CHAPTER 5
IMPLEMENTATION AND TESTING

5.1 Implementation

In the home automation project is implemented on the prototype home. Here is the hardware that is implemented on the prototype:

1. LED Room 1
2. LED Room 2
3. LED Living Room 1
4. LED Living Room 2
5. LED Terrace
6. LDR sensor
7. DHT11 Sensor
8. Motor Servo Garage

The above hardware is the device used in completing this home automation project. There are LED, DHT11 sensor, LDR sensor and Motor Servo. In this project, made a prototype to illustrate how conditions or shapes, from a house that is controlled using a red node. Here is a picture of the prototype made in this project:
Illustration 5.1: Prototype Home

The picture above is the shape of the prototype made in this project, on the prototype there are two rooms, one garage, and two living room. In the prototype which will be controlled or monitoring using Node Red.

Illustration 5.2: Flow Node Red
From the picture flow above, can be seen schematics of the sensors used. Flow is a process creation and user interface used to control the sensors. For user interface viewed through Node Red dashboard.

5.1.1 LED Room 1

![Illustration 5.3: Implementation LED Room 1](image)

Implementation performed on room 1 on this project is, use 1 LED, LEDs installed in room 1 function to illuminate room 1 space. The LEDs installed in room 1 are controlled using Node Red, where Node Red will publish topic on MQTT server, while the client (microcontroller) subscribe on the MQTT server. The MQTT server will send the publish result from Node Red to the client, so the LED can be ON or OFF.
5.1.2 LED Room 2

Implementation performed on room 2 is also the same as that used in room 1, is use 1 LED. The LED applied in room 2 serve to illuminate the space of room 2. The LEDs applied in room 2 are controlled using Node Red, where Node Red will publish topic on MQTT server, while the client (microcontroller) subscribe on the MQTT server. The MQTT server will send the publish result from Node Red to the client, so the LED can be ON or OFF.
5.1.3 LED Living Room 1

Illustration 5.5: Implementation LED Living Room 1

Implementation performed on the living room 1 is the same as rooms 1 and 2, which uses 1 LED to illuminate the living room 1. LEDs in the living room 1 are controlled using Node Red, where Node Red publish topic on MQTT server, while the client (microcontroller) subscribe topic on MQTT server, so LED can ON or OFF.
5.1.4 LED Living Room 2

Implementation performed on the living room 2 is the same as room 1, room 2, and living room 1, where using 1 LED to illuminate the living room 2. LEDs applied to living room 2 are controlled using Node Red, where Node Red publish topic on MQTT server, while the client (microcontroller) subscribe topic on MQTT server, so LED can ON or OFF.
5.1.5 LED Terrace

Implementation is done on the terrace that is using 3 LED. The LED are applied to the terrace serve to illuminate the terrace. Terrace LED can be controlled using Node Red, where Node Red publish topic on MQTT server, while the client (microcontroller) subscribe topic on MQTT server, so LED can ON or OFF.
5.1.6 LDR Sensor

Implementation of LDR sensor hardware, done on the location of the living room 2, where the LDR sensor serves to monitor the LED lighting, whether the LED is on or off. The LDR sensor is monitored through Node Red dashboard, where the LDR sensor sends the lighting based on the time specified. On this project, Arduino Uno or client will send a lighting sensor every 5000 ms to MQTT server, while Node Red subscribe to the MQTT server, so monitoring of LED lighting can be monitored through Node Red.
5.1.7 DHT 11 Sensor

Implementation hardware DHT11 Sensor, performed on the location of the living room 2, where the DHT11 sensor serves to monitor the room temperature. The DHT 11 sensor is monitored through Node Red, where the DHT11 sensor sends room temperature based on the time specified. On this project, Arduino UNO or client will send a temperature of every 3000 ms on the MQTT server, while in Node Red subscribe to the MQTT server, so that the monitoring of room temperature can be monitored through the Red Node.
5.1.8 Motor Servo Garage

Implementation is done in hardware servo motor placed in the garage, where the servo motor is controlled using Node Red. Functions the same as LEDs are ON or OFF only, but on the servo motor when opened or ON servo motor will rotate as much as 110 degrees, and when closed servo motor will return to its original position which is 0 degree. On servo motor Node Red publish on MQTT server, while the client (microcontroller) subscribe on the MQTT server, so that the servo motor can be moved as much as 110 degrees or 0 degrees.
5.2 Testing

In this project two tests were performed, Where testing is used to determine the ability of this project. The two testing conducted among others, namely:

5.2.1 Functional Testing

In functional testing is done to determine whether the system created on this project is running well or not. To test the system works well, publish topics or subscribe topics from Dashboard Node Red.

![Illustration 5.11: Screenshot Node Red Dashboard]

5.2.1.1 Functional Testing LED

In testing performed on LED room 1 is, Node Red is used to turn LED on or off. In this project Node Red is used to publish topic on MQTT server, while the client that subscribe on MQTT server is client microcontroller or Arduino Uno. Below is a picture of the screenshot of the project created.
The picture above is a picture showing Node Red Publish topic on the MQTT server. If the publish is successful, then the Switch image is changed as in the picture above. Below is shown screenshot image of output used in this project succeed or not.
And with other LED, used in completing this project, how the system works the same as that used LED room 1, then from that on the other LED only displayed screenshot only.
Illustration 5.18: Screenshot Publish Topic Node Red Dashboard Living Room 1

Illustration 5.19: Screenshot Arduino IDE Living Room 1
Illustration 5.20: Output Living Room 1

Illustration 5.21: Screenshot Publish Topic Node Red Dashboard Living Room 2
Illustration 5.22: Screenshot Arduino IDE Living Room 2

Illustration 5.23: Output Living Room 2
Illustration 5.24: Screenshot Publish Topic Node Red Dashboard Terrace

Illustration 5.25: Screenshot Arduino IDE Terrace
5.2.1.2 Functional Testing Motor Servo

In testing Servo Motor basically the same concept as LED that is, Node
Red publish topic on the MQTT Server, and client microcontroller subscribe on
the MQTT server. Here is a screenshot of the servo motor used in this project.

Illustration 5.27: Screenshot Publish Topic Node Red Dashboard Motor Servo Garage
Illustration 5.28: Screenshot Arduino IDE Garage

Illustration 5.29: Output Garage
5.2.1.3 Functional Testing DHT 11

On DHT sensor 11 client microcontroller publish topic on the MQTT Server, while in Node Red graphically displayed condition temperature read by DHT11 sensor. In this project the temperature read by the DHT 11 sensor is displayed graphically and update according to the set time of 3000 ms. Here are Node Red Dashboard screenshots read DHT11 sensor.

Illustration 5.30: Screenshot Node Red Dashboard DHT11

5.2.1.4 Functional Testing LDR

In the LDR client microcontroller or Arduino Uno publish topic on the MQTT server, while in the Red Node only displayed brightness LEDs that are read by the LDR sensor, and in the Node Red dashboard it is notified that the LED is on or off.
5.2.2 Performance Testing

The ability test performed on this project is to test the server capabilities used in this project. In this testing is done publish topic on the MQTT server as much as 1000, 10000, 50000, and 100000, to find out how much memory and CPU is being used by the MQTT server.
Illustration 5.33: Publish Topic 10000

Illustration 5.34: Publish Topic 50000
Illustration 5.35: Publish Topic 100000

The picture above is a screenshot of publish topic on the MQTT server. From the publish on the MQTT server obtained average data from the use of CPU, memory, and how long publish topic on the MQTT server with publish 1000, 10000, 50000, and 100000. Here is the average data obtained.

<table>
<thead>
<tr>
<th>Publish Topic</th>
<th>% CPU</th>
<th>% Memory</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000 Topic</td>
<td>3.4 %</td>
<td>0.1 %</td>
<td>17 Second</td>
</tr>
<tr>
<td>10000 Topic</td>
<td>5.2 %</td>
<td>0.1 %</td>
<td>01.27 Minute</td>
</tr>
<tr>
<td>50000 Topic</td>
<td>5.1 %</td>
<td>0.1 %</td>
<td>07.46 Minute</td>
</tr>
<tr>
<td>100000 Topic</td>
<td>4.9 %</td>
<td>0.1 %</td>
<td>15.42 Minute</td>
</tr>
</tbody>
</table>

Table 5.1: Table Performance Testing MQTT Server

In accordance with the results of the test publish on the MQTT server, the memory used by this MQTT server is quite small. The MQTT protocol itself has several optimizations, among others:

1. Pressing the data packet size as small as possible so that traffic can be increased.
2. Minimizes computational processes for encoding and decoding of data packet.
3. Data uses only the smallest possible storage space.