รายงาน Mini Project

Smart CoE Monitoring

จัดทำโดย

นายพชรพล เกตุแก้ว รหัสนักศึกษา 6610110190

นายภควัฒน์ พรหมหอม รหัสนักศึกษา 6610110227

รายงานนี้เป็นส่วนหนึ่งรายวิชา 240-371 IoT Developer Module

ภาคเรียนที่ 2 ปีการศึกษา 2568

มหาวิทยาลัยสงขลานครินทร์

คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของรายวิชา 240-371 IoT Developer Module ภาคเรียนที่ 2 ปีการศึกษา 2568 โดยมีวัตถุประสงค์เพื่อสรุปผลโครงงาน “Smart CoE Monitoring” ซึ่งเป็นระบบเฝ้าระวังสภาพแวดล้อมอัจฉริยะภายในอาคารภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ โครงงานมุ่งแก้ปัญหาการตรวจสอบแบบเดิมที่ล่าช้า โดยนำเทคโนโลยี IoT ร่วมกับสถาปัตยกรรมเครือข่ายแบบลำดับชั้น Edge Gateway และระบบคลาวด์ (AWS Serverless) เพื่อให้สามารถตรวจวัด แจ้งเตือน และแสดงผลข้อมูลได้แบบเรียลไทม์

เนื้อหาครอบคลุมการออกแบบระบบ การสื่อสารข้อมูล และผลการทดสอบ พร้อมทั้งอภิปรายปัญหาข้อจำกัด และแนวทางพัฒนาในอนาคต ทั้งนี้คณะผู้จัดทำหวังว่ารายงานฉบับนี้จะเป็นประโยชน์ต่อการพัฒนาและศึกษาด้าน IoT และ Cloud Computing ต่อไป หากมีข้อผิดพลาดประการใด ขออภัยมา ณ ที่นี้

คณะผู้จัดทำ

นายพชรพล เกตุแก้ว

นายภควัฒน์ พรหมหอม

สารบัญ

1 บทนำ 1

1.1 ความเป็นมาและความสำคัญ 1

1.2 ปัญหาที่ต้องการแก้ 2

2. วัตถุประสงค์ 4

3. การออกแบบระบบ 5

3.1 Architecture diagram 5

3.2 Hardware design 7

3.3 Communication design 9

4. การพัฒนา 11

4.1 การเชื่อมต่ออุปกรณ์ 11

4.2 Software/Cloud 11

5. การทดสอบระบบ 16

5.1 ผลการทดลอง 16

5.2 ความเสถียร 23

6. ผลลัพธ์และอภิปราย 24

7. ปัญหาและข้อจำกัด 25

8. แนวทางพัฒนาต่อ 26

9. บทบาทหน้าที่หลักของแต่ละคน 27

10. บทสรุป 30

สารบัญภาพ

รูป 1 Architecture Diagram 5
รูป 2 Detailed Architecture Diagram 5
รูป 3 Full Schematic of Multimodal Board 7
รูป 4 Schematic of Multimodal Board 7
รูป 5 Communication Design Diagram 9
รูป 6 MQTT Communication Design 10
รูป 7 ตัวอย่างโปรแกรมที่ทำหน้าประมวลผลและส่งต่อข้อมูลบน Raspberry Pi 12
รูป 8 การไหลของข้อมูลบน MQTT Server 12
รูป 9 รายละเอียด DynamoDB บน AWS 13
รูป 10 ฟังก์ชั่นบน AWS Lambda ที่มีการใช้งานในเว็บ 13
รูป 11 ตัวอย่างฟังก์ชั่นการทำงานของ ParseAndSaveIoTData 14
รูป 12 หน้าเว็บ Smart CoE Monitoring 14
รูป 13 API Gateway บน AWS ที่เว็บไซต์มีการเรียกใช้งาน 15
รูป 14 เว็บไซต์สำหรับการตั้งค่าระบบ 15
รูป 15 Multimodal Board ที่ห้อง R200 16
รูป 16 Multimodal Board ที่ห้อง R303 17
รูป 17 Multimodal Board ที่ห้อง R201 17
รูป 18 Multimodal Board ที่ห้อง AI Co-Working Space 18
รูป 19 Sensor Gateway ที่ห้อง CoE Co-Working Space 18
รูป 20 ESP-CAM ที่ห้อง AI Co-Working Space 19
รูป 21 Edge Gateway ทั้งสอบที่ห้อง PUPA 19
รูป 22 ข้อมูลบน DynamoDB ของตารางที่ชื่อว่า IoTData 20
รูป 23 Metrics การทำงานของ GetSensorData บน AWS Lambda 20
รูป 24 การแจ้งเตือนบน Discord 21
รูป 25 แดชบอร์ดการแสดงผลบนเว็บไซต์ 22
รูป 26 บันทึกการทำงาน 1 (พชรพล) 27
รูป 27 บันทึกการทำงาน 2 (พชรพล) 28
รูป 28 บันทึกการทำงาน 1 (ภควัฒน์) 29
รูป 29 บันทึกการทำงาน 2 (ภควัฒน์) 29

1 บทนำ

1.1 ความเป็นมาและความสำคัญ

อาคารภาควิชาวิศวกรรมคอมพิวเตอร์ (CoE) คือศูนย์รวมของห้องปฏิบัติการ ห้องเรียน และพื้นที่วิจัยที่เต็มไปด้วยอุปกรณ์มูลค่าสูงและงานวิจัยที่ต้องการสภาพแวดล้อมที่แม่นยำตลอด 24 ชั่วโมง แต่ในความเป็นจริง การดูแลพื้นที่เหล่านี้ยังคงพึ่งพาการเดินตรวจด้วยมือหรืออุปกรณ์วัดแบบแยกส่วน ที่ไม่สามารถมองเห็นภาพรวมได้ ไม่ว่าจะเป็นอุณหภูมิในห้องแม่ข่ายที่พุ่งสูงเกินขีดจำกัด หรือควันไฟที่เริ่มก่อตัวในห้องแล็บ ทุกอย่างล้วนถูกรับรู้ช้าเกินไปเสมอ

โครงการ Smart CoE Monitoring กำเนิดขึ้นเพื่อเปลี่ยนสมการนี้อย่างสิ้นเชิง ด้วยการนำพลังของ Internet of Things (IoT) มาหลอมรวมกับสถาปัตยกรรมโครงข่ายแบบ Multi-protocol และระบบคลาวด์ที่ออกแบบมาเพื่อความเสถียรและประหยัดสูงสุด ทำให้ทุกมุมของอาคารกลายเป็นข้อมูลเรียลไทม์ที่พร้อมตอบสนองทุกความผิดปกติก่อนที่ปัญหาจะลุกลาม

ความสำคัญของโครงการนี้ครอบคลุมใน 3 มิติหลัก ดังนี้:

1. **ด้านความปลอดภัยและการรักษาสภาพแวดล้อม** ระบบติดตั้งเซ็นเซอร์ตรวจวัดแบบพหุรูปแบบ (Multimodal) ทั้งเซ็นเซอร์อุณหภูมิและความชื้น (DHT22) และเซ็นเซอร์ตรวจจับเปลวไฟ ที่ทำงานเชื่อมต่อกับ AWS Lambda & SES เพื่อส่งการแจ้งเตือนอัตโนมัติไปยังผู้ดูแลระบบแบบทันทีทันใด ไม่ว่าเหตุการณ์จะเกิดขึ้นที่มุมใดของอาคาร ระบบจะรู้ก่อน แจ้งเตือนก่อน และพร้อมรับมือก่อนที่ความเสียหายจะเกิดขึ้น
2. **ด้านประสิทธิภาพเทคโนโลยีโครงข่าย** ไม่มีโปรโตคอลใดที่ "ดีที่สุด" สำหรับทุกสถานการณ์ โครงการนี้จึงออกแบบโครงข่ายแบบ Right Protocol, Right Place โดยในชั้น 2 อุปกรณ์ Multimodal Board ที่ติดตั้งในห้อง R201, Co-Ai และ R200 จะส่งข้อมูลผ่านโปรโตคอล ESP-NOW หรือ nRF Communication มารวมกันที่ Sensor Gateway ณ ห้อง Co-CoE ก่อนส่งต่อขึ้นสู่ระบบ Cloud ในชั้น 3 โครงการยกระดับความเสถียรด้วยสถาปัตยกรรม High Availability Dual Gateway โดยติดตั้ง Network Gateway จำนวน 2 ตัวที่ห้อง PUPA ซึ่งรับข้อมูลจากโหนดเซ็นเซอร์ในห้อง R303 ผ่าน nRF Communication พร้อมกันทั้งสองเส้นทาง หากเกตเวย์ตัวใดตัวหนึ่งขัดข้อง ข้อมูลจะยังคงไหลต่อเนื่องขึ้นสู่ Cloud ได้โดยไม่สะดุด ผลลัพธ์คือโครงข่ายที่ไม่มีจุดล้มเหลวจุดเดียว (No Single Point of Failure)
3. **ด้านการบริหารจัดการข้อมูลและต้นทุน** สถาปัตยกรรมคลาวด์บน AWS ถูกออกแบบให้ข้อมูลไหลอย่างชาญฉลาด โดยใช้ AWS Lambda เป็นเครื่องยนต์ประมวลผลกลางที่รับข้อมูลจาก IoT Core แปลงผล และจัดเก็บลงสู่ DynamoDB อย่างเป็นระเบียบ รองรับข้อมูลแบบ Time-series ในปริมาณมหาศาล ข้อมูลทั้งหมดนี้ไม่ได้หยุดอยู่แค่การแสดงผลบน Next.js Dashboard แต่ยังเป็นเชื้อเพลิง

สำคัญสำหรับ Predictive Analytics ที่จะขับเคลื่อนการบำรุงรักษาเชิงรุกและการประหยัดพลังงานอย่างยั่งยืนในอนาคต

1.2 ปัญหาที่ต้องการแก้

ในการบริหารจัดการอาคารและห้องปฏิบัติการของภาควิชาวิศวกรรมคอมพิวเตอร์ (CoE) ที่ผ่านมาพบประเด็นปัญหาสำคัญที่ส่งผลต่อประสิทธิภาพและความปลอดภัย ดังนี้:

1. **การขาดระบบติดตามตรวจสอบสภาพแวดล้อมแบบเรียลไทม์** ในปัจจุบันการตรวจสอบสภาพแวดล้อมภายในห้องปฏิบัติการ เช่น อุณหภูมิ ความชื้น หรือความเรียบร้อยของห้อง มักดำเนินการโดยเจ้าหน้าที่หรือผู้ใช้งานในพื้นที่ ซึ่งเป็นการตรวจสอบเป็นครั้งคราว (Manual Check) ทำให้ไม่สามารถรับทราบสภาวะที่เกิดขึ้นจริง ณ ขณะนั้นได้ทันที โดยเฉพาะในช่วงนอกเวลาทำการหรือวันหยุด ซึ่งหากเกิดสภาวะผิดปกติ เช่น ระบบปรับอากาศขัดข้องจนทำให้อุณหภูมิสูงเกินเกณฑ์ จะไม่มีผู้รับทราบจนกว่าจะเกิดความเสียหายขึ้น
2. **ความเสี่ยงต่อความเสียหายของอุปกรณ์ราคาสูง** ห้องปฏิบัติการทางคอมพิวเตอร์ เช่น ห้องแม่ข่าย (Server Room) หรือห้องห้องปฏิบัติการคอมพิวเตอร์มีการติดตั้งอุปกรณ์อิเล็กทรอนิกส์ที่มีมูลค่าสูงและมีความไวต่อสภาพแวดล้อม (Environmental Sensitivity) ปัญหาความร้อนสะสม ความชื้นที่สูงเกินไป หรือความผันผวนของกระแสไฟฟ้า อาจส่งผลให้อุปกรณ์เสื่อมสภาพเร็วขึ้นหรือเกิดการหยุดทำงานของระบบ (Downtime) ซึ่งส่งผลกระทบโดยตรงต่อการเรียนการสอนและการวิจัย
3. **ข้อจำกัดด้านโครงสร้างทางกายภาพและการเชื่อมต่อ** การติดตั้งระบบ IoT ภายในอาคารที่มีโครงสร้างเป็นคอนกรีตและมีพื้นที่กว้างขวาง มักประสบปัญหาสัญญาณรบกวนและจุดบอดของสัญญาณ (Dead Zones) การใช้ระบบ Wi-Fi เพียงอย่างเดียวอาจไม่ครอบคลุมและไม่มีความเสถียรเพียงพอ รวมถึงการลากสายสัญญาณสื่อสารไปยังทุกจุดเป็นเรื่องที่ทำได้ยากและมีค่าใช้จ่ายสูง ระบบเดิมจึงขาดความยืดหยุ่นในการขยายตัว (Scalability) เพื่อรองรับเซ็นเซอร์จำนวนมากในตำแหน่งที่แตกต่างกัน
4. **ระบบแจ้งเตือนที่ล่าช้าและไม่มีประสิทธิภาพ** เมื่อเกิดเหตุการณ์วิกฤต เช่น อุณหภูมิพุ่งสูงเกินขีดจำกัดในห้องแม่ข่าย ระดับเสียงผิดปกติในพื้นที่ที่ควรเงียบสงบ หรือความเข้มแสงที่เปลี่ยนแปลงอย่างผิดปกติ ระบบแจ้งเตือนแบบเดิมมักเป็นเพียงการแสดงผลเฉพาะที่ซึ่งหากไม่มีผู้อยู่ในบริเวณนั้นเหตุการณ์อาจลุกลามจนไม่สามารถควบคุมได้ การขาดระบบแจ้งเตือนอัตโนมัติที่สามารถส่งข้อมูลไปยังอุปกรณ์พกพาของผู้ดูแลระบบได้โดยตรงถือเป็นช่องโหว่สำคัญด้านความปลอดภัยที่ต้องได้รับการแก้ไข
5. **ความซับซ้อนและต้นทุนในการบริหารจัดการข้อมูลคลาวด์** การส่งข้อมูลจากเซ็นเซอร์จำนวนมหาศาลขึ้นสู่ระบบคลาวด์มักมีค่าใช้จ่ายสูงหากไม่มีการออกแบบสถาปัตยกรรมที่ดี ปัญหาที่พบคือ

การประมวลผลข้อมูลที่ซ้ำซ้อนและการใช้ทรัพยากรคลาวด์ที่เกินความจำเป็น (Over-provisioning) ทำให้โครงการที่ต้องดำเนินการในระยะยาวประสบปัญหาด้านงบประมาณในการบำรุงรักษาระบบคลาวด์

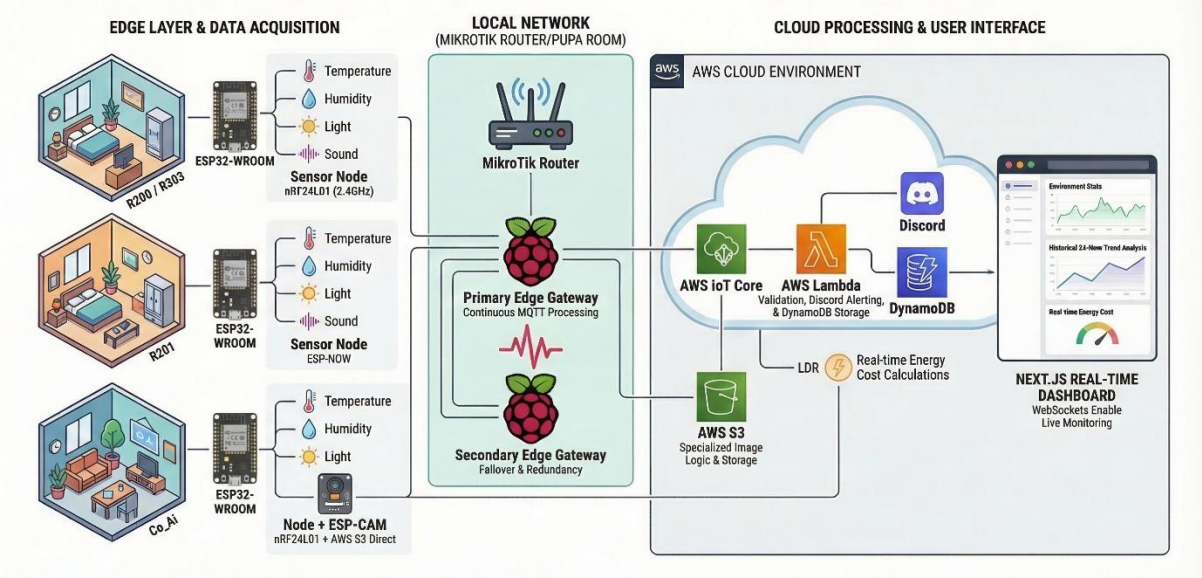
6. **จุดอ่อนของการมีเกตเวย์เดี่ยว** ในระบบ IoT ทั่วไป หากเกตเวย์หลัก (Main Gateway) ที่ทำหน้าที่ส่งข้อมูลขึ้นคลาวด์เกิดขัดข้อง จะทำให้ข้อมูลจากเซ็นเซอร์ทั้งหมดในพื้นที่นั้นสูญหายและไม่สามารถติดตามสถานะได้ การขาดระบบสำรองข้อมูล (Redundancy) ในระดับเกตเวย์จึงเป็นปัญหาสำคัญที่ทำให้ระบบขาดความน่าเชื่อถือ (Reliability) ในการใช้งานจริง

2. วัตถุประสงค์

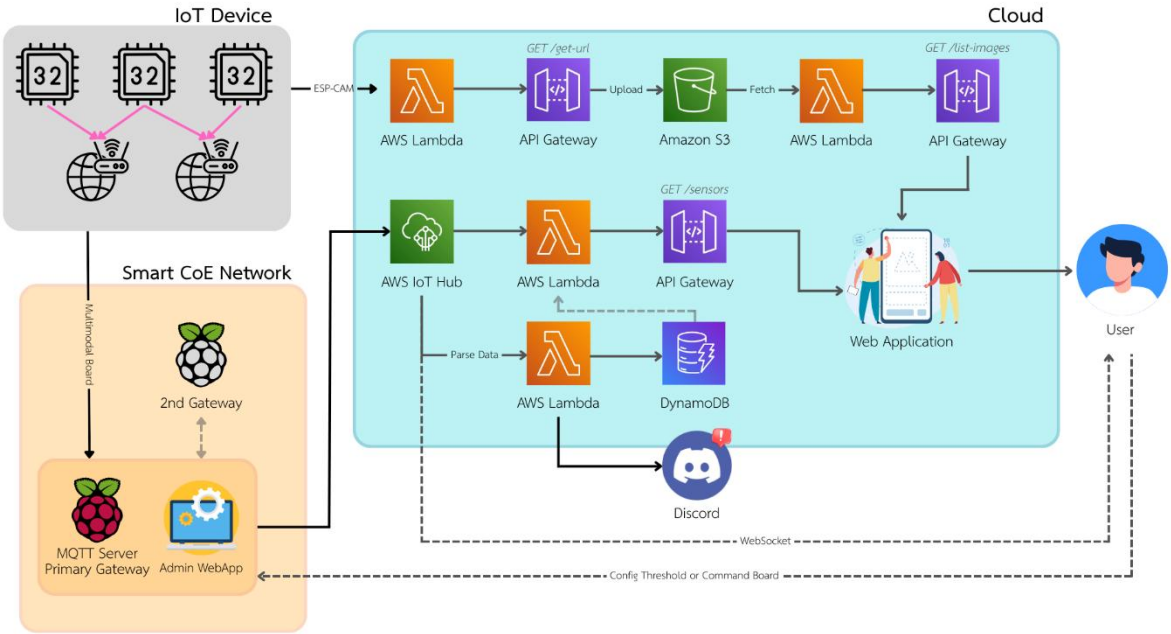
1. **เพื่อออกแบบและพัฒนาระบบ IoT สำหรับเฝ้าระวังสภาพแวดล้อมภายในอาคารภาควิชาวิศวกรรมคอมพิวเตอร์แบบครบวงจร** โดยครอบคลุมการเก็บข้อมูลอุณหภูมิ ความชื้น ระดับแสง และระดับเสียง ผ่านเซ็นเซอร์ที่ติดตั้งในห้องต่างๆ
2. **เพื่อออกแบบสถาปัตยกรรมเครือข่ายแบบลำดับชั้น** ที่รองรับโปรโตคอลการสื่อสารหลายรูปแบบ ได้แก่ nRF24L01 และ ESP-NOW เพื่อให้ครอบคลุมพื้นที่ใช้งานในแต่ละชั้นอาคารได้อย่างมีประสิทธิภาพ
3. **เพื่อพัฒนา Edge Gateway ที่มีความพร้อมใช้งานสูง** โดยใช้ Raspberry Pi 5 เป็น Primary Gateway และ Raspberry Pi 3 เป็น Secondary Gateway พร้อมกลไก Failover อัตโนมัติ เพื่อให้ระบบสามารถรับส่งข้อมูลได้อย่างต่อเนื่องไม่ขาดหาย
4. **เพื่อบูรณาการระบบกับโครงสร้างพื้นฐานคลาวด์ของ AWS** ประกอบด้วย AWS IoT Core, AWS Lambda, Amazon DynamoDB และ Amazon S3 เพื่อรองรับการจัดเก็บ ประมวลผล และแสดงผลข้อมูลแบบ Serverless
5. **เพื่อพัฒนาเว็บแอปพลิเคชันสำหรับแสดงผลและวิเคราะห์ข้อมูลแบบเรียลไทม์** รวมถึงการดูประวัติข้อมูลย้อนหลัง การวิเคราะห์การใช้พลังงานไฟฟ้า และการแสดงภาพถ่ายภายในห้องจาก ESP-CAM
6. **เพื่อพัฒนาระบบแจ้งเตือนอัตโนมัติ** เมื่อค่าเซ็นเซอร์เกินเกณฑ์ที่กำหนด ผ่านทั้ง Passive Buzzer บนตัวบอร์ดและการแจ้งเตือนผ่าน Discord เพื่อให้ผู้ดูแลระบบสามารถรับรู้และตอบสนองต่อสถานการณ์ผิดปกติได้อย่างทันท่วงที

3. การออกแบบระบบ

3.1 Architecture diagram



รูป 1 Architecture Diagram



รูป 2 Detailed Architecture Diagram

Smart CoE Monitoring ถูกออกแบบด้วยสถาปัตยกรรมแบบลำดับชั้น (Hierarchical IoT Architecture) เพื่อให้ระบบมีความเสถียร รองรับการสำรองการทำงานและบริหารจัดการข้อมูลได้อย่างมีประสิทธิภาพ การทำงานเริ่มต้นที่ชั้นการตรวจวัดและรวบรวมข้อมูล (Edge Layer) ซึ่งประกอบด้วย Multimodal Board ที่พัฒนาบน ESP32 ติดตั้งกระจายตามพื้นที่ต่างๆ เช่น ห้อง R200, R201, R303 และ

AI Co-Working Space บอร์ดเหล่านี้ทำหน้าที่อ่านค่าจากเซ็นเซอร์วัดอุณหภูมิ ความชื้น แสง และเสียง นอกจากนี้ในห้อง AI Co-Working Space ยังมีการติดตั้งอุปกรณ์ ESP-CAM เพื่อทำหน้าที่ถ่ายภาพสภาพแวดล้อมภายในห้องอีกด้วย

ข้อมูลสภาพแวดล้อมที่อ่านได้จะถูกส่งผ่านโปรโตคอลไร้สาย (nRF24L01 สำหรับพื้นที่ส่วนใหญ่ และ ESP-NOW สำหรับห้อง R201) ไปรวบรวมที่ Sensor Gateway ประจำชั้น จากนั้นข้อมูลจะถูกแปลงเป็นโปรโตคอล MQTT และส่งผ่านวงเครือข่ายท้องถิ่นที่ถูกแยกส่วนเพื่อความปลอดภัยด้วย MikroTik Router ไปยัง Dual Edge Gateway ที่ตั้งอยู่ในห้อง PUPA โดยโครงสร้างของศูนย์กลางขอบเครือข่ายนี้ออกแบบมาเพื่อรองรับการทำงานที่ต่อเนื่อง ดังนี้:

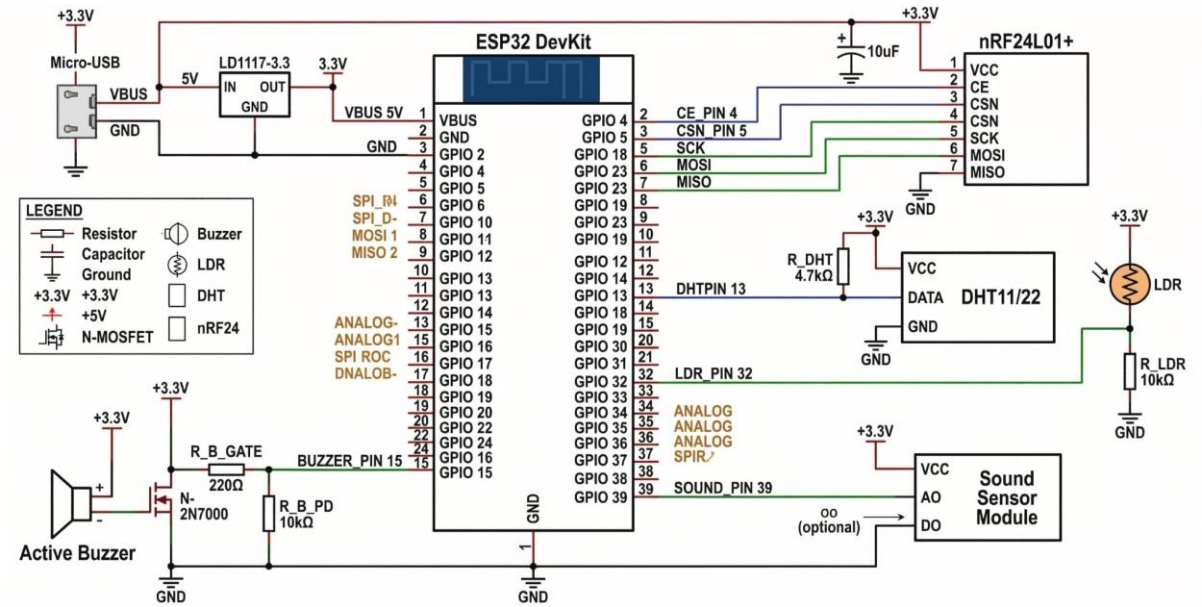
- **Primary Gateway (Raspberry Pi 5):** ทำหน้าที่เป็นเซิร์ฟเวอร์ MQTT หลัก และรับผิดชอบการประมวลผลข้อมูลที่ขอบเครือข่าย (Edge Processing) ก่อนส่งขึ้นระบบคลาวด์
- **Secondary Gateway (Raspberry Pi 3):** ทำหน้าที่เป็นระบบสำรอง (High Availability/Failover) ซึ่งจะคอยตรวจสอบสถานะการทำงานและพร้อมรับช่วงต่อทันทีหากระบบหลักเกิดข้อขัดข้อง

สำหรับการประมวลผลส่วนกลาง ระบบเลือกใช้สถาปัตยกรรมคลาวด์แบบ Serverless บนแพลตฟอร์ม AWS ข้อมูลจาก Edge Gateway จะถูกส่งผ่านการเชื่อมต่อที่ปลอดภัย (mTLS) ไปยัง AWS IoT Core จากนั้น AWS Lambda จะรับหน้าที่ตรวจสอบและบันทึกข้อมูลลงในฐานข้อมูล DynamoDB พร้อมทั้งประเมินเงื่อนไขเพื่อส่งการแจ้งเตือนไปยังแอปพลิเคชัน Discord เมื่อพบค่าเซ็นเซอร์ที่ผิดปกติ ในส่วนของข้อมูลภาพจาก ESP-CAM ระบบจะทำการขอ Presigned URL ผ่าน API Gateway และ Lambda เพื่อให้อุปกรณ์สามารถอัปโหลดภาพตรงเข้าสู่ Amazon S3 ได้อย่างปลอดภัย

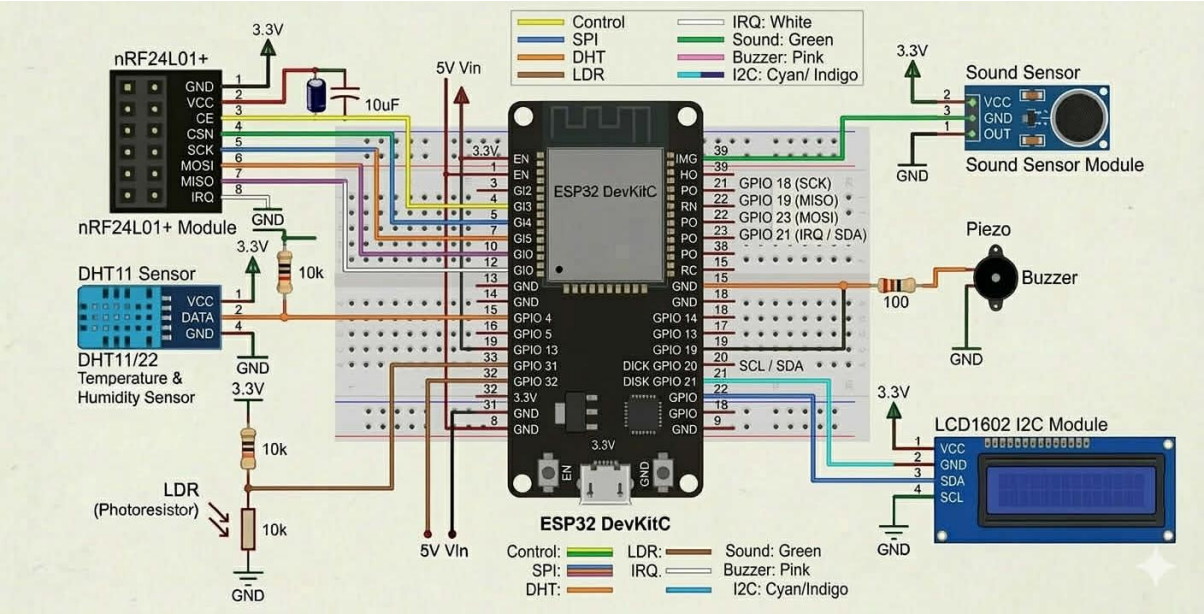
ท้ายที่สุด ข้อมูลที่ถูกจัดเก็บและประมวลผลทั้งหมดจะถูกนำไปแสดงผลบนเว็บแอปพลิเคชันที่พัฒนาด้วย Next.js โดยมีคุณสมบัติหลักดังนี้:

- **การแสดงผลแบบเรียลไทม์:** ระบบใช้ WebSockets เชื่อมต่อกับ AWS IoT Core โดยตรง ทำให้สามารถแสดงสถานะเซ็นเซอร์ปัจจุบันได้ทันที
- **การดูข้อมูลย้อนหลังและภาพถ่าย:** ระบบจะเชื่อมต่อผ่าน API Gateway เพื่อดึงข้อมูลประวัติจาก DynamoDB และดึงรายการภาพจาก S3 มาแสดงผลบนหน้าต่างประวัติการใช้งาน
- **การวิเคราะห์การใช้พลังงาน:** ระบบสามารถประเมินและคำนวณค่าใช้จ่ายพลังงานไฟฟ้า (Real-time Energy Cost) โดยอ้างอิงจากสถานะของเซ็นเซอร์แสง (LDR) ที่ติดตั้งไว้

3.2 Hardware design



รูป 3 Full Schematic of Multimodal Board



รูป 4 Schematic of Multimodal Board

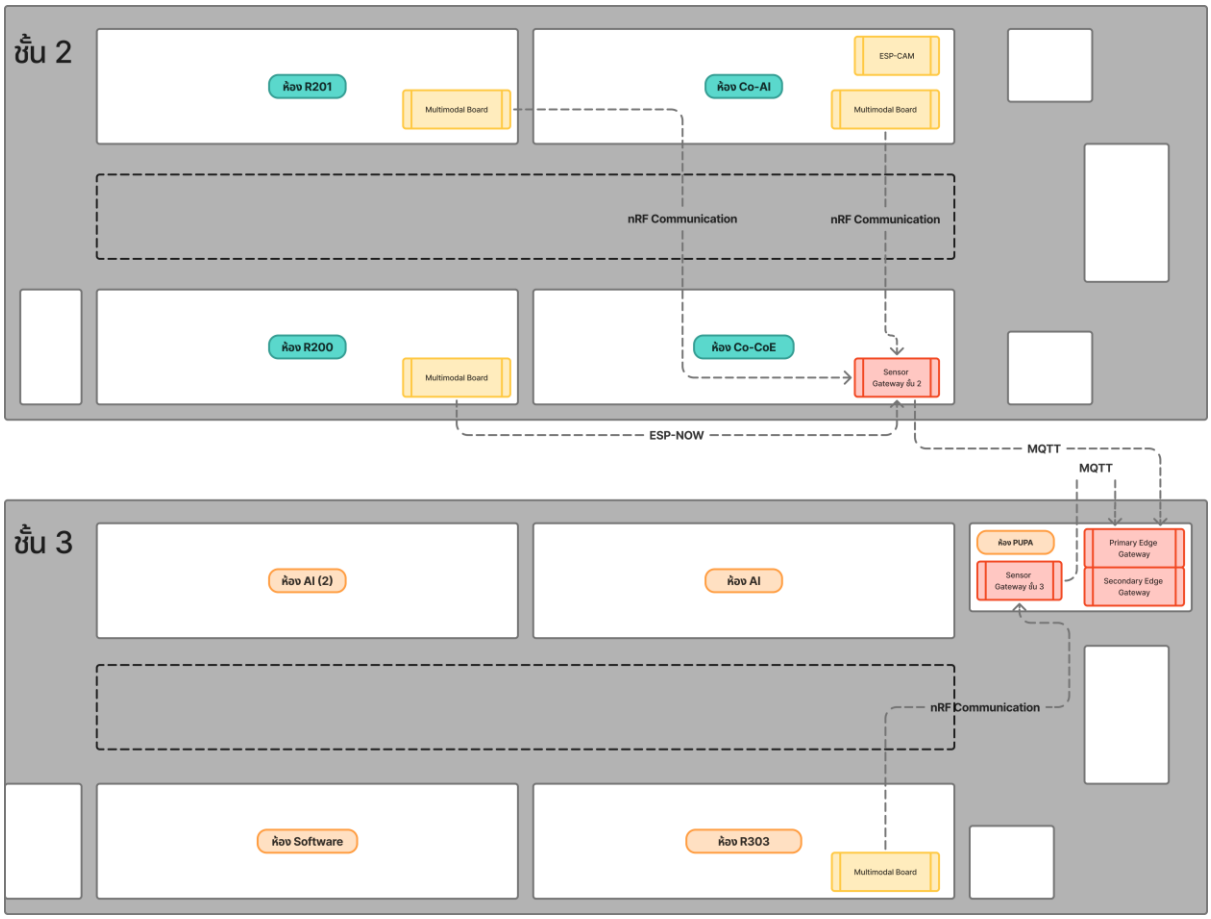
ระบบใช้การออกแบบศูนย์กลางการตรวจวัดสภาพแวดล้อมผ่าน Multimodal Board ซึ่งพัฒนาขึ้นโดยมีตัวควบคุมหลักเป็นไมโครคอนโทรลเลอร์ ESP32 ในส่วนของการจัดการพลังงาน บอร์ดจะรับแหล่งจ่ายไฟผ่านช่องเชื่อมต่อ Micro-USB เพื่อนำพลังงานไปเลี้ยงวงจรและอุปกรณ์เซ็นเซอร์ส่วนประกอบต่างๆ ภายในระบบสำหรับการตรวจวัดสภาพแวดล้อม บอร์ดได้รวบรวมเซ็นเซอร์หลายชนิดเพื่อการเก็บข้อมูลที่ครอบคลุม ดังนี้:

- **เซ็นเซอร์อุณหภูมิและความชื้น:** ใช้โมดูลเซ็นเซอร์รุ่น DHT11 แบบสำเร็จรูป ซึ่งมีการติดตั้งวงจรตัวต้านทานแบบ Pull-up ไว้ในตัวโมดูลแล้ว ทำให้สามารถเชื่อมต่อและรับส่งข้อมูลแบบ One-Wire ผ่านขา GPIO 13 ได้โดยตรง
- **เซ็นเซอร์แสง:** ใช้โมดูลเซ็นเซอร์แสง (LDR Module) แบบสำเร็จรูป ซึ่งมีวงจรแบ่งแรงดัน (Voltage Divider) ประกอบมาให้ภายในตัว โดยจะส่งสัญญาณแอนะล็อกเข้าไปยังขา GPIO 32 เพื่อให้ระบบนำไปประเมินสถานะการเปิด-ปิดแสงสว่าง
- **เซ็นเซอร์เสียง:** ใช้โมดูลเซ็นเซอร์เสียงแบบแอนะล็อกเชื่อมต่อสัญญาณเอาต์พุต (AO) เข้ากับ GPIO 39 ของ ESP32 เพื่อใช้วัดระดับความดังของเสียงภายในพื้นที่
- **โมดูลกล้อง (เฉพาะพื้นที่):** พื้นที่ AI Co-Working Space (Co_Ai) มีการติดตั้งบอร์ด ESP32-CAM แยกต่างหากเพื่อใช้ในการบันทึกและส่งข้อมูลภาพสภาพแวดล้อมภายในห้อง

ในด้านการแสดงผลและการแจ้งเตือนบนตัวบอร์ด มีการติดตั้งหน้าจอ LCD ขนาด 16x2 ผ่านบัสการสื่อสาร I2C (GPIO 21 สำหรับ SDA และ GPIO 22 สำหรับ SCL) เพื่อแสดงค่าเซ็นเซอร์ปัจจุบัน พร้อมทั้งมี Passive Buzzer สำหรับส่งเสียงเตือนเมื่อค่าสภาพแวดล้อมเกินเกณฑ์ที่กำหนด โดยการทำงานของ Buzzer จะถูกควบคุมผ่าน GPIO 15

ระบบการสื่อสารข้อมูลแบบไร้สายของโหนดถูกออกแบบให้สอดคล้องกับข้อจำกัดของพื้นที่ โดยโหนดเซ็นเซอร์ส่วนใหญ่จะติดตั้งโมดูลรับส่งสัญญาณวิทยุ nRF24L01+ ที่สื่อสารผ่านบัส SPI (ขา SCK, MOSI, MISO, CE และ CSN) ซึ่งในวงจรได้เพิ่มตัวเก็บประจุขนาด 10μF เข้าที่สายไฟ 3.3V เพื่อกรองสัญญาณรบกวนให้การสื่อสารมีความเสถียร อย่างไรก็ตาม สำหรับห้องปฏิบัติการ R201 เนื่องจากมีพื้นที่ตั้งอยู่ใกล้กับ Sensor Gateway จึงได้ปรับเปลี่ยนมาใช้โปรโตคอล ESP-NOW ในการส่งข้อมูลโดยตรงโดยไม่ได้ผ่านเครือข่าย Wi-Fi ทำให้ไม่มีการติดตั้งโมดูล nRF24L01+ บนบอร์ดในพื้นที่ดังกล่าว

3.3 Communication design



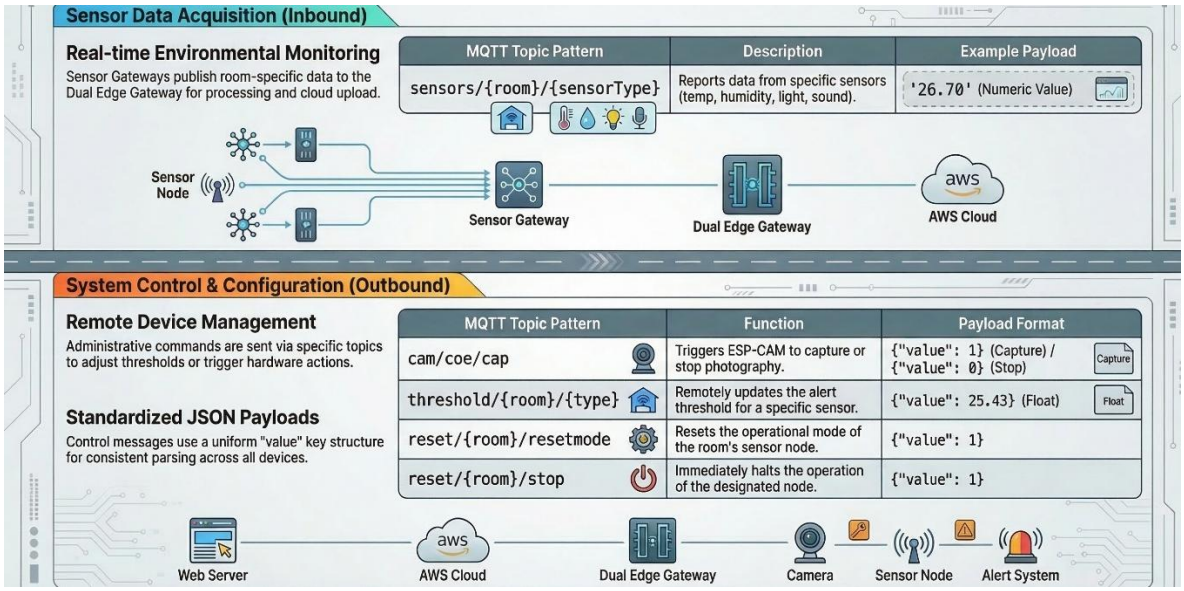
รูป 5 Communication Design Diagram

การออกแบบระบบสื่อสารข้อมูลถูกจัดโครงสร้างแบบลำดับชั้นเพื่อให้ครอบคลุมพื้นที่ใช้งานและมีความเสถียรสูง โดยระบบได้กำหนดเส้นทางการไหลของข้อมูล (Data Flow) จากจุดตรวจวัดแต่ละพื้นที่ไปยังศูนย์กลางอย่างเป็นระบบ ดังนี้:

- **ห้อง R200 และ AI Co-Working Space (ชั้น 2):** โหนดเซ็นเซอร์ส่งข้อมูลผ่านคลื่นวิทยุด้วยโปรโตคอล nRF24L01 ไปรวบรวมที่ Sensor Gateway ประจำชั้น 2 ซึ่งตั้งอยู่ที่ห้อง CoE Co-Working Space
- **ห้อง R201 (ชั้น 2):** เนื่องจากตำแหน่งติดตั้งอยู่ใกล้กับเกตเวย์ จึงใช้โปรโตคอล ESP-NOW ในการส่งข้อมูลไปยัง Sensor Gateway ชั้น 2 โดยตรง
- **ห้อง R303 (ชั้น 3):** โหนดเซ็นเซอร์ส่งข้อมูลผ่านคลื่นวิทยุด้วยโปรโตคอล nRF24L01 ไปยัง Sensor Gateway ประจำชั้น 3
- **การแปลงและส่งต่อข้อมูล (Sensor Gateway):** เมื่อ Sensor Gateway ของทั้งชั้น 2 และชั้น 3 ได้รับข้อมูลสภาพแวดล้อมแล้ว จะทำหน้าที่รวบรวมและแปลงข้อมูลให้อยู่ในรูปแบบโปรโตคอล

MQTT ก่อนส่งต่อไปยังศูนย์กลางการประมวลผลแบบขอบเครือข่าย (Dual Edge Gateway) ที่ตั้งอยู่ในห้อง PUPA

- **การรักษาความปลอดภัย:** โหนดเซ็นเซอร์ (Multimodal Board) ในทุกห้องจะทำงานในระบบปิด โดยไม่มีการเชื่อมต่ออินเทอร์เน็ตโดยตรง มีเพียง Sensor Gateway เท่านั้นที่เชื่อมต่อเครือข่ายเพื่อส่งข้อมูลไปยังเซิร์ฟเวอร์ MQTT ซึ่งระบบได้จำกัดสิทธิ์ให้เข้าถึงเซิร์ฟเวอร์ MQTT ได้ผ่านเทคนิค Port Forwarding บนวงเครือข่ายที่แยกไว้เฉพาะเท่านั้น เพื่อป้องกันการเข้าถึงจากภายนอก



รูป 6 MQTT Communication Design

รูปที่ 6 กล่าวถึงการออกแบบหัวข้อการสื่อสาร (MQTT Topic) ของระบบถูกจัดโครงสร้างแบบลำดับชั้นและแบ่งทิศทางการไหลของข้อมูลออกเป็น 2 ทิศทางหลักอย่างชัดเจน ได้แก่ ขาเข้า (Inbound) และขาออก (Outbound) โดยมีรายละเอียดการออกแบบดังนี้:

- **ทิศทางข้อมูลขาเข้า (Inbound - Sensor Data):** เป็นเส้นทางการส่งข้อมูลสภาพแวดล้อมจากโหนดเซ็นเซอร์เข้าสู่ระบบส่วนกลาง โดยใช้ Topic ในรูปแบบ **sensors/{ชื่อห้อง}/{ประเภทเซ็นเซอร์}** (เช่น sensors/R200/temperature) ซึ่งข้อมูลที่ส่งมา (Payload) จะอยู่ในรูปแบบ JSON ที่ประกอบด้วยค่าของเซ็นเซอร์ที่อ่านได้ในขณะนั้นและสถานะการประเมินความผิดปกติ (isAlert) เพื่อให้ระบบนำไปจัดเก็บและแสดงผลต่อไป

- **ทิศทางข้อมูลขาออก (Outbound - Command & Control):** เป็นเส้นทางการส่งคำสั่งจากเว็บเซิร์ฟเวอร์บน Primary Gateway หรือระบบส่วนกลาง เพื่อกลับไปควบคุมและตั้งค่าการทำงานของโหนดปลายทางแบบเฉพาะเจาะจง โดยใช้ Topic รูปแบบพิเศษตามวัตถุประสงค์การสั่งการ เช่น การตั้งค่าเกณฑ์แจ้งเตือนของเซ็นเซอร์ **(threshold/{ชื่อห้อง}/{ประเภทเซ็นเซอร์})**, การสั่งให้โมดูล

ESP-CAM ถ่ายภาพสภาพแวดล้อม **(cam/coe/cap)**, การสั่งรีเซ็ตโหมดการทำงานของห้อง **(reset/{ชื่อห้อง}/resetmode)**, และการสั่งหยุดการทำงานของโหนด **(reset/{ชื่อห้อง}/stop)**

4. การพัฒนา

4.1 การเชื่อมต่ออุปกรณ์

การพัฒนาในส่วนของการเชื่อมต่ออุปกรณ์มุ่งเน้นที่การบูรณาการฮาร์ดแวร์และการสื่อสารข้อมูลภายในพื้นที่อย่างปลอดภัยและมีประสิทธิภาพ อุปกรณ์ปลายทาง (Sensor Node) ถูกพัฒนาขึ้นในรูปแบบ Multimodal Board โดยใช้ไมโครคอนโทรลเลอร์ ESP32 เป็นศูนย์กลาง รับพลังงานผ่านช่องเชื่อมต่อ Micro-USB และเชื่อมต่อกับโมดูลเซ็นเซอร์สำเร็จรูปต่างๆ ได้แก่ โมดูลเซ็นเซอร์อุณหภูมิและความชื้น (DHT11/22) ผ่านขา GPIO 13, โมดูลเซ็นเซอร์แสง (LDR) ผ่านขา GPIO 32, และโมดูลเซ็นเซอร์เสียงผ่านขา GPIO 39 พร้อมทั้งติดตั้งหน้าจอ LCD 16x2 ผ่านบัส I2C และ Passive Buzzer ที่ขา GPIO 15 สำหรับแสดงผลและแจ้งเตือนสถานะบนตัวบอร์ด

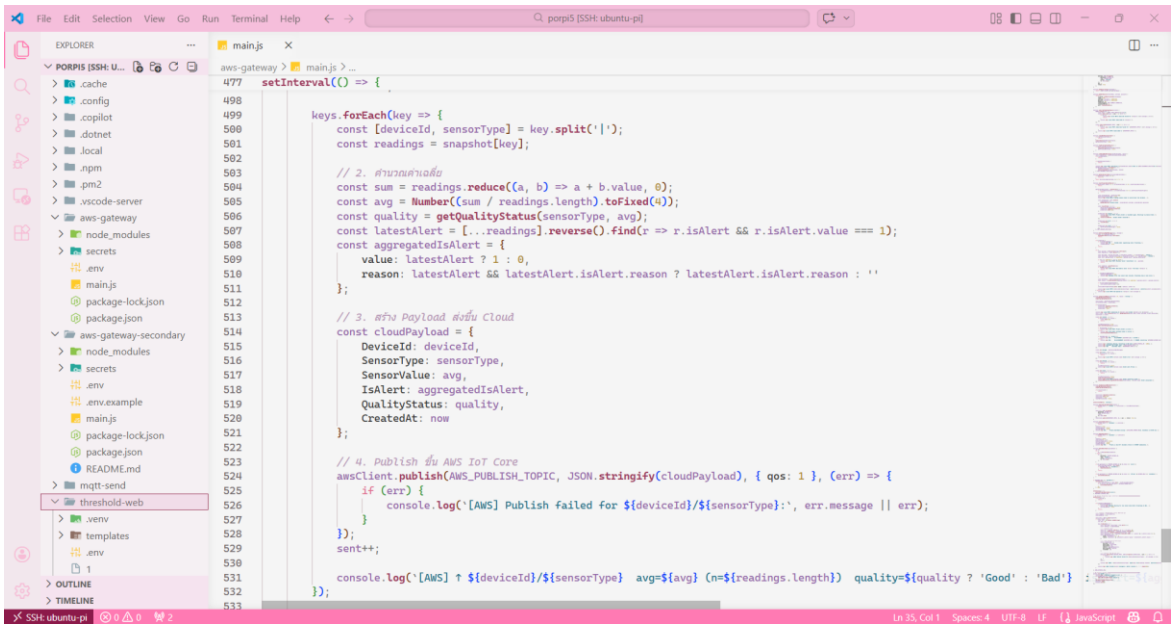
ในด้านการส่งข้อมูล โหนดเซ็นเซอร์ทั้งหมดจะทำงานในระบบปิดโดยไม่มีการเชื่อมต่ออินเทอร์เน็ตโดยตรง เพื่อความปลอดภัยสูงสุดของระบบ โดยมีรูปแบบการเชื่อมต่อดังนี้:

- **การสื่อสารผ่านคลื่นวิทยุ:** พื้นที่ติดตั้งส่วนใหญ่ (ห้อง R200, AI Co-Working Space และ R303) มีการเชื่อมต่อโมดูลรับส่งสัญญาณวิทยุ nRF24L01+ ผ่านบัส SPI พร้อมเพิ่มตัวเก็บประจุ 10μF เพื่อความเสถียรของสัญญาณ ในการส่งข้อมูลไปยัง Sensor Gateway ประจำชั้น
- **การสื่อสารผ่าน ESP-NOW:** พื้นที่ห้อง R201 ใช้โปรโตคอล ESP-NOW ในการเชื่อมต่อเพื่อส่งข้อมูลตรงไปยัง Sensor Gateway ผ่านวงจรภายในตัว ESP32 โดยไม่ต้องติดตั้งโมดูล nRF24L01+ เพิ่มเติม
- **การเชื่อมต่อเข้าสู่เครือข่ายหลัก:** เมื่อ Sensor Gateway ได้รับข้อมูลแล้ว จะแปลงเป็นโปรโตคอล MQTT และเชื่อมต่ออินเทอร์เน็ตเพื่อส่งข้อมูลเข้าสู่เซิร์ฟเวอร์ MQTT ที่อยู่ในวงเครือข่ายที่ถูกแยกส่วน โดยการเข้าถึงนี้จะทำผ่านเทคนิค Port Forwarding บน MikroTik Router เท่านั้น เพื่อป้องกันการเข้าถึงจากภายนอกที่ไม่ได้รับอนุญาต

4.2 Software/Cloud

ระบบซอฟต์แวร์และคลาวด์ถูกพัฒนาขึ้นเพื่อรองรับการไหลของข้อมูลตั้งแต่ขอบเครือข่ายไปจนถึงการจัดเก็บและการแสดงผล โดยใช้สถาปัตยกรรมแบบ Serverless บนแพลตฟอร์ม AWS เพื่อให้ระบบสามารถขยายขนาดและดูแลรักษาได้ง่าย ซึ่งมีกระบวนการพัฒนาหลักดังนี้:

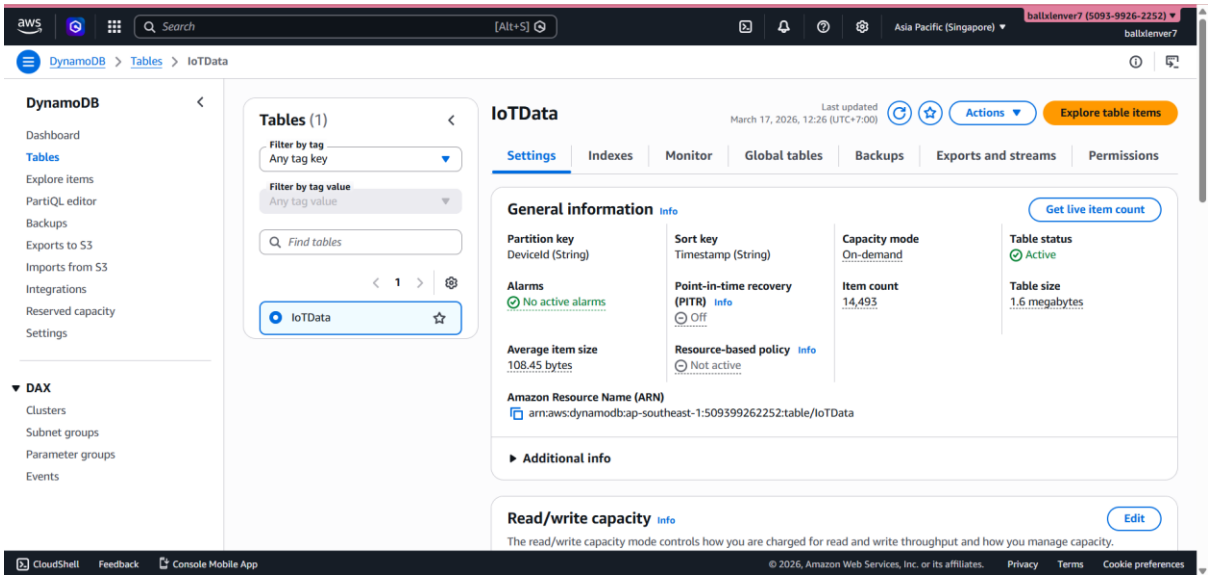
- **การประมวลผลที่ขอบเครือข่าย (Edge Processing):** มีการพัฒนาซอฟต์แวร์บน Dual Edge Gateway (Raspberry Pi 5 เป็นเครื่องหลัก และ Raspberry Pi 3 เป็นเครื่องสำรองแบบ High Availability) ทำหน้าที่อ่านค่าการตั้งค่าจากไฟล์ .env รับข้อมูลผ่าน MQTT ตรวจสอบความถูกต้อง จัดกลุ่ม และคำนวณค่าเฉลี่ยก่อนจัดส่งข้อมูลขึ้นสู่คลาวด์



รูป 7 ตัวอย่างโปรแกรมที่ทำหน้าประมวลผลและส่งต่อข้อมูลบน Raspberry Pi

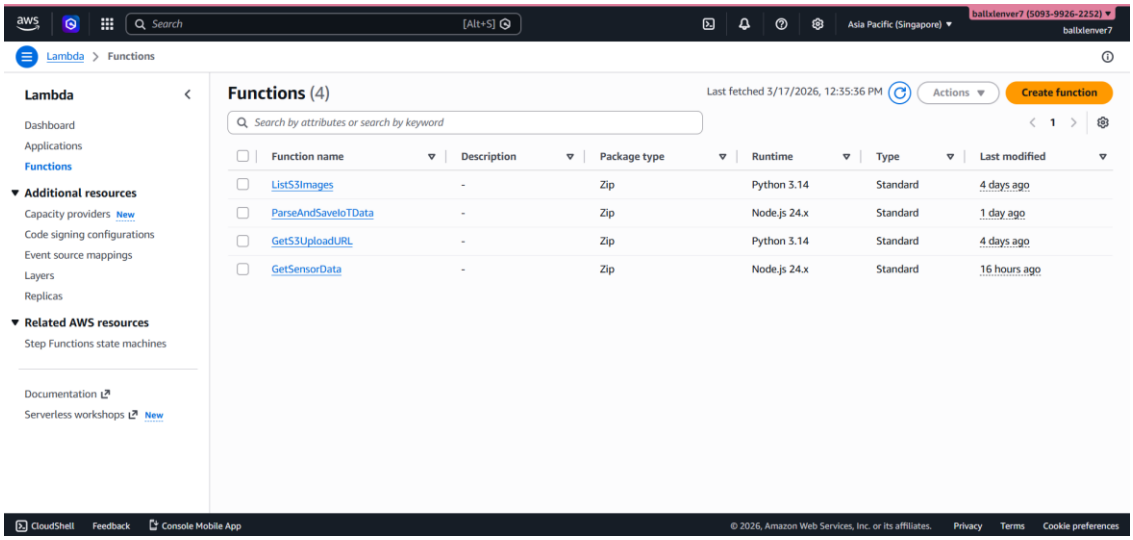
รูป 8 การไหลของข้อมูลบน MQTT Server

- **การจัดเก็บข้อมูลผ่าน AWS Serverless:** ข้อมูลจากเกตเวย์จะถูกส่งผ่านช่องทางที่เข้ารหัส (mTLS) ไปยัง AWS IoT Core จากนั้นกฎของ IoT (IoT Rules Engine) จะเรียกใช้งาน AWS Lambda เพื่อทำการบันทึกข้อมูลสถานะต่างๆ ลงในฐานข้อมูล Amazon DynamoDB โดยมีการตั้งค่าอายุของข้อมูล (TTL) ไว้ที่ 7 วัน เพื่อล้างข้อมูลเก่าอัตโนมัติ และหลังจากครบ 7 วัน ข้อมูลประวัติดังกล่าวจะถูกย้ายไปจัดเก็บไว้บน Amazon S3 แทน เพื่อช่วยลดต้นทุนในการจัดเก็บข้อมูลระยะยาว



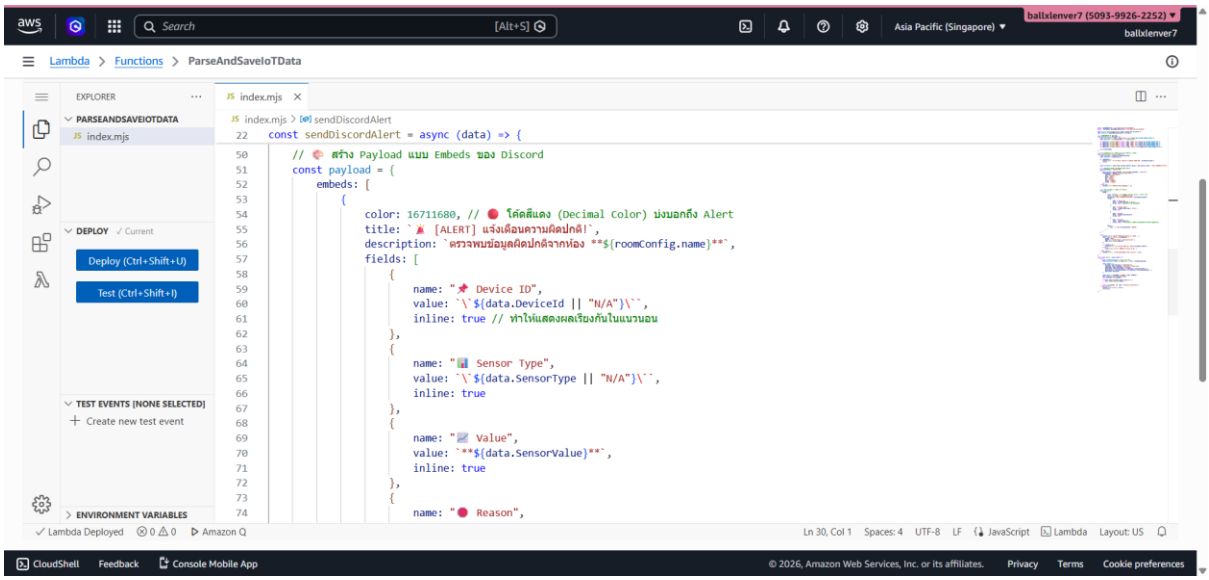
รูป 9 รายละเอียด DynamoDB บน AWS

- **การจัดการรูปภาพจาก ESP-CAM:** พัฒนาฟังก์ชันการทำงานให้โมดูลกล้องในห้อง AI Co-Working Space สามารถร้องขอ Presigned URL (มีอายุ 5 นาที) ผ่าน API Gateway และ Lambda (GetS3UploadURL) เพื่อใช้อัปโหลดภาพถ่ายตรงเข้าสู่ระบบจัดเก็บ Amazon S3 ได้อย่างปลอดภัย จากนั้นหน้าเว็บจะทำการร้องขอผ่าน API Gateway และ Lambda ที่มีชื่อว่า ListS3Images



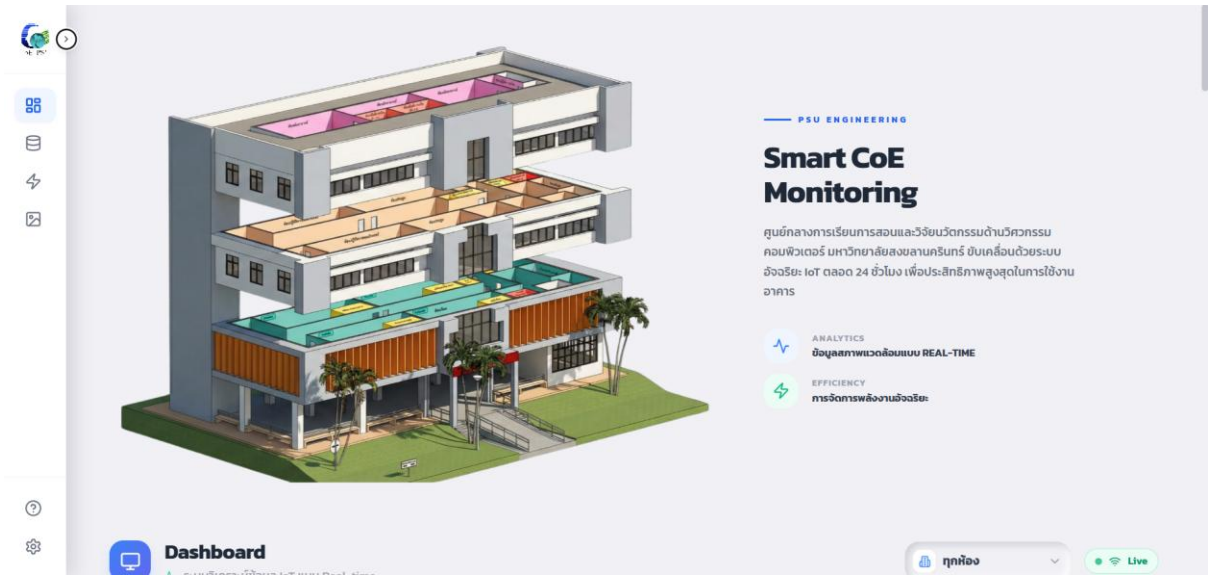
รูป 10 ฟังก์ชั่นบน AWS Lambda ที่มีการใช้งานในเว็บ

- **ระบบแจ้งเตือนความผิดปกติ:** มีการเขียนโปรแกรมบน AWS Lambda (ParseAndSaveIoTData) ให้ตรวจสอบค่าสถานะการแจ้งเตือน หากพบค่าเซ็นเซอร์ที่ผิดเกณฑ์ ระบบจะส่ง Webhook พร้อมจัดรูปแบบข้อความเพื่อแจ้งเตือนไปยังแอปพลิเคชัน Discord ทันที และบันทึกข้อมูลลงใน DynamoDB

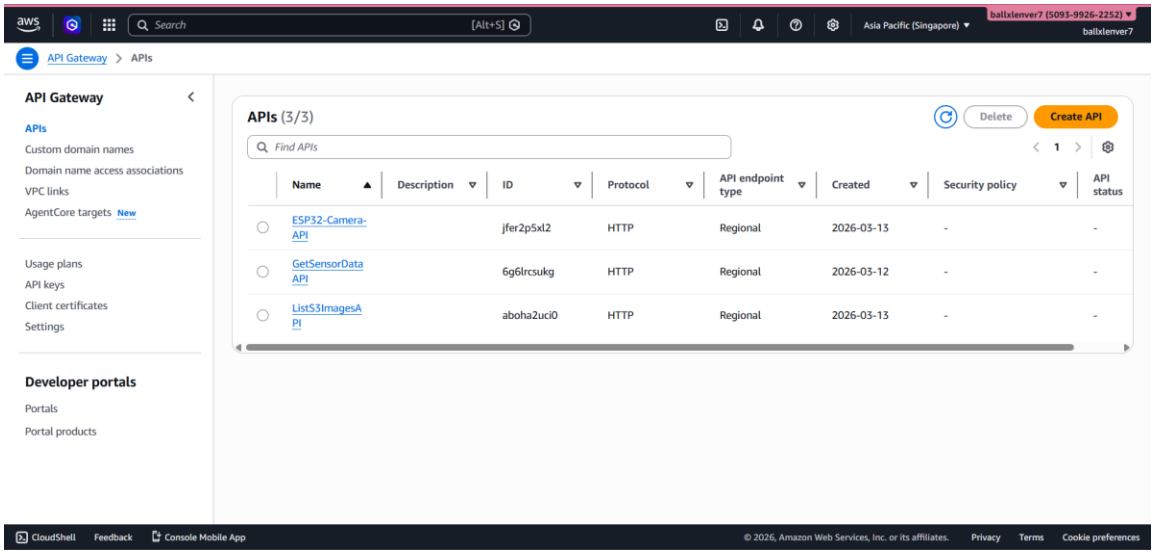


รูป 11 ตัวอย่างฟังก์ชั่นการทำงานของ ParseAndSaveIoTData

- **เว็บแอปพลิเคชันสำหรับ Monitoring:** ส่วนติดต่อผู้ใช้งานถูกพัฒนาด้วยกรอบการทำงาน Next.js โดยรองรับการดึงข้อมูลเพื่อแสดงผลแบบเรียลไทม์ผ่าน WebSocket ที่ต่อตรงกับ AWS IoT Core และรองรับการดึงข้อมูลย้อนหลัง (ประวัติเซ็นเซอร์และรูปภาพ) ผ่าน API Gateway รวมทั้งสามารถคำนวณและประเมินค่าใช้จ่ายพลังงานไฟฟ้าได้แบบเรียลไทม์

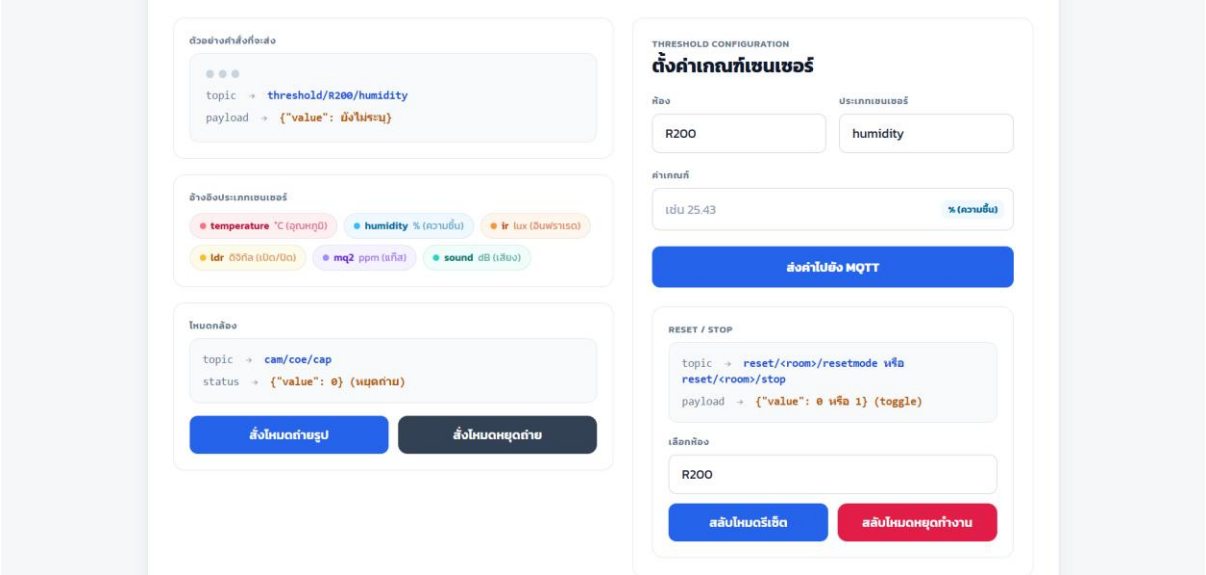


รูป 12 หน้าเว็บ Smart CoE Monitoring



รูป 13 API Gateway บน AWS ที่เว็บไซต์มีการเรียกใช้งาน

- **เว็บแอปพลิเคชันสำหรับการตั้งค่าระบบ:** ทำงานอยู่บนเว็บเซิร์ฟเวอร์ของ Primary Edge Gateway (Raspberry Pi 5) ซึ่งถูกจำกัดการเข้าถึงให้อยู่เฉพาะภายในวงเครือข่ายปิดของ Router (หรือเข้าถึงผ่าน VPN) เพื่อความปลอดภัยสูงสุด โดยระบบนี้ทำหน้าที่ส่งคำสั่งควบคุมการทำงานผ่าน MQTT Topic ไปยังโหนดปลายทาง เช่น การปรับเปลี่ยนเกณฑ์การแจ้งเตือน (Threshold) การสั่งถ่ายภาพจากกล้อง ESP-CAM และการสั่งรีเซ็ตหรือหยุดการทำงานของโหนดเซ็นเซอร์ในแต่ละพื้นที่



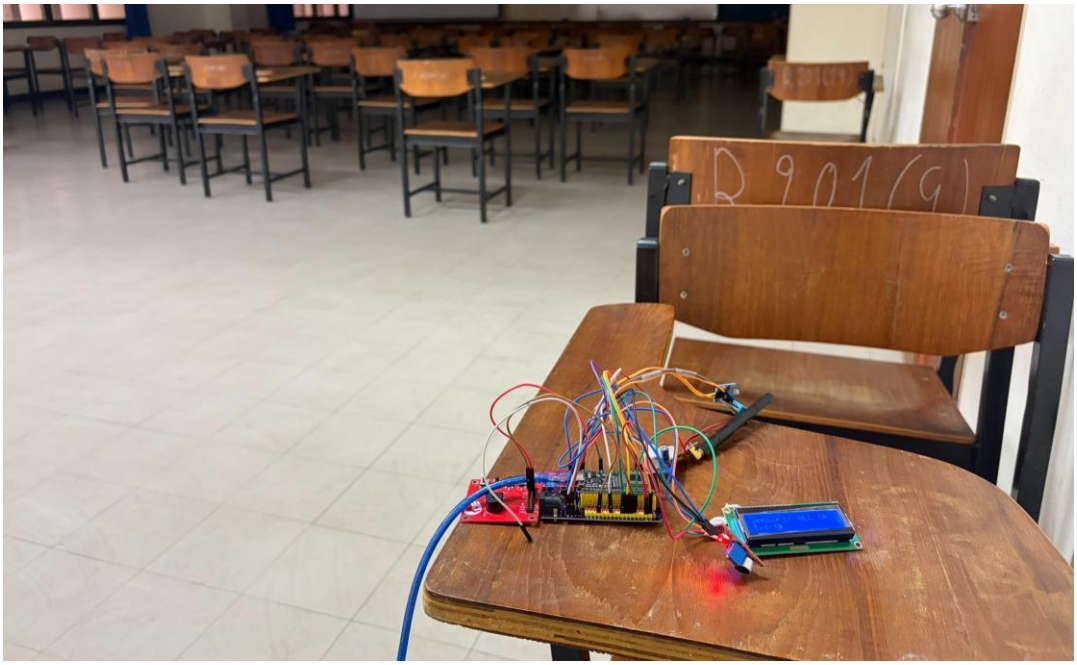
รูป 14 เว็บไซต์สำหรับการตั้งค่าระบบ

5. การทดสอบระบบ

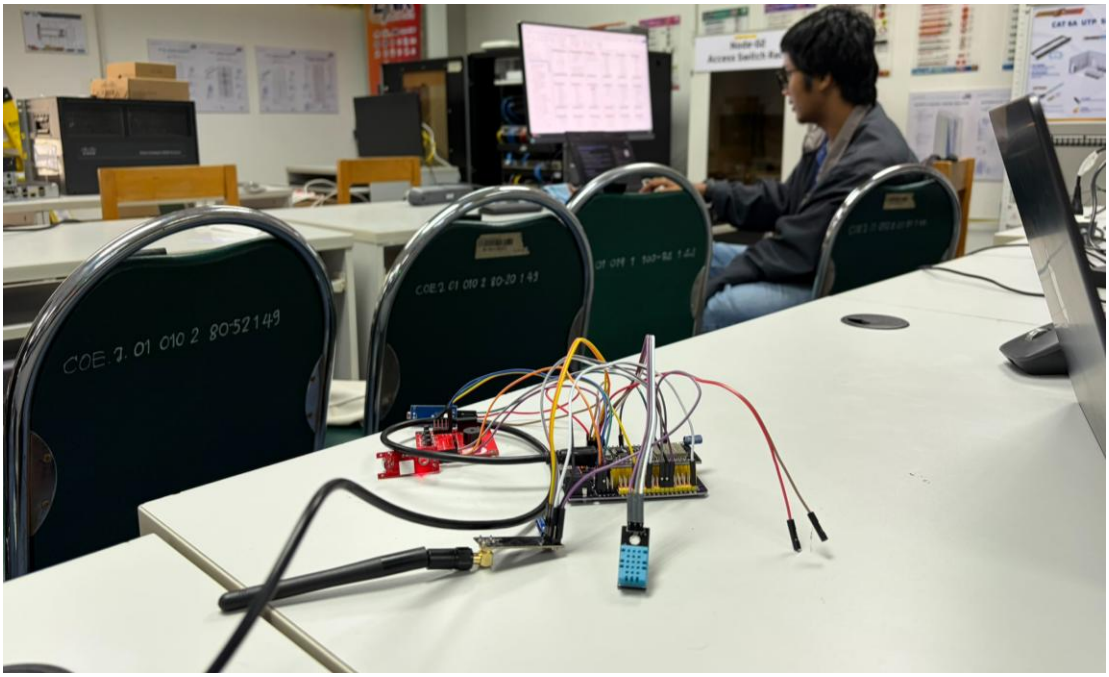
5.1 ผลการทดลอง

จากการนำระบบ Smart CoE Monitoring ไปติดตั้งและทดสอบการใช้งานจริงในพื้นที่อาคารภาควิชาวิศวกรรมคอมพิวเตอร์ พบว่าฮาร์ดแวร์และซอฟต์แวร์สามารถทำงานร่วมกันได้อย่างสมบูรณ์ตามวัตถุประสงค์ โดยมีผลการดำเนินการดังนี้:

- **การติดตั้งและรวบรวมข้อมูลในพื้นที่จริง:** จากการนำโหนดเซ็นเซอร์ (Multimodal Board) ไปติดตั้งในแต่ละพื้นที่เป้าหมาย พบว่าอุปกรณ์สามารถอ่านค่าอุณหภูมิ ความชื้น ระดับแสง และระดับเสียงได้อย่างถูกต้องตามสภาพแวดล้อมจริง โดยมีรายละเอียดการติดตั้งและการสื่อสารดังนี้:
    - **พื้นที่ห้อง R200 และ R303:** บอร์ดเซ็นเซอร์ที่ติดตั้งในพื้นที่ดังกล่าว (ดังรูปที่ 15, 16) สามารถรวบรวมข้อมูลและส่งผ่านคลื่นวิทยุด้วยโปรโตคอล nRF24L01 ไปยัง Sensor Gateway ประจำชั้นได้อย่างมีประสิทธิภาพ

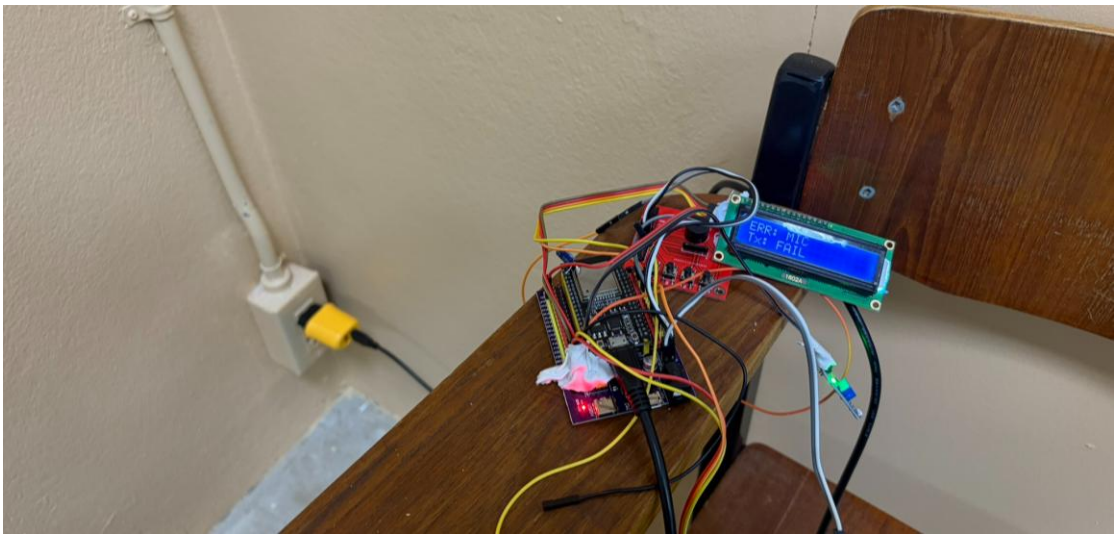


รูป 15 Multimodal Board ที่ห้อง R200



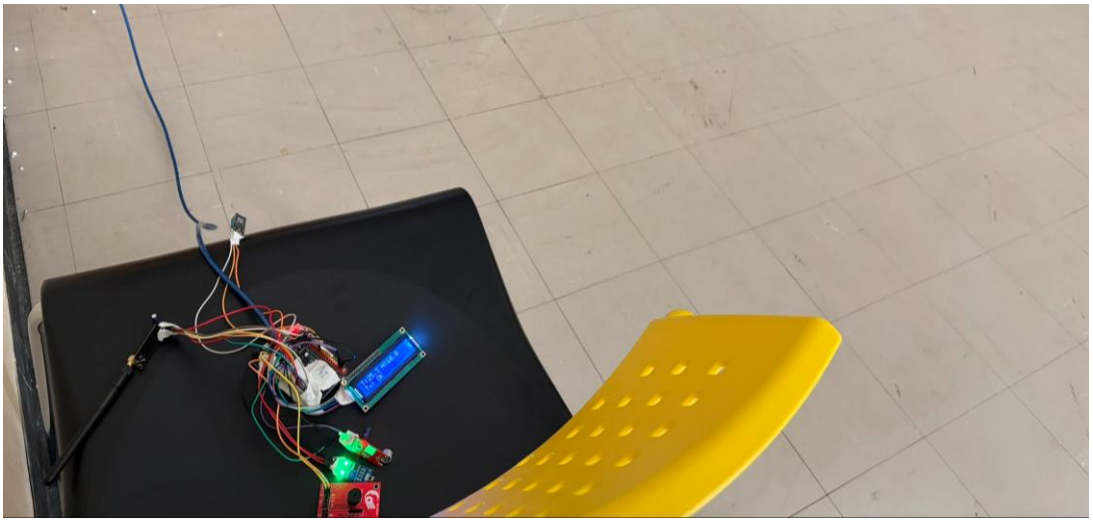
รูป 16 Multimodal Board ที่ห้อง R303

- **พื้นที่ห้อง R201:** โหนดเซ็นเซอร์ (ดังรูปที่ 17) สามารถสื่อสารผ่านโปรโตคอล ESP-NOW เพื่อส่งข้อมูลตรงไปยัง Sensor Gateway ชั้น 2 ได้ตามที่ออกแบบไว้ โดยไม่ต้องอาศัยโมดูลเสริม



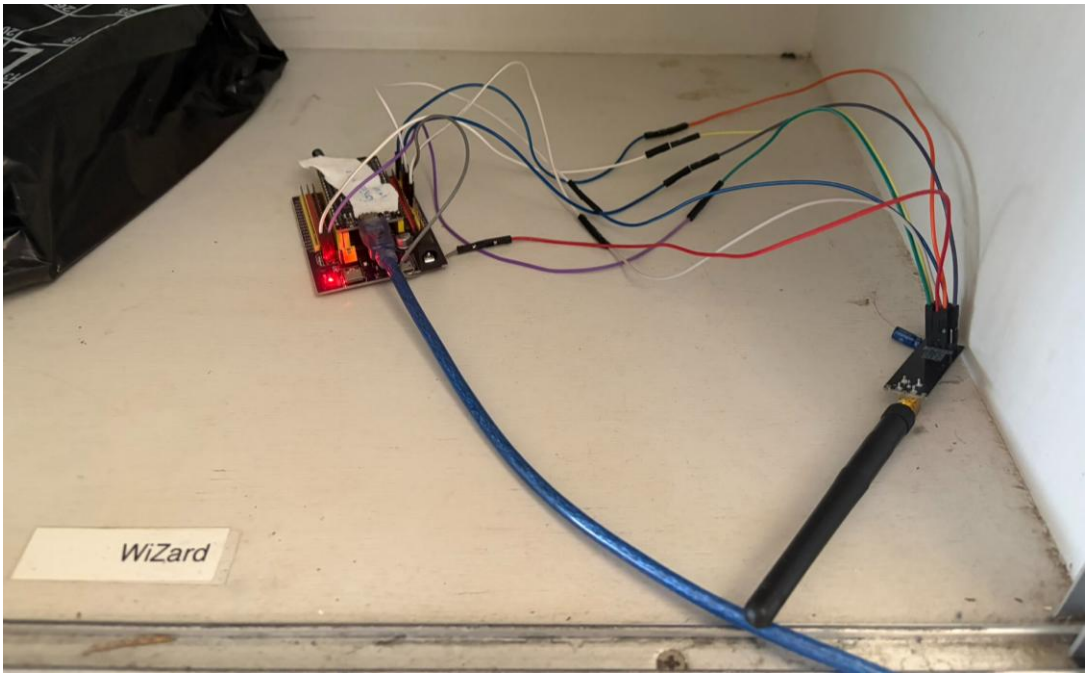
รูป 17 Multimodal Board ที่ห้อง R201

- **พื้นที่ AI Co-Working Space (Co_Ai):** บอร์ดเซ็นเซอร์ทำงานควบคู่กับโมดูลกล้อง ESP-CAM (ดังรูปที่ 18) โดยสามารถตรวจวัดสภาพแวดล้อมและส่งคำร้องขอ Presigned URL เพื่ออัปโหลดภาพถ่ายตรงเข้าสู่ระบบ Amazon S3 ได้สำเร็จโดยไม่ต้องผ่านเกตเวย์

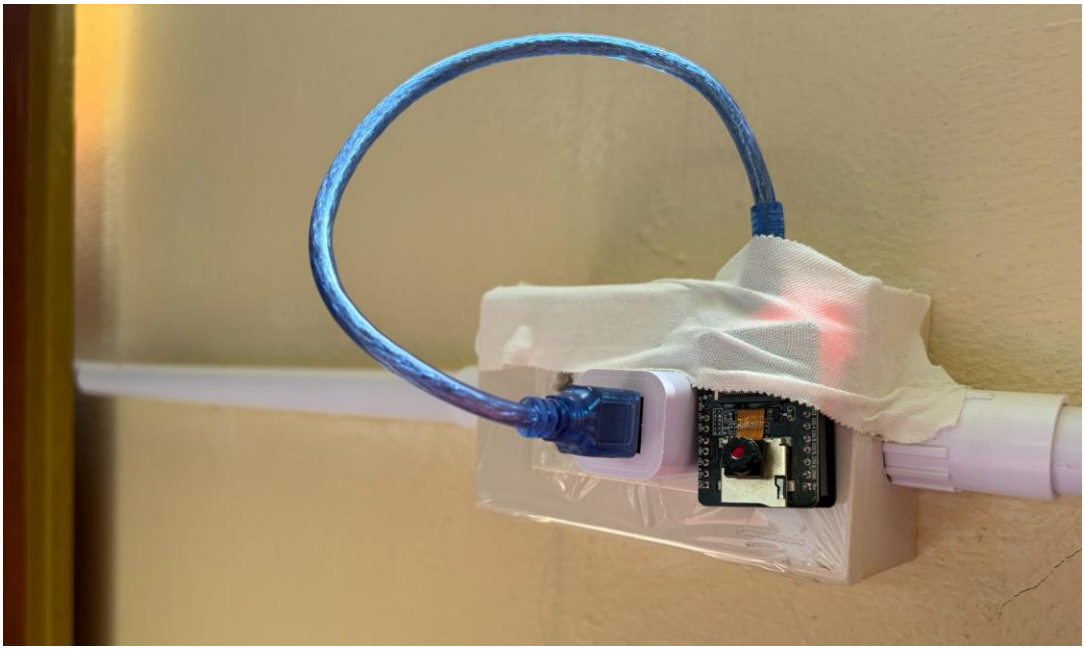


รูป 18 Multimodal Board ที่ห้อง AI Co-Working Space

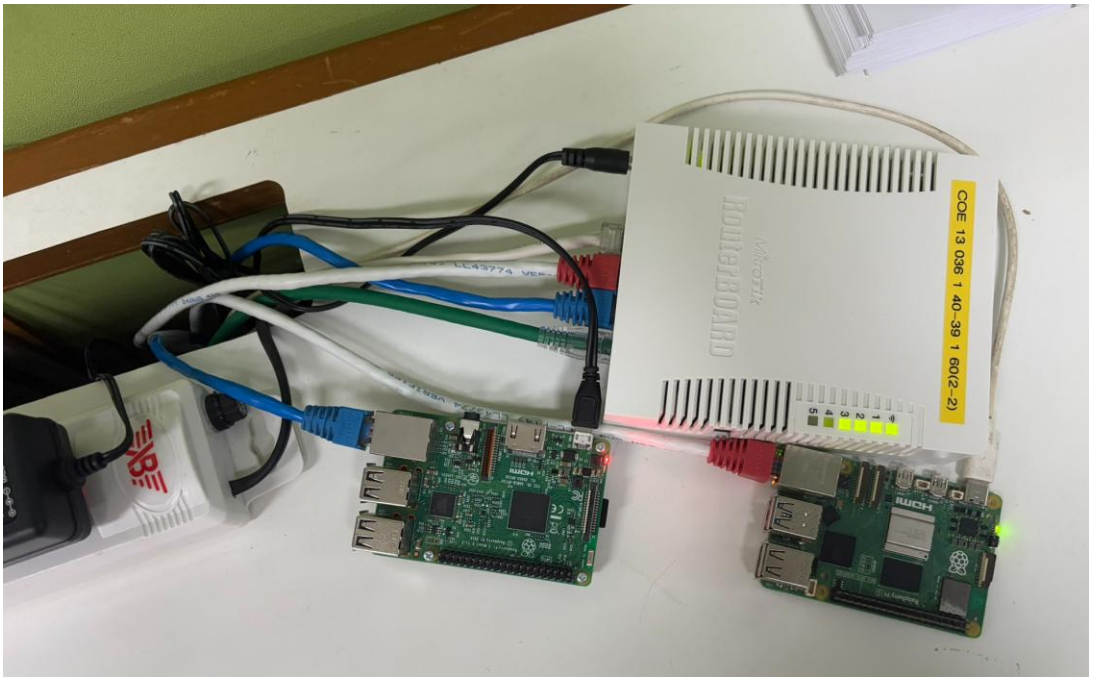
- **พื้นที่ศูนย์กลางข้อมูล (ห้อง PUPA และ CoE Co-Working Space):** การติดตั้งอุปกรณ์ Dual Edge Gateway และ Sensor Gateway (ดังรูปที่ 19, 20, 21) สามารถทำหน้าที่รับข้อมูลจากโหนดปลายทาง แปลงเป็นโปรโตคอล MQTT และจัดส่งขึ้นคลาวด์ได้อย่างสมบูรณ์



รูป 19 Sensor Gateway ที่ห้อง CoE Co-Working Space

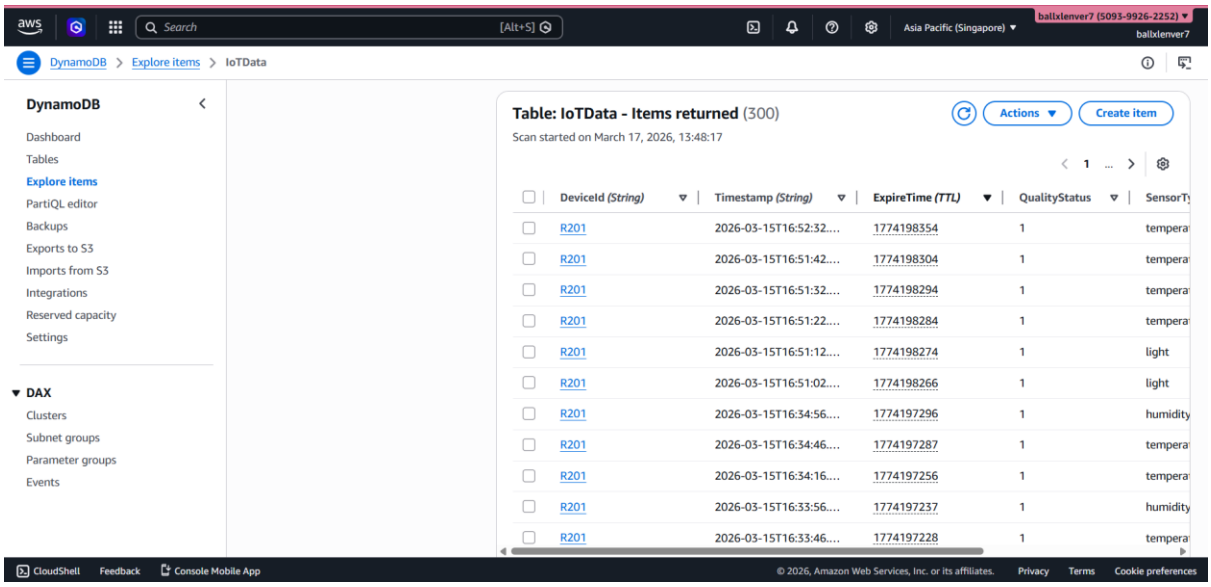


รูป 20 ESP-CAM ที่ห้อง AI Co-Working Space

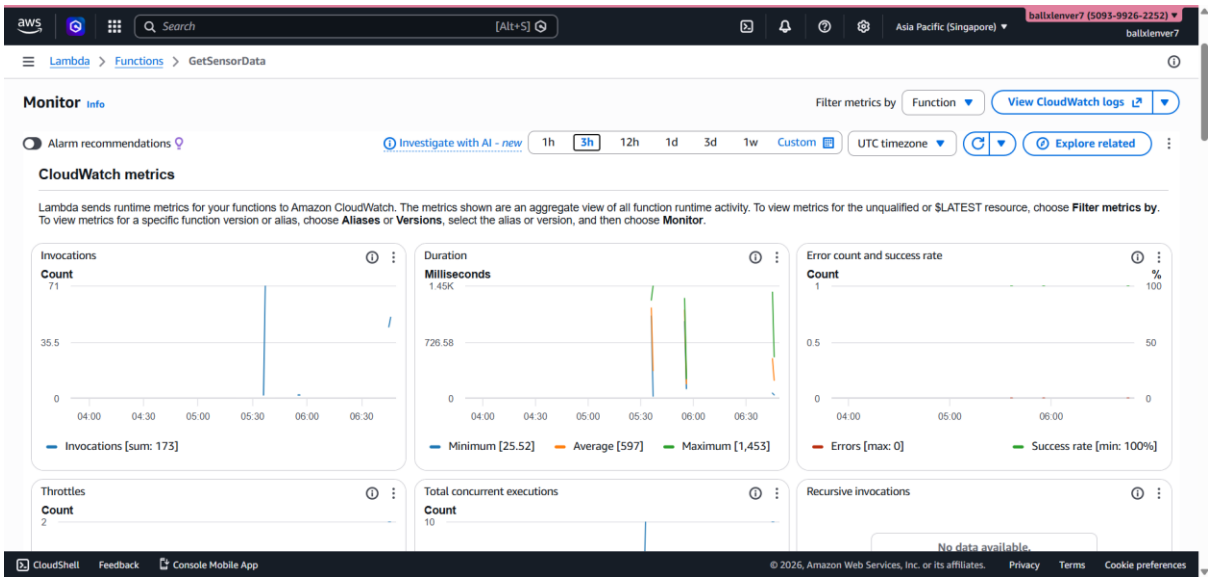


รูป 21 Edge Gateway ทั้งสอบที่ห้อง PUPA

- **การประมวลผลและการจัดเก็บข้อมูล:** ข้อมูล MQTT จาก Edge Gateway ถูกส่งขึ้นไปยัง AWS IoT Core ผ่านช่องทางที่เข้ารหัส (mTLS) ได้อย่างปลอดภัย และฟังก์ชัน AWS Lambda สามารถทำงานตามกฎที่ตั้งไว้ เพื่อบันทึกข้อมูลลงฐานข้อมูล Amazon DynamoDB ได้อย่างถูกต้อง พร้อมทั้งระบบสามารถล้างข้อมูลเก่าโดยอัตโนมัติตามระยะเวลา (TTL) 7 วันที่กำหนดไว้

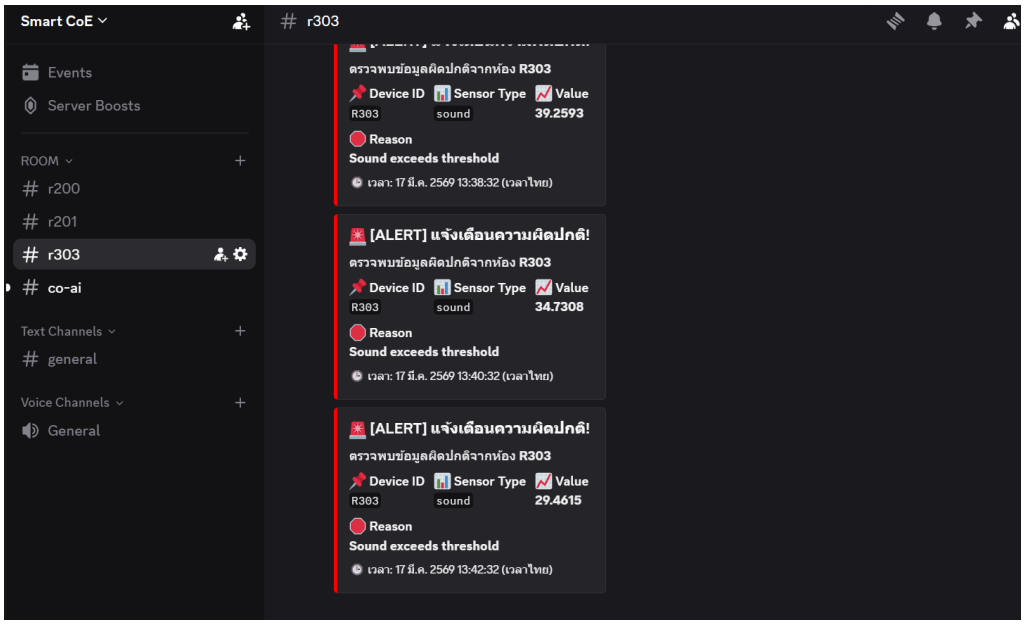


รูป 22 ข้อมูลบน DynamoDB ของตารางที่ชื่อว่า IoTData



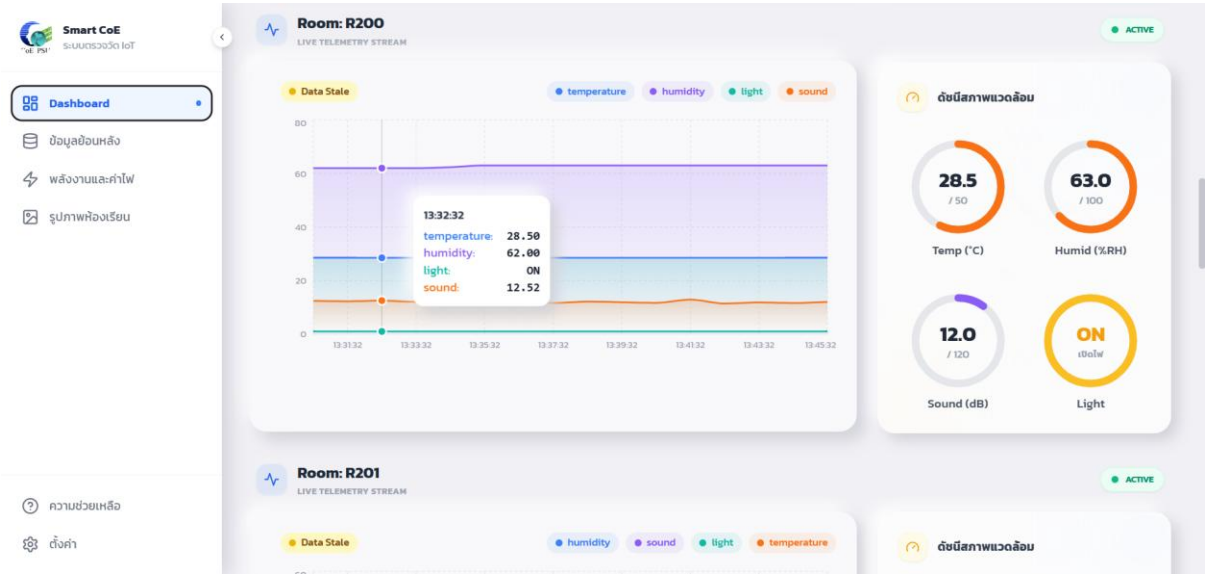
รูป 23 Metrics การทำงานของ GetSensorData บน AWS Lambda

- **ระบบการแจ้งเตือนอัตโนมัติ:** เมื่อเซ็นเซอร์ในพื้นที่ใดๆ ตรวจพบค่าสภาพแวดล้อมที่ผิดปกติหรือเกินกว่าเกณฑ์ (Threshold) ที่ผู้ดูแลระบบได้กำหนดไว้ผ่านเว็บแอปพลิเคชันสำหรับการตั้งค่าระบบ ฟังก์ชัน Lambda จะทำงานร่วมกับ Webhook เพื่อส่งข้อความแจ้งเตือนเข้าสู่แอปพลิเคชัน Discord ได้ทันที โดยสามารถระบุรายละเอียดอุปกรณ์ ห้อง ค่าที่ผิดปกติ และเวลาเกิดเหตุได้อย่างแม่นยำ



รูป 24 การแจ้งเตือนบน Discord

- **การแสดงผลบนส่วนติดต่อผู้ใช้งาน:** หน้าแดชบอร์ดสามารถแสดงผลข้อมูลเซ็นเซอร์จากบอร์ดที่ติดตั้งในทุกห้องได้แบบเรียลไทม์โดยปราศจากความหน่วง ผ่านการเชื่อมต่อ WebSockets รวมถึงสามารถแสดงรายการรูปภาพจากห้อง AI Co-Working Space ดึงประวัติข้อมูลย้อนหลัง และคำนวณประเมินค่าใช้จ่ายพลังงานไฟฟ้าได้อย่างถูกต้อง



รูป 25 แดชบอร์ดการแสดงผลบนเว็บไซต์

5.2 ความเสถียร

ระบบเราถูกออกแบบโดยให้ความสำคัญกับความพร้อมใช้งานสูงเพื่อลดความเสี่ยงจากการสูญหายของข้อมูล จุดเด่นสำคัญด้านความเสถียรคือการใช้สถาปัตยกรรมแบบ Dual Edge Gateway ที่ประกอบด้วย Raspberry Pi 5 เป็นเกตเวย์หลัก และ Raspberry Pi 3 เป็นเกตเวย์สำรอง โดยทั้งสองอุปกรณ์มีกลไกตรวจสอบสถานะการทำงานระหว่างกันตลอดเวลา หากเกตเวย์หลักเกิดข้อขัดข้อง เกตเวย์สำรองจะทำหน้าที่รับช่วงต่อในการส่งข้อมูลขึ้นระบบคลาวด์ทันทีซึ่งช่วยขจัดปัญหาจุดล้มเหลวจุดเดียวและทำให้ระบบสามารถเฝ้าระวังสภาพแวดล้อมได้อย่างต่อเนื่องไม่สะดุด

ในด้านฮาร์ดแวร์และเครือข่าย ระบบได้เพิ่มความเสถียรของการสื่อสารไร้สายด้วยการติดตั้งตัวเก็บประจุขนาด 10μF ให้กับโมดูล nRF24L01+ เพื่อช่วยกรองสัญญาณรบกวนทางไฟฟ้า ควบคู่ไปกับการเลือกใช้โปรโตคอลการสื่อสารให้เหมาะสมกับระยะทางของแต่ละพื้นที่ เช่น การนำโปรโตคอล ESP-NOW มาใช้กับห้องที่อยู่ใกล้กับเกตเวย์ อย่างไรก็ตาม จากการทำงานในสภาพแวดล้อมจริงพบว่า โหนดเซ็นเซอร์ในห้องที่อยู่ห่างไกลจาก Sensor Gateway ยังคงพบปัญหาข้อมูลสูญหาย เกิดขึ้นบ้างในบางครั้ง ซึ่งเป็นผลมาจากข้อจำกัดด้านระยะทางและการลดทอนสัญญาณจากผนังอาคาร อย่างไรก็ตามในภาพรวมระบบยังคงได้รับการบริหารจัดการความเสถียรผ่านการแยกวงเครือข่ายสำหรับอุปกรณ์ IoT โดยเฉพาะด้วย MikroTik Router ซึ่งช่วยป้องกันปัญหาความหนาแน่นและลดการรบกวนจากทราฟฟิกอินเทอร์เน็ตทั่วไปภายในอาคารได้เป็นอย่างดี

สำหรับการจัดการข้อมูลและการทำงานของระบบคลาวด์ ซอฟต์แวร์บนเกตเวย์ถูกพัฒนาให้รองรับกลไก MQTT Failover โดยสามารถสลับไปเชื่อมต่อกับเซิร์ฟเวอร์สำรองได้อัตโนมัติหากเซิร์ฟเวอร์หลักไม่สามารถให้บริการได้ รวมถึงการเลือกใช้สถาปัตยกรรมโครงสร้างพื้นฐานแบบ Serverless บนแพลตฟอร์ม AWS (เช่น AWS IoT Core, Lambda และ DynamoDB) ซึ่งไม่เพียงแต่ช่วยลดภาระการบำรุงรักษาเซิร์ฟเวอร์แบบเดิม แต่ยังให้ความเสถียรระดับสูงในการรองรับและประมวลผลข้อมูลปริมาณมหาศาลที่ถูกส่งเข้ามาตลอด 24 ชั่วโมงได้อย่างมีประสิทธิภาพ

6. ผลลัพธ์และอภิปราย

ผลการทดสอบแสดงให้เห็นว่าระบบสามารถทำงานร่วมกันในทุกส่วนได้อย่างสมบูรณ์ โดยโหนดเซ็นเซอร์ทั้งหมดสามารถตรวจวัดค่าสภาพแวดล้อมและส่งข้อมูลผ่านโครงข่ายไร้สาย nRF24L01+ และ ESP-NOW เข้าสู่ Sensor Gateway ประจำชั้นได้อย่างถูกต้อง จากนั้นข้อมูลได้ถูกส่งต่อมายัง Dual Edge Gateway เพื่อส่งขึ้นไปประมวลผลบนระบบคลาวด์ AWS ซึ่งระบบสามารถทำการแจ้งเตือนความผิดปกติผ่านแอปพลิเคชัน Discord ได้แบบเรียลไทม์ รวมทั้งเว็บแอปพลิเคชันที่พัฒนาขึ้นสามารถแสดงผลข้อมูลเซ็นเซอร์ ประวัติย้อนหลัง ภาพถ่ายภายในห้อง และประเมินค่าใช้จ่ายพลังงานไฟฟ้าได้อย่างแม่นยำ

จากผลการดำเนินงานและการทดสอบระบบในสถานที่จริง ผู้จัดทำมีประเด็นอภิปรายผลดังนี้:

- **ความเสถียรของสถาปัตยกรรมระบบ:** การออกแบบเครือข่ายแบบลำดับชั้นร่วมกับการใช้สถาปัตยกรรม Dual Edge Gateway ที่มีการทำ High Availability (HA) ระหว่างอุปกรณ์ Raspberry Pi ช่วยลดภาระการประมวลผลของระบบคลาวด์และสามารถแก้ปัญหาจุดล้มเหลวจุดเดียวได้อย่างชัดเจน ทำให้ระบบมีความเสถียรและพร้อมใช้งานตลอดเวลา
- **ข้อจำกัดของสัญญาณไร้สายทางกายภาพ:** จากการนำไปติดตั้งจริงพบว่า คลื่นความถี่ 2.4 GHz ที่ระบบเลือกใช้ มีข้อจำกัดเมื่อต้องส่งสัญญาณผ่านสิ่งกีดขวางที่เป็นผนังอาคารหรือระยะทางที่ไกลเกินไป ส่งผลให้เกิดปัญหาข้อมูลหล่นหายในโหนดเซ็นเซอร์บางจุด ซึ่งชี้ให้เห็นว่าในอนาคตอาจต้องพิจารณาเพิ่มอุปกรณ์ทวนสัญญาณ หรือเปลี่ยนไปใช้เทคโนโลยีการสื่อสารระยะไกลเพื่อแก้ปัญหานี้
- **การบริหารจัดการค่าใช้จ่ายบนคลาวด์:** เนื่องจากบริการคลาวด์มีค่าใช้จ่าย ผู้จัดทำจึงได้ออกแบบระบบให้จัดเก็บข้อมูลใน Amazon DynamoDB เพียง 7 วัน (ผ่านการตั้งค่า TTL) และย้ายข้อมูลประวัติที่เก่ากว่านั้นไปไว้ใน Amazon S3 ซึ่งผลลัพธ์แสดงให้เห็นว่าเป็นวิธีการที่ช่วยลดต้นทุนการจัดเก็บข้อมูลในระยะยาวได้อย่างมีประสิทธิภาพและเหมาะสมกับข้อจำกัดด้านงบประมาณ
- **ความปลอดภัยของเครือข่าย IoT:** การแยกวงเครือข่ายโดยใช้ MikroTik Router เพื่อให้อุปกรณ์ IoT ทำงานในระบบปิด และอนุญาตให้เชื่อมต่อได้เฉพาะผ่าน Port Forwarding ที่กำหนด ถือเป็นมาตรการที่ช่วยป้องกันการถูกโจมตีทางไซเบอร์จากเครือข่ายภายนอกได้เป็นอย่างดี

7. ปัญหาและข้อจำกัด

จากการติดตั้งและเปิดใช้งานระบบ Smart CoE Monitoring ในสภาพแวดล้อมจริง พบปัญหาและข้อจำกัดบางอย่างของระบบที่พัฒนาขึ้น ดังนี้:

- **ข้อจำกัดด้านสิ่งกีดขวางและการส่งข้อมูล:** การสื่อสารข้อมูลไร้สายด้วยคลื่นความถี่ 2.4 GHz ทั้งในส่วนของโปรโตคอล nRF24L01+ และ ESP-NOW มีความไวต่อการถูกลดทอนสัญญาณเมื่อต้องทะลุผ่านผนังคอนกรีตหนาหรือเมื่อมีระยะห่างที่มากเกินไป ส่งผลให้โหนดเซ็นเซอร์ในบางพื้นที่ที่อยู่ห่างจาก Sensor Gateway ประจำชั้น เกิดปัญหาข้อมูลสูญหายในบางรอบของการส่งข้อมูล
- **ข้อจำกัดด้านความละเอียดของฮาร์ดแวร์และการจัดการพลังงาน:** โมดูลเซ็นเซอร์อุณหภูมิและความชื้นที่เลือกใช้ (รุ่น DHT11) มีความแม่นยำในระดับพื้นฐาน ซึ่งอาจมีข้อจำกัดหากนำไปประยุกต์ใช้ในพื้นที่ที่ต้องการความละเอียดอ่อนสูง นอกจากนี้ โหนดเซ็นเซอร์และเกตเวย์ทั้งหมดในปัจจุบันยังต้องพึ่งพาแหล่งจ่ายไฟหลักของอาคารโดยไม่มีระบบพลังงานสำรองทำให้ระบบจะสูญเสียความสามารถในการเฝ้าระวังและไม่สามารถส่งแจ้งเตือนได้ทันทีหากเกิดเหตุการณ์ไฟฟ้าดับ
- **ข้อจำกัดด้านการตอบสนองอัตโนมัติ:** ระบบในปัจจุบันถูกออกแบบมาสำหรับการเฝ้าระวังและการส่งเสียงแจ้งเตือนเป็นหลัก โดยยังไม่มีการเชื่อมต่อกับอุปกรณ์ควบคุมภายนอก เช่น โมดูล Relay หรือรีโมทอินฟราเรด ทำให้ระบบไม่สามารถดำเนินการแก้ไขปัญหาเบื้องต้นได้ด้วยตนเอง เช่น ไม่สามารถสั่งเปิดเครื่องปรับอากาศสำรองหรือพัดลมระบายอากาศได้เมื่ออุณหภูมิในห้องเซิร์ฟเวอร์สูงเกินวิกฤต

8. แนวทางพัฒนาต่อ

แม้ระบบจะสามารถแก้ปัญหาการตรวจสอบสภาพแวดล้อมได้อย่างมีประสิทธิภาพ แต่ยังมีช่องทางในการพัฒนาต่อยอดเพื่อให้ระบบมีความสมบูรณ์และตอบโจทย์การใช้งานระดับองค์กรมากยิ่งขึ้น ดังนี้:

1.**การใช้งานปัญญาประดิษฐ์และการวิเคราะห์เชิงทำนาย:** ข้อมูล Time-series ปริมาณมหาศาลที่ถูกจัดเก็บใน DynamoDB และ S3 สามารถนำมาใช้เป็นชุดข้อมูล (Dataset) สำหรับฝึกสอนโมเดล Machine Learning เพื่อทำ Predictive Maintenance เช่น การทำนายล่วงหน้าว่าระบบปรับอากาศในห้อง Server (R303) กำลังจะทำงานผิดปกติจากแนวโน้มอุณหภูมิที่ค่อยๆ สูงขึ้น หรือการวิเคราะห์พฤติกรรมการใช้พื้นที่ห้อง AI Co-Working Space เพื่อปรับปรุงประสิทธิภาพการใช้พลังงาน

2.การยกระดับความแม่นยำของฮาร์ดแวร์เซ็นเซอร์: เพื่อความแม่นยำขั้นสูงในการตรวจจับ ควรพิจารณาอัปเกรดเซ็นเซอร์อุณหภูมิและความชื้นไปใช้รุ่นที่มีความแม่นยำและทนทานสูงขึ้น (เช่น จาก DHT11 เป็น BME280 หรือ SHT31) รวมถึงการเพิ่มระบบพลังงานสำรอง (UPS หรือ Battery Module) ให้กับโหนดเซ็นเซอร์และ Gateway เพื่อให้ระบบสามารถเฝ้าระวังและแจ้งเตือนได้แม้ในกรณีที่ไฟฟ้าในอาคารดับ

3.การเพิ่มระบบตอบสนองอัตโนมัติ: ในปัจจุบันระบบทำได้เพียงการแจ้งเตือนผ่านเสียงและผ่านระบบแจ้งเตือน (AWS SES) การพัฒนาต่อยอดสามารถเพิ่มโมดูล Relay หรือ IR Blaster เข้าไปที่โหนด เพื่อให้ระบบสามารถสั่งการแก้ไขปัญหาได้อัตโนมัติ เช่น หากอุณหภูมิห้อง Server สูงเกินขีดวิกฤต ให้ระบบสั่งเปิดพัดลมระบายอากาศสำรอง หรือส่งสัญญาณอินฟราเรดไปสั่งเปิดเครื่องปรับอากาศตัวสำรองทันที

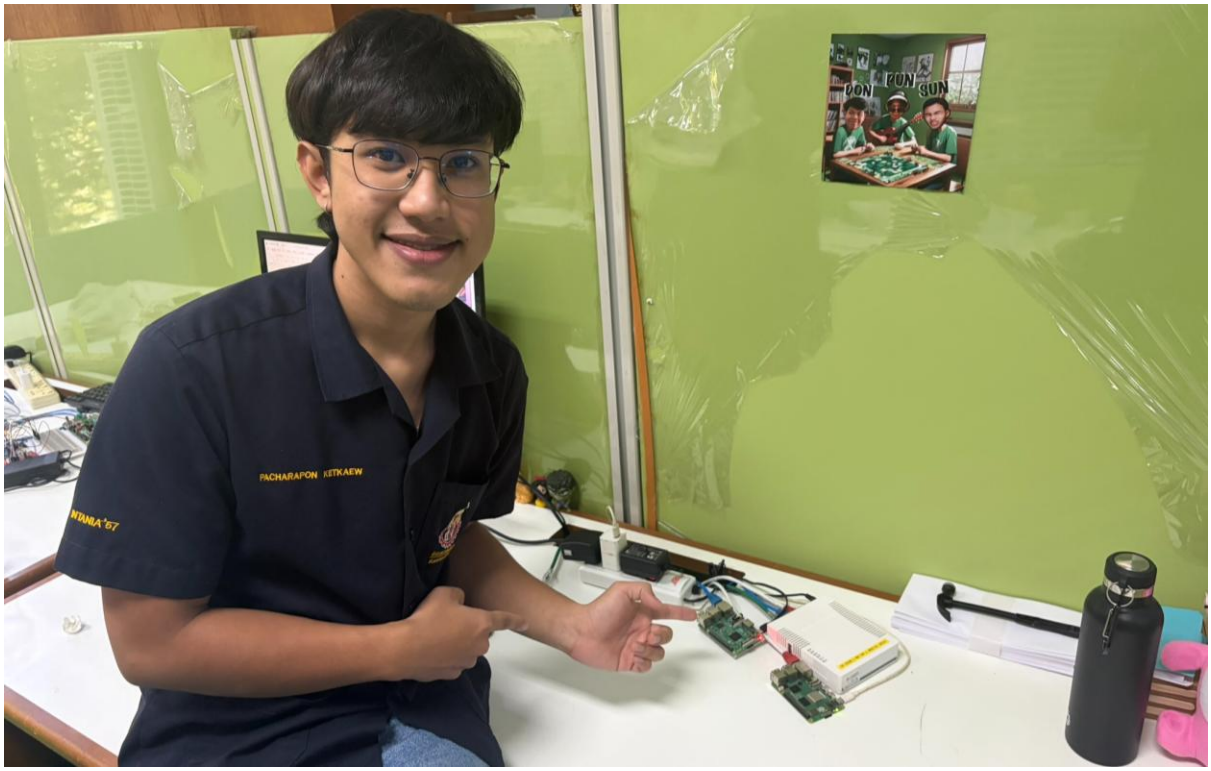
4.การขยายช่องทางการแจ้งเตือนแบบมัลติแชนเนล: นอกเหนือจากการแจ้งเตือนผ่านอีเมล (AWS SES) ควรมีการเชื่อมต่อ API เข้ากับแอปพลิเคชันสนทนาที่ผู้ดูแลระบบใช้งานเป็นประจำ เช่น LINE Notify, Slack หรือ Microsoft Teams เพื่อลดความหน่วงในการรับข่าวสารและทำให้การเข้าระงับเหตุทำได้รวดเร็วยิ่งขึ้น

9. บทบาทหน้าที่หลักของแต่ละคน

นายพชรพล เกตุแก้ว รหัสนักศึกษา 6610110190 รับผิดชอบด้านโครงสร้างพื้นฐานเซิร์ฟเวอร์ สถาปัตยกรรมคลาวด์ เครือข่าย และการพัฒนาส่วนติดต่อผู้ใช้งาน โดยมีรายละเอียดดังนี้:

- **Network & Infrastructure:** ออกแบบโครงสร้างเครือข่าย IoT แบบลำดับชั้น รองรับการสื่อสารหลายโปรโตคอล (nRF24L01, ESP-NOW, MQTT) พร้อมตั้งค่า Router, Port Forwarding และระบบ High Availability ระหว่าง Gateway เพื่อให้ระบบมีความเสถียรและรองรับการ Failover
- **Edge Gateway:** พัฒนาและดูแลระบบ Edge Processing ด้วย Raspberry Pi 5 (Primary) และ Pi 3 (Secondary) รองรับ MQTT, Data Validation และ Data Aggregation
- **AWS Cloud (Serverless):** ออกแบบระบบบน AWS IoT Core, Lambda, DynamoDB และ S3 พร้อมทำ Cost Optimization และ API สำหรับอุปกรณ์
- **Web Application:** พัฒนา Dashboard ด้วย Next.js แสดงผลแบบ Real-time และข้อมูลย้อนหลัง พร้อมระบบคำนวณค่าไฟ
- **Alert System:** พัฒนาระบบแจ้งเตือนอัตโนมัติผ่าน Discord Webhook เมื่อพบค่าผิดปกติ

รูป 26 บันทึกการทำงาน 1 (พชรพล)

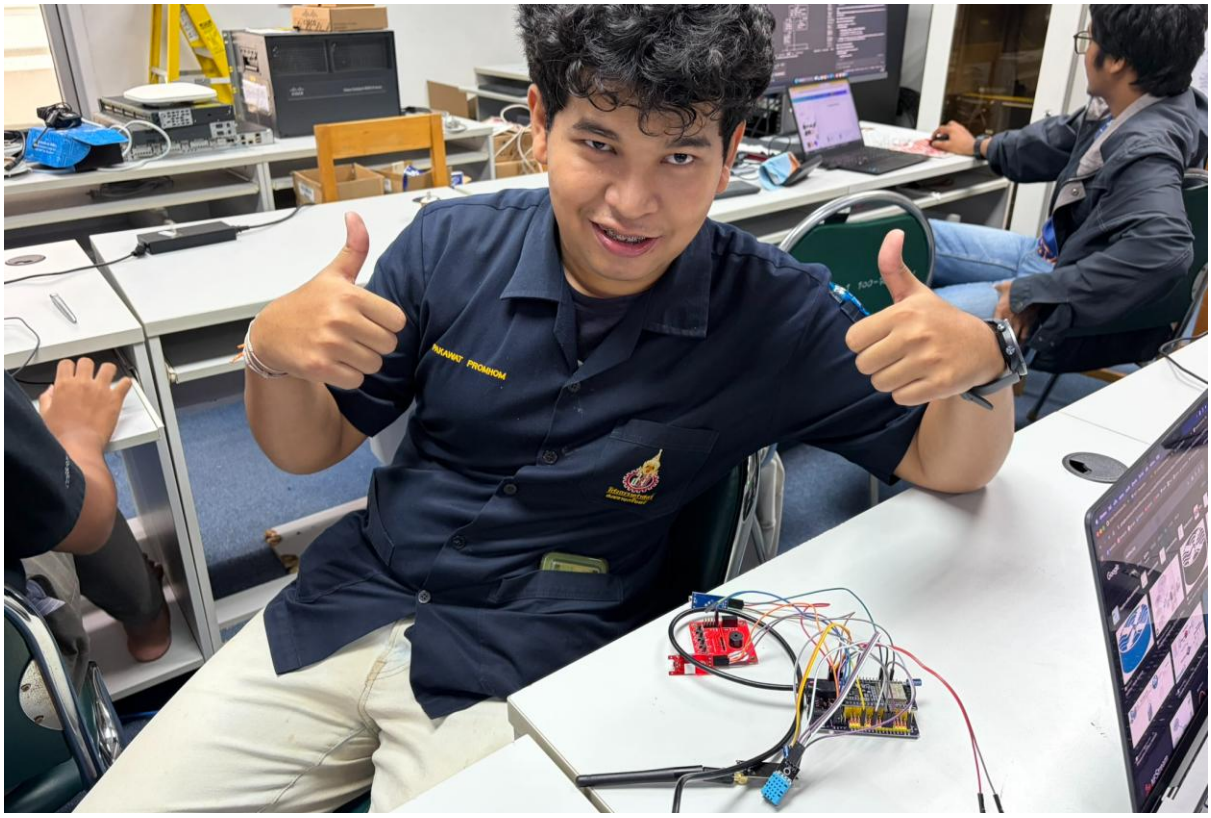


รูป 27 บันทึกการทำงาน 2 (พชรพล)

นายภควัฒน์ พรหมหอม รหัสนักศึกษา 6610110227 รับผิดชอบด้านฮาร์ดแวร์ระบบสื่อสารไร้สาย และการจัดการ Sensor Gateway โดยมีรายละเอียดดังนี้:

- **Hardware Design:** ออกแบบ Schematic และประกอบ Multimodal Board ด้วย ESP32 เชื่อมต่อเซ็นเซอร์ (DHT11/22, LDR, Sound) พร้อม LCD (I2C) และ Buzzer รวมถึงติดตั้ง ESP-CAM
- **Wireless Communication:** ออกแบบระบบสื่อสารแบบลำดับชั้นใน Network ปิด ใช้ nRF24L01+ (R200, R303, CoE Co-Working Space) และ ESP-NOW (R201)
- **Sensor Gateway & Network:** พัฒนาและติดตั้ง Gateway สำหรับรับข้อมูลและแปลงเป็น MQTT
- **MQTT Architecture:** ออกแบบโครงสร้าง Topic และ Payload ทั้งฝั่งรับ (sensors/{room}/{type}) และส่งคำสั่ง (เช่น threshold, cam, reset)

รูป 28 บันทึกการทำงาน 1 (ภควัฒน์)



รูป 29 บันทึกการทำงาน 2 (ภควัฒน์)

10. บทสรุป

โครงงาน Smart CoE Monitoring ได้บรรลุวัตถุประสงค์ในการพัฒนาระบบเฝ้าระวังสภาพแวดล้อมภายในอาคารภาควิชาวิศวกรรมคอมพิวเตอร์แบบครบวงจร ระบบถูกออกแบบโดยผสานเทคโนโลยี Internet of Things (IoT) ตั้งแต่อุปกรณ์ฮาร์ดแวร์ปลายทาง สถาปัตยกรรมเครือข่าย ไปจนถึงระบบคลาวด์เข้าด้วยกัน ทำให้สามารถตรวจวัดค่าอุณหภูมิ ความชื้น แสง เสียง และถ่ายภาพสภาพแวดล้อมเพื่อนำมาวิเคราะห์และแสดงผลแบบเรียลไทม์ได้อย่างมีประสิทธิภาพ

จุดเด่นสำคัญของโครงงานคือการออกแบบสถาปัตยกรรมเครือข่ายแบบลำดับชั้นที่เลือกประยุกต์ใช้โปรโตคอลการสื่อสาร (nRF24L01+ และ ESP-NOW) ให้เหมาะสมกับสภาพพื้นที่ รวมทั้งการสร้างสถาปัตยกรรมศูนย์กลางข้อมูลแบบ Dual Edge Gateway ที่มีกลไก High Availability (HA) ซึ่งช่วยขจัดปัญหาจุดล้มเหลวจุดเดียว ทำให้โครงข่ายมีความเสถียรและสามารถรับส่งข้อมูลได้อย่างต่อเนื่อง นอกจากนี้การประยุกต์ใช้โครงสร้างพื้นฐานแบบ Serverless บน AWS (IoT Core, Lambda, DynamoDB และ S3) ยังช่วยยกระดับประสิทธิภาพการประมวลผล ช่วยบริหารจัดการต้นทุนการจัดเก็บข้อมูล และทำให้ระบบสามารถส่งข้อความแจ้งเตือนความผิดปกติไปยังแอปพลิเคชัน Discord เพื่อให้ผู้ดูแลระบบรับรู้และระงับเหตุได้อย่างทันท่วงที

แม้ในการนำไปติดตั้งใช้งานจริงจะพบข้อจำกัดทางกายภาพบางประการ เช่น การลดทอนสัญญาณของคลื่นความถี่ 2.4 GHz เมื่อทะลุผ่านสิ่งกีดขวางจนทำให้เกิดปัญหาข้อมูลสูญหายในบางโหนด ตลอดจนข้อจำกัดของการไม่มีระบบพลังงานสำรอง แต่ข้อค้นพบเหล่านี้ล้วนเป็นข้อมูลสำคัญที่นำไปสู่การกำหนดแนวทางการพัฒนาในอนาคต เช่น การอัปเกรดเซ็นเซอร์ การขยายความสามารถด้วยปัญญาประดิษฐ์เพื่อทำ Predictive Maintenance และการเพิ่มอุปกรณ์ให้ระบบสามารถตอบสนองต่อปัญหาได้โดยอัตโนมัติ

โดยสรุป โครงงานนี้ถือเป็นความสำเร็จในการนำเทคโนโลยี IoT มาแก้ปัญหาข้อจำกัดของการตรวจสอบสภาพแวดล้อมแบบเดิม สามารถลดความเสี่ยงต่อการเกิดความเสียหายของอุปกรณ์ราคาสูงภายในห้องปฏิบัติการ และสร้างรากฐานของระบบติดตามอัจฉริยะที่สามารถปรับขยายและยกระดับประสิทธิภาพการบริหารจัดการทรัพยากรของภาควิชาฯ ได้อย่างยั่งยืน