Bachelor of Science (Honours)  
Software Development  

Home Automation System  
Design Specification  

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

This purpose of this document is to provide a detail of the system design of the Home Automation System. The system architecture and functional design outlined as core part on the system and describe each function in detail. This document first presents the architecture of Home Automation System and then given out a structure of whole system functions that presents the key modules.
Contents

1. Introduction ................................................................................................................. 6

2. Overview ....................................................................................................................... 6
   2.1. Purpose ..................................................................................................................... 6
   2.2. Goal .......................................................................................................................... 6

3. System Architecture .................................................................................................. 6
   3.1. IoT Three Layers Architecture ............................................................................. 6
      3.1.1. Overview ........................................................................................................... 6
      3.1.2. Sensing Layer ................................................................................................... 7
      3.1.3. Network Layer ................................................................................................ 12
      3.1.4. Application Layer ............................................................................................. 13
   3.2. Topology Architecture ......................................................................................... 14

4. System Design ............................................................................................................ 15
   4.1. Modules and Functions Layout .............................................................................. 15
   4.2. UI Components Structure ..................................................................................... 16
   4.3. Device Control Structure ....................................................................................... 17
   4.4. Networks Communication ..................................................................................... 18
      4.4.1. Wi-Fi ................................................................................................................ 18
      4.4.2. MQTT ................................................................................................................. 18
      4.4.3. Internet ............................................................................................................. 19

5. Modules Design .......................................................................................................... 19
   5.1. Appliances Control ................................................................................................. 19
      5.1.1. Light .................................................................................................................. 19
      5.1.2. Fan ..................................................................................................................... 30
      5.1.3. Shutter ............................................................................................................... 35
   5.2. Security Monitoring ............................................................................................... 41
      5.2.1. CCTV .................................................................................................................. 41
      5.2.2. Door Access ....................................................................................................... 46
   5.3. Sensors Control ...................................................................................................... 54
      5.3.1. Temperature ...................................................................................................... 54
      5.3.2. Humidity .......................................................................................................... 60
Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IoT architecture</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>temperature &amp; humidity sensor</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>DC Motor Driver 1</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>DC Motor Driver 2</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Raspberry Pi Camera</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>USB Camera</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>Relay Module</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>Fan DC Motor</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>Shutter DC Motor 2</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>Buzzer</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>Push Button</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>Table Light</td>
<td>11</td>
</tr>
<tr>
<td>13</td>
<td>Bulb</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>E-Lock</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td>Home Router</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>Raspberry Pi 4</td>
<td>13</td>
</tr>
<tr>
<td>17</td>
<td>MQTT Clients &amp; Broker</td>
<td>14</td>
</tr>
<tr>
<td>18</td>
<td>Topology Architecture</td>
<td>15</td>
</tr>
<tr>
<td>19</td>
<td>Modules &amp; Functions Layout</td>
<td>16</td>
</tr>
<tr>
<td>20</td>
<td>UI Components Structure</td>
<td>17</td>
</tr>
<tr>
<td>21</td>
<td>Device Control Scripts Structure</td>
<td>18</td>
</tr>
<tr>
<td>22</td>
<td>Light Use Case</td>
<td>20</td>
</tr>
<tr>
<td>23</td>
<td>Light Class Diagram</td>
<td>21</td>
</tr>
<tr>
<td>24</td>
<td>Light Hardware Design</td>
<td>22</td>
</tr>
<tr>
<td>25</td>
<td>Light Python Scripts Class Diagram</td>
<td>23</td>
</tr>
<tr>
<td>26</td>
<td>Turn Light On/Off Sequence Diagram 1</td>
<td>25</td>
</tr>
<tr>
<td>27</td>
<td>Turn Light On/Off Sequence Diagram 2</td>
<td>26</td>
</tr>
<tr>
<td>28</td>
<td>Turn Light On/Off Sequence Diagram 3</td>
<td>28</td>
</tr>
<tr>
<td>29</td>
<td>Turn Light On/Off Sequence Diagram 4</td>
<td>29</td>
</tr>
<tr>
<td>30</td>
<td>Fan Use Case</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>Fan Class Diagram</td>
<td>31</td>
</tr>
<tr>
<td>32</td>
<td>Fan Hardware Design</td>
<td>32</td>
</tr>
<tr>
<td>33</td>
<td>Fan Python Scripts Class Diagram</td>
<td>33</td>
</tr>
<tr>
<td>34</td>
<td>Turn Fan On/Off Sequence Diagram</td>
<td>34</td>
</tr>
<tr>
<td>35</td>
<td>Adjust Fan Speed Sequence Diagram</td>
<td>35</td>
</tr>
<tr>
<td>36</td>
<td>Shutter Use Case</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>Shutter Class Diagram</td>
<td>37</td>
</tr>
<tr>
<td>38</td>
<td>Shutter Hardware Design</td>
<td>38</td>
</tr>
<tr>
<td>39</td>
<td>Shutter Python Scripts Class Diagram</td>
<td>39</td>
</tr>
<tr>
<td>40</td>
<td>Move Shutter to Left/Right Sequence Diagram</td>
<td>40</td>
</tr>
<tr>
<td>41</td>
<td>Pause Shutter Class Diagram</td>
<td>41</td>
</tr>
<tr>
<td>42</td>
<td>CCTV Use Case</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>CCTV Class Diagram</td>
<td>43</td>
</tr>
<tr>
<td>44</td>
<td>CCTV Hardware Design</td>
<td>44</td>
</tr>
<tr>
<td>45</td>
<td>CCTV Python Scripts Class Diagram</td>
<td>44</td>
</tr>
<tr>
<td>46</td>
<td>Turn CCTV On/Off Sequence Diagram</td>
<td>46</td>
</tr>
</tbody>
</table>
Figure 47 Door Access Use Case ................................................................. 47
Figure 48 Door Access Class Diagram ....................................................... 48
Figure 49 Door Access Hardware Design .................................................... 49
Figure 50 Door Access Python Scripts Class Diagram .................................. 50
Figure 51 Doorbell Sequence Diagram ....................................................... 51
Figure 52 Door Lock Sequence Diagram .................................................... 52
Figure 53 Door Camera Sequence Diagram ................................................ 53
Figure 54 Temperature Sensor Use Case ..................................................... 54
Figure 55 Temperature & Humidity Sensor Class Diagram ............................ 55
Figure 56 Temperature & Humidity Sensor Hardware Design ......................... 56
Figure 57 Temperature & Humidity Sensor Scripts Class Diagram .................. 57
Figure 58 Indoor Temperature & Humidity Sensor Sequence Diagram ............. 58
Figure 59 Outdoor Temperature & Humidity Sensor Sequence Diagram .......... 60
Figure 60 Humidity Sensor Use Case .......................................................... 61
1. Introduction
Home automation as an important application of the IoT field has been constantly moving forward. In life, people can also easily buy smart devices and control them remotely through the App installed on the mobile phone. Meanwhile, there are many open-source home automation systems constantly emerging. However, these existing products or open source systems cannot fully meet the personalized customization needs of different users.

Therefore, I hope to develop a home automation system that is closer to people’s lives. So, this project carried out preliminary research on home automation, and step by step to realize a system prototype involving management and control included appliances, security monitoring, sensors and entertainment.

2. Overview

2.1. Purpose
The purpose of the home automation system design is to break down the whole system into different modules and functions and describe them in detail conduct how the system should be implemented by the development stage.

2.2. Goal
The home automation system is designed used to remote control involved fields that included appliances, sensors, security monitoring, entertainment, through networks. This document describes the system architecture and system module design details, explains the system components and working principles.

3. System Architecture

3.1. IoT Three Layers Architecture

3.1.1. Overview
The three layers architecture is a classic design in the IoT field that simply describes how it works in its own responsibility of between different components.
3.1.2. Sensing Layer

3.1.2.1. MEMS
MEMS is an acronym for Micro-electromechanical System that is one of the critical techniques that was applied in various sensors or smart devices in the IoT field. The advantages are a microchip and circuit they have. In this project, the temperature and humidity sensor, DC motor driver controller and cameras are used.

- Temperature & Humidity Sensor
This sensor used to monitor the environment to obtain temperature and humidity in real-time. It is small size and works with low power.
• DC Motor Driver Speed Controller

The DC motor driver speed controller can convert each other between the analog signal and digit signal so that they can read and write data from the equipment.

![Figure 3 DC Motor Driver 1](image1)

![Figure 4 DC Motor Driver 2](image2)

• RaspberryPi Camera

This is a raspberry Pi camera V2 in the project. It has 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images and supports high resolution video.
- **USB Camera**

This is a Microsoft Lifecam NX-3000 USB camera used in this project. It could be replaced with any type of USB camera.

![USB Camera](image)

### 3.1.2.2. Electronic Components

Those components have a circuit control system that is controlled through the analog signal. The way they work based on electromagnetic induction and mechanical principles.

- **Relay Module**

It can control the home appliances turn on or off through the 5 to 12 voltage that powered by battery or raspberry Pi. Please read the Wikipedia if you would like to understand how the relay module works.
• DC Motor

The DC motors work with 5 to 12 voltage that the maximum of RMP is 15,000. They can be controlled by the DC motor driver controller which is MEMS.

• Buzzer

As a doorbell that is the low-level signal trigger in door access system.
• Push Button
As a doorbell push button in door access system. It is a switch. It is switched on if it is pressed down then.

• Desk Lamp
As a home appliance, it is turned on or off when a high-level signal is received through the relay module.
E-Lock

As a lock in door access system, it is locked or unlocked when a high-level signal is received through the relay module.

3.1.3. Network Layer

3.1.3.1. Wi-Fi

Wi-Fi is a local network communication protocol that takes advantage of the 802.11 standards to define service, clients, access points. It supports the maximum number of the client nodes is 32. The Wi-Fi is used for communication between IoT devices in this project.

3.1.3.2. Router

As a gateway that is a networking device, it forwards data packets between networks. It also provides the Wi-Fi feature which is local networks. In this project, it is necessary to support the OpenWRT that is an open-source project for an embedded operating system based on Linux used to route network traffic and we can login into it via SSH to install the software which I want to.

Here I am using a GL.iNET GL-MT300N-V2 Mini Travel Router, any router support OpenWRT can replace it in the project.
3.1.4. Application Layer

3.1.4.1. Raspberry Pi 4B
It is an open-source single board hardware platform base on the Linux operating system that easily extends to develop a device’s control system in the IoT field through the GPIO feature is provided. In this project, the Raspberry Pi is a device control centre to manage the IoT devices through Python GPIO feature.

3.1.4.2. Mobile Phone
The mobile phone as a terminal control device that remotely manages the IoT devices through a mobile app developed in this project. It could be an Android or Apple iOS operating system within the mobile phone, however, that just is tested with the Android operating system in this project so far.

Currently, I am using the brand of mobile phone that is Samsung SM-A320FL base on the Android 7.0 version. It can be replaced with the same Android version of any mobile phone.

3.1.4.3. Mobile App
It is an app was developed by Google Flutter mobile app development framework that used Dart programming language. Once the app development is complete, it can be easily deployed on both Android and iOS mobile operating systems. The app mainly provides features through the network to remotely control the IoT devices and obtain environment data both from the sensors.
3.1.4.4. **MQTT**
It is a lightweight message queue mid-ware applied in the system. It was developed by IBM that widely used in the IoT field to solve the problems of real-time communication between devices. It consists of MQTT Client and Broker. The MQTT client could publish the message to or subscribe to the message from the broker.

In this project, the flutter app installed in the mobile phone and Raspberry Pi both are MQTT client and the router is a MQTT broker.

![MQTT Clients & Broker](image)

3.1.4.5. **AWS ActiveMQ**
AWS ActiveMQ is a cloud service provided by Amazon corporate company. It bases on the open source ‘s Apache ActiveMQ used to communicate and exchange information between different systems. Its function is like MQTT and it also compatible with MQTT as a broker, so it as a service is used to communicate across the internet between devices in this project.

3.2. **Topology Architecture**
In the whole project, there are four components are mainly involved that listed below.

1. The smart phone as a control terminal has already installed an app that was developed base on Flutter mobile framework.
2. The AWS cloud provides the ActiveMQ service as a broker to solve the problems of across internet real-time communication between mobile phone and Raspberry Pi.
3. The wireless router in the home provides a local network service using Wi-Fi protocol between mobile phone and Raspberry Pi and it also as a gateway to communicate to the internet.
4. The Raspberry Pi as a control centre that manages and controls the devices which are home appliances, monitoring equipment, sensors and etc.
4. System Design

4.1. Modules and Functions Layout
The modules and functions of each module are listed below. It will be described in detail later in the document.
4.2. UI Components Structure
The mobile app UI components and structure listed below. The widget is a concept in Flutter mobile framework that is abstracted as an object which is corresponding to a device control panel in this project.
4.3. Device Control Structure

The Raspberry Pi as a device control centre has a controller to manage the terminal devices to communicate to a mobile app through the network. Those were implemented by python scripts.
4.4. Networks Communication

4.4.1. 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 project that is not too matter.

4.4.2. 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 can 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.
4.4.3. 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.

5. Modules Design
In this chapter, it describes the functions of each module in detail to explain how the function works through the design planning.

5.1. Appliances Control
There are 3 types of home appliances are applied in the project, light, fan and shutter that will be described in the next sections.

5.1.1. Light
This section described how to control the light by the mobile app through both local network and internet.

5.1.1.1. Use Case
The use case diagram below is clearly shown the process of the light control between the mobile phone, Raspberry Pi.
5.1.1.2. Class Diagram
The light module classes are described using UML class diagram below.
5.1.1.3. Wiring Diagram

The diagram described how the components connect to each other. There is useful information provided below.
1. Raspberry Pi and Relay Module

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Relay Module</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>5v Pin</td>
<td>Positive Pin</td>
<td>Red</td>
</tr>
<tr>
<td>GND Pin</td>
<td>Negative Pin</td>
<td>Grey</td>
</tr>
<tr>
<td>BCM 21</td>
<td>I/O Pin</td>
<td>Green</td>
</tr>
</tbody>
</table>

2. Relay Module and Light

<table>
<thead>
<tr>
<th>Relay Module</th>
<th>Light</th>
<th>Charge of Light</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Open Pin</td>
<td>Positive of Charge</td>
<td>Not Used</td>
<td>Red</td>
</tr>
<tr>
<td>COM Pin</td>
<td>Not Used</td>
<td>Positive of Charge</td>
<td>Red</td>
</tr>
</tbody>
</table>

5.1.1.4. Device Control Script

There are Python scripts implements to control the Light device through the Raspberry Pi. The Python script is described using class diagram below.
5.1.1.5. Functional Description

In this section, it will describe that functions design of the light in detail.

5.1.1.5.1. Turn Light On/Off via Wi-Fi

- **Description**
  The householder could click the button on the mobile phone screen to remotely control the light to turn on or off through local network.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>switch button on</td>
<td>Wi-Fi</td>
<td>publish</td>
<td>topic/light/control</td>
<td>on</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/light/control</td>
<td>[not applicable]</td>
<td>on</td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>switch button off</td>
<td></td>
<td>publish</td>
<td>topic/light/control</td>
<td>off</td>
<td>not applicable</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/light/control</td>
<td>[not applicable]</td>
<td>off</td>
</tr>
</tbody>
</table>

- **Sequence Diagram**
5.1.1.5.2. Turn Light On/Off via Internet

- **Description**
  The householder could click the button on the mobile phone screen to remotely control the light turn on or off between local network and Internet.

- **Key Parameters**
<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>switch button on</td>
<td>Internet</td>
<td>publish</td>
<td>topic/light/control</td>
<td>on</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/light/control</td>
<td>[not applicable]</td>
<td>on</td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>switch button off</td>
<td></td>
<td>publish</td>
<td>topic/light/control</td>
<td>off</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/light/control</td>
<td>[not applicable]</td>
<td>off</td>
</tr>
</tbody>
</table>

- **Sequence Diagram**
Figure 27 Turn Light On/Off Sequence Diagram 2
5.1.1.5.3. Turn Light On/Off with Voice via Wi-Fi

- **Description**
The householder could speak to the mobile phone to remotely control the light turn on or off between local network. The keywords are ‘light on’ or ‘light off’ should be spoke to mobile phone.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Said ‘light on’ to phone</td>
<td>Wi-Fi</td>
<td>publish</td>
<td>topic/light/control</td>
<td>on</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Said ‘light off’ to phone</td>
<td></td>
<td>publish</td>
<td>topic/light/control</td>
<td>off</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/light/control</td>
<td></td>
<td>off</td>
</tr>
</tbody>
</table>

- **Sequence Diagram**
5.1.1.5.4. Turn Light On/Off with Voice via Internet

- **Description**
The householder could speak to the mobile phone to remotely control the light turn on or off through internet. The keywords are ‘light on’ or ‘light off’ should be spoke to mobile phone.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Said ‘light on’ to phone</td>
<td>Internet</td>
<td>publish</td>
<td>topic/light/control</td>
<td>on</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td>subscribe</td>
<td>topic/light/control</td>
<td>[not applicable]</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Said 'light off'</td>
<td>publish</td>
<td>topic/light/control</td>
<td>off</td>
<td>[not applicable]</td>
<td></td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td>subscribe</td>
<td>topic/light/control</td>
<td>[not applicable]</td>
<td>off</td>
<td></td>
</tr>
</tbody>
</table>

**Sequence Diagram**

![Sequence Diagram](image)

*Figure 29 Turn Light On/Off Sequence Diagram 4*
5.1.2. Fan
The Fan module that manages their status is to turn on, turn off and adjust running speed.

5.1.2.1. Use Case
The use case diagram below is clearly shown the process of the Fan control between the Mobile app, Raspberry Pi.

5.1.2.2. Class Diagram
The Fan module classes are described using UML class diagram below.
5.1.2.3. Wiring Diagram

The diagram described how the components connect to each other. There is useful information provided below.
1. Raspberry Pi and Motor Driver Module

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Motor Driver Module</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V Pin</td>
<td>Positive Pin</td>
<td>Red</td>
</tr>
<tr>
<td>GND Pin</td>
<td>Negative Pin</td>
<td>Grey</td>
</tr>
<tr>
<td>BCM 12</td>
<td>In2 Pin</td>
<td>Green</td>
</tr>
</tbody>
</table>

2. Motor Driver Module and Fan Motor

<table>
<thead>
<tr>
<th>Motor Driver Module</th>
<th>Fan Motor</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Pin in Slot</td>
<td>Positive</td>
<td>Red</td>
</tr>
<tr>
<td>Fourth Pin Slot</td>
<td>Negative</td>
<td>Grey</td>
</tr>
</tbody>
</table>

5.1.2.4. Device Control Script

There are Python scripts implements to control the Light device through the Raspberry Pi. The Python script is described using class diagram below.
5.1.2.5. Functional Description
In this section, it will describe that functions design of the Fan in detail.

5.1.2.5.1. Turn Fan On/Off

- **Description**
The householder could click the button on the mobile phone screen to remotely control the fan to turn on or off through local network.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>switch button on</td>
<td>WI-FI</td>
<td>publish</td>
<td>topic/fan/control</td>
<td>on</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/fan/control</td>
<td>[not applicable] on</td>
<td></td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>switch button off</td>
<td></td>
<td>publish</td>
<td>topic/fan/control</td>
<td>off</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/fan/control</td>
<td>[not applicable] off</td>
<td></td>
</tr>
</tbody>
</table>

- **Sequence Diagram**
5.1.2.5.2. Adjust Fan Speed

- **Description**
The householder could slide the progressing bar on the mobile phone screen to remotely control the fan running speed through local network.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
</table>

*Figure 34 Turn Fan On/Off Sequence Diagram*
5.1.3. Shutter
The Shutter module that manages their directions to move and pause itself.
5.1.3.1. Use Case
The use case diagram below is clearly shown the process of the Shutter control between the Mobile phone, Raspberry Pi.

![Use Case Diagram](image)

*Figure 36 Shutter Use Case*

5.1.3.2. Class Diagram
The Shuter module classes are described using UML class diagram below.
5.1.3.3. Wiring Diagram

The diagram described how the components connect to each other. There is useful information provided below.
1. Raspberry Pi and Motor Driver Module

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Motor Driver Module</th>
<th>Wire Colour</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5v Pin</td>
<td>Positive Pin</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>GND Pin</td>
<td>Negative Pin</td>
<td>Grey</td>
<td></td>
</tr>
<tr>
<td>BCM 4</td>
<td>DIR Pin</td>
<td>Blue</td>
<td>Control motor direction</td>
</tr>
<tr>
<td>BCM 13</td>
<td>PWM Pin</td>
<td>Green</td>
<td>Control motor speed</td>
</tr>
<tr>
<td>GND Pin</td>
<td>GND Pin</td>
<td>Grey</td>
<td></td>
</tr>
</tbody>
</table>

2. Motor Driver Module and Fan Motor

<table>
<thead>
<tr>
<th>Motor Driver Module</th>
<th>Shutter Motor</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Pin</td>
<td>Positive</td>
<td>Red</td>
</tr>
<tr>
<td>B Pin</td>
<td>Negative</td>
<td>Grey</td>
</tr>
</tbody>
</table>

5.1.3.4. Device Control Script

There are Python scripts implements to control the Shutter device through the Raspberry Pi. The Python script is described using class diagram below.
5.1.3.5. Functional Description
In this section, it will describe that functions design of the Shutter in detail.

5.1.3.5.1. Move to Left/Right

- **Description**
The householder could click the left or right arrow button on the mobile phone screen to remotely control the Shutter moving to left or right through local network.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>click the left arrow button</td>
<td>Wi-Fi</td>
<td>publish</td>
<td>topic/shutter/control</td>
<td>left</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/shutter/control</td>
<td>[not applicable]</td>
<td>left</td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>click the right arrow button</td>
<td></td>
<td>publish</td>
<td>topic/shutter/control</td>
<td>right</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/shutter/control</td>
<td>[not applicable]</td>
<td>right</td>
</tr>
</tbody>
</table>

- **Sequence Diagram**
5.1.3.5.2. Pause the Shutter

- **Description**
The householder could click pause button on the mobile phone screen to remotely control the Shutter to stop the shutter through local network.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>click the pause button</td>
<td>Wi-Fi</td>
<td>publish</td>
<td>topic/shutter/control</td>
<td>stop</td>
<td>[not applicate]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicate]</td>
<td>subscribe</td>
<td>topic/shutter/control</td>
<td>[not applicate]</td>
<td>stop</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 40 Move Shutter to Left/Right Sequence Diagram*
5.2. Security Monitoring
There are two parts in this module, one part is a CCTV to capture the live view and transfer to mobile phone. another one is a Door Access to manage the doorbell, camera, and lock that will be described in the next sections.

5.2.1. CCTV
The CCTV module that manages their status is to turn on and turn off.
5.2.1.1. Use Case
The use case diagram below is clearly shown the process of the CCTV control between the mobile phone, Raspberry Pi.

5.2.1.2. Class Diagram
The CCTV module classes are described using UML class diagram below.
5.2.1.3. Wiring Diagram

The diagram described how the components connect to each other. There is useful information provided below.
1. Raspberry Pi and USB Camera

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>USB Camera</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB Interface</td>
<td>USB Cable</td>
<td>Green</td>
</tr>
</tbody>
</table>

5.2.1.4. Device Control Script

There are Python scripts implements to control the CCTV camera through the Raspberry Pi. The Python script is described using class diagram below.

5.2.1.5. Functional Description

In this section, it will describe that functions design of the CCTV in detail.
5.2.1.5.1. Turn CCTV On/Off

- **Description**
The householder could click the button on the mobile phone screen to remotely control the CCTV camera to turn on or off through local network.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>switch button on</td>
<td></td>
<td>publish</td>
<td>topic/cctv/control</td>
<td>on</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td>Wi-Fi</td>
<td>subscribe</td>
<td>topic/cctv/control</td>
<td>[not applicable] on</td>
<td></td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>switch button off</td>
<td></td>
<td>publish</td>
<td>topic/fan/control</td>
<td>off</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/cctv/control</td>
<td>[not applicable] off</td>
<td></td>
</tr>
</tbody>
</table>

- **Sequence Diagram**
5.2.2. Door Access
The Door Access module that manages the door status when a visitor stands at the door.

5.2.2.1. Use Case
The use case diagram below is clearly shown the process of the Door Access control between the mobile phone, Raspberry Pi.
5.2.2.2. Class Diagram
The CCTV module classes are described using UML class diagram below.
5.2.2.3. Wiring Diagram
The diagram described how the components connect to each other. There is useful information provided below.
1. Raspberry Pi and Buzzer (Doorbell)

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Buzzer (Doorbell)</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3v Pin</td>
<td>VCC Pin</td>
<td>Red</td>
</tr>
<tr>
<td>GND Pin</td>
<td>GND Pin</td>
<td>Grey</td>
</tr>
<tr>
<td>BCM 5</td>
<td>I/O Pin</td>
<td>Green</td>
</tr>
</tbody>
</table>

2. Raspberry Pi and Doorbell Button

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Doorbell Button</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3v Pin</td>
<td>VCC Pin</td>
<td>Red</td>
</tr>
<tr>
<td>GND Pin</td>
<td>GND Pin</td>
<td>Grey</td>
</tr>
<tr>
<td>BCM 15 (RXD)</td>
<td>I/O Pin</td>
<td>Green</td>
</tr>
</tbody>
</table>

3. Raspberry Pi and Relay Module

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Relay Module</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>5v Pin</td>
<td>Positive Pin</td>
<td>Red</td>
</tr>
<tr>
<td>GND Pin</td>
<td>Negative Pin</td>
<td>Grey</td>
</tr>
<tr>
<td>BCM 23</td>
<td>I/O Pin</td>
<td>Green</td>
</tr>
</tbody>
</table>

4. Relay Module, Lock and Battery
### Relay Module

<table>
<thead>
<tr>
<th></th>
<th>Lock</th>
<th>Battery</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Open Pin</td>
<td>Positive of Charge</td>
<td><em>Not Used</em></td>
<td>Red</td>
</tr>
<tr>
<td>COM Pin</td>
<td><em>Not Used</em></td>
<td>Positive</td>
<td>Red</td>
</tr>
</tbody>
</table>

5. **Raspberry Pi and Raspberry Camera**

<table>
<thead>
<tr>
<th><strong>Raspberry Pi</strong></th>
<th><strong>Raspberry Camera</strong></th>
<th><strong>Wire Colour</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Raspberry Camera Interface</td>
<td>Camera Cable</td>
<td>Green</td>
</tr>
</tbody>
</table>

5.2.2.4. **Device Control Script**

There are Python scripts implements to control the Door Access through the Raspberry Pi. The Python script is described using class diagram below.

![Class Diagram](image)

**Figure 50 Door Access Python Scripts Class Diagram**

5.2.2.5. **Functional Description**

In this section, it will describe that functions design of the Door Access in detail.

5.2.2.5.1. **Doorbell On/Off**

- **Description**
  
  The visitor could press the doorbell button on the door to enable the doorbell beep that can notify the householder to hear it.

- **Key Parameters**

<table>
<thead>
<tr>
<th>Device</th>
<th>Role</th>
<th>Action</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doorbell Button</td>
<td>sender</td>
<td>Press the doorbell button</td>
<td></td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>Notify the doorbell beep</td>
<td>bellOn()</td>
</tr>
<tr>
<td>Doorbell Button</td>
<td>sender</td>
<td>Release the doorbell button</td>
<td></td>
</tr>
</tbody>
</table>
5.2.2.5.2. E-Lock Locked/Unlocked

- **Description**
  When the householder hears the doorbell is beeping, the householder could click the lock button on the mobile phone screen to remotely control the door lock to unlock and lock through the local network.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Click lock button on</td>
<td>Wi-Fi</td>
<td>publish</td>
<td>topic/door/lock/control</td>
<td>on</td>
<td>[not applicate]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicate]</td>
<td></td>
<td>subscribe</td>
<td>topic/door/lock/control</td>
<td>[not applicate]</td>
<td>on</td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Click lock button off</td>
<td></td>
<td>publish</td>
<td>topic/door/lock/control</td>
<td>off</td>
<td>[not applicate]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicate]</td>
<td></td>
<td>subscribe</td>
<td>topic/door/lock/control</td>
<td>[not applicate]</td>
<td>off</td>
</tr>
</tbody>
</table>
5.2.2.5.3. Turn Door Camera On/Off

- **Description**
The householder enters the door access control panel screen on the mobile phone that could remotely control the door camera to turn on or off through the local network.

**Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Enter the door access control panel</td>
<td>Wi-Fi</td>
<td>publish</td>
<td>topic/door/camera/control</td>
<td>on</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>(not applicable)</td>
<td></td>
<td>subscribe</td>
<td>topic/door/camera/control</td>
<td>[not applicable]</td>
<td>on</td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Exit the door access control panel</td>
<td></td>
<td>publish</td>
<td>topic/door/camera/control</td>
<td>off</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>(not applicable)</td>
<td></td>
<td>subscribe</td>
<td>topic/door/camera/control</td>
<td>[not applicable]</td>
<td>off</td>
</tr>
</tbody>
</table>

**Sequence Diagram**

![Figure 53 Door Camera Sequence Diagram](image-url)
5.3. Sensors Control
The Sensor module maintains the sensor status which interacts with the Raspberry Pi. Currently, there are two data from sensors will be delivered to Raspberry Pi that is temperature and humidity both measurements respectively indoor and outdoor.

5.3.1. Temperature
The temperature measurement can be read from the temperature sensor by Raspberry Pi then publish it to the MQTT Broker. The mobile app also can subscribe to the message which temperature measurement sent by Raspberry Pi.

5.3.1.1. Use Case
The use case diagram below is clearly shown the process of the Temperature Sensor data exchange between the mobile phone, Raspberry Pi.

![Temperature Sensor Use Case](image)

*Figure 54 Temperature Sensor Use Case*
5.3.1.2. Class Diagram
The temperature module classes are described using UML class diagram below.

Figure 55 Temperature & Humidity Sensor Class Diagram

5.3.1.3. Wiring Diagram
The diagram described how the components connect to each other. There is useful information provided below.
1. Raspberry Pi and Indoor Sensor

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Indoor Sensor</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>5v Pin</td>
<td>Positive Pin</td>
<td>Red</td>
</tr>
<tr>
<td>GND Pin</td>
<td>Negative Pin</td>
<td>Grey</td>
</tr>
<tr>
<td>BCM 17</td>
<td>I/O Pin</td>
<td>Green</td>
</tr>
</tbody>
</table>

2. Raspberry Pi and Outdoor Sensor

<table>
<thead>
<tr>
<th>Raspberry Pi</th>
<th>Indoor Sensor</th>
<th>Wire Colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>5v Pin</td>
<td>Positive Pin</td>
<td>Red</td>
</tr>
<tr>
<td>GND Pin</td>
<td>Negative Pin</td>
<td>Grey</td>
</tr>
<tr>
<td>BCM 27</td>
<td>I/O Pin</td>
<td>Green</td>
</tr>
</tbody>
</table>

5.3.1.4. Device Control Script

There are Python scripts implemented to control the Temperature Sensor through the Raspberry Pi. The Python script is described using class diagram below.
5.3.1.5. Functional Description
In this section, it will describe the functions design of the Temperature Sensor in detail.

5.3.1.5.1. Read Indoor Temperature Measurement

- **Description**
The household enters the temperature panel screen on the mobile phone that should display the indoor temperature measurement in real-time.

- **Key Parameters**

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Enter the temperature screen</td>
<td></td>
<td>publish</td>
<td>topic/indoor/measurement/read</td>
<td>on</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td>Wi-Fi</td>
<td>subscribe</td>
<td>topic/indoor/measurement/read</td>
<td>[not applicable]</td>
<td>on</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>sender</td>
<td>Read the measurement from sensor</td>
<td></td>
<td>publish</td>
<td>topic/indoor/measurement/data</td>
<td>measurement</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/indoor/measurement/data</td>
<td>[not applicable]</td>
<td>measurement</td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Click lock button off</td>
<td></td>
<td>publish</td>
<td>topic/indoor/measurement/read</td>
<td>off</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver</td>
<td>[not applicable]</td>
<td></td>
<td>subscribe</td>
<td>topic/indoor/measurement/read</td>
<td>[not applicable]</td>
<td>off</td>
</tr>
</tbody>
</table>

Note: when the Raspberry Pi received the message of the topic 'topic/indoor/measurement/read' is 'on', then the Raspberry Pi will continuously publish the data to the topic 'topic/indoor/measurement/data' interval 1 second until it received the message which is 'off' of the topic 'topic/indoor/measurement/read'.

- **Sequence Diagram**
5.3.1.5.2. Read Outdoor Temperature Measurement

- **Description**
  The householder enters the temperature panel screen on the mobile phone that should display the outdoor temperature measurement in real-time.
### Key Parameters

<table>
<thead>
<tr>
<th>MQTT Client</th>
<th>Role</th>
<th>Action</th>
<th>Network</th>
<th>Method</th>
<th>Topic Name</th>
<th>Message</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Enter the temperature screen</td>
<td></td>
<td>publish</td>
<td>topic/outdoor/measurement/read</td>
<td>on</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver [not applicable]</td>
<td></td>
<td>Wi-Fi</td>
<td>subscribe</td>
<td>topic/outdoor/measurement/read</td>
<td>[not applicable]</td>
<td>on</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>sender</td>
<td>Read the measurement from sensor</td>
<td></td>
<td>publish</td>
<td>topic/outdoor/measurement/data</td>
<td>measurement</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver [not applicable]</td>
<td></td>
<td></td>
<td>subscribe</td>
<td>topic/outdoor/measurement/data</td>
<td>[not applicable]</td>
<td>measurement</td>
</tr>
<tr>
<td>Flutter App</td>
<td>sender</td>
<td>Exit the temperature screen</td>
<td></td>
<td>publish</td>
<td>topic/outdoor/measurement/read</td>
<td>off</td>
<td>[not applicable]</td>
</tr>
<tr>
<td>Raspberry Pi</td>
<td>receiver [not applicable]</td>
<td></td>
<td></td>
<td>subscribe</td>
<td>topic/outdoor/measurement/read</td>
<td>[not applicable]</td>
<td>off</td>
</tr>
</tbody>
</table>

Note: when the Raspberry Pi received the message of the topic 'topic/outdoor/measurement/read' is 'on', then the Raspberry Pi will continuously publish the data to the topic 'topic/outdoor/measurement/data' interval 1 second until it received the message which is 'off' of the topic 'topic/outdoor/measurement/read'.

### Sequence Diagram
5.3.2. Humidity

The humidity measurement can be read from the humidity sensor by Raspberry Pi then publish it to the MQTT Broker. The mobile app also can subscribe to the message which humidity measurement sent by Raspberry Pi.
5.3.2.1. Use Case
The use case diagram below is clearly shown the process of the Humidity Sensor data exchange between the mobile phone, Raspberry Pi.

![Humidity Sensor Use Case Diagram](image)

*Figure 60 Humidity Sensor Use Case*

5.3.2.2. Class Diagram
The temperature module classes are described using UML class diagram that is the same to temperature module classes diagram.

Please back to Temperature Class Diagram

5.3.2.3. Wiring Diagram
The diagram described how the components connect to each other. There is useful information provided is the same to Temperature module’s wiring diagram.

Please back to Temperature Wiring Diagram

5.3.2.4. Device Control Script
There are Python scripts implements to control the Temperature Sensor through the Raspberry Pi. The Python script is described using class diagram that is the same to Temperature’s.
5.3.2.5. Functional Description
In this section, it will describe the design of the Humidity Sensor in detail.

5.3.2.5.1. Read Indoor Humidity Measurement
The householder enters the humidity panel screen on the mobile phone that should display the indoor humidity measurement in real-time. The detail design which is the same to Temperature module.

Please back to Read Indoor Temperature Measurement

5.3.2.5.2. Read Outdoor Humidity Measurement
The householder enters the humidity panel screen on the mobile phone that should display the outdoor humidity measurement in real-time. The detail design which is the same to Temperature module.

Please back to Read Outdoor Temperature Measurement