

ระบบตรวจสอบสถานะและสั่งการแสงสว่างในอาคารโดยใช้ ESP32

Building Lighting Control and Monitoring System using ESP32

โยธกา ตั้งตระกูล¹ ฐานิต แสงจินดา¹ วรวิธน์นัทธ์ ภักดีรัตน์¹ และ อติราช สุขสวัสดิ์^{1*}

Yotaka Tungtragul¹ Thanit Saengjinda¹ Warithnun Phakdeerat¹ and Atirarj Suksawad^{1*}

¹สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น

¹Department of Computer Engineering, Faculty of Engineering, Rajamangala University of Technology Isan, Khon Kaen, Campus, Thailand,

*Corresponding Author E-mail: atirarj.su@rmuti.ac.th

Received: 16/02/25, Revised: 11/03/25, Accepted: 12/03/25

บทคัดย่อ

บทความนี้นำเสนอระบบตรวจสอบและควบคุมแสงสว่างในอาคารโดยใช้ ESP32 เพื่อเพิ่มประสิทธิภาพการบริหารพลังงาน ระบบใช้ MOC3021 และ BT136 เป็นสวิตช์ควบคุม พร้อมวงจรป้องกันแรงดันไฟฟ้ากระชาก (Snubber Circuit) สำหรับการควบคุม ESP32 เชื่อมต่อกับเซิร์ฟเวอร์และฐานข้อมูล รองรับการสั่งงานผ่านเว็บไซต์และสวิตช์ไฟปกติ เว็บไซต์สามารถตรวจสอบสถานะ สั่งเปิด-ปิดไฟ ตั้งเวลาการทำงาน และสร้างรายงานพลังงานในรูปแบบ PDF และ Excel ผลการทดลองยืนยันว่าระบบทำงานได้อย่างมีประสิทธิภาพและเหมาะสำหรับการใช้งานในอาคารอัจฉริยะ

คำสำคัญ: ESP32 ระบบควบคุมแสงสว่าง อาคารอัจฉริยะ

Abstract

This paper presents a lighting monitoring and control system for buildings using ESP32 to enhance energy management efficiency. The system utilizes MOC3021 and BT136 as switching components, along with a snubber circuit to protect against voltage surges. The ESP32 microcontroller communicates with a server and database, enabling control via both a web-based interface and a physical switch. The web interface allows users to monitor system status, toggle lights, schedule operations, and generate energy usage reports in PDF and Excel formats. Experimental results confirm that the system operates efficiently and is well-suited for smart building applications.

Keywords: ESP32, Lighting Control System, Smart Building

1. บทนำ

ปัจจุบันเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง (Internet of Things : IoT) ได้เข้ามามีบทบาทสำคัญในการพัฒนาระบบอัตโนมัติและการควบคุมจากระยะไกล โดยเชื่อมต่ออุปกรณ์ต่าง ๆ ผ่านเครือข่ายอินเทอร์เน็ตเพื่อให้สามารถส่งข้อมูล ตรวจสอบสถานะและสั่งการ

อุปกรณ์ได้แบบเรียลไทม์จึงทำให้เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งถูกนำมาใช้ในหลากหลายด้าน เช่น บ้านอัจฉริยะ (Smart Home) [1], อาคารอัจฉริยะ (Smart Building) [2], ระบบเกษตรอัจฉริยะ (Smart Agriculture) [3], ระบบควบคุมอุตสาหกรรม (Industrial IoT) [4] และระบบดูแลสุขภาพอัจฉริยะ (Smart Healthcare) [5] ซึ่งช่วยเพิ่มประสิทธิภาพในการทำงาน ลดการใช้พลังงานและเพิ่มความสะดวกสบายให้กับผู้ใช้งาน

แนวคิดของเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งได้ถูกนำไปประยุกต์ใช้ในอาคารอัจฉริยะหลายระบบที่เกี่ยวข้องกับการควบคุมอุปกรณ์ไฟฟ้าและการบริหารจัดการพลังงาน ตัวอย่างเช่น ระบบตรวจสอบและควบคุมภายในบ้านอัจฉริยะ [1] ระบบนี้มุ่งเน้นไปที่การควบคุมและตรวจสอบอุปกรณ์ภายในบ้านผ่านแพลตฟอร์มออนไลน์ เช่น สมาร์ทโฟนหรือคอมพิวเตอร์ ผู้ใช้สามารถเปิด-ปิดไฟ ควบคุมเครื่องใช้ไฟฟ้าและตรวจสอบสถานะอุปกรณ์ได้จากทุกที่ผ่านเครือข่ายอินเทอร์เน็ตซึ่งช่วยเพิ่มความสะดวกสบายและลดการใช้พลังงานที่ไม่จำเป็น ระบบควบคุมอุณหภูมิของตู้ปลา Nano ผ่านเว็บแอปพลิเคชัน [6] ระบบนี้ใช้เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งในการตรวจสอบและควบคุมอุณหภูมิของตู้ปลาผ่านเซ็นเซอร์อุณหภูมิที่เชื่อมต่อกับไมโครคอนโทรลเลอร์ซึ่งสามารถประมวลผลและควบคุมการทำงานของเครื่องทำความร้อน (Heater) ได้โดยอัตโนมัติผู้ใช้สามารถตรวจสอบและตั้งค่าอุณหภูมิผ่านเว็บแอปพลิเคชันซึ่งช่วยให้สามารถรักษาสภาวะแวดล้อมที่เหมาะสมสำหรับสิ่งมีชีวิตในตู้ปลาได้อย่างมีประสิทธิภาพ ระบบตรวจสอบและควบคุมอุปกรณ์ภายในบ้านโดยใช้ไมโครคอนโทรลเลอร์ รุ่น ESP32 [7] ระบบนี้นำ ESP32 มาใช้เป็นตัวกลางในการเชื่อมต่อและควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน เช่น หลอดไฟ พัดลม หรือเครื่องใช้ไฟฟ้าต่าง ๆ โดยสามารถสั่งงานผ่าน Wi-Fi และอินเทอร์เน็ต ระบบสามารถแสดงสถานะของอุปกรณ์ในรูปแบบเรียลไทม์ และสามารถตั้งค่าให้ทำงานอัตโนมัติตามเงื่อนไขที่กำหนดซึ่งช่วยเพิ่มความสะดวกและลดการใช้พลังงาน การติดตามและควบคุมสถานะมอเตอร์

ป้อนผ่าน NETPIE และ Node-red [8] โดยระบบสามารถควบคุมความถี่ของมอเตอร์ได้

จากแนวคิดของเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งที่ถูกนำไปประยุกต์ใช้ในระบบต่าง ๆ งานวิจัยนี้มุ่งเน้นไปที่การพัฒนาตรวจสอบและควบคุมแสงสว่างในอาคารโดยใช้ไมโครคอนโทรลเลอร์รุ่น ESP32 เพื่อเพิ่มประสิทธิภาพในการบริหารจัดการพลังงานและเพิ่มความสะดวกในการควบคุม โดยมีการพัฒนาระบบควบคุมผ่านเว็บไซต์ อีกทั้งรองรับการควบคุมแบบ Manual ผ่านสวิตช์ไฟปกติ ทำให้สามารถใช้งานหากขาดการเชื่อมต่อกับระบบอินเทอร์เน็ตรวมถึงช่วยให้ผู้ใช้สามารถควบคุมแสงสว่างภายในอาคารผ่านเครือข่ายอินเทอร์เน็ตได้สะดวกมากขึ้น อีกทั้งยังสามารถตั้งเวลาเปิด-ปิดหลอดไฟเพื่อลดการใช้พลังงานที่ไม่จำเป็นและสามารถสร้างรายงานข้อมูลการใช้พลังงานเพื่อนำข้อมูลมาช่วยการวางแผนและบริหารพลังงานในอาคารได้อย่างมีประสิทธิภาพ

2. การออกแบบระบบ

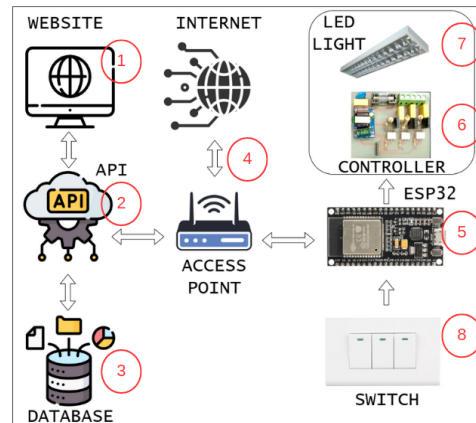
ระบบที่พัฒนาขึ้นได้รับการออกแบบให้ทำงานเป็นส่วนเสริมของสวิตช์ไฟเพื่อเพิ่มความสามารถในการควบคุมการเปิด-ปิดไฟผ่านสวิตช์ไฟแบบเดิมและผ่านการควบคุมผ่านเว็บไซต์ โดยโครงสร้างของระบบแสดงดังรูปที่ 1 มีองค์ประกอบหลัก ดังนี้ 1) เว็บเบราว์เซอร์: ผู้ใช้สามารถเข้าถึงระบบผ่านเว็บเบราว์เซอร์ ทำหน้าที่ เป็นส่วนติดต่อผู้ใช้หลัก 2) API (Node.js): เว็บเบราว์เซอร์สื่อสารกับ API ที่พัฒนาด้วย Node.js ทำหน้าที่ เป็นตัวกลางในการจัดการคำขอและการตอบสนอง 3) เซิร์ฟเวอร์และฐานข้อมูล: API เชื่อมต่อกับเซิร์ฟเวอร์และฐานข้อมูลเพื่อจัดเก็บและเรียกใช้ข้อมูลที่ต้องการสำหรับการทำงานของระบบ 4) Access Point: API ส่งข้อมูลไปยัง Access Point ทำหน้าที่ เป็นจุดเชื่อมต่อไร้สายเพื่อเชื่อมโยงระบบควบคุมกับอุปกรณ์ต่าง ๆ 5) ESP32: Access Point เชื่อมต่อกับไมโครคอนโทรลเลอร์ ESP32 ที่เป็นหน่วยประมวลผลหลักของระบบ 6) ชุดควบคุม: ESP32 ควบคุมการทำงานของชุดควบคุมหลอดไฟ ทำหน้าที่ แทนสวิตช์ไฟเดิม 7) หลอดไฟ: ชุดควบคุมเชื่อมต่อกับหลอดไฟโดยตรงทำให้สามารถดำเนินการเปิด-ปิดได้ผ่านระบบ 8) สวิตช์ไฟ: ESP32 สามารถรับข้อมูลจากสวิตช์ไฟเดิมเพื่อรองรับการสั่งงานเปิด-ปิดในลักษณะดั้งเดิม

การออกแบบระบบดังกล่าวช่วยให้สามารถควบคุมแสงสว่างได้อย่างยืดหยุ่นทั้งการใช้งานแบบดั้งเดิมและผ่านระบบดิจิทัลซึ่งจะช่วยเพิ่มประสิทธิภาพในการจัดการพลังงานและรองรับการทำงานในสภาพแวดล้อมที่หลากหลาย

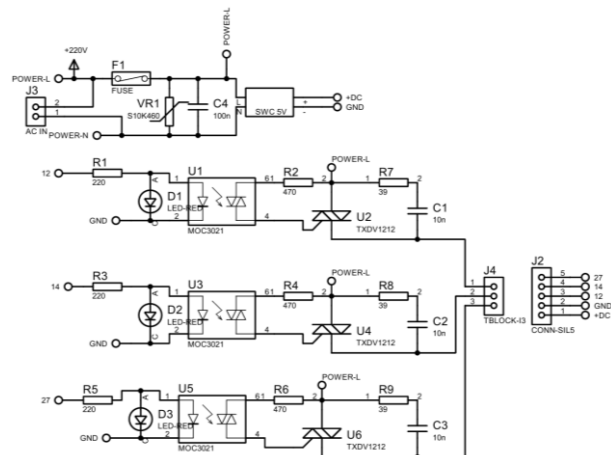
2.1 การออกแบบชุดควบคุมหลอดไฟ

การออกแบบวงจรชุดควบคุม ดังรูปที่ 2 ประกอบไปด้วย วงจรภาคแปลงไฟ ทำหน้าที่ แปลงไฟฟ้ากระแสสลับเป็นกระแสตรง โดยวงจรมีวารีสเตอร์ 10D471k และคาปาซิเตอร์ MKP 0.1μF เพื่อเป็นวงจรกันไฟกระชากก่อนเข้าสวิตช์ซึ่งเป็นแหล่งจ่ายไฟให้กับวงจรควบคุมและ

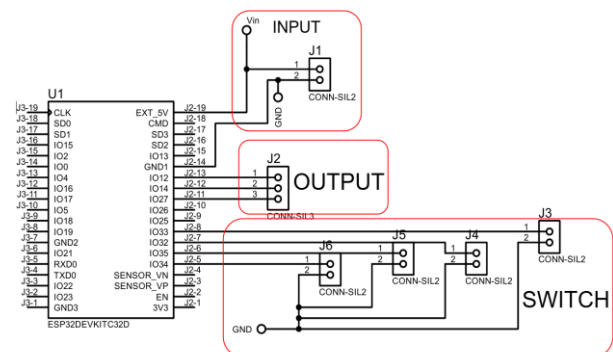
ไมโครคอนโทรลเลอร์ ถัดมาเป็นวงจรควบคุมการทำงานหลอดไฟ ประกอบไปด้วย ออปโตคัปเปิลเลอร์ MOC3021 และไดรแอค BT136 ที่ทำงานในลักษณะของโซลิดสเตตรีเลย์ (Solid-state relay : SSR) โดยมีวงจรป้องกันไฟกระชากแบบ RC Snubber ซึ่งทำหน้าที่หลักในการป้องกันและลดผลกระทบจากแรงดันไฟฟ้าหรือกระแสไฟฟ้าสูงชั่วขณะ (Transient) ที่เกิดขึ้น ใน วงจรในแต่ละเอาต์พุต วงจรนี้จะถูกสั่งการจากไมโครคอนโทรลเลอร์ รุ่น ESP 32 (ขา IO12, IO14, IO27) เพื่อควบคุมการทำงานของหลอดไฟ



รูปที่ 1 โครงสร้างของระบบ



รูปที่ 2 วงจรภาคจ่ายไฟและควบคุมหลอดไฟ

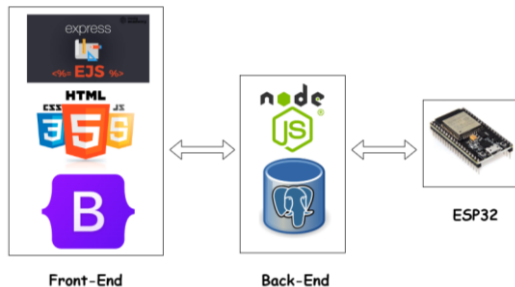


รูปที่ 3 วงจร ESP32 และสวิตช์

วงจรไมโครคอนโทรลเลอร์ ดังรูปที่ 3 ประกอบไปด้วยการออกแบบ 3 ส่วน ได้แก่ ส่วนที่ 1 เป็นอินพุต (input) รับจากภาคจ่ายไฟ ส่วนที่ 2 เป็นเอาต์พุต (output) ที่เชื่อมต่อไปยังวงจรควบคุมการทำงานหลอดไฟ ส่วนที่ 3 ส่วนรับค่าจากสวิตช์ไฟเดิม ที่มีการเชื่อมต่อกับไมโครคอนโทรลเลอร์รุ่น ESP32 (ขา IO32 IO33 IO35) เพื่อควบคุมการทำงานของหลอดไฟในรูปแบบ Manual ที่สามารถใช้สวิตช์ไฟเดิมหรือติดตั้งใหม่และมีสวิตช์รีเซ็ต (IO34) เพื่อใช้ในการตั้งค่าการทำงานของบอร์ดไมโครคอนโทรลเลอร์

2.2 การพัฒนาด้านซอฟต์แวร์

การพัฒนาด้านซอฟต์แวร์มีโครงสร้างของระบบแสดงดังรูปที่ 4 ประกอบด้วย 3 ส่วนหลัก ได้แก่ ส่วนแสดงผล (Front-End) ส่วนประมวลผลเบื้องหลัง (Back-End) และส่วนควบคุมอุปกรณ์ (Microcontroller) โดยส่วน Front-End เป็นส่วนของหน้าเว็บไซต์ที่ผู้ใช้งานมองเห็นและได้ตอบ โดยใช้เทคโนโลยี HTML, CSS, JavaScript, Bootstrap และ EJS ในการออกแบบและพัฒนาให้เชื่อมโยงกับผู้ใช้งานพร้อมทั้งใช้เฟรมเวิร์กที่ชื่อ Express.js สำหรับการจัดการการแสดงผล ส่วน Back-End ใช้เซิร์ฟเวอร์เป็น Node.js และใช้ PostgreSQL เป็นระบบจัดการฐานข้อมูลหลักในการเก็บและบริหารข้อมูลต่าง ๆ สุดท้ายเป็นส่วนควบคุมอุปกรณ์ซึ่งเลือกใช้ไมโครคอนโทรลเลอร์รุ่น ESP32 สำหรับเชื่อมต่อกับระบบ Back-End ผ่านเครือข่าย โดยทำหน้าที่รับและส่งข้อมูลเพื่อควบคุมการทำงานของระบบ เช่น การสั่งเปิด-ปิดหลอดไฟ



รูปที่ 4 โครงสร้างเว็บไซต์

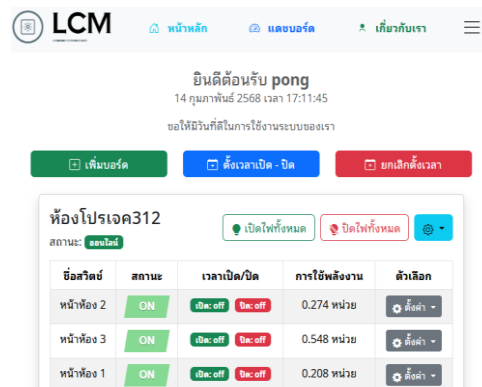
3. ผลการดำเนินงาน

ระบบตรวจสอบสถานะและสั่งการแสงสว่างในอาคารโดยใช้ ESP32 สามารถควบคุมหลอดไฟได้ทั้งผ่านการกดสวิตช์และผ่านการสั่งงานผ่านเว็บไซต์ โดยหน้าเว็บไซต์ของระบบในรูปที่ 5 นั้นสามารถแสดงสถานะการทำงานของหลอดไฟที่ทำการติดตั้ง และสามารถตั้งเวลาเปิดและเวลาปิด เพื่อใช้คำนวณพลังงานได้ อีกทั้งยังสามารถเพิ่มจำนวนบอร์ดควบคุมโดยการกรอกโทเค็นที่ได้จากบอร์ดควบคุม รูปที่ 6-7 แสดงผลการรายงานผ่านกราฟที่สามารถเลือกช่วงวันที่ต้องการได้

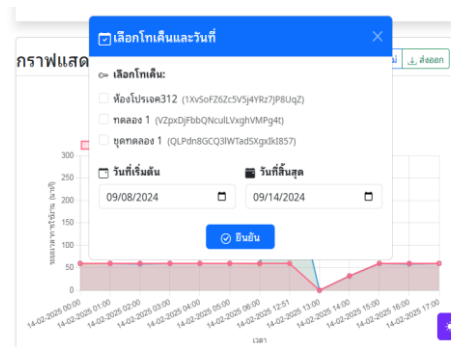
ในส่วนของการพัฒนาเว็บไซต์ของระบบสามารถส่งออกรายงานการใช้ไฟฟ้าในรูปแบบ ไฟล์ PDF ซึ่งจะสรุปการใช้งานของหลอดไฟของแต่ละช่องทุก 60 นาที เพื่อที่จะนำมาคำนวณหน่วยการใช้ไฟฟ้า พร้อมทั้งแสดงวันเวลารายละเอียดรายงานการใช้ไฟฟ้า ดังรูปที่ 8

ในการทดสอบระบบได้มีการจำลองให้คอมพิวเตอร์ขนาดเล็ก โดยเลือกใช้ Raspberry Pi 4 ทำหน้าที่ เป็นเซิร์ฟเวอร์จำลองเพื่อทดสอบการเรียกใช้งาน API จากไมโครคอนโทรลเลอร์ โดยมีวัตถุประสงค์เพื่อประเมินความสามารถในการรองรับการทำงานในกรณีที่มียุโรปและผู้ใช้งานเพิ่มขึ้น (ระบบควบคุมหลอดไฟ) สำหรับอุปกรณ์ Raspberry Pi 4 ที่ใช้ในการทดสอบ มีสเปคดังนี้

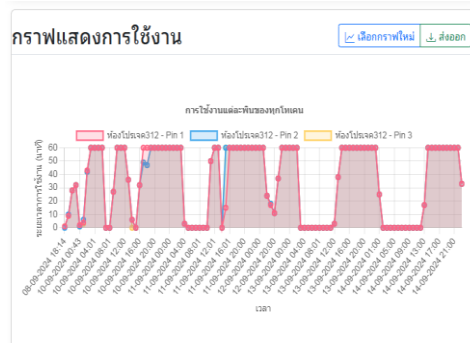
- CPU: Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
- RAM: 4GB LPDDR4-3200 SDRAMWW
- ROM: SD CARD 64GB



รูปที่ 5 หน้าเว็บไซต์ของระบบ



รูปที่ 6 การเลือกวันเวลาที่ต้องการแสดงผล



รูปที่ 7 กราฟแสดงช่วงเวลาการใช้หลอดไฟในช่วง 7 วัน

ในการทดสอบโหลด API ได้สร้างหน้าเว็บเพจขึ้นมาเพื่อใช้ในการทดสอบส่งค่าขอเป็นจำนวน จาก 1 ถึง 1000 ครั้ง โดยใช้การทดสอบ 3 รูปแบบแสดงดังตารางที่ 1

LCM

รายงานการใช้ไฟฟ้า
ชื่อองค์กร: หอประชุม312

ลำดับ	หมายเลขห้อง	ชื่อบอร์ด	วันเวลา	ระยะเวลาใช้งาน (นาที)	หน่วยการใช้ไฟฟ้า (kWh)
1	2	หอประชุม312	08/09/2024 18:14:00	0.00	0.000
2	3	หอประชุม312	08/09/2024 18:14:00	1.00	0.001
3	1	หอประชุม312	08/09/2024 18:14:00	1.00	0.000
4	2	หอประชุม312	08/10/2024 00:00:00	60.00	0.018
5	1	หอประชุม312	08/10/2024 00:00:00	60.00	0.018
6	3	หอประชุม312	08/10/2024	60.00	0.036

รูปที่ 8 การแสดงผลรายงานการใช้ไฟฟ้าในรูปแบบไฟล์ PDF

ตารางที่ 1 การเปรียบเทียบการทดสอบจำนวนการส่ง Request

Program	Test 1	Test 2	Test 3
Number of Requests	1	100	1000
Total time	61.90 ms	1,948.60 ms	17,069.10 ms
Average time per request	61.90 ms	19.49 ms	17.07 ms
CPU Speed	700-800MHz	700-1,000MHz	900-1,800MHz
CPU On Load	99.10%	99.10%	99.10%
CPU No Load	0.90%	0.90%	0.90%
Memory	532.84 MB	548.54 MB	548.26 MB

จากตารางที่ 1 พบว่า เวลาเฉลี่ยต่อคำขอ (Request) มีแนวโน้มลดลงเมื่อจำนวนคำขอเพิ่มขึ้นขณะที่ความเร็วของ CPU ปรับตัวสูงขึ้นตามจำนวนคำขอที่ส่งเข้ามา นอกจากนี้ พบว่า CPU Load ขณะทำงานคงที่อยู่ที่ 99.10% ข้อมูลในตารางได้มาจากการทดสอบผ่านหน้าเว็บเพจที่พัฒนาขึ้นสำหรับตรวจสอบประสิทธิภาพของ API เมื่อ Raspberry Pi ส่งคำตอบ (Response) สำเร็จ จากนั้นระบบจะบันทึกค่าและแสดงผลผ่านหน้าเว็บดังกล่าวโดยแสดงค่าการใช้งาน CPU และ RAM รวมถึงระยะเวลาในการประมวลผลตามจำนวนคำขอที่กำหนด จากการทดสอบพบว่าการใช้หน่วยความจำเพิ่มขึ้นเพียงเล็กน้อย แม้ว่าจำนวนคำขอจะเพิ่มขึ้นอย่างมาก ซึ่งแสดงให้เห็นว่าระบบจะต้องรองรับคำขอจาก API จำนวนมาก แต่เซิร์ฟเวอร์ยังคงสามารถทำงานได้อย่างมีประสิทธิภาพ

4. สรุป

ระบบตรวจสอบสถานะและควบคุมการทำงานของระบบแสงสว่างภายในอาคารโดยใช้ ESP32 ได้รับการพัฒนาเพื่อเพิ่มประสิทธิภาพในการบริหารจัดการพลังงานไฟฟ้า โดยระบบประกอบด้วยวงจรควบคุมหลอดไฟที่ใช้ออปโตคัปเปลอร์ (MOC3021) และไดรเอก (BT136) เป็นสวิตช์ควบคุม พร้อมติดตั้งวงจรป้องกันแรงดันไฟฟ้ากระชาก (Snubber Circuit) เพื่อป้องกันความเสียหายที่อาจเกิดขึ้นกับโหลดที่เชื่อมต่อ การควบคุมระบบดำเนินการผ่านไมโครคอนโทรลเลอร์ ESP32 ซึ่งสามารถสื่อสารกับเซิร์ฟเวอร์และฐานข้อมูล อีกทั้งยังรองรับการควบคุมแบบดั้งเดิมผ่านสวิตช์ไฟ สำหรับระบบเว็บไซต์ ได้รับการออกแบบให้

สามารถตรวจสอบสถานะหลอดไฟ สั่งการเปิด-ปิด และกำหนดตารางการทำงานโดยเลือกวันและเวลาได้ตามต้องการ นอกจากนี้ ระบบยังสามารถสร้างรายงานการใช้พลังงานในรูปแบบ PDF และ Excel เพื่อใช้ในการวิเคราะห์และติดตามผล ผลการทดสอบแสดงให้เห็นว่าการสั่งงานผ่านเว็บไซต์และการตั้งเวลาทำงานสามารถดำเนินการได้อย่างมีประสิทธิภาพ และระบบยังสามารถรองรับการควบคุมด้วยสวิตช์ไฟแบบปกติได้อย่างสมบูรณ์

เอกสารอ้างอิง

- [1] V. Pravalika and Ch. Rajendra Prasad, "Internet of things based home monitoring and device control using Esp32," *International Journal of Recent Technology and Engineering*, vol. 8, pp. 58-62, 2019.
- [2] A. Verma, S. Prakash, V. Srivastava, A. Kumar and S. Chandra Mukhopadhyay, "Sensing, controlling, and IoT infrastructure in smart building: A review," *IEEE Sensors Journal*, pp. 9036-9046, 2019.
- [3] K. A. Patil, and N. R. Kale. "A model for smart agriculture using IoT," *International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*, pp. 543-545, 2016.
- [4] E. Sisinni, A. Saifullah, S. Han, U. Jennehag and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE transactions on industrial informatics*, pp. 4724-4734, 2018.
- [5] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of things for smart healthcare: Technologies, challenges, and opportunities," *Ieee Access* 5 , pp. 26521-26544, 2017.
- [6] Y. Tungtragul, J. Wongkham, S. Matarach, A. Suksawad, K. Puntsri, and P. Tangkittipon, (2024, May), "Development of an IoT Web-Based Water Temperature Control System for Nano Fish Tanks," In *2024 21st International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 1-5, 2024.
- [7] V. Pravalika and C. R. Prasad, "Internet of things based home monitoring and device control using Esp32," *International Journal of Recent Technology and Engineering*, vol. 8, pp. 58-62, 2019.
- [8] ณรงค์ชัย ทศพร, ณัฐวิทย์ มหาสังข์, เพ็ญญา เจริญดีและชัยวุฒ ชูรักษ์, "ระบบควบคุมและแสดงค่าพลังงานของมอเตอร์ปั้มน้ำด้วย IOT," *วิศวกรรมไฟฟ้าไทย*, เล่มที่ 3 ฉบับที่ 1, หน้า 8-11, 2023.