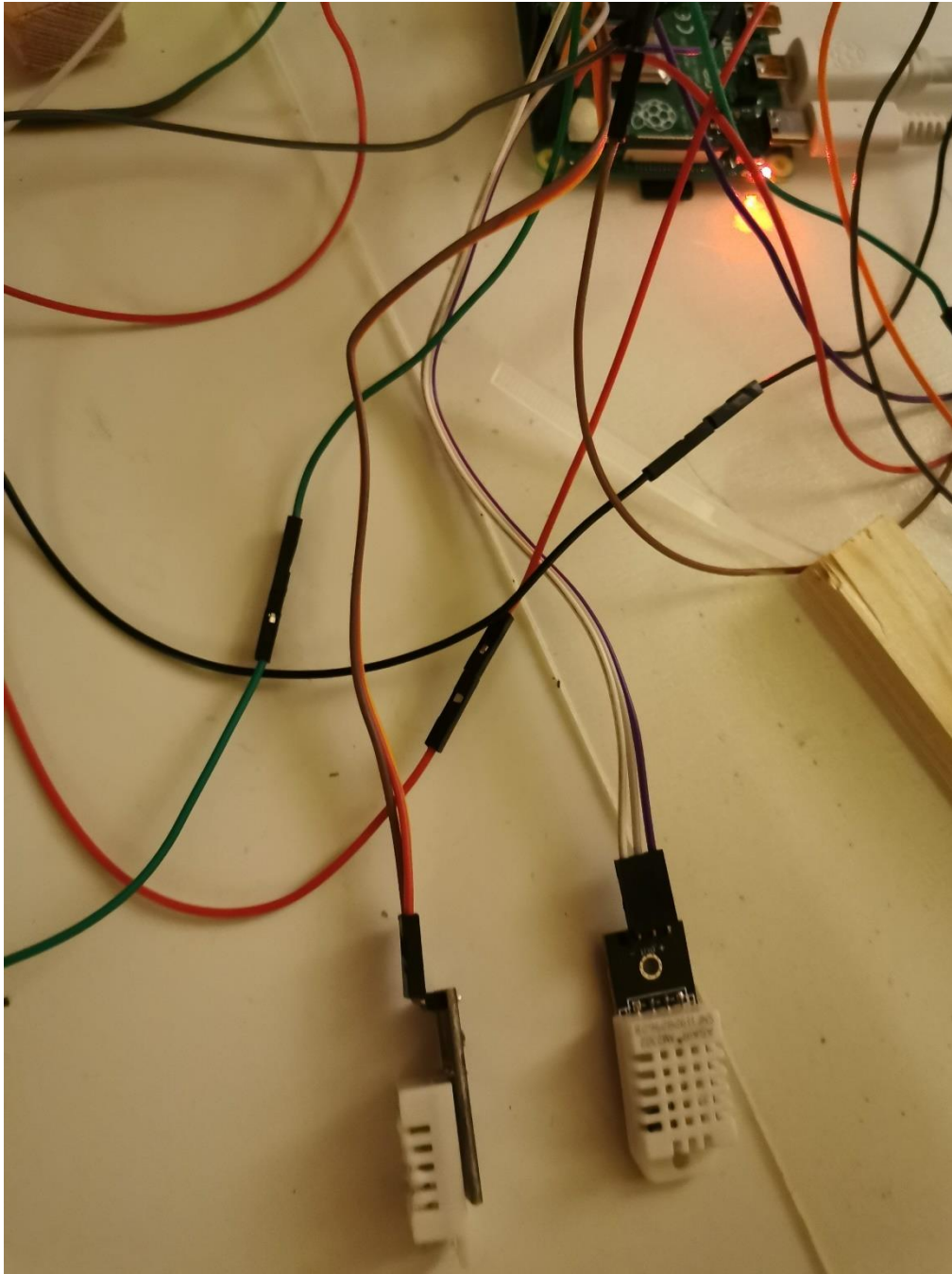


Figure 20 temperature & humidity sensor hardware implementation

- Key Hardware Parameters

1. Raspberry Pi and Indoor Sensor

Raspberry Pi	Indoor Sensor	Wire Colour
5v Pin	Positive Pin	Red
GND Pin	Negative Pin	Grey
BCM 17	I/O Pin	Green

2. Raspberry Pi and Outdoor Sensor

Raspberry Pi	Indoor Sensor	Wire Colour
5v Pin	Positive Pin	Red
GND Pin	Negative Pin	Grey

### 9.1.4. Hardware Integration

To integrate all components of this project, that should need to more jump wire and a breadboard to extend the input or output pins on where the Raspberry Pi.

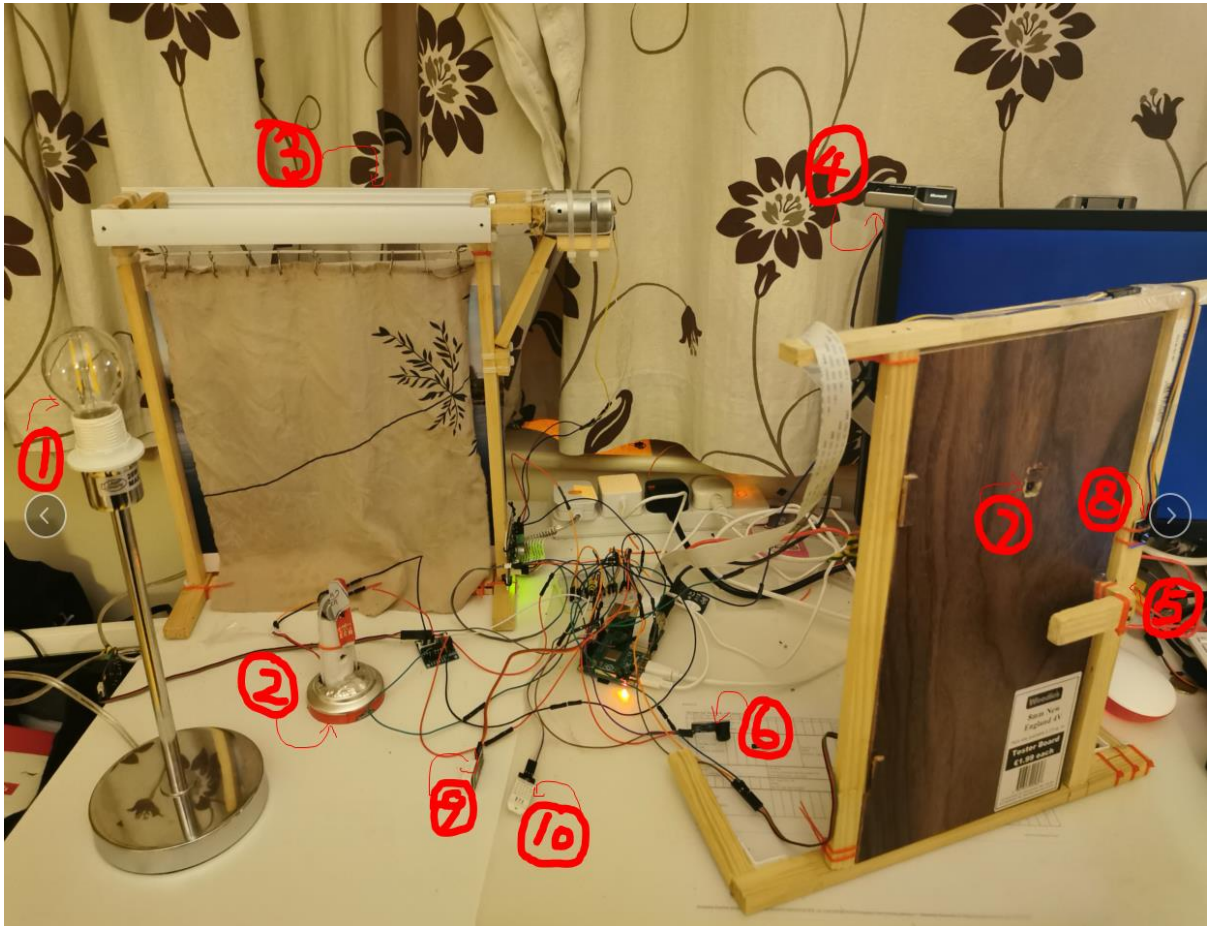


Figure 21 whole project devices integration

#### Description

1	Light	6	Doorbell
2	Fan	7	Door Camera
3	Shutter	8	Doorbell Button
4	CCTV	9	Indoor Sensor
5	Door Lock	10	Outdoor Sensor

## 9.2. Software Asset

At the end of the project should produce the software to run as an application. There are two executable programs to delivery that are a flutter mobile app and python scripts.

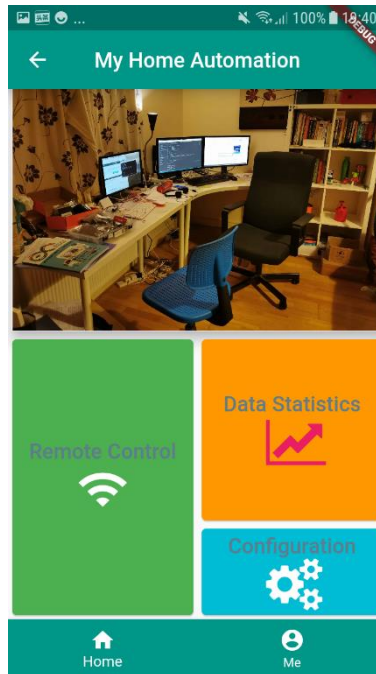
### 9.2.1. Flutter App

As a part of the software asset, the screencasts of a flutter mobile app are shown below.

9.2.1.1. *Screencasts*

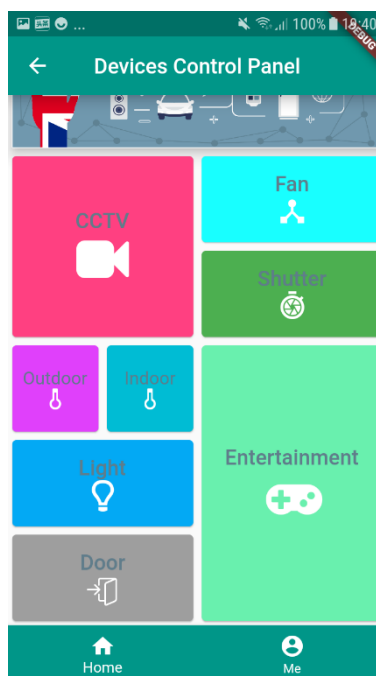
9.2.1.1.1. *Navigator Screens*

1) Home Screen



*Figure 22 home screen*

2) Devices Control Panel Screen



*Figure 23 devices control panel screen*

3) Network Options Screen

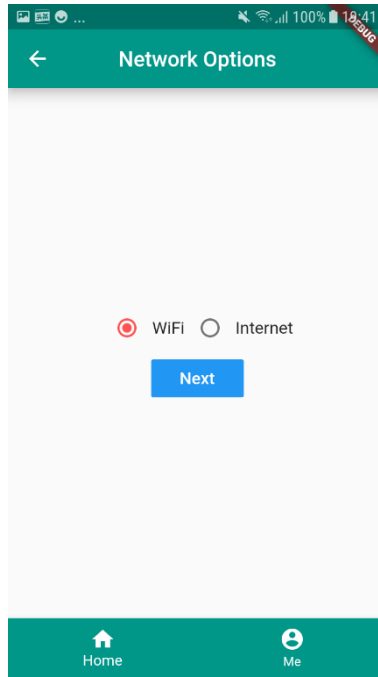


Figure 24 network options

#### 9.2.1.1.2. Home Appliances Control

##### 1) Light Control Panel

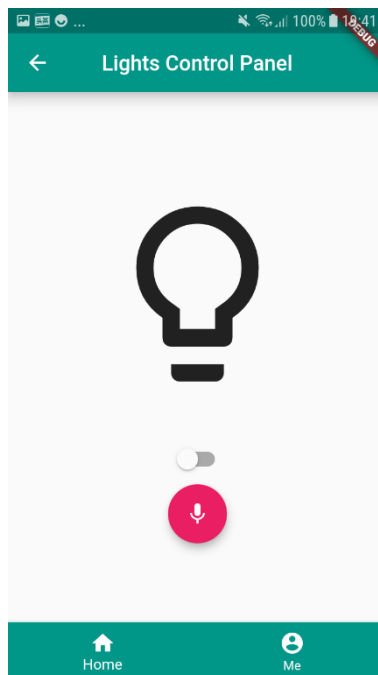


Figure 25 light control panel 1

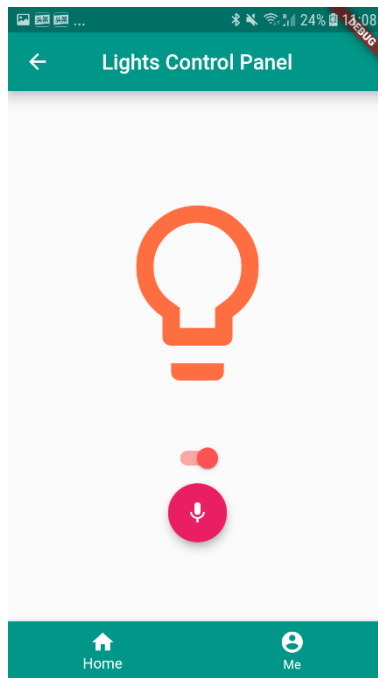


Figure 26 light control panel 2

## 2) Fan Control Panel Screen

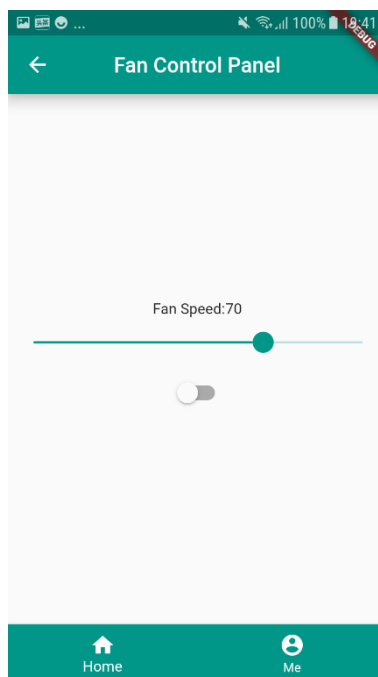


Figure 27 fan control panel 1

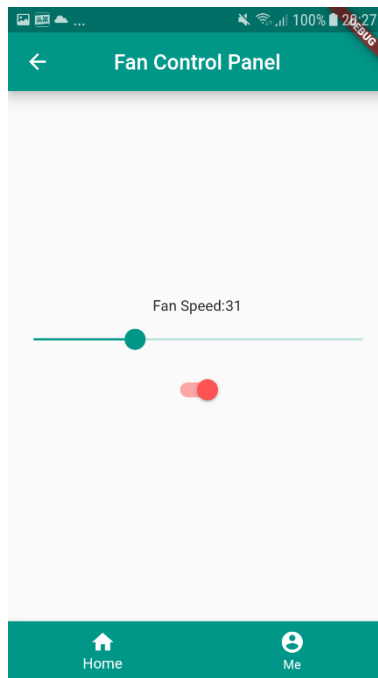


Figure 28 fan control panel 2

### 3) Shutter Control Panel Screen

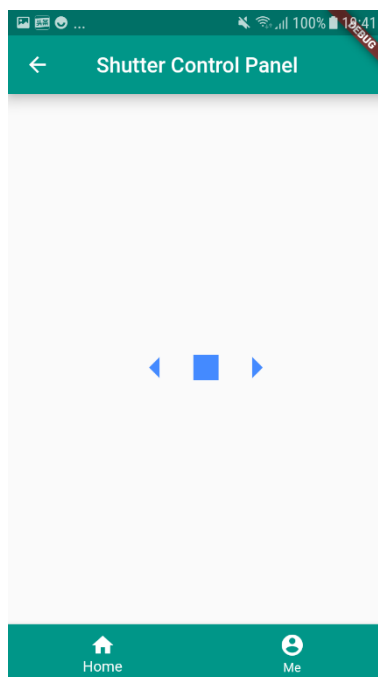


Figure 29 shutter control panel

#### 9.2.1.1.3. Security Monitoring

##### 1) CCTV Control Panel Screen



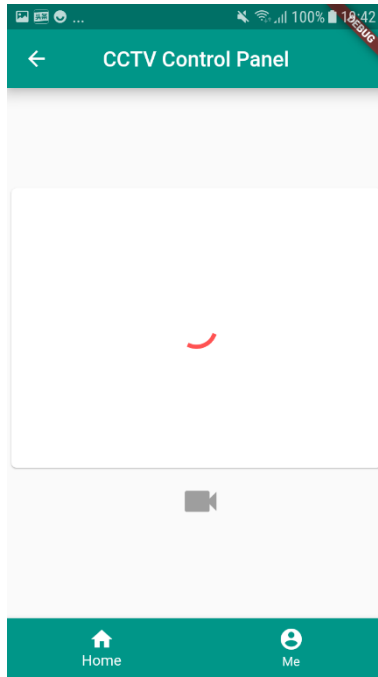


Figure 30 cctv control panel 1

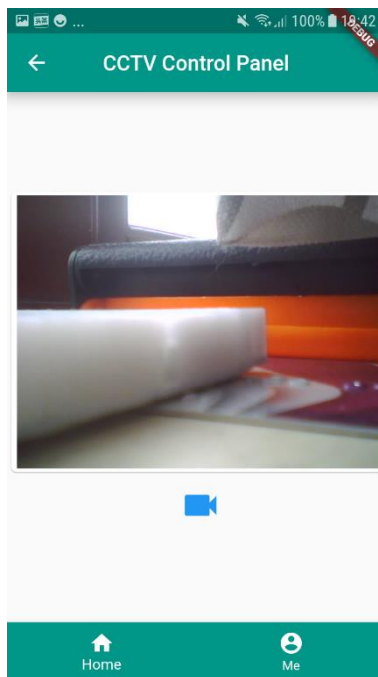


Figure 31 cctv control panel 2

## 2) Door Access Control Panel Screen

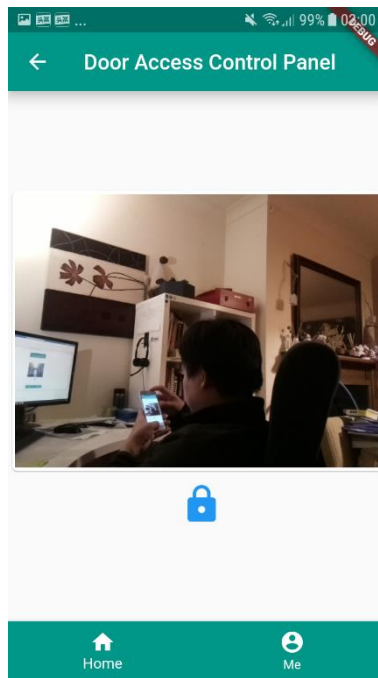


Figure 32 door access control panel 1

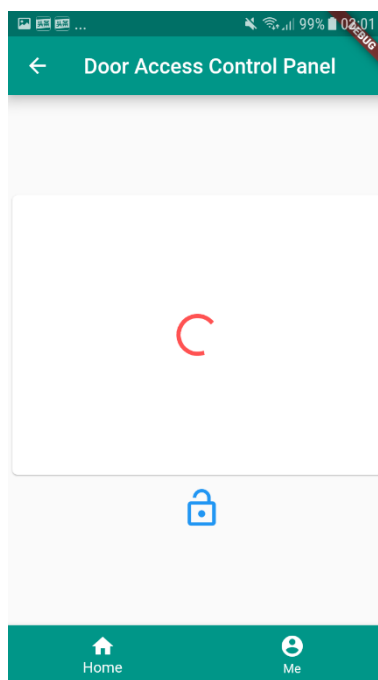


Figure 33 door access control panel 2

#### 9.2.1.1.4. Sensors Control

##### 1) Indoor Atmosphere Panel Screen

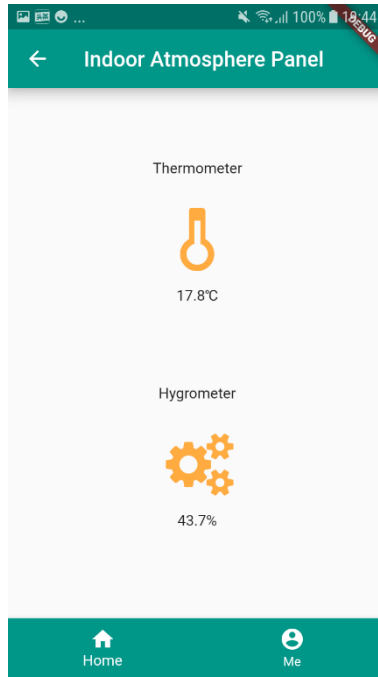


Figure 34 indoor atmosphere panel

## 2) Outdoor Atmosphere Panel Screen

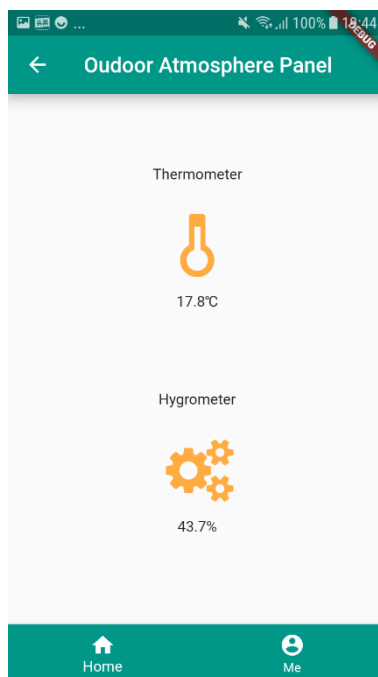


Figure 35 outdoor atmosphere panel

### 9.2.1.1.5. Entertainment

#### 1) Game Options Screen

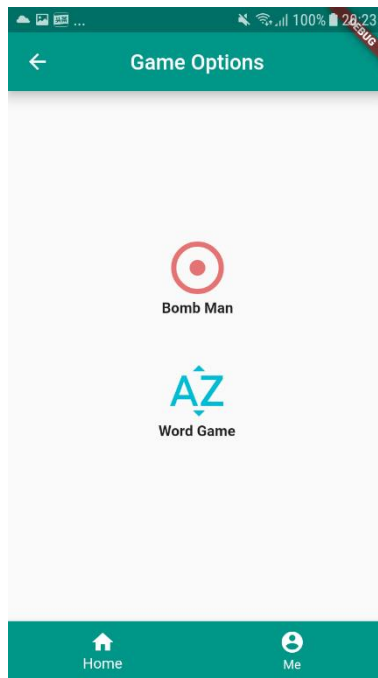


Figure 36 game option

### 9.2.2. Raspberry Pi Python Scripts

A number of python scripts that will be delivered to maintain the status of the terminal devices when they receive a signal sent by the Raspberry Pi.

#### 9.2.2.1. Terminal Device Scripts

Module Name	Device Name	Script Location
Home Appliances	Light	\$project_home/rpiboard/scripts/light/control.py
	Fan	\$project_home/rpiboard/scripts/fan/control.py
	Shutter	\$project_home/rpiboard/scripts/shutter/control.py
Security Monitoring	CCTV	\$project_home/rpiboard/scripts/cctv/control.py
	Doorbell	\$project_home/rpiboard/scripts/door/bell/control.py
	Door Camera	\$project_home/rpiboard/scripts/door/camera/control.py
Sensors Control	Door Lock	\$project_home/rpiboard/scripts/door/lock/control.py
	Temperature & Humidity Sensor	\$project_home/rpiboard/scripts/atmosphere/inddor/control.py
	Temperature & Humidity Sensor	\$project_home/rpiboard/scripts/atmosphere/outdoor/control.py
	Temperature & Humidity Sensor	\$project_home/rpiboard/scripts/atmosphere/atmosphereData.py

#### 9.2.2.2. Entry & Schedule Scripts

Name	Purpose	Script Location
------	---------	-----------------

controller.py	Main entry	\$project_home/rpiboard/controller.py
localMain.py	Scheduled on local home network	\$project_home/rpiboard/localMain.py
cloudMain.py	Scheduled on between Internet and local home network	\$project_home/rpiboard/cloudMain.py

### 9.2.2.3. Global Configuration Scripts

Name	Script Location
config.py	\$project_home/rpiboard/config.py

## 10. Not Yet Achieved

### 10.1. Unit Test

Due to lacking the unit tests that cause many problems after I have several times refactor my entire project at the end of the development phase. A number of the new bugs are introduced when I do the refactor code, which will further affect my confidence in the entire project. Because of I found the issue at the end of development phase, so I do not have much time to consider that write the unit test for several core classes.

So if the unit test is done from the beginning should be good for the system delivery in time at high quality. Meanwhile, the TDD is an abbreviation for Test Driven Development is worth trying.

### 10.2. Entertainment Module Implementation

Because of the overall evaluation of the whole project at the beginning of the project in which the estimated cycle of the project is not accurate. At the same time, it took too long to resolve unforeseen problems during the project. All of those have caused this module in the project to not be completed in time.

So, if the tasks of the whole project that break down into the size enough small at the beginning of the project then this issue that I met should be reduced slightly

## 11. Deviation

There is a deviation in this project that according to the original design, the router should be a bridge between the Raspberry Pi and AWS ActiveMQ that transfer the message. The diagram explains below.

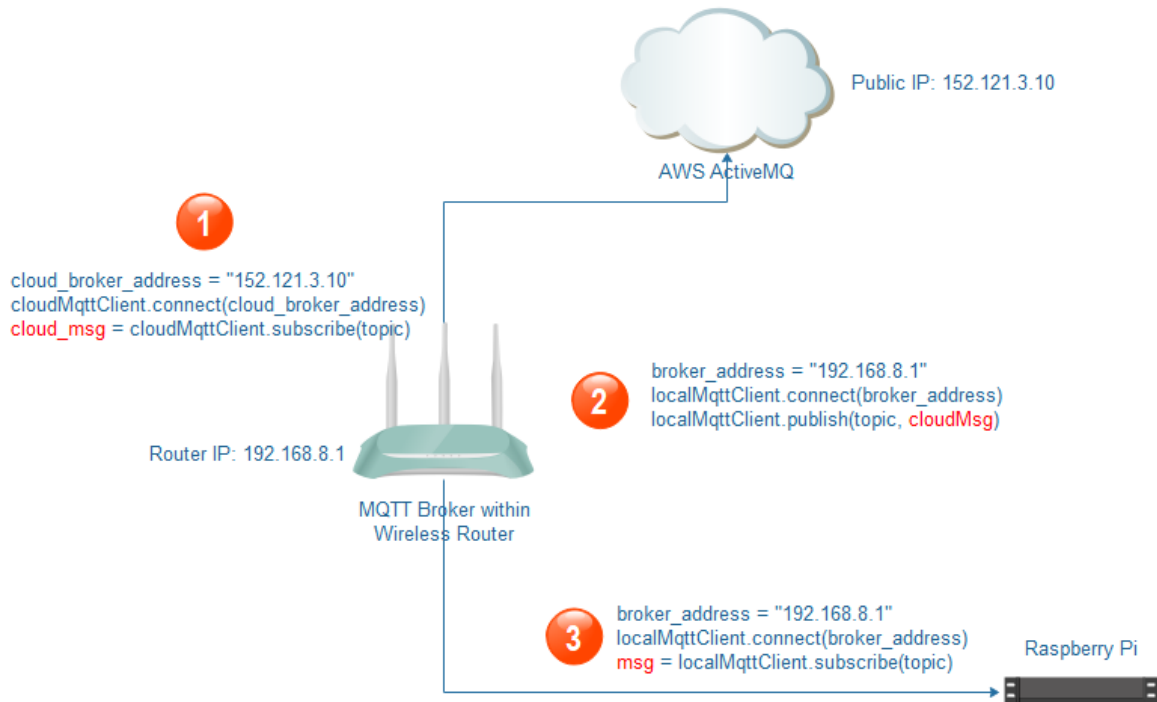


Figure 37 original design

But currently, the implementation of the project just not strictly conform to the design. In principle, the idea of the design should be better than the current implementation. Because when a new Raspberry Pi is added into the home local network to control other devices, both Raspberry Pi are in an unequal position, the new Raspberry Pi as a client-side to communicate with another one which as a server-side.

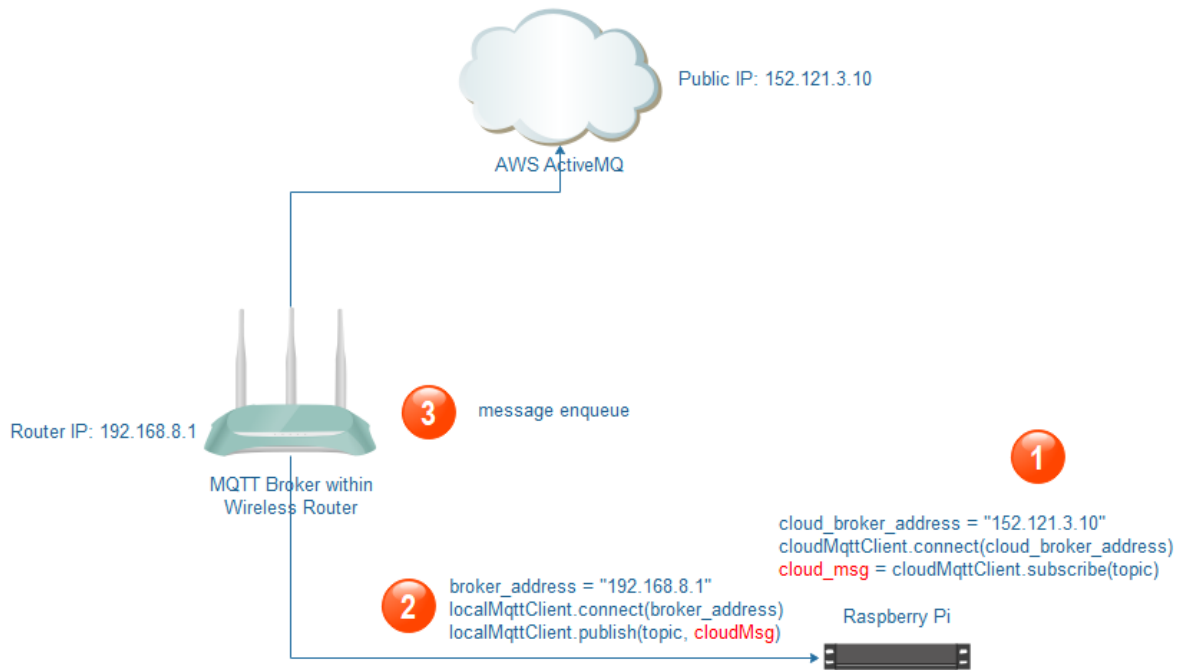


Figure 38 a real implementation

## 12. Process Methodology

The Scrum is a software development process approach used throughout my whole project. In this process approach, I almost had 2 sprints for every week that ensure I had delivery each week. Further ensure that when I met my supervisor, I had some stuff can demo with him every week, so I can get feedback and solution in time from my supervisor. Meanwhile, that ensures my supervisor can know the progress of the project in which I am doing.

After each of the sprint, I would like to refine the requirement and then put them into the backlogs. Once the new sprint will be launched that one of the backlogs would be picked up with the high priority and assign the points to it which the points to decide the length of this sprint.

There is a template that describes the user story is used for this project.

**As a** [role],  
**I want to** [do something]  
**So that** [can get benefit]

The user story for each sprint shown in diagram below.

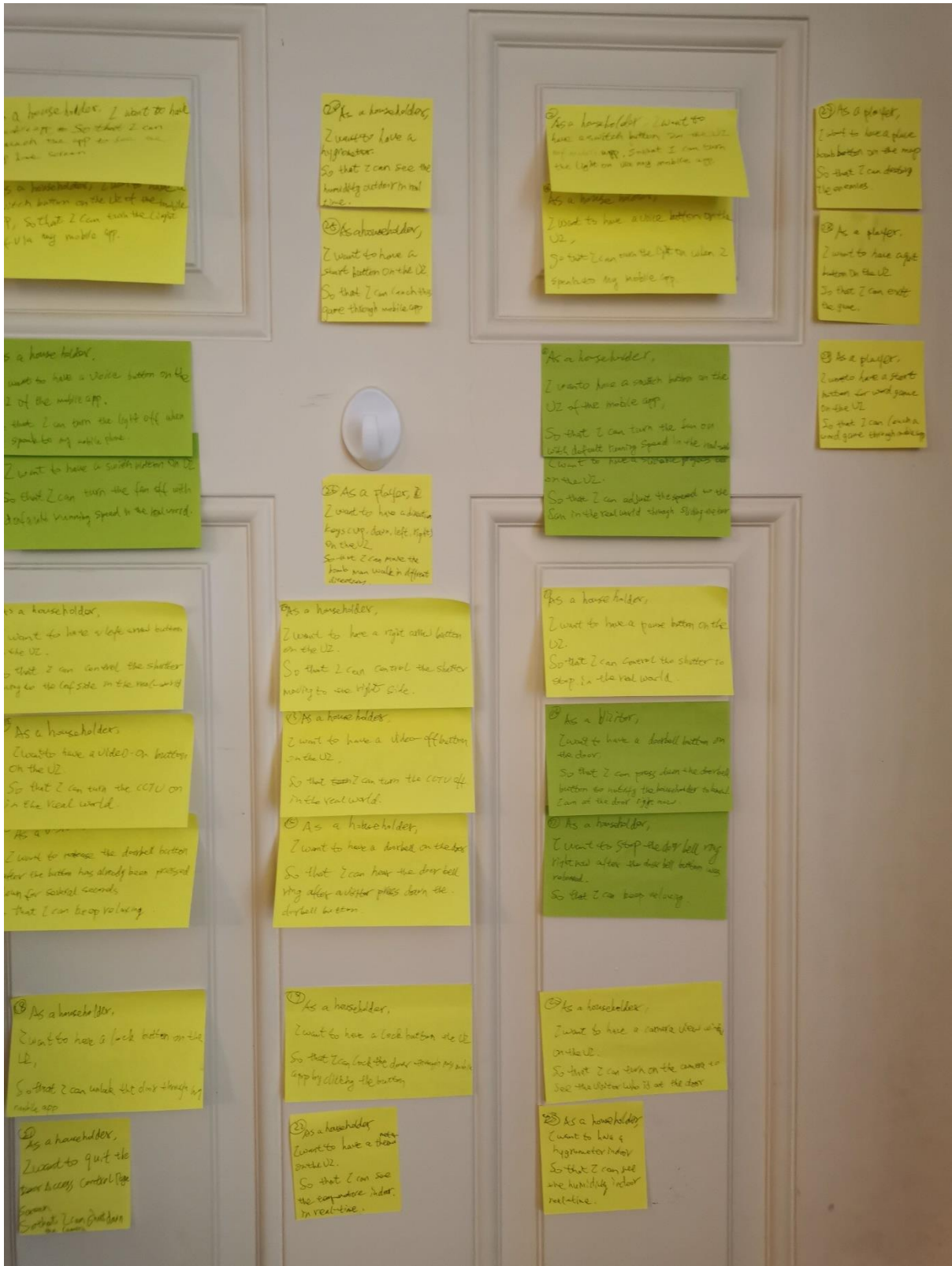


Figure 39 user story wall



### 13. Milestone

In this project implementation, each delivery considered a milestone. Therefore, there are 13 milestones totally. The delivery as a sign that means the stuff has already demo with my supervisor.

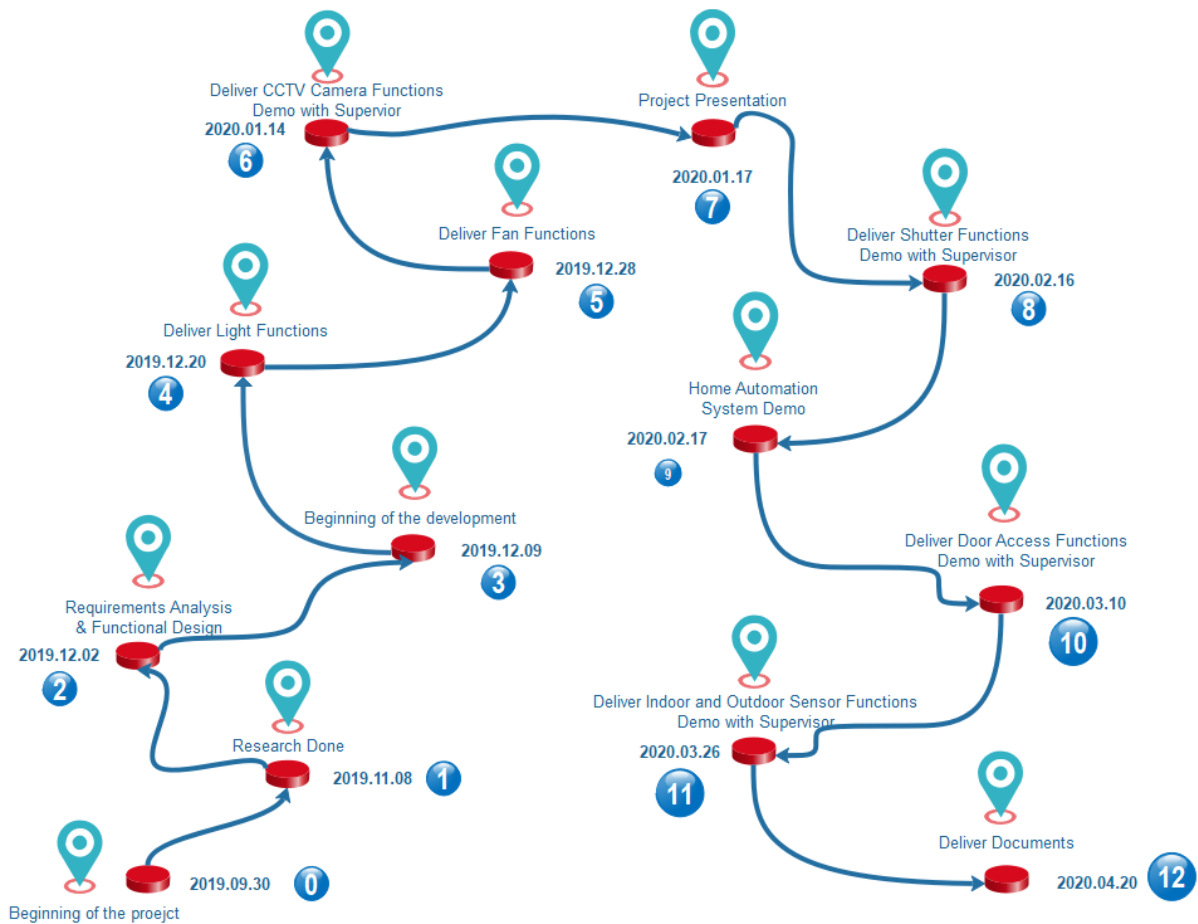


Figure 40 milestone

#### Description

No.	Title	Date	Description
0	Beginning of the project	2019.09.30	Start date
1	Research Report	2019.11.08	Research report submission
2	Requirement & Functional	2019.12.02	Requirements analysis & Functional Design
3	Development	2019.12.09	Start the development of the project
4	Deliver Light Functions	2019.12.20	Complete the Light remotely control functions
5	Deliver Fan Functions	2019.12.28	Complete the Fan remotely control functions
6	Deliver CCTV Functions	2020.01.03	Complete the CCTV camera remotely control functions
7	Project Presentation	2020.01.17	Presentation in UNUM
8	Delivery Shutter Functions	2020.02.16	Demo the shutter with supervisor
9	Home automation system demo	2020.02.17	Demo the system with lecturers in UNUM
10	Deliver Door Access Functions	2020.03.10	Demo the Door Access Functions with supervisor

11	Deliver Indoor and Outdoor Sensor Functions	2020.03.26	Send the video to my supervisor for demo
12	Deliver Documents	2020.4.20	Submission the documents and build a website

## 14. Problems Encountered

During the development of the project, there are problems I encountered listed below.

1. In the home local network, the mobile app interacts to the Raspberry Pi, how to solve the communication issue?  
To add a router that connects the home local network, and this router should support the OpenWRT firmware to let the developer to login via SSH, then installing the software that the system required. Such as MQTT broker.
2. Read the camera live data stream and then shown the stream as a video on the mobile app screen.  
To install a program called Mjpeg-streamer as a command on the Raspberry Pi that can be executed on the command line of the Raspberry Pi.
3. How to communicate across the Internet when mobile app tends to send a message to the Raspberry Pi?  
In this project, I used the AWS ActiveMQ as a cloud broker. Due to the AWS ActiveMQ has already supported the MQTT protocol, so I did not need to do any changes to communicate to the AWS ActiveMQ broker.
4. The home router has a public IP address but never used this public IP address as an address to communicate with other devices that have not existed in home local networks. Because this public IP address is dynamical got from ISP. If you would like to get a fixed public IP address to your home router, communicate with your ISP is required. In this project, the system did not use the public IP address provided by ISP.

## 15. Further Work

To ensure this Home Automation System I implemented to friendly interact with, and smarter to service the householder, currently implemented system should be improved in the future. I think that at least 3 parts below should finish later.

### 15.1. Device Configuration Subsystem

This subsystem is a foundation of whole Home Automation System. It provides a friendly configuration system that lets the householder add their own home appliances, sensors, security monitoring equipment that the householder wants to join dynamically. The devices are added through this configuration subsystem at least persistence the information that seems like below.

1. Setup the House information.
2. Setup Rooms in which house.

3. Setup Devices in which Room.
4. Setup Device parameters
  - a. Name
  - b. Brand
  - c. Watts
  - d. IP
  - e. etc.
5. Track and collect each device behaviours.
6. If it requires, upload the data to cloud.
7. Monitoring each device status.

## 15.2. Data Statistical Analysis Subsystem

This subsystem to assist the householder to manage the devices more conveniently and more reasonably. It provides data visualization functions to keep the householder more intuitive understanding of home appliances or other devices. Here, I would like to advise some recommendations shown below.

1. Given the Devices energy used data visualization, such as:

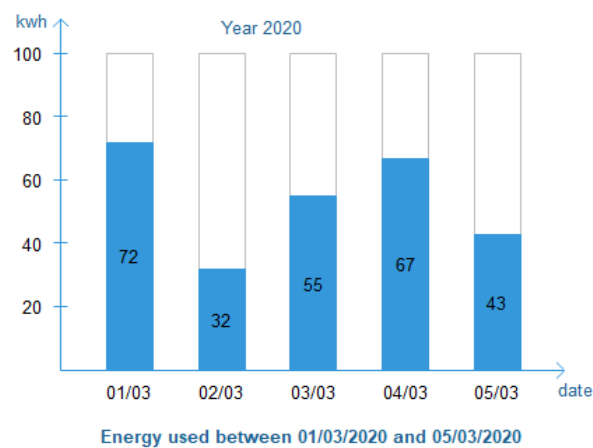
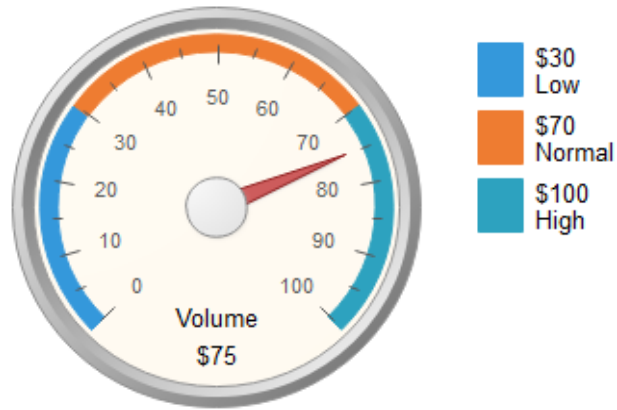


Figure 41 visualization 1

2. Setup the threshold and alarms if it meets

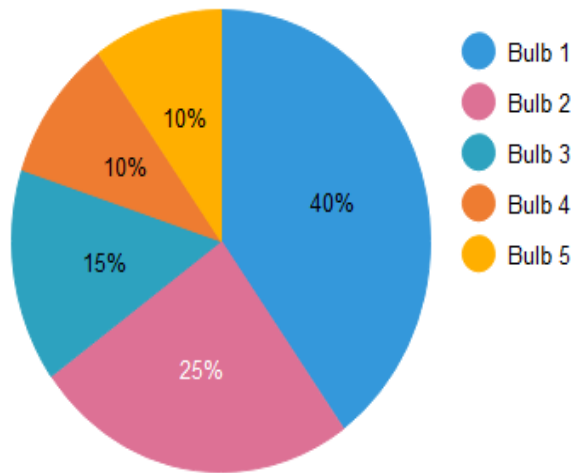


**Threshold Alarms**

*Figure 42 visualization 2*

3. Comparison between home appliances

April 2020

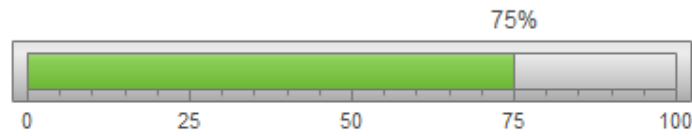


**Comparison between Bulbs**

*Figure 43 visualization 3*

4. Rewards if the householder more save energy

### Energy Used in April 2020



### How many users did you beat?

Figure 44 visualization 4

5. Given some recommendations, for example:  
“**80%** of householders choose **Samsung** refrigerators to save **€20** per month compared to other brands”.
6. Send periodically the statistics and analysis data to mobile app, if the householder subscribed.

### 15.3. The Router as a Bridge

The router as a bridge transfer the messages between the Raspberry Pi and AWS ActiveMQ to conform strictly the idea of design. That idea has already described in the Deviation.

## 16. Challenges

Through this project studied there are four aspects challenges I met:

1. The underlying hardware circuit working principle.
2. The home appliance industry standards are not uniform.
3. There are many communication protocols that need to be considered for compatibility.
4. Protect user privacy and information security.

## Acknowledgement

I have to thank my supervisor, Dr. Paul Barry, for the patient guidance, and always encourage me. Gave a lot of help and freedom in the project I studied. The happiest thing I do during this project is to meet with him every week to tell him the recent results of my project's research and development and give a demo.

Finally, I would like to thank the lecturers who taught me in the four years of studying at Institute of Technology Carlow, and my beloved family who are far away in China.

# Plagiarism Declaration

## Declaration

- I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.
- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.
- I have provided a complete bibliography of all works and sources used in the preparation of this submission.
- I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offense.

**Student Name:** XIAOHUI LING

**Student Number:** C00212235

**Signature:**

**X**   
XIAOHUI LING  
Student

Date: 20/04/2020