

เว็บแอปพลิเคชันควบคุมอุปกรณ์ไอโอทีภายในบ้าน

นางสาวอมรรัตน์ วุฒิเจริญบุรี

นายศนันท์ ชินรักษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์ (คอมพิวเตอร์)

ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์

วิทยาลัยเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

พ.ศ. 2566

# Web Application Control Home IOT Devices

Miss Amornrat Woothicharoenpooree

Mr. Yodsanan Chinarak

Project Report Submitted in Partial Fulfillment of the Requirements

for the Bachelor's Degree of Industrial Technology in

Electronics Technology (Computer)

Department of Electronics Engineering Technology

College of Industrial Technology

King Mongkut's University of Technology North Bangkok

2023

หัวข้อปัญญานิพนธ์ : เว็บแอปพลิเคชันควบคุมอุปกรณ์ไอโอทีภายในบ้าน  
โดย : นางสาวอมรรัตน์ วุฒิเจริญฤทธิ์  
นายศนันท์ ชินารักษ์  
ที่ปรึกษาปัญญานิพนธ์ : อาจารย์ดำรงเกียรติ แซ่ลี้ม  
สาขาวิชา : เทคโนโลยีอิเล็กทรอนิกส์ (คอมพิวเตอร์)  
ภาควิชา : เทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์  
ปีการศึกษา : 2566

---

วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ อนุมัติให้นับ ปัญญา  
นิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

..... คณบดีวิทยาลัยเทคโนโลยีอุตสาหกรรม  
(รองศาสตราจารย์ ดร.สมิตร ส่งพิริยะกิจ)

คณะกรรมการสอบปัญญานิพนธ์

..... ประธานกรรมการ  
(อาจารย์ ดร. เลอสรณ์ กิรสมุทธานนท์)

..... กรรมการ  
(อาจารย์ดำรงเกียรติ แซ่ลี้ม)

..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. พิสิทธิ วิสุทธิเมธีกร)

Project Title : Web Application Control Home IOT Devices

By : Miss Amornrat Woothicharoenpooree  
Mr. Yodsanan Chinarak

Project Advisor : Mr. Damrongkiat Lim

Major Field : Electronics Technology (Computer)

Department : Electronics Engineering Technology

Academic Year : 2023

---

Accepted by the College of Industrial Technology, King Mongkut's University of Technology North  
Bangkok in Partial Fulfillment of the Requirements for the Bachelor's Degree of Industrial Technology

..... Dean of College of Industrial Technology  
(Assoc. Prof. Dr. Smith Songpiriyakij)

Project Committee

..... Chairperson  
(Dr. Lerson Kirasamutiranon)

..... Member  
(Mr. Damrongkiat Lim)

..... Member  
(Dr. Pisit Wisutmetheekorn )

## กิตติกรรมประกาศ

ปริญญานิพนธ์เว็บแอปพลิเคชันควบคุมอุปกรณ์ไอโอทีภายในบ้าน สำเร็จลุล่วงไปได้ด้วยดีเนื่องด้วยได้รับความช่วยเหลือเป็นอย่างดีจากอาจารย์ดำรงเกียรติ แซ่ลี้ม อาจารย์ที่ปรึกษาที่ให้คำแนะนำและข้อคิดต่าง ๆ ของการจัดทำปริญญานิพนธ์ และการแก้ไขข้อบกพร่องต่าง ๆ มาโดยตลอด

ขอกราบขอบพระคุณบุพการีเป็นอย่างสูง ซึ่งให้การสนับสนุนในทุก ๆ ด้านเป็นแรงผลักดัน และให้กำลังใจคอยสนับสนุนแก่ผู้จัดทำเสมอมาจนสำเร็จการศึกษา ขอขอบพระคุณคณะอาจารย์สาขาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ แขนงวิชาคอมพิวเตอร์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และให้ความช่วยเหลือด้านเทคนิคหลาย ๆ อย่างเป็นอย่างดี และขอขอบพระคุณทุกท่านและผู้ที่มีส่วนเกี่ยวข้องกับความสำเร็จแต่มิได้เอ่ยนามทุกท่านมา ณ ที่นี้ด้วย

สุดท้ายนี้ทางคณะผู้จัดทำ ต้องขอขอบพระคุณท่านกรรมการสอบปริญญานิพนธ์ทุกท่านเป็นอย่างสูง ที่ได้ช่วยพิจารณาและให้คำแนะนำในการตรวจทานแก้ไข อนุโมทนา ปริญญานิพนธ์นี้สำเร็จเป็นไปตามวัตถุประสงค์และขอบเขตที่ตั้งไว้ทุกประการ ซึ่งคณะผู้จัดทำหวังว่าปริญญานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อผู้ที่ใช้งานเว็บแอปพลิเคชัน

คณะผู้จัดทำ

## เว็บแอปพลิเคชันควบคุมอุปกรณ์ไอโอทีภายในบ้าน

อมรรัตน์ วุฒิเจริญฤทธิ์<sup>1</sup> ยศนันท์ ชินารักษ์<sup>2</sup> และ คำรงเกียรติ แซ่ลิ้ม<sup>3</sup>

### บทคัดย่อ

ในปัจจุบันมีการใช้เทคโนโลยีไอโอทีจำนวนมากในชีวิตประจำวัน และมีหลากหลายชนิดในแต่ละอุปกรณ์ จึงมีการพัฒนาอุปกรณ์เครื่องมืออำนวยความสะดวกมากมาย ทำให้มีความสะดวกสบายมากขึ้น โดยเฉพาะในยุคที่มีอินเทอร์เน็ตที่มีการใช้งานอย่างกว้างขวางและได้มีอุปกรณ์ที่เชื่อมต่อกับระบบอินเทอร์เน็ต เพื่อการใช้งานตลอดเวลามากขึ้น

โครงการปริญญานิพนธ์นี้จะพัฒนาระบบเว็บแอปพลิเคชันอุปกรณ์ไอโอทีภายในบ้านที่มีการสั่งงานผ่านอินเทอร์เน็ตโดยใช้บอร์ด Raspberry Pi เป็นตัวกลางในการเชื่อมต่ออุปกรณ์ไอโอทีในแต่ละชนิด เข้าด้วยกันโดยผ่านโพรโทคอล MQTT โดยจะส่งค่าจาก Raspberry Pi โดยเชื่อมต่อไปยังตัวลูกที่รองรับคือบอร์ดไมโครคอนโทรลเลอร์ ESP32 โดยจะรองรับคำสั่งการจากตัว แดชบอร์ดที่ใช้สร้างขึ้นมาจาก Node-Red และจะส่งการทำงานต่อไปยังอุปกรณ์ที่เราต้องการใช้

คำสำคัญ : Raspberry Pi, เว็บแอปพลิเคชัน, IOT, Node Red

---

<sup>1,2</sup>นักศึกษา, <sup>3</sup>อาจารย์ที่ปรึกษาภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์, วิทยาลัยเทคโนโลยีอุตสาหกรรม, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

## Web Application Control Home IOT Devices

Amornrat Woothicharoenpooree<sup>1</sup> Yodsanan Chinarak<sup>2</sup> and Damrongkiat Lim<sup>3</sup>

### Abstract

Nowadays, devices that are internet or IoT devices are used in many ways. Both measurements from the environment or electronic device control But because each type of IoT device has many types cause different communication agreements And each type of IoT device cannot exchange data directly. Therefore, we want to develop an intermediary device that acts as a gateway for building a system of IoT devices that supports a variety of protocols or standards that can be easily deployed and used. This includes adjusting settings and monitoring the status of gateways and local IoT devices through a browser. In addition, IoT devices can communicate with each other in the event of a lack of internet connection.

This project develops an application system that can be commanded via the Internet using an ESP32 microcontroller board that can connect to a Wi-Fi network. ESP32 is used for controlling various devices in the farm. for optimum temperature and humidity. In addition, the developed system consists of server computer A web application developed by Node Red. Each farm uses a microcontroller to control the devices. And read the temperature and humidity values from the DHT22 sensor to send data. At the same time, the temperature and humidity values will be displayed on the web application. Commands from the device can be set via a web application and command values are simultaneously saved in the database.

**Keywords :** Raspberry Pi, Web Application, IOT, Node Red

---

<sup>1,2</sup>students, <sup>3</sup>advisors from Department of Electronics Engineering Technology, College of Industrial Technology, King Mongkut's University of Technology North Bangkok

## 1. บทนำ

ในปัจจุบันมีการใช้เทคโนโลยีไอโอทีจำนวนมากในชีวิตประจำวัน และมีหลากหลายชนิดในแต่ละอุปกรณ์ จึงต้องทำการโหลดแอปพลิเคชันก่อนข้างหลากหลายเพื่อทำการสื่อสารกับอุปกรณ์นั้นๆ ซึ่งทำให้เกิดความลำบากในการใช้งานอุปกรณ์เนื่องจากต้องสลับแอปพลิเคชันเพื่อเปิดอุปกรณ์แต่ละแอปพลิเคชัน

จากปัญหาที่เกิดขึ้น จึงเกิดแนวคิดที่จะแก้ไขปัญหาที่เกิดขึ้น โดยมีแนวคิดที่ว่าควรจับอุปกรณ์ที่หลากหลายมาสั่งการภายในแอปพลิเคชันเดียว โดยทำการสร้าง Raspberry Pi ขึ้นมาหนึ่งตัวโดยอุปกรณ์ทั้งหมดจะต้องผ่าน Raspberry Pi เดียวกันและสั่งการโดยแอปพลิเคชันตัวเดียวกันและต้องใช้งานได้ทั้งคอมพิวเตอร์และมือถือจึงเป็นที่มาของการทำเป็นเว็บแอปพลิเคชันควบคุมอุปกรณ์ไอโอทีภายในบ้าน

โดยโครงการปริญญานิพนธ์นี้คาดหวังว่าจะลดภาระของผู้ใช้งานและเพิ่มความสะดวกสบายของผู้ใช้งานได้โดยที่สั่งการผ่านเว็บแอปพลิเคชันตัวเดียว และสามารถเช็คข้อมูลต่างๆผ่านทางเว็บแอปพลิเคชันได้

## 2. เครื่องมือพัฒนาระบบ

### 2.1 Raspberry Pi

Raspberry Pi [2] คือชุดคอมพิวเตอร์บอร์ดเดี่ยวขนาดเล็กซึ่งประกอบไปด้วย หน่วยประมวลผลกลาง, หน่วยความจำ, ตัวรับสัญญาณไวไฟและตัวรับ

สัญญาณบลูทูธ ด้วยเหตุนี้จึงนำมาใช้งานแทนคอมพิวเตอร์เครื่องหนึ่งได้ จึงเหมาะกับการทำเป็นตัว Server สำหรับการใช้งานกับโครงการปริญญานิพนธ์นี้

### 2.2 ESP32

ESP32 [3] คือ ไมโครคอนโทรลเลอร์ ที่ออกแบบมาให้มีความสามารถในการเชื่อมต่อเครือข่าย WiFi และรับส่ง ข้อมูลผ่าน Bluetooth มีความปลอดภัยในการรับ-ส่ง ข้อมูลรวดเร็ว และประหยัดทรัพยากร

### 2.3 PostgreSQL

PostgreSQL [4] คือ เป็นระบบการจัดการฐานข้อมูลเชิงวัตถุ-สัมพันธ์ (object-relational) แบบ ORDBMS โดยสามารถใช้รูปแบบคำสั่งของภาษา SQL ได้เกือบทั้งหมด นอกจากนี้ยังเป็นระบบฐานข้อมูลที่ทันสมัยที่สุดของ OpenSource ที่สามารถนำไปใช้งานได้โดยไม่มีค่าใช้จ่ายใดๆ สามารถทำงานบนระบบปฏิบัติการได้ทั้ง Linux, UNIX (AIX, BSD, HP-UX, SGI Irix, Mac OS X, Solaris, Tru64) และ Windows

### 2.4 MQTT

MQTT [5] คือ โพรโตคอลการส่งข้อความที่อิงตามมาตรฐาน หรือชุดของกฎที่ใช้สำหรับการสื่อสารระหว่างเครื่องต่อเครื่อง ซึ่งในด้าน IoT มักจะต้องส่งและรับข้อมูลผ่านเครือข่ายที่มีข้อจำกัดด้านทรัพยากร



จึงใช้งาน MQTT เนื่องจากมันใช้งานง่ายและสามารถ  
สื่อสารข้อมูล IOT ได้อย่างมีประสิทธิภาพ

## 2.5 Node Red

Node Red [6] คือ เครื่องมือสำหรับนักพัฒนา  
โปรแกรมในการเชื่อมต่ออุปกรณ์ฮาร์ดแวร์เข้ากับ APIs  
(Application Programming Interface) ซึ่งเป็น การ  
พัฒนาโปรแกรมแบบ Flow-Based Programming ที่มี  
หน้า UI สำหรับนักพัฒนา ใช้งานผ่าน Web Browser

## 2.6 Javascript

Javascript [7] คือ เป็นภาษาสคริปต์ ที่มีลักษณะ  
การเขียนแบบโพรโทไทป์ ส่วนมากใช้ในหน้าเว็บเพื่อ  
ประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน แต่ก็ยังมีใช้เพื่อ  
เพิ่มเติมความสามารถในการเขียนสคริปต์โดยฝังอยู่ใน  
โปรแกรมอื่นๆ

## 2.7 Python

Python [8] คือ ภาษาโปรแกรมระดับสูงที่ใช้กัน  
อย่างแพร่หลาย ถูกออกแบบเพื่อให้มีโครงสร้างและ  
ไวยากรณ์ของภาษาที่ไม่ซับซ้อน เข้าใจง่าย มีการใช้  
พัฒนาแอปพลิเคชัน เว็บไซต์ รวมถึงแอปบนมือถือ  
หรือ อุปกรณ์เคลื่อนที่ด้วย หน้าที่ของ Python ก็คือการ  
ทำงานแปลชุดคำสั่งทีละบรรทัดเพื่อป้อนเข้าสู่หน่วย  
ประมวลผล

## 2.8 C++

C++ [9] คือ เป็นภาษาคอมพิวเตอร์เพื่อ  
วัตถุประสงค์ทั่วไป ซึ่งสามารถเขียนโปรแกรมได้ทั้ง

แบบออบเจ็กต์ และการเขียนแบบปกติทั่วไป และยังมี  
เครื่องมืออำนวยความสะดวกในการจัดการและเข้าถึง  
เขียนโปรแกรมแบบต่างๆ มากมาย

## 2.9 Arduino

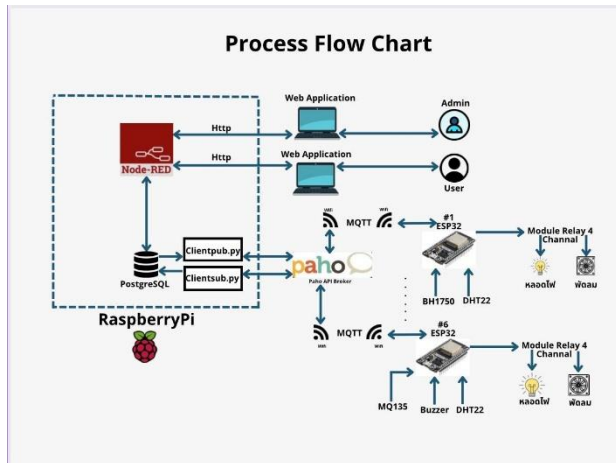
Arduino [10] คือ โปรแกรมที่ใช้สำหรับการ  
เขียนโปรแกรมควบคุมและสามารถทำการแปลงไฟล์  
ดังกล่าวเพื่อนำไปอัปโหลดลงยังบอร์ดไมโครคอน  
โทรลเลอร์ โดยโปรแกรมนี้นี้เป็นโปรแกรมที่ใช้งานใน  
ลักษณะ Open Source

## 3. วิธีการดำเนินงาน

เว็บแอปพลิเคชันควบคุมอุปกรณ์ไอโอที  
ภายในบ้านในปริญญานิพนธ์นี้มีวิธีการดำเนินงาน  
ผ่าน เว็บแอปพลิเคชัน เพื่อเข้าไปเซตค่าในตัว  
อุปกรณ์ ในการควบคุมและป้อนคำสั่ง โดยวิธีการ  
ป้อนคำสั่งจะป้อนผ่านเว็บแอปพลิเคชันโดยใช้ UI  
จาก Node Red ในหน้าเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันจะมี Raspberry Pi เป็น  
ตัวกลางในการเชื่อมต่ออุปกรณ์ต่างๆ โดยจะ  
เชื่อมต่อจาก WiFi และส่งข้อมูลให้อุปกรณ์อื่นๆ  
ผ่าน Wireless ซึ่งจะใช้การเชื่อมต่ออุปกรณ์แบบ

MQTT ในการรับส่งข้อมูลที่ได้มาจาก  
หน้าเว็บแอปพลิเคชัน ดังรูปที่ 1



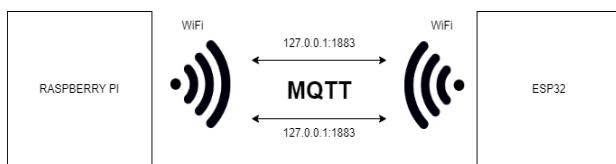
รูปที่ 1 ภาพไดอะแกรมการทำงานโดยรวม

### 3.1 หลักการทำงานของระบบ

การทำงานของระบบจะแบ่งออกเป็นส่วนหลักๆ ดังนี้

#### 3.1.1 การออกแบบอุปกรณ์ที่ใช้เป็นตัวกลาง

อุปกรณ์ที่ใช้เป็นตัวกลางในที่นี้เราเลือกบอร์ด Raspberry Pi เป็นอุปกรณ์ในการส่งค่าและรับค่าไปยังอุปกรณ์ตัวลูก โดยในที่นี้เราจะใช้บอร์ด ESP32 เป็นตัวรับค่าจาก Raspberry Pi โดยจะรับส่งค่าผ่าน Wireless เดียวกัน โดยจะส่งข้อมูลผ่านโปรโตคอล MQTT โดยจะใช้พอร์ต 1883 ซึ่งเป็นพอร์ตพื้นฐานของโปรโตคอล MQTT ดังรูปที่ 2

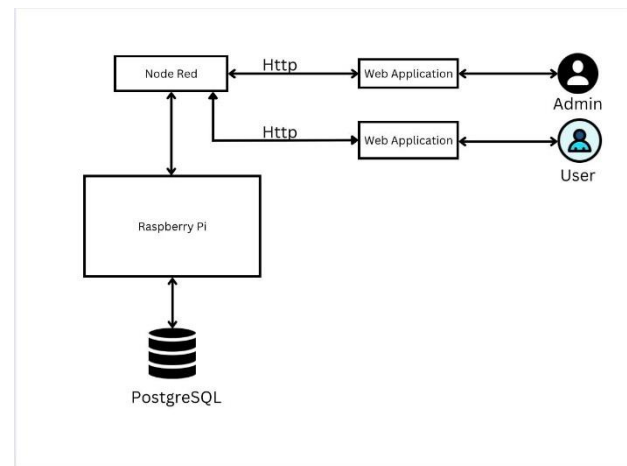


รูปที่ 2 ภาพแสดงตัวอย่างการรับส่งข้อมูล

#### 3.1.2 การออกแบบในส่วนของการติดต่อผู้ใช้งาน

นอกจาก Raspberry Pi จะเป็นเกตเวย์ในการติดต่อกันระหว่างอุปกรณ์แล้ว ยังมีหน้าที่ในการเป็น Web Server ในการทำแคชบอร์ดติดต่อกับผู้ใช้ ซึ่งจะใช้ Node-Red ในการทำงานในส่วนของ UI บนเว็บแอปพลิเคชัน ดังรูปที่ 3

โดย Node-Red จะใช้ไลบรารีของ PostgreSQL ในการเก็บข้อมูล



รูปที่ 3 ภาพแสดงตัวอย่างการเชื่อมต่อและเก็บข้อมูล

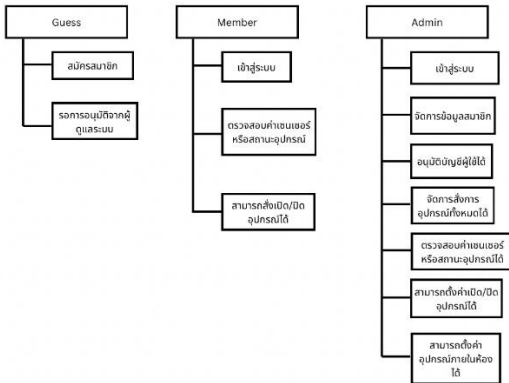
#### 3.1.3 การออกแบบในส่วนของการซอฟต์แวร์

ในส่วนของการซอฟต์แวร์โดยในส่วนหลักจะใช้ภาษา Python ในการเขียนโปรแกรมบน Raspberry Pi และภาษา Javascript ในการเขียนเว็บแอปพลิเคชันและภาษา C++ ในการเขียนโปรแกรมบน Arduino

บอร์ด Raspberry Pi จะมีการติดตั้ง Node-Red, PostgreSQL โดยจะเขียนสคริปต์ขึ้นมาสองสคริปต์โดยการทำงานแต่ละสคริปต์จะเป็นดังนี้



อื่นได้ถ้าผู้ดูแลระบบไม่ให้สิทธิ์การใช้งานโดยสิทธิ์  
ผู้ใช้งานจะเป็นดังรูปที่ 5

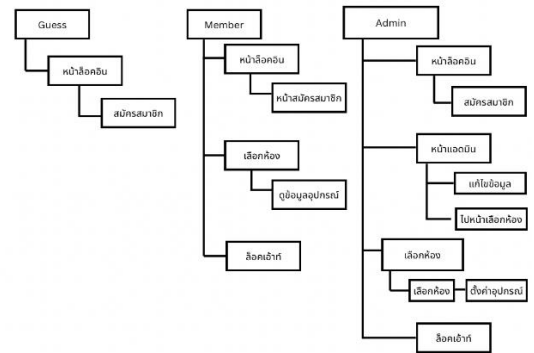


รูปที่ 5 สิทธิ์ผู้ใช้งาน

### 3.2.3 การใช้งานหน้าเว็บแอปพลิเคชัน

เนื่องจากเว็บแอปพลิเคชันได้กำหนดสถานะ  
ผู้ใช้งานแตกต่างกัน จึงทำให้การเข้าใช้งานหน้าเว็บ  
แอปพลิเคชันต่างกัน

โดยที่ผู้ใช้งานทั่วไปเมื่อทำการเข้าเว็บแอป  
พลิเคชันจะต้องทำการสมัครสมาชิกและรอผู้ดูแล  
ระบบอนุมัติ เมื่ออนุมัติและทำการเข้าสู่ระบบจะพาไป  
ยังหน้าเมนูหลัก ซึ่งแตกต่างจากผู้ดูแลระบบที่จะเข้าไป  
ยังหน้าจัดการสมาชิกก่อนเป็นอันดับแรกและสามารถ  
แก้ไขข้อมูลสมาชิกได้ในหน้านั้น ก่อนที่จะเข้าเมนูหลัก  
เพื่อเลือกห้องที่จะต้องการใช้งานดังรูปที่ 6



รูปที่ 6 การเข้าใช้งานในแต่ละสถานะ

### 3.3 การตั้งค่าห้อง

ในงานวิจัยนี้อุปกรณ์ที่เราใช้ในการเชื่อมต่อ  
เกตเวย์ที่เป็น Raspberry Pi ก็คือ ESP32 โดย ESP32 จะ  
เชื่อมกับ Module Relay 4 Channel และเซนเซอร์  
DHT22, MQ135, BH1750 ในการรับ-ส่งค่าของอุปกรณ์  
ส่งไปยัง Raspberry Pi

โดยตัวห้อง จะทำการรับไฟที่ได้จากตัว  
Switching Power Supply 12V โดยนำมาผ่าน Module  
DC Stepdown เพื่อมาแปลงไฟเป็น 5V เท่ากับการรับ  
ไฟของบอร์ด ESP32 และนำไปต่อเข้ากับพอร์ต Vin  
ของ ESP32 เพื่อนำมาไฟมาเลี้ยงที่บอร์ด

โดยในส่วนของ Module Relay 4 Channel จะ  
ทำการเชื่อมต่อกับ Module DC Stepdown เพื่อนำไฟ  
5V มาเลี้ยงตัวบอร์ด จากนั้นจะทำการติดตั้งเอาต์พุตซึ่ง  
จะเป็นหลอดไฟ กับ พัดลมและ โซลินอยด์วาล์ว

เมื่อติดตั้งอุปกรณ์เรียบร้อยแล้วในห้องตัวบอร์ด  
ESP32 จะทำการรอรับค่าคำสั่งจากตัว Raspberry Pi  
เพื่อนำมาสั่งการตัวรีเลย์ตามคำสั่งที่ส่งมา และจะนำ  
ค่าที่ได้จากเซนเซอร์ส่งไปยังตัวบอร์ด Raspberry Pi

เพื่อเก็บในฐานข้อมูลและนำไปแสดงผลบนแดชบอร์ด  
เพื่อโชว์ให้ผู้ใช้งาน

### 3.4 ระบบฐานข้อมูล

ในงานวิจัยนี้จะทำการเก็บข้อมูลใน  
PostgreSQL โดยจะแบ่งการเก็บข้อมูลเป็นดังนี้

ตารางที่ 1 โครงสร้างตาราง Memberdb ซึ่งเป็นตารางที่  
เก็บข้อมูลสมาชิก

ชื่อฟิลด์	ประเภทข้อมูล	ข้อมูล
username	String	ชื่อสมาชิก
password	String	รหัสผ่านสมาชิก
status	String	สถานะสมาชิก

ตารางที่ 2 โครงสร้างตาราง rooms ซึ่งเป็นตารางที่เก็บ  
ชื่อของแต่ละห้อง

ชื่อฟิลด์	ประเภทข้อมูล	ข้อมูล
id	Integer	เลขลำดับห้อง
name	String	ชื่อห้อง
Relay1	String	ชื่ออุปกรณ์ชุด 1
Relay2	String	ชื่ออุปกรณ์ชุด 2
Relay3	String	ชื่ออุปกรณ์ชุด 3
Relay4	String	ชื่ออุปกรณ์ชุด 4
Sensor1	String	ชื่อเซนเซอร์ที่ 1
Sensor2	String	ชื่อเซนเซอร์ที่ 2
Sensor3	String	ชื่อเซนเซอร์ที่ 3
Sensor4	String	ชื่อเซนเซอร์ที่ 4

ตารางที่ 3 โครงสร้างตาราง Modules ซึ่งเป็นตารางที่  
เก็บสถานะของอุปกรณ์แต่ละตัวไว้

ชื่อฟิลด์	ประเภทข้อมูล	ข้อมูล
id	Integer	เลขลำดับห้อง
Relay1	False	ชื่ออุปกรณ์ชุด 1
Relay2	False	ชื่ออุปกรณ์ชุด 2
Relay3	False	ชื่ออุปกรณ์ชุด 3
Relay4	False	ชื่ออุปกรณ์ชุด 4

ตารางที่ 4 โครงสร้างตาราง sensors ซึ่งเป็นตารางที่เก็บ  
ข้อมูลเซนเซอร์ของแต่ละแดชบอร์ด

ชื่อฟิลด์	ประเภทข้อมูล	ข้อมูล
Dashboard_id	Integer	ลำดับ dashboard
Sensor1	String	ชื่อเซนเซอร์ที่ 1
Sensor2	String	ชื่อเซนเซอร์ที่ 2
Sensor3	String	ชื่อเซนเซอร์ที่ 3
Sensor4	String	ชื่อเซนเซอร์ที่ 4

### 3.5 การเขียนรับส่งค่าระหว่างบอร์ด

การรับส่งค่าเราจะเขียนไฟล์ส่งไว้ที่ตัวบอร์ด  
ESP32 และ Raspberry Pi โดยแบ่งเป็นทั้งหมด 2 ฟังก์ชัน  
แบ่งดังนี้

#### 3.5.1 Raspberry Pi

ตัว Raspberry Pi จะเขียนด้วยภาษา Python  
โดยแบ่งเป็น 2 ไฟล์ดังนี้

1. Clientsub.py คือ Subscribe รับค่าเซนเซอร์  
ที่ส่งมาจาก Raspberry Pi และนำมาเก็บไว้ใน  
ฐานข้อมูล PostgreSQL ใน sensors โดยใช้ Paho MQTT  
Broker ในการสื่อสารกับบอร์ด ESP32 โดยเขียน  
ฟังก์ชัน Callback ที่เก็บค่าเซนเซอร์ที่รับมาจาก ESP32  
มาไว้ในตัวแปร sensor1 ดังรูปที่ 7 จากนั้นนำค่าที่ได้ไป  
เก็บไว้ในฐานข้อมูล โดยสร้างฟังก์ชัน update\_sensor

ผังรูปที่ 8 ซึ่งเราจะอัปเดตค่าเข้าฟังก์ชันโดยการเขียน  
update\_sensor(เลข แดชบอร์ด,ค่าเซนเซอร์ 1,ค่า  
เซนเซอร์2,ค่าเซนเซอร์3,ค่าเซนเซอร์4) ซึ่งจะอยู่ใน  
While True เพื่ออัปเดตค่าตลอด โดยการดำเนินงานจะ  
เป็นไปตามผังรูปที่ 9

```
def callback_esp32_sensor1(client, userdata, msg):
    global sensor1
    sensor1 = msg.payload.decode('utf-8')
    #print('TEMP: ', msg.payload.decode('utf-8'))

def callback_esp32_sensor2(client, userdata, msg):
    global sensor2
    sensor2 = msg.payload.decode('utf-8')
    #print('HUMIDITY: ', str(msg.payload.decode('utf-8'))))

def callback_esp32_sensor3(client, userdata, msg):
    global sensor3
    sensor3 = msg.payload.decode('utf-8')
    #print('SOIL: ', str(msg.payload.decode('utf-8'))))

def callback_esp32_sensor4(client, userdata, msg):
    global sensor4
    sensor4 = msg.payload.decode('utf-8')
    #print('SOIL: ', str(msg.payload.decode('utf-8'))))

def callback_esp32_sensor5(client, userdata, msg):
    global sensor5
    sensor5 = msg.payload.decode('utf-8')

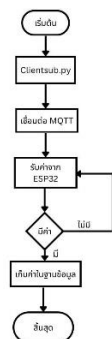
def callback_esp32_sensor6(client, userdata, msg):
    global sensor6
    sensor6 = msg.payload.decode('utf-8')
```

รูปที่ 7 ฟังก์ชันการรับค่าจาก ESP32

```
def update_sensor(board_id, sensor1, sensor2, sensor3, sensor4):
    """ update vendor name based on the vendor id """
    sql = """ UPDATE sensors
    SET sensor1 = %s,sensor2 = %s,sensor3 = %s,sensor4 = %s
    WHERE dashboard_id = %s"""

    conn = None
    updated_rows = 0
    try:
        # read database configuration
        #params = config()
        # connect to the PostgreSQL database
        conn=psycopg2.connect(host='localhost',database='smartroom',user='pi',password='1234')
        # create a new cursor
        cur = conn.cursor()
        # execute the UPDATE statement
        cur.execute(sql, (sensor1,sensor2,sensor3,sensor4,board_id))
        # get the number of updated rows
        updated_rows = cur.rowcount
        # commit the changes to the database
        conn.commit()
        # Close communication with the PostgreSQL database
        cur.close()
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
    finally:
        if conn is not None:
            conn.close()
    return updated_rows
```

รูปที่ 8 ฟังก์ชันการอัปเดตค่า PostgreSQL



รูปที่ 9 Flow Chart การทำงานของไฟล์ Clientsub.py

2. Clientpub.py คือตัว Publisher ส่งค่าที่รับมาจากการ  
สั่งการบน Node-Red ส่งไปยัง PostgreSQL โดยตัว  
ไฟล์จะอ่านค่าสถานะ รีเลย์ ที่เก็บไว้ใน PostgreSQL มา  
สั่งการตัว ESP32 โดยจะสร้างตัวแปรเพื่อรับค่าที่อ่าน  
จากฐานข้อมูล modules ผังรูปที่ 10 จากนั้นจะนำไป  
เทียบค่าที่ได้ ผังรูปที่ 11 เพื่อส่งไปยัง ESP32 ตาม  
Topic ที่ตั้งไว้ ผังรูปที่ 12 โดยการทำงานจะเป็นไปตาม  
ผังรูปที่ 13

```
conn=psycopg2.connect(host='localhost',database='smartroom',user='pi',password='1234')
sql1 = """ SELECT relay1 FROM modules WHERE id=0 """
sql11 = """ SELECT relay1 FROM modules WHERE id=0 """
sql12 = """ SELECT relay3 FROM modules WHERE id=0 """
sql13 = """ SELECT relay4 FROM modules WHERE id=0 """
#Board2
sql14 = """ SELECT relay1 FROM modules WHERE id=1 """
sql15 = """ SELECT relay2 FROM modules WHERE id=1 """
sql16 = """ SELECT relay3 FROM modules WHERE id=1 """
sql17 = """ SELECT relay4 FROM modules WHERE id=1 """
#Board3
sql18 = """ SELECT relay1 FROM modules WHERE id=2 """
sql19 = """ SELECT relay2 FROM modules WHERE id=2 """
sql110 = """ SELECT relay3 FROM modules WHERE id=2 """
sql111 = """ SELECT relay4 FROM modules WHERE id=2 """
```

รูปที่ 10 การอ่านค่ามาเก็บในตัวแปร

```
try:
    if relay1_1 == False:
        msg = 'OFF1'
    else:
        msg = 'ON1'

    if relay2_1 == False:
        msg1 = 'OFF2'
    else:
        msg1 = 'ON2'

    if relay3_1 == False:
        msg2 = 'OFF3'
    else:
        msg2 = 'ON3'
    if relay4_1 == False:
        msg3 = 'OFF4'
    else:
        msg3 = 'ON4'
    if relay1_2 == False:
        msg4 = 'OFF5'
    else:
        msg4 = 'ON5'
    if relay2_2 == False:
        msg5 = 'OFF6'
    else:
        msg5 = 'ON6'
```

รูปที่ 11 การอ่านค่าที่รับมาและเก็บไว้ในตัวแปรที่จะ  
ส่งให้ ESP32



```
pubMsg = client.publish(
    topic='rpi/relay1_1',
    payload=msg.encode('utf-8'),
    qos=0,
)
pubMsg.wait_for_publish()
print(pubMsg.is_published())
pubMsg1 = client.publish(
    topic='rpi/relay1_2',
    payload=msg1.encode('utf-8'),
    qos=0,
)
pubMsg1.wait_for_publish()
print(pubMsg1.is_published())
pubMsg2 = client.publish(
    topic='rpi/relay1_3',
    payload=msg2.encode('utf-8'),
    qos=0,
)
pubMsg2.wait_for_publish()
print(pubMsg2.is_published())
pubMsg3 = client.publish(
    topic='rpi/relay1_4',
    payload=msg3.encode('utf-8'),
    qos=0,
)
pubMsg3.wait_for_publish()
```

รูปที่ 12 การส่งค่าไปยัง ESP32 ผ่าน Topic



รูปที่ 13 Flow Chart การทำงานของไฟล์ Clientpub.py

### 3.5.2 ESP32

ตัว ESP32 เราจะเขียนโค้ดโดยใช้ภาษา C++ ผ่านโปรแกรม Arduino โดย ESP32 ทำการเชื่อมต่อ IP Address ของตัว MQTT BROKER และทำการเรียกใช้ไลบรารี PubsubClient.h จากนั้นทำการเขียนฟังก์ชัน connect\_mqttServer ในการเชื่อมต่อและกำหนด Topic ที่จะรอรับจากตัวบอร์ด Raspberry Pi โดยใช้ client.subscribe("rpi/relay1\_1") ... ("rpi/relay1\_2") เป็น

ต้น โดยแต่ละบอร์ด Topic ที่รอรับจะไม่เหมือนกัน ดังรูปที่ 14 จากนั้นจะนำค่าที่ได้มาจาก Topic มาเข้าฟังก์ชัน callback และนำมาเทียบในการสั่งรีเลย์ดังรูปที่ 15 ส่วนค่าที่ได้จากเซนเซอร์จะส่งผ่าน client.publish ในการส่งไปยังตัว Raspberry Pi ตาม Topic ที่ตั้งไว้ดังรูปที่ 16 โดยการทำงานจะเป็นไปตามดังรูปที่ 17

```
void connect_mqttServer() {
    // Loop until we're reconnected
    while (!client.connected()) {

        //first check if connected to wifi
        if(WiFi.status() != WL_CONNECTED){
            //if not connected, then first connect to
            setup_wifi();
        }

        //now attempt to connect to MQTT server
        Serial.print("Attempting MQTT connection...
        // Attempt to connect
        if (client.connect("ESP32_client1")) { // C
            //attempt successful
            Serial.println("connected");
            // Subscribe to topics here
            client.subscribe("rpi/relay1_1");
            client.subscribe("rpi/relay1_2");
            client.subscribe("rpi/relay1_3");
            client.subscribe("rpi/relay1_4");
            //client.subscribe("rpi/xyz"); //subscrib
        }
        else {
            //attempt not successful
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" trying again in 2 second

            blink_led(3,200); //blink LED three times
            // Wait 2 seconds before retrying
            delay(2000);
        }
    }
}
```

รูปที่ 14 ฟังก์ชัน connect\_mqttServer

```
void callback(char* topic, byte* message, unsigned int length) {
    Serial.print("Message arrived on topic: ");
    Serial.print(topic);
    Serial.print(". Message: ");
    String messageTemp;

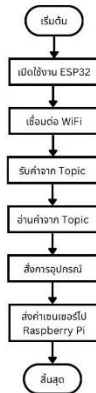
    for (int i = 0; i < length; i++) {
        Serial.print((char)message[i]);
        messageTemp += (char)message[i];
    }
    Serial.println();

    // Check if a message is received on the topic "rpi/broadcast"
    if (String(topic) == "rpi/relay1_1") {
        if(messageTemp == "OFF1"){
            digitalWrite(RL1,0);
        }
        else if(messageTemp == "ON1"){
            digitalWrite(RL1,1);
        }
    }
    if (String(topic) == "rpi/relay1_2") {
        Serial.println("Coming");
        if(messageTemp == "OFF2"){
            digitalWrite(RL2,0);
        }
        else if(messageTemp == "ON2"){
            digitalWrite(RL2,1);
        }
    }
}
```

รูปที่ 15 ฟังก์ชัน callback

```
long now = millis();
if (now - lastMsg > 4000) {
  lastMsg = now;
  snprintf (msg, MSG_BUFFER_SIZE, "%ld", t);
  snprintf (msg2, MSG_BUFFER_SIZE, "%ld", h);
  snprintf (msg3, MSG_BUFFER_SIZE, "%ld", lux);
  snprintf (msg4, MSG_BUFFER_SIZE, "%ld", map(analogRead(34), 0, 4095, 0, 100));
  client.publish("esp32/sensor1", msg); //topic name (to which this ESP32 publishes
  client.publish("esp32/sensor2", msg2); //topic name (to which this ESP32 publishes
  client.publish("esp32/sensor3", msg3); //topic name (to which this ESP32 publishes
  client.publish("esp32/sensor4", msg4); //topic name (to which this ESP32 publishes
}
```

รูปที่ 16 การส่งค่าไปยัง Raspberry Pi



รูปที่ 17 Flow Chart การทำงานของบอร์ด ESP32

#### 4. ผลการดำเนินงาน

จากการจำลองสถานการณ์ในการใช้งานเว็บแอปพลิเคชันควบคุมอุปกรณ์ไอโอทีภายในบ้าน เป็นข้อสรุปได้ว่า การใช้งานเว็บแอปพลิเคชันช่วยอำนวยความสะดวกในการเชื่อมต่ออุปกรณ์แต่ละอุปกรณ์ผ่าน Wireless และเว็บแอปพลิเคชันสามารถเข้าผ่านได้ทั้งคอมพิวเตอร์และโทรศัพท์มือถือผ่านเว็บเบราว์เซอร์

##### 4.1 หน้าจอแดชบอร์ด

หน้าจอแดชบอร์ดจะแบ่งการทำงานดังนี้

##### 4.1.1 หน้าจอล็อกอินและสมัครสมาชิก

เมื่อเริ่มต้นการใช้งานเว็บแอปพลิเคชัน

ผู้ใช้งานจะต้องสมัครสมาชิกเพื่อเข้าใช้งานดังรูปที่ 18

เมื่อทำการสมัครสมาชิกแล้วจะสามารถล็อกอินเข้าสู่ระบบโดยผู้ใช้งานเว็บแอปพลิเคชันจะแบ่งออกเป็น 2 สถานะดังนี้

1. สถานะ Member คือ สถานะที่ได้รับการอนุมัติแล้วจากผู้ดูแลระบบ

2. สถานะ Admin คือ สถานะของผู้ดูแลระบบ โดยเมื่อแรกเริ่มหลังจากสมัครสมาชิกเรียบร้อยแล้วผู้ใช้งานจะต้องรอผู้ดูแลระบบทำการอนุมัติเข้าใช้งาน โดยเมื่อล็อกอินจะขึ้นป๊อปอัพดังรูปที่ 19

รูปที่ 18 หน้าสมัครสมาชิก

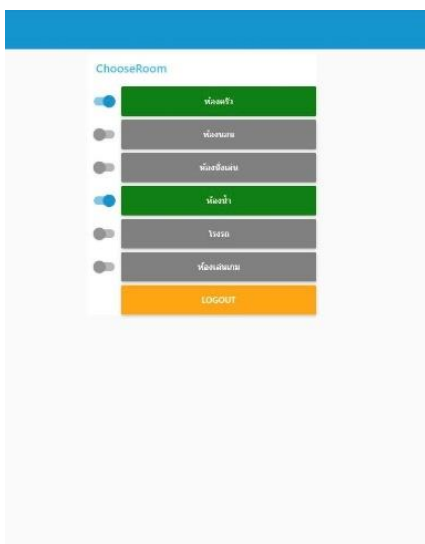
รูปที่ 19 การแจ้งเตือนล็อกอินในการรออนุมัติ



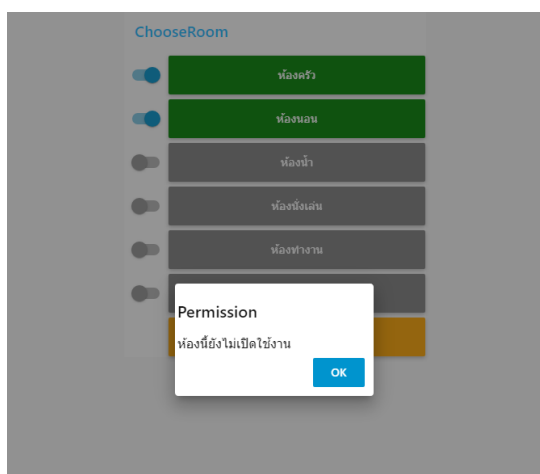
#### 4.1.2 การเข้าใช้งานห้อง

เมื่อทำการเข้าใช้งานโดยมีสถานะอนุมัติแล้ว จะขึ้นหน้าต่างให้เลือกเข้าใช้งานห้องในแต่ละชุดซึ่งในแต่ละผู้ใช้งานจะสามารถเข้าใช้ห้องได้โดยห้องที่จัดเตรียมไว้จะเข้าใช้งานได้ 6 ห้อง ดังรูปที่ 20

ในกรณีที่ไม่ได้เปิดใช้งานห้องจะขึ้นแจ้งเตือน ดังรูปที่ 21



รูปที่ 20 หน้าการเข้าใช้ห้อง



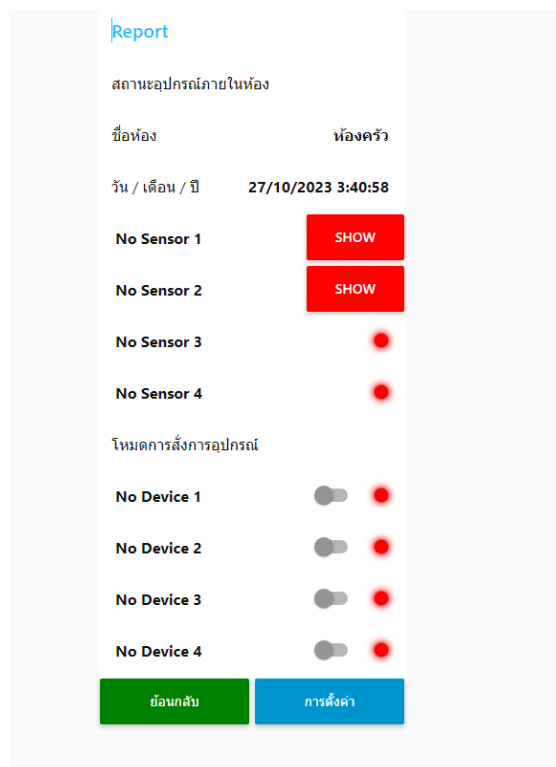
รูปที่ 21 การแจ้งเตือนเมื่อไม่ได้เปิดใช้งานห้อง

#### 4.1.3 การใช้งานในแต่ละห้อง

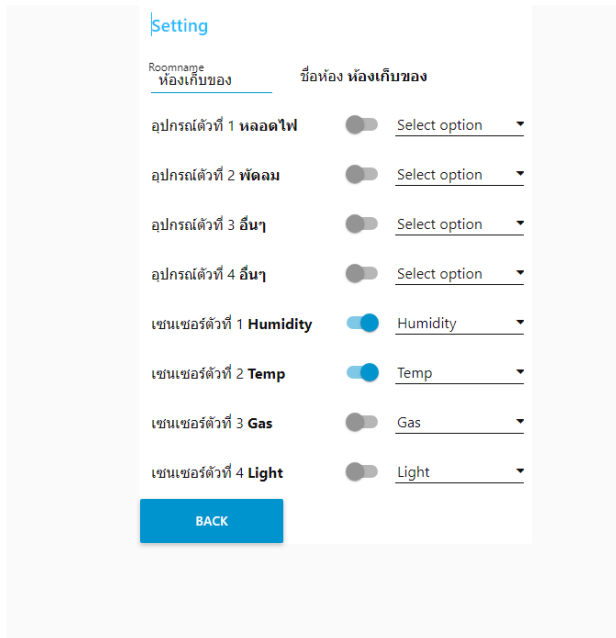
หลังจากเลือกห้องเสร็จเรียบร้อยแล้วระบบจะนำเข้าสู่การตั้งค่าของบอร์ด โดยจะแบ่งเป็น 2 ส่วน ดังนี้

1. การแสดงชื่อห้องและวันที่ รวมถึงสถานะการเปิดใช้งาน การสั่งเปิด-ปิด อุปกรณ์ และค่าเซนเซอร์ ดังรูปที่ 22

2. เมื่อกดปุ่ม Setting จะทำการแสดงการควบคุมอุปกรณ์ให้เปิดใช้งานตามต้องการ ดังรูปที่ 23



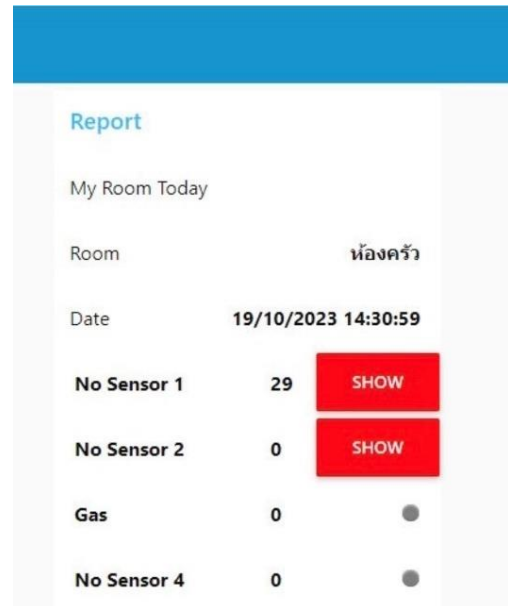
รูปที่ 22 หน้าแสดงสถานะอุปกรณ์และเซนเซอร์



รูปที่ 23 หน้าแสดงการเปิด/ปิด อุปกรณ์ภายในบ้าน



รูปที่ 24 แบบจำลองห้องเปิดหลอดไฟ



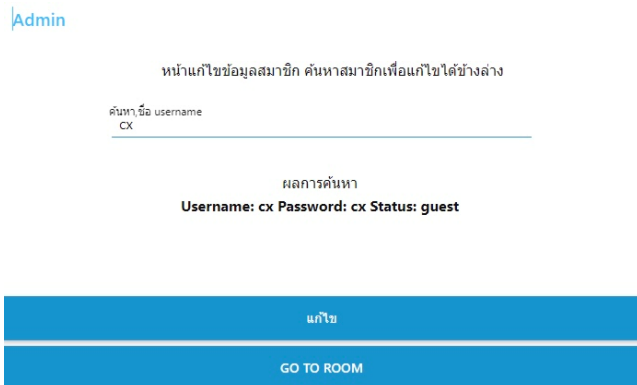
รูปที่ 25 ตัวอย่างการรับค่าเซนเซอร์

#### 4.1.4 หน้าการจัดการสมาชิกของผู้ดูแลระบบ

โดยผู้ดูแลระบบหลังจากเข้าไปใช้งานเว็บไซต์ จะเข้าสู่หน้าผู้ดูแลระบบก่อนที่จะเข้าบ้าน โดยในหน้าผู้ดูแลระบบจะมีช่องค้นหาชื่อสมาชิกเพื่อทำการแก้ไขสมาชิกและหน้าเข้าสู่หน้าเลือกห้องดังรูปที่ 26

ซึ่งผู้ดูแลระบบจะแก้ไขข้อมูลสมาชิกผ่านการค้นหาชื่อโดยพิมพ์ในช่องค้นหาและกดปุ่มแก้ไข เพื่อแก้ไขข้อมูลสมาชิก จากนั้นจะทำการตั้งค่าดังต่อไปนี้

1. แก้ไข Password ของผู้ใช้งาน
2. แก้ไขสถานะบัญชีของผู้ใช้งาน



รูปที่ 26 หน้าแสดงการจัดการของผู้ดูแลระบบ

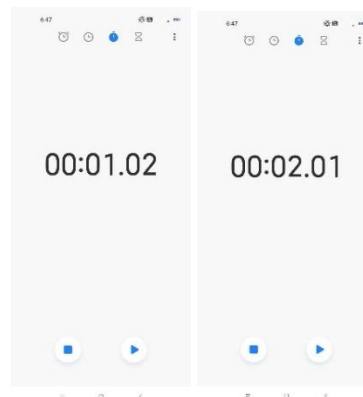
#### 4.1.5 หน้าห้อง

ในหน้าเลือกห้องดังรูปที่ 20 จะเห็นได้ว่ามีห้องให้ใช้ทั้งหมด 6 ชุดเนื่องมาจากการทดลองการส่งข้อมูลและการตั้งข้อสันนิษฐานโดยผลการทดลองจะเป็นดังตารางที่ 5

ห้อง	ความเร็ว
1	01.02
2	02.01

ตารางที่ 5 ผลการทดลองการจับเวลา

จากผลการทดลองในตารางที่ 5 จะเห็นได้ว่ายิ่งจำนวนห้องเพิ่ม จะทำให้การส่งข้อมูลช้าประมาณ 1 วินาทีเนื่องจากการเชื่อมต่อในหลายๆ Access และปัจจัยภายนอกอื่นๆ ทำให้ความเร็วช้าลงเรื่อยๆ จึงได้ทำการตั้งข้อสันนิษฐานและทำการกำหนดห้องให้ใช้จำนวน 6 ชุดโดยการเปรียบเทียบจะเป็นดังรูปที่ 27



รูปที่ 27 การจับเวลาผลการทดลอง

#### 4.1.6 การทดลองวัดอุณหภูมิและความชื้น

การทดลองการวัดอุณหภูมิจะทดลองผ่าน DHT22 ซึ่งเป็นเซนเซอร์วัดอุณหภูมิและความชื้น และทดลองเปรียบเทียบค่ากับตัวมิเตอร์ HTC-01 โดยจะมีคุณสมบัติดังนี้

1. ช่วงวัดอุณหภูมิ  $-10^{\circ}\text{C} \sim +50^{\circ}\text{C}$  ( $-14^{\circ}\text{F} \sim +122^{\circ}\text{F}$ )
2. ความแม่นยำของอุณหภูมิ  $\pm 1^{\circ}\text{C}$  ( $1.8^{\circ}\text{F}$ )
3. ความละเอียดของการวัดอุณหภูมิ  $0.1^{\circ}\text{C}$  ( $0.2^{\circ}\text{F}$ )
4. ช่วงวัดความชื้น 20%~99% RH
5. ความแม่นยำในการวัดความชื้น  $\pm 5\%$  RH
6. ความละเอียดในการวัดความชื้น 1% RH

โดยจะนำมิเตอร์ HTC-01 และเซนเซอร์ DHT22 นำไปวัดทั้งหมด 10 จุด ในสถานที่เดียวกันและเปรียบเทียบกับค่าที่โชว์บนแดชบอร์ด

การทดลองการวัดอุณหภูมิ โดยจะแบ่งออกเป็น 10 ครั้ง ดังตารางที่ 6

ครั้งที่	เซนเซอร์ (อุณหภูมิ)	มิเตอร์ (อุณหภูมิ)	ความผิดพลาด (อุณหภูมิ)
1	30	29.9	0.1
2	31	32.0	1
3	28	28.5	0.5
4	27	27.5	0.5
5	26	26.8	0.8
6	30	30	0
7	32	32.6	0.6
8	29	29.7	0.7
9	31	32.0	1
10	28	28.9	0.9

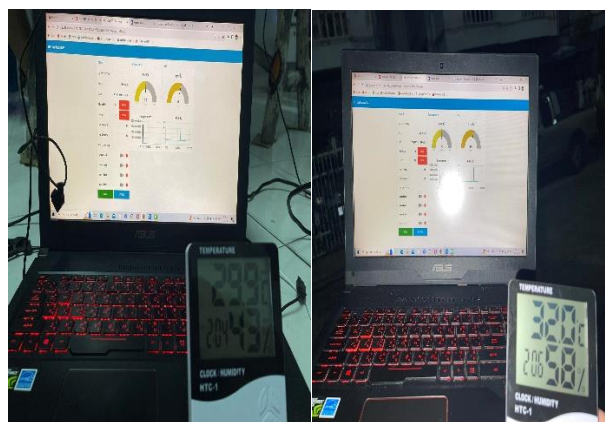
ตารางที่ 6 ผลการทดลองการวัดอุณหภูมิ

การทดลองการวัดความชื้น โดยจะแบ่งออกเป็น 10 ครั้ง ดังตารางที่ 7

ครั้งที่	เซนเซอร์ (ความชื้น)	มิเตอร์ (ความชื้น)	ความผิดพลาด (ความชื้น)
1	42	43	1
2	59	58	1
3	45	47	2
4	58	56	2
5	57	57	0
6	52	53	1
7	49	48	1
8	55	54	1
9	54	55	1
10	47	48	1

## ตารางที่ 7 ผลการทดลองการวัดความชื้น

จากผลการทดลองในตารางที่ 6 จะเห็นได้ว่าค่าที่วัดจากเครื่องวัดดังตารางที่ 6 เมื่อเทียบกับเซนเซอร์ที่นำมาใช้วัดภายในตัวบ้านมีค่าที่คลาดเคลื่อนโดยประมาณที่ 0.5 – 1 ทั้งอุณหภูมิและความชื้นโดยตัวอย่างการทดสอบจะเป็นดังรูปที่ 19



รูปที่ 28 การวัดอุณหภูมิและความชื้น

## 4.2 ผลการประเมินจากการความพึงพอใจและการทดลองการใช้งานเว็บแอปพลิเคชัน

แบบสอบถามประเมินความพึงพอใจนี้เป็นส่วนหนึ่งของปริญญานิพนธ์โดยข้อมูลที่ได้รับจะนำไปพัฒนา ปรับปรุงเว็บแอปพลิเคชันให้ดียิ่งขึ้น จากผลประเมินจำนวน 14 คนดังตารางที่ 7

ตารางที่ 6 ค่าเฉลี่ยผลการประเมินความพึงพอใจและการทดลองการใช้งานเว็บแอปพลิเคชันจำนวน 14 คน โดยค่าคะแนนเฉลี่ยแต่ละหัวข้อประเมินเต็ม 5 คะแนน

หัวข้อประเมิน	ค่าคะแนนเฉลี่ย
<b>1.ด้านความอำนวยความสะดวก</b>	4.35
1.1 หน้าแดชบอร์ดเข้าใช้งานได้ง่าย	
1.2 ขั้นตอนการสั่งการง่าย	4.14
1.3 สามารถสั่งการอุปกรณ์ได้รวดเร็ว	4.14
1.4 สามารถดูค่าสถานะต่างๆได้	4.42
<b>2. ด้านการออกแบบ</b>	4.5
2.1 สามารถใช้งานได้ทั้งคอมพิวเตอร์และโทรศัพท์	
2.2 ขนาดและรูปแบบพอดีกับความต้องการ	4.35
2.3 ความสวยงามของตัวเว็บแอปพลิเคชัน	4.42
<b>3. ด้านอุปกรณ์ใช้งาน</b>	4.5
3.1 สามารถเชื่อมต่ออุปกรณ์ในการใช้งานได้	
3.2 สามารถตั้งค่าในส่วนของซอฟต์แวร์ได้	4.5
3.3 เชื่อมต่ออุปกรณ์ได้หลากหลาย	4.5
3.4 สามารถประยุกต์นำอุปกรณ์อื่นๆมาเชื่อมต่อได้	4.57
<b>คะแนนเฉลี่ยรวม</b>	4.39

จากคะแนนเฉลี่ยหัวข้อประเมินทั้งหมดนำมาคำนวณคะแนนความพึงพอใจและการใช้งานโดยรวมของเว็บแอปพลิเคชันได้ 4.39 เต็ม 5 คะแนน

## 5. สรุปผลการดำเนินงาน

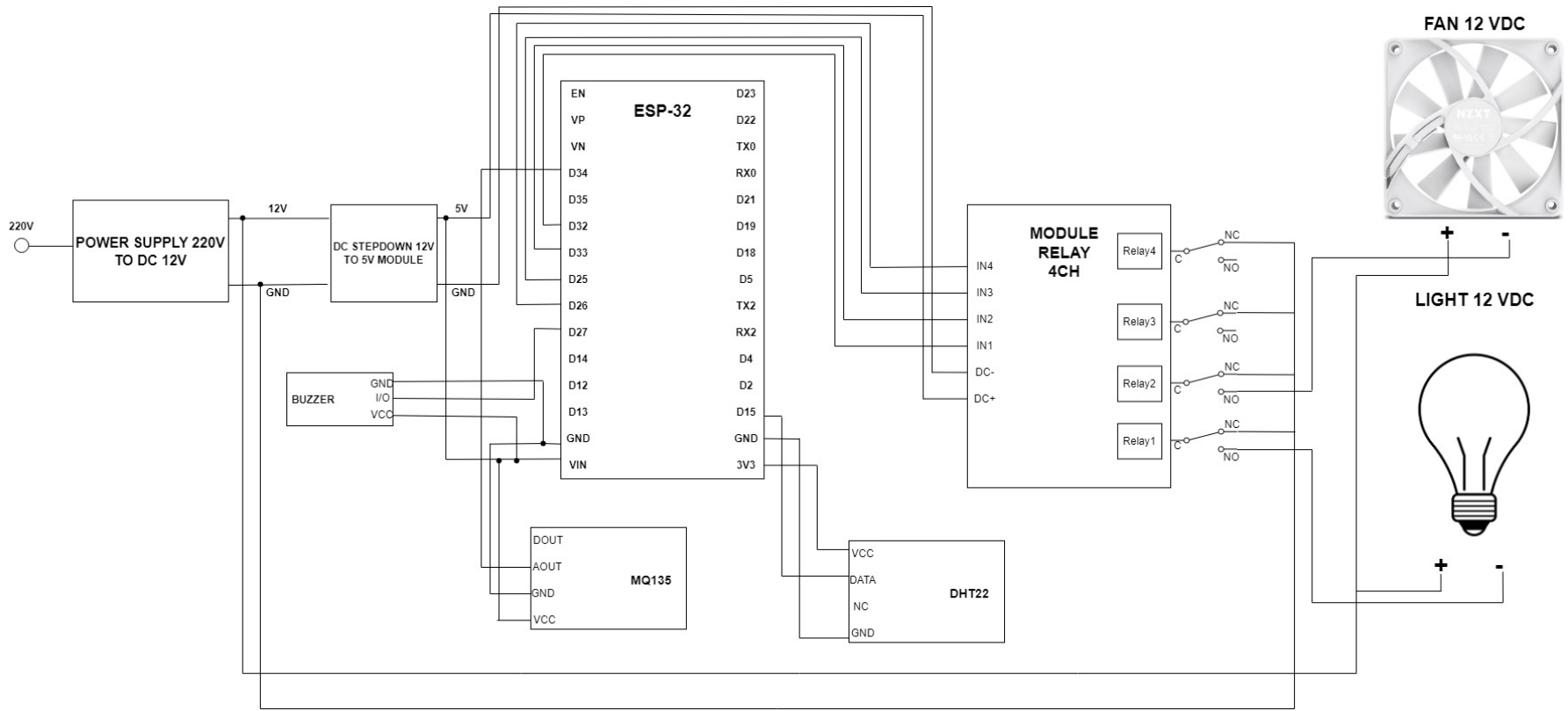
โครงการปริญญานิพนธ์นี้ได้นำเสนอเว็บแอปพลิเคชันควบคุมอุปกรณ์ไอโอทีภายในบ้านโดยบอร์ด Raspberry Pi พัฒนาโปรแกรมส่วนติดต่อฮาร์ดแวร์ด้วยภาษา Python และ เว็บแอปพลิเคชันโดย Node-Red พัฒนาด้วยภาษา JavaScript โดยจะใช้งานฐานข้อมูลคือ PostgreSQL ในการจัดเก็บข้อมูลและใช้บอร์ด Esp32 ในการติดต่อระหว่าง Raspberry Pi และ อุปกรณ์ฮาร์ดแวร์ พัฒนาด้วยภาษา C++ ผ่านโปรแกรม Arduino

การเข้าใช้งานเว็บแอปพลิเคชันผู้ใช้งานต้องเข้าสู่ระบบโดยการสมัครบัญชีผู้ใช้และได้รับการอนุมัติจากผู้ดูแลระบบจึงจะสามารถใช้งานเว็บแอปพลิเคชันได้และจะต้องได้รับสิทธิ์ในการเข้าใช้ห้องผ่านผู้ดูแลระบบอีกชั้นหนึ่งเพื่อป้องกันการเข้าใช้งานจากบุคคลอื่น

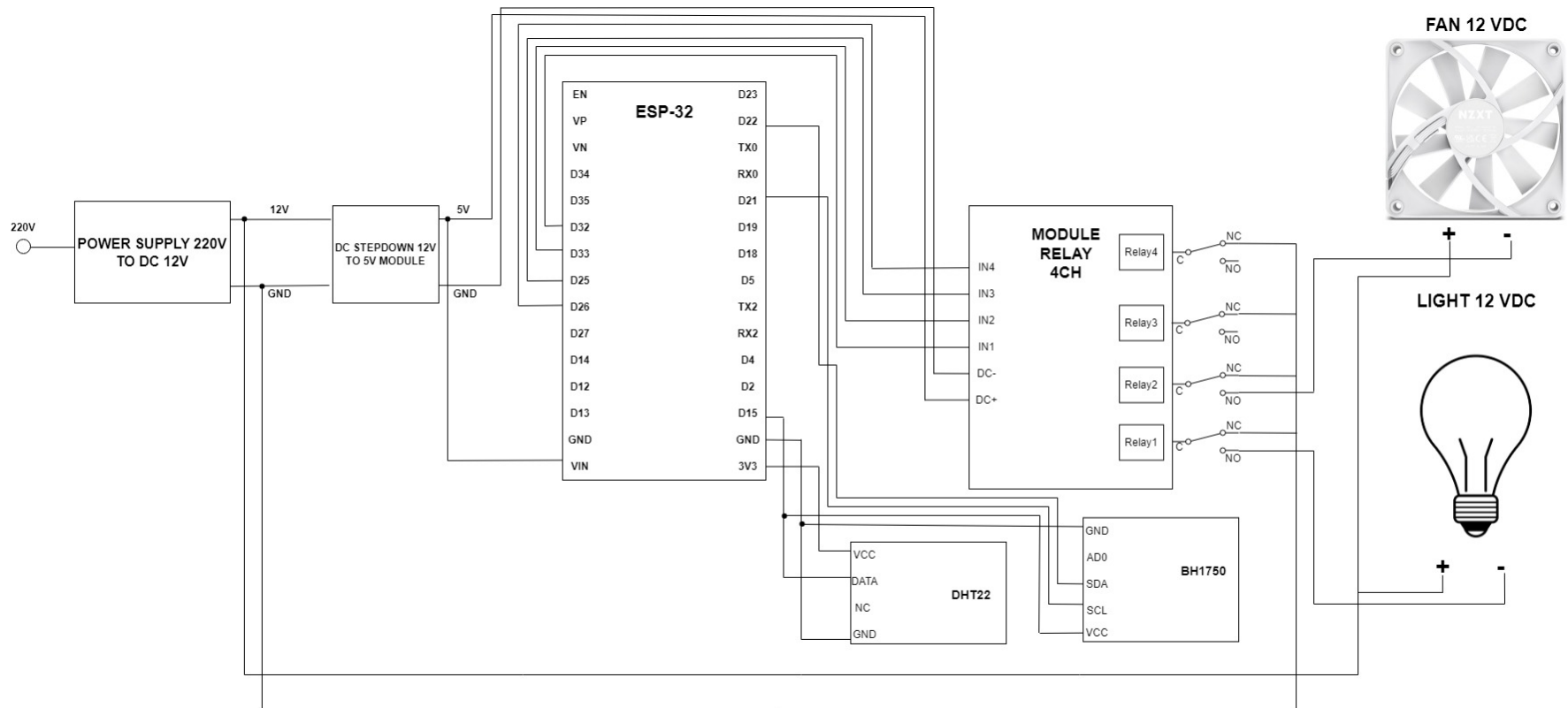
จากการทดสอบพบว่าสามารถสั่งการห้องได้สะดวกและสามารถนำอุปกรณ์อื่นมาใช้งานได้ในงานอื่นโดยแก้ไขหรือเพิ่มอุปกรณ์บนหน้าแดชบอร์ดได้หลากหลายโดยไม่จำเป็นต้องใช้ห้องเฉพาะที่กำหนด

## 6. เอกสารอ้างอิง

- [1] ตอนที่ 1 รู้จักกับบอร์ด Raspberry Pi - งานไฟฟ้า และอิเล็กทรอนิกส์, 2016. [Online]. Available: <http://know2learning.blogspot.com/2016/02/1-raspberry-pi.html> [Accessed: 10- June-2021]
- [2] Python คืออะไร - ภาษา python ใช้ทำอะไร, [Online]. Available: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2417-python-คืออะไร.html> [Accessed: 10- June-2021]
- [3] JavaScript คืออะไร จาวา สคริปต์ คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต, 2017. [Online]. Available: <https://www.mindphp.com/คู่มือ/73-คืออะไร/2187-java-javascript-คืออะไร.html> [Accessed: 10- June-2021]
- [4] Firebase คืออะไร | 4 Xtreme Co.,Ltd. 2020. [Online]. Available: <https://www.4xtreme.com/2020/11/20/firebase-คืออะไร/> [Accessed: 11- June-2021]
- [5] ทำความรู้จัก Firebase และผลิตภัณฑ์ต่าง ๆ ในช่วงต้นปี 2019 กัน . 2019. [Online]. Available: <https://medium.com/@sirawit/firebase-คืออะไร-ทำความรู้จัก-firebase-ในช่วงต้นปี-2019-กัน-473a8e8699fb> [Accessed: 11- June-2021]
- [6] รู้จัก Cloud Functions for Firebase ตั้งแต่ Zero จนเป็นHero, 2017. [Online]. Available: <https://medium.com/firebase-thailand/รู้จัก-cloud-functions-for-firebase-ตั้งแต่-zero-จนเป็น-hero-1c94acbb55af> [Accessed: 11- June-2021]
- [7] Google Cloud Firestore ระบบฐานข้อมูลเก็บเอกสารแบบ NoSQL เข้าสู่สถานะ GA แล้ว, 2017. [Online]. Available: <https://www.blognone.com/node/107898> [Accessed: 11- June-2021]
- [8] MQTT. Retrieved December 4, 2022, accessed from : <https://www.mindphp.com/บทความ-mqtt.html>
- [9] Node Red คืออะไร. Retrieved December 4, 2022, accessed from : <http://pantamitsombaddee.blogspot.com/p/node-red-node-red-apis-application.html>
- [10] ESP32 คืออะไร. Retrieved , 2022, accessed from : <https://techtalk2apply.com/what-is-esp32/>
- [11] DHT22 คืออะไร. Retrieved , 2022, accessed from : <https://www.arduino4.com/product/23/dht22-am2302-temperature-humidity-sensor-โมดูลวัดอุณหภูมิความชื้น>



ภาคผนวก (ก) แผงวงจรที่ 1



ภาคผนวก (ข) แผงวงจรที่ 2



## ประวัติผู้จัดทำ



ชื่อ – นามสกุล : นางสาวอมรรัตน์ วุฒิเจริญฤทธิ์

อีเมล : s6203052412164@email.kmutnb.ac.th

### ประวัติการศึกษา

พ.ศ. 2566	ปริญญตรี เทคโนโลยีอิเล็กทรอนิกส์ (คอมพิวเตอร์) มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
พ.ศ. 2561	ประกาศนียบัตรวิชาชีพชั้นสูง คอมพิวเตอร์ธุรกิจ วิทยาลัยเทคนิคราชบุรี จังหวัดราชบุรี

## ประวัติผู้จัดทำ



ชื่อ – นามสกุล : นายศนันท์ ชินารักษ์

อีเมล : s6203052422178@email.kmutnb.ac.th

### ประวัติการศึกษา

- |           |  |
|-----------|--|
| พ.ศ. 2566 | ปริญญาตรี เทคโนโลยีอิเล็กทรอนิกส์ (คอมพิวเตอร์)<br>มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ        |
| พ.ศ. 2561 | ประกาศนียบัตรวิชาชีพชั้นสูง อิเล็กทรอนิกส์อุตสาหกรรม<br>วิทยาลัยเทคนิคสมุทรปราการ จังหวัดสมุทรปราการ |