IoT-Based Smart Detection and Control System for Warehouse Management

ในปัจจุบันคลังสินค้าถือเป็นส่วนสำคัญของห่วงโซ่อุปทาน (Supply Chain) ซึ่งต้องการการควบคุมด้านความปลอดภัยและคุณภาพสภาพแวดล้อมอย่างมีประสิทธิภาพ ปัญหาที่มักพบ ได้แก่ อุณหภูมิสูงผิดปกติ ความชื้นเกินระดับ ไฟไหม้ การบุกรุก หรือความล้มเหลวของระบบควบคุม การตรวจสอบแบบใช้แรงงานมนุษย์เพียงอย่างเดียวไม่สามารถตอบสนองต่อสถานการณ์ฉุกเฉินได้ทันเวลา

เทคโนโลยี IoT ได้เข้ามามีบทบาทสำคัญในการบริหารจัดการคลังสินค้า โดยสามารถเชื่อมโยงอุปกรณ์เซนเซอร์และอุปกรณ์ควบคุมให้ทำงานผ่านอินเทอร์เน็ต ส่งข้อมูลแบบ Real-time และตอบสนองต่อเหตุการณ์ได้อัตโนมัติ ช่วยให้ผู้จัดการคลังสามารถเข้าถึงข้อมูลจากทุกที่ ลดความเสี่ยงจากเหตุไม่คาดคิด และเพิ่มความปลอดภัยโดยรวม

ดังนั้น ระบบที่นำเสนอในงานนี้จึงมุ่งพัฒนาคลังสินค้าอัจฉริยะที่ผสานเซนเซอร์และอุปกรณ์ต่าง ๆ เพื่อให้สามารถตรวจจับและควบคุมสถานการณ์แบบ Real-time พร้อมความสามารถทำงานอัตโนมัติในสถานการณ์ที่มีความเสี่ยงสูง

วัตถุประสงค์ของโครงงาน (Objectives)

1. เพื่อควบคุมอุปกรณ์ภายในคลังสินค้าแบบ Real-time
2. เพื่อตรวจจับและแจ้งเตือนอัคคีภัยอย่างทันท่วงที
3. เพื่อป้องกันและตรวจจับการโจรกรรมหรือการบุกรุก
4. เพื่อพัฒนาต้นแบบระบบคลังสินค้าอัจฉริยะที่สามารถนำไปต่อยอดได้ในอนาคต

II. LITERATURE SURVEY

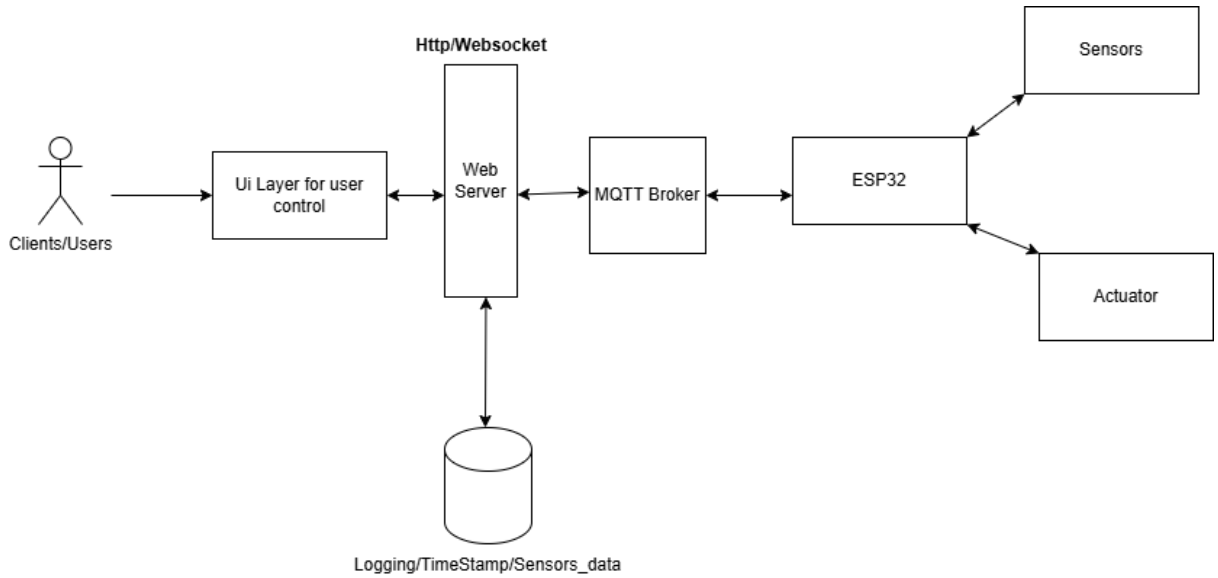
การประยุกต์ใช้เทคโนโลยี Internet of Things (IoT) ในงานด้านคลังสินค้าอัจฉริยะได้รับการศึกษาอย่างแพร่หลายในช่วงไม่กี่ปีที่ผ่านมา โดยมีงานวิจัยจำนวนหนึ่งที่มุ่งเน้นด้านระบบตรวจจับและควบคุมแบบ Real-time ซึ่งสอดคล้องกับระบบที่พัฒนาในงานวิจัยฉบับนี้

งานของ Prabha et al. [1] ได้ทำการ *systematic review* ที่ครอบคลุมการพัฒนาเครือข่ายเซนเซอร์อัจฉริยะในคลังสินค้า โดยเน้นสถาปัตยกรรมของระบบ IoT, ประเภทของเซนเซอร์, รูปแบบการสื่อสาร รวมถึงการตรวจจับสภาวะผิดปกติ เช่น อุณหภูมิสูงเกิน การเกิดเปลวไฟ และการเคลื่อนไหวที่ไม่ได้รับอนุญาต งานนี้ชี้ให้เห็นว่า IoT มีบทบาทสำคัญต่อระบบคลังสินค้าอัจฉริยะ ทั้งในด้านความปลอดภัยและการลดความเสี่ยง ซึ่งสอดคล้องโดยตรงกับระบบตรวจจับอัคคีภัยและป้องกันการบุกรุกที่ถูกพัฒนาในโครงงานนี้

นอกจากนี้ Jarašūnienė et al. [2] ได้ทำการทบทวนเชิงลึกเกี่ยวกับผลกระทบของ IoT ต่อการจัดการคลังสินค้า โดยระบุว่า IoT สามารถเพิ่มความสามารถในการตรวจสอบแบบ Real-time ช่วยให้ผู้ดูแลคลังสามารถควบคุมอุปกรณ์จากระยะไกล และลดความผิดพลาดจากแรงงานมนุษย์ งานวิจัยยังกล่าวถึงการใช้เซนเซอร์ตรวจวัดสภาพแวดล้อมและเซนเซอร์ตรวจจับการบุกรุกซึ่งมีความสอดคล้องกับระบบที่นำเสนอในงานนี้อย่างชัดเจน

จากการสำรวจแบบรวมกลุ่มของงานวิจัยทั้งสองฉบับ พบว่าองค์ประกอบสำคัญของคลังสินค้าอัจฉริยะ ได้แก่ ระบบตรวจจับอุณหภูมิ, ระบบตรวจจับอัคคีภัย, การตรวจจับการเคลื่อนไหว และการควบคุมอุปกรณ์แบบ Real-time ซึ่งตรงกับวัตถุประสงค์ของระบบ IoT-Based Smart Detection and Control System ที่พัฒนาในงานนี้อย่างครบถ้วน

กระบวนการพัฒนาระบบ



1. การวิเคราะห์ความต้องการ (Requirement Analysis)

ทำการรวบรวมและวิเคราะห์ความต้องการของระบบจากโจทย์คลังอัจฉริยะ ซึ่งประกอบด้วยความต้องการหลัก ได้แก่

1. ตรวจวัดข้อมูลสภาพแวดล้อมแบบเรียลไทม์ เช่น อุณหภูมิ ความชื้น เปลวไฟ การเคลื่อนไหว
2. ควบคุมอุปกรณ์ตอบสนอง เช่น ประตู พัดลม หรือไฟ
3. เฝ้าระวังการบุกรุกและเหตุฉุกเฉิน
4. บันทึกข้อมูลลงฐานข้อมูล
5. แสดงผลผ่านเว็บและสามารถควบคุมอุปกรณ์ได้ทันที

ผลลัพธ์คือการกำหนดองค์ประกอบของระบบเซนเซอร์, อุปกรณ์ควบคุม, โปรโตคอลสื่อสาร, โครงสร้างเครือข่าย และความต้องการของผู้ใช้

2. การออกแบบสถาปัตยกรรมระบบ (System Architecture Design)

ระบบถูกออกแบบตามหลัก IoT architecture 4 ชั้น ได้แก่ Device Layer, Network Layer, Application Layer และ Storage Layer โดยมีองค์ประกอบดังนี้

2.1 ส่วนอุปกรณ์ (Device Layer)

ประกอบด้วย **ESP32** ทำหน้าที่เป็นไมโครคอนโทรลเลอร์หลักในการอ่านค่าจากเซนเซอร์ เช่น

- เซนเซอร์ตรวจจับอุณหภูมิ/ความชื้น
- เซนเซอร์ตรวจจับเปลวไฟ
- เซนเซอร์ตรวจจับการเคลื่อนไหว
- รวมถึงควบคุม Actuator เช่น Servo Motor หรือพัดลม

ESP32 จะอ่านค่าจากเซนเซอร์และส่งข้อมูลไปยังระบบผ่านโปรโตคอล MQTT

2.2 ส่วนสื่อสารข้อมูล (Network Layer)

ใช้ **MQTT Broker** เป็นตัวกลางในการรับ-ส่งข้อมูลระหว่างอุปกรณ์กับ Web Server

- ESP32 to Publish ค่าจากเซนเซอร์

- Web Server to Subscribe เพื่อรับข้อมูลแบบทันที
- Web Server to Publish คำสั่งควบคุม เช่น เปิดพัดลม / เปิดประตู
- ESP32 to Subscribe คำสั่งและสั่งงาน actuator

MQTT ถูกเลือกเนื่องจากมีน้ำหนักเบา ใช้แบนด์วิธต่ำ เหมาะสำหรับอุปกรณ์ IoT

2.3 ส่วนประมวลผลและให้บริการ (Application Layer)

ใช้ **Web Server (พัฒนาโดยภาษา Go)** ทำหน้าที่เป็นตัวกลางระหว่างผู้ใช้และระบบ IoT โดยมีหน้าที่หลักดังนี้

- ติดต่อกับ MQTT Broker เพื่อรับข้อมูลจาก ESP32
- ประมวลผลเหตุการณ์ เช่น อุณหภูมิสูง ⟶ แจ้งเตือน
- เปิดบริการ API เพื่อให้ UI Layer เรียกใช้งาน
- เปิด WebSocket สำหรับส่งข้อมูลแบบ Real-time ไปยังผู้ใช้
- จัดการสิทธิ์ผู้ใช้งานและการตรวจสอบคำขอ

ระบบถูกออกแบบให้มีประสิทธิภาพสูง รองรับการเชื่อมต่อพร้อมกันหลายผู้ใช้

2.4 ส่วนฐานข้อมูล (Storage Layer)

มีการจัดเก็บข้อมูลสำคัญดังนี้

- ข้อมูลผู้ใช้ (Users)
- ข้อมูลเหตุการณ์และการควบคุม (Event Logs)
- ข้อมูลวันที่เวลา (Timestamp)
- ค่าที่อ่านจากเซนเซอร์ทุกประเภท (Sensors Data)

ฐานข้อมูลถูกเชื่อมต่อโดยตรงกับ Web Server เพื่อให้ระบบมีจุดควบคุมการจัดเก็บข้อมูลที่ชัดเจนและปลอดภัย

2.5 ส่วนแสดงผลและควบคุม (UI Layer)

UI Layer ทำหน้าที่เป็น Dashboard แบบ Interactive สำหรับผู้ใช้งาน โดยมีองค์ประกอบดังนี้
UI สื่อสารกับ Web Server ผ่าน REST API และ WebSocket เพื่อให้สามารถตอบสนองข้อมูลแบบทันที

3. การพัฒนาและเชื่อมต่อระบบ (Implementation)

ประกอบด้วยขั้นตอนดังนี้

1. พัฒนาโปรแกรมบน ESP32 สำหรับอ่านค่าเซนเซอร์และควบคุม actuator
2. เชื่อมต่อ ESP32 กับ MQTT Broker และกำหนด Topic ต่าง ๆ
3. พัฒนา Web Server ด้วยภาษา Go พร้อมส่วนติดต่อ MQTT
4. สร้าง API สำหรับ UI ในการควบคุมอุปกรณ์
5. พัฒนา WebSocket channel เพื่อส่งข้อมูลแบบ Real-time
6. พัฒนา UI Dashboard สำหรับผู้ใช้
7. เชื่อมต่อฐานข้อมูลและทดสอบการบันทึกข้อมูล

4. การทดสอบระบบ (Testing)

ทดสอบตามกรณีใช้งานจริง เช่น

- ทดสอบค่าจากเซนเซอร์หลายประเภท
- ทดสอบการส่งข้อมูลผ่าน MQTT ทั้ง publish/subscribe
- ทดสอบการแสดงผลแบบ Real-time บน UI
- ทดสอบคำสั่งควบคุมอุปกรณ์ (Actuator Control)
- ทดสอบความเสถียรของ Web Server
- ทดสอบความถูกต้องของข้อมูลบนฐานข้อมูล

5. การประเมินผลระบบ (Evaluation)

ประเมินผลจาก

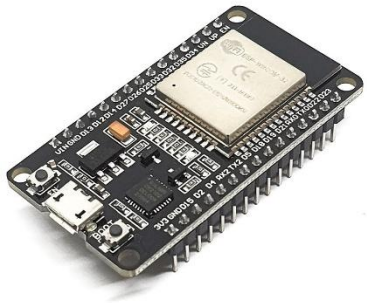
- ความถูกต้องของการตรวจวัด
- ความเร็วในการตอบสนองของระบบ Real-time
- ความเสถียรของระบบสื่อสาร MQTT
- ความสามารถในการควบคุมอุปกรณ์ผ่าน UI

- การบันทึกข้อมูลลงฐานข้อมูล

Hardware & Tools

อุปกรณ์ฮาร์ดแวร์ (Hardware Components)

1) บอร์ดไมโครคอนโทรลเลอร์ ESP32



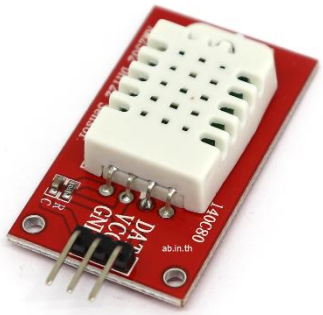
ESP32 เป็นไมโครคอนโทรลเลอร์ที่รองรับ Wi-Fi และ Bluetooth ในตัว ใช้สำหรับอ่านค่าจากเซนเซอร์ทุกชนิดและสื่อสารข้อมูลผ่านโปรโตคอล MQTT ไปยังเซิร์ฟเวอร์

หน้าที่ในระบบ:อ่านค่าเซนเซอร์อุณหภูมิ, ความชื้น, เปลวไฟ และตรวจจับการเคลื่อนไหว

ส่งข้อมูลไปยัง MQTT Broker

รับคำสั่งเพื่อควบคุมอุปกรณ์ Actuator เช่น Servo Motor หรือรีเลย์

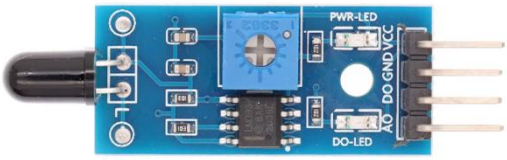
2) เซนเซอร์วัดอุณหภูมิและความชื้น DHT22



- ตรวจวัดอุณหภูมิและความชื้นแบบ Real-time

- ส่งข้อมูลให้ ESP32 เพื่อประเมินเหตุการณ์ เช่น ความร้อนสูง

3) เซนเซอร์ตรวจจับเปลวไฟ (IR Flame Detection Sensor)



เซนเซอร์อินฟราเรดสำหรับตรวจจับแสงอินฟราเรดจากเปลวไฟ

หน้าที่ในระบบ:

- ตรวจจับการเกิดไฟไหม้
- ส่งสัญญาณให้ ESP32 เพื่อแจ้งเตือนและสั่งงาน Actuator เช่น เปิดประตูอัตโนมัติ

4) เซนเซอร์ตรวจจับการเคลื่อนไหว PIR (Passive Infrared Sensor)

หน้าที่ในระบบ:

- ตรวจจับการบุกรุกในคลังสินค้า
- ส่งข้อมูลไปยัง ESP32 เพื่อแสดงสถานะบน Dashboard

5) มอเตอร์เซอร์โว (Servo Motor / MG995)

หน้าที่ในระบบ:

ใช้เป็นอุปกรณ์เปิด–ปิดประตูคลังสินค้าอัตโนมัติ

ทำงานตามคำสั่งจาก Web Server ผ่าน MQTT

6) โมดูลรีเลย์ (Relay Module)

อุปกรณ์สวิตช์ไฟฟ้า ใช้ควบคุมโหลดแรงดันสูง เช่น พัดลม หรือไฟส่องสว่าง

หน้าที่ในระบบ:

รับคำสั่งจาก ESP32 เพื่อเปิด–ปิดอุปกรณ์ไฟฟ้า

ใช้ร่วมกับพัดลมระบายอากาศภายในคลัง

7) พัดลมกระแสตรง DC 12V

เปิดอัตโนมัติเมื่ออุณหภูมิสูง

ช่วยรักษาสภาพแวดล้อมของคลังสินค้าให้เหมาะสม

8) หลอดไฟ LED

หน้าที่ในระบบ:

แสดงสถานะ เช่น ไฟแดง = อันตราย, ไฟเขียว = ปกติ

9) ตัวต้านทาน (Resistor)

หน้าที่ในระบบ:

- ปรับแรงดันให้เหมาะสม
- ป้องกันความเสียหายต่ออุปกรณ์อิเล็กทรอนิกส์

Software Tools

ซอฟต์แวร์และเครื่องมือ (Software Tools)

ภาษาพัฒนา Web Server: Go

โปรโตคอลสื่อสาร: MQTT

ระบบจัดการฐานข้อมูล: MySQL

บอร์ดโปรแกรม ESP32: Arduino IDE / PlatformIO

ระบบแสดงผล: Web Dashboard + WebSocket

สรุปผลการดำเนินงาน

โครงงานนี้ประสบความสำเร็จในการพัฒนา “ระบบตรวจจับและควบคุมอัจฉริยะสำหรับคลังสินค้า” โดยใช้เทคโนโลยี IoT ระบบสามารถ

- ตรวจวัดสภาวะแวดล้อมได้แบบ Real-time
- ตรวจจับไฟไหม้และการบุกรุก
- ควบคุมอุปกรณ์ตอบสนองอัตโนมัติ เช่น พัดลม ประตู หรือไฟ
- แสดงผลผ่าน Dashboard ได้ทันที
- จัดเก็บข้อมูลสำหรับอ้างอิงย้อนหลัง

ระบบสามารถทำงานได้อย่างเสถียร และตอบสนองเหตุการณ์ภายในเวลาต่ำกว่า 1 วินาทีในกรณีฉุกเฉิน

อ้างอิง

[1] S. R. Prabha, et al., “A systematic review of recent advances in IoT-based sensor networks for warehouse management,” Results in Engineering, vol. 29, 109195, 2026. doi: 10.1016/j.reng.2026.109195.

[2] A. Jarašūnienė, K. Čižiūnienė, and A. Čereška, “Research on Impact of IoT on Warehouse Management,” Sensors, vol. 23, no. 4, p. 2213, 2023. doi: 10.3390/s23042213