

ความก้าวหน้าของโครงงาน

การพัฒนาระบบ Data Transmission ในไลน์การผลิตเครื่องยนต์ด้วยเทคโนโลยี IoT

STM SIAM TOYOTA MANUFACTURING co. ltd

 TK24-17 วิทยาลัยเทคนิคสัตหีบ



สถานะปัจจุบัน

สำเร็จเรียบร้อย (ระบบหลัก)

- ปัจจุบันทำการย้ายข้อมูลไปยัง PostgreSQL ทั้ง 4 ไลน์เป็นที่เรียบร้อยแล้ว
- สร้าง Databases บน PostgreSQL ครบทั้ง 4 ไลน์
- พร้อมนำไปใช้งานกับ Genesis หรืออื่น ๆ



Raspberry pi 4 Model B/
1 อุปกรณ์: ใช้เป็น Gateway
ในการย้ายข้อมูล **(ยืนยันทดสอบ)**

ยังไม่ได้ทำ (ระบบสำรอง)

- ส่งข้อมูลผ่าน MQTT

สถานที่ติดตั้ง

- Plant 1 Line G6



WIFI extender TP link
205 / 1 อุปกรณ์: ใช้ในการขยาย
สัญญาณ WIFI **(ยืนยันทดสอบ)**





แนวคิดผลงานใหม่

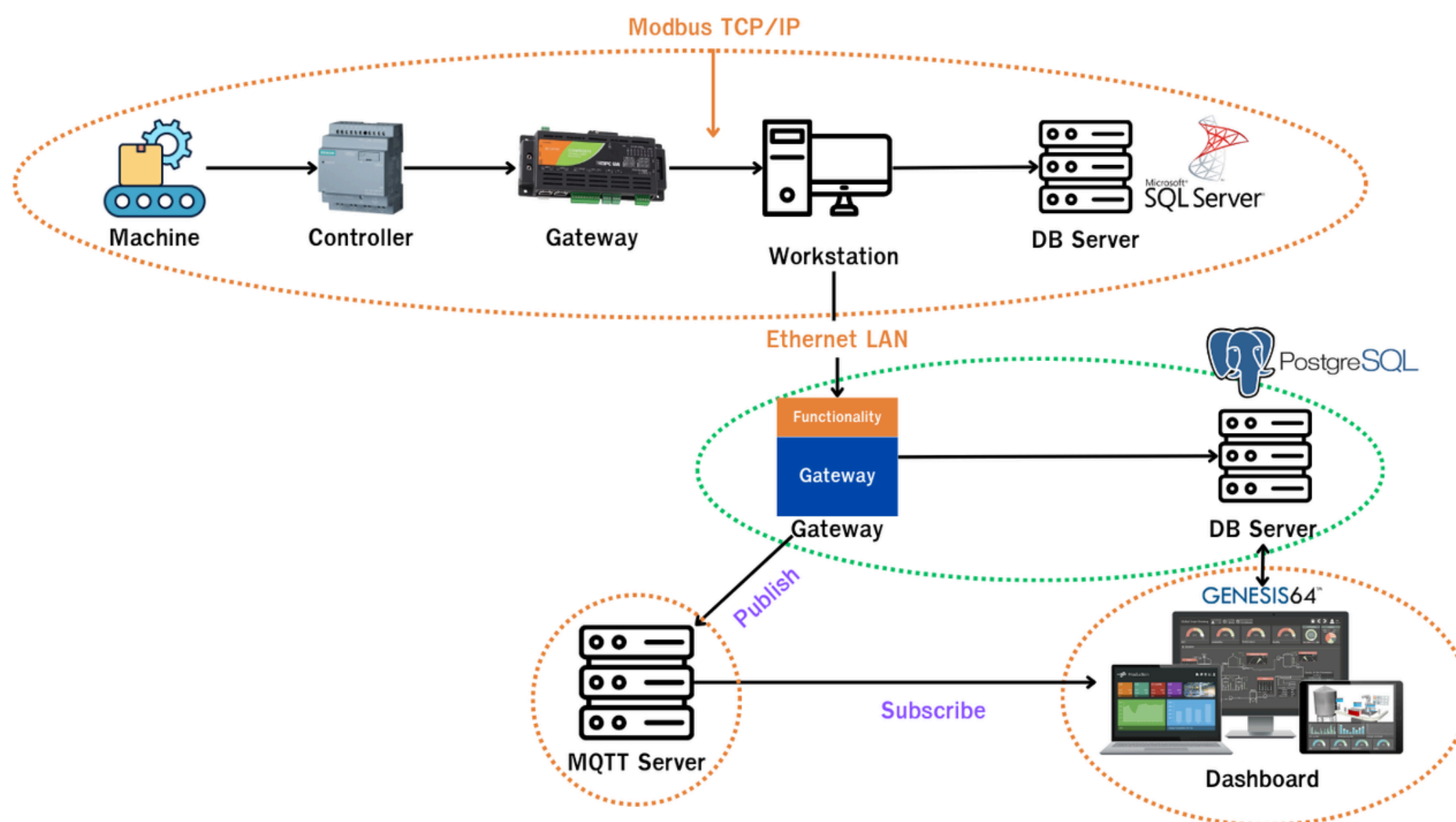


TK24-17



STM

New Concept



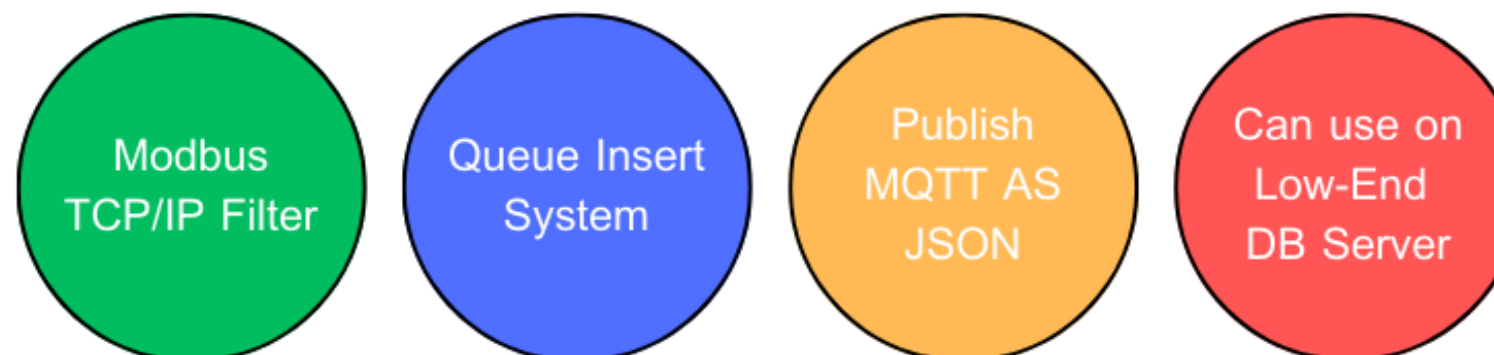
ปัญหาจริงๆ ที่สถานประกอบการพบเจอ

- เกิด timeout จาก Node-RED ในการส่งเข้า DB Server (ข้อมูลที่ต้องนำมาทำ OEE ศูนย์หาย)
- การส่งข้อมูล เป็นการส่งทุกครั้งที่ได้จาก Modbus และไม่มีการทำ Delay หรือ Queue สำหรับกรองข้อมูลก่อนส่ง เข้า DB Server
- MSSQL ใช้ทรัพยากรเครื่องหนักมาก
- ทั้ง 4 Line นำข้อมูลเข้า Server เครื่องเดียวกัน (แค่แยก DB)



* ความสามารถของ Gateway ที่พัฒนาขึ้นมา

สามารถกรองข้อมูลที่
Read ได้จาก Modbus TCP/IP



สามารถใช้บน DB Server ที่สเปคต่ำ

มีระบบ Queue ข้อมูลก่อน
Insert ข้อมูลไป DB Server

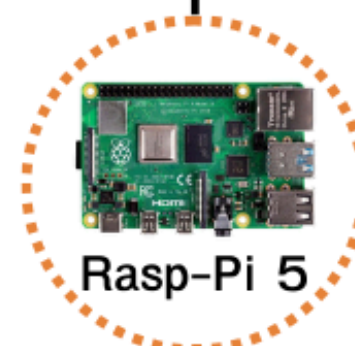
มีการส่งข้อมูลในรูปแบบ JSON
ผ่านโปรโตคอล MQTT

Functionality

Gateway

Hardware

Software



อะไรช่วยแก้ปัญหาและทำให้ดีขึ้นได้อย่างไร

Gateway ที่พัฒนาขึ้น

- มีการส่งข้อมูลผ่านโปรโตคอล MQTT เพื่อเป็นเส้นทางสำรองในการส่งข้อมูลเพื่อลดอัตรา การสูญเสียหายของข้อมูลที่เกิดจากการ timeout ระหว่างการส่งข้อมูลไป DB Server
- มีระบบกรองข้อมูล Modbus ให้มีการส่งต่อเมื่อค่าเปลี่ยน
- มีระบบ Queue เพื่อแก้ปัญหาข้อมูลเข้ามาพร้อมกันจำนวนมาก

PostgreSQL

- ตัว PostgreSQL กินทรัพยากรเครื่องต่ำ

ดีขึ้นอย่างไร

- ข้อมูลในการนำไปทำ OEE มีความ Real-Time มากขึ้น
- สามารถอ่านค่าจาก Modbus ได้ถี่ขึ้นและไม่ส่งผลเสียกับ DB Server
- สามารถนำ DB Server ตัวเก่ามาแยกทรัพยากรได้ (Visual Machine)

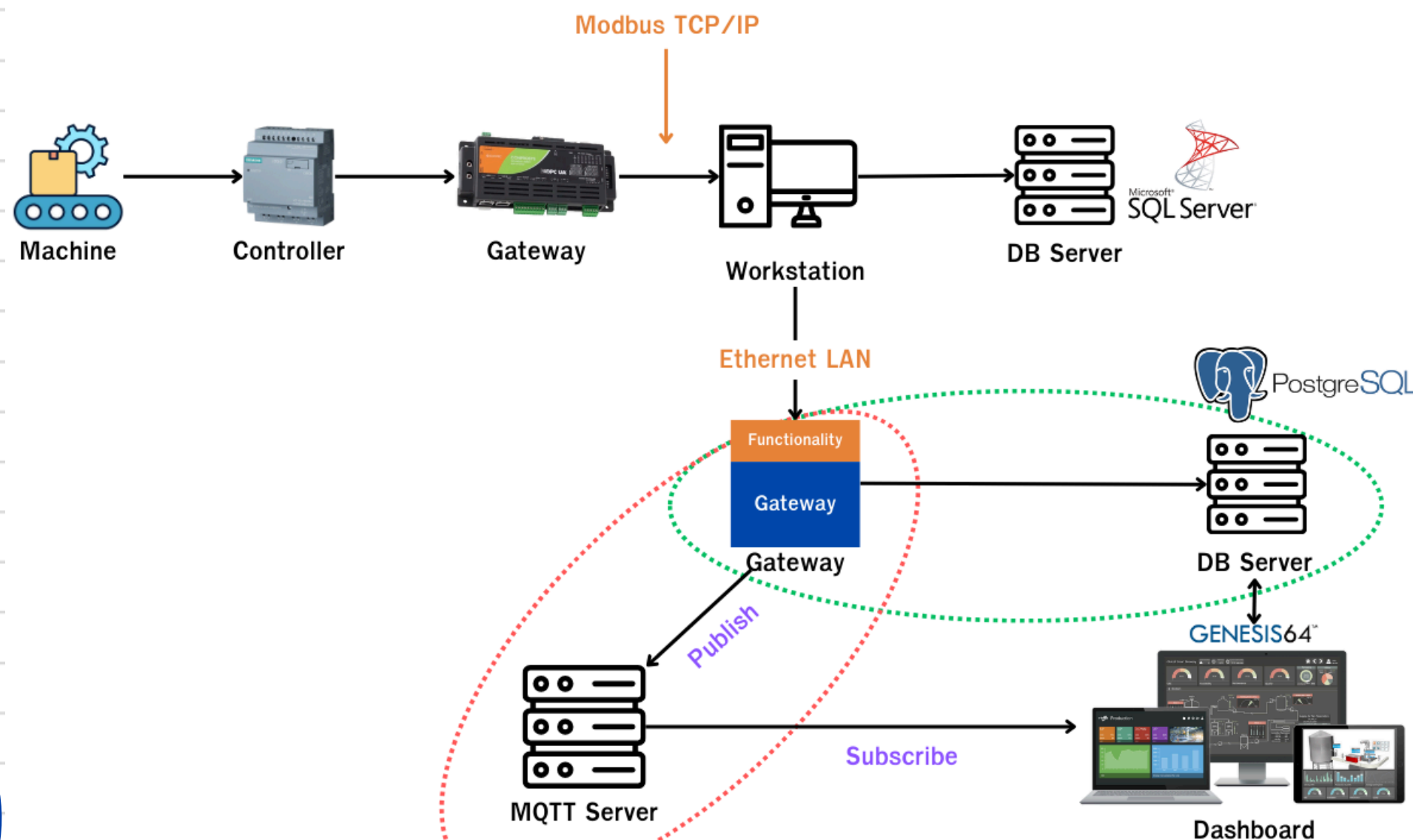




ความคืบหน้าในผลงานตอนนี้

● สำเร็จ

● ระหว่างดำเนินการ



ตารางความคืบหน้า

ข้อที่	ระบบ	ความสำคัญ	สำเร็จ
1	สร้าง DB บน PostgreSQL และ Gateway สามารถดึง Modbus ไปเก็บ DB ได้ (ระบบหลัก)	30%	✓
2	ระบบกรองข้อมูล Modbus (ระบบหลัก)	20%	✓
3	ระบบ Queue ก่อนส่งเข้า DB Server (ระบบหลัก)	20%	✓
4	การ Publish ข้อมูลออกในรูป JSON ผ่านโปรโตคอล MQTT (ระบบสำรอง)	30%	×
รวมความคืบหน้าที่สำเร็จทั้งหมด		70%	

ปัญหา/อุปสรรค

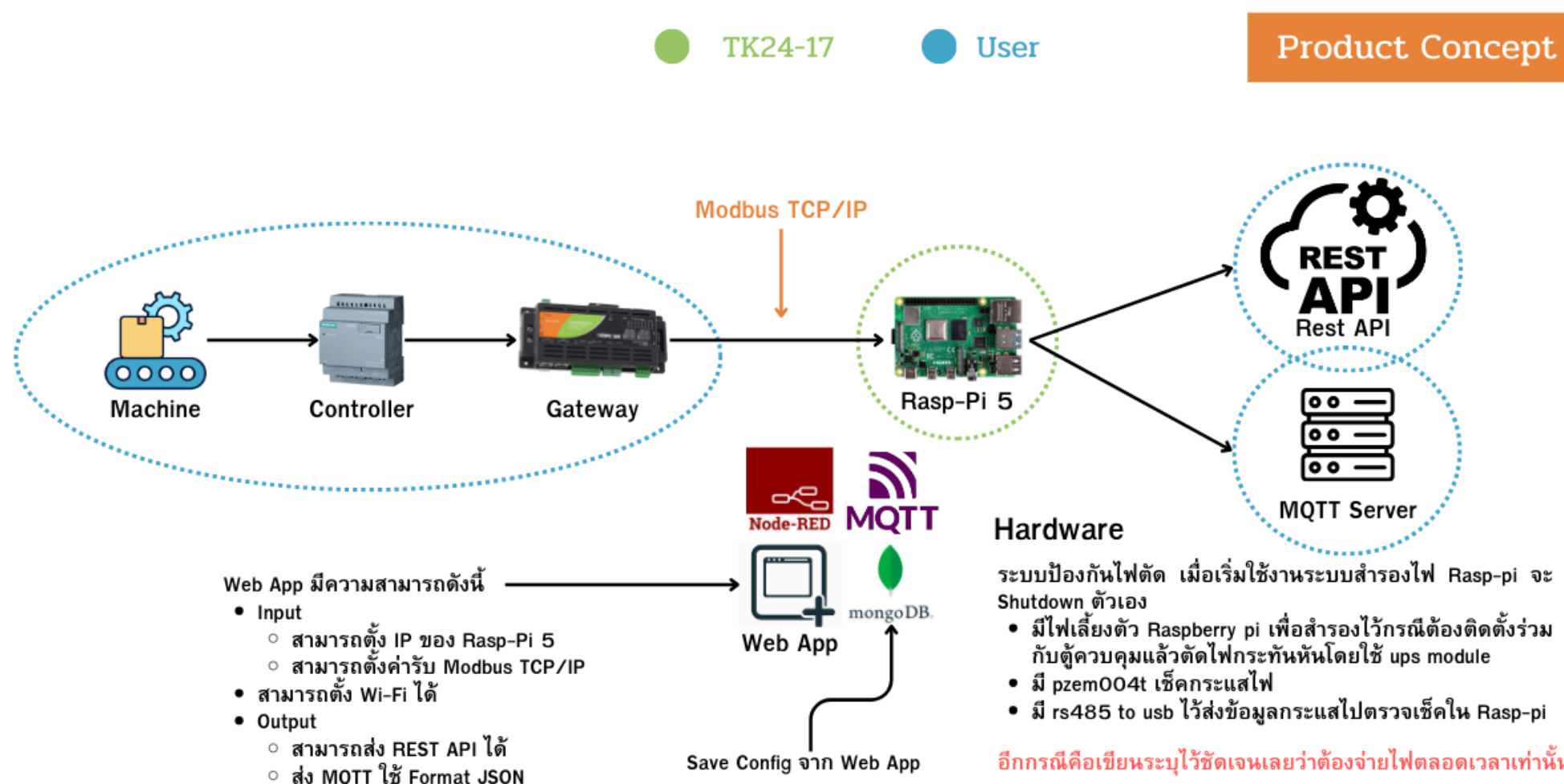


ปัญหาทางด้านเครือข่ายสัญญาณทำให้ **Insert** เข้า **Server** ช้าและมีโอกาสเกิด **timeout** สูงรวมถึงสัญญาณขาดหายทำให้ไม่สามารถเชื่อมต่อ **DB Server** ได้

- แก้ปัญหาด้วยการใช้ Wi-Fi Extender เพื่อให้งานสามารถเดินได้ระหว่างรอฟัง Infra แก้ไข



แนวคิดในการพัฒนาต่อยอด



Hardware

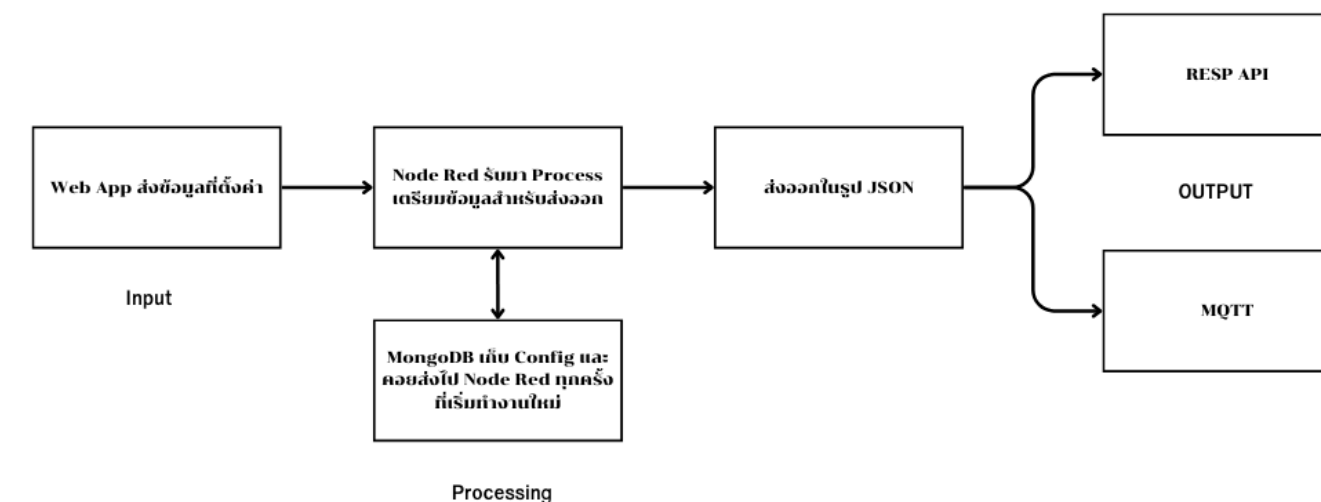
ระบบป้องกันไฟตัด เมื่อเริ่มใช้งานระบบสำรองไฟ Rasp-pi จะ Shutdown ตัวเอง

- มีไฟเลี้ยงตัว Raspberry pi เพื่อสำรองไว้กรณีต้องติดตั้งร่วมกับตู้ควบคุมแล้วตัดไฟกระชากกันโดยใช้ ups module
- มี pzem004t เซ็นเซอร์กระแสไฟ
- มี rs485 to usb ไว้ส่งข้อมูลกระแสไปตรวจเช็คใน Rasp-pi

อีกกรณีคือเขียนระบุไว้ชัดเจนเลยว่าต้องจ่ายไฟตลอดเวลาเท่านั้น

ความสามารถของระบบที่พัฒนาต่อจะเน้นให้มีความสามารถในการปรับแต่งการรับส่งข้อมูลได้มากขึ้นหรือก็คือให้มีความสามารถพอที่จะนำไปใช้ที่ไหนก็ได้

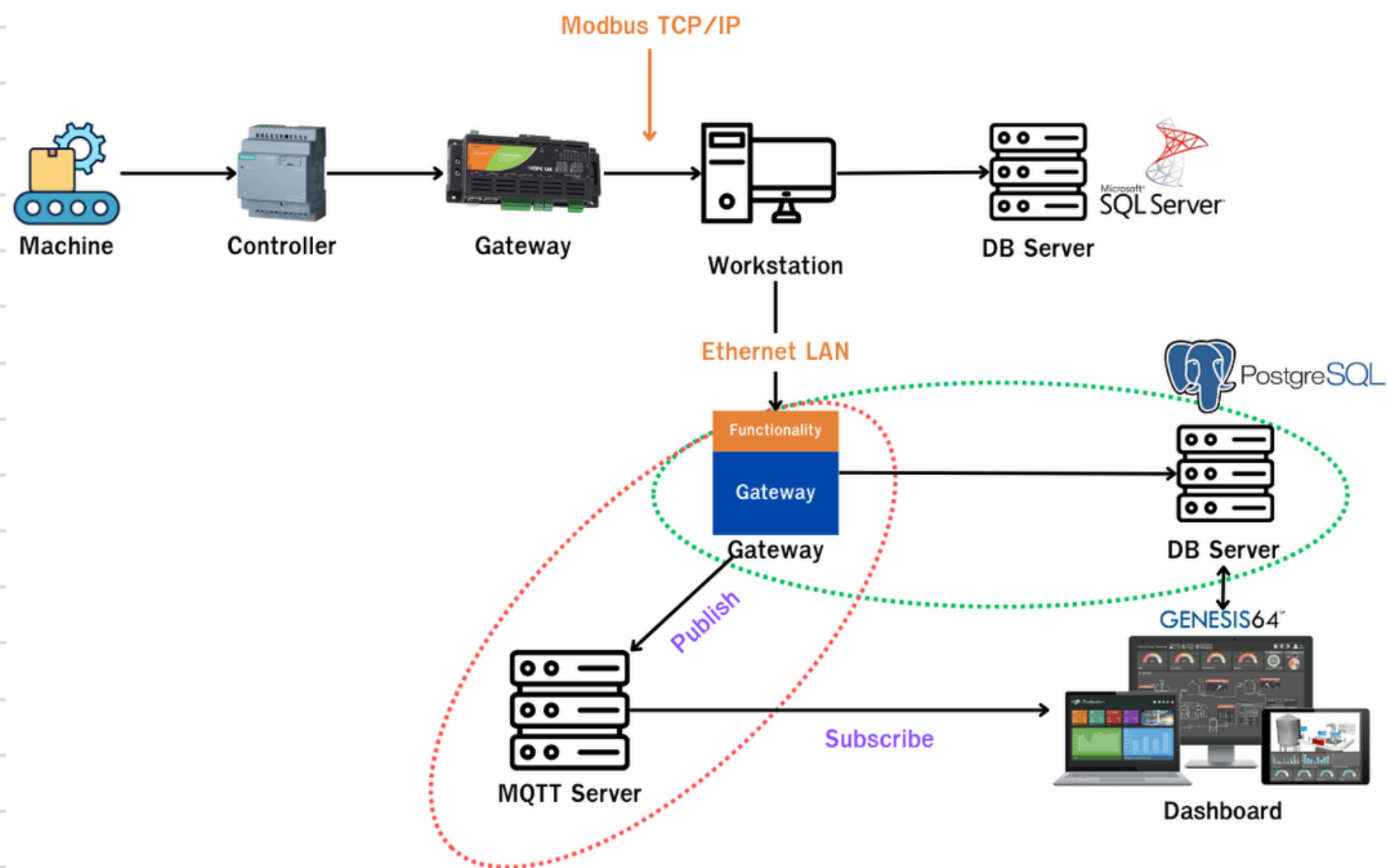
- มี Web App สามารถปรับแต่งหรือ config ระบบได้ดังนี้
 - ตั้งค่าการรับค่าจาก Modbus TCP/IP
 - มีการส่งข้อมูลแบบ Rest API
 - มีการส่งข้อมูลแบบ MQTT
- ความสามารถด้าน Hardware
 - สามารถใช้ในระบบที่ตัดไฟได้ทันที
 - ติดตั้งง่ายเพราะตัวอุปกรณ์มีขนาดเล็ก



สรุปความก้าวหน้า

● สำเร็จ

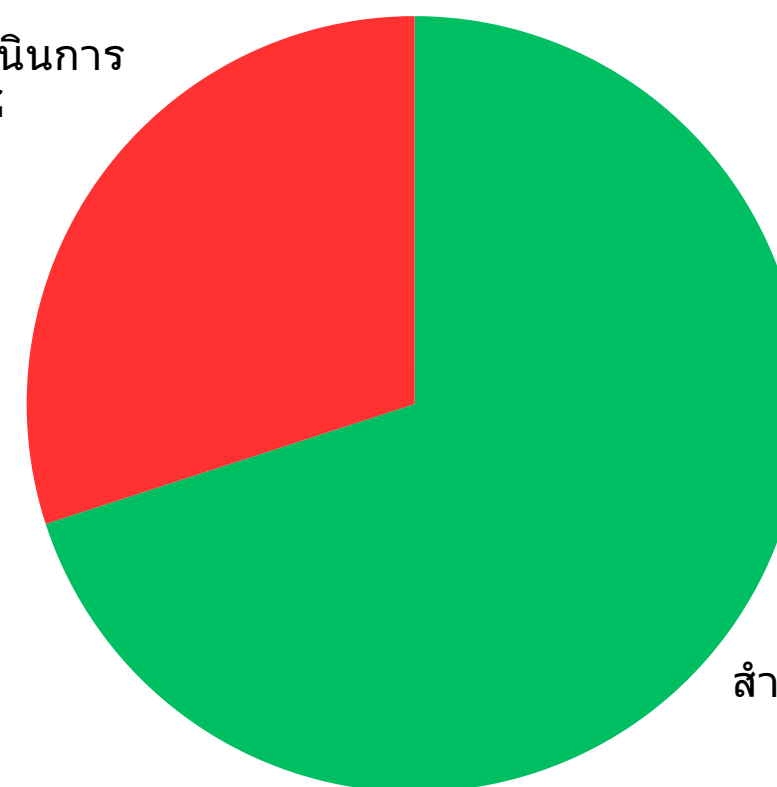
● ระหว่างดำเนินการ



โดยสรุปความก้าวหน้าจะมีดังนี้

- การเปลี่ยนแปลง Concept งานเพื่อให้สอดคล้องกับปัญหามากขึ้น
- ความคืบหน้าของตัวชิ้นงานที่เสร็จเกินครึ่ง
- มีแนวคิดที่จะพัฒนาระบบต่อ

ระหว่างดำเนินการ
30%



สำเร็จแล้ว
70%



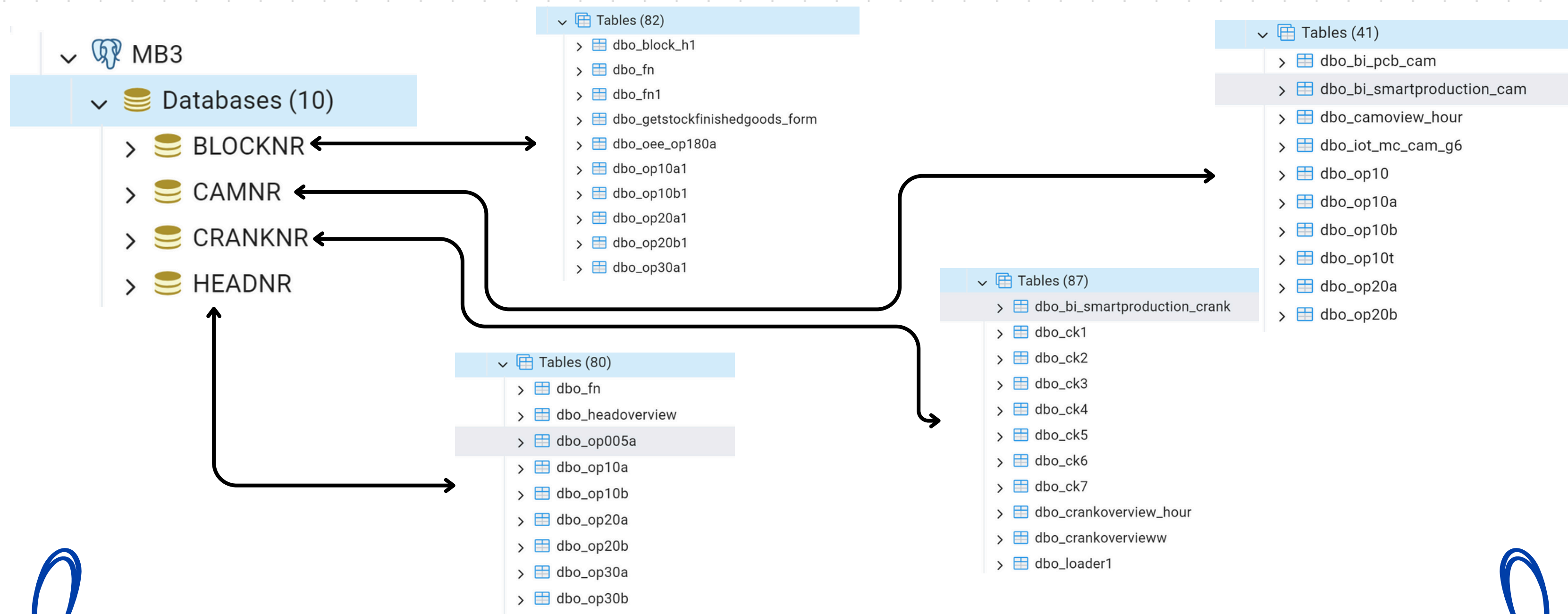
ภาพผลงาน



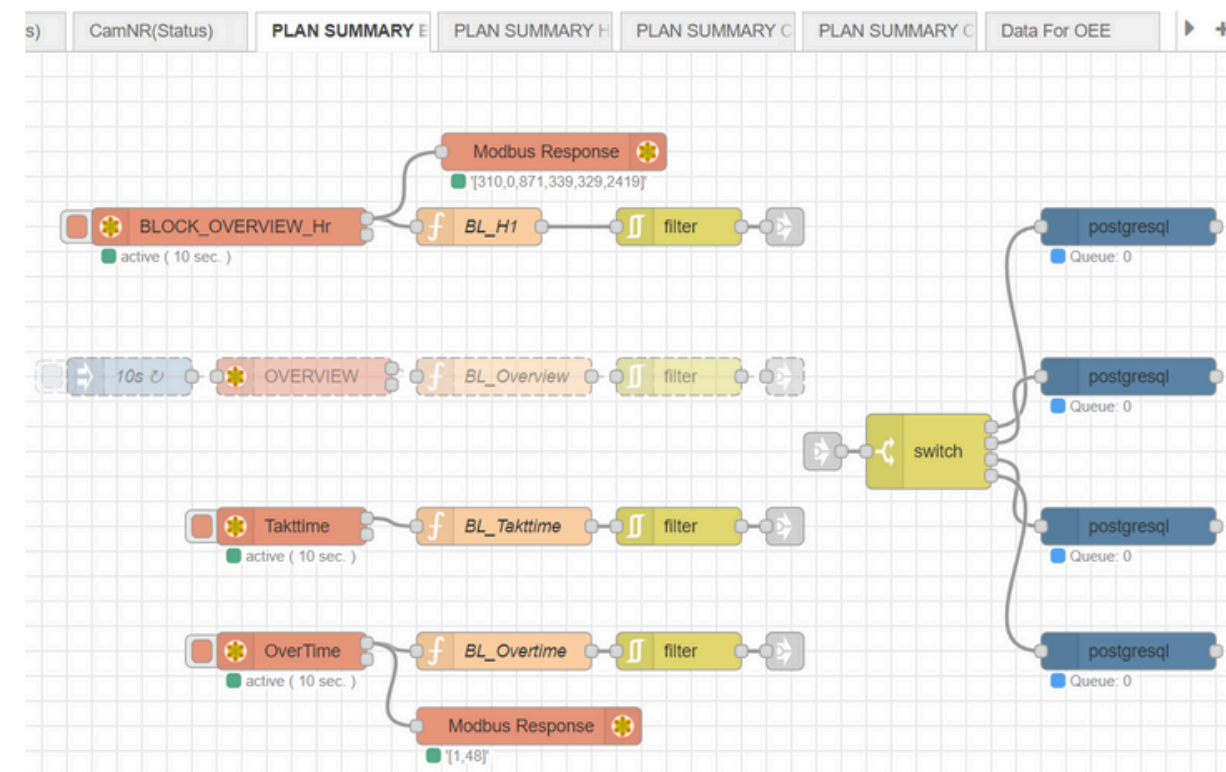
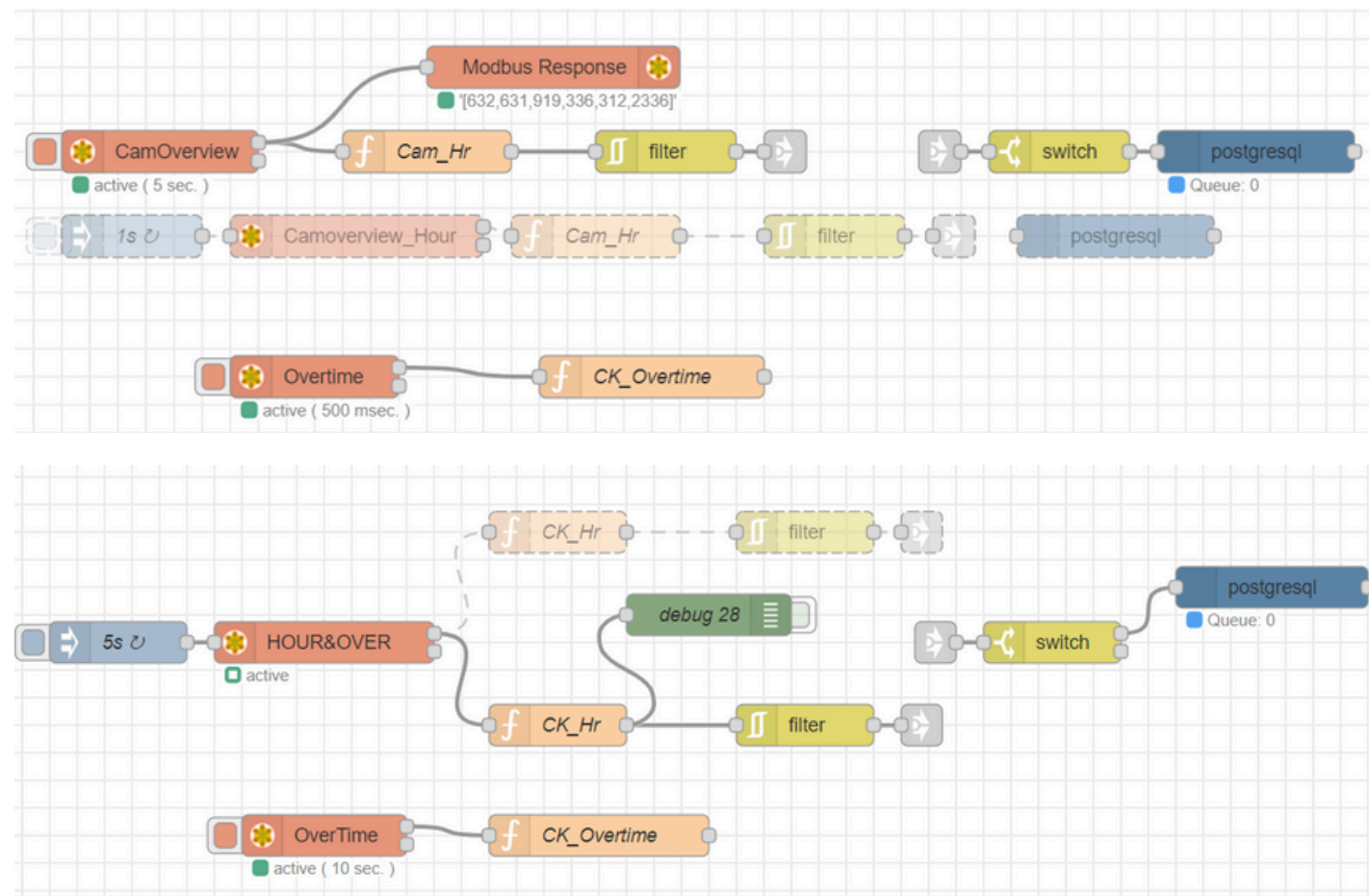
install

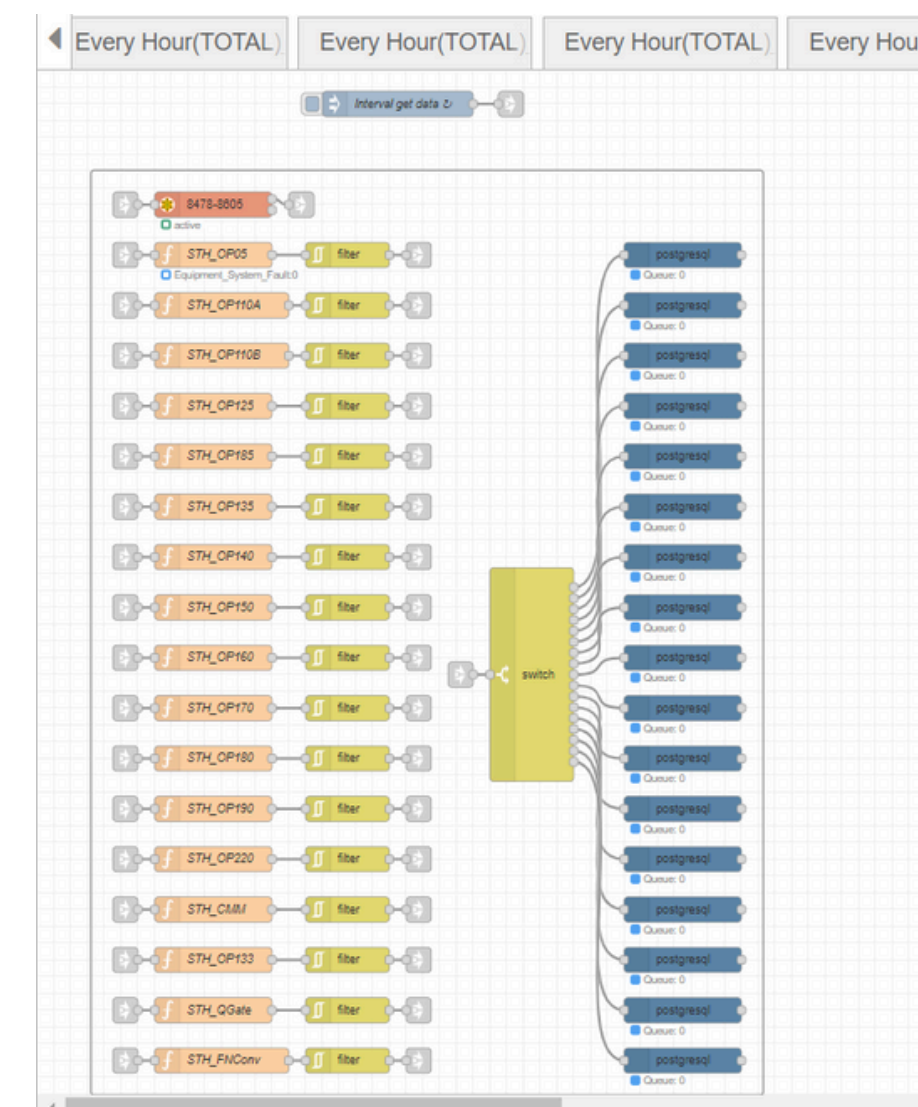
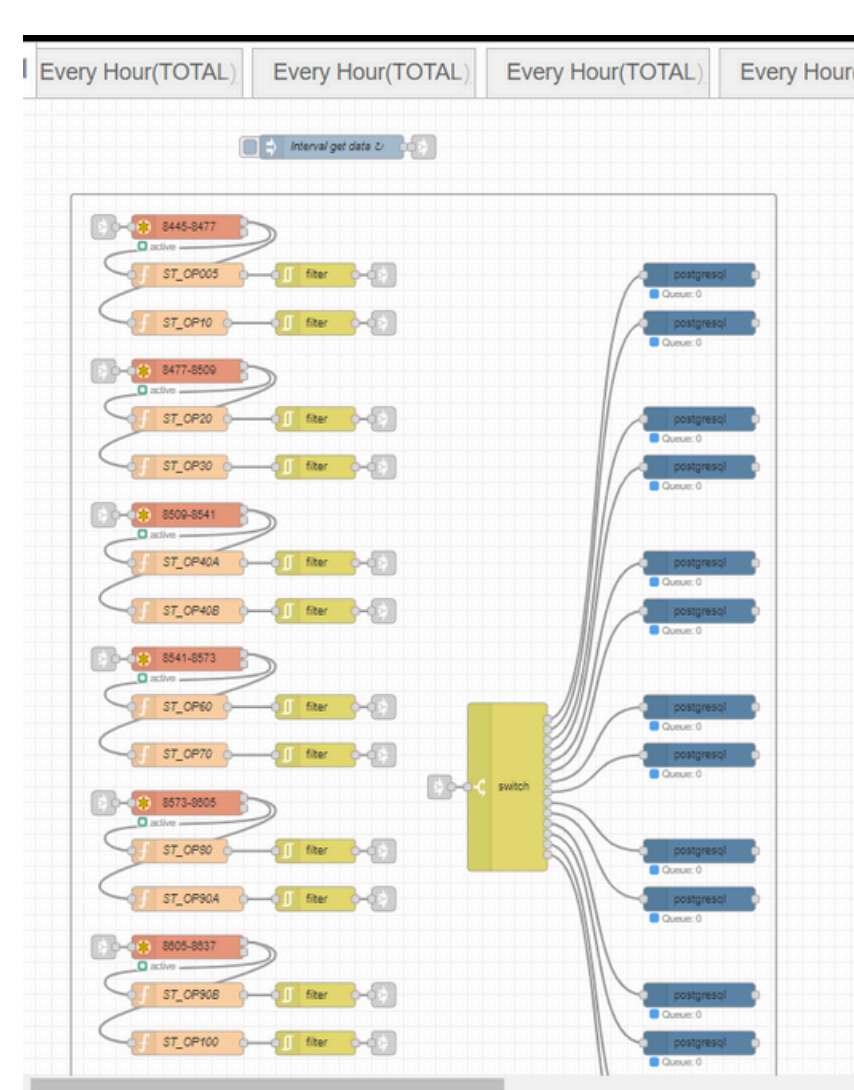
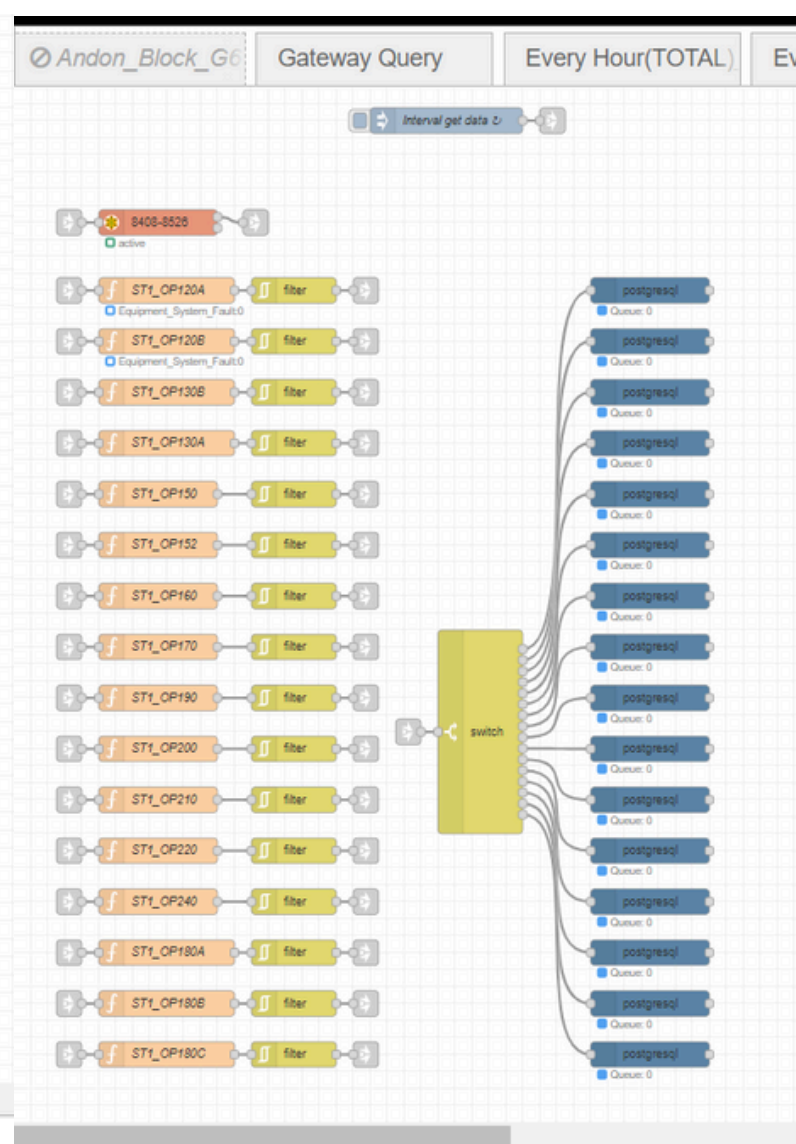
ติดตั้ง Raspberry pi และ
Wi-Fi Extenderติดตั้งที่
Plant 1 Line G6

ภาพผลงาน



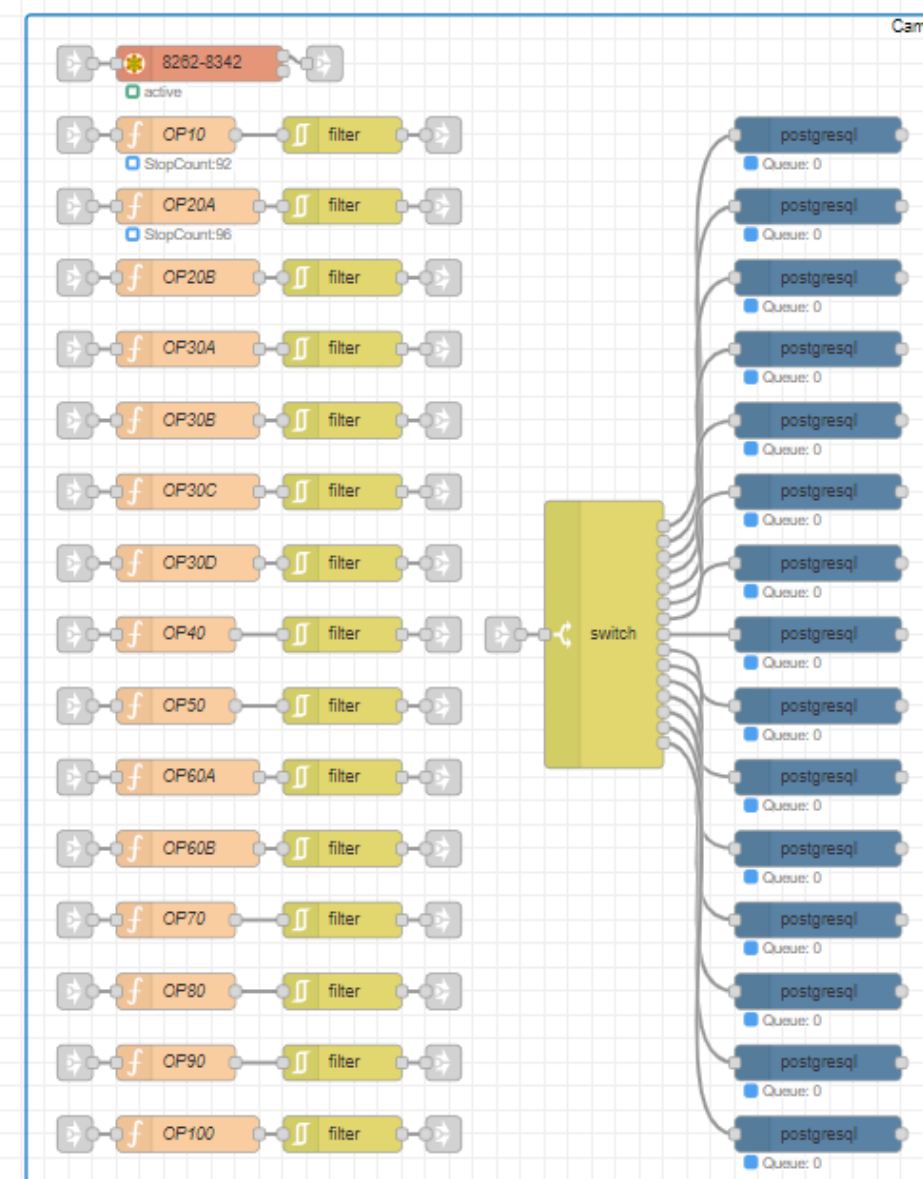
ภาพผลงาน







Interval get data 1



MQTT Publish

MQTT Subscribe

Andon_Block_G6

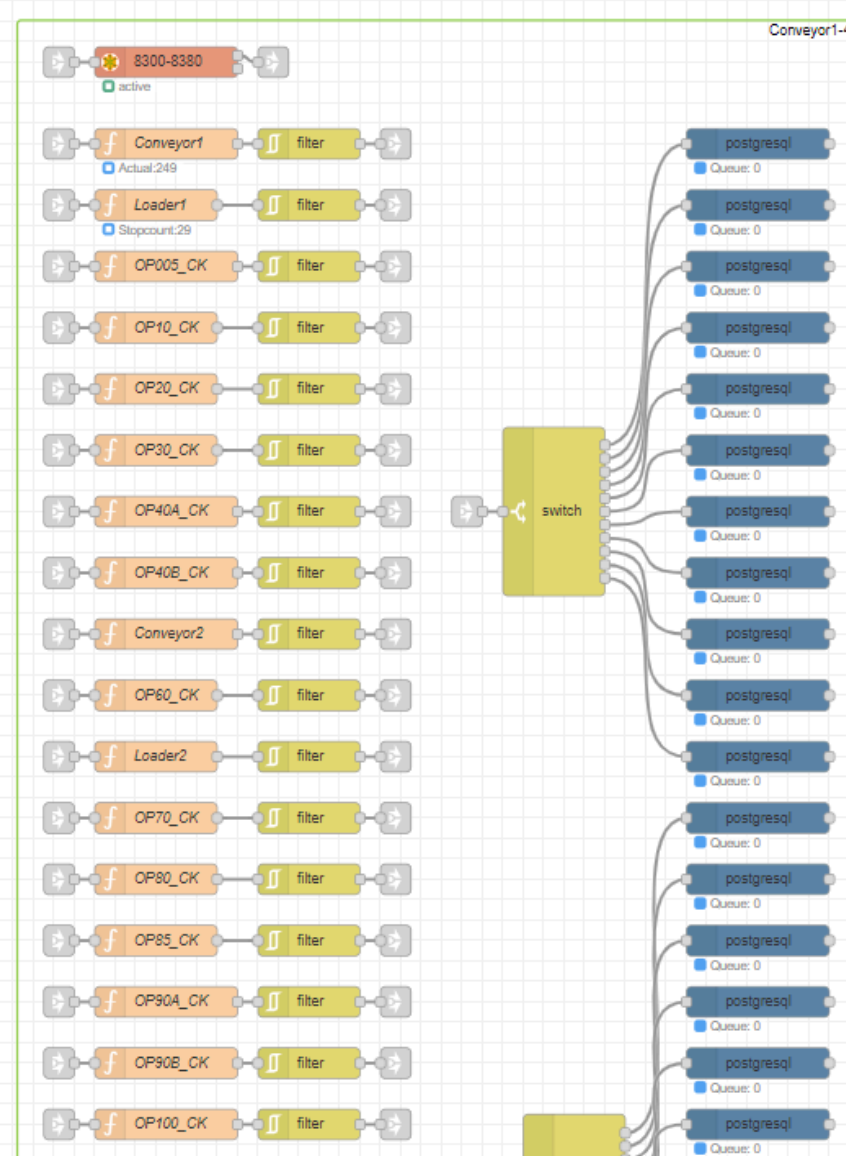
Gateway Query

Every Hour(TOTAL)

Every Hour(TOTAL)

Every Hour(TOTAL)

Interval get data 1



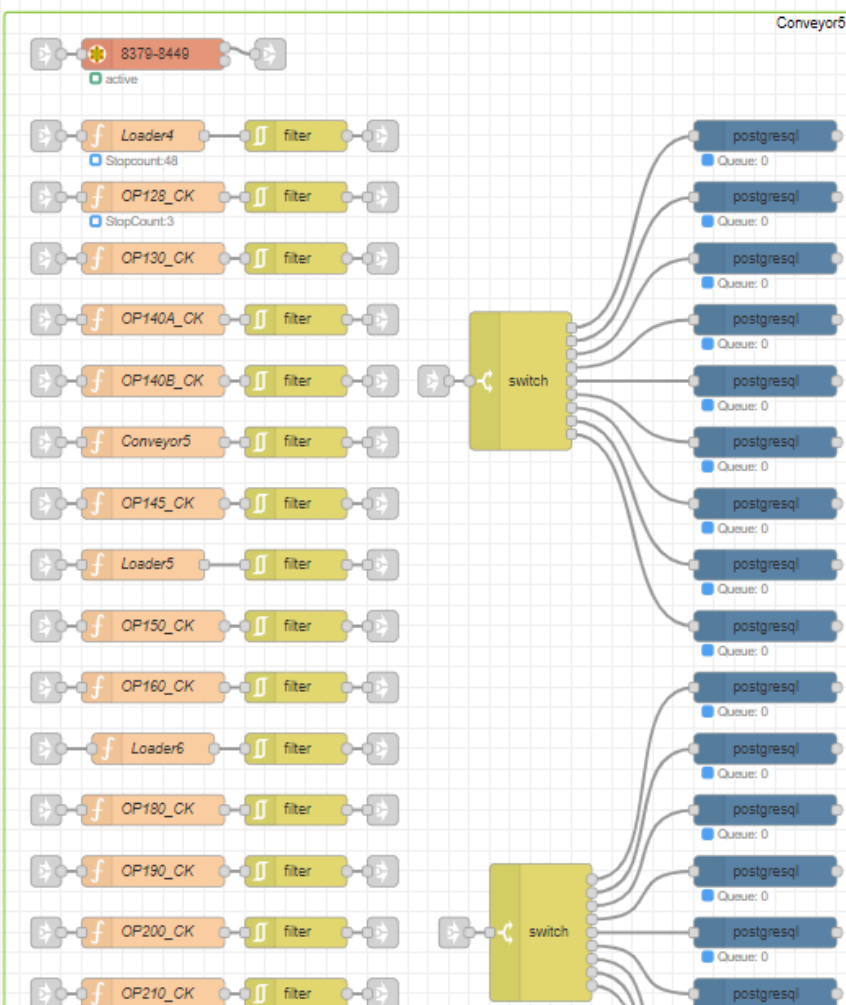
Queue_Query

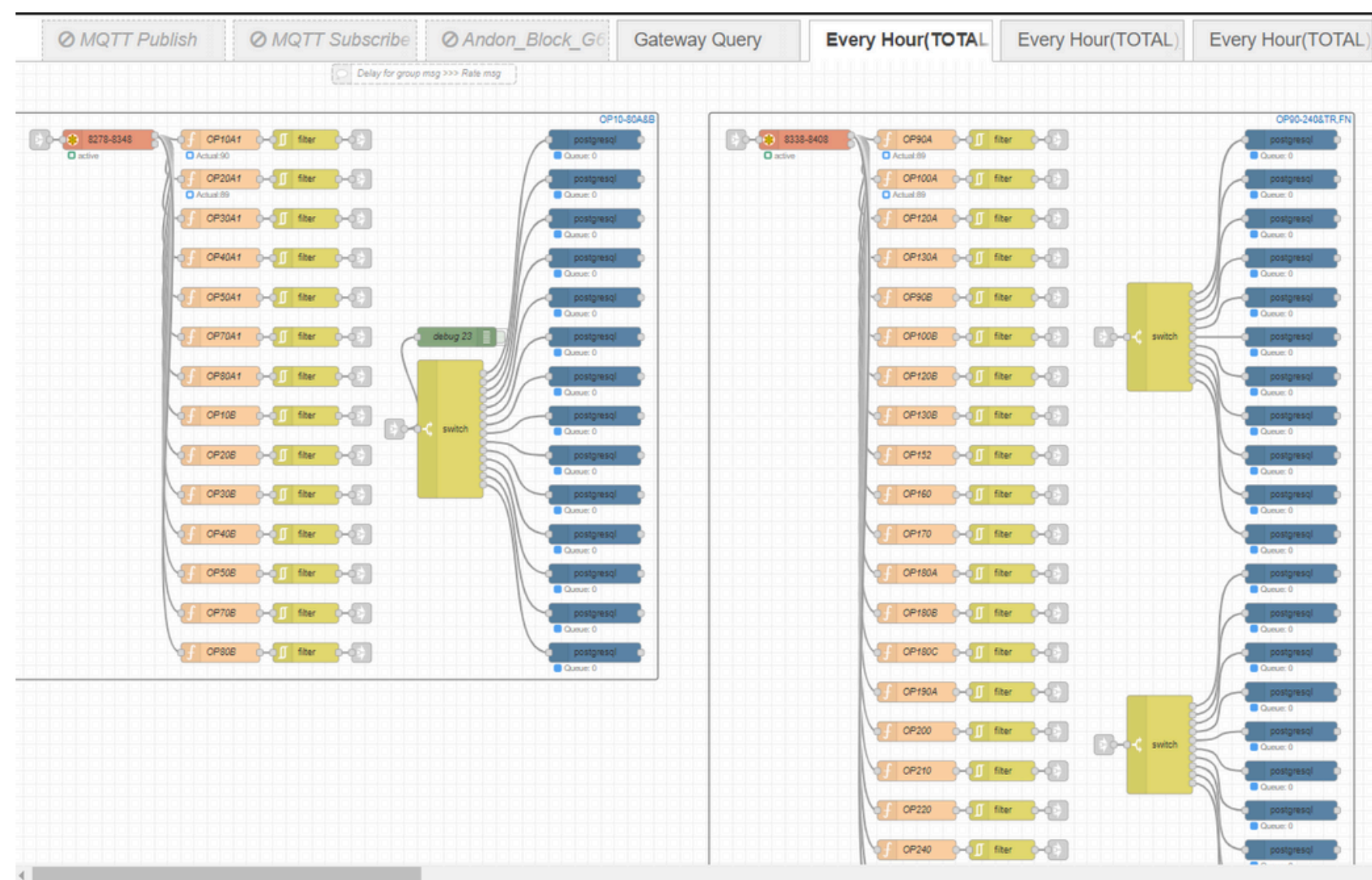
split

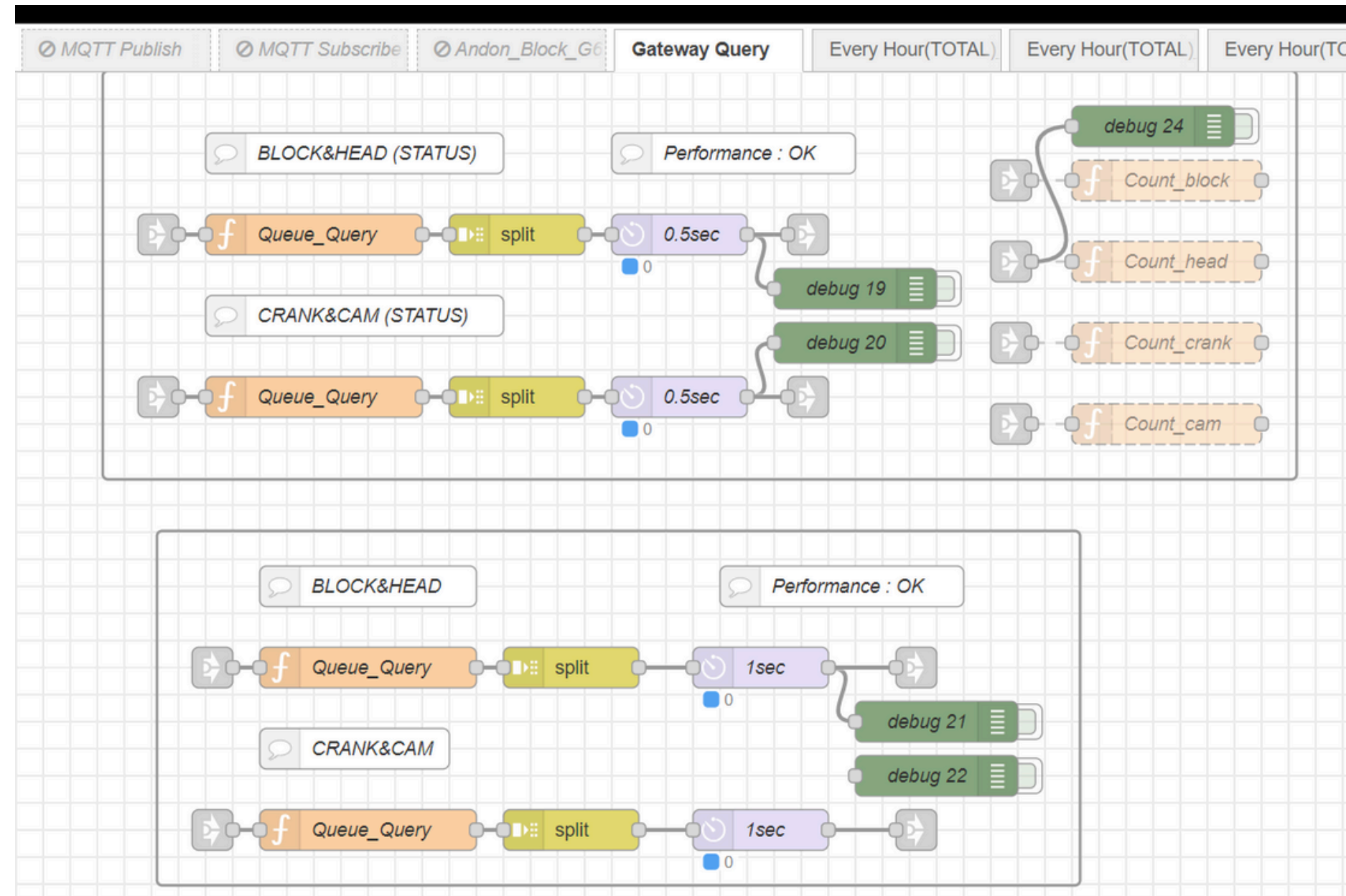
1s

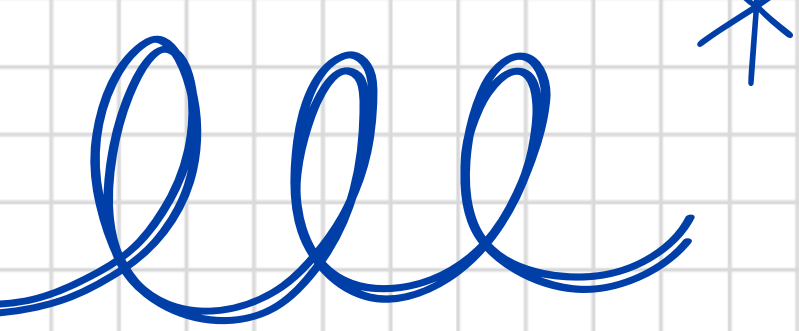
R3s

debug 16









Thank you

