บ้านอัจฉริยะ

SMART HOME

พัสกร พงศ์ภูวิวรรธน์

สรวิชญ์ ราชวงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

บ้านอัจฉริยะ

SMART HOME

พัสกร พงศ์ภูวิวรรธน์

สรวิชญ์ ราชวงษ์

b00264495
TB00010

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง บ้านอัจฉริยะ

SMART HOME

ผู้จัดทำ

1. นายพัสกร พงศ์ภูวิวรรธน์ รหัสนักศึกษา 57010885

2. นายสรวิชญ์ ราชวงษ์ รหัสนักศึกษา 57011311

_____อาจารย์ที่ปรึกษา

(ผศ.ธนา หงษ์สุวรรณ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บ้านอัจฉริยะ

นายพัสกร พงศ์ภูวิวรรธน์ 57010885
นายสรวิชญ์ ราชวงษ์ 57011311
ผศ.ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ปีการศึกษา 2560

บทคัดย่อ

โครงงานชิ้นนี้ได้เห็นถึงการเข้ามามีส่วนร่วมของเทคโนโลยีสำหรับการใช้งานในชีวิตประจำวัน ซึ่งเทคโนโลยีในอดีตไม่สามารถตอบสนองความต้องการทั้งหมดของมนุษย์ได้และมีค่าใช้จ่ายที่สูง ไม่ว่าจะเป็นโทรศัพท์มือถือที่ในสมัยก่อนสามารถทำได้เพียงใช้ในการโทรศัพท์หรือรับสายจากผู้อื่น แต่ในปัจจุบันความสามารถของโทรศัพท์มือถือนั้นได้พัฒนาให้ตอบสนองความต้องการของผู้ใช้งานได้หลากหลายและมีประสิทธิภาพมากยิ่งขึ้น ไม่เว้นแม้แต่อุปกรณ์อิเล็กทรอนิกส์อื่น ๆ ที่มีความสามารถเพิ่มขึ้นตามไปด้วย เช่น การนำเซ็นเซอร์มาใช้ทำงานร่วมกับอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ เพื่ออำนวยความสะดวกแก่ผู้ใช้งาน และการเชื่อมต่อเพื่อควบคุมอุปกรณ์อิเล็กทรอนิกส์ผ่านอินเตอร์เน็ต จากการพัฒนาทางด้านเทคโนโลยีที่เกิดขึ้น นำไปสู่เทคโนโลยีรูปแบบใหม่ที่ชื่อว่า IoT หรือ Internet of Things

ทางผู้จัดทำจึงได้นำเทคโนโลยี IoT มาพัฒนาอุปกรณ์เพื่อความสะดวกสบายภายในบ้านโดยการควบคุมอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ด้วยบอร์ด Raspberry Pi ผ่านแอปพลิเคชันในโทรศัพท์มือถือ ซึ่งบอร์ด Raspberry Pi ทำหน้าที่เป็นเซิร์ฟเวอร์ขนาดเล็กภายในบ้านสำหรับส่งคำสั่งเพื่อควบคุมการทำงานของอุปกรณ์ต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Smart Home

Mr. Phasakorn	Phongbhuviwat	57010885
Mr. Sorawit	Rachawong	57011311
Asst.Prof. Thana	Hongsuwan	Advisor

Academic Year 2560

ABSTRACT

This project has seen the participation of technology for everyday use. Technology in the past can not meet all human needs and has a high cost. Whether it is a mobile phone that in the past can only be used to make calls or receive calls from others. Today, the capabilities of mobile phones have evolved to meet the needs of many users and more efficient. No other electronic devices have the ability to increase, such as the use of sensors to work with electronic devices to facilitate the user. And the connection to control electronic devices through the Internet. The development of technology occurs. Leads to a new technology called IoT or Internet of Things.

IoT technology has developed the device for convenience in home with Raspberry Pi Boards through mobile applications, the Raspberry Pi board acts as a small home server to send commands to control the operation of various devices

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานชิ้นนี้ไม่อาจประสบความสำเร็จหากขาดความช่วยเหลือและชี้แนะแนวทางจาก ผศ.ธนา หงษ์สุวรรณ ที่ช่วยแก้ปัญหาที่เกิดขึ้นขณะทำการทดลองเพื่อพัฒนาโครงงานตลอดระยะเวลาการทำงาน คอยดูแลติดตามความก้าวหน้าของโครงงานเพื่อให้เป็นไปตามแบบแผนสำหรับการพัฒนาที่ได้วางไว้

ขอขอบพระคุณคณะอาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ให้ความรู้และตอบคำถามที่สงสัยขณะทำการศึกษาอยู่ตลอดระยะเวลา 4 ปี สำหรับนำไปใช้ในการพัฒนาโครงงานและการทำงานในอนาคตเพื่อให้ให้ผ่านอุปสรรคต่าง ๆ ที่เกิดขึ้นได้

ขอขอบพระคุณบิดาและมารดาที่คอยสนับสนุนและให้กำลังใจตลอดระยะเวลาที่ทำการศึกษาจนสำเร็จลุล่วงไปได้ด้วยดี

สุดท้าย หวังว่าโครงงานชิ้นนี้จะเป็นประโยชน์ต่อผู้ที่สนใจสำหรับนำไปใช้งานหรือการศึกษาเพื่อพัฒนาต่อในอนาคต หากเกิดความผิดพลาดประการใด ทางผู้จัดทำต้องขออภัยมา ณ ที่นี้ด้วย

พัสกร พงศ์ภูวิวรรธน์
สรวิชญ์ ราชวงษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

สารบัญ IV

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 2

1.4 วิธีการดำเนินงาน 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

บทที่ 2 ความรู้และเทคโนโลยีที่เกี่ยวข้อง 4

2.1 ทฤษฎีเกี่ยวกับ Hardware 4

2.1.1 Internet of Things 4

2.1.2 Raspberry Pi 4

2.1.3 NodeMCU 5

2.1.4 รีเลย์ (Relay) 6

2.1.5 เราเตอร์ (Router) 7

2.1.6 ESP8266 8

2.2 ทฤษฎีเกี่ยวกับ Software 9

2.2.1 Raspbian jessie 9

2.2.2 Node-RED 9

2.2.3 JSON 10

2.2.4 HTTP 10

2.2.5 MQTT 10

2.2.6 React-Native 11

2.2.7 Android 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและพัฒนา 14
3.1 แนวคิดในการพัฒนา 14
3.2 เครื่องมือที่ใช้ในการพัฒนา 14
3.2.1 ภาษาที่ใช้ในการพัฒนา 14
3.2.2 เทคโนโลยีที่ใช้ในการพัฒนา 14
3.2.3 อุปกรณ์ที่ใช้ในการพัฒนา 15
3.3 ส่วนต่อประสานกับผู้ใช้ 15
3.3.1 Application 15
3.3.2 Dashboard 20
3.4 โครงสร้างฮาร์ดแวร์ 23
3.4.1 Arduino 23
3.4.2 PCB 34
3.5 ส่วนจัดการระบบ 35
3.5.1 Node-RED 35

บทที่ 4 การทดลองและผลการทดลอง 40
4.1 การทดลองส่วน Application และ Server 40
4.1.1 การเรียกดูหน้า Home 40
4.1.2 การเรียกดูหน้า Room 42
4.1.3 การเพิ่ม/ลบห้อง 44
4.1.4 การเพิ่ม/ลบอุปกรณ์ 46
4.1.5 การสั่งเปิดปิด 49
4.1.6 การตั้งการทำงานอัตโนมัติ 50
4.2 การทดลองส่วนฮาร์ดแวร์ 52
4.2.1 การเปิด/ปิดอุปกรณ์ 52
4.2.2 การรับค่าจากเซ็นเซอร์ 53
4.2.3 การรับส่งข้อมูลกับเซิร์ฟเวอร์ 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 5 บทสรุปและข้อเสนอแนะ 55
5.1 บทสรุปของโครงงาน 55
5.2 ปัญหา อุปสรรค และแนวทางแก้ไข 55
5.3 แนวทางในการพัฒนาต่อ 55

บรรณานุกรม 56

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป **หน้า**

รูป 2.1 บอร์ด Raspberry Pi 4
รูป 2.2 ตัวอย่าง NodeMCU 5
รูป 2.3 ESP-01 Datasheet 6
รูป 2.4 ESP-12e Datasheet 6
รูป 2.5 การทำงานของรีเลย์ 7
รูป 2.6 ตัวอย่าง Router 8
รูป 2.7 ESP-8266 9
รูป 2.8 MQTT 11
รูป 3.1 หน้า Home 16
รูป 3.2 หน้าสำหรับเพิ่มห้อง 17
รูป 3.3 หน้าอุปกรณ์ ในห้อง 18
รูป 3.4 หน้าเพิ่มอุปกรณ์ 19
รูป 3.5 หน้าตั้งค่าการทำงานอัตโนมัติ 20
รูป 3.6 หน้า Home ของ Dashboard 21
รูป 3.7 Guage ของหน้า Room ของ Dashboard 21
รูป 3.8 Graph ของหน้า Room ของ Dashboard 22
รูป 3.9 การทำงานของสวิทช์ 23
รูป 3.10 การทำงานของฟังก์ชั่น callback ในสวิทช์ 24
รูป 3.11 การทำงานของปลั๊ก 29
รูป 3.12 การทำงานของฟังก์ชั่น callback ในปลั๊ก 30
รูป 3.13 PCB ของ Switch 34
รูป 3.14 PCB ของ Plug 34
รูป 3.15 ภาพรวมโฟลวการทำงานของ Node-RED 35
รูป 3.16 โฟลวการส่งคำร้องขอสำหรับเปิดหน้า Home 36
รูป 3.17 โฟลวการส่งคำร้องขอสำหรับเปิดหน้า Room 36
รูป 3.18 โฟลวการสั่งการสำหรับลบ Room 36
รูป 3.19 โฟลวการสั่งการสำหรับเพิ่ม Room 37
รูป 3.20 โฟลวการสั่งเปิดปิดอุปกรณ์ 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป **หน้า**

รูป 3.21 โฟลวการลบ Device 37
รูป 3.22 โฟลวการเพิ่ม Device 38
รูป 3.23 โฟลวการตั้งค่าการทำงานอัตโนมัติ 38
รูป 3.24 โฟลวการเปลี่ยนค่าในฐานข้อมูลเมื่อรับค่าจาก MQTT 38
รูป 3.25 โฟลวการทำงาน Dashboard 39
รูป 4.1 ผลลัพท์การส่งคำร้องขอของหน้า Home 41
รูป 4.2 ผลลัพท์ที่ application แสดงในหน้า Home 42
รูป 4.3 request ที่ application ส่งให้ NodeRed 42
รูป 4.4 ผลลัพท์การส่งคำร้องขอของหน้า Room 43
รูป 4.5 ผลลัพท์ที่ application แสดงในหน้า Room 44
รูป 4.6 การทดลองเพิ่มห้อง 45
รูป 4.7 request ที่ application ส่งเมื่อทดลองเพิ่มห้อง 45
รูป 4.8 request ที่ application ส่งไปเมื่องกดลบห้อง 46
รูป 4.9 ผลลัพท์ที่ application แสดงหลังจาก ลบห้อง 46
รูป 4.10 การทดลองเพิ่มอุปกรณ์ 47
รูป 4.11 request ที่ application ส่งไปเมื่อทดลองเพิ่มอุปกรณ์ 47
รูป 4.12 ผลลัพท์ที่ application แสดงหลังจากทำการทดลองเพิ่มอุปกรณ์ 48
รูป 4.13 request ที่ application ส่งไปเมื่อทดลองลบอุปกรณ์ 48
รูป 4.14 ผลลัพท์ที่ application แสดงหลังจาก ลบอุปกรณ์ 49
รูป 4.15 request ที่ application ส่งไปเมื่อทดลองสั่งเปิด 49
รูป 4.16 request ที่ application ส่งไปเมื่อทดลองสั่งปิดอุปกรณ์ 50
รูป 4.17 ผลลัพท์ การส่ง mqtt ของ NodeRed เมื่อทำการสั่งปิดอุปกรณ์ 50
รูป 4.18 การทดลองการตั้งค่าอัตโนมัติ 50
รูป 4.19 request ที่ application ส่งไปเมื่องตั้งค่าการทำงานอัตโนมัติ 51
รูป 4.20 อุปกรณ์ในขณะปิดอยู่ 52
รูป 4.21 อุปกรณ์ในขณะเปิดอยู่ 52
รูป 4.22 ค่าการใช้งานสวิทช์จาก Serial Monitor 53
รูป 4.23 ค่าที่เซิร์ฟเวอร์ได้รับ ในช่วงเวลาหนึ่ง ๆ ที่เปิดการทำงานของอุปกรณ์สวิทช์ 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป **หน้า**

รูป 4.24 ค่าของสวิทช์จาก Serial Monitor 54

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

เทคโนโลยีในอดีตไม่สามารถตอบสนองความต้องการทั้งหมดของมนุษย์ได้และมีค่าใช้จ่ายที่สูง ไม่ว่าจะเป็นโทรศัพท์มือถือที่ในสมัยก่อนสามารถทำได้เพียงใช้ในการโทรศัพท์หรือรับสายจากผู้อื่น แต่ในปัจจุบันความสามารถของโทรศัพท์มือถือนั้นได้พัฒนาให้ตอบสนองความต้องการของผู้ใช้งานได้หลากหลายและมีประสิทธิภาพมากยิ่งขึ้น ไม่เว้นแม้แต่อุปกรณ์อิเล็กทรอนิกส์อื่น ๆ ที่มีความสามารถเพิ่มขึ้นตามไปด้วย เช่น การนำเซ็นเซอร์มาใช้ทำงานร่วมกับอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ เพื่ออำนวยความสะดวกแก่ผู้ใช้งาน และการเชื่อมต่อเพื่อควบคุมอุปกรณ์อิเล็กทรอนิกส์ผ่านอินเตอร์เน็ต จากการพัฒนาทางด้านเทคโนโลยีที่เกิดขึ้น

Internet of Things (IoT) เป็นระบบเครือข่ายที่เชื่อมต่ออุปกรณ์ต่างๆทั้งหมด เช่น อุปกรณ์เครื่องใช้ไฟฟ้า รถยนต์ โทรศัพท์มือถือ เครื่องมือสื่อสาร เครื่องมือทางการเกษตร อาคาร บ้านเรือน เครื่องใช้ในชีวิตประจำวันต่างๆ ผ่านเครือข่ายอินเตอร์เน็ต และให้อุปกรณ์เหล่านี้เก็บและแลกเปลี่ยนข้อมูลกันได้ ทำให้สามารถสั่งการได้ง่าย สะดวกสบายต่อการใช้ชีวิตมากยิ่งขึ้น และนี้คือการประยุกต์ใช้ร่วมกันตามที่กล่าวข้างบนมานั้นเอง และปัจจุบันราคาของอุปกรณ์อิเล็กทรอนิกส์ต่างๆนั้นมีราคาถูกลงเลยทำให้เรื่องเหล่านี้สามารถทำได้ง่ายขึ้นเข้าถึงผู้คนได้มากขึ้น ทำให้เรื่องเหล่านี้ใกล้ตัวมากขึ้น และการดำเนินชีวตของเราสะดวกมากยิ่งขึ้น

จากที่กล่าวมา ทางผู้จัดทำจึงมีแนวคิดที่จะสร้างระบบบ้านอัจฉริยะ โดยการใช้เทคโนโลยี Internet of things เพื่อควบคุมการทำงานของอุปกรณ์ต่าง ๆ ภายในบ้านอย่างอัตโนมัติ โดยพิจารณาจากเซ็นเซอร์ต่าง ๆ ได้แก่ (1) เซ็นเซอร์วัดความเข้มของแสง ซึ่งจะใช้สำหรับควบคุมการเปิดและปิดไฟผ่านการรับแสงในขณะนั้น (2) เซ็นเซอร์วัดอุณหภูมิ ซึ่งใช้ในการรับค่าของอุณหภูมิมาพิจารณาเพื่อทำการเปิดพัดลมและใช้งานระบบระบายอากาศในบ้านเมื่ออุณหภูมิอยู่ในค่าที่กำหนด (3) เซ็นเซอร์ตรวจจับการเคลื่อนไหว ใช้ในการตรวจจับว่ามีผู้อยู่อาศัยหรือไม่ เพื่อสั่งการให้เครื่องใช้ไฟฟ้าภายในบ้านทำงาน โดยจะทำการเชื่อมต่อระบบกับแอปพลิเคชั่นบนโทรศัพท์มือถือในระบบปฏิบัติการ Android เพื่อลดข้อจำกัดทางด้านฮาร์ดแวร์ เพื่อให้สามารถควบคุมการเปิดและปิดอุปกรณ์เครื่องใช้ไฟฟ้า อีกทั้งยังสามารถเรียกดูข้อมูลต่าง ๆ ในรูปแบบของกราฟได้ เพื่อให้ผู้อยู่อาศัยได้ใช้ชีวิตอย่างสะดวกสบายและมีความปลอดภัยมากยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อลดการใช้พลังงานไฟฟ้าภายในบ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เพื่อเพิ่มความสะดวกสบายในการใช้ชีวิตประจำวัน
3) เพื่อป้องกันอุบัติเหตุที่อาจเกิดขึ้นจากการใช้งาน

1.3 ขอบเขตของโครงงาน

สร้างระบบ Smart Home โดยที่ระบบจะทำงานอัตโนมัติตามที่ได้โปรแกรมไว้ โดยผู้ใช้งานนั้นสามารถควบคุมการทำงานและสามารถดูสถานะหรือค่าต่าง ๆ ของอุปกรณ์และเซ็นเซอร์ที่เชื่อมต่อกับระบบ Smart Home ผ่านแอพพลิเคชั่นจากที่ใดก็ได้ โดยระบบจะมีขอบเขตของการทำงาน ดังนี้

1) ใช้ NodeMCU อ่านข้อมูลจากเซ็นเซอร์ที่เชื่อมต่อ โดยจะใช้เซ็นเซอร์วัดอุณหภูมิ ความชื้น เซ็นเซอร์วัดความเข้มแสง และเซ็นเซอร์ตรวจจับการเคลื่อนไหว เป็นต้น และเชื่อมต่อกับ Relay เพื่อควบคุมการทำงานของเครื่องใช้ไฟฟ้า
2) MQTT (Message Queue Telemetry Transport) เป็นโปรโตคอลสำหรับ Internet of Things โปรโตคอลนี้ใช้วิธีการเหมือนกับ Message Queue ปกติ แต่พัฒนาให้รองรับงานด้าน Internet of Things นอกจากจะรอรับและอ่านค่าแล้ว ยังสามารถสั่งงานอุปกรณ์ที่ใช้ MQTT ได้ด้วย การใช้งานโปรโตคอลนี้ต้องจะต้องมีอุปกรณ์ที่ฝัง MQTT Broker มาเพื่อทำหน้าที่รับส่งข้อมูลไปยังอุปกรณ์อื่นที่ใช้ MQTT เหมือนกัน ส่งข้อมูลจาก NodeMCU ผ่าน MQTT protocol ไปยัง MQTT broker ซึ่งในที่นี้จะใช้ Raspberry Pi เป็น MQTT broker และเก็บข้อมูลที่ส่งเข้ามาใน MQTT Broker ไว้ใน Raspberry Pi ซึ่งสามารถใส่หน่วยความจำเป็น SD Card ได้
3) ตั้งค่า Forward port ที่ Wireless Router เพื่อให้สามารถสั่งการอุปกรณ์ภายในบ้านเมื่อผู้ใช้งานอยู่นอกบ้านได้
4) สร้างแอพพลิเคชั่นเพื่อให้ผู้ใช้สามารถเชื่อมต่อกับระบบ Smart Home โดยใช้ Raspberry Pi เป็น Server โดยจะมีความสามารถดังนี้
    - สามารถควบคุมสั่งการเปิดปิดเครื่องใช้ไฟฟ้าภายในบ้านได้
    - สามารถดูข้อมูลการใช้งานของอุปกรณ์และค่าที่วัดได้จากเซ็นเซอร์ โดยจะแสดงเป็นกราฟเพื่อให้ง่ายต่อการเข้าใจและตรวจสอบ
    - สามารถปิดระบบอัตโนมัติของแต่ละส่วนได้ ในกรณีที่เซ็นเซอร์มีปัญหา (วัดค่าไม่ได้ตามจริง)
    - สามารถเลือกโหมดภายในห้องนอนได้ โดยจะปรับแสงภายในห้องตามโหมดที่ได้เลือก
5) ระบบเพื่อความสะดวกสบายและการทำงานของเครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รับอุณหภูมิผ่านเซ็นเซอร์วัดอุณหภูมิ อุณหภูมิเมื่อถึงจุดที่กำหนดจะทำการเปิดพัดลมและใช้งานระบบระบายอากาศในบ้าน
- ตรวจจับความสว่างผ่านเซ็นเซอร์วัดความเข้มแสงจะใช้ในการควบคุมการเปิด-ปิดไฟ
- เมื่อไม่มีใครอยู่ภายในบ้าน ระบบจะทำการตัดไฟเครื่องใช้ไฟฟ้าที่ไม่จำเป็นเพื่อการประหยัดพลังงาน
- ระบบจะเปิดไฟอัตโนมัติเมื่อมีคนเข้าไปในระยะการตรวจจับของเซ็นเซอร์ตรวจจับการเคลื่อนไหว และจะปิดไฟเมื่อไม่มีคนเคลื่อนไหวแล้วระยะหนึ่ง
- ระบบจะปรับความสว่างและสีของแสงไฟ และองศาของมูลี่ภายในห้องนอนตามโหมดที่เลือกในแอพพลิเคชั่น โดยพิจารณาจากเซนเซอร์รับแสงและเวลา

1.4 วิธีการดำเนินงาน

1) กำหนดขอบเขตการทำงาน วัตถุประสงค์ อุปกรณ์ที่จะใช้ในการรับและแสดงผลรวมทั้งภาษาที่จะใช้ในการเขียนโปรแกรม
2) วิเคราะห์ระบบจากการศึกษาหลักการ ทฤษฎีและเอกสารที่เกี่ยวข้อง
3) ออกแบบระบบในส่วนต่าง ๆ เช่น โครงสร้างภาพรวมของระบบ โครงสร้างของส่วนจัดเก็บข้อมูลทดสอบ
4) เก็บข้อมูลที่ใช้ในการทดลอง
5) พัฒนาโปรแกรมประยุกต์
6) ทำผลงานจำลองออกมาใช้จัดแสดง
7) ทดสอบโปรแกรมและแก้ไขข้อบกพร่องให้สมบูรณ์และถูกต้องที่สุด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ช่วยประหยัดค่าไฟฟ้าภายในบ้านได้
2) มีระบบรักษาความปลอดภัยภายในบ้าน
3) ระบบสามารถแสดงค่าและสถานะต่าง ๆ ผ่านแอพพลิเคชั่นในรูปแบบกราฟหรือตารางได้เพื่อให้ผู้ใช้สามารถเข้าใจและตรวจสอบได้ง่าย
4) เข้าใจการรับ-ส่งและการดึงข้อมูลมาเก็บผ่าน MQTT protocol
5) เข้าใจการสร้างและการทำงานของแอพพลิเคชั่น และสามารถเขียนให้ส่งข้อมูลผ่าน MQTT Protocol ได้
6) สามารถนำไปประยุกต์ใช้กับระบบอื่น ๆ ที่เกี่ยวข้องกับ IoT หรือ MQTT Protocol ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้และเทคโนโลยีที่เกี่ยวข้อง

2.1 ทฤษฎีเกี่ยวกับ Hardware

2.1.1 Internet of Things

Internet of Things (IoT) เป็นระบบเครือข่ายที่เชื่อมต่ออุปกรณ์ต่าง ๆ ทั้งหมด เช่น อุปกรณ์เครื่องใช้ไฟฟ้า รถยนต์ โทรศัพท์มือถือ เครื่องมือสื่อสาร เครื่องมือทางการเกษตร อาคารบ้านเรือน เครื่องใช้ในชีวิตประจำวันต่าง ๆ ผ่านเครือข่ายอินเตอร์เน็ตและให้อุปกรณ์เหล่านี้เก็บและแลกเปลี่ยนข้อมูลกันได้ ทำให้สามารถสั่งการได้ง่าย

IoT มีชื่อเรียกอีกชื่อหนึ่งว่า M2M ย่อมาจาก Machine to Machine คือเทคโนโลยีอินเตอร์เน็ตที่เชื่อมต่ออุปกรณ์กับเครื่องมือต่าง ๆ เข้าไว้ด้วยกัน

โดยการทำงานจะอาศัยอุปกรณ์ประเภท RFID และ Sensor เป็นหลักโดยจะเชื่อมต่อกันผ่านทางเครือข่ายอินเตอร์เน็ต เพื่อให้รับค่า จัดเก็บ และดึงข้อมูลนำไปใช้ได้ โดยสามารถจัดเก็บลง Database ได้ด้วย

2.1.2 Raspberry Pi

รูป 2.1 บอร์ด Raspberry Pi

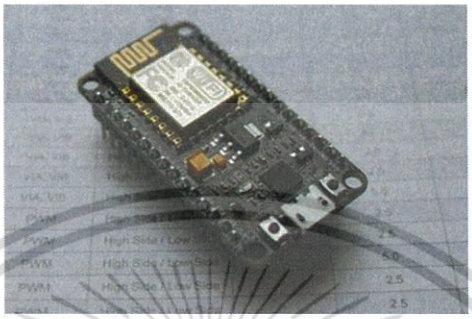
บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงงานทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมล หรือเล่นเกมส์ อีกทั้งยังสามารถเล่นไฟล์วีดีโอความละเอียดสูง (High-Definition) ได้

บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน SD Card ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่น ๆ ได้

2.1.3 NodeMCU

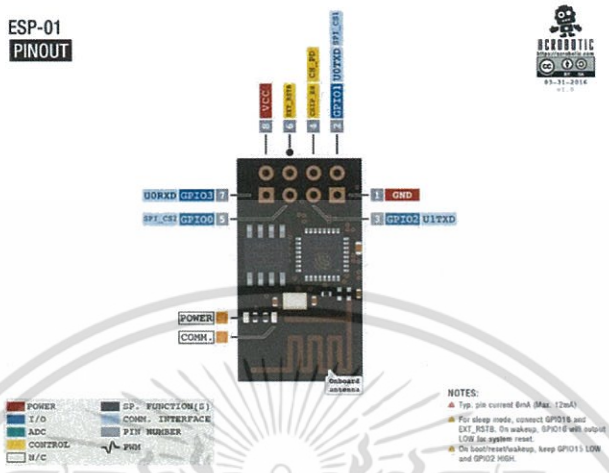


รูป 2.2 ตัวอย่าง NodeMCU

แพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างโปรเจค Internet of Things (IoT) ที่ประกอบไปด้วย Development Kit (ตัวบอร์ด) และ Firmware (Software บนบอร์ด) ที่เป็น open source สามารถเขียนโปรแกรมด้วยภาษา Lau ได้ ทำให้ใช้งานได้ง่ายขึ้น มาพร้อมกับโมดูล WiFi (ESP8266) ซึ่งเป็นหัวใจสำคัญในการใช้เชื่อมต่อกับอินเตอร์เน็ตนั่นเอง ตัวโมดูล ESP8266นั้นมีอยู่ด้วยกันหลายรุ่นมาก ตั้งแต่ Version แรกที่เป็น ESP-01 ไล่ไปเรื่อยๆจนปัจจุบันมีถึง ESP-12 แล้ว และที่ฝังอยู่ใน NodeMCU version แรกนั้นก็เป็น ESP-12 แต่ใน Version 2 นั้นจะใช้เป็น ESP-12E แทน ซึ่งการใช้งานโดยรวมก็ไม่แตกต่างกันมากนัก NodeMCUนั้นมีลักษณะคล้ายกับ Arduino ตรงที่มีพอร์ต Input Output built inมาในตัว สามารถเขียนโปรแกรมคอนโทรลอุปกรณ์ I/O ได้โดยไม่ต้องผ่านอุปกรณ์อื่นๆ และเมื่อไม่นานมานี้ก็มีนักพัฒนาที่สามารถทำให้ Arduino IDE ใช้งานร่วมกับ Node MCU ได้ จึงทำให้ใช้ภาษา C/C++ ในการเขียนโปรแกรมได้ ทำให้เราสามารถใช้งานมันได้หลากหลายมากยิ่งขึ้น NodeMCUตัวนี้สามารถทำอะไรได้หลายอย่างมากโดยเฉพาะเรื่องที่เกี่ยวข้องกับ IoT ไม่ว่าจะเป็นการทำ Web Server ขนาดเล็ก การควบคุมการเปิดปิดไฟผ่านWiFi และอื่น ๆ อีกมากมาย

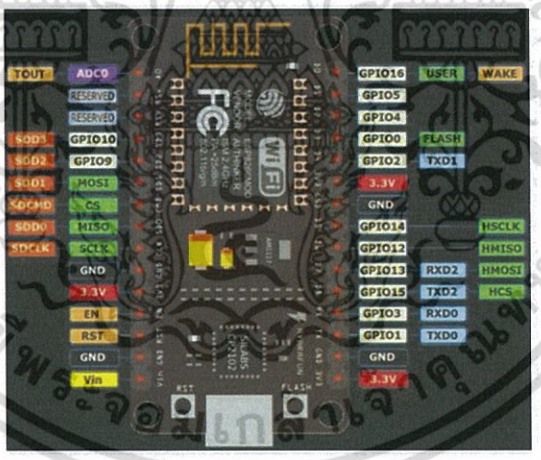
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.1 ESP-01



รูป 2.3 ESP-01 Datasheet

2.1.3.2 ESP-12E



รูป 2.4 ESP-12e Datasheet

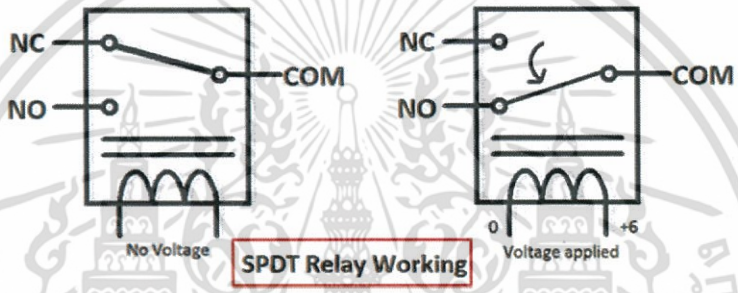
2.1.4 รีเลย์ (Relay)

รีเลย์ (Relay) คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เป็นสวิตซ์ตัด-ต่อวงจร โดยใช้แม่เหล็กไฟฟ้าและการที่จะให้ทำงานก็ต้องจ่ายไฟให้ตามที่กำหนด เพราะเมื่อจ่ายไฟให้กับตัวรีเลย์จะทำให้หน้าสัมผัสติดกันกลายเป็นวงจรปิด และตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้ก็จะกลายเป็นวงจรเปิด ไฟที่เราใช้ป้อนให้กับตัวรีเลย์ก็จะเป็นไฟที่มาจากเพาเวอร์ซัพพลายของเครื่องเรา ดังนั้นทันทีที่เปิดเครื่อง ก็จะทำให้รีเลย์ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วยส่วนสำคัญ 2 ส่วนหลัก คือ

1) ส่วนของขดลวด (coil) เหนี่ยวนำกระแสต่ำ ทำหน้าที่สร้างสนามแม่เหล็กไฟฟ้าให้แกนโลหะไปกระทุ้งให้หน้าสัมผัสต่อกัน ทำงานโดยการรับแรงดันจากภายนอกต่อคร่อมที่ขดลวดเหนี่ยวนำนี้ เมื่อขดลวดได้รับแรงดัน (ค่าแรงดันที่รีเลย์ต้องการขึ้นกับชนิดและรุ่นตามที่ผู้ผลิตกำหนด) จะเกิดสนามแม่เหล็กไฟฟ้าทำให้แกนโลหะด้านในไปกระทุ้งให้แผ่นหน้าสัมผัสต่อกัน
2) ส่วนของหน้าสัมผัส (contact) ทำหน้าที่เหมือนสวิตช์จ่ายกระแสไฟให้กับอุปกรณ์ที่เราต้องการนั่นเอง



รูป 2.5 การทำงานของรีเลย์

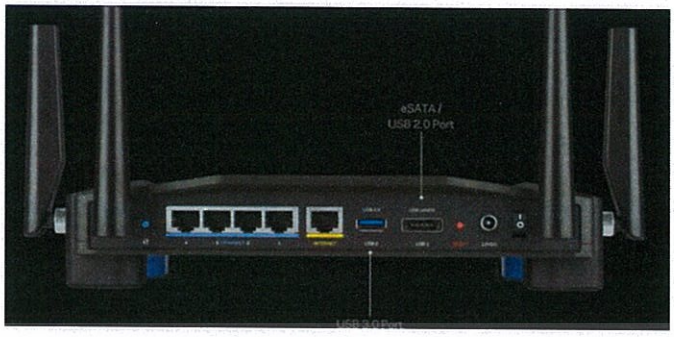
โดยรีเลย์จะมีขาอยู่ 3 ขาด้วยกัน คือ

1) ขา NC ย่อมาจาก normal close หมายความว่าปกติปิด หรือ หากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา
2) ขา NO ย่อมาจาก normal open หมายความว่าปกติเปิด หรือหากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิดปิด เช่น โคมไฟสนามเหนือหน้าบ้าน
3) ขา C ย่อมากจาก common คือจุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ

2.1.5 เราเตอร์ (Router)

เป็นอุปกรณ์คอมพิวเตอร์ที่ทำหน้าที่หาเส้นทางและส่ง (forward) แพ็คเกตข้อมูลระหว่างเครือข่ายคอมพิวเตอร์ไปยังเครือข่ายปลายทางที่ต้องการ เราเตอร์ทำงานบนเลเยอร์ที่ 3 ตามมาตรฐานของ OSI Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.6 ตัวอย่าง Router

เราเตอร์มีลักษณะการใช้งานคล้ายกับสวิตช์ (Switch) ที่มีความสามารถแจกไอพีได้ โดยเราเตอร์สามารถเชื่อมต่อเข้ากับสองเส้นทางหรือมากกว่าจากเครือข่ายที่แตกต่างกัน เมื่อแพ็คเกตของข้อมูลเข้ามาจากเส้นทางหนึ่ง เราเตอร์จะอ่านข้อมูล address ที่อยู่ในแพ็คเกตเพื่อค้นหาปลายทางสุดท้าย จากนั้น ด้วยข้อมูลในตารางเส้นทางหรือนโยบายการส่ง จะส่งแพ็กเกตไปยังเครือข่ายข้างหน้าตามเส้นทางนั้น เราเตอร์จะดำเนินการ "กำกับการจราจร" บนเส้นทางนั้นด้วย

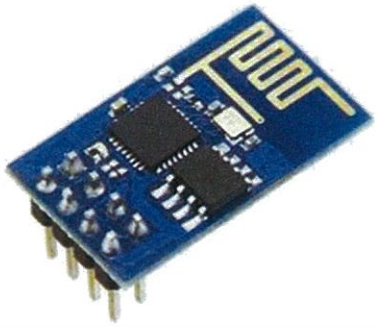
แพ็คเกตข้อมูลโดยทั่วไปจะถูกส่งจากเราเตอร์หนึ่งไปยังอีกเราเตอร์หนึ่งผ่านเครือข่ายที่เป็น Internetwork จนกว่าจะถึงโหนดปลายทาง เราเตอร์ประเภทที่คุ้นเคยมากที่สุดคือ เราเตอร์ที่บ้านและสำนักงานขนาดเล็ก ที่เพียงส่งผ่านข้อมูลเช่นหน้าเว็บ อีเมล์ IM และวิดีโอระหว่างเครื่องคอมพิวเตอร์ที่บ้านและอินเทอร์เน็ต เราเตอร์ดังกล่าวอาจเป็นเคเบิลโมเด็มหรือ DSL โมเด็มที่เชื่อมต่อกับอินเทอร์เน็ตผ่าน ISP เราเตอร์ที่มีความซับซ้อนมากขึ้นเช่นเราเตอร์ขององค์กรธุรกิจเชื่อมต่อกับธุรกิจขนาดใหญ่หรือกับเครือข่ายผู้ให้บริการอินเทอร์เน็ต เข้ากับคอร์เราเตอร์กำลังสูงที่สามารถส่งข้อมูลไปข้างหน้าด้วยความเร็วสูงตามแนวเส้นใยแก้วนำแสงของอินเทอร์เน็ตแบ็คโบน แม้ว่าเราเตอร์โดยปกติจะเป็นอุปกรณ์ที่ทำงานด้วยฮาร์ดแวร์ก็ตาม การใช้เราเตอร์ที่ทำงานด้วยซอฟต์แวร์มีการเจริญเติบโตมากขึ้น

2.1.6 ESP8266

ESP8266 คือ โมดูล wifi ที่ภายในมีเฟิร์มแวร์ทำงานในลักษณะ Serial-to-WiFi ที่ช่วยให้อุปกรณ์อื่น ๆ เช่น MCU สามารถต่อเข้ากับ internet ได้โดยใช้ port serial (ขา Tx, ขา Rx) และใช้คำสั่ง AT ในการควบคุมการทำงาน

ต่อมาผู้พัฒนาได้พัฒนาเฟิร์มแวร์ NodeMcu ให้เป็น platform และใช้ภาษา LUA ในการเขียนโปรแกรม ด้วยความที่เป็น platform ที่สะดวกต่อการใช้งาน ทางผู้พัฒนาจึงได้นำ NodeMcu (ESP8266) ใส่เป็นบอร์ดหนึ่งใน Arduino IDE และได้พัฒนาให้สามารถเขียนโปรแกรมด้วยภาษา C/C++ สำหรับผู้ที่ใช้งาน Arduino อยู่แล้วสามารถใช้งานบน Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.7 ESP-8266

ปัจจุบัน IoT(internet of things) กำลังมาแรง ESP8266 จึงเป็นที่นิยมเป็นอย่างมาก เพราะเป็นอุปกรณ์ที่มีขนาดเล็ก ราคาถูก ใช้งานง่ายเหมาะสำหรับนำมาพัฒนา IoT

2.2 ทฤษฎีเกียวกับ Software

2.2.1 Raspbian jessie

เป็นระบบปฏิบัติการที่มีฐานอยู่บนชุดของซอฟต์แวร์เสรีที่พัฒนาโดยอาสาสมัครภายใต้โครงการเดเบียน ภายใต้โครงการนี้มีเดเบียนลินุกซ์ (Debian GNU/Linux) ที่ใช้ลินุกซ์เป็นเคอร์เนล และใช้เครื่องมือต่าง ๆ ในโครงการ GNU ประกอบกันเป็นระบบปฏิบัติการ โดย Raspbian จะเป็นระบบปฏิบัติการเพื่อการใช้งานบน Raspberry Pi โดยเฉพาะ

เดเบียนมีชื่อเสียงในฐานะที่เป็นลินุกซ์ดิสทริบิวชันแรกที่ประสบความสำเร็จในการพัฒนาด้วยชุมชนโดยไม่มีเอกชนอยู่เบื้องหลัง มีการสร้างสัญญาประชาคมบทนิยามซอฟต์แวร์เสรีและนโยบายที่ชัดเจนทั้งทางเทคนิคและการบริหารงานกลายเป็นต้นแบบให้กับชุมชนอื่น ๆ ต่อมา รวมถึงปริมาณแพ็คเกตในโครงการมากกว่า 37,000 แพ็คเกตและรองรับสถาปัตยกรรมคอมพิวเตอร์มากกว่า 11 ชนิด ซึ่งครอบคลุมตั้งแต่คอมพิวเตอร์ฝังตัวไปจนถึงเมนเฟรม มีลินุกซ์ดิสทริบิวชันจำนวนมากที่นำเดเบียนไปพัฒนาต่อ

2.2.2 Node-RED

เป็นเครื่องมือสำหรับนักพัฒนาโปรแกรมในการเชื่อมต่ออุปกรณ์ฮาร์ดแวร์เข้ากับ APIs (Application Programming Interface) ซึ่งเป็นการพัฒนาโปรแกรมแบบ Flow-Based Programming ที่มีหน้า UI สำหรับนักพัฒนาให้ใช้งานผ่าน Web Browser ทำให้การเชื่อมต่อเส้นทางการไหลของข้อมูลนั้นง่ายขึ้น

เนื่องจาก Node-RED เป็น Flow-Based Programming นั้นทำให้เราแทบจะไม่ต้องเขียน Code ในการพัฒนาโปรแกรมเลย แค่เพียงเลือก Node มาวางแล้วเชื่อมต่อก็สามารถควบคุม I/O ได้ โดย Node-RED จะมี Node ให้เลือกใช้งานอย่างหลากหลาย สามารถสร้างฟังก์ชัน JavaScript

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้โดยใช้ Text Editor ที่มีอยู่ใน Node-RED และยังสามารถบันทึก Function, Templates, Flows เพื่อไปใช้งานกับงานอื่นต่อไป

Node-RED นั้นทำงานบน Node.js ทำให้เหมาะสำหรับการใช้งานกับ Raspberry Pi เนื่องจากใช้ทรัพยากรน้อย ขนาดไฟล์ไม่ใหญ่และ Node.js ยังทำหน้าที่เป็นตัวกลางให้ Raspberry Pi สามารถติดต่อกับ Web Browser และอุปกรณ์อื่น ๆ ได้

2.2.3 JSON

JSON หรือ Java Script Object Notation คือการนำภาษา Javascript มาใช้ในการแลกเปลี่ยนข้อมูลกับ Sever ได้ง่ายขึ้น โดยมีหลักการเขียนคล้ายกับภาษาพื้นฐานต่าง ๆ เช่น ภาษา C ภาษา C++ ภาษา Python ฯลฯ

โดย JSON จะมีโครงสร้างลักษณะเป็นรูปแบบคล้าย string จัดเก็บแบบอ่านค่าแล้วเข้าใจง่าย เสมือนรูปแบบ Array และสามารถจัดเก็บได้อีกแบบในรูปแบบของการจับคู่เรียงลำดับชื่อของชุดข้อมูลและค่าของชุดข้อมูลให้มีความสัมพันธ์กัน

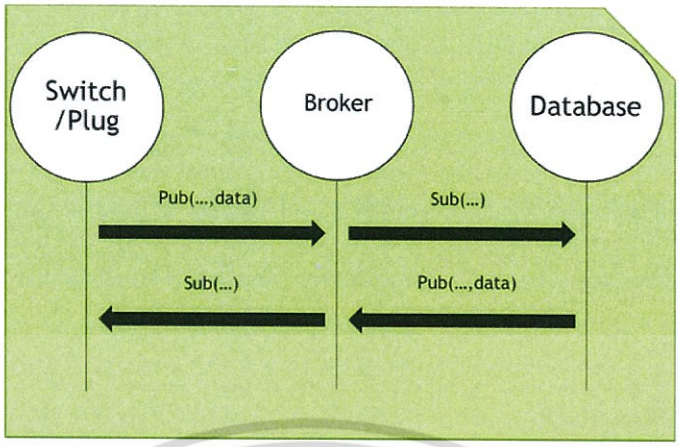
2.2.4 HTTP

HTTP หรือ HyperText Transfer Protocol คือ Protocol ชนิดหนึ่งที่อยู่บนชั้นของ Application Layer โดยจะมีลักษณะการทำงานในการร้องขอและการตอบรับระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่าย ซึ่งเครื่องลูกข่ายคือผู้ใช้ปลายทาง (end-user) และเครื่องแม่ข่ายคือเว็บไซต์ เครื่องลูกข่ายจะสร้างการร้องขอเอชทีทีพีผ่านทางเว็บเบราว์เซอร์ เว็บครอว์เลอร์ หรือเครื่องมืออื่น ๆ ที่จัดว่าเป็น ตัวแทนผู้ใช้ (user agent) ส่วนเครื่องแม่ข่ายที่ตอบรับ ซึ่งเก็บบันทึกหรือสร้าง ทรัพยากร (resource) อย่างเช่นไฟล์เอชทีเอ็มแอลหรือรูปภาพ จะเรียกว่า เครื่องให้บริการต้นทาง (origin server) ในระหว่างตัวแทนผู้ใช้กับเครื่องให้บริการต้นทางอาจมีสื่อกลางหลายชนิด อาทิ พร็อกซี เกตเวย์ และทุนเนล เอชทีทีพีไม่ได้จำกัดว่าจะต้องใช้ชุดเกณฑ์วิธีอินเทอร์เน็ต (TCP/IP) เท่านั้น แม้ว่าจะเป็นการใช้งานที่นิยมมากที่สุดบนอินเทอร์เน็ตก็ตาม โดยแท้จริงแล้วเอชทีทีพีสามารถนำไปใช้ได้บนโพรโทคอลอินเทอร์เน็ตอื่น ๆ หรือบนเครือข่ายอื่นก็ได้ เอชทีทีพีคาดหวังเพียงแค่การสื่อสารที่เชื่อถือได้ นั่นคือโพรโทคอลที่มีการรับรองเช่นนั้นก็สามารถใช้งานได้

2.2.5 MQTT

MQTT หรือ Message Queuing Telemetry Transport เป็น Protocol ที่อาศัยการทำงานผ่าน wifi ใช้ในอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก อาศัยช่วง Bandwidth ที่ต่ำใช้ในการส่ง โดยเป็นการเชื่อมต่อแบบ M2M (Machine to Machine) เหมาะแก่การใช้งาน เช่น IOT เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.8 MQTT

โดยการส่งจะอาศัยหลักการแบบ Publisher กับ Subscriber โดยจะมีลักษณะคล้ายกับหลักการที่ใช้บน Web Service โดยจะมี 2 ส่วนหลัก คือ Client กับ Broker

2.2.5.1 Broker

จะมีลักษณะแบบ Path ของ File คือจะสามารถแยกลำดับชั้นด้วยการเครื่องหมาย "/" เช่น fibo/floor3/room304/temperature ซึ่ง Client สามารถเลือกทำการ Publish, Subscribe เฉพาะ Topic หรือว่า Publisher/Subscriber หลาย ๆ Topic ในเวลาเดียวกันได้เลย โดยใช้ Wildcard

- Single-Level Wildcard (+) ตัวอย่างเช่น fibo/floor3/+/temperature หมายถึงการขอเขียนหรือรับข้อมูล tamperature จากทุก ๆ ห้องเลยใน fibo/floor3
- Multi-Level Wildcard (#) ตัวอย่างเช่น fibo/floor3/# หมายถึงการขอเขียนหรือรับข้อมูลทั้งหมดที่มีใน Topic ที่ขึ้นต้นด้วย fibo/floor3

2.2.5.2 Client

จะมี 2 ส่วนหลัก คือ Publisher กับ Subscriber โดยสามารถเป็นทั้ง 2 ส่วนได้ในเวลาเดียวกัน โดย Client จะต้องเป็นอุปกรณ์ที่สามารถรัน MQTT Client Library บน TCP/IP ได้ การใช้โมเดล Publisher/Subscriber นั้น หน้าที่คิดคำนวณจะถูกผลักไปยัง Broker Client จำเป็นที่จะต้องเปิดการเชื่อมต่อ TCP/IP ไว้ตลอดเวลาเพื่อที่ Broker จะสามารถส่งข้อมูลมาให้ได้ หากการเชื่อมต่อมีปัญหา Broker ก็จะเก็บข้อมูลนั้นไว้จนกว่า Client จะกลับมาออนไลน์อีกรอบ

2.2.6 React-Native

เป็น React ชนิดหนึ่ง แต่จะใช้ native components แทนที่ใช้ web components ตามปกติ ในส่วนของการสร้าง การจะเข้าใจและเรียนรู้การใช้งานของ React-Native นั้นจะต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าใจพื้นฐานของการทำงานของ React ก่อน ซึ่งมีลักษณะคล้ายกับ JSX (JavaScript ที่เอาไปรวมกับ XML) ทั้งในส่วนของ components state และ props

2.2.7 Android

Android คือ ระบบปฏิบัติการ หรือ OS ชนิดหนึ่งที่รากฐานมาจาก Linux โดยในอดีตถูกออกแบบมาใช้เป็นระบบปฏิบัติการของอุปกรณ์ที่ใช้การสัมผัส เช่น จอ Touchscreen บอร์ด Arduino เป็นต้น

2.2.7.1 User Interface

ส่วนต่อประสานกับผู้ใช้ (user interface) ของแอนดรอยด์ มีพื้นฐานอยู่บนอินเตอร์เฟซแบบไดเรกต์มานิพูเลชัน (Direct manipulation) ซึ่งจะใช้การสัมผัสที่สอดคล้องกับการกระทำในโลกความจริง เช่น การปัด การแตะ การกวาดนิ้ว รวมไปถึงการใช้นิ้วหมุนบนหน้าจอ การตอบสนองการสัมผัสนี้ได้รับการออกแบบมาอย่างดี และมักจะใช้การสั่นของอุปกรณ์ตอบโต้ว่าผู้ใช้ได้สัมผัสแล้ว ฮาร์ดแวร์เพิ่มเติมภายในเช่น เข็มทิศดิจิทัล ไจโรสโคป และ เซ็นเซอร์วัดแสง จะได้รับการนำมาใช้เพิ่มเติมในการตอบสนองต่าง ๆ กับผู้ใช้ เช่นการหมุนหน้าจอจากแนวตั้งเป็นแนวนอน หรือการเล่นเกมแข่งรถที่ต้องใช้การหมุนอุปกรณ์ เป็นต้น

อุปกรณ์แอนดรอยด์จะบูตเข้าหน้าหลัก ซึ่งเป็นหน้าจอหลักในการนำทางไปทุก ๆ ที่ในอุปกรณ์เหมือนกับเดสก์ท็อปบนเครื่องคอมพิวเตอร์ หน้าจอหลักของแอนดรอยด์จะสามารถวางไอคอนของแอปพลิเคชันและวิดเจ็ต โดยไอคอนของแอปพลิเคชันนั้นสามารถแตะเพื่อกดเข้าแอปพลิเคชันได้โดยตรง สภาพอากาศและพยากรณ์อากาศที่อัปเดตอยู่ตลอดเวลา กล่องขาเข้าของอีเมล รวมไปถึงหน้าจอข่าวด้วย หน้าจอหลักสามารถสร้างได้หลายหน้า โดยผู้ใช้สามารถปัดเพื่อเลื่อนไป-มาระหว่างหน้าได้ แม้ว่าหน้าจอหลักของแอนดรอยด์ที่จะสามารถให้ผู้ใช้สามารถปรับแต่งได้ตามต้องการ เพื่อที่จะให้ผู้ใช้รู้สึกดีตามรสนิยมของตนเอง แอปพลิเคชันอื่น ๆ มีให้ดาวน์โหลดบนกูเกิลเพลย์ และแอปหลายตัวสามารถที่จะเปลี่ยนรูปแบบหรือธีมของหน้าจอหลักได้ แม้กระทั่งการเปลี่ยนหน้าจอเลียนแบบระบบปฏิบัติการอื่น ๆ เช่น วินโดวส์โฟน ผู้ผลิตต่าง ๆ และผู้ให้บริการเครือข่ายบางรายจะปรับแต่งให้หน้าตาของหน้าจอหลักเปลี่ยนไปจากเดิมเพื่อให้แตกต่างจากคู่แข่งของพวกเขา

ด้านบนของหน้าจอจะเป็นแถบสถานะ ซึ่งจะแสดงถึงข้อมูลต่าง ๆ เกี่ยวกับอุปกรณ์และการเชื่อมต่อต่าง ๆ แถบสถานะสามารถดึงลงมาเพื่อที่จะแสดงการแจ้งเตือนบนหน้าจอเมื่อแอปพลิเคชันแจ้งเตือนข้อมูลหรือมีอัปเดต เช่น การได้รับข้อความใหม่ ในรุ่นก่อน ๆ ของแอนดรอยด์สามารถแตะที่การแจ้งเตือนเพื่อเปิดแอปพลิเคชันได้โดยตรง แต่รุ่นล่าสุดได้เพิ่มคุณสมบัติการทำงานที่มากขึ้น เช่น ความสามารถในการโทรกลับจากการแจ้งเตือนสายที่ไม่ได้รับโดยไม่ต้องเปิดแอปโทรศัพท์ การแจ้งเตือนจะหายไปเมื่อผู้ใช้อ่านหรือทำการลบการแจ้งเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7.2 แอปพลิเคชัน

แอนดรอยด์มีแอปพลิเคชันที่เติบโตขึ้นเรื่อย ๆ ซึ่งผู้ใช้สามารถซื้อและดาวน์โหลดได้จากกูเกิลเพลย์หรือแอมะซอนแอปสโตร์ และสามารถที่จะดาวน์โหลดไฟล์ APK ได้จากเว็บไซต์ต่าง ๆ แอปพลิเคชันจากเพลย์สโตร์อนุญาตให้ผู้ใช้สามารถดาวน์โหลดและอัปเดตได้จากกูเกิลและนักพัฒนาที่พัฒนาแอปนั้น ๆ รวมไปถึงความสามารถในการติดตั้งกับอุปกรณ์ที่สามารถเข้ากันได้กับแอปพลิเคชัน ซึ่งนักพัฒนาอาจจำกัดด้วยเหตุผลทางด้านอุปกรณ์ ประเทศ หรือเหตุผลทางธุรกิจ เมื่อซื้อแอปแล้วสามารถขอคืนเงินได้ภายใน 15 นาที หลังจากการดาวน์โหลด และบางผู้ให้บริการจะเก็บเงินด้วยใบเสร็จจากการซื้อแอปพลิเคชันบนกูเกิลเพลย์ ซึ่งจะคิดเงินเพิ่มเติมจากค่าใช้บริการรายเดือนปกติ ในเดือนกันยายน พ.ศ. 2555 แอปพลิเคชันสำหรับแอนดรอยด์มีมากถึง 675,000 แอปพลิเคชัน และมียอดดาวน์โหลดแอปพลิเคชันจากเพลย์สโตร์ทั้งหมด 2.5 พันล้านครั้ง

แอปพลิเคชันจะเขียนโดยใช้ภาษาจาวาและใช้แอนดรอยด์ซอฟต์แวร์เดเวล็อปเมนต์คิต (Android software development kit) หรือ SDK โดยเอสดีเคจะประกอบด้วยชุดเครื่องมือต่าง ๆ ในการพัฒนาแอปพลิเคชัน รวมไปด้วยตัวดีบัก แหล่งรวมซอฟต์แวร์ต่าง ๆ ตัวจำลองแฮนด์เซต โค้ดจำลอง และวิธีใช้ต่าง ๆ

ส่วนในประเทศจีนนั้น จะมีการจำกัดการใช้อินเทอร์เน็ตต่าง ๆ ของทางรัฐ โดยอุปกรณ์แอนดรอยด์ที่วางขายในประเทศจีนนั้นจะถูกจำกัดบริการบางอย่าง และจะมีเพียงแค่บริการที่ได้รับอนุมัติจากรัฐบาลแล้วเท่านั้น

2.2.7.3 การจัดการหน่วยความจำ

อุปกรณ์แอนดรอยด์นั้นจะมีการใช้งานแบตเตอรี ทำให้แอนดรอยด์ได้รับการออกแบบเพื่อจัดการหน่วยความจำหรือแรมสำหรับการใช้พลังงานที่น้อยที่สุด ในทางตรงข้ามกันกับคอมพิวเตอร์ตั้งโต๊ะ ซึ่งจะมีพลังงานให้ใช้ได้อย่างไม่จำกัด เมื่อแอปพลิเคชันของแอนดรอยด์ไม่ได้ใช้งาน ระบบจะจัดการจัดเก็บไว้ในหน่วยความจำ (เมื่อเปิดแอปพลิเคชันทิ้งไว้ในการใช้งาน)

แอนดรอยด์จะจัดการแอปพลิเคชันในหน่วยความจำอัตโนมัติ ซึ่งเมื่อแรมเหลือน้อย ระบบจะจัดการปิดแอปพลิเคชันและบริการต่าง ๆ ที่กำลังทำงานอยู่ทันที โดยกระบวนการนี้ผู้ใช้จะไม่สามารถมองเห็นได้ อย่างไรก็ตามจะมีแอปพลิเคชันบนกูเกิลเพลย์ที่จะสามารถจัดการและปิดแอปพลิเคชันได้ ซึ่งคาดกันว่าให้ผลร้ายมากกว่าผลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนา

การออกแบบระบบ Smart Home มีอุปกรณ์ 2 ส่วน คือ สวิทช์และปลั๊ก โดยสวิทช์จะเป็นอุปกรณ์ที่ทำหน้าที่สั่งการการทำงานของไฟและวัดค่าอุณหภูมิ ความชื้นและการเคลื่อนไหว ส่วนปลั๊ก เป็นอุปกรณ์สำหรับการควบคุมการทำงานของอุปกรณ์เครื่องใช้ไฟฟ้าต่าง ๆ ในส่วนของ sever นั้นถูกออกแบบให้เป็นอย่างง่ายและไม่ยุ่งยากโดยใช้ Node-RED

3.1 แนวคิดในการพัฒนา

ในส่วนของ Sever นั้นมีหน้าที่ในการรับส่งข้อมูลผ่าน MQTT Protocol โดยใช้ Node- RED ในการจัดการทางด้าน Backend ทั้งในส่วนของ Database และ Application โดยในส่วนของ Database ใช้ Json ในการเขียนโดยเก็บเป็น Files Local และส่วนของ Application นั้นใช้ React Native ในการเขียนและรับส่งข้อมูลเป็น HTTP Protocol และทาง Switch และ Plug นั้นทั้งคู่ทำงานโดยการรับส่งค่าผ่าน MQTT Protocol โดยในส่วนของทั้ง Switch และ Plug นั้นประกอบด้วย NodeMCU โดยใช้ ESP8266 เป็นหลัก และยังควบคุมการทำงานของอุปกรณ์ไฟฟ้าต่าง ๆ

3.2 เครื่องมือที่ใช้ในการพัฒนา

การพัฒนาระบบ Smart Home มีการกำหนดเทคโนโลยี เครื่องมือและภาษาที่ใช้ในการพัฒนาให้มีความเหมาะสม ดังนี้

3.2.1 ภาษาที่ใช้ในการพัฒนา

ภาษาที่ใช้ในการพัฒนาระบบ Smart Home สามารถแบ่งได้ตามการทำงาน 3 ส่วน คือ ส่วนแอพพลิเคชั่นบนสมาร์ทโฟน ส่วนดำเนินงานกับอุปกรณ์ฮาร์ดแวร์ และส่วนจัดการระบบหลังบ้าน มีการใช้ภาษาดังนี้

1) ภาษา C ใช้เขียนคำสั่งควบคุมอุปกรณ์ผ่านโปรแกรม Arduino IDE
2) ภาษา JAVASCRIPT ใช้เขียนคำสั่งควบคุมระบบหลังบ้านผ่าน Node-RED
3) JSON
   - ใช้ส่งข้อมูลระหว่างระบบหลังบ้านกับแอพพลิเคชั่น
   - ใช้เขียน0การทำงานของแอพพลิเคชั่นผ่าน react native

3.2.2 เทคโนโลยีที่ใช้ในการพัฒนา

1) React Native ใช้ในการสร้างแอพพลิเคชั่น
2) MQTT ใช้เป็น protocol ในการสื่อสารระหว่างระบบหลังบ้านกับอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 อุปกรณ์ที่ใช้ในการพัฒนา

1) Node-RED ใช้ในการออกแบบการทำงานของ Raspberry Pi
2) Android SDK (Software Development Kit) คือเครื่องมือที่ใช้สำหรับการพัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์ ซึ่งมีไลบรารีต่าง ๆ ที่จำเป็นต่อการพัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์
3) AVD (Android Virtual Device) คืออีมูเลเตอร์ในการรันโปรแกรมที่เขียนด้วยแอนดรอยด์ โดยตัวอีมูเลเตอร์จะจำลอง Virtual OS ของแอนดรอยด์รุ่นต่างๆมาไว้บนเครื่องคอมพิวเตอร์ของเรา
4) ADB (Android Debug Bridge) ใช้ในการติดตั้งแอพพลิเคชั่นลงบนโทรศัพท์มือถือแอนดรอยด์

3.3 ส่วนต่อประสานกับผู้ใช้

ส่วนต่อประสานกับผู้ใช้แอพพลิเคชั่นเป็นส่วนที่ผู้ใช้สามารถที่จะเรียกดูข้อมูลการทำงาน สั่งการการทำงาน ตั้งค่าระบบอัตโนมัติ โดยจะมี 2 ส่วน คือ ส่วนแอพพลิเคชั่นและส่วน dashboard บนหน้าจอสัมผัส

3.3.1 Application

แอพพลิเคชั่นได้ทำการสร้างและพัฒนาด้วย react native โดยมีหน้าแอพพลิเคชั่นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.1 หน้า Home

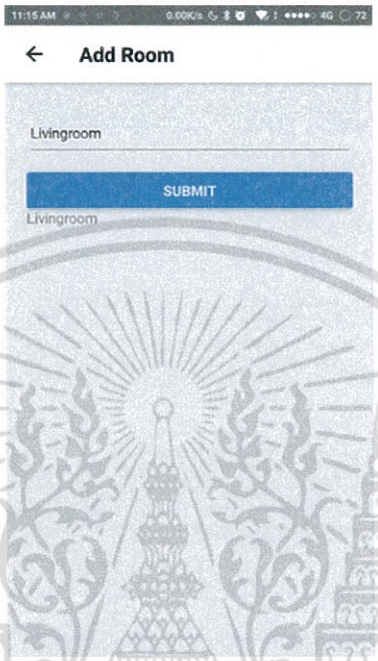
หน้าแรกจะปรากฏปุ่มเพื่อไปหน้าห้องที่สร้างไว้ และปุ่มลบห้องนั้น ๆ อยู่ด้านหลัง

รูป 3.1 หน้า Home

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2 หน้าการเพิ่มห้อง

หน้านี้จะปรากฏ Text input เพื่อกรอกชื่อห้องที่ต้องการสร้าง และมีปุ่ม Submit เพื่อยืนยัน แล้วจะกลับไปที่หน้าโฮมอีกครั้ง



รูป 3.2 หน้าสำหรับเพิ่มห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.3 หน้าอุปกรณ์ในห้อง

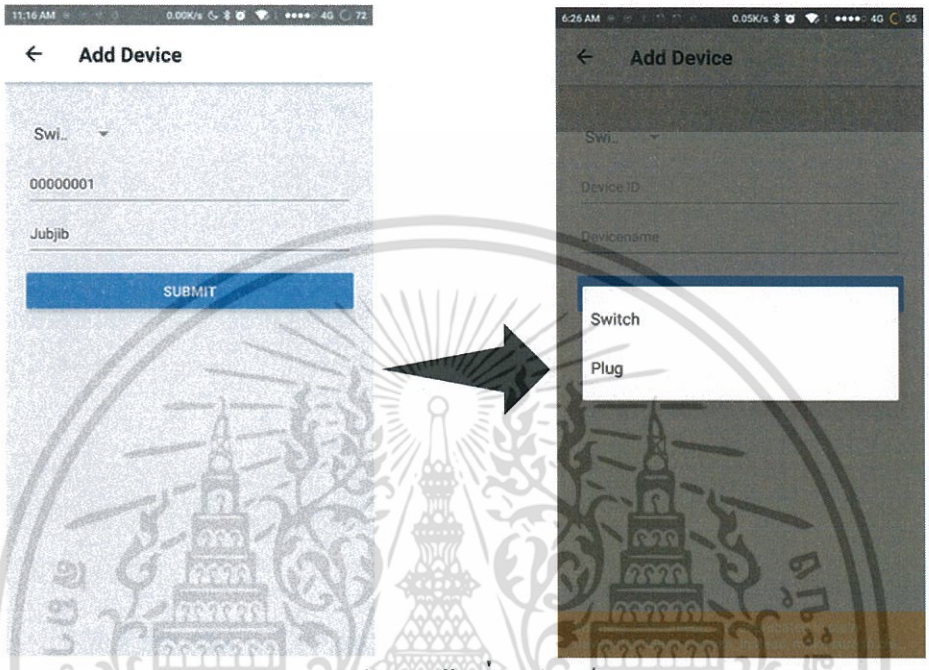
หน้านี้จะปรากฏ อุณหภูมิที่วัดจากอุปกรณ์ และข้อมูลของอุปกรณ์ ประกอบด้วย ชื่อสวิทช์ ปุ่มสั่งการและแสดงสถานะการทำงานของอุปกรณ์ ปุ่มตั้งค่าการทำงานอัตโนมัติของอุปกรณ์นั้นๆ และปุ่มลบอุปกรณ์นั้นออกจากห้อง และด้านล่างของหน้านี้ จะปรากฏปุ่มเพิ่มอุปกรณ์ในห้องนั้น

รูป 3.3 หน้าอุปกรณ์ในห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.4 หน้าเพิ่มอุปกรณ์

หน้าเพิ่มอุปกรณ์จะปรากฏ dropbox ให้เลือกว่าจะเพิ่มสวิทช์หรือปลั๊กดังรูป กรอกไอดีของอุปกรณ์ ชื่อของอุปกรณ์ที่จะตั้ง และปุ่ม Submit



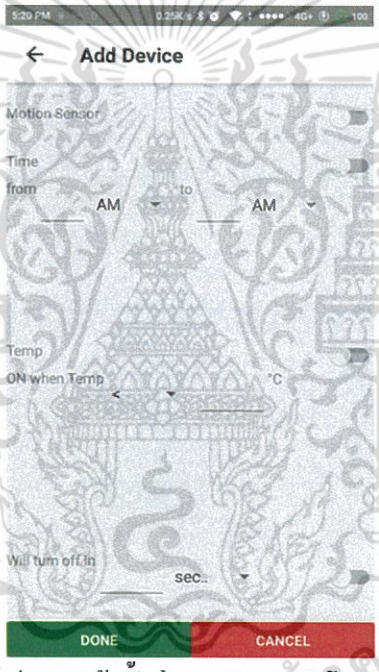
รูป 3.4 หน้าเพิ่มอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.5 หน้าตั้งค่าการทำงานอัตโนมัติ

หน้าตั้งค่าอัตโนมัติ จะปรากฏหน้าที่แบ่งเป็น 4 ส่วน

- ส่วน Motion sensor ประกอบด้วย สวิทช์เปิดปิด
- ส่วน Time ประกอบด้วย สวิทช์เปิดปิด และ text input ให้กรอกเวลา ที่จะให้ระบบทำงาน
- ส่วน Temp ประกอบด้วย สวิทช์เปิดปิด เครื่องหมาย < หรือ >= ให้เลือก และ text input ให้กรอกอุณหภูมิ
- ส่วน Delay ประกอบด้วย สวิทช์เปิดปิด text input ให้กรอกเวลาที่จะให้อุปกรณ์ปิดเมื่อไม่ตรงเงื่อนไข และเลือกหน่วยเวลาเป็น นาที หรือ วินาที



รูป 3.5 หน้าตั้งค่าการทำงานอัตโนมัติ

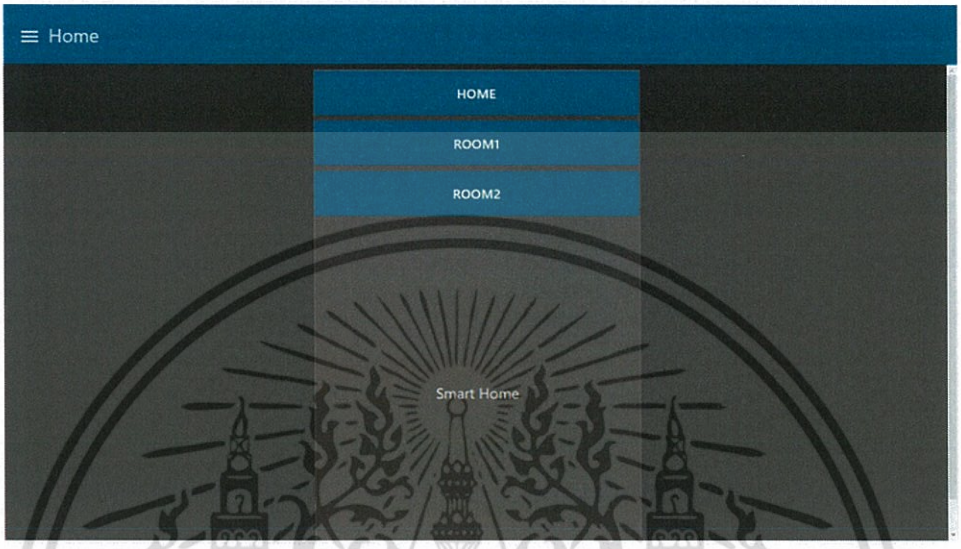
3.3.2 Dashboard

เป็นส่วนต่อประสานกับผู้ใช้ที่สร้างโดยการใช้ Node-RED ซึ่งใช้ในการเขียนการทำงานของ Raspberry Pi และยังสามารถนำมาใช้ในการสร้างหน้าส่วนต่อประสานกับผู้ใช้งานได้อีกด้วย โดยส่วน Dashboard มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.1 หน้า Home ของ Dashboard

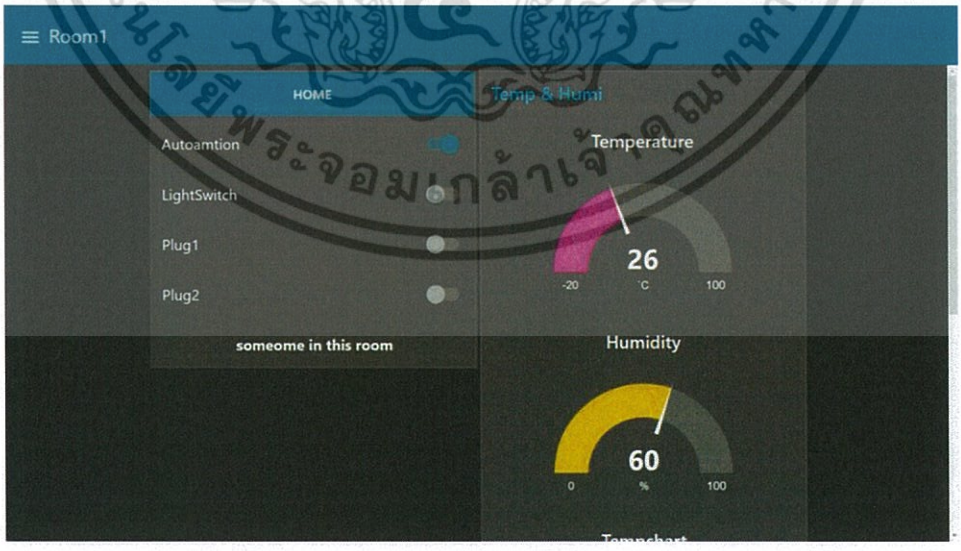
จากรูปแสดงหน้าแดชบอร์ดเริ่มต้น โดยจะมีปุ่ม ที่ลิ้งไปที่ หน้า Room1 และ Room2



รูป 3.6 หน้า Home ของ Dashboard

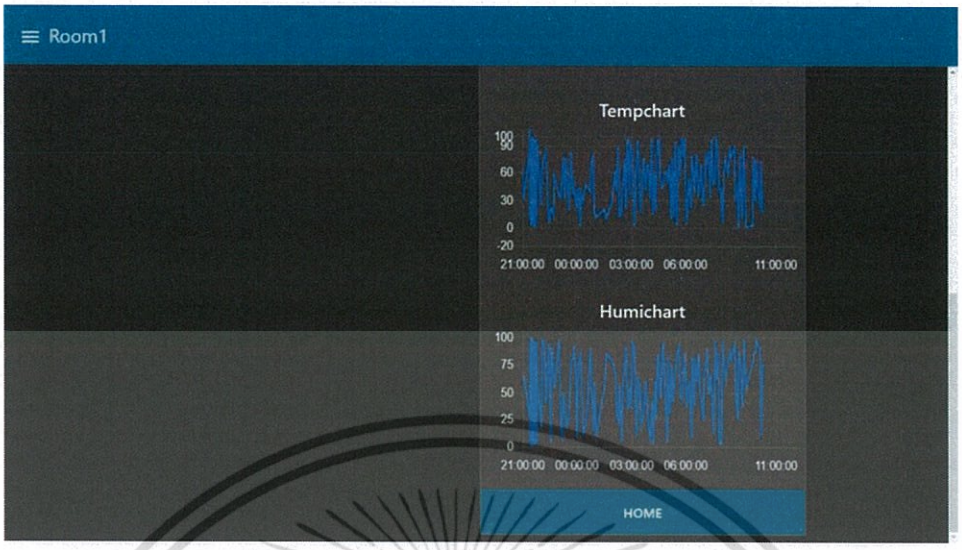
3.3.2.2 หน้า Room ของ Dashboard

จากรูป แสดงหน้าแดชบอร์ดที่สามารถควบคุมสวิทช์ และปลั๊กไฟที่เชื่อมต่อในห้อง Room1 และแสดงค่าของเซนเซอร์ที่รับได้จาก Smart Switch ได้



รูป 3.7 Guage ของหน้า Room ของ Dashboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.8 Graph ของหน้า Room ของ Dashboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โครงสร้างฮาร์ดแวร์

ส่วนฮาร์ดแวร์ของระบบ Smart Home จะมี 2 ประเภทคือ สวิทช์และปลั๊ก โดยสวิทช์เป็นอุปกรณ์ที่ทำหน้าที่เป็นเซ็นเซอร์และเป็นอุปกรณ์ที่ควบคุมการเปิด-ปิดไฟ และปลั๊กเป็นอุปกรณ์ที่ทำหน้าที่ควบคุมการทำงานของอุปกรณ์เครื่องใช้ไฟฟ้าต่างๆ

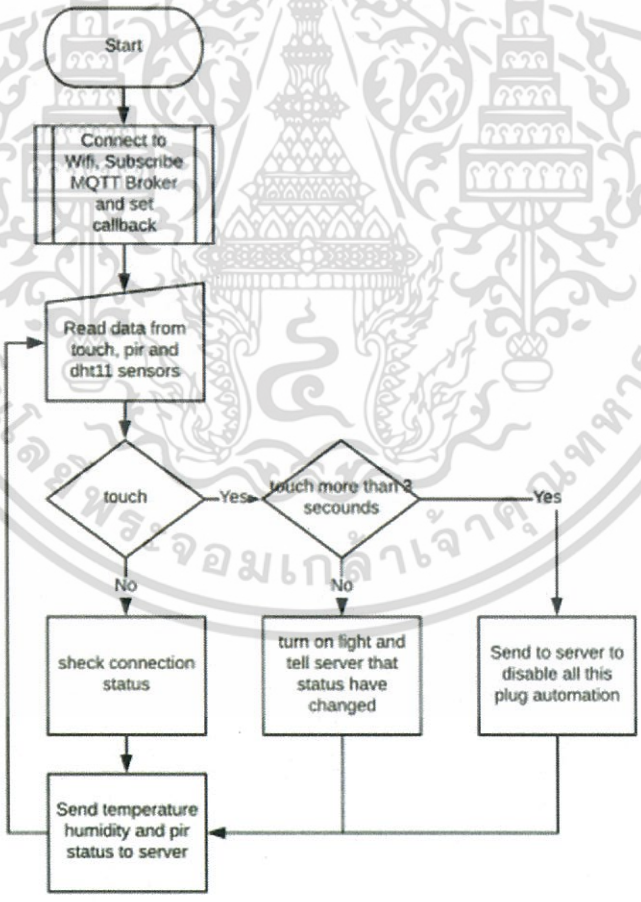
3.4.1 Arduino

การเขียนโปรแกรมเพื่อควบคุมการทำงานของสวิทช์ และปลั๊ก โดยใช้โปรแกรม Arduino

3.4.1.1 Switch

Flowchart

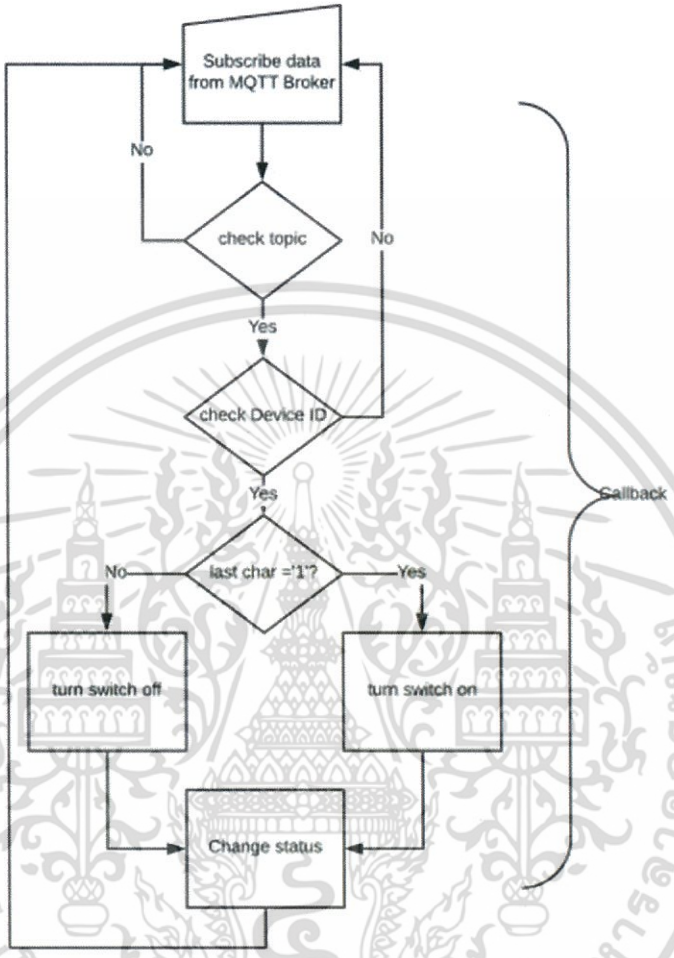
การออกแบบการทำงานของสวิทช์มีการทำงานดังนี้



รูป 3.9 การทำงานของสวิทช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของฟังก์ชั่น callback ในสวิทช์



รูป 3.10 การทำงานของฟังก์ชั่น callback ในสวิทช์

1) Source Code

โปรแกรม 3.1 การยกเลิกระบบอัตโนมัติของสวิทช์

```
for(int i = 0; i<30 && touch ;i++){//เช็คว่ากดนานเกิน3วินาทีหรือเปล่า
     touch = digitalRead(touchPin);
    if(i>=30){                    // ถ้าเกิน ให้ ส่งไปปิดการทำงานอัตโนมัติทั้งหมดของอุปกรณ์นี้

      strcat(message3,"c");
      client.publish("switch", message3);

     Serial.println("2");
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วนนี้คือจะเช็คว่ามีการกดที่สวิทช์สัมผัสหรือไม่ ถ้ามีแล้วแตะค้างนานกว่า 3 วินาที จะเป็นการกดเพื่อยกเลิกการทำงานอัตโนมัติของอุปกรณ์นั้นๆ

แต่ถ้าแตะไม่ถึง 3 วินาที ก็จะเป็นการกดเปิด ปิด สวิทช์ปกติ ดังโปรแกรม 3.2

โปรแกรม 3.2 การสั่งเปิด-ปิดไฟ ด้วยเซ็นเซอร์สัมผัส

```
else if(touch==0){  //ถ้าไม่เกิน ให้เช็คว่าสถานะของรีเลย์เป็นอย่างให้ ให้เปลี่ยนเป็นอีกอย่างหนึ่ง
       if(switstat){
         digitalWrite(relayPin, LOW);
         i = 30;

      Serial.println("2");
         strcat(message3, "0");
         client.publish("switch", message3);
       }
       else{
         digitalWrite(relayPin, HIGH);
         i = 30;
      Serial.println("2");
         strcat(message3, "1");
         client.publish("switch", message3);
   //ส่งข้อมูลสถานะของรีเลย์ให้เซิร์ฟเวอร์
       }
```

โปรแกรม 3.3 การตรวจสอบสถานะการเชื่อมต่อของสวิทช์

```
while (!client.connected()){//เช็คสถานการณ์เชื่อมต่อกับ wifi และ mqtt
      Serial.print("Attempting MQTT connection...");
  //Attempt to connect
      if (client.connect(CLIENT_ID)) {
      Serial.println("connected");
      client.publish("switch","connected");
      client.subscribe(topic);
      } else {
         Serial.print("failed, rc=");
         Serial.print(client.state());
         Serial.println(" try again in 5 seconds");
         delay(5000);
      }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโปรแกรม 3.3 เป็นโค้ดที่จะเช็คสถานะการเชื่อมต่อของทั้ง Wifi และ MQTT ว่าเชื่อมต่ออยู่หรือไม่ ถ้ายังเชื่อมต่ออยู่ก็จะดำเนินการโค้ดในส่วนอื่นๆต่อไป แต่ถ้าไม่มีการเชื่อมต่อ ก็จะทำการเชื่อมต่อใหม่ จนกว่าจะสามารถเชื่อมต่อได้

โปรแกรม 3.4 การอ่านค่าจากเซ็นเซอร์

```
chk = DHT.read(DHT11_PIN);      //อ่านค่าจากเซ็นเซอร์
    pirValue = digitalRead(pirPin);
```

โค้ดส่วนนี้ เป็นโค้ดที่ทำการอ่านค่าจากเซ็นเซอร์วัดอุณหภูมิ ความดัน และ เซ็นเซอร์ตรวจจับการเคลื่อนไหว

โปรแกรม 3.5 การส่งข้อมูลการตรวจจับการเคลื่อนไหว

```
    char message[50]="";
    char ir[10];
    if(second() % 1==0 && s!= second()){
 //ส่งข้อมูล motion sensor ทุกๆ 1 วินาที
     strcat(message, CLIENT_ID);
     strcat(message, " ");
     strcat(message, "pirs");
     strcat(message, " ");
     itoa(pirValue, ir, 10);
     strcat(message, ir);
     client.publish("switch", message);
      s = second();
    }
```

โค้ดนี้เป็นโค้ดที่ทำหน้าที่ส่งข้อมูลที่อ่านได้ จากเซ็นเซอร์จับการเคลื่อนไหว โดยจะส่งทุกๆ 1 วินาที และส่งผ่าน MQTT protocal โดยมี topic เป็น switch และมี payload เป็น CLIENT_ID +' pirs '+ค่าที่อ่านได้จากเซ็นเซอร์

โปรแกรม 3.6 การส่งข้อมูลอุณหภูมิและความชื้น

```
    char message2[50]="";
    char it[10];
    if(second() % 10==0 && t!= second()){
//ส่งข้อมูลอุณหภูมิและความชื้น ทุกๆ 10 วินาที
       strcat(message2, CLIENT_ID);
       strcat(message2, " ");
       strcat(message2, "tehu");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 3.6 การส่งข้อมูลอุณหภูมิและความชื้น(ต่อ)

```
        itoa(DHT.temperature, it, 10);
        strcat(message2, " ");
        strcat(message2, it);
        itoa(DHT.humidity, it, 10);
        strcat(message2, " ");
        strcat(message2, it);
        client.publish("switch", message2);
        t = second();
    }
```

โค้ดนี้เป็นโค้ดที่ทำหน้าที่ส่งข้อมูลที่อ่านได้ จากเซ็นเซอร์วัดอุณหภูมิและความชื้น โดยจะส่งทุกๆ 10 วินาที และส่งผ่าน MQTT protocal โดยมี topic เป็น switch และมี payload เป็น CLIENT_ID +' tehu '+ค่าอุณหภูมิที่อ่านได้+' '+ ค่าความชื้นที่อ่านได้

จากการเขียนโปรแกรมควบคุมการทำงานให้เมื่อบอร์ดได้รับค่าจาก MQTT Broker บอร์ดจะเรียกใช้ฟังก์ชั่น callback

โปรแกรม 3.7 การรับคำสั่งจาก NodeRed ของสวิทช์

```
void callback(char* topic, byte* payload,
unsigned int length) {
int i = 0 ,j = 0;
     char Number[10]="";
     if(String(topic) == "switch" && length <=10){
   //เช็ค topic ว่าใช่ของอุปกรณ์นี้หรือเปล่า
          for(i=0;  payload[i] != ' '; i++) {
               message_buff[i] = payload[i];
     // นับจำนวนตัวจนถึง " "
          }
          for (j = 0; j < i; j++){
               Number[j] = payload[j];
     // เก็บตัวก่อนหน้า " " ใส่ใน array Number
          }
          Number[j] = '\0';
          if(!strcmp(Number,Num_device)&&
            length== 10){
     // เช็คว่า ตัวที่ได้ตรงกับรหัสของบอร์ดนี้หรือไม่
               i++;
          if(payload[i]== '1'){
       //เช็คตัวหลังจาก " " ถ้าเป็น '1' ให้เปิดไฟ และปริ้น
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 3.7 การรับคำสั่งจาก NodeRed ของสวิทช์(ต่อ)

```
                    //"Switch status : 1"
                digitalWrite(relayPin, HIGH);
                Serial.println("OK");
                switstat = 1;
                Serial.print("Switch status: ");
                Serial.println(switstat);
            }
            else if(payload[i]== '0'){
        //ถ้าเป็น '0' ให้ปิดไฟ และปริ้น
        //"Switch status : 0"
                digitalWrite(relayPin, LOW);
                switstat = 0;
                Serial.print("Switch status: ");
                Serial.println(switstat);
            }
            }
            }
        }
}
```

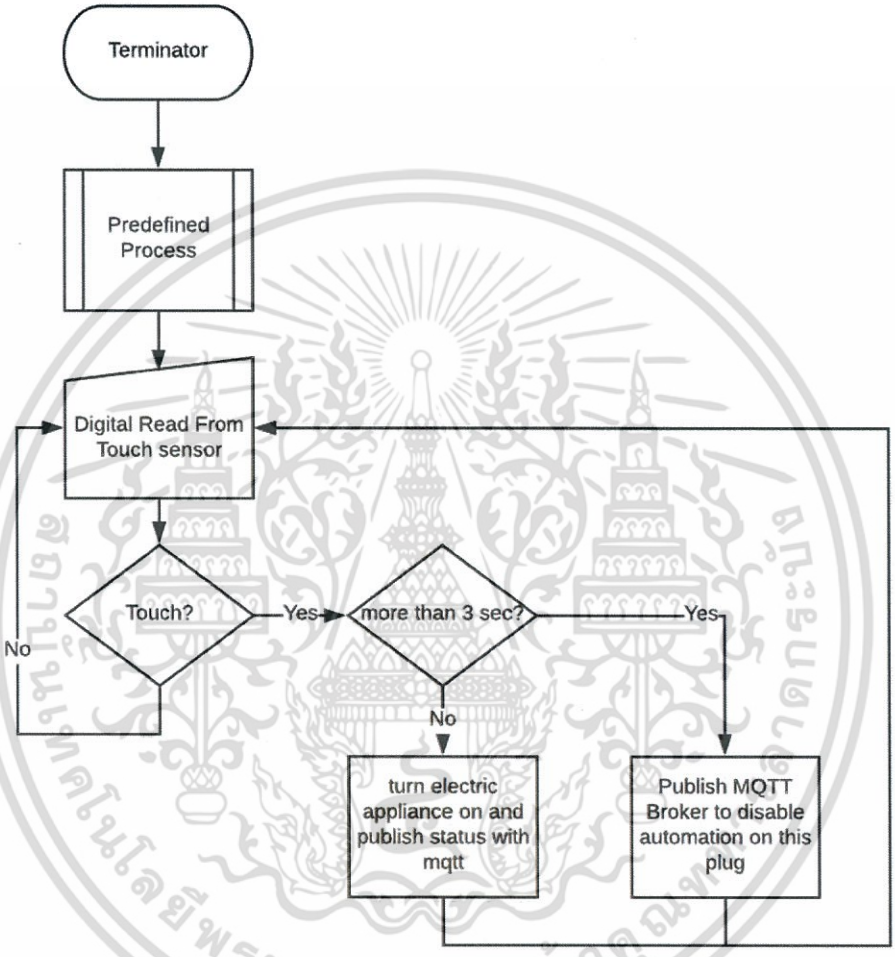
โดยฟังก์ชั่น callback จะทำงานเมื่อมีการส่งข้อมูลใน MQTT broker ฟังก์ชั่นจะทำการตรวจสอบว่าข้อมูลที่ส่งมา เป็นของอุปกรณ์นั้นหรือไม่ โดยจะเช็คจาก topic ที่ส่งมา ว่าตรงกับ switch หรือไม่ และ payload 8 บิทแรก โดยนำมาเทียบกับไอดีของอุปกรณ์ เมื่อตรงกับอุปกรณ์ของตนแล้ว จึงอ่านบิทถัดไปเพื่อสั่งเปิด-ปิดการทำงานของ Relay ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1.2 Plug

1) Flowchart

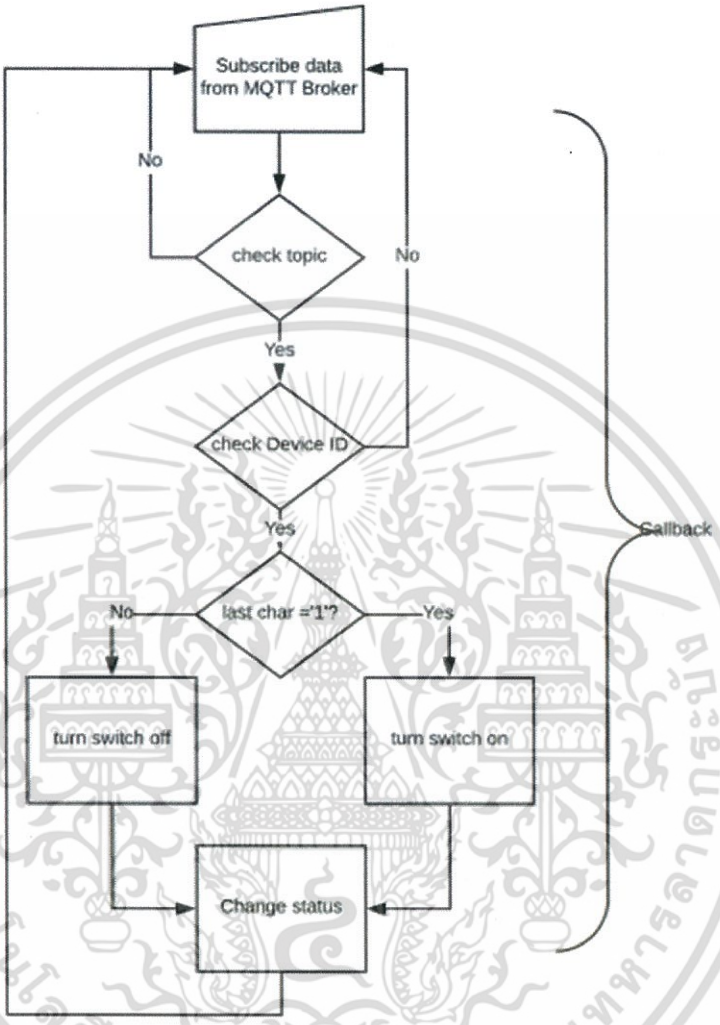
การทำงานของปลั๊ก มีการทำงานดังนี้



รูป 3.11 การทำงานของปลั๊ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของฟังก์ชั่น callback ในปลั๊ก



รูป 3.12 การทำงานของฟังก์ชั่น callback ในปลั๊ก

2) Source code

โปรแกรม 3.8 การยกเลิกระบบอัตโนมัติของปลั๊ก

```
for(int i = 0; i<=30 && touch ;i++){


    touch = digitalRead(touchPin);
    if(i>=30){

      strcat(message,"c");
      client.publish("plug", message);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนนี้คือจะเช็คว่ามีการกดที่สวิทช์สัมผัสหรือไม่ ถ้ามีแล้วแตะค้างนานกว่า 3 วินาที จะเป็นการกดเพื่อยกเลิกการทำงานอัตโนมัติของอุปกรณ์นั้นๆ

แต่ถ้าแตะไม่ถึง 3 วินาที ก็จะเป็นการกดเปิด ปิด สวิทช์ปกติ ดังโค้ดข้างล่าง

โปรแกรม 3.9 การสั่งเปิด-ปิดเครื่องใช้ไฟฟ้า ของปลั๊ก

```
else if(touch==0){
      if(plugstat){
        digitalWrite(relayPin, LOW);
        i = 30;
        strcat(message, "0");
        client.publish("plug", message);
      }
      else{
        digitalWrite(relayPin, HIGH);
        i = 30;
        strcat(message, "1");
        client.publish("plug", message);
      }
}
```

โปรแกรม 3.10 การตรวจสอบสถานะการเชื่อมต่อของปลั๊ก

```
while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
 //Attempt to connect
    if (client.connect(CLIENT_ID)) {
        Serial.println("connected");
        client.publish("switch","connected");
        client.subscribe(topic);
    } else {
        Serial.print("failed, rc=");
        Serial.print(client.state());
        Serial.println(" try again in 5 seconds");
        delay(5000);
    }
  }
```

จากโปรแกรม 3.10 เป็นโค้ดที่จะเช็คสถานะการเชื่อมต่อของทั้ง Wifi และ MQTT ว่าเชื่อมต่ออยู่หรือไม่ ถ้ายังเชื่อมต่ออยู่ก็จะดำเนินการโค้ดในส่วนอื่นๆต่อไป แต่ถ้าไม่มีการเชื่อมต่อ ก็จะทำการเชื่อมต่อใหม่ จนกว่าจะสามารถเชื่อมต่อได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 3.11 การส่งข้อมูลสถานะของเครื่องใช้ไฟฟ้า

```
if(minute() % 1==0 && t!= minute()){
strcat(message2, CLIENT_ID);
     strcat(message2, " ");
     itoa(plugstat, it, 10);
     strcat(message2, it);
     client.publish("plug", message2);
     t = minute();
}
Serial.print("sec =");
Serial.println(second());
delay(100);
client.loop();
```

โปรแกรม 3.11 จะทำการส่งสถานะการทำงานของ รีเลย์ ด้วยทุกๆ 1 นาที โดยมีรูปแบบการส่งเป็น topic plug มี payload เป็น CLIENT_ID +' '+สถานะของรีเลย์ แล้วทำการวนลูป

จากการเขียนโปรแกรมควบคุมการทำงานให้เมื่อบอร์ดได้รับค่าจากMQTT Broker บอร์ดจะเรียกใช้ฟังก์ชั่น callback

โปรแกรม 3.12 การรับคำสั่งจาก NodeRed ของปลั๊ก

```
void callback(char* topic, byte* payload,
unsigned int length) {
int i = 0 ,j = 0;
     char Number[10]="";
     if(String(topic) == "plug" && length <=10){
   //เช็ค topic ว่าใช่ของอุปกรณ์นี้หรือเปล่า
          for(i=0;  payload[i] != ' '; i++) {
               message_buff[i] = payload[i];
    // นับจำนวนตัวจนถึง " "
          }
          for (j = 0; j < i; j++){
               Number[j] = payload[j];
    // เก็บตัวก่อนหน้า " " ใส่ใน array Number
          }
          Number[j] = '\0';
          if(!strcmp(Number,Num_device)&&
            length== 10){
    // เช็คว่า ตัวที่ได้ตรงกับรหัสของบอร์ดนี้หรือไม่
               i++;
          if(payload[i]== '1'){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม 3.12 การรับคำสั่งจาก NodeRed ของปลั๊ก(ต่อ)

```
        //เช็คตัวหลังจาก " " ถ้าเป็น '1' ให้เปิดไฟ และปริ้น
                        //"Switch status : 1"
                    digitalWrite(relayPin, HIGH);
                    Serial.println("OK");
                    plugstat = 1;
                    Serial.print("Plug status: ");
                    Serial.println(plugstat);
                }
                else if(payload[i]== '0'){
        //ถ้าเป็น '0' ให้ปิดไฟ และปริ้น
        //"Switch status : 0"
                    digitalWrite(relayPin, LOW);
                    plugstat = 0;
                    Serial.print("Plug status: ");
                    Serial.println(plugstat);
                }
                }
                }
        }
}
```

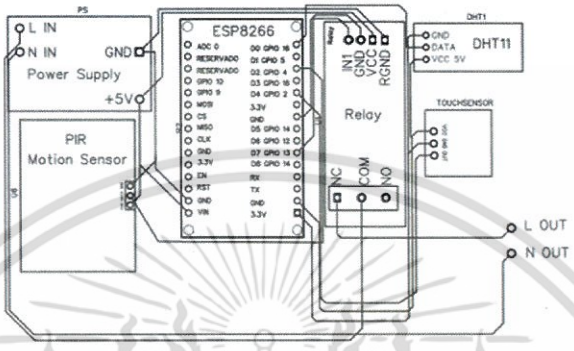
โดยฟังก์ชั่น callback จะทำงานเมื่อมีการส่งข้อมูลใน MQTT broker ฟังก์ชั่นจะทำการตรวจสอบว่าข้อมูลที่ส่งมา เป็นของอุปกรณ์นั้นหรือไม่ โดยจะเช็คจาก topic ที่ส่งมา ว่าตรงกับ plug หรือไม่ และ payload 8 บิทแรก โดยนำมาเทียบกับไอดีของอุปกรณ์ เมื่อตรงกับอุปกรณ์ของตนแล้ว จึงอ่านบิทถัดไปเพื่อสั่งเปิด-ปิดการทำงานของ Relay ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 PCB

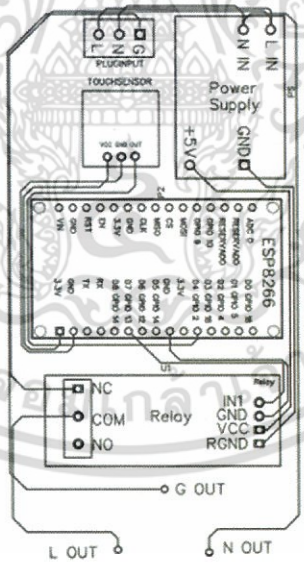
หลังจากออกแบบวงจรทั้งสองส่วนแล้ว จึงนำวงจรมาออกแบบลายวงจรผ่านเว็บไซต์ easyeda.com เพื่อทำแผ่น PCB

3.4.2.1 Switch



รูป 3.13 PCB ของ Switch

3.4.2.2 Plug



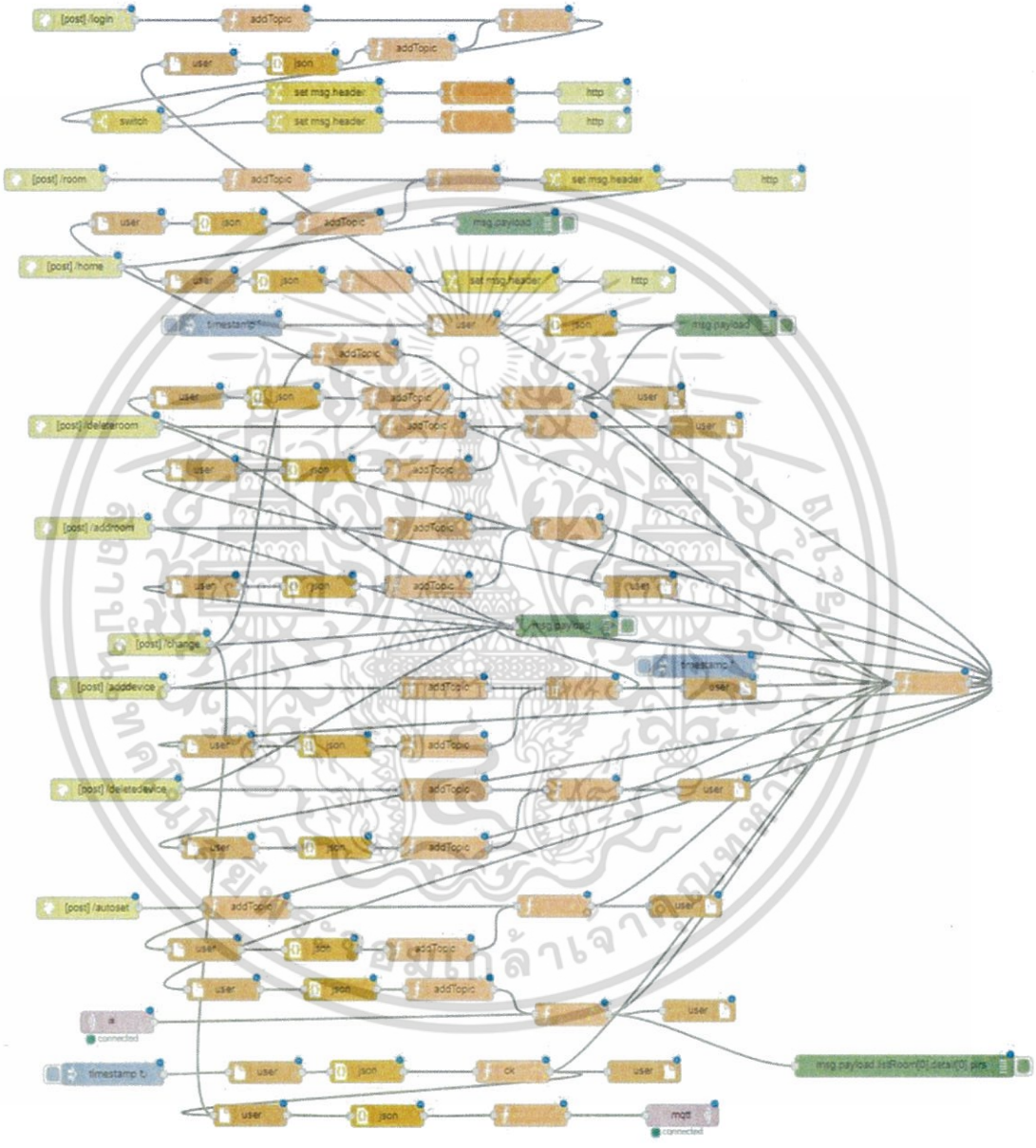
รูป 3.14 PCB ของ Plug

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ส่วนจัดการระบบ

3.5.1 Node-RED

3.5.1.1 Backend



รูป 3.15 ภาพรวมโฟลวการทำงานของ Node-RED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.2 การเรียกหน้า Home

จะทำงานเมื่อมีการส่งคำร้องขอ POST method /home ก็จะทำการสั่งให้ไปเรียกดูข้อมูลจากไฟล์ .JSON ที่กำหนด แล้วนำมาเลือกส่งข้อมูลห้องทั้งหมดที่อยู่ในผู้ใช้งานนี้ แล้วรีเทิร์นกลับไปทาง http

รูป 3.16 โฟลวการส่งคำร้องขอสำหรับเปิดหน้า Home

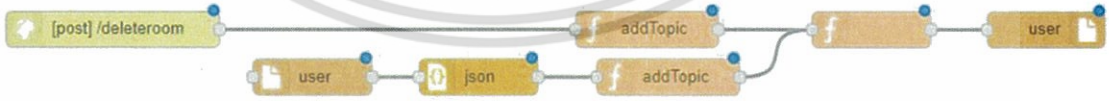
3.5.1.3 การเรียกดูหน้า Room

จะทำงานเมื่อมีการส่งคำร้องขอ POST method /room โดยส่งชื่อห้องที่กดไป ก็จะทำการสั่งให้ไปเรียกดูข้อมูลจากไฟล์ .JSON ที่กำหนด แล้วนำมาเทียบหาชื่อห้องเดียวกันกับที่ request ส่งมา แล้วรีเทิร์นข้อมูลอุปกรณ์ทั้งหมดที่อยู่ในห้องนั้นกลับไปทาง http

รูป 3.17 โฟลวการส่งคำร้องขอสำหรับเปิดหน้า Room

3.5.1.4 การลบ Room

จะทำงานเมื่อมีการส่งคำร้องขอ POST method /deleteroom โดยส่งชื่อห้องที่กดไป ก็จะทำการสั่งให้ไปเรียกดูข้อมูลจากไฟล์ .JSON ที่กำหนด แล้วนำมาเทียบหาชื่อห้องเดียวกันกับที่ request ส่งมาแล้วทำการลบข้อมูลที่ห้องนั้นทั้งหมดออกจากไฟล์

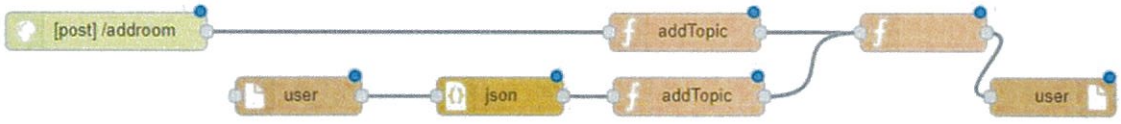


รูป 3.18 โฟลวการสั่งการสำหรับลบ Room

3.5.1.5 การเพิ่ม Room

จะทำงานเมื่อมีการส่งคำร้องขอ POST method /addroom โดยส่งชื่อห้องที่ต้องการสร้างมาด้วย ก็จะทำการสั่งให้ไปเรียกดูข้อมูลจากไฟล์ .JSON ที่กำหนด แล้วเพิ่มห้องใหม่เข้าไป โดยมีชื่อตามที่ส่ง request มา แล้วทำการอัพเดตข้อมูลในไฟล์นั้นใหม่

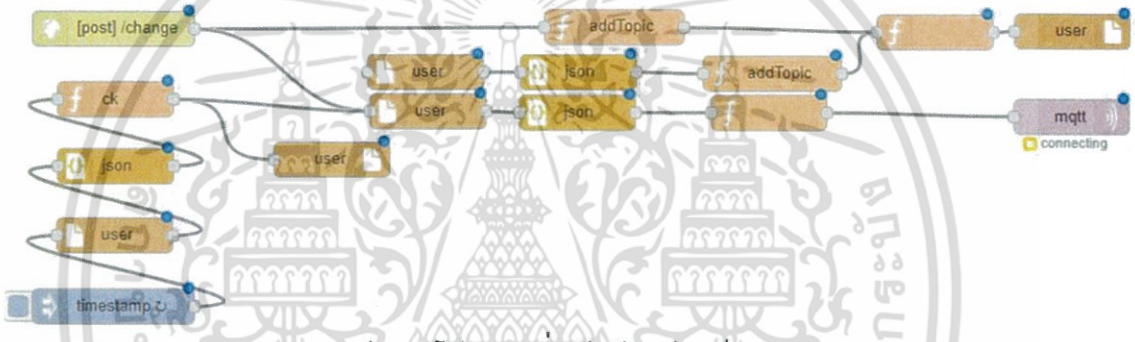
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.19 โฟลวการสั่งการสำหรับเพิ่ม Room

3.5.1.6 การสั่งเปิดปิดอุปกรณ์

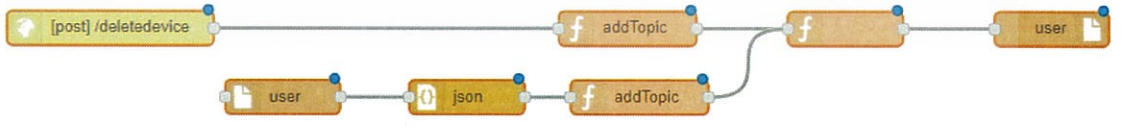
จะทำงานเมื่อมีการส่งคำร้องขอ POST method /change โดยส่งข้อมูลชื่อห้องที่อุปกรณ์นั้นๆอยู่ ไอดีของอุปกรณ์ ประเภทของอุปกรณ์ และสถานะที่ต้องการให้เป็น ก็จะทำการ 2 ส่วน คือ สั่งให้ไปเรียกดูข้อมูลจากไฟล์ .JSON ที่กำหนด แล้วนำมาเทียบหาห้องเดียวกัน รหัสอุปกรณ์เดียวกัน และประเภทอุปกรณ์เดียวกัน แล้วทำการอัพเดตข้อมูลสถานะที่ต้องการให้เป็น



รูป 3.20 โฟลวการสั่งเปิดปิดอุปกรณ์

3.5.1.7 การลบ Device

จะทำงานเมื่อมีการส่งคำร้องขอ POST method /deletedevice โดยส่งชื่อห้อง ไอดีอุปกรณ์ และประเภทของอุปกรณ์ที่กดไป ก็จะทำการสั่งให้ไปเรียกดูข้อมูลจากไฟล์ .JSON ที่กำหนด แล้วนำมาเทียบหาชื่อห้องเดียวกัน ไอดีอุปกรณ์เดียวกัน และประเภทอุปกรณ์เดียวกับที่ request ส่งมาแล้วทำการลบข้อมูลของอุปกรณ์นั้นจากไฟล์

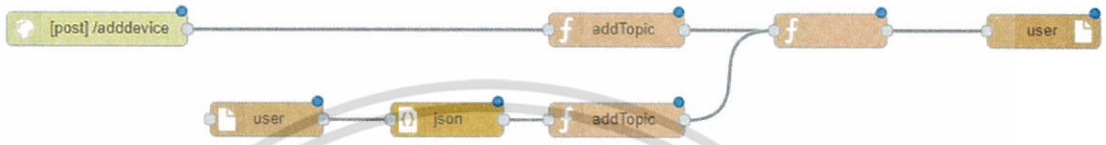


รูป 3.21 โฟลวการลบ Device

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.8 การเพิ่ม Device

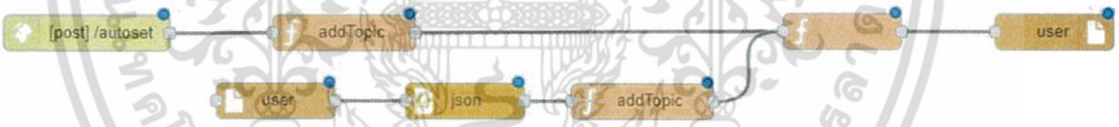
จะทำงานเมื่อมีการส่งคำร้องขอ POST method /adddevice โดยส่งชื่อห้อง ไอดีอุปกรณ์ ประเภทของอุปกรณ์ และชื่ออุปกรณ์ที่ตั้งขึ้นมา ก็จะทำการสั่งให้ไปเรียกดูข้อมูลจากไฟล์ .JSON ที่กำหนด แล้วเทียบหาห้องที่ส่งมาแล้วเพิ่มข้อมูลอุปกรณ์ที่ส่งมาทั้งหมดลงไป โดยมีชื่อตามที่ส่ง request มา แล้วทำการอัพเดตข้อมูลในไฟล์นั้นใหม่



รูป 3.22 โฟลวการเพิ่ม Device

3.5.1.9 การตั้งค่าการทำงานอัตโนมัติ

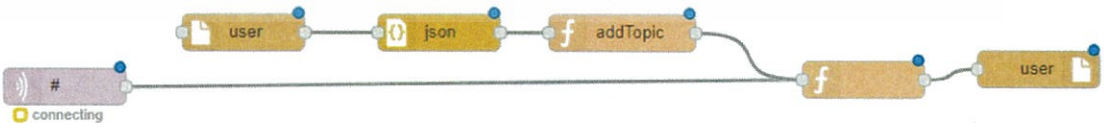
จะทำงานเมื่อมีการส่งคำร้องขอ POST method /autoset โดยส่งชื่อห้อง ไอดีอุปกรณ์ ประเภทของอุปกรณ์ ชื่ออุปกรณ์ ข้อมูลการตั้งค่าการทำงานอัตโนมัติ ก็จะทำการสั่งให้ไปเรียกดูข้อมูลจากไฟล์ .JSON ที่กำหนด แล้วเทียบหาห้องที่ส่งมา ไอดีอุปกรณ์ ประเภทอุปกรณ์ แล้วอัพเดตข้อมูลการทำงานอัตโนมัติของอุปกรณ์ลงไป



รูป 3.23 โฟลวการตั้งค่าการทำงานอัตโนมัติ

3.5.1.10 การเปลี่ยนค่าในฐานข้อมูลเมื่อรับค่าจาก MQTT

ทำงานเมื่อได้รับการส่งข้อมูลผ่าน MQTT Broker จะทำการสั่งให้ไปเรียกดูข้อมูลจากไฟล์ .JSON ที่กำหนดแล้วอัเดตข้อมูล ในดาต้าเบส

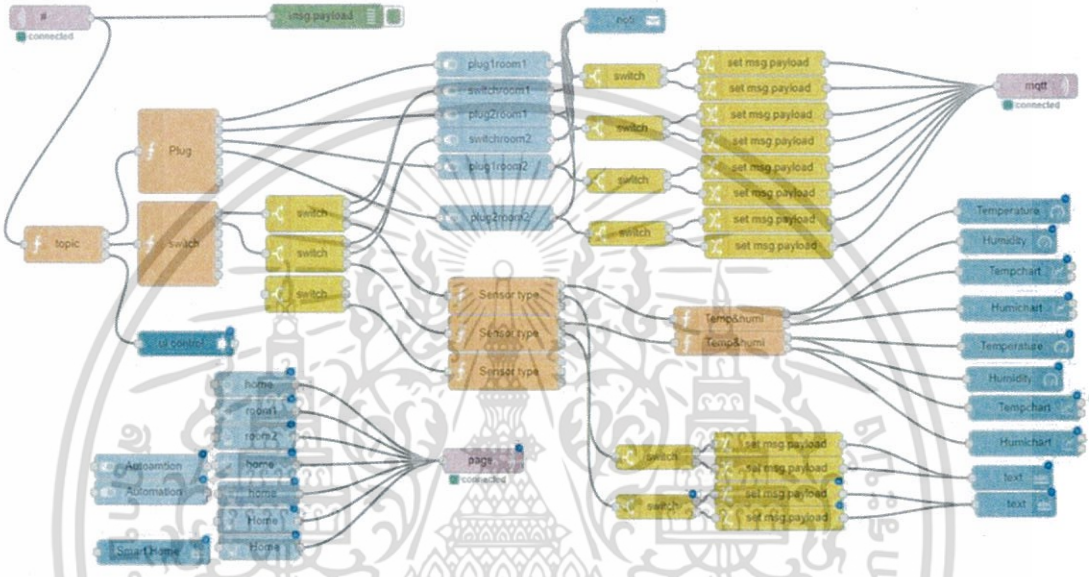


รูป 3.24 โฟลวการเปลี่ยนค่าในฐานข้อมูลเมื่อรับค่าจาก MQTT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1.11 Dashboard

จากรูป แสดงการออกแบบทำงานใน ส่วนของ Dashboard บนหน้าจอ touchscreen ผ่านโปรแกรม NodeRED โดยจะรับข้อมูลผ่าน mqtt broker แล้วนำมาตรวจสอบ ว่าเป็น topic ใด หลังจากผ่านการกรอง topic แล้ว ก็จะตรวตสอบว่ามีเลข device id เป็นเท่าไร แล้วจึงแยกกันไปทำงานตามที่ออกแบบไว้



รูป 3.25 โฟลวการทำงาน Dashboard

โดยส่วนของ topic จะมีทั้งหมด 3 topic ได้แก่ “plug” , “switch” และ “page”

“plug” คือข้อมูลที่ถูกส่งมาจาก Smart plug โดยข้อมูลจะมีรูปแบบเดียว คือการสั่งเปิดและปิดสวิทช์ โดยจะมี payload เป็น device id ตามด้วย คำสั่งเปิด หรือปิด

“switch” คือข้อมูลที่ถูกส่งมาจาก Smart switch โดยข้อมูลจะมี 2 รูปแบบ

1) การส่งข้อมูลจากเซนเซอร์ โดยจะมี payload เป็น device id ตามด้วยชื่อเซนเซอร์ และค่า ตามลำดับ แล้วนำมาแสดงในรูปแบบของกราฟในหน้าแดชบอร์ด
2) การสั่งการการทำงานของสวิทช์ไฟ จะมี payload เป็น device id ตามด้วย คำสั่ง เปิด-ปิด

“page” คือข้อมูลที่ถูกส่งมาจาก ตัว Raspberry Pi เอง ผ่านการกดปุ่มต่างๆ ผ่าน dashboard โดยจะมีไว้เพื่อควบคุมการแสดงผลdashboard มี payload เป็น ลำดับหน้าของ dashboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดสอบระบบต่างๆเช่น ทดสอบแอพพลิเคชั่นโดยนำแอพพลิเคชั่นไปติดตั้งในโทรศัพท์มือถือแอนดรอยด์รุ่น Xiaomi mi 5s ทดสอบการทำงานของ NodeRed และทดสอบการทำงานของอุปกรณ์สวิทช์และปลั๊ก

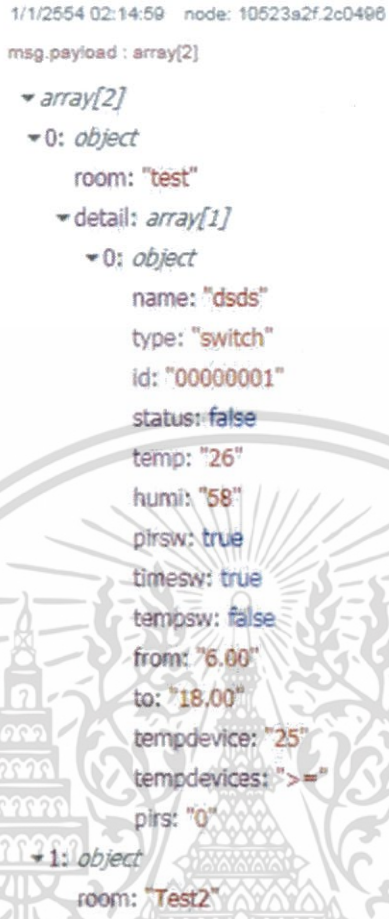
4.1 การทดลองส่วน Application และ Server

การทดลองส่วนแอพพลิเคชั่นจะทำการลงแอพพลิเคชั่นในโทรศัพท์มือถือรุ่น Xiaomi mi 5s ซึ่งใช้ระบบปฏิบัติการแอนดรอยด์ เวอร์ชั่น 7.0 (Nougat) และมีขนาดหน้าจอ 5.15 นิ้ว

4.1.1 การเรียกดูหน้า Home

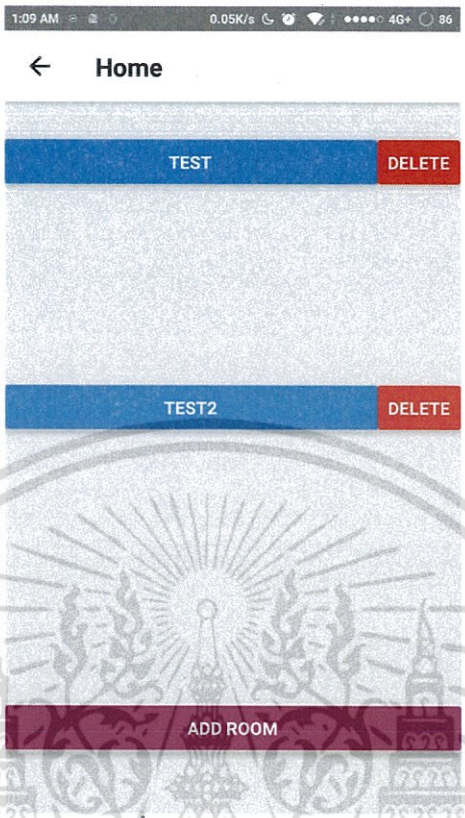
การทดลองเรียกดูหน้า Home เป็นการทดลองการเรียกข้อมูลจาก NodeRed ผ่าน POST method ว่าจะส่งข้อมูลอะไรกลับมา แล้วได้ผลลัพธ์บนหน้าแอพพลิเคชั่นอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.1 ผลลัพท์การส่งคำร้องขอของหน้า Home

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.2 ผลลัพท์ที่ application แสดงในหน้า Home

ผลลัพท์ที่ application แสดง

4.1.2 การเรียกดูหน้า Room

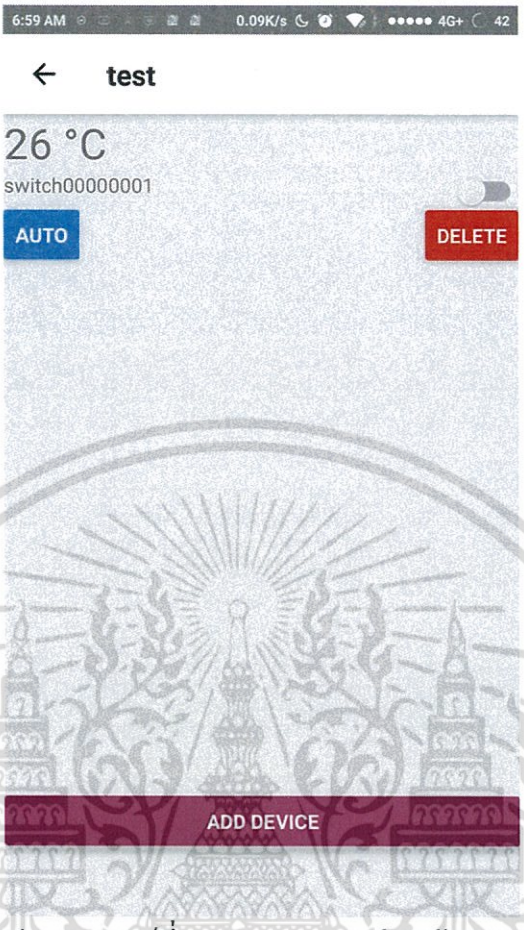
การทดลองเรียกดูหน้า Room เป็นการทดลองเรียกดูหน้า Room โดยจะรีเควส POST method แล้วดูว่า ได้ส่งอะไรไปบ้างและ NodeRed จะส่งข้อมูลอะไรกลับมาบ้าง แล้วได้ผลลัพธ์บนหน้าแอพพลิเคชั่นอย่างไร

รูป 4.3 request ที่ application ส่งให้ NodeRed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.4 ผลลัพท์การส่งคำร้องขอของหน้า Room

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.5 ผลลัพท์ที่ application แสดงในหน้า Room

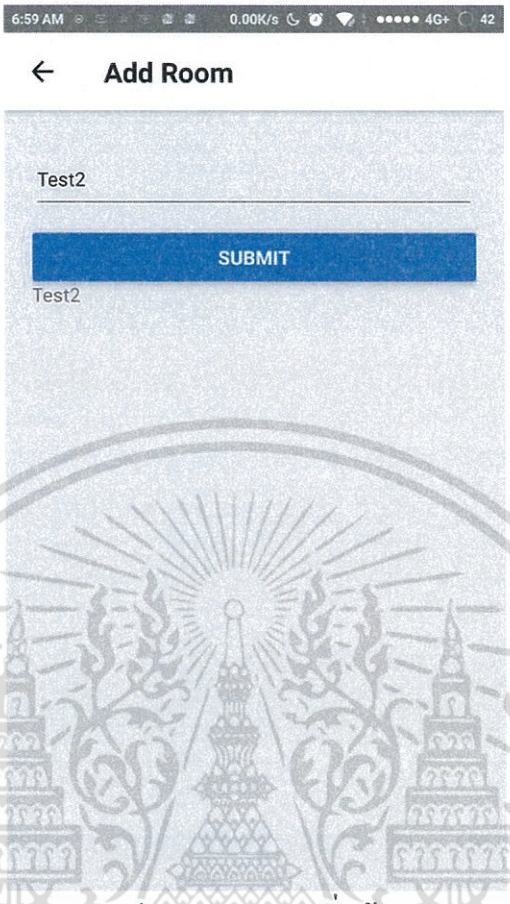
4.1.3 การเพิ่ม/ลบห้อง

การทดลองเรียกใช้ฟังก์ชั่น เพิ่มห้อง และลบห้อง เป็นการทดลองที่จะทำการ request ผ่าน POST method แล้วดูว่า ได้ส่งอะไรไปและ NodeRed จะส่งข้อมูลอะไรกลับมาบ้าง แล้วได้ผลลัพธ์บนหน้าแอพพลิเคชั่นอย่างไร

4.1.3.1 การเพิ่มห้อง

การเพิ่มห้อง จะต้องเข้าจากหน้า Home และกดปุ่ม Addroom

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



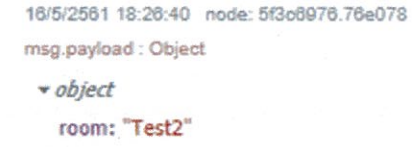
รูป 4.6 การทดลองเพิ่มห้อง

รูป 4.7 request ที่ application ส่งเมื่อทดลองเพิ่มห้อง

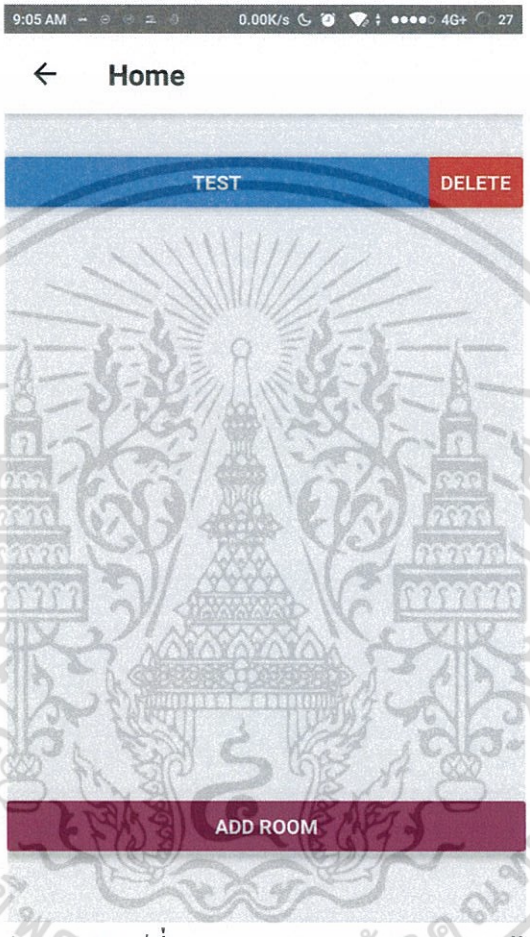
4.1.3.2 การลบห้อง

การลบห้อง จะสามารถเรียกใช้ฟังก์ชั่นนี้โดยจากหน้า Home โดยกดที่ปุ่ม Delete ด้านหลังของห้องที่ต้องการจะลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.8 request ที่ application ส่งไปเมื่องกดลบห้อง



รูป 4.9 ผลลัพท์ที่ application แสดงหลังจาก ลบห้อง

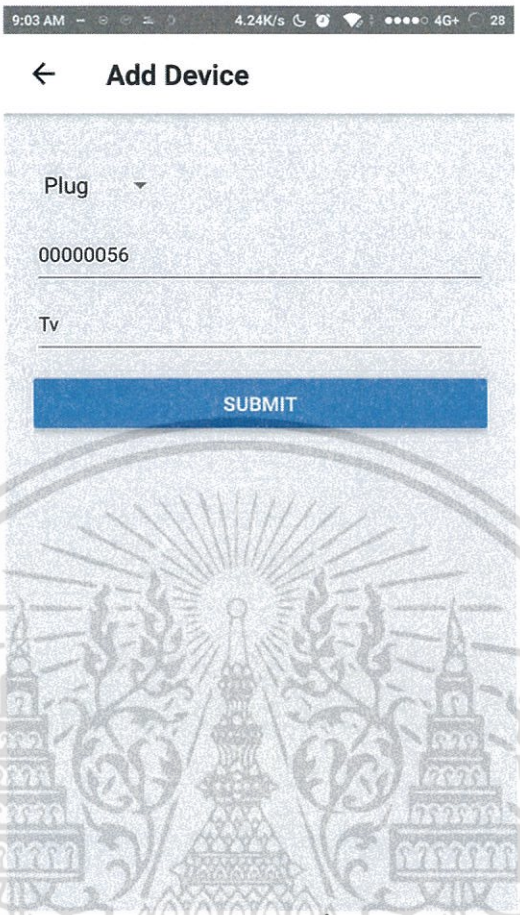
4.1.4 การเพิ่ม/ลบอุปกรณ์

การทดลองเรียกใช้ฟังก์ชั่น เพิ่มอุปกรณ์ และลบอุปกรณ์ เป็นการทดลองที่จะทำการ request ผ่าน POST method แล้วดูว่า ได้ส่งอะไรไปและ NodeRed จะส่งข้อมูลอะไรกลับมาบ้าง แล้วได้ผลลัพธ์บนหน้าแอพพลิเคชั่นอย่างไร

4.1.4.1 การเพิ่มอุปกรณ์

การเพิ่มอุปกรณ์ สามารถเรียกใช้ได้จากหน้า Room โดยกดปุ่ม Add Device ที่อยู่ด้านล่างของหน้า Room

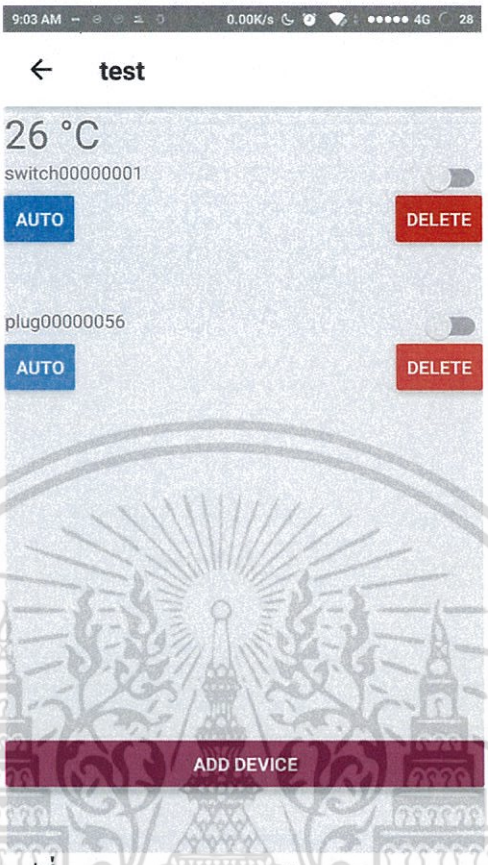
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.10 การทดลองเพิ่มอุปกรณ์

รูป 4.11 request ที่ application ส่งไปเมื่อทดลองเพิ่มอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



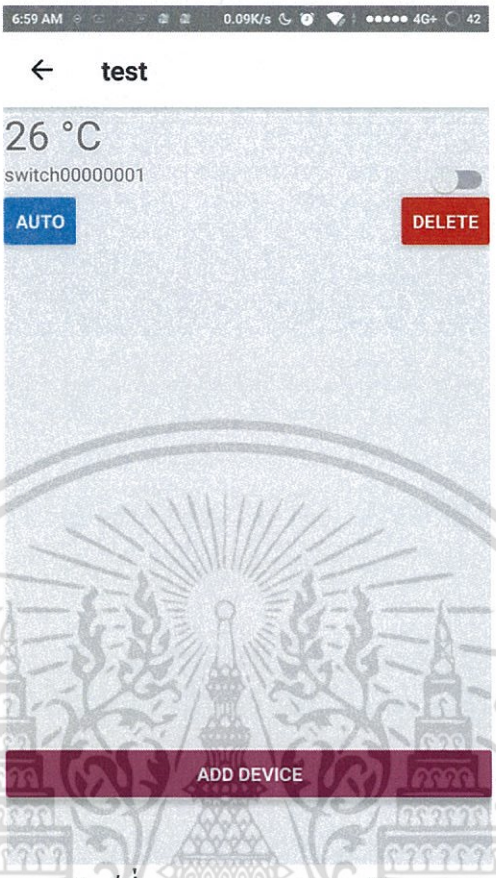
รูป 4.12 ผลลัพท์ที่ application แสดงหลังจากทำการทดลองเพิ่มอุปกรณ์

4.1.4.2 การลบอุปกรณ์

การลบอุปกรณ์

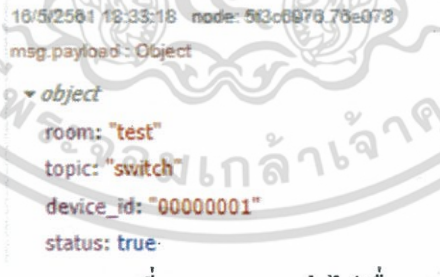
รูป 4.13 request ที่ application ส่งไปเมื่อทดลองลบอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



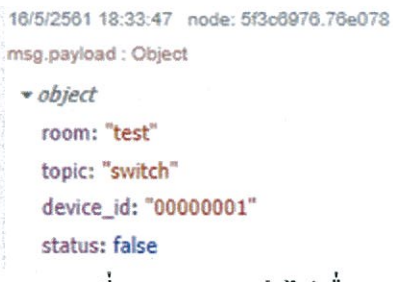
รูป 4.14 ผลลัพท์ที่ application แสดงหลังจาก ลบอุปกรณ์

4.1.5 การสั่งเปิดปิด

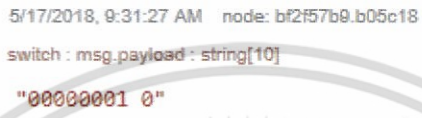


รูป 4.15 request ที่ application ส่งไปเมื่อทดลองสั่งเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

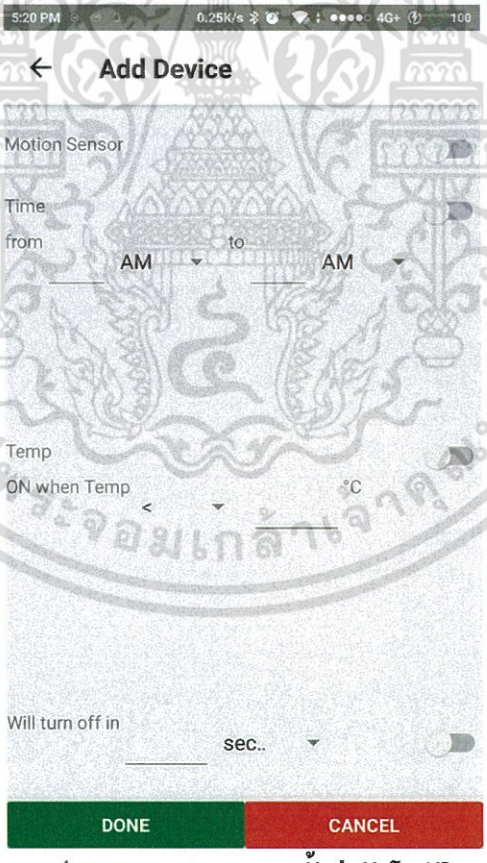


รูป 4.16 request ที่ application ส่งไปเมื่อทดลองสั่งปิดอุปกรณ์



รูป 4.17 ผลลัพท์ การส่ง mqtt ของ NodeRed เมื่อทำการสั่งปิดอุปกรณ์

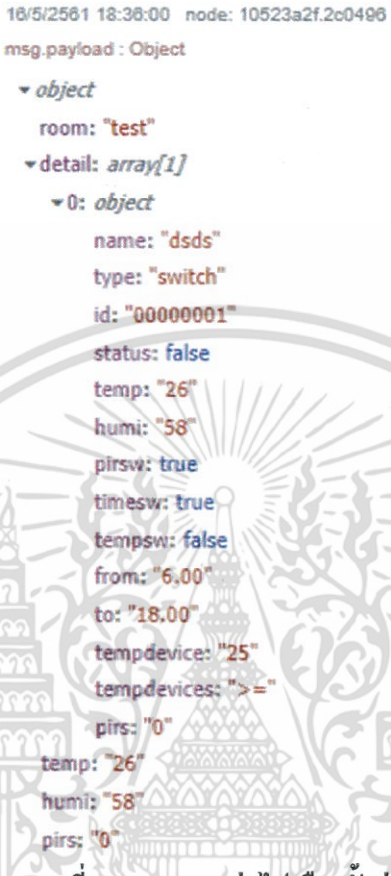
4.1.6 การตั้งการทำงานอัตโนมัติ



รูป 4.18 การทดลองการตั้งค่าอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองการตั้งค่าอัตโนมัติ

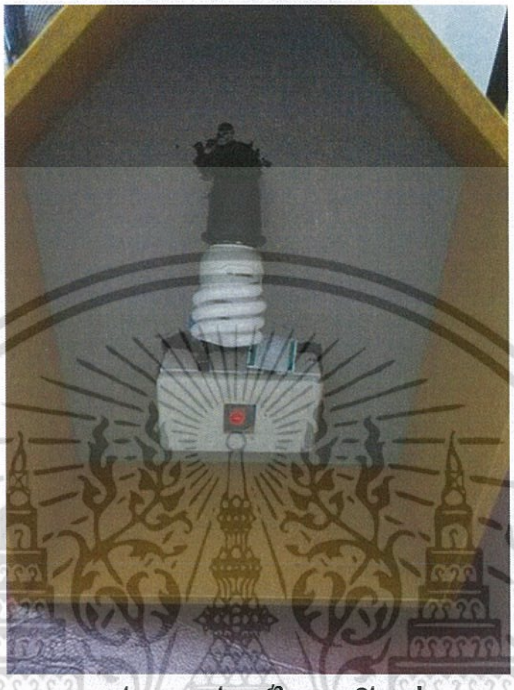


รูป 4.19 request ที่ application ส่งไปเมื่อตั้งค่าการทำงานอัตโนมัติ

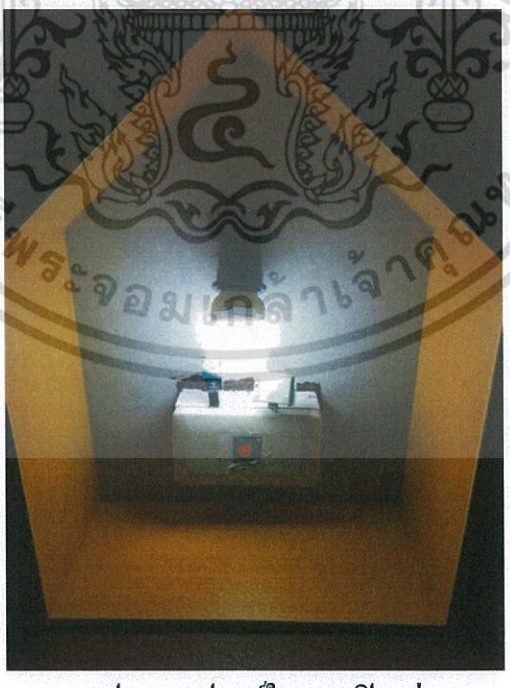
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองส่วนฮาร์ดแวร์

4.2.1 การเปิด/ปิดอุปกรณ์



รูป 4.20 อุปกรณ์ในขณะปิดอยู่



รูป 4.21 อุปกรณ์ในขณะเปิดอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การรับค่าจากเซ็นเซอร์

ทดลองโดยเปิดการใช้งานสวิทช์หนึ่งตัว แล้วอ่านค่าจาก Serial Monitor ได้ดังนี้

```
sec =3
temp : 30   humi : 61   Pir : 1

sec =4
temp : 30   humi : 61   Pir : 1

sec =5
temp : 30   humi : 62   Pir : 1

sec =6
temp : 30   humi : 62   Pir : 1

sec =7
temp : 30   humi : 61   Pir : 1

sec =8
temp : 30   humi : 61   Pir : 1

sec =9
temp : 30   humi : 61   Pir : 1

sec =10
temp : 30   humi : 61   Pir : 1

sec =11
temp : 30   humi : 61   Pir : 1

sec =13
temp : 30   humi : 61   Pir : 1
```

รูป 4.22 ค่าการใช้งานสวิทช์จาก Serial Monitor

4.2.3 การรับส่งข้อมูลกับเซิร์ฟเวอร์

4.2.3.1 การส่งข้อมูลไปยังเซิร์ฟเวอร์

ทดลองโดย อ่านค่าที่เซิร์ฟเวอร์ได้รับ ในช่วงเวลาหนึ่ง ๆ ที่เปิดการทำงานของอุปกรณ์สวิทช์หนึ่งตัว ได้ดังนี้

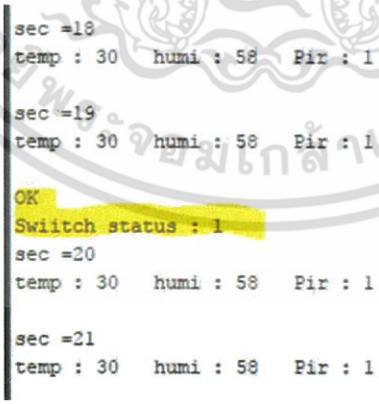
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 4.23 ค่าที่เซิร์ฟเวอร์ได้รับ ในช่วงเวลาหนึ่ง ๆ ที่เปิดการทำงานของอุปกรณ์สวิทช์

จะเห็นว่า มีการส่งข้อมูล ผ่าน protocol MQTT ใน topic “switch” โดยมี payload “00000001 “ ตามด้วย “pir “ ’1’ หรือ ‘0’ หรือ “tehu 30 61” ซึ่งคือ ค่าที่อุปกรณ์ต้องการบอกเซนเซอร์ ว่า Pir ไม่ตรวจพบ

4.2.3.2 การรับข้อมูลจากเซิร์ฟเวอร์

ทดลองโดยการให้ส่งข้อมูลผ่าน protocol MQTT ด้วย topic “switch” และ payload “00000001 1” คือ ต้องการสั่งสวิทช์ที่มีไอดี 00000001 ทำการหยุดจ่ายไฟไปที่หลอดไฟแล้วอ่านค่าจาก Serial Monitor



รูป 4.24 ค่าของสวิทช์จาก Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุปของโครงงาน

จากการได้ลงมือปฏิบัติจริงนั้น ได้ผลลัพธ์ที่ได้และความคืบหน้าในชิ้นงานนั้น อยู่เกณฑ์ที่คาดการไว้ สามารถเชื่อมต่ออุปกรณ์ รับคำสั่ง จัดเก็บ และนำข้อมูลมาประยุกต์ใช้ได้สำเร็จตามเป้าหมาย

5.2 ปัญหา อุปสรรค และแนวทางแก้ไข

จากการทำงานจะมีปัญหาในการทดลองใช้ โปรแกรมใหม่ๆที่ไม่เคยใช้และการได้ลงมือทำในรูปแบบที่ปกติไม่เคยทำ ทำให้ต้องเวลาในการศึกษาและหาความรู้ใหม่มาเพิ่มเพื่อใช้ในการทำงานให้สำเร็จ และมีปัญหาด้านอุปกรณ์ที่ไม่สามารถใช้งานในบางส่วนได้

5.3 แนวทางในการพัฒนาต่อ

โดยปัจจุบันสามารถนำไปต่อยอดในการทำหน้า UI ได้เพื่อเพิ่มทางเลือกในการใช้งาน และทางด้านแอปพิเคชั่นนั้นสามารถเพิ่มความยืดหยุ่นในการใช้งานได้มากกว่านี้ ในอนาคตข้างถ้านำมาทำต่อสามารถปรับปรุงให้ดีขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

อานนท์ ณ หนองคาย. **Embedded Systems.** [Online].

Available: https://embeddedsystem2558.wordpress.com

Arduino. **Arduino.** [Online].

Available: https://www.arduino.cc/en/Guide/Introduction

JS Foundation. **Node-RED.** [Online].

Available: https://nodered.org/

RASPBERRY PI FOUNDATION. **RASPBERRY PI.** [Online].

Available: https://www.raspberrypi.org/

Smart Home Asia. **ระบบบ้านอัจฉริยะ (Home automation) คืออะไร?.** [Online].

Available: https://www.smarthomeasia.in.th/

Wikipedia. **ESP8266.** [Online].

Available: https://en.wikipedia.org/wiki/ESP8266

Wikipedia. **Internet of things.** [Online].

Available: https://en.wikipedia.org/wiki/Internet_of_things

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้