รายงานโครงงาน

ระบบควบคุมและสั่งการ IoT หลายบอร์ดแบบรวมศูนย์

Universal Multi-Board IoT Control & Automation System

จัดทำโดย

นายธนภัทร เลิศพิทักษ์สิทธิ์ 67332310017-9

เสนอ

อาจารย์ ประภาส ผ่องสนาม

รายงานนี้เป็นส่วนหนึ่งของรายวิชาการออกแบบระบบอินเทอร์เน็ตของสรรพสิ่ง

คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น

ภาคเรียนที่ 1 ปีการศึกษา 2568

รายงานโครงงาน

ระบบควบคุมและสั่งการ IoT หลายบอร์ดแบบรวมศูนย์

Universal Multi-Board IoT Control & Automation System

จัดทำโดย

นายธนภัทร เลิศพิทักษ์สิทธิ์ 67332310017-9

เสนอ

อาจารย์ ประภาส ผ่องสนาม

รายงานนี้เป็นส่วนหนึ่งของรายวิชาการออกแบบระบบอินเทอร์เน็ตของสรรพสิ่ง

คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น

ภาคเรียนที่ 1 ปีการศึกษา 2568

กิตติประกาศ

โครงงานเรื่อง “ระบบควบคุมและสั่งการ IoT หลายบอร์ดแบบรวมศูนย์” ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดีด้วยความกรุณาและความช่วยเหลืออย่างดียิ่งจาก **อาจารย์ประภาส ผ่องสนาม** อาจารย์ประจำวิชา ที่ได้กรุณาให้คำปรึกษา แนะนำแนวทางในการแก้ไขปัญหา ตลอดจนตรวจสอบและปรับปรุงเนื้อหาของรายงานฉบับนี้ให้มีความสมบูรณ์ยิ่งขึ้น ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณภาควิชาการออกแบบระบบอินเทอร์เน็ตของสรรพสิ่ง คณะวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น ที่ได้สนับสนุนทรัพยากร อุปกรณ์ และสถานที่ในการดำเนินงานวิจัย รวมถึงคณาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ ซึ่งเป็นรากฐานสำคัญในการพัฒนาโครงงานชิ้นนี้

นอกจากนี้ ขอขอบคุณเพื่อน ๆ ที่คอยให้กำลังใจและแลกเปลี่ยนความรู้ และที่สำคัญที่สุด ขอกราบขอบพระคุณบิดา มารดา ที่คอยให้การสนับสนุนทั้งด้านกำลังใจและทุนทรัพย์ในการศึกษาเสมอมา

คุณค่าและประโยชน์ใด ๆ ที่เกิดจากโครงงานฉบับนี้ ผู้จัดทำขอมอบเป็นเครื่องบูชาพระคุณบิดา มารดา ครูบาอาจารย์ และผู้มีพระคุณทุกท่านที่มีส่วนร่วมในการส่งเสริมให้การดำเนินงานครั้งนี้ประสบความสำเร็จ

คณะผู้จัดทำ

นายธนภัทร เลิศพิทักษ์สิทธิ์

(22/02/2569)

สารบัญ

บทที่ 1: บทนำ (Introduction)

1.1 ที่มาและความสำคัญ

ในปัจจุบันการใช้งานอุปกรณ์ Internet of Things (IoT) มีการแพร่หลายอย่างมาก แต่ปัญหาที่ผู้ใช้งานมักพบคือ "ความกระจัดกระจาย" ของการควบคุมอุปกรณ์ หากมีบอร์ดควบคุมหลายตัว ผู้ใช้อาจต้องเข้าใช้งานผ่านหลายหน้าจอหรือต้องตั้งค่าแยกกัน โครงงานนี้จึงพัฒนา **Centralized Dashboard** เพื่อจัดการบอร์ด ESP32 หลายตัวพร้อมกันผ่านหน้าเว็บเดียว โดยใช้ MAC Address เป็นตัวระบุตัวตนอัตโนมัติ ช่วยลดความยุ่งยากในการเขียนโปรแกรมใหม่ทุกครั้งที่เพิ่มอุปกรณ์

1.2 วัตถุประสงค์

1. เพื่อสร้างระบบควบคุมส่วนกลาง (Centralized Control) ที่เข้าถึงได้ผ่านอินเทอร์เน็ต
2. เพื่อพัฒนาระบบสั่งการอัตโนมัติที่ยืดหยุ่น ทั้งแบบตั้งช่วงเวลา (Timer) และนับเวลาถอยหลัง (Duration)
3. เพื่อตรวจสอบสถานะการเชื่อมต่อ (Online/Offline) และบันทึกประวัติการทำงานแบบ Real-time
4. เพื่อศึกษาการใช้งาน FreeRTOS ในการจัดการ Multitasking บน ESP32

1.3 ขอบเขตของเทคโนโลยีที่ใช้

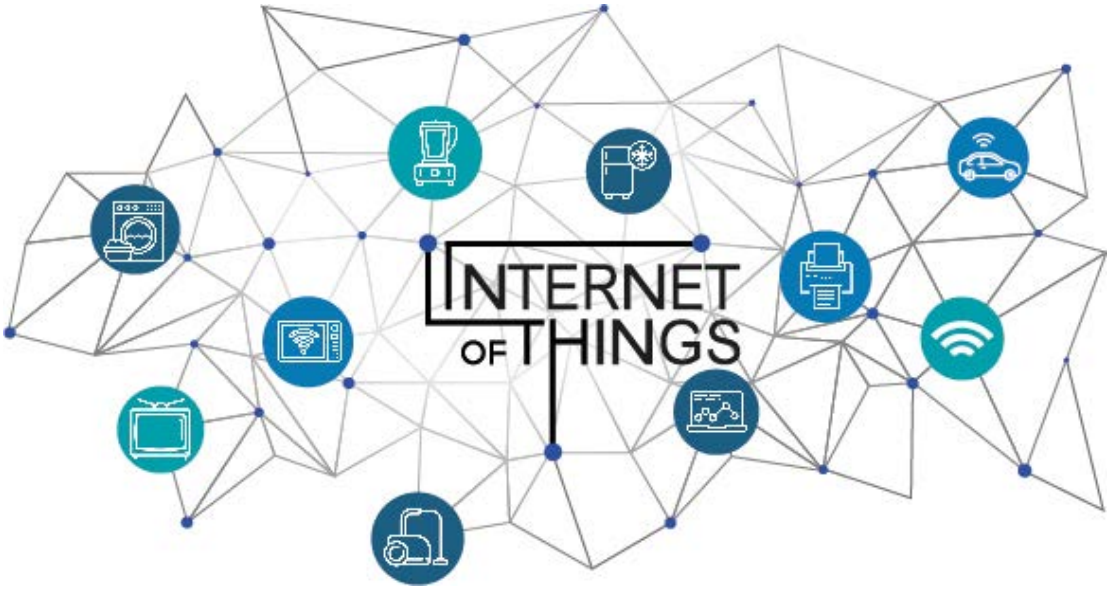
- **Hardware:** ESP32 Microcontroller, Relay Module, LED Actuators
- **Software:** C++ (Arduino/FreeRTOS), PHP (Backend), MySQL (Database), Bootstrap 5 (UI)
- **Protocol:** RESTful API สื่อสารด้วยรูปแบบ JSON

บทที่ 2: ทฤษฎีและงานวิจัยที่เกี่ยวข้อง (Literature Review)

2.1 แนวคิด Internet of Things (IoT)

IoT คือเครือข่ายของอุปกรณ์ที่ฝังเซนเซอร์และซอฟต์แวร์เพื่อแลกเปลี่ยนข้อมูล กระบวนการทำงานประกอบด้วย:

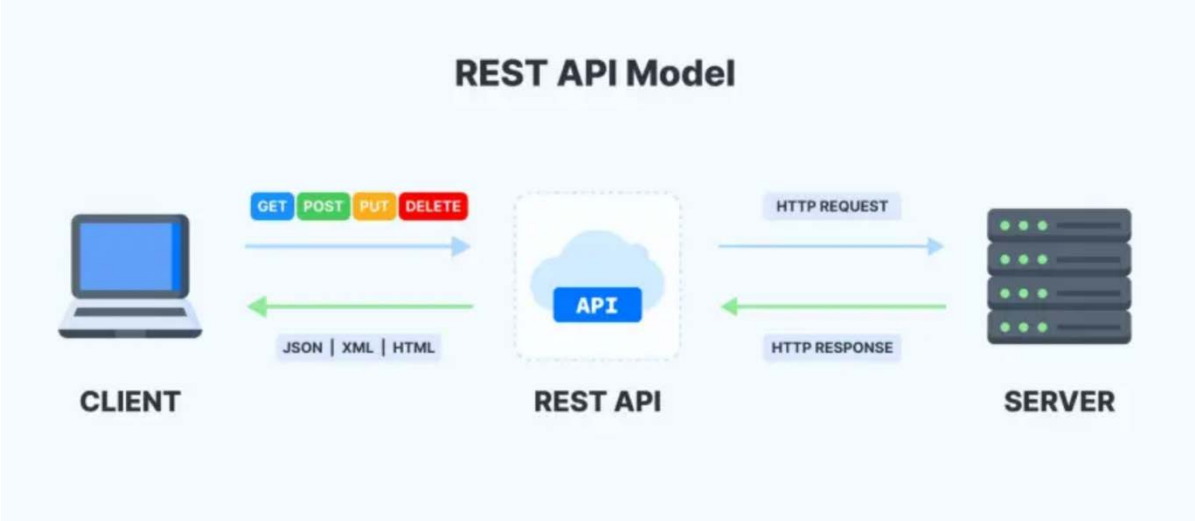
- **Sensor/Actuator:** รับค่าหรือทำงานตามสั่ง
- **Gateway:** เชื่อมต่ออินเทอร์เน็ต
- **Cloud/Server:** ประมวลผลและเก็บข้อมูล
- **User Interface:** ส่วนติดต่อผู้ใช้เพื่อสั่งการ



ภาพที่ 1 อินเตอร์เน็ตของสรรพสิ่ง

2.2 สถาปัตยกรรม Client-Server และ RESTful API

ระบบนี้ใช้โมเดล **Client-Server** โดยที่ ESP32 ทำหน้าที่เป็น Client ส่ง Request ไปยัง PHP Server ที่ทำหน้าที่เป็นผู้ให้บริการ ข้อมูลที่รับส่งจะอยู่ในรูปแบบ **JSON** ซึ่งมีโครงสร้างที่อ่านง่าย เช่น {"status": "on", "duration": 30}



ภาพที่ 2 Rest APL Model

2.3 FreeRTOS และ Multitasking

เพื่อให้ ESP32 ทำงานได้อย่างเสถียร ระบบจึงใช้ **FreeRTOS** ในการแบ่ง Priority ของงาน (Tasks) เช่น งานเชื่อมต่อ WiFi จะไม่ถูกขัดจังหวะด้วยการอ่านค่าเซนเซอร์ ทำให้ระบบไม่ค้างแม้มีการส่งข้อมูลจำนวนมาก

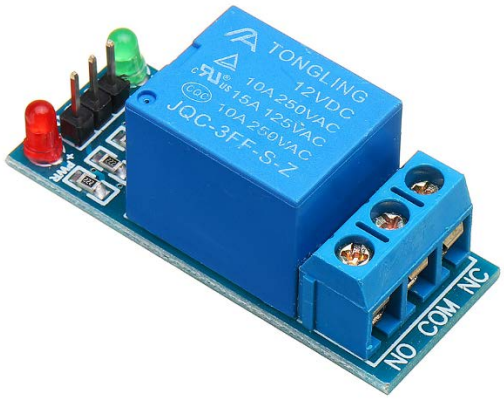
2.4 Introduction to ESP32

ESP32 มาพร้อมกับไมโครคอนโทรลเลอร์ 32 บิตในตัว พร้อมคุณสมบัติ Wi-Fi, Bluetooth และ BLE ที่รองรับการใช้งานหลากหลายประเภท เป็นผลิตภัณฑ์ในกลุ่มอุปกรณ์ประหยัดพลังงานและราคาประหยัดที่พัฒนาโดย Espressif Systems

ภาพที่ 3 Introduction to ESP32

2.6 Relay module

รีเลย์คือสวิตช์ที่เปิดหรือปิดวงจรไฟฟ้าเมื่อได้รับสัญญาณระหว่างอุปกรณ์อิเล็กทรอนิกส์ดิจิทัลกำลังต่ำและอุปกรณ์กำลังสูง มันมีประโยชน์เมื่อสิ่งที่คุณต้องการควบคุมต้องการพลังงาน (แรงดัน/กระแส) ที่สูงกว่าที่ไมโครคอนโทรลเลอร์สามารถให้ได้ Arduino สามารถให้ได้สูงสุดเพียง 5V และ 40 mA เท่านั้น ในบทเรียนนี้ผมจะใช้ปั๊มน้ำและโมดูลรีเลย์ 1 ช่อง 5V เป็นตัวอย่าง แต่สามารถนำไปใช้กับสิ่งอื่นๆ ได้อีกมากมาย เช่น ไฟและแอคชูเอเตอร์ โมดูลรีเลย์อาจแตกต่างกันไปในแต่ละรุ่น และมีพิกัดแรงดันและกระแสสูงสุด รวมถึงข้อกำหนดด้านพลังงานที่แตกต่างกัน

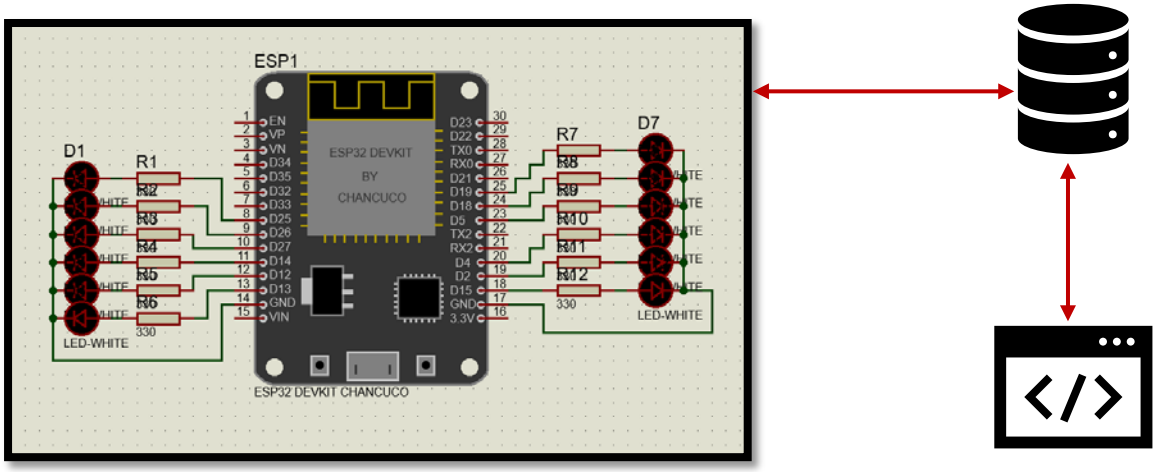


ภาพที่ 4 Relay Module

บทที่ 3: การออกแบบและสถาปัตยกรรมระบบ (System Architecture)

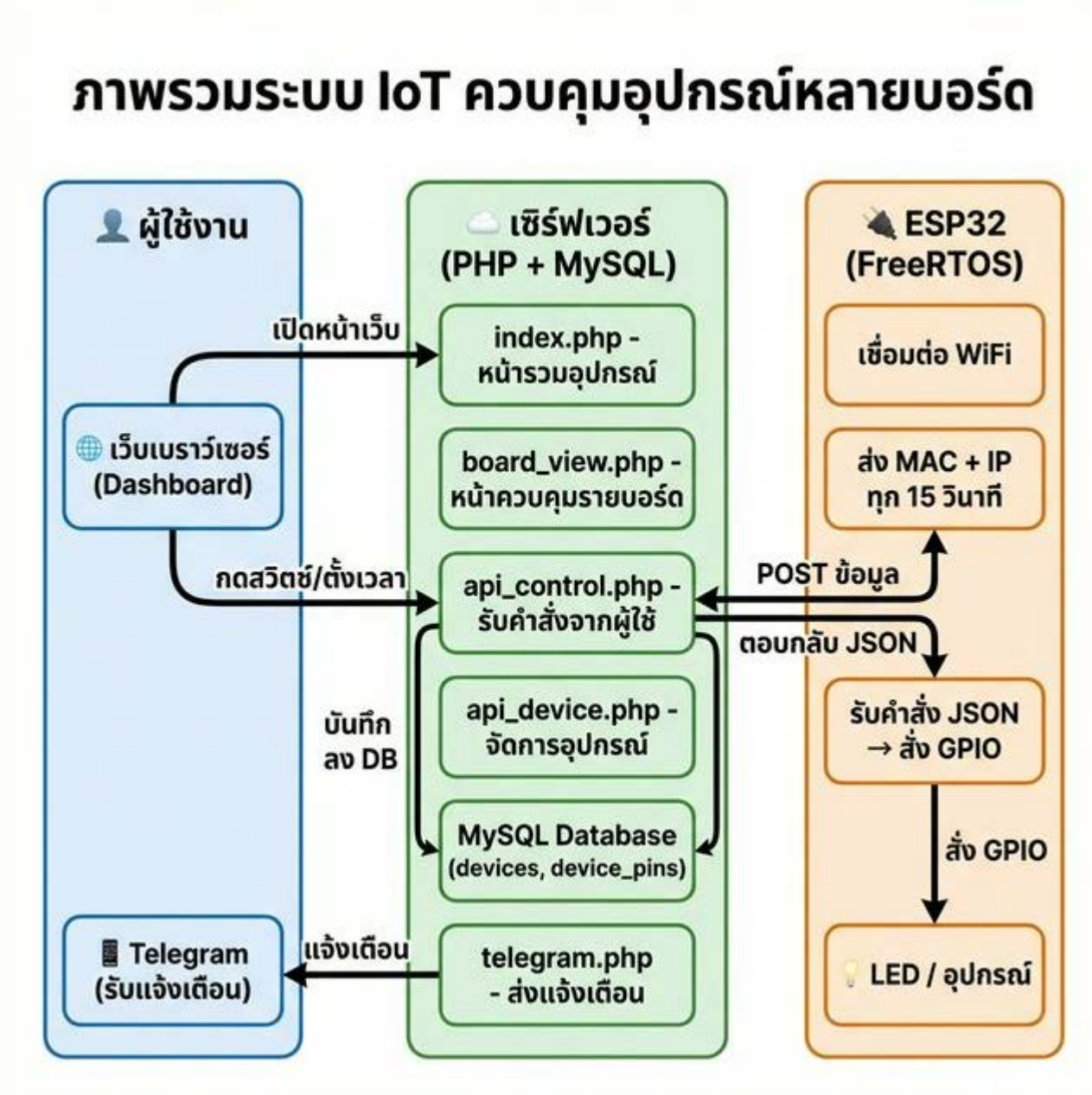
3.1 การออกแบบ

3.1.1 การออกแบบการทำงานของโครงงานเบื้องต้น



ภาพที่ 5 แนวคิดและการออกแบบเบื้องต้น

3.1.2 การออกแบบการทำงานของโครงงานฉบับสมบูรณ์



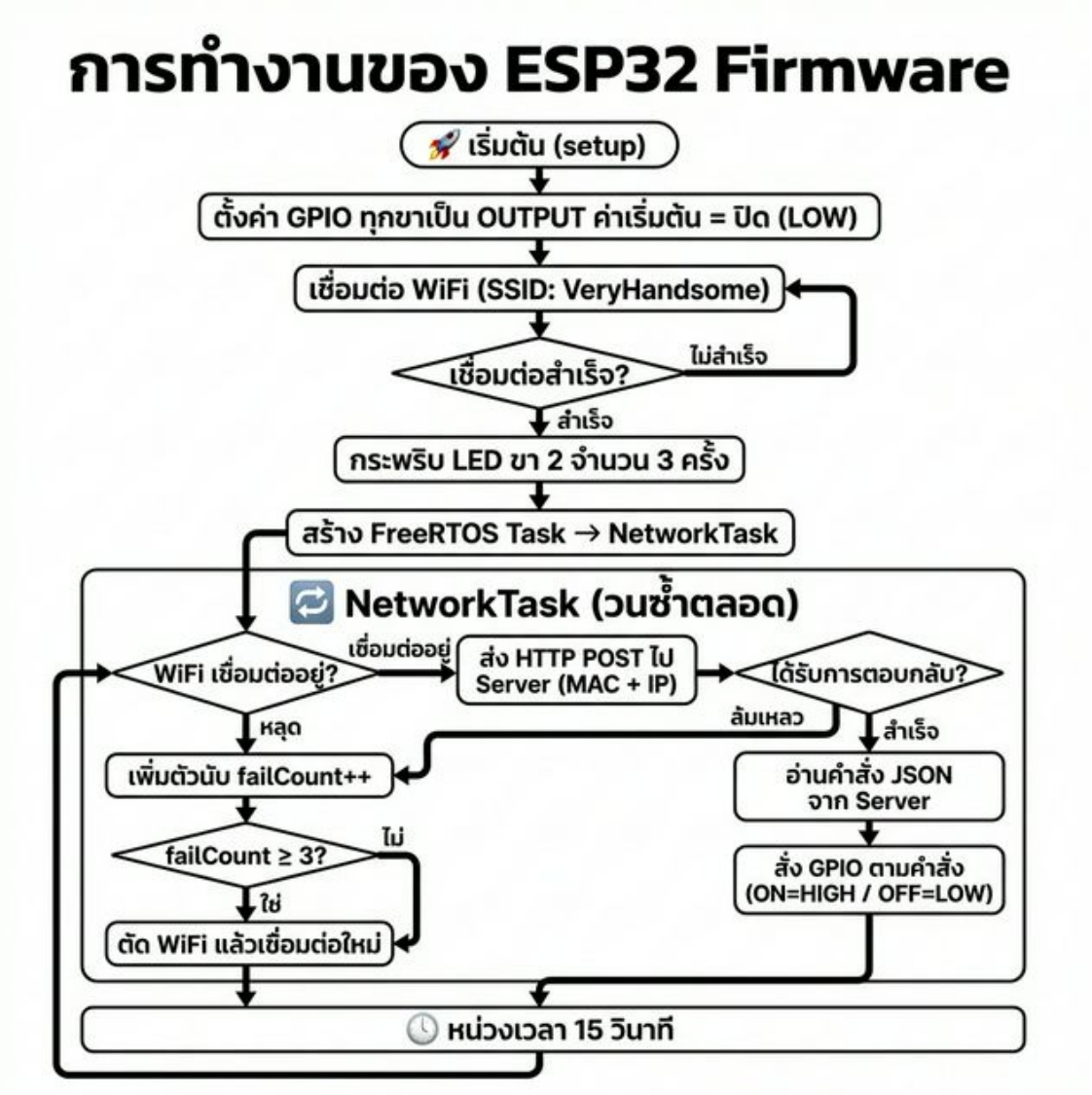
ภาพที่ 6 ภาพรวมระบบ IoT ควบคุมอุปกรณ์หลายบอร์ด

3.2 ภาพรวมการทำงาน (System Logic)

ระบบถูกออกแบบให้ Server เป็น "สมองส่วนกลาง" (Brain) ในขณะที่ ESP32 เป็น "แขนขา" (Actuators):

1. **ESP32** จะส่ง MAC Address ไปที่ Server ทุก 15 วินาที (Polling)
2. **Server** ตรวจสอบใน Database ว่าบอร์ดนี้มีคำสั่งใหม่หรือไม่ (Manual/Timer/Duration)
3. **Server** ส่งคำตอบกลับเป็น JSON

4. **ESP32** รับคำสั่งไปควบคุม Relay และส่งสถานะกลับมายืนยัน



ภาพที่ 7 การทำงานของ ESP32 Firmware

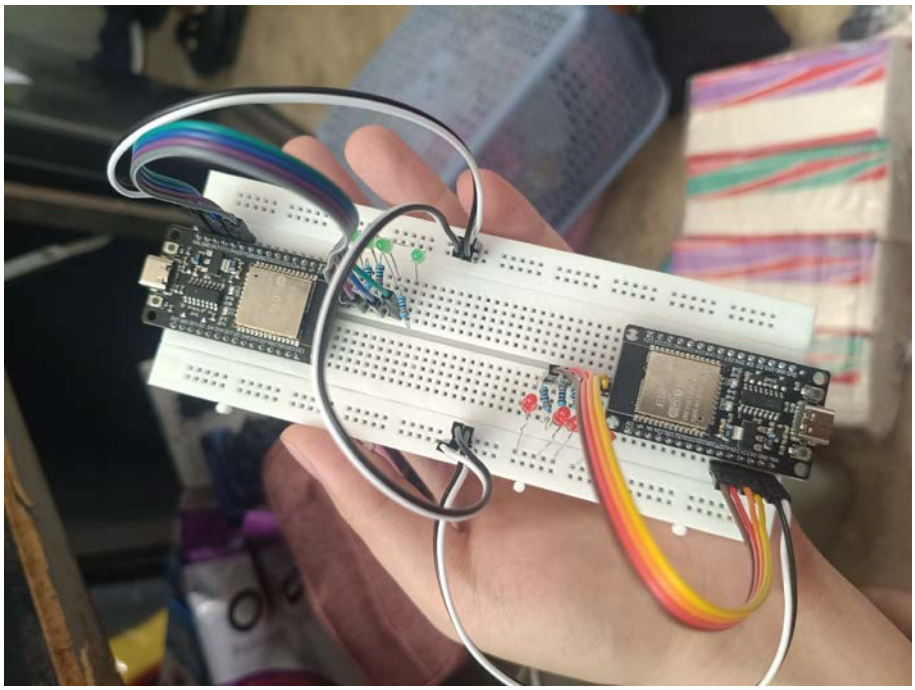
บทที่ 4: วิธีดำเนินงานและคู่มือการใช้งาน (Methodology & User Manual)

4.1 รายการอุปกรณ์ (Hardware Specifications)

- **ESP32:** หน่วยประมวลผลหลัก รองรับ Dual-core
- **Relay Module:** สวิตช์อิเล็กทรอนิกส์สำหรับควบคุมเครื่องใช้ไฟฟ้า
- **Power Supply:** แหล่งจ่ายไฟ 5V/2A เพื่อความเสถียร

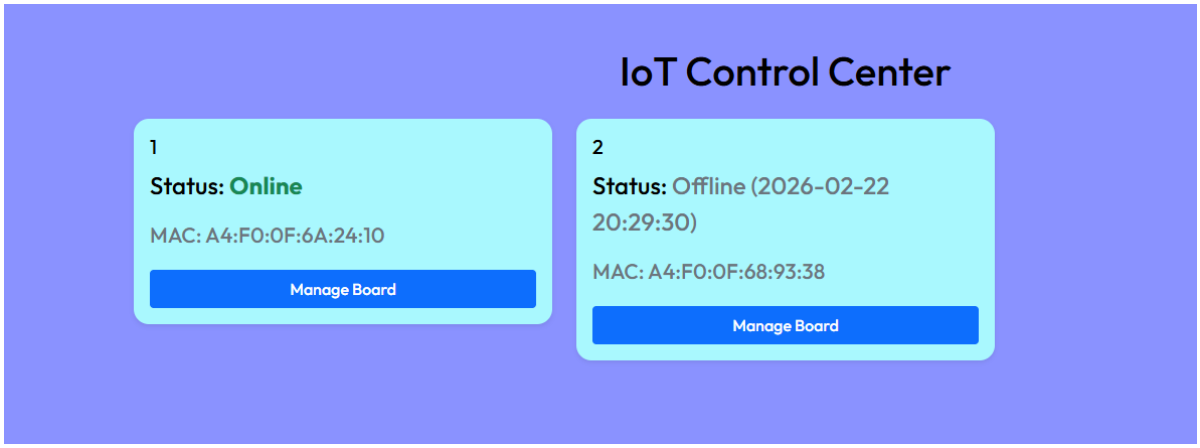
4.2 คู่มือการใช้งานระบบ

1. **การติดตั้ง:** จ่ายไฟให้บอร์ด ESP32 ระบบจะเชื่อมต่อ WiFi ที่ตั้งค่าไว้โดยอัตโนมัติ



ภาพที่ 8 การติดตั้งและทดสอบด้วย LED

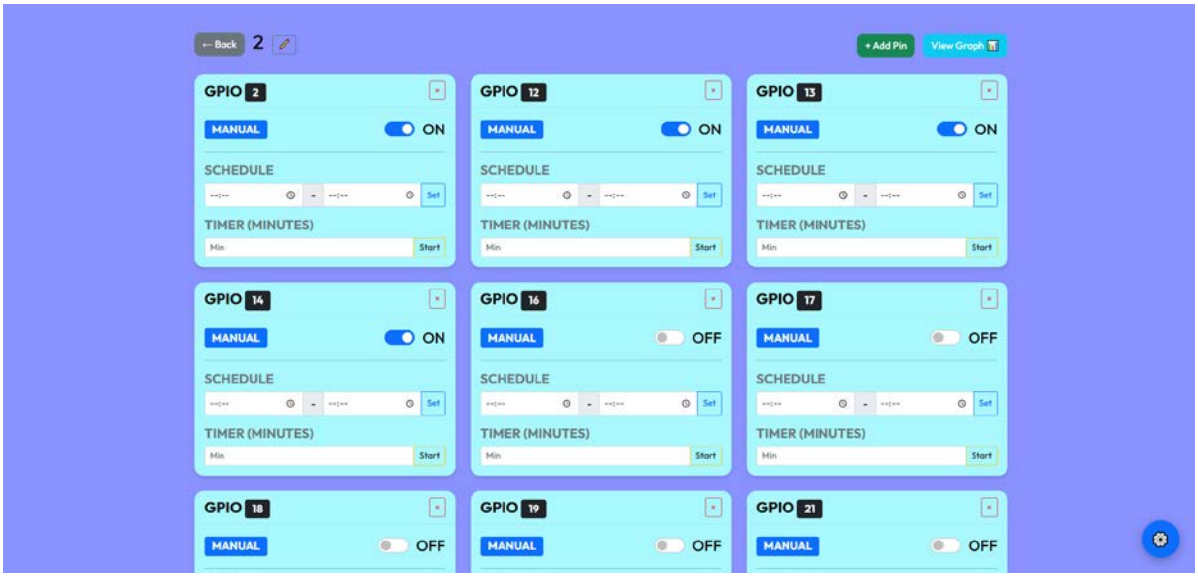
2. **การตรวจสอบสถานะ:** เข้าหน้า Dashboard เพื่อดูรายการบอร์ดทั้งหมด



ภาพที่ 9 Dashboard (ออนไลน์,ออฟไลน์)

- **สีเขียว:** ออนไลน์ พร้อมสั่งการ
- **สีเทา:** ออฟไลน์ (ขาดการติดต่อ)

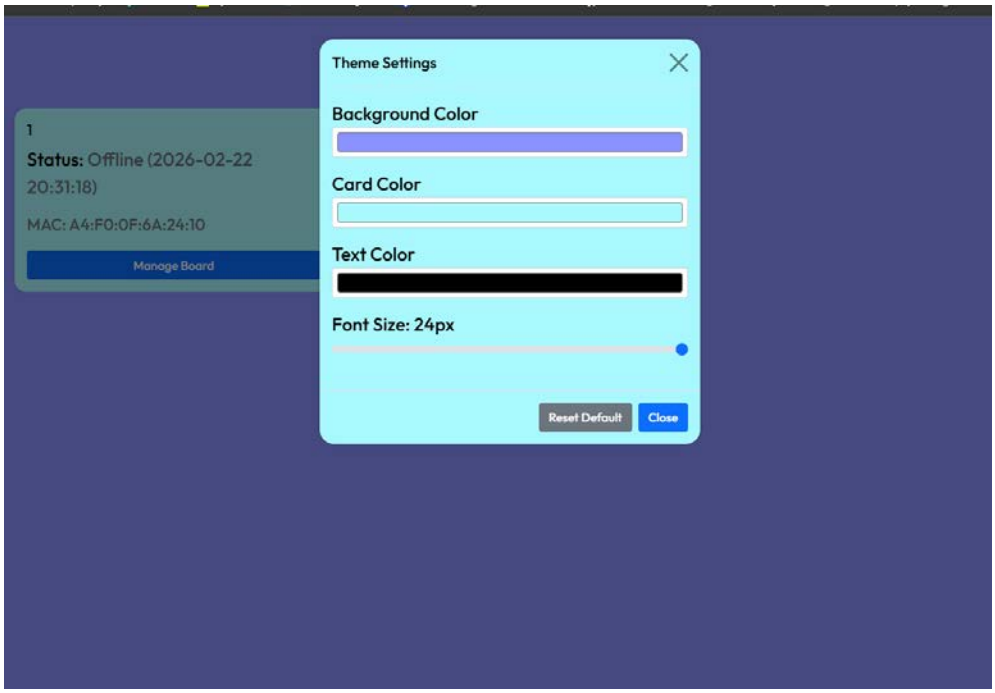
3. **การสั่งการ:** * **Manual:** กดเปิด-ปิดผ่านหน้าเว็บทันที



ภาพที่ 10 Dashboard-สั่งการ

- **Timer:** ตั้งเวลาล่วงหน้า (เช่น เปิด 08:00 ปิด 17:00)
- **Duration:** ตั้งให้ทำงานเป็นระยะเวลาหนึ่งแล้วดับเอง

4. **การปรับแต่ง:** สามารถเลือกธีม Dark/Light Mode และปรับขนาดตัวอักษรได้ผ่านเมนู Settings



ภาพที่ 11 Dashboard-ปรับแต่ง

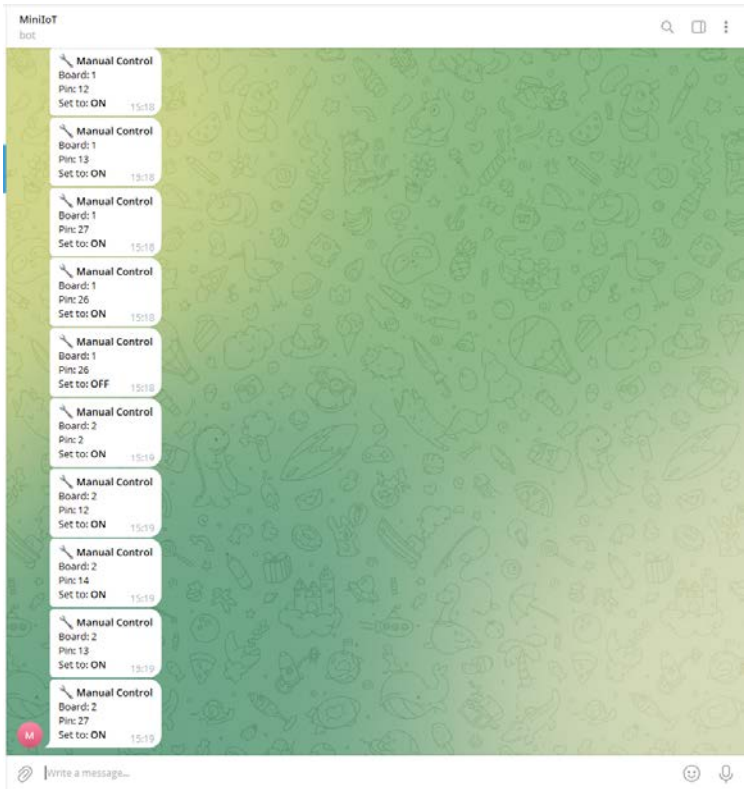
บทที่ 5: สรุปผลและการอภิปราย (Conclusion)

โครงงานนี้ประสบความสำเร็จในการสร้างระบบควบคุมแบบรวมศูนย์ (Centralized Control) ซึ่งช่วยแก้ปัญหาความซับซ้อนในการจัดการอุปกรณ์ IoT หลายตัว อย่างไรก็ตาม มีข้อควรระวังเรื่อง **Single Point of Failure** หาก Server ล่ม ระบบทั้งหมดจะไม่สามารถสั่งการได้ ในอนาคตจึงควรพัฒนาให้มีการประมวลผลที่ขอบเครือข่าย (Edge Computing) ร่วมด้วย

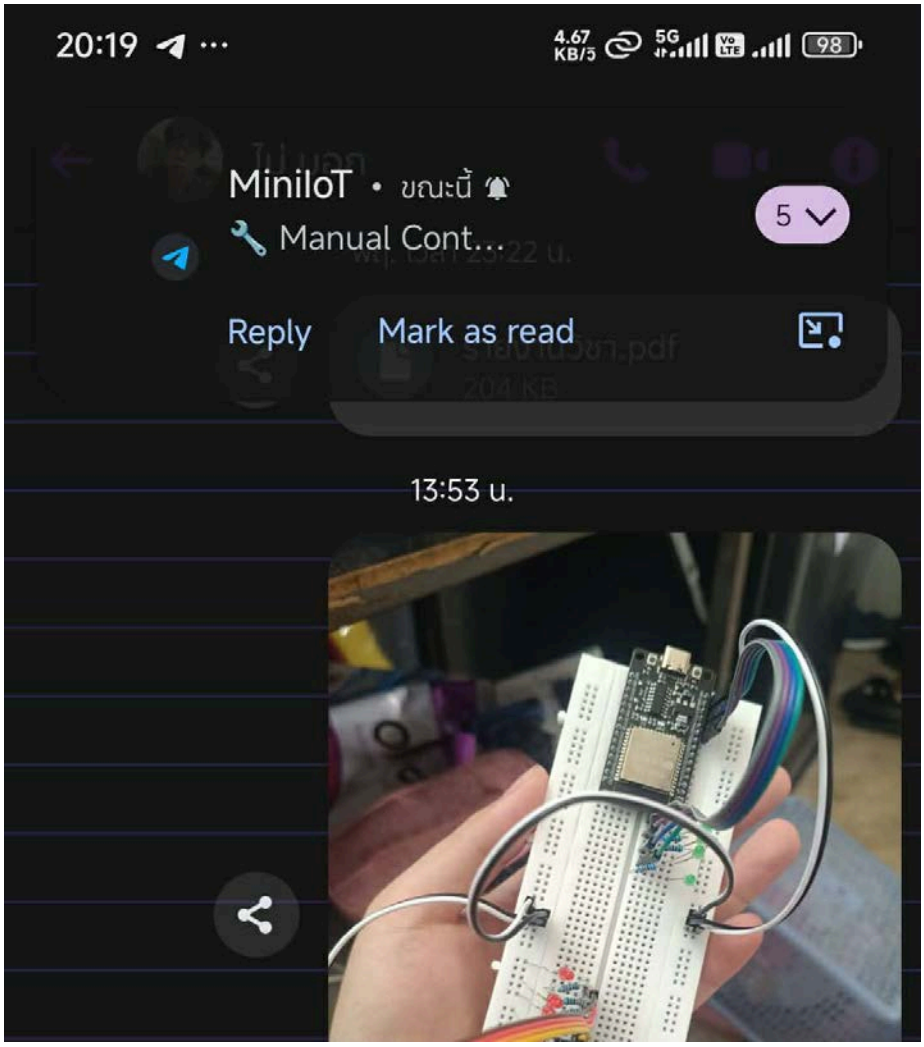
ภาคผนวก (Appendix)

- **การแจ้งเตือน:** ระบบเชื่อมต่อ Telegram Bot API เพื่อแจ้งเตือนสถานะเมื่อบอร์ด Online/Offline

ภาพที่ 12 การแจ้งเตือน Telegram Bot API PC

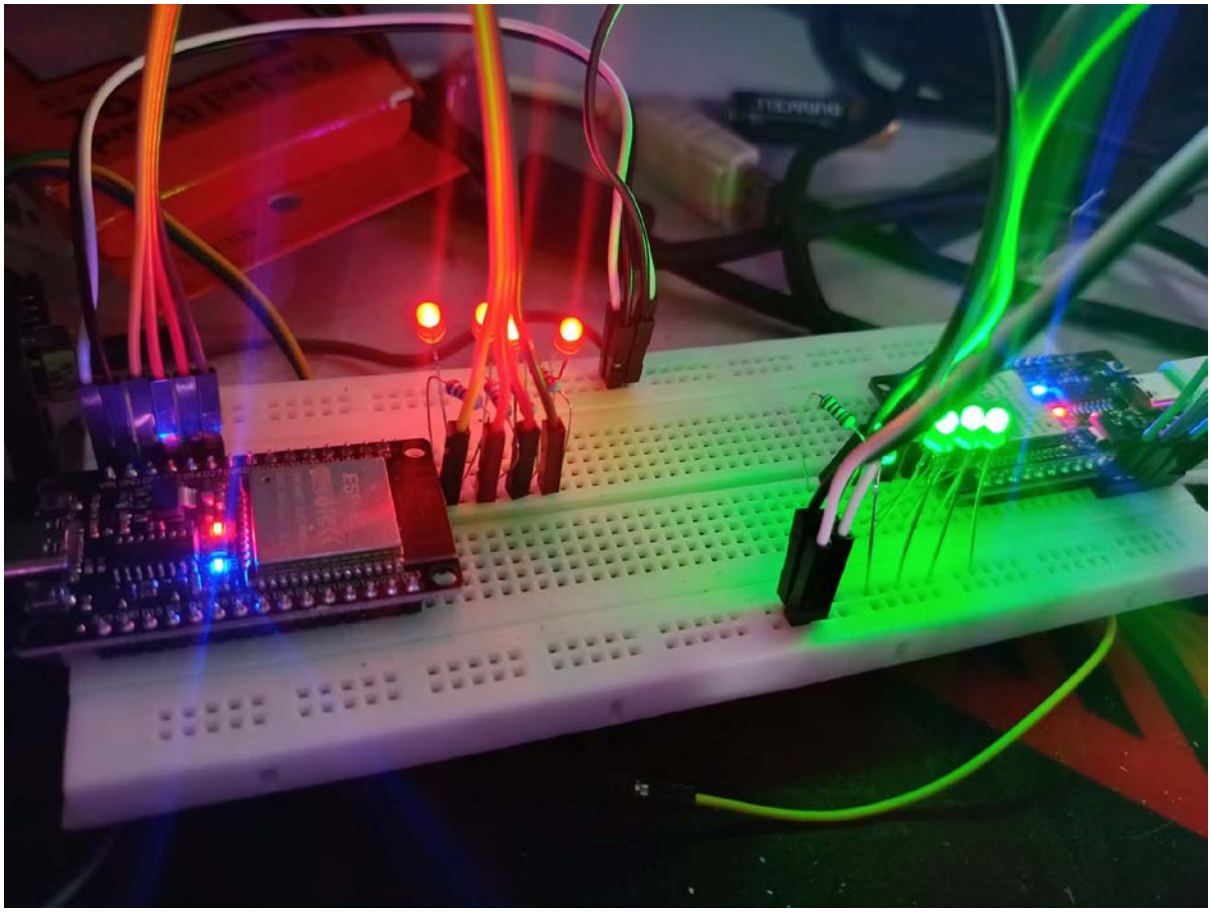


ภาพที่ 13 การแจ้งเตือน ใน Telegram Bot API PC



ภาพที่ 13 การแจ้งเตือน Telegram Bot API Mobile

- **ความปลอดภัย:** มีการใช้ระบบ Authentication ในหน้า Dashboard เพื่อป้องกันการเข้าถึงจากบุคคลภายนอก
- **การแสดงผล:** เมื่อ ESP32 เชื่อมต่อ WiFi ก็จะแสดงผมตาม Database ที่เก็บไว้



ภาพที่ 14 ESP32(โครงงาน) กำลังทำงาน

เอกสารอ้างอิง (References)

[1] **Thanapat Lertpitaksit.** (2025). *Universal Multi-Board IoT Control & Automation System*[Online]. Available: https://github.com/VeryHandsome520/MiniProject.git.

[2] **Espressif Systems.** (2023). *ESP32 Series Datasheet v4.1*. [Online]. Available: https://www.espressif.com. (ข้อมูลทางเทคนิคของไมโครคอนโทรลเลอร์ ESP32)

[3] **Richard Barry.** (2022). *Mastering the FreeRTOS Real Time Kernel: A Hands-On Tutorial Guide*. Real Time Engineers Ltd. (ทฤษฎีการจัดการ Multitasking และระบบปฏิบัติการแบบเรียลไทม์)

[4] **Fielding, R. T.** (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD dissertation, University of California, Irvine. (ที่มาของสถาปัตยกรรม RESTful API)

[5] **Ecma International.** (2017). *The JSON Data Interchange Syntax (ECMA-404 2nd Edition)*. (มาตรฐานรูปแบบการรับส่งข้อมูล JSON)

[6] **Luke Welling and Laura Thomson.** (2016). *PHP and MySQL Web Development (5th Edition)*. Addison-Wesley Professional. (แนวทางการพัฒนา Backend และการจัดการฐานข้อมูลสำหรับระบบควบคุม)

[7] **Mark Otto and Jacob Thornton.** (2023). *Bootstrap Documentation: Frontend framework for fast, responsive development*. [Online]. Available: https://getbootstrap.com. (การออกแบบ UI/UX แบบ Responsive สำหรับ Dashboard)

[8] **Telegram Messenger Inc.** (2024). *Telegram Bot API Documentation*. [Online]. Available: https://core.telegram.org/bots/api. (การเชื่อมต่อระบบแจ้งเตือนผ่าน Telegram)

[9] **V. K. Rao et al.** (2024). "Smart Energy Meter by Using IOT," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 12, no. V, pp. 620-623.