บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ตามพระราชบัญญัติการศึกษาแห่งชาติ พ.ศ. 2542 และที่แก้ไขเพิ่มเติม ได้กำหนดแนวทางในการจัดการศึกษาที่ต้องมุ่งเน้นการฝึกทักษะ กระบวนการคิด การจัดการ และการประยุกต์ความรู้ด้านเทคโนโลยีสารสนเทศเพื่อรองรับการพัฒนาประเทศ โดยเฉพาะการเตรียมความพร้อมกำลังคนเข้าสู่ภาคอุตสาหกรรมในยุคดิจิทัล ดังนั้น การจัดทำโครงงานระบบ IIoT สำหรับติดตามค่า OEE จึงเป็นการฝึกทักษะการประยุกต์เทคโนโลยีสารสนเทศและระบบอัตโนมัติ ให้สอดคล้องกับการเตรียมกำลังคนสู่ภาคอุตสาหกรรมในยุคดิจิทัล

ทว่าในสภาพความเป็นจริงของภาคการผลิตในปัจจุบัน โดยเฉพาะกลุ่มวิสาหกิจขนาดกลางและขนาดย่อม (SME) การประเมินประสิทธิภาพเครื่องจักรยังคงพึ่งพากระบวนการแบบดั้งเดิม (Manual) ที่ให้พนักงานหน้างานจดบันทึกยอดการผลิตและเวลาที่เครื่องจักรหยุดทำงานลงบนกระดาษ สภาพปัจจุบันนี้ทำให้เกิดปัญหาความล่าช้าของข้อมูล (Data Latency) และมีความคลาดเคลื่อนสูงจากความผิดพลาดของมนุษย์ (Human Error) ซึ่งไม่สอดคล้องกับเจตนารมณ์ในการพัฒนาอุตสาหกรรมด้วยเทคโนโลยีสมัยใหม่

สภาพที่ต้องการและมีความมุ่งหวังในยุคอุตสาหกรรม 4.0 คือการเปลี่ยนผ่านกระบวนการผลิตไปสู่ "โรงงานอัจฉริยะ (Smart Factory)" ที่เครื่องจักรและระบบคอมพิวเตอร์สามารถเชื่อมโยงข้อมูลกันได้อย่างอัตโนมัติ (Automated Data Pipeline) การวัดค่าประสิทธิผลโดยรวมของเครื่องจักร หรือค่า OEE (Overall Equipment Effectiveness) ซึ่งเป็นดัชนีชี้วัดระดับสากล จะต้องสามารถคำนวณและแสดงผลได้แบบใกล้เวลาจริง (Near Real-time Monitoring) โดยอัปเดตข้อมูลประมาณทุก 1 วินาที เพื่อให้ผู้ควบคุมงานและวิศวกรสามารถมองเห็นสถานะหน้างานได้อย่างโปร่งใสและนำข้อมูลไปใช้วางแผนการผลิตได้อย่างมีประสิทธิภาพ

สาเหตุสำคัญที่ทำให้ภาคอุตสาหกรรม SME ไม่สามารถบรรลุความมุ่งหวังดังกล่าวได้ เกิดจากระบบการจัดเก็บข้อมูลและประมวลผลเชิงพาณิชย์ในท้องตลาดมีต้นทุนในการจัดหาและการติดตั้งค่อนข้างสูง ซึ่งเป็นอุปสรรคทางการเงินที่สำคัญ หากปล่อยปัญหานี้ทิ้งไว้ โรงงานอุตสาหกรรม จะไม่สามารถรับรู้ถึง "ความสูญเปล่าที่มองไม่เห็น" เช่น การหยุดชะงักย่อยของสายพานลำเลียง (Minor Stoppage) หรืออัตราการเกิดชิ้นงานเสียที่สะสมในแต่ละวัน ส่งผลให้ต้นทุนการผลิต บานปลาย

สูญเสียวัตถุดิบโดยเปล่าประโยชน์ และทำให้อุตสาหกรรมไทยสูญเสียขีดความสามารถ ในการแข่งขัน

คณะผู้จัดทำจึงได้เลือกวิธีการพัฒนาระบบเทคโนโลยีอินเทอร์เน็ตของอุตสาหกรรม (IIoT) บนสถาปัตยกรรมแบบแยกส่วน (Decoupling Architecture) โดยแยกงานนับข้อมูลที่ ESP32 และงานคำนวณ OEE ที่ Node-RED เพื่อให้ปรับสูตรและแสดงผลได้โดยไม่ต้องแก้ firmware บ่อย การออกแบบและสร้าง "โมเดลสายพานลำเลียงเพื่อการประเมินประสิทธิภาพเครื่องจักรและแสดงผลแบบใกล้เวลาจริง" ขึ้นมาเป็นต้นแบบสำหรับการศึกษา โครงงานนี้เลือกใช้ฮาร์ดแวร์ต้นทุนต่ำอย่างบอร์ด ESP32 Development Board ร่วมกับระบบเซนเซอร์อินฟราเรด 3 จุดแยกอิสระ (Total, Good, Waste) ส่งข้อมูลผ่านโปรโตคอล MQTT (HiveMQ Cloud) ควบคู่กับซอฟต์แวร์ประมวลผลลอจิก Node-RED เพื่อจัดเก็บลงฐานข้อมูล Influx DB v2 และแสดงผลผ่าน Grafana Dashboard การดำเนินการโครงงานนี้นอกจากจะสร้างต้นแบบราคาประหยัดให้กับอุตสาหกรรมขนาดเล็กแล้ว ยังช่วยให้คณะผู้จัดทำเกิดทักษะความเชี่ยวชาญด้านวิศวกรรมข้อมูล (Data Engineering) และระบบเครือข่าย ซึ่งเป็นประโยชน์ต่อการนำไปต่อยอดในระดับวิชาชีพต่อไป

1.2 วัตถุประสงค์

1. เพื่อออกแบบและประกอบชุดจำลองสายพานการผลิต (Mini Conveyor Prototype) ที่สามารถนับจำนวนชิ้นงานทั้งหมด Total/Good นับอัตโนมัติ ส่วน Waste ผ่าน Manual QC (พนักงานจับชิ้นงานเสียให้โดนเซนเซอร์กลาง) เพื่อพัฒนาระบบส่งข้อมูลแบบเรียลไทม์ (Real-time Data Pipeline) จากบอร์ด ESP32 ผ่านโปรโตคอล MQTT บนคลาวด์ ไปยังฐานข้อมูลอนุกรมเวลา (Time-Series Database)
2. เพื่อสร้างระบบสมองกลคำนวณและหน้าจอแสดงผล (Logic Engine & Dashboard) ผ่านซอฟต์แวร์ Node-RED และ Grafana ที่สามารถประเมินสถานะเครื่องจักรและคำนวณค่า OEE ได้อัตโนมัติ

1.3 ขอบเขตของโครงงาน

1.3.1 ขอบเขตด้านประชากรและกลุ่มตัวอย่าง (Target Group Scope)

- **กลุ่มตัวอย่างเชิงระบบ (System Testing):** ทำการทดสอบประสิทธิภาพของเซนเซอร์โดยการปล่อยชิ้นงานผ่านสายพานจำนวน 105 ชิ้น (แบ่งเป็น 3 รอบ รอบละ 35 ชิ้น) และทดสอบความเสถียรของระบบเครือข่าย โดยการเปิดระบบเชื่อมต่อทิ้งไว้ต่อเนื่องเป็นเวลา 1 ชั่วโมง (60 นาที)
- **กลุ่มตัวอย่างบุคคล (Expert Evaluation)**: ประเมินความพึงพอใจและประสิทธิภาพการใช้งานโดยผู้เชี่ยวชาญด้านเทคโนโลยีสารสนเทศหรือช่างเทคนิคอุตสาหกรรม จำนวน 3-5 ท่าน (ใช้วิธีเลือกแบบเจาะจง)

1.3.2 ขอบเขตด้านฮาร์ดแวร์และอุปกรณ์ (Edge Layer)

- **ชุดสายพานลำเลียงจำลอง:** ขับเคลื่อนด้วยมอเตอร์เกียร์ DC (ใช้โครงสร้างอะลูมิเนียมสำเร็จรูป) จำนวน 1 ชุด
- **บอร์ดประมวลผล:** ESP32 (รุ่น 38 Pin) จำนวน 1 บอร์ด
- **เซนเซอร์อินฟราเรด (IR Sensor) 3 จุด:**
    - จุดที่ 1 (ต้นสายพาน/ GPIO 2): ตรวจจับชิ้นงานทั้งหมด (Total Count)
    - จุดที่ 2 (จุดคัดออกโดยพนักงาน QC / GPIO 5): ติดตั้งบริเวณคานกลางสายพานสำหรับให้พนักงาน QC ทำการสแกนหรือนำชิ้นงานที่พบตำหนิมาผ่านหน้าเซนเซอร์ด้วยตนเอง เพื่อบันทึกเป็นชิ้นงานเสีย (Waste Count)
    - จุดที่ 3 (ปลายสายพาน/ GPIO 4): ตรวจจับชิ้นงานที่ผ่านกระบวนการสมบูรณ์ (Good Count)

1.3.3 ขอบเขตด้านซอฟต์แวร์และระบบเครือข่าย

- **Hardware Programming:** ภาษา C/C++ (Arduino IDE) โดยออกแบบลูปหลักให้ตรวจจับเซนเซอร์ด้วย millis() และ debounce เพื่อลดการบล็อกการทำงานของระบบ พร้อมขยาย MQTT Buffer Size เป็น 512 bytes เพื่อรองรับการเชื่อมต่อ TLS และลดโอกาสข้อมูลสูญหายจาก BufferOverflow
- **Message Broker:** โปรโตคอล MQTT ผ่านระบบ HiveMQ Cloud พร้อมมาตรฐานความปลอดภัย TLS/SSL (พอร์ต 8883)
- **Logic Engine:** ซอฟต์แวร์ Node-RED สำหรับรับข้อมูล (Subscribe), บริหารสถานะเครื่องจักร (Running/IDLE โดย Timeout 60 วินาที), และคำนวณสมการ OEE โดยอัปเดตค่าประมาณทุก 1 วินาที
- **Database:** ซอฟต์แวร์ InfluxDB v2 (Time-Series Database) ใช้ bucket ชื่อ conveyor_oee
- **Data Visualization:** ซอฟต์แวร์ Grafana ประมวลผลข้อมูลด้วยภาษา Flux Query แสดงผลแดชบอร์ดแบบใกล้เวลาจริง (Refresh ทุก 5 วินาที)

1.3.4 ขอบเขตด้านสถานที่ (Place Scope)

- แผนกวิชาเทคโนโลยีสารสนเทศ วิทยาลัยเทคนิคพัทลุง

1.3.5 ขอบเขตด้านเวลา (Time Scope)

- ดำเนินการศึกษาและจัดทำโครงงานตลอดภาคเรียนที่ 1 ปีการศึกษา 2569 เป็นระยะเวลา 11 สัปดาห์

1.4 แนวความคิดในการออกแบบโครงงาน

โครงงานนี้ถูกออกแบบตามโครงสร้างสถาปัตยกรรม IIoT (Industrial Internet of Things) มีแนวคิดการทำงานแบบ 4 เลเยอร์ ดังนี้:

1. Edge Node Layer: ESP32 อ่านค่าจากเซนเซอร์ IR 3 จุด เพื่อตรวจสอบการไหลของวัตถุดิบและส่งข้อมูลสถานะเข้าสู่คลาวด์
    - ในส่วนการตรวจจับชิ้นงานเสีย ระบบถูกออกแบบให้ทำงานร่วมกับพนักงานตรวจสอบคุณภาพ (Manual QC Station) โดยเมื่อพนักงานตรวจพบชิ้นงานที่ไม่ผ่านมาตรฐานบนสายพาน จะทำการหยิบชิ้นงานนั้นมานำผ่านเซนเซอร์จุดคัดออก

(จุดที่ 2) บริเวณกลางสายพาน เพื่อส่งสัญญาณนับยอด Waste Count เข้าสู่ระบบอัตโนมัติ

2. Communication Layer: ส่งข้อมูลข้อความ (String) โดยแยกตามหัวข้อ (Topic) ผ่าน HiveMQ Broker แบบใกล้เวลาจริง / Near Real-time ได้แก่
    - factory/conveyor/total
    - factory/conveyor/good
    - factory/conveyor/waste
3. Logic & Processing Layer: ใช้ Node-RED เป็นสมองกลในการวิเคราะห์ข้อมูลตามหลักการ OEE สากล ดังนี้:
    - **Machine Status:** หากเซนเซอร์ไม่มีการทำงานเกิน 60 วินาที จะปรับสถานะเป็น IDLE (STOPPED) อัตโนมัติ
    - **Availability (A):** เปรียบเทียบเวลาที่สายพานทำงานจริง (Running Time) กับเวลาเปิดระบบทั้งหมด (Planned Time / Total Time) ซึ่งในโครงงานนี้คำนวณจากเวลาตั้งแต่ Node-RED เริ่มทำงานจนถึงปัจจุบัน
    - **Performance (P)**: กำหนดเวลามาตรฐาน (Ideal Cycle Time) ที่ 5 วินาทีต่อชิ้น เปรียบเทียบกับความเร็วที่ผลิตได้จริง
    - **Quality (Q):** วัดสัดส่วนของชิ้นงานดี (Good Count) เทียบกับชิ้นงานทั้งหมด (Total Count)
4. Storage & Visualization Layer: ข้อมูลพิกัดเวลา (Time-Series) ถูกบันทึกลง Influx DB v2 และดึงมาสร้างกราฟิกมาตรวัด (Gauge) และกราฟแนวโน้ม (Trend) ใน Grafana เพื่อประเมินค่า OEE รวมตามสมการ:

$$OEE(\%) = \left(\frac{Availability}{100}\right) \times \left(\frac{Performance}{100}\right) \times \left(\frac{Quality}{100}\right) \times 100$$

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการทำโครงงาน

1. ได้ชุดโมเดลสายพานลำเลียงจำลองที่สามารถประเมินดัชนีชี้วัด OEE แบบอัตโนมัติตามมาตรฐานอุตสาหกรรม

2. สามารถตรวจสอบสถานะการทำงาน (RUNNING / IDLE หรือ STOPPED) และยอดการผลิตได้แบบเรียลไทม์ ช่วยลดปัญหาความล่าช้าและข้อผิดพลาดจากมนุษย์
3. คณะผู้จัดทำได้รับทักษะวิชาชีพชั้นสูงด้าน IIoT ครบทุกขั้นตอน ตั้งแต่ Hardware, Network/Cloud, Logic Scripting, และ Data Query
4. ได้ต้นแบบระบบการผลิตอัจฉริยะราคาประหยัด (Cost-Effective Blueprint) ที่มีความเสถียรสูง สามารถนำไปประยุกต์ใช้เพื่อยกระดับโรงงาน SME ในอนาคต

1.6 นิยามศัพท์เฉพาะ

คำศัพท์	ความหมาย
OEE	ดัชนีประสิทธิผลโดยรวมของเครื่องจักร
Availability	สัดส่วนเวลาที่เครื่องพร้อมทำงานจริง เทียบกับเวลาเปิดระบบ (Planned Time)
Performance	สมรรถนะการผลิตเทียบกับรอบเวลาอุดมคติ (Ideal Cycle Time = 5 วินาที/ชิ้น)
Quality	สัดส่วนชิ้นงานดีต่อชิ้นงานทั้งหมด
MQTT	Message Queuing Telemetry Transport เป็นโปรโตคอลการรับส่งข้อความแบบน้ำหนักเบา ที่ออกแบบมาสำหรับอุปกรณ์ Internet of Things (IoT) และระบบอุตสาหกรรม (IIoT) ทำงานด้วยโมเดล Publish/Subscribe ผ่านตัวกลางที่เรียกว่า Broker เหมาะกับเครือข่ายที่จำกัดแบนด์วิดท์และต้องการความเร็วสูง

คำศัพท์	ความหมาย
Planned Time	ในโครงงานนี้หมายถึงเวลารวมตั้งแต่ระบบเริ่มทำงานจนถึงปัจจุบัน (now - startTime)
Watchdog	กลไกจับเวลาเมื่อไม่มีชิ้นงานผ่าน เพื่อเปลี่ยนสถานะเป็น IDLE
IDLE	สภาวะเครื่องไม่มีการไหลของชิ้นงานตามเงื่อนไขเวลา
Near Real-time	การอัปเดตข้อมูลใกล้เวลาจริง โดยฝั่ง Logic อัปเดตประมาณทุก 1 วินาที และ Dashboard Refresh ทุก 5 วินาที
Manual QC	กระบวนการที่คนใช้สายตา การสัมผัส หรืออุปกรณ์พื้นฐานตรวจเช็คสินค้า งาน หรือระบบ เพื่อหาข้อผิดพลาด ก่อนส่งมอบให้ลูกค้า โดยเน้นความยืดหยุ่นและการตัดสินใจจากประสบการณ์ของมนุษย์

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการจัดทำโครงงาน "โมเดลสายพานลำเลียงเพื่อการประเมินประสิทธิภาพเครื่องจักรและแสดงผลแบบเรียลไทม์ (Conveyor Belt Model for Real-Time Overall Equipment Effectiveness (OEE) Evaluation)" คณะผู้จัดทำได้ศึกษาหลักการ ทฤษฎี และเอกสารวิชาการที่เกี่ยวข้องเพื่อเป็นแนวทางในการดำเนินงาน โดยแบ่งเนื้อหาออกเป็นหัวข้อหลักดังต่อไปนี้:

2.1 แนวคิดและส่วนประกอบของระบบสายพานลำเลียง

2.2 ทฤษฎีการประเมินประสิทธิผลโดยรวมของเครื่องจักร (OEE)

2.3 เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง (IoT) และอุปกรณ์ฮาร์ดแวร์

2.4 ซอฟต์แวร์ระบบเปิด (Open-source) และโปรโตคอลการสื่อสารเรียลไทม์

2.5 งานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและส่วนประกอบของระบบสายพานลำเลียง

ระบบสายพานลำเลียง (Conveyor System) เป็นอุปกรณ์เคลื่อนย้ายวัสดุที่มีบทบาทสำคัญอย่างยิ่งในกระบวนการผลิตและระบบโลจิสติกส์อัตโนมัติยุคปัจจุบัน ในเชิงอุตสาหกรรมและการศึกษา ระบบสายพานมักถูกใช้เป็นเครื่องมือต้นแบบในการทดสอบการไหลของวัสดุ เนื่องจากระบบสายพานเป็นจุดเชื่อมต่อระหว่างเครื่องจักรแต่ละสถานี ทำให้พฤติกรรมและการทำงานของสายพานสามารถสะท้อนถึงระดับผลิตภาพและความเสถียรของเครื่องจักรในสายการผลิตได้โดยตรง

2.1.1 ประเภทของสายพานที่นิยมใช้ในโมเดลจำลอง

1. สายพานแบบพื้นเรียบ (Flat Belt Conveyor): เป็นโมเดลที่นิยมใช้มากที่สุด มีพื้นผิวเรียบ เหมาะสำหรับการขนย้ายสินค้าทั่วไป เช่น กล่อง หรือชิ้นงานทดสอบ
2. สายพานโมดูลาร์ (Modular Belt): ทำจากชิ้นส่วนพลาสติกมาประกอบกันคล้ายการก่ออิฐ ทนทาน แข็งแรง และซ่อมแซมเฉพาะจุดได้ง่าย
3. สายพาน PVC (PVC Conveyor Belt): ผิวมัน ยืดหยุ่นได้ดี ราคาประหยัด เหมาะสำหรับโครงงานนวัตกรรมและงานแพ็คเกจจิ้งทั่วไป

2.1.2 องค์ประกอบเชิงกลของโมเดลสายพานลำเลียง

- โครงสร้างและฐานรองรับ (Frame): โครงหลักเพื่อยึดอุปกรณ์ ใช้วัสดุอะลูมิเนียมโปรไฟล์สำเร็จรูปที่มีน้ำหนักเบาและมีความแข็งแรงสูง

- ชุดขับเคลื่อน (Drive Unit): มอเตอร์เกียร์กระแสตรง (DC Geared Motor) ทำหน้าที่ให้แรงบิดและความเร็วที่เหมาะสมในการหมุนเพลาขับ
- ลูกกลิ้งประคอง (Idler Roller): ลูกกลิ้งที่ติดตั้งอยู่ปลายสุดอีกด้านหนึ่งเพื่อรักษาระดับความตึงและช่วยให้สายพานหมุนวนได้อย่างราบรื่น

2.2 ทฤษฎีการประเมินประสิทธิผลโดยรวมของเครื่องจักร (OEE)

การประเมินประสิทธิภาพเครื่องจักรในงานวิจัยนี้อ้างอิงดัชนีชี้วัดสากลที่เรียกว่า ประสิทธิผลโดยรวมของเครื่องจักร หรือ OEE (Overall Equipment Effectiveness) ซึ่งเป็นเครื่องมือในการค้นหาความสูญเสียในกระบวนการผลิต โดยค่า OEE เกิดจากการคำนวณความสัมพันธ์ของปัจจัยหลัก 3 ประการ ดังนี้:

$$OEE(\%) = \left(\frac{Availability}{100}\right) \times \left(\frac{Performance}{100}\right) \times \left(\frac{Quality}{100}\right) \times 100$$

2.2.1 อัตราความพร้อมใช้งาน (Availability: A)

เป็นการวัดความสูญเสียด้านเวลา (Downtime) ที่เกิดจากเครื่องจักรหยุดทำงานเปรียบเทียบกับเวลาที่วางแผนไว้ โดยในโครงงานนี้มีการกำหนดเงื่อนไขลอจิกเวลาการทำงาน (Running Time Logic) คือ หากเซนเซอร์ไม่มีการอัปเดตข้อมูลชิ้นงานนานเกิน 60 วินาที ระบบจะเปลี่ยนสถานะเครื่องจักรเป็น IDLE (STOPPED) ทันที และหยุดสะสมเวลา Running Time

$$Availability(\%) = \left(\frac{\text{Running Time}}{\text{Planned Time}}\right) \times 100$$

หมายเหตุ: Planned Time ในโครงงานนี้ หมายถึงเวลารวมตั้งแต่ Node-RED เริ่มทำงานจนถึงปัจจุบัน (now - startTime) ไม่ใช่เวลากะการผลิตที่ผู้ใช้กรอกเอง ส่วน Running Time คือเวลาที่ระบบถือว่าเครื่องอยู่ในสถานะ RUNNING และมีการสะสมเวลาอย่างต่อเนื่อง

2.2.2 ประสิทธิภาพการเดินเครื่อง (Performance: P)

เป็นการเปรียบเทียบความเร็วในการผลิตจริงกับความเร็วมาตรฐาน โดยนับจำนวนชิ้นงานทั้งหมดที่ผ่านเซนเซอร์ต้นสายพาน (Total Count) ซึ่งในโครงงานนี้กำหนดมาตรฐานการผลิต (Ideal Cycle Time) ไว้ที่ 5 วินาทีต่อชิ้น

$$Performance(\%) = \left(\frac{\text{Total Count}}{\text{Expected Output}}\right) \times 100$$

2.2.3 อัตราคุณภาพ (Quality: Q)

เป็นการวัดอัตราส่วนของชิ้นงานที่ดีเทียบกับชิ้นงานที่ผลิตออกมาได้ทั้งหมด โดยระบบจะคำนวณจากจำนวนชิ้นงานดี (Good Count) ที่ผ่านเซนเซอร์ปลายสายพาน หารด้วยยอดจำนวนชิ้นงานรวม (Total Count) ส่วนชิ้นงานเสีย (Waste) บันทึกผ่านกระบวนการ Manual QC ที่จุดกึ่งกลางสายพาน

$$Quality(\%) = \left(\frac{\text{Good Count}}{\text{Total Count}}\right) \times 100$$

2.3 เทคโนโลยี IIoT และอุปกรณ์ฮาร์ดแวร์

การเปลี่ยนผ่านไปสู่โรงงานอัจฉริยะ (Smart Factory) ได้เปลี่ยนรูปแบบการเก็บข้อมูล OEE จากเดิมที่เป็นการบันทึกด้วยเอกสาร (Manual) มาเป็นการจัดเก็บข้อมูลอัตโนมัติด้วย เทคโนโลยีอินเทอร์เน็ตของสรรพสิ่งในอุตสาหกรรม (Industrial Internet of Things: IIoT) โดยมีฮาร์ดแวร์หลักดังนี้

2.3.1 ESP32 Development Board

ESP32: บอร์ดไมโครคอนโทรลเลอร์ สถาปัตยกรรม 32-bit มีโมดูล Wi-Fi ในตัว ทำหน้าที่เป็นอุปกรณ์ปลายทาง (Edge Device) รับสัญญาณจากเซนเซอร์ ประมวลผลลอจิกเบื้องต้น และส่งข้อมูลเข้าสู่ระบบเครือข่าย

2.3.2 เซนเซอร์อินฟราเรดตรวจจับวัตถุ (IR Infrared Sensor)

ทำงานโดยการส่งและรับรังสีอินฟราเรด เมื่อมีวัตถุตัดผ่านลำแสงจะเกิดการเปลี่ยนสถานะทางลอจิก โครงงานนี้ใช้เซนเซอร์ 3 ตัวติดตั้งตามตำแหน่งบนสายพาน ดังแสดงในตาราง

ตารางที่ 2.1 ตำแหน่งและการตั้งค่าเซนเซอร์อินฟราเรดบนสายพานลำเลียง

ตำแหน่งบนสายพาน	**ขา GPIO**	**หน้าที่**	**วิธีนับ**	**MQTT Topic**
จุดที่ 1: ต้นสายพาน	GPIO 2	นับยอดรวม (Total)	อัตโนมัติเมื่อชิ้นงานตัดผ่าน	factory/conveyor/total

ตำแหน่งบนสายพาน	ขา GPIO	หน้าที่	วิธีนับ	MQTT Topic
จุดที่ 2: กึ่งกลางสายพาน	GPIO 5	นับชิ้นงานเสีย (Waste)	Manual QC — พนักงานตรวจด้วยสายตาแล้วจับชิ้นงานเสียให้โดนเซนเซอร์	factory/conveyor/waste
จุดที่ 3: ปลายสายพาน	GPIO 4	นับชิ้นงานดี (Good)	อัตโนมัติเมื่อชิ้นงานดีไหลถึงปลายสาย	factory/conveyor/good

- **กระบวนการ Manual QC:** พนักงาน QC ประจำจุดกึ่งกลางสายพานทำหน้าที่ตรวจสอบคุณภาพชิ้นงานด้วยสายตา หากพบชิ้นงานที่ไม่ได้มาตรฐาน จะจับชิ้นงานนั้นให้โดนเซนเซอร์กลางเพื่อบันทึก Waste Count จากนั้นชิ้นงานดีที่เหลือจะไหลต่อไปยังปลายสายพานเพื่อบันทึก Good Count
- **การป้องกันสัญญาณรบกวน:** ระบบตั้งค่าพอร์ตเป็น INPUT_PULLUP และทำงานเมื่อเกิดการเปลี่ยนแปลงสถานะแบบขอบขาลง (Falling Edge: HIGH → LOW) พร้อมใช้ millis() ตรวจสอบเวลาเพื่อหน่วงเวลากรองสัญญาณรบกวน (Debounce Delay) ไว้ที่ 150 มิลลิวินาที ป้องกันปัญหาสัญญาณเบิ้ลซ้ำได้อย่างมีประสิทธิภาพ

2.4 ซอฟต์แวร์ระบบเปิดและโปรโตคอลการสื่อสารเรียลไทม์

เพื่อตอบโจทย์การลดต้นทุนสำหรับอุตสาหกรรม SME และรองรับ สถาปัตยกรรมแบบแยกส่วน (Decoupling Architecture) โครงงานนี้เลือกใช้ซอฟต์แวร์ระบบเปิด (Open-source) เป็นหลัก ได้แก่:

2.4.1 โปรโตคอล MQTT และ HiveMQ Broker

โครงงานนี้ใช้ไลบรารี PubSubClient ในการส่งข้อมูลผ่านโปรโตคอล MQTT ไปยัง HiveMQ Cloud โดยเชื่อมต่อผ่านพอร์ต 8883 ซึ่งรองรับมาตรฐานความปลอดภัย TLS/SSL ในระบบต้นแบบใช้ setInsecure() เพื่อความสะดวกในการทดสอบ หากนำไปใช้งานจริงควรติดตั้ง CA Certificate ให้ครบถ้วน และทำการขยายขนาด Buffer ของ Client เป็น 512 bytes เพื่อรองรับขนาดแพ็กเกจข้อมูลที่มีการเข้ารหัส ข้อมูลจะถูกแยกส่งตามหัวข้อ (Topic) อย่างเป็นระเบียบ เช่น factory/conveyor/total, factory/conveyor/waste และ factory/conveyor/good

2.4.2 Node-RED

ซอฟต์แวร์รูปแบบ Flow-based Programming ทำหน้าที่เป็นระบบประมวลผลหลัก (Logic Layer) คอยรับข้อมูลจากโปรโตคอล MQTT มาตรวจสอบสถานะเครื่องจักรและคำนวณสมการ OEE โดยมีการประยุกต์ใช้หน่วยความจำระบบ (Flow Context) เพื่อบันทึกค่าเวลาทำงานสะสม (Running Time) ไม่ให้สูญหาย ใช้เทคนิค Data Clamping ในการจำกัดเพดานผลลัพธ์ของสมการไม่ให้เกิน 100% หรือติดลบ ก่อนทำการแยกชุดข้อมูล (Data Routing) ส่งไปยังหน้าจอแสดงผลและฐานข้อมูลต่อไป โดยอัปเดตค่าประมาณทุก 1 วินาที

2.4.3 Influx DB v2

ฐานข้อมูลอนุกรมเวลา (Time-Series Database) เหมาะสำหรับการเก็บข้อมูลเซนเซอร์ที่มีสแตมป์เวลา (Timestamp) ผูกติดอยู่ โครงงานนี้ใช้ Bucket ชื่อ conveyor_oee และ Measurement oee_metrics ทำให้สามารถจัดเก็บข้อมูลแบบใกล้เวลาจริงได้อย่างมีประสิทธิภาพ

2.4.4 Grafana

เครื่องมือวิเคราะห์และสร้าง Dashboard ทำหน้าที่เป็นส่วนแสดงผล (Presentation Layer) โดยใช้ภาษา Flux ในการ Query ดึงข้อมูลจาก Influx DB มาแสดงผลในรูปแบบกราฟ เกจวัด และมาตรวัด OEE โดย Dashboard Refresh ประมาณทุก 5 วินาที

2.5 งานวิจัยที่เกี่ยวข้อง

จากการสืบค้นเอกสารและงานวิจัยวิชาการในฐานข้อมูลระดับชาติและนานาชาติ คณะผู้จัดทำได้นำงานวิจัยที่มีความเกี่ยวข้องกับการประยุกต์ใช้เทคโนโลยี IoT และสถาปัตยกรรมระบบในการวัดค่า OEE บนระบบสายพานลำเลียงมาเป็นแนวทางดังนี้:

1. **Wicaksono และคณะ (2024)** ได้เผยแพร่งานวิจัยเรื่อง "Implementation of MQTT Protocol on ESP32-Based OEE Analysis Development Board" ในวารสารวิชาการ Emitor: Jurnal Teknik Elektro (Vol. 24, No. 2) งานวิจัยนี้ได้พัฒนาและทดสอบบอร์ดประมวลผลวิเคราะห์ค่า OEE โดยใช้ไมโครคอนโทรลเลอร์ ESP32 เป็นอุปกรณ์ปลายทาง (Edge Device) ในการตรวจจับสัญญาณพฤติกรรมของเครื่องจักรจากหน้างาน

และส่งข้อมูลผ่านโปรโตคอล MQTT ผลการศึกษาแสดงให้เห็นว่า การใช้ฮาร์ดแวร์ต้นทุนต่ำอย่างบอร์ด ESP32 ร่วมกับระบบส่งข้อมูล MQTT สามารถนำมาคำนวณและประเมินค่า OEE เพื่อเป็นทางเลือกในการวัดประสิทธิภาพเครื่องจักรแบบเรียลไทม์แทนการจดบันทึกแบบเดิมได้อย่างมีเสถียรภาพ คณะผู้จัดทำจึงนำสถาปัตยกรรมฮาร์ดแวร์และโปรโตคอล MQTT นี้มาใช้เป็นแกนหลักในโครงงาน

2. **ชนม์รัตน์ ตติยะวรนันท์ (2568) Engineering Transactions มหาวิทยาลัยเทคโนโลยีมหานคร Vol. 28 No. 2 หน้า 126–136 งานวิจัยในวารสาร** เรื่อง "Workpiece Inspection and Sorting System using Image Processing on CiRA CORE Integrated with PLC and IoT" ได้นำเสนอการพัฒนาและประเมินประสิทธิภาพเครื่องจักรในระบบคัดแยกชิ้นงานบนสายพานลำเลียง โดยระบบมีการเชื่อมต่อเข้ากับโปรแกรม Node-RED ผ่านเทคโนโลยี IoT เพื่อทำหน้าที่เป็นตัวกลาง (Middleware) ในการรับข้อมูล แปลงค่า และแสดงผลกระบวนการทำงานรวมถึงคำนวณค่าประสิทธิผลโดยรวมของเครื่องจักร (OEE) แบบเรียลไทม์ ซึ่งแสดงให้เห็นว่า Node-RED มีประสิทธิภาพและความยืดหยุ่นสูงในการประมวลผลลอจิกคณิตศาสตร์ คณะผู้จัดทำจึงได้นำแนวทางการประยุกต์ใช้ Node-RED มาสร้างฟังก์ชันคำนวณค่า Availability, Performance และ Quality ในโครงงานนี้

3. **งานวิจัยในวารสารวิชาการพระจอมเกล้าพระนครเหนือ** เรื่อง "การเพิ่มประสิทธิภาพการผลิตด้วยการวางแผนการบำรุงรักษา กรณีศึกษาเครื่องจักรสายพานลำเลียงภายในห้องรับ-จ่ายวัตถุดิบ" ได้ศึกษาการประเมินประสิทธิภาพโดยอ้างอิงดัชนีชี้วัดประสิทธิผลโดยรวมของเครื่องจักร (OEE) บนระบบเครื่องจักรสายพานลำเลียงโดยเฉพาะ เพื่อค้นหาสาเหตุของการสูญเสียในกระบวนการและลดการหยุดชะงักของสายพาน (Downtime) ผลการวิจัยระบุว่าการวิเคราะห์ปัจจัยย่อยของ OEE ช่วยให้ผู้ควบคุมมองเห็นจุดบกพร่องและแก้ไขปัญหาเชิงกลของสายพานลำเลียงได้อย่างแม่นยำ คณะผู้จัดทำจึงนำหลักทฤษฎีการจำแนกประเภทความสูญเสียของสายพานจากงานวิจัยเล่มนี้ มาเป็นแนวทางในการกำหนดรูปแบบการรับสัญญาณนับชิ้นงานดีและชิ้นงานเสียจากเซนเซอร์อินฟราเรด

4. **สมชาย เปรียงพรม, สุชาดี ธำรงสุข และวรรณลภย์ อนันตเจริญโชติ (2564)** ได้เผยแพร่งานวิจัยเรื่อง "การเพิ่มประสิทธิภาพการผลิตด้วยการวางแผนการบำรุงรักษา กรณีศึกษาเครื่องจักรสายพานลำเลียงภายในห้องรับ-จ่ายวัตถุดิบ" ในวารสารวิชาการพระจอมเกล้าพระนครเหนือ (ปีที่ 31 ฉบับที่ 2 หน้า 201–215) งานวิจัยนี้ศึกษาเครื่องจักรสายพานลำเลียงโดยตรง โดยวิเคราะห์ค่า OEE เพื่อค้นหาความสูญเสียในกระบวนการ พบว่าการบำรุงรักษา

แบบซ่อมเมื่อเสีย (Breakdown Maintenance) ทำให้เกิด Downtime และสูญเสียกำลังการผลิต นอกจากนี้ยังพบปัญหาของเสียจากวัตถุดิบตกหล่นในกระบวนการจ่ายวัตถุดิบ หลังปรับปรุงด้วยการวางแผนบำรุงรักษาและติดตั้งอุปกรณ์ลดของเสีย (แปรงปัดเศษวัตถุดิบและแผ่นประคอง) สามารถเพิ่มค่า OEE จากร้อยละ 75 เป็นร้อยละ 79 และลดปริมาณของเสียได้มากกว่าร้อยละ 90

จากแนวคิดของงานวิจัยเล่มนี้ คณะผู้จัดทำนำมาประยุกต์ในโครงงาน 2 ประเด็นหลัก ดังนี้:

ด้าน Availability / Downtime: งานวิจัยเน้นว่าต้องมองเห็นและลดช่วงที่สายพานหยุดชะงัก คณะผู้จัดทำจึงออกแบบกลไก Watchdog 60 วินาทีใน Node-RED หากไม่มีชิ้นงานผ่านเซนเซอร์เกินเวลาที่กำหนด ระบบจะเปลี่ยนสถานะเป็น IDLE และหยุดสะสม Running Time เพื่อไม่ให้นับว่าเครื่องยังทำงานทั้งที่ของไม่ไหล

ด้าน Quality / ของเสีย: งานวิจัยแสดงให้เห็นว่าการแยกวัดและลดของเสียมีผลต่อค่า OEE โดยตรง คณะผู้จัดทำจึงออกแบบการนับชิ้นงานแยกเป็น Total / Good / Waste โดยจุดของเสียใช้กระบวนการ Manual QC คือพนักงานตรวจสอบด้วยสายตาแล้วจับชิ้นงานเสียให้โดนเซนเซอร์กึ่งกลางสายพาน เพื่อให้แกนคุณภาพ (Quality) มีข้อมูลจริงนำไปคำนวณได้ ไม่ใช่การประมาณจากยอดรวมอย่างเดียว

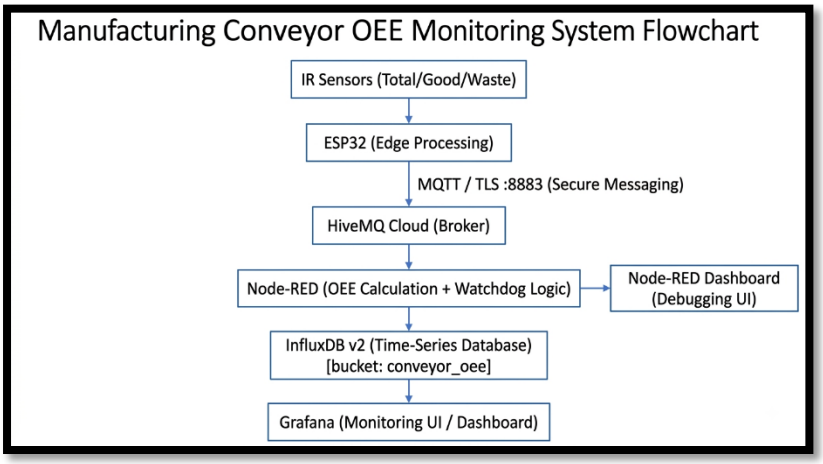
บทที่ ๓
วิธีดำเนินการโครงงาน

การจัดทำโครงงาน "โมเดลสายพานลำเลียงเพื่อการประเมินประสิทธิภาพเครื่องจักรและแสดงผลแบบเรียลไทม์" คณะผู้จัดทำได้กำหนดแผนการดำเนินงานและขั้นตอนการออกแบบระบบอย่างเป็นลำดับขั้น เพื่อให้บรรลุวัตถุประสงค์ที่ตั้งไว้ โดยมีรายละเอียดวิธีดำเนินการดังต่อไปนี้

3.1 ขั้นตอนการดำเนินงานโดยรวม

1. ศึกษาปัญหาและกำหนดขอบเขตโปรเจค
2. ศึกษาทฤษฎี OEE และเทคโนโลยีที่เกี่ยวข้อง
3. ออกแบบสถาปัตยกรรมระบบและกำหนด MQTT topics
4. พัฒนา firmware ESP32 สำหรับนับชิ้นงานและส่งข้อมูล
5. พัฒนา Flow ใน Node-RED สำหรับคำนวณ OEE และ Watchdog
6. ตั้งค่า Influx DB และ Grafana Dashboard
7. ทดสอบระบบแบบครบวงจรและปรับปรุง
8. จัดทำเอกสารรายงานและคู่มือใช้งาน

3.2 ผังงานและสถาปัตยกรรมของระบบ (System Architecture)



รูปที่ 3.1 ผังงานการทำงานของโปรแกรม (Flowchart)

3.3 การออกแบบฮาร์ดแวร์

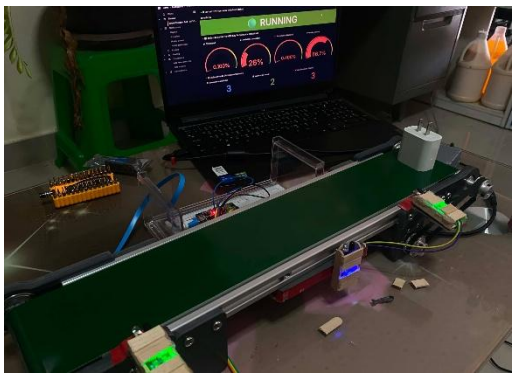
3.3.1 หลักการออกแบบ

- แยกชั้น Edge กับ Logic ให้ชัด เพื่อปรับสูตร OEE ได้โดยไม่ต้องแฟลชบอร์ดใหม่บ่อย ๆ
- ใช้ MQTT topics แยกตามชนิดข้อมูล เพื่อง่ายต่อการขยาย
- เก็บข้อมูลแบบ time-series เพื่อวิเคราะห์แนวโน้ม
- แสดงผลด้วยเครื่องมือมาตรฐานอุตสาหกรรมเพื่อการสาธิตที่น่าเชื่อถือ

3.3.2 การออกแบบและประกอบฮาร์ดแวร์ (Hardware Setup)

- ประกอบชุดจำลองสายพานลำเลียง: การประกอบโครงสร้างอะลูมิเนียมโปรไฟล์ การติดตั้งมอเตอร์เกียร์ DC และลูกกลิ้งประคองสายพาน
- การต่อวงจรไมโครคอนโทรลเลอร์: การเชื่อมต่อบอร์ด ESP32
- การติดตั้งเซนเซอร์อินฟราเรด (IR Sensor): การวางตำแหน่งและต่อสายสัญญาณเซนเซอร์ 3 จุด เข้ากับพอร์ตดิจิทัลของ ESP32 ได้แก่:
    - ขา 2: ตรวจจับชิ้นงานทั้งหมด (Total)
    - ขา 4: ตรวจจับชิ้นงานดี (Good)
    - ขา 5: ตรวจจับชิ้นงานเสีย (Waste) Manual QC

รูปที่ 3.2 การปรึกษาและประกอบโครงสร้างชุดสายพานลำเลียงร่วมกับอาจารย์ที่ปรึกษา



รูปที่ 3.3 การทดสอบการส่งผ่านข้อมูลและการแสดงผล OEE แบบเรียลไทม์บน Grafana Dashboard

3.3.3 การกำหนดขาพินและตำแหน่งเซนเซอร์

ตำแหน่งบนสายพาน	อุปกรณ์	GPIO	หน้าที่	วิธีนับ
จุดที่ 1 ต้นสายพาน	IR Sensor Total	2	นับชิ้นงานเข้าสู่สายพาน	อัตโนมัติ
จุดที่ 2 กึ่งกลางสายพาน	IR Sensor Waste	5	นับชิ้นงานเสีย (Manual QC)	พนักงานจับชิ้นงานเสียให้โดนเซนเซอร์
จุดที่ 3 ปลายสายพาน	IR Sensor Good	4	นับชิ้นงานดี	อัตโนมัติ

กระบวนการ Manual QC: พนักงาน QC ตรวจสอบชิ้นงานด้วยสายตา หากเป็นของเสียจะจับชิ้นงานให้โดนเซนเซอร์กลางเพื่อบันทึก Waste จากนั้นชิ้นงานดีไหลต่อไปยังปลายสายเพื่อบันทึก Good

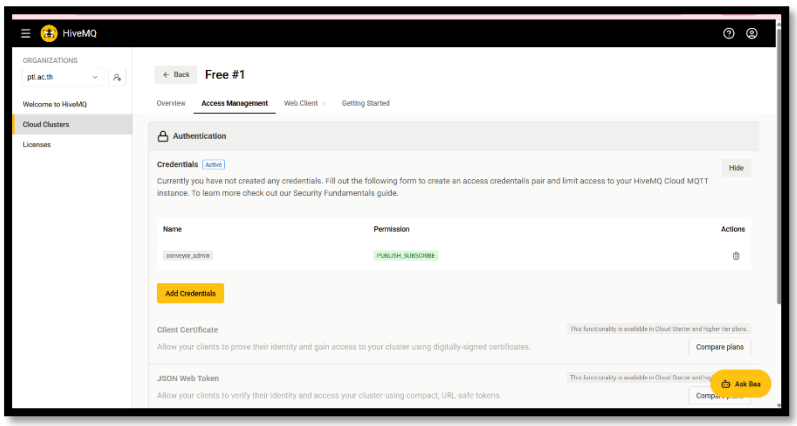
3.3.4 ตรรกะฝั่งเฟิร์มแวร์ (Firmware)

1. อ่านสถานะเซนเซอร์แบบ INPUT_PULLUP
2. ตรวจจังหวะขอบขาลง (Falling Edge) เมื่อวัตถุตัดผ่าน
3. ใช้ debounce ประมาณ 150 มิลลิวินาที
4. เพิ่มตัวนับและ publish ค่าขึ้น MQTT ทันที
5. มีกลไก reconnect Wi-Fi/MQTT เมื่อการเชื่อมต่อหลุด

3.4 การออกแบบการสื่อสารข้อมูล

3.4.1 การจัดการระบบเครือข่ายและคลาวด์ (Network & Communication)

- การตั้งค่า HiveMQ Cloud: การสร้าง Cluster, การกำหนด Username/Password สำหรับเชื่อมต่อ
- การกำหนดโครงสร้างหัวข้อ (Topic Routing): การออกแบบชื่อ Topic สำหรับแยกประเภทข้อมูลที่ส่งจาก ESP32



รูปที่ 3.4 การตั้งค่า HiveMQ Cloud

3.4.2 MQTT Topics

Topic	ความหมาย
factory/conveyor/total	ยอดนับชิ้นงานรวม
factory/conveyor/good	ยอดนับชิ้นงานดี
factory/conveyor/waste	ยอดนับชิ้นงานเสีย

3.5 การออกแบบซอฟต์แวร์ฝั่งหลังบ้าน (Node-RED)

3.5.1 ฟังก์ชันหลัก

1. Subscribe MQTT ทั้ง 3 topics
2. เก็บค่าล่าสุดใน flow context
3. ทำงานเป็นรอบทุก 1 วินาทีเพื่ออัปเดตตัวชี้วัด
4. คำนวณ Availability, Performance, Quality และ OEE
5. ตรวจ Watchdog 60 วินาที เพื่อกำหนดสถานะ RUNNING/IDLE
6. บันทึกลง Influx DB และส่งไปแสดงผล UI

3.5.2 สูตรที่ใช้ในระบบ

Quality

Quality = \frac{good}{total} \times 100

Availability

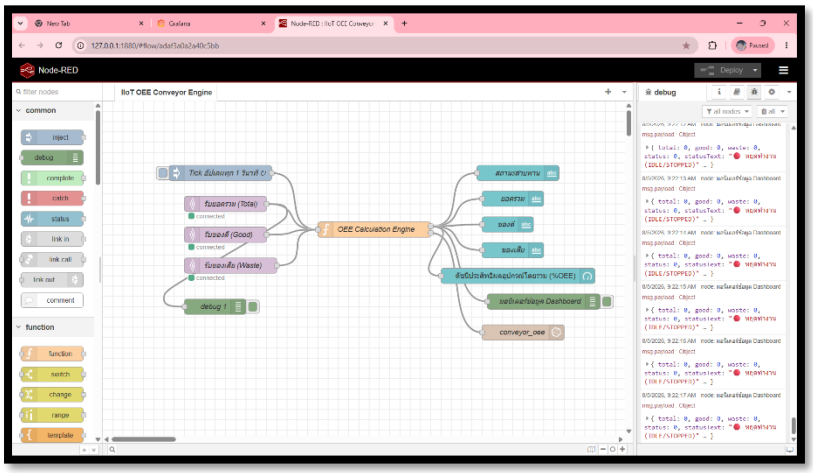
Availability = \frac{runningTime}{totalPlannedTime} \times 100

Performance

- Ideal Cycle Time = 5 วินาที/ชิ้น
- Expected Output = runningTime / 5
- Performance = (total / expectedOutput) × 100

OEE

OEE = Availability \times Performance \times Quality



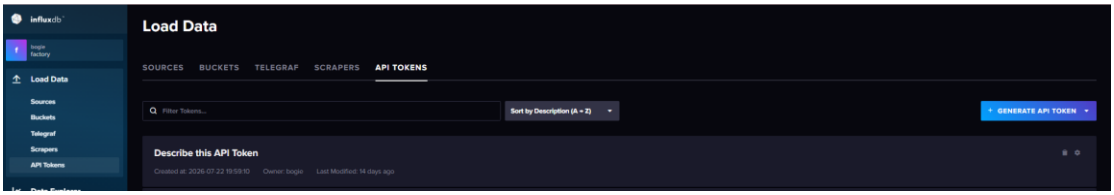
รูปที่ 3.5 การสร้าง Flow ใน Node-RED

3.6 การออกแบบฐานข้อมูลและ Dashboard

3.6.1 Influx DB

- Bucket: conveyor_oee
- Measurement: oee_metrics
- ตัวอย่างฟิลด์: total, good, waste, status, availability, performance, quality, oee, running_time_sec, total_time_sec
- ตัวอย่าง tag: conveyor_id = LINE_01

การเชื่อมต่อ Influx DB v2: การสร้าง Bucket, การจัดการ API Token และการบันทึกข้อมูลแบบอนุกรมเวลา (Time-Series)



รูปที่ 3.6 การเชื่อมต่อ Influx DB v2: การสร้าง Bucket, การจัดการ API Token และการบันทึกข้อมูลแบบอนุกรมเวลา (Time-Series)

3.6.2 การออกแบบหน้าจอแสดงผล (Dashboard Visualization)

- การตั้งค่า Grafana: การเชื่อมต่อ Data Source มายัง InfluxDB v2
- การเขียน Flux Query: ตัวอย่างคำสั่งที่ใช้ดึงข้อมูลมาแสดงผล

```
from(bucket: "conveyor_oee")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "oee_metrics")
  |> filter(fn: (r) => r["_field"] == "availability")
  |> last()
```

- การออกแบบส่วนแสดงผล (Panels): การจัดวางเกจวัด (Gauge) สำหรับค่า OEE, กราฟแนวโน้ม (Time-series Graph), และป้ายสถานะเครื่องจักร (RUNNING/IDLE) โดยกำหนดค่า Refresh อัตโนมัติทุก 5 วินาที

กลุ่มแผงแสดงผลหลัก:

1. สถานะเครื่องและ Running Time
2. เกจ OEE / Availability / Performance / Quality
3. ยอด Total / Good / Waste
4. กราฟแนวโน้มการผลิต

รูปที่ 3.7 การจัดรูปหน้าจอการแสดงผล OEE แบบเรียลไทม์บน Grafana Dashboard

3.7 ขั้นตอนติดตั้งระบบ

1. ตั้งค่า Wi-Fi และ MQTT ใน firmware แล้วอัปโหลดลง ESP32
2. เปิด Node-RED ที่พอร์ต 1880 แล้วนำเข้า flow
3. ตั้งค่า MQTT broker และ Influx DB ให้ตรงกับสภาพแวดล้อมจริง
4. สร้าง bucket conveyor_oee ใน Influx DB (พอร์ต 8086)
5. นำเข้า dashboard ใน Grafana (พอร์ต 3000)
6. ทดสอบการนับชิ้นงานและการอัปเดตกราฟ

3.8 เครื่องมือที่ใช้พัฒนา

รายการ	รายละเอียด
Arduino IDE	พัฒนาและอัปโหลด firmware
Node-RED	พัฒนาตรรกะ OEE
InfluxDB v2	เก็บข้อมูลเวลา
Grafana	สร้าง Dashboard
HiveMQ Cloud	MQTT Broker
GitHub	จัดเก็บซอร์สโค้ดและเอกสาร

3.9 การทดสอบระบบและการประเมินผล (System Testing & Evaluation)

- แผนการทดสอบระบบต่างๆ ประกอบไปด้วย
    - นับ Total
    - นับ Waste (Manual QC)
    - นับ Good
    - ส่ง MQTT
    - คำนวณ Quality
    - Idle Watchdog
    - การบันทึกของ influx DB
    - Grafana Refresh
    - Reconnect

- การทดสอบความแม่นยำของเซนเซอร์: วิธีการทดสอบปล่อยชิ้นงานผ่านสายพานจำนวน 105 ชิ้น (แบ่งเป็น 3 รอบ รอบละ 35 ชิ้น) และการหาค่าความคลาดเคลื่อนทางสถิติ
- การทดสอบความเสถียรของเครือข่าย: การเดินระบบต่อเนื่องเป็นเวลา 60 นาที เพื่อตรวจสอบปัญหาข้อมูลสูญหาย (Data Loss)
- การประเมินโดยผู้เชี่ยวชาญ: เครื่องมือที่ใช้ แบบสอบถาม และเกณฑ์การประเมินจากผู้เชี่ยวชาญจำนวน 3-5 ท่าน