แพลตฟอร์ม IoT ในอุตสาหกรรม และ หน่วยตรวจวัดระยะไกลแบบอเนกประสงค์

Industrial IoT Platform and Universal Remote Terminal Unit

นายบุริศร์ หวังธิติธีรกุล

นายพรชัย อรชร

นายเมธัส วิรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์

บัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ปริญญานิพนธ์ปีการศึกษา 2563

ภาควิชาวิศวกรรมคอมพิวเตอร์

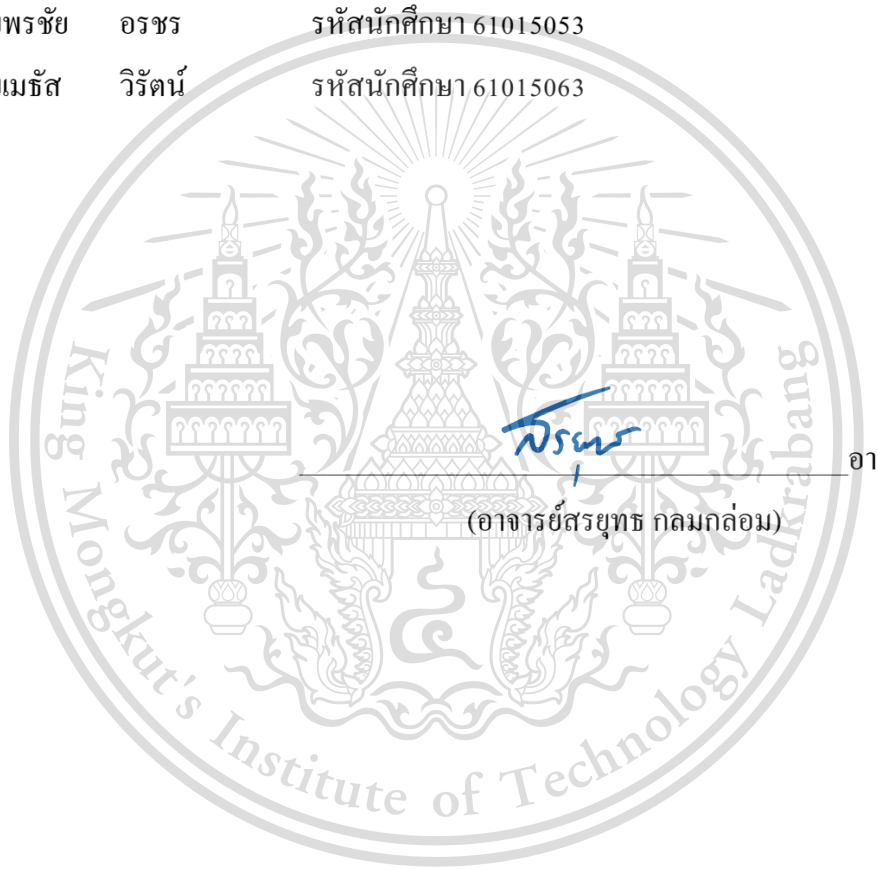
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แพลตฟอร์ม IoT ในอุตสาหกรรม และ หน่วยตรวจวัดระยะไกลแบบอเนกประสงค์

Industrial IoT Platform and Universal Remote Terminal Unit

ผู้จัดทำ

1. นายบุริศร์ หวังธิติธีรกุล รหัสนักศึกษา 60010558
2. นายพรชัย อรชร รหัสนักศึกษา 61015053
3. นายเมธัส วิรัตน์ รหัสนักศึกษา 61015063



_____ อาจารย์ที่ปรึกษา

(อาจารย์สรยุทธ กลมกล่อม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

แพลตฟอร์ม IoT ในอุตสาหกรรม และ
หน่วยตรวจวัดระยะไกลแบบอเนกประสงค์

นายบุริศร์	หวังธิติธีรกุล	60010558
นายพรชัย	อรชร	61015053
นายเมธัส	วิรัตน์	61015063
อาจารย์สรยุทธ	กลมกล่อม	อาจารย์ที่ปรึกษา

ปีการศึกษาที่ 2563

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบ และพัฒนาแพลตฟอร์ม IoT ในอุตสาหกรรม และ หน่วยตรวจวัดระยะไกลแบบอเนกประสงค์ ที่มีลักษณะที่เป็นอุปกรณ์อ่านข้อมูลจากเซนเซอร์ในงานอุตสาหกรรมและต้องทำงานอย่างต่อเนื่องเป็นเวลาติดต่อกันยาวนาน เพราะการตรวจสอบดูแลการทำงานของเครื่องจักรเหล่านี้มีความจำเป็นอย่างยิ่ง เพื่อให้การผลิตเป็นไปอย่างต่อเนื่อง การตรวจสอบดูแลการทำงานของเครื่องจักรหรือเรียกว่า Machine Monitoring System จำเป็นต้องใช้เครื่องมือวัดประเภทต่างๆ ในการตรวจวัดและรายงานผลตลอดจนสามารถวิเคราะห์ผลจากสภาวะการทำงานของเครื่องจักรให้ได้ด้วย โครงงานชิ้นนี้จึงได้พัฒนา Monitoring System ที่สามารถเก็บข้อมูลและวัดประสิทธิภาพการทำงานของเครื่องจักร โดยใช้ IIoT (Industrial Internet of Things) ที่เรียกว่า uRTU (Universal Remote Terminal Unit) เป็นอุปกรณ์สำหรับเก็บรวบรวมข้อมูลและวัดประสิทธิภาพการทำงานของเครื่องจักรต่างๆ โดย uRTU ได้ถูกออกแบบมาให้ใช้งานกับอุปกรณ์ตรวจจับสัญญาณหลายประเภท สามารถเพิ่มหรือลดจำนวนอินพุท ของ uRTU ได้ตามความต้องการ โดยจะนำมาใช้ในการรวบรวมข้อมูลจากแบบจำลอง Output จาก Sensor ในภาคอุตสาหกรรม และส่งข้อมูลที่ได้ไปยัง Server และประมวลผลและวิเคราะห์ตามมาตรฐาน OEE (Overall Equipment Effectiveness) สำหรับเครื่องจักรนั้นๆ แล้วแสดงผลขึ้นบน Dashboard แบบ Realtime เพื่อให้ผู้ประกอบการสามารถตรวจสอบการทำงาน ได้ในทุกที่ทุกเวลา และยังสามารถนำข้อมูล เชิงลึกมาพัฒนาเพื่อเพิ่มประสิทธิภาพในการผลิตให้มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Industrial IoT Platform and Universal Remote Terminal Unit

Mr.Burit	Wangthititheerakul	60010558
Mr.Phornchai	Aorachorn	61015053
Mr.Matus	Wirat	61015063
Mr.Sorayut	Glomglome	Advisor

Academic Year 2563

ABSTRACT

This Project presents the design and development of Industrial IoT Platform and Universal Remote Terminal Unit (IIoT&uRTU). uRTU is data reading device from sensors in Industrial and must work continuously for a long time because monitoring of the operation of these machines is essential so that the production is uninterrupted. Machine monitoring system required use different types of measuring tools in measurement,reporting and analyze results from machine operation. This project was developed Monitoring system that can collect data and measure the performance of the machine using uRTU (Universal Remote Terminal Unit) for collecting data and measuring the performance of various machines. uRTU is designed to be used with a wide variety of signal detectors and the number of inputs output can be increased or decreased as required, It will be used to collect data from sensor output models in the industry and send the received information to server processing and analysis as OEE standard (Overall Equipment Effectiveness) and real time monitoring on dashboard. So the operator can check the work anytime, anywhere and bring insights to develop for increase the efficiency of production even more.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดีด้วยความช่วยเหลือจากหลายฝ่ายทั้งในทางตรงและทางอ้อมปริญญานิพนธ์ฉบับ นี้จะสำเร็จลงไม่ได้หากปราศจากความช่วยเหลือของบุคคลเหล่านี้

ขอขอบคุณ อาจารย์ที่ปรึกษา คือ อาจารย์สรยุทธ กลมกล่อม เป็นผู้ให้คำแนะนำคำปรึกษาและให้ความช่วยเหลือตลอดการทำโครงงาน ซึ่งทำให้การทำงานต่างๆ เป็นไปได้อย่างราบรื่นและสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณอาจารย์และบุคลากรต่างๆ ในสาขาวิชาวิศวกรรมคอมพิวเตอร์ที่ได้ให้คำแนะนำและคำสั่งสอนความรู้ต่างๆ มาโดยตลอด

ขอขอบคุณรุ่นพี่และเพื่อนหลายๆ คน ในภาควิชาวิศวกรรมคอมพิวเตอร์ได้ให้คำแนะนำและคำปรึกษาอย่างเต็มที่และแบ่งปันความรู้ในทุกๆ ด้าน

ในท้ายที่สุดนี้ขอขอบพระคุณ บิดา มารดา และครอบครัวที่เลี้ยงดูสั่งสอน และให้การสนับสนุน พร้อมทั้งให้โอกาสในการศึกษาและให้กำลังใจเสมอมา

บุริศร์ หวังธิติธีรกุล
พรชัย อรชร
เมธัส วิรัตน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญ

หน้า

บทคัดย่อ.....I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญ.....1

1.2 วัตถุประสงค์โครงงาน.....2

1.3 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 2

1.4 ขอบเขตของโครงงาน 2

1.5 ขั้นตอนการดำเนินงาน.....2

1.6 ตารางการดำเนินงาน 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....5

2.1 ทฤษฎีที่เกี่ยวข้อง.....5

2.1.1 หน่วยตรวจวัดระยะไกลยูนิเวอร์แซล.....5

2.1.2 Modbus Protocol.....5

2.1.3 MQTT.....6

2.1.4 TIG Stack.....7

2.2 งานวิจัยที่เกี่ยวข้องที่เกี่ยวข้อง.....9

2.2.1 Design and Development of Modbus/MQTT Gateway for Industrial IoT Cloud Applications Using Raspberry Pi.....9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

สารบัญ (ต่อ)

หน้า

2.2.2 Lightweight & Secure Industrial IoT Communications via the MQ Telemetry Transport Protocol 10

บทที่ 3 การออกแบบและการพัฒนา.....11

3.1 ภาพรวมการทำงานในระบบ 11

3.1.1 uRTU.....11

3.1.2 Simulator.....11

3.1.3 TIG Stack.....11

3.2 การทำงานโดยรวมของระบบ Monitoring เมื่อมีการเปิดใช้งาน uRTU จะมีขั้นตอนการทำงานและส่งข้อมูล..... 11

3.3 โครงสร้างของระบบ.....12

3.3.1 uRTU.....12

3.3.2 Simulator.....12

3.3.3 TIG Stack.....13

3.4 Flowchart การทำงานของระบบ.....14

3.5 Use case Diagram.....15

3.6 ER diagram.....15

บทที่ 4 การทดลอง.....16

4.1 การทดลองส่งข้อมูลผ่านโปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบINT และ Float.....16

4.2 การทดลองส่งข้อมูลผ่านโปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบJSON…….....18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญ (ต่อ)

หน้า

4.3 การทดลองส่งข้อมูลผ่านโปรโตคอล MQTT จากเครื่องคอมพิวเตอร์ Client ขึ้นไปบน Server……..…...……19
4.4 การทดลองปรับค่าความถี่ของ PWM ผ่านแอพพลิเคชั่นBlynk.....21
4.5 การทดลองควบคุม I/O บน STM32 ด้วย Web server และ LwIP.....21
4.6 การทดลองส่งข้อมูลผ่านโปรโตคอลMQTT ด้วย LwIP จาก STM32 ขึ้นไปบน Server.....22
4.7 การทดลองการทำงานแบบ Multi Task บน FreeRTOS.....23
4.8 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จาก Mega 2560 ขึ้นไปบน Server.....24
4.9 การทดลองอ่านค่า Modbus RTU จาก Simulator ด้วย Mega 2560.....25
4.10 การทดลองคำนวณมาตรฐาน OEE โดยใช้ข้อมูลจาก InfluxDB.....26

บทที่ 5 สรุปผลการดำเนินงานและแนวทางการพัฒนา.....28
5.1 ผลการดำเนินงาน.....25
5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข.....28
5.3 แนวทางการพัฒนา.....29

บรรณานุกรม.....30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญรูป

รูปที่ หน้า

1.1 ตารางการดำเนินการ....4

2.1 การสื่อสารระหว่าง Master กับ Slave....6

2.2 MQTT....7

2.3 ส่วนปรกอบของ TIG Stack....8

2.4 แผนภาพแสดงภาพรวมระดับสูงของการออกแบบและสถาปัตยกรรมเกตเวย์ Modbus / MQTT บนคลาวด์....9

2.5. กราฟแสดงการเปรียบเทียบการใช้ทรัพยากรและพลังงาน....10

3.1ไดอะแกรมของระบบ....12

3.2 Application Blynk....13

3.3 Grafana Dashboard....13

3.4 Flowchart การทำงานของระบบ....14

3.5 Use case Diagram....15

3.6 ER Diagram....15

4.1 Grafana Dashboard....17

4.2 ผลทดลองการส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบ INT และ Float....17

4.3 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบ JSON....18

4.4 ผลทดลองการส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบJSON....19

4.5 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จากเครื่องคอมพิวเตอร์ Client ขึ้นไปบนServer.20

4.6 ผลทดลองการส่งข้อมูลผ่าน โปรโตคอล MQTT จากเครื่องคอมพิวเตอร์ Client ขึ้นไปบน....20

4.7 ผลทดลองการปรับค่าความถี่ของ PWM ผ่านแอพพลิเคชั่นBlynk....21

4.8 การทดลองควบคุม I/O บน STM32 ด้วย Web server และ LwIP....22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญรูป (ต่อ)

4.9 ผลทดลองการควบคุม I/O บน STM32 ด้วย Web server และ LwIP.....22
4.10 ผลการทดลองโปรแกรม uRTU แบบ Multi Tasks โดยใช้ FreeRTOS.....24
4.11 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จาก Mega 2560 ขึ้นไปบน Server.....24
4.12 ผลการทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จาก Mega 2560 ขึ้นไปบน Server.....25
4.13 ผลการทดลองอ่านค่า Modbus RTU จาก Simulator ด้วย Mega 2560.....26
4.14 ผลการแสดงข้อมูลจาก InfluxDB.....27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันประเทศไทย เป็นฐานการผลิตสินค้าเพื่อใช้สอยภายในประเทศและส่งออกไปยังต่างประเทศ โดยมีทั้งธุรกิจของคนไทยและธุรกิจของต่างประเทศ ด้วยการขยายตัวอย่างต่อเนื่องของภาคอุตสาหกรรมทำให้การผลิตหลายๆอย่าง มีการนำเครื่องจักรเข้ามาใช้งานแทนแรงงานคน ถึงแม้ว่าเครื่องจักรจะช่วยในการเพิ่มประสิทธิภาพการผลิต แต่ก็ยังจำเป็นต้องมีคนคอยควบคุมงาน ณ สายการผลิตอยู่ดี เพื่อไม่ให้เกิดความผิดพลาดหรือความสูญเสียเมื่อเครื่องจักรนั้นทำงานผิดพลาด ในอุตสาหกรรมที่มีเครื่องจักรติดตั้งใช้งานและต้องทำงานอย่างต่อเนื่องเป็นเวลาติดต่อกันยาวนาน การตรวจสอบดูแลการทำงานของเครื่องจักรเหล่านี้มีความจำเป็นอย่างยิ่ง เพื่อให้การผลิตเป็นไปอย่างต่อเนื่อง การตรวจสอบดูแลการทำงานของเครื่องจักรหรือเรียกว่า Machine Monitoring System จำเป็นต้องใช้เครื่องมือวัดประเภทต่างๆ ในการตรวจวัดและรายงานผลตลอดจนสามารถวิเคราะห์ผลจากสภาวะการทำงานของเครื่องจักรให้ได้ด้วย

ด้วยกระแสของ Digital Transformation ที่กำลังเกิดขึ้น ทำให้เทคโนโลยีเข้ามามีบทบาทสำคัญในทุกอุตสาหกรรมและทุกธุรกิจ โดยเฉพาะด้านการผลิต ทำให้ข้อมูลเกี่ยวกับการผลิตแต่ละขั้นตอนสามารถเข้าถึงเพื่อใช้งานและสามารถแบ่งปันข้อมูลได้อย่างปลอดภัยจากทุกที่และทุกอุปกรณ์ โดยมี IIoT Platform เป็นส่วนสำคัญ

โครงงานชิ้นนี้จึงได้พัฒนา Monitoring System ที่สามารถเก็บข้อมูลและวัดประสิทธิภาพการทำงานของเครื่องจักร โดยใช้ IIoT ที่เรียกว่า uRTU เป็นอุปกรณ์สำหรับเก็บรวบรวมข้อมูลและวัดประสิทธิภาพการทำงานของเครื่องจักรต่างๆ โดย uRTU ได้ถูกออกแบบมาให้ใช้งานกับอุปกรณ์ตรวจจับสัญญาณหลายประเภท สามารถเพิ่มหรือลดจำนวน Input โดยการเพิ่มหรือลดโมดูลขยาย (Expansion Module) ของ uRTU ได้ตามความต้องการ โดยจะนำมาใช้ในการรวบรวมข้อมูลจากแบบจำลอง Output จาก Sensor ในภาคอุตสาหกรรม และส่งข้อมูลที่ได้ไปยัง Server และให้ Server ทำการประมวลผลและวิเคราะห์มาตรฐาน OEE สำหรับเครื่องจักรนั้นๆ แล้วแสดงผลขึ้นบน Dashboard แบบ Realtime เพื่อให้ผู้ประกอบการสามารถตรวจสอบการทำงาน ได้ในทุกที่ทุกเวลา และยังสามารถนำข้อมูล เชิงลึกมาพัฒนาเพื่อเพิ่มประสิทธิภาพในการผลิตให้มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาการทำงานและสร้าง Monitoring System ในภาคอุตสาหกรรม ที่มีประสิทธิภาพ

2) เพื่อศึกษาวิธีการส่งข้อมูลของ IIoT Platform

3) สร้างระบบที่สามารถตรวจสอบการทำงานของเครื่องจักรในภาคอุตสาหกรรม ด้วยมาตรฐาน OEE

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1) ภาคอุตสาหกรรมสามารถตรวจสอบและวัดผลการทำงานของเครื่องจักรโดยไม่ต้องอยู่ในสายการผลิต

2) Monitoring System ที่สามารถใช้งานได้อย่างมีประสิทธิภาพ และมีความยืดหยุ่นในการนำไปใช้ให้เหมาะสมกับงานแต่ละประเภท

3) ฝึกฝนทักษาะทางด้าน การทำ IIOT และ Monitoring System

1.4 ขอบเขตของโครงงาน

1) สร้างแบบจำลอง Output จาก Sensor ในภาคอุตสาหกรรม และ Web Application สำหรับควบคการทำงานแบบ Realtime

2) สร้างอุปกรณ์ uRTU ที่สามารถรวบรวม Output จาก Sensor แล้วส่งข้อมูลไปยัง Server แบบ Realtime

3) สร้าง Dashboard ที่สามารถคำนวณและแสดงผลข้อมูลแบบ Realtime

4) สร้าง Monitoring System ที่มีการวัดมาตรฐาน OEE

1.5 ขั้นตอนการดำเนินงาน

1) เสนอแนวทางและเลือกใช้เทคโนโลยีที่เหมาะกับโครงงาน

1.1) uRTU พัฒนาบนบอร์ด Arduino Mega 2560 ที่ใช้ชิพ ATmega2560

1.2) TIG Stack เพื่อใช้ในการประมวลผลและแสดงผลข้อมูลที่อ่านได้จาก uRTU

1.3) MQTT เป็นโปรโตคอลในการสื่อสารระหว่าง uRTU และ TIG Stack

1.4) Modbus RTU เป็นโปรโตคอลสื่อสารระหว่าง เครื่องจักรอุตสาหกรรมและ uRTU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2) ศึกษาเทคโนโลยีที่เกี่ยวข้อง

2.1) FreeRTOS

2.2) TIG Stack (Telegraf , Influx DB ,Grafana Dashboard)

2.3) MQTT

2.4) Modbus RTU

3) ศึกษาและออกแบบระบบโดยรวม

3.1) ระบบจำลอง Output จากเครื่องจักรอุตสาหกรรม (Simulator)

3.2) ระบบรวบรวมและส่งข้อมูลจาก Sensor ไปยัง Server โดยใช้ uRTUที่ประกอบด้วย FreeRTOS

3.3) ระบบแสดงผลข้อมูลและคำนวณ OEE แบบ Realtime

4) สร้างระบบย่อยได้แก่

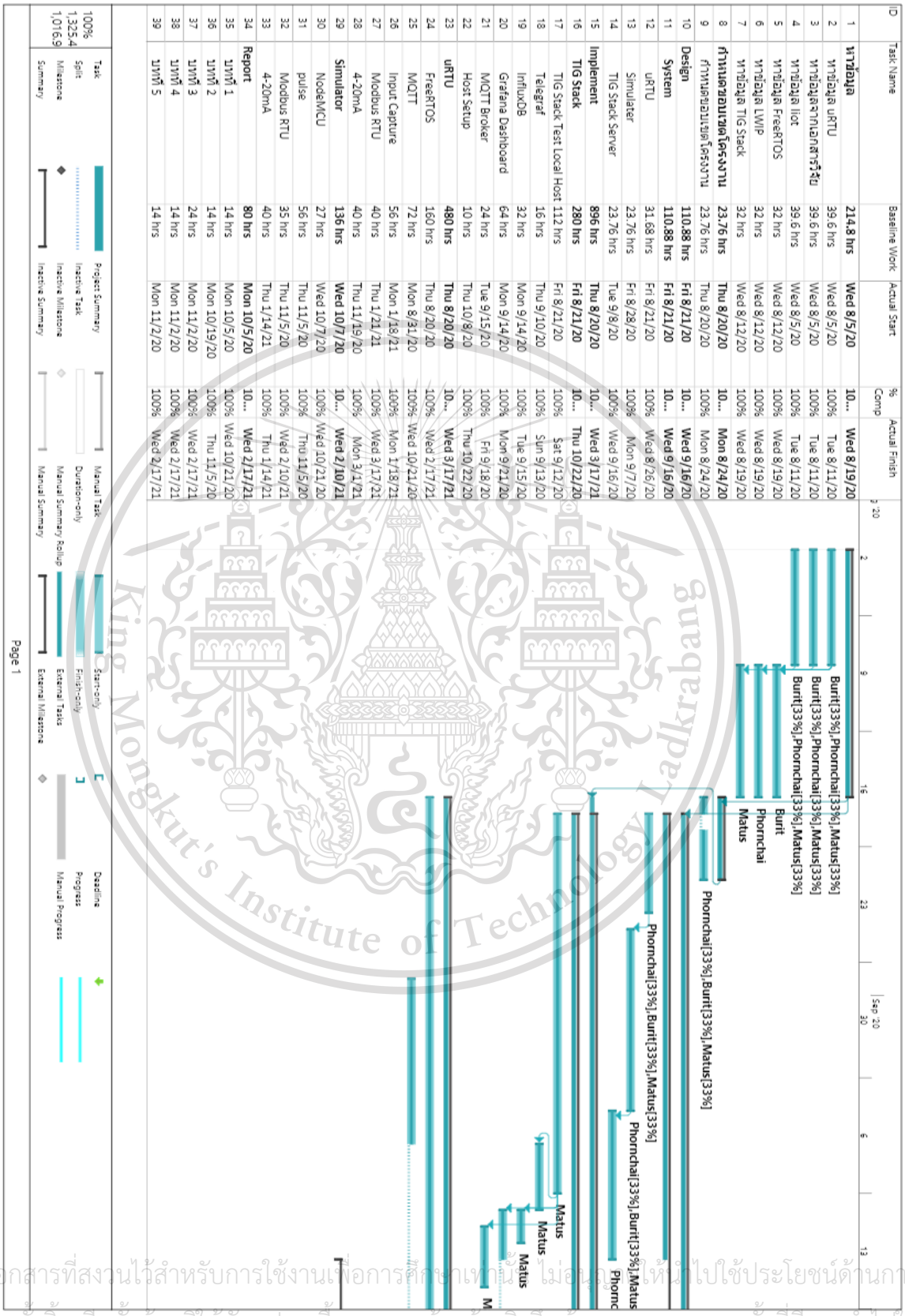
4.1) แบบจำลอง Output จากเครื่องจักรอุตสาหกรรม

4.2) ระบบรวบรวมและส่งข้อมูลจาก Sensor ไปยัง Server โดยใช้ uRTU

4.3) ระบบแสดงผลข้อมูลและคำนวณ OEE แบบ Realtime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

1.6 ตารางการดำเนินงาน



รูปที่ 1.1 ตารางการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 หน่วยตรวจวัดระยะไกลยูนิเวอร์แซล

หน่วยตรวจวัดระยะไกลยูนิเวอร์แซล หรือ Universal Remote Terminal Unit (uRTU) เป็นอุปกรณ์ที่พัฒนามาจาก Remote Terminal Unit (RTU) เป็นอุปกรณสำหรับการตรวจสอบระยะไกลและควบคุมอุปกรณ์ภาคสนาม และเชื่อมต่อกับระบบควมคุมโรงงานหรือ SCADA (Supervisory Control & Data Acquisition System) ซึ่งมีปัญหาในการเพิ่ม และปรับเปลี่ยนช่องสัญญาณ และคุณสมบัติได้ โดย uRTU ถูกพัฒนาให้สามารถเพิ่มหรือลดExpansion module ได้ตามความต้องการ โดยมีคุณลักษณะ ดังนี้

1. สามารถเปลี่ยนสัญญาณข้อมูล Analog ในรูปแบบสัญญาณมาตรฐานทางไฟฟ้าที่ได้รับจาก Sensor เช่น กระแสไฟฟ้า แรงดัน ความถี่ ให้เป็นข้อมูล Digital เพื่อใช้ในการประมวลผลหรือเปลี่ยนข้อมูล Digital ที่ได้รับจากคอมพิวเตอร์เพื่อใช้สั่งงานการเปิดปิดอุปกรณ์ต่างๆ ทั้งยังสามารถเก็บข้อมูลโดยอัตโนมัติ (Data logger)

2. สามารถติดต่อกับคอมพิวเตอร์ผ่านช่องทางการสื่อสารต่างๆได้ เช่น LAN, WiFi, 3/4G, Long Range (LoRa) เป็นต้น

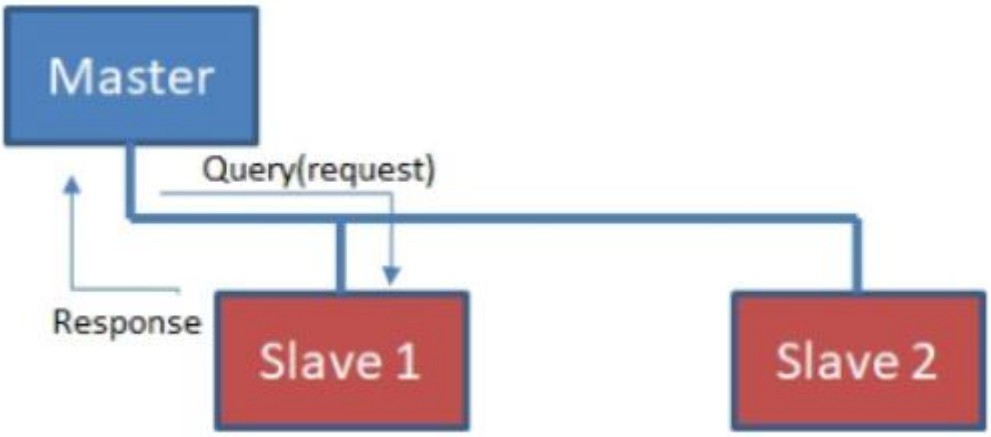
2.1.2 Modbus Protocol

Modbus คือ โปรโตคอล (Protocol) ซึ่งเป็นการส่งข้อมูลระหว่างอุปกรณ์อิเล็กทรอนิกส์แบบดิจิตอลอนุกรม อุปกรณ์หลักที่ต้องการข้อมูลเรียกว่า Modbus Master (Server) และอุปกรณ์ที่ส่งข้อมูลเรียกว่า Modbus Slave (Client) โดยใน Modbus Network ที่เป็นมาตรฐานนั้นจะมีอุปกรณ์ Master ตัวเดียวส่วนอุปกรณ์ Slave มีได้มากถึงได้ถึง 247 ตัว โดยอุปกรณ์ Slave แต่ละตัวจะมี ID ระบุตั้งแต่ 1 ถึง 247 และ Master สามารถเขียนข้อมูลไปยังอุปกรณ์ Slave ได้

Modbus เป็น Open Protocol ที่บุคคลทั่วไปสามารถใช้หรือพัฒนาได้โดยไม่มีค่าใช้จ่าย Modbus จึงเป็น ที่นิยมและใช้อย่างแพร่หลายในระบบอุตสาหกรรม

Slave เป็นอุปกรณ์ต่อพ่วงอื่นๆ เช่น วาล์ว (Valve), Input/Output Transducer, Inverter, อุปกรณ์บันทึกข้อมูล (Data Logger) หรือเป็นอุปกรณ์เครื่องมือวัดอื่นๆ ซึ่งทำการประมวลผลและส่งข้อมูลไปให้ Master

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 2.1 การสื่อสารระหว่าง Master กับ Slave

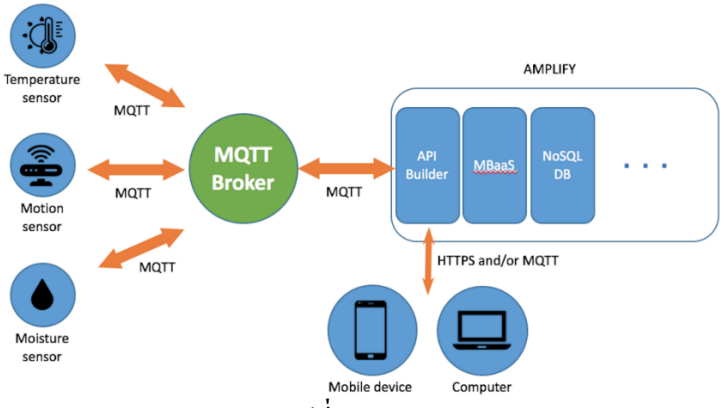
Master เป็นอุปกรณ์ที่สามารถติดต่อกับ Slave แต่ละตัวได้หรือสามารถส่งเป็น Message ถึง Slave ทุกตัว และอุปกรณ์ Slave จะตอบสนองสิ่งที่อุปกรณ์ Master ต้องการ สิ่งที่อุปกรณ์ Master ส่งให้จะประกอบไปด้วย Slave Address, Function Code, Data และ Checksum ส่วนข้อมูลที่ Slave ส่งกลับมาคือคำสั่งที่สั่งให้กระทำหรือข้อมูลต่างๆ

Modbus แบ่งออกเป็น Modbus Serial: ASCII/RTU (เป็นพอร์ตการสื่อสารแบบอนุกรม RS232, RS485, RS422) และ Modbus TCP/IP (LAN) เพื่อให้สอดคล้องกับแนวโน้มการพัฒนาการสื่อสารในปัจจุบันและทุกอย่างที่สามารถเชื่อมต่อกับเครือข่าย Ethernet หรือ Internet เพื่อส่งข้อมูล

2.1.3 MQTT

MQTT (Message Queue Telemetry Transport) คือ โปรโตคอลในการส่งข้อมูลที่ถูกพัฒนามาเพื่อให้ใช้ในระบบ IoT มันทำงานแบบ Broker and Clients Network ถูกออกแบบให้สามารถส่งข้อมูลแบบ Real-Time ในปริมาณข้อมูลที่น้อย ทำให้ใช้พลังงานต่ำและถูกพัฒนามาจาก TCP/IP ที่มีการส่งข้อมูลแบบ One-To-One ทำให้สิ้นเปลืองทรัพยากรณ์มากจึงไม่เหมาะกับระบบ IoT เนื่องจากระบบ IoT มีการส่งข้อมูลตลอดเวลา 1 อุปกรณ์อาจรับหรือส่งข้อมูลไปยังหลายอุปกรณ์ หรือการส่งข้อมูลแบบ One-To-All โดยอุปกรณ์ทุกตัวที่ทำการ Subscriber ไปยัง Topic บน Broker จะได้รับข้อมูลที่ Publisher ส่งให้ Topic นั้นๆบน Broker ทั้งหมด โดยมันถูกคิดค้นเมื่อปี 1999 โดย Andy Stanford-Clark จาก IBM และ Arlen Nipper จาก Cirrus Link โดยถูกใช้เพื่อตรวจสอบท่อส่งน้ำมันในทะเลทรายโดยเป้าหมายคือ เป็นโปรโตคอลที่มีประสิทธิภาพสูง ส่งข้อมูลขนาดไม่มาก ใช้พลังงานต่ำ เนื่องจากต้องเชื่อมต่อผ่านดาวเทียมซึ่งมีราคาในการส่งข้อมูลสูงมากในขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 2.2 MQTT

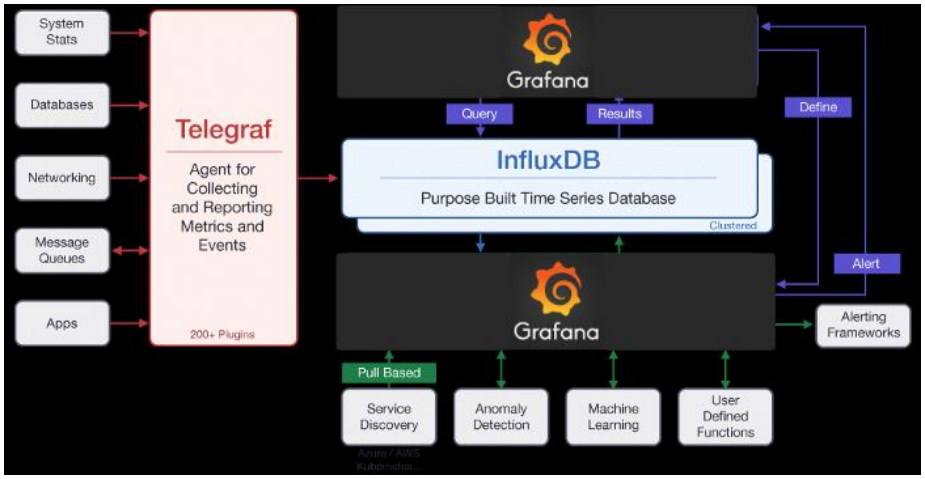
MQTT ประกอบด้วย

- Broker (Server) คือตัวกลางในการรับข้อมูลจาก Publisher และส่งข้อมูลให้กับ Subscriber
- Clients (Subscriber / Publisher) คือ ผู้ใช้งาน MQTT
- Publisher คือ ตัวส่งข้อมูลให้กับ Topic ที่อยู่ใน Broker เรียกว่าการ Publish
- Subscriber คือ ตัวรับข้อมูลจาก Topic ที่อยู่ใน Broker เรียกว่าการ Subscribe
- Topic คือ หัวเรื่องที่ต้องการรับส่งข้อมูล ระหว่าง Publisher กับ Subscriber

2.1.4 TIG Stack

เมื่อแพลตฟอร์ม (platform) หรือแอปพลิเคชันของเราเริ่มมีผู้ใช้มากขึ้นสิ่งหนึ่งที่สำคัญและขาดไม่ได้คือการตรวจสอบประสิทธิภาพ (monitoring) ของแอปพลิเคชันหรือเครื่องเซิฟเวอร์ที่ทำหน้าที่ให้บริการแอปพลิเคชันแก่ผู้ใช้ของเราว่าได้ถูกใช้อย่างคุ้มค่าพอแล้วหรือไม่ควรที่จะได้เวลาเพิ่มความสามารถของเครื่องเดิมโดยการเพิ่ม Memory, Hardisk ให้แก่เครื่องเดิมหรือที่เรียกว่า Vertical Scaling หรือจะเพิ่มขึ้นแบบเอาเครื่องหลายๆ เครื่องมาช่วยกันประมวลผล Horizontal Scaling ทั้งสองทางเลือกนี้ก็เป็นหนึ่งในตัวเลือกที่ช่วย Scale แอปของเราได้ดีทีเดียวขึ้นอยู่กับสถานการณ์และความคุ้มค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 2.3 ส่วนประกอบของ TIG Stack

TIG Stack ประกอบด้วย

Telegraf ตัวนี้เป็นตัวที่ทำหน้าที่คอยเก็บข้อมูลบนเครื่องของเรา (agent) เช่น cpu, memory, disk i/o, processes หรือแม้กระทั่งแอปพลิเคชันต่างๆ ที่รันอยู่บนเครื่องของเราเช่น Nginx, Apache, Docker, K8 และทำการส่งข้อมูลไปเก็บยังจุดหมาย (endpoint)ที่เรากำหนดไว้เช่น InfluxDB, MySQL, MongoDB, JSON และอื่นๆ

InfluxDB ทำหน้าที่เป็น Time Series Database คอยเก็บข้อมูลที่จะไหลเข้ามาจาก Telegraf นั่นเองหรือข้อมูลอะไรก็ได้ซึ่งด้วยความที่มันเป็น Time Series Database ทำให้มันมีความสามารถจัดการกับข้อมูลที่เป็นวันและเวลาได้อย่างรวดเร็วกว่าพวก database ทั่วไปแถมยังสามารถทำ Retention Policy ที่สามารถกำหนดอายุของข้อมูลที่จะอยู่ใน database ได้เช่น 3 วันเป็นต้น

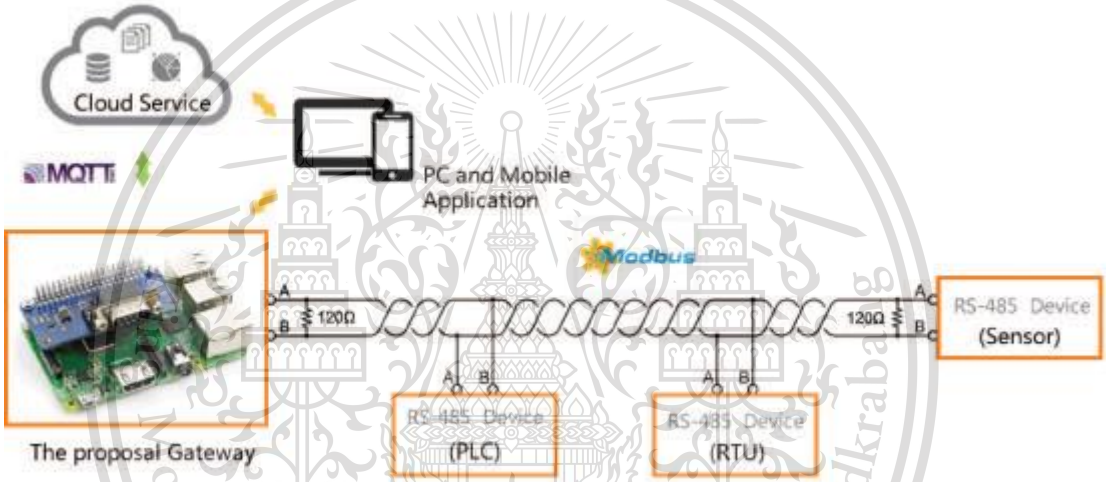
Grafana คือ dashboard tools ซึ่งจะแสดงผลออกมาเป็นพวก กราฟข้อมูล metrix ต่างๆ โดยความเทพของมันคือมันจะดึงข้อมูลออกมาได้ในระดับ realtime โดยมันสามารถดึงข้อมูลมาจาก datasource ที่เป็นที่นิยมได้อย่างมาก เช่น influxdb, Prometheus, elasticsearch, AWS CloudWatch และ datasource ตัวอื่นที่เป็นที่นิยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.2 งานวิจัยที่เกี่ยวข้องที่เกี่ยวข้อง

2.2.1 Design and Development of Modbus/MQTT Gateway for Industrial IoT Cloud Applications Using Raspberry Pi

เป็นงานวิจัยเกี่ยวกับการทำ IoT Gateway โดยใช้ Raspberry Pi ในการดึงข้อมูลจาก RS485 Device (Sensor) ในภาคอุตสาหกรรม ผ่านโปรโตคอล Modbus RTU และนำขึ้น Clound Service เช่น Amazon AWS , Alibaba Cloud หรือ MQTT Broker โดยใช้โปรโตคอล MQTT ส่งในรูปแบบ JSON และให้ Cloud Service สื่อสารกับ PC หรือ Mobile Device ผ่านทาง MQTT เพื่อแสดงผลข้อมูล



รูปที่ 2.4 แผนภาพแสดงภาพรวมระดับสูงของการออกแบบและสถาปัตยกรรมเกตเวย์ Modbus / MQTT บนคลาวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.2.2 Lightweight & Secure Industrial IoT Communications via the MQ Telemetry Transport Protocol

เป็นงานวิจัยที่เกี่ยวของกับการสื่อสารของ IoT ผ่านโปรโตคอลต่างๆ รายละเอียดของโปรโตคอลนั้นๆ และการเปรียบเทียบของโปรโตคอลทางด้าน IoT หลักๆ เช่น DPWS, XMPP, COAP, MQTT เป็นต้น โดยที่ส่วนใหญ่จะเน้นไปทางด้านโปรโตคอล MQTT เป็นหลัก ทั้งการก่อตั้ง วัตถุประสงค์ รายละเอียดต่างๆ รวมไปถึงความปลอดภัยของโปรโตคอล MQTT มีการทดลอง security options ต่างๆ มาเปรียบเทียบกัน และการประเมิณค่า Average rount trip time (RTT) ระหว่างขนาดของ payload ที่ต่างกัน การใช้ RAM และ ROM บนทรัพยากรของเครื่องที่จำกัด การใช้พลังงานด้วยกลไลการเข้ารหัสที่ต่างกัน และการประยุกต์ใช้โปรโตคอล MQTT กับกังหันลม เพื่อส่งค่าเซนเซอร์ต่างๆ ไปทำการ monitor ตรวจสอบสถานะ สภาพแวดล้อม หรือเหตุการณ์ต่างๆ

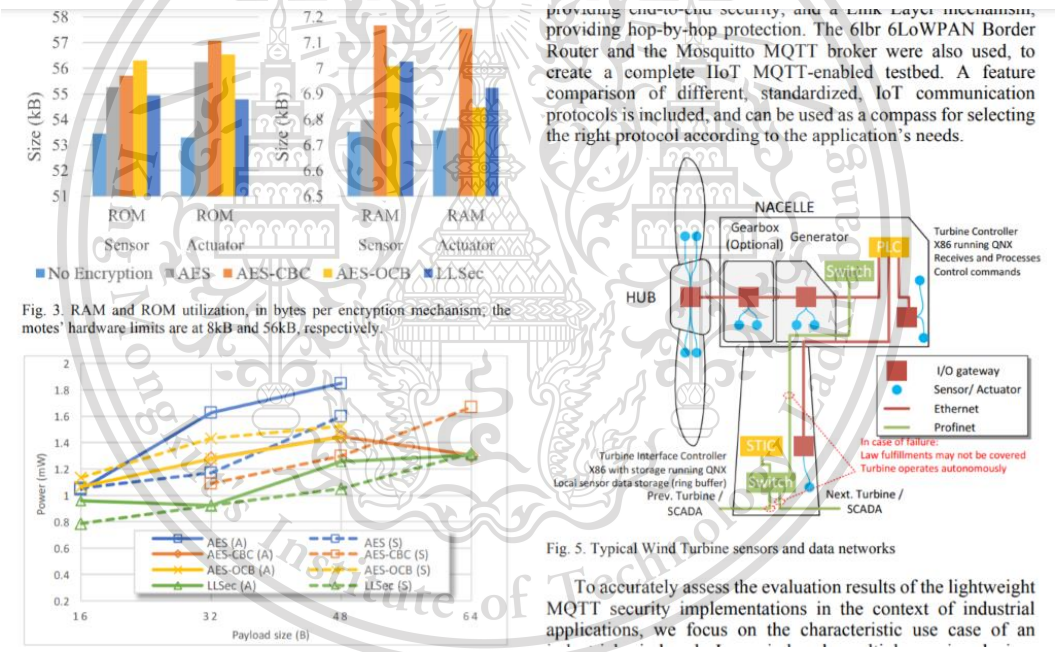


Fig. 3. RAM and ROM utilization, in bytes per encryption mechanism; the motes' hardware limits are at 8kB and 56kB, respectively.

Fig. 5. Typical Wind Turbine sensors and data networks

To accurately assess the evaluation results of the lightweight MQTT security implementations in the context of industrial applications, we focus on the characteristic use case of an

รูปที่ 2.5. กราฟแสดงการเปรียบเทียบการใช้ทรัพยากรและพลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 3
การออกแบบและการพัฒนา

3.1 ภาพรวมการทำงานในระบบ

3.1.1 uRTU

1) สามารถรับอินพุทในรูปแบบ แอนาลอก และดิจิตอลได้

2) สามารถสื่อสารกับ Simulator ผ่าน Modbus RTU โดยทำหน้าที่เป็น Master

3) สามารถส่งข้อมูลขึ้นไปยังServer โดยใช้ MQTT ได้

3.1.2 Simulator

1) สามารถจำลองเอาท์พุทสัญญาณ PWM และสัญญาณแอนาลอก 4-20 mA ได้

2) สามารถสื่อสารกับ uRTU ผ่าน Modbus RTU โดยทำหน้าที่เป็น Slave ได้

3.1.3 TIG Stack

1) สามารถรวบรวมข้อมูลจาก MQTT ได้

2) สามารถบันทึกข้อมูลลงบน Database ได้

3) สามารถดึงข้อมูลจาก Database มาแสดงใน Dashboard ได้

3.2 การทำงานโดยรวมของระบบ Monitoring เมื่อมีการเปิดใช้งาน uRTU จะมีขั้นตอนการทำงานและส่งข้อมูล

1) ADC Task จะทำการอ่านค่า 4-20 mA และคิดเป็นร้อยละ

2) Modbus Task จะกำหนด IP ให้กับ Simulator แล้วอ่านข้อมูลจาก Slave

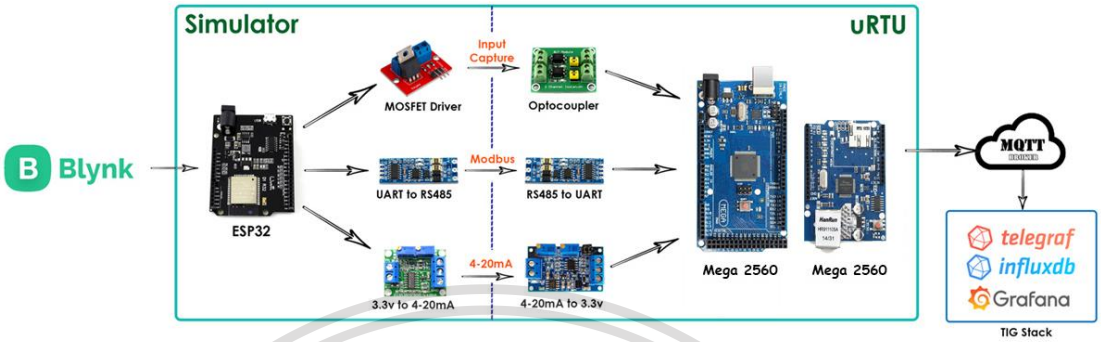
3) เมื่อมีสัญญาณ ISR Input Capture Task จะทำการเก็บค่า ความถี่ต่อเวลาหนึ่ง ของ จากสัญญาณ PWM ที่มาจาก Simulator

4) ทุกๆ เวลาหนึ่ง MQTT Task จะรวบรวมข้อมูล จากแต่ละ Task เพื่อส่งไปยัง Server Agent (Telegraf) รวบรวมข้อมูลและบันทึกลง Database (InfluxDB)

5) Grafana ทำการ Query ข้อมูลจาก Database มาแสดงบน Dashboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.3 โครงสร้างของระบบ



รูป 3.1 ไดอะแกรมของระบบ

3.3.1 uRTU ทำงานบนบอร์ด Atmel Atmega2650 เป็นอุปกรณ์รวบรวม Input และส่งค่าไปยังServer โดยการทำงานภายในจะประกอบด้วย

3.3.1.1 FreeRTOS เป็นตัวกลางในการควบคุมและจัดลำดับการทำงานของส่วนต่างๆ ใน uRTU

3.3.1.2 Input Capture Task ใช้สำหรับ รับค่า Digital (PWM) และคำนวณข้อมูลออกเป็นความถี่

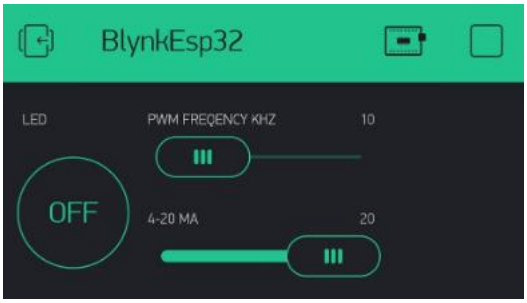
3.3.1.3 ADC Task ใช้สำหรับ รับค่าแอนาลอก 4-20 mA และ คำนวณข้อมูลออกเป็น 0-100%

3.3.1.4 Modbus Task ใช้สำหรับเชื่อมต่อ Modbus RTU กับ Simulator

3.3.1.5 MQTT Task เพื่อรวบรวมข้อมูล จากส่วนอื่นๆ แล้วส่งขึ้น Server ผ่าน MQTT

3.3.2 Simulator เป็นตัวจำลองอินพุทให้ uRTU ในรูปแบบ แอนาลอก (4-20 mA) และดิจิตอล (PWM) และสื่อสารกันผ่านทาง Modbus RTU โดย ควบคุมผ่าน Mobile Application Blynk

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



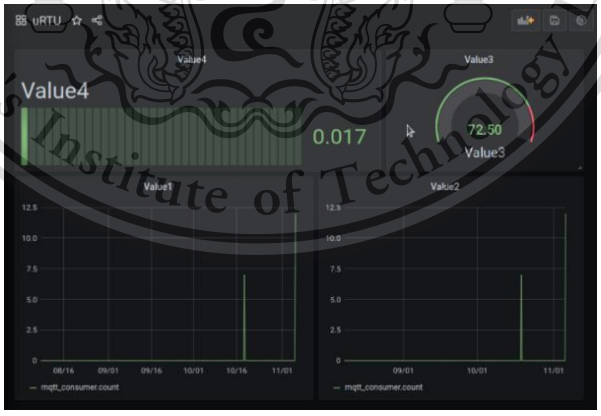
รูป 3.2 Application Blynk

3.3.3 TIG Stack

T ย่อมาจาก Telegraf ทำหน้าที่คอยเก็บรวบรวบข้อมูลจากแหล่งที่เราต้องการ (agent) และทำการส่งไปเก็บที่จุดหมาย (endpoint) ที่ต้องการเช่น MangoDB, MySQL, InfluxDB, JSON เป็นต้น

I ย่อมาจาก InfluxDB เป็น Time Series Database ที่จะคอยเก็บบันทึกข้อมูลที่ส่งมาจาก Telegraf และด้วยความที่ InfluxDB เป็น Time Series Database ทำให้สามารถจัดการกับข้อมูลที่เป็นวันเวลาได้อย่างรวดเร็วกว่า Database ทั่วไป และยังสามารถทำ Retention Policy กำหนดอายุของข้อมูลที่จัดเก็บได้

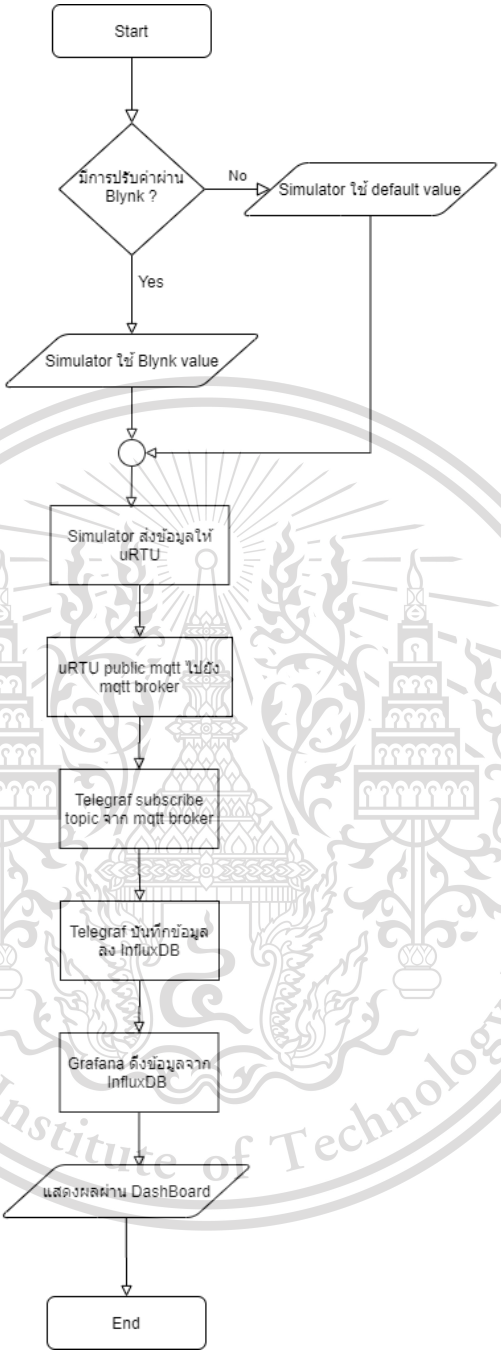
G ย่อมาจาก Grafana เป็นเครื่องมือที่ใช้ทำ Data visualization เป็น open source dashboard ที่สามารถทำงานร่วมกับ Datasource ต่างๆได้อย่างหลากหลาย มีความสวยงามและใช้งานง่าย



รูป 3.3 Grafana Dashboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

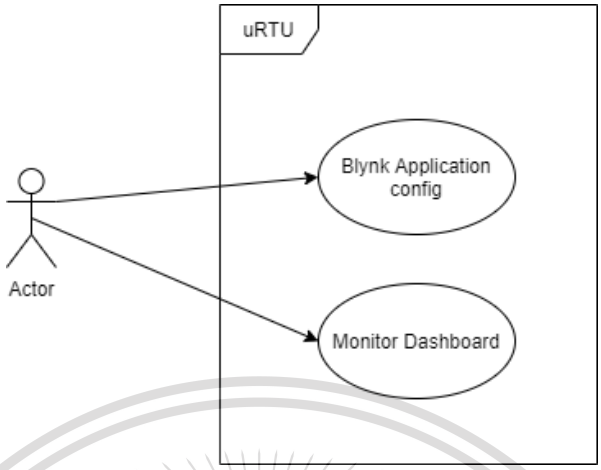
3.4 Flowchart การทำงานของระบบ



รูปที่ 3.4 Flowchart การทำงานของระบบ

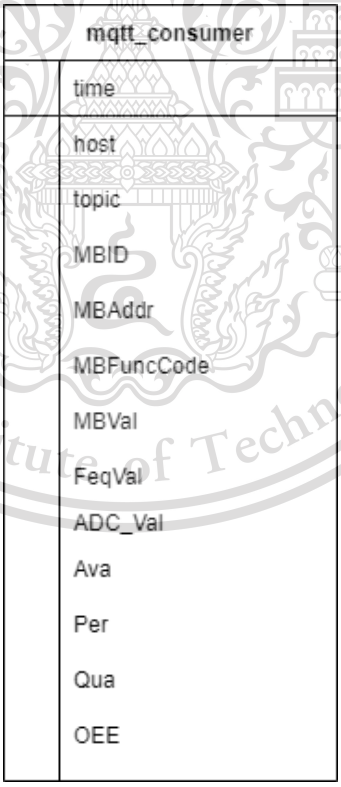
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.5 Use case Diagram



รูปที่ 3.5 Use Case Diagram

3.6 ER Diagram



รูปที่ 3.6 ER Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 4

การทดลอง

บทนี้เป็นการทดลองในส่วนของ TIG Stack และ simulator โดยดำเนินการทดลองดังนี้

1) การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบ INT และ Float

2) การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูแบบ JSON

3) การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จากเครื่องคอมพิวเตอร์ Client ขึ้นไปบน Server

4) การทดลองปรับค่าความถี่ของ PWM ผ่านแอพพลิเคชั่น Blynk

5) การทดลองควบคุม I/O บน STM32 ด้วย Web server และ LwIP

6) การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT ด้วย LwIP จาก STM32 ขึ้นไปบน Server

7) การทดลองการทำงานแบบ Multi Task บน FreeRTOS

4.1 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบ INT และ Float

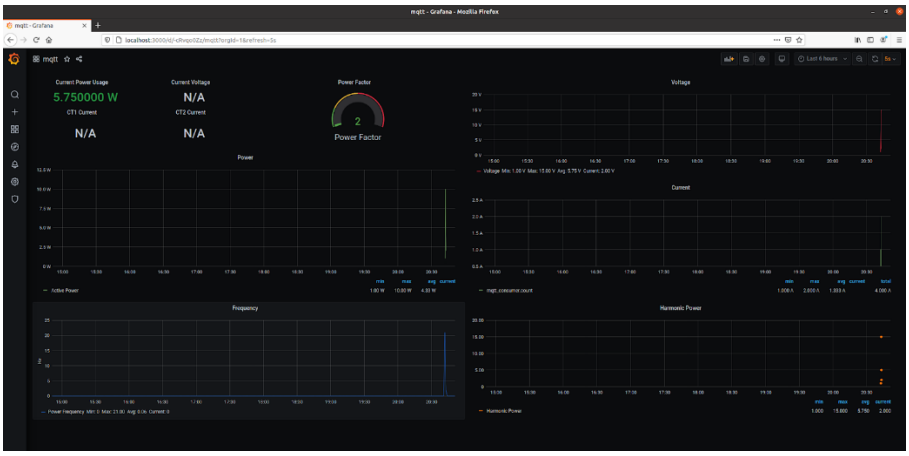
4.1.1 วัตถุประสงค์

เพื่อทดสอบการทำงานของ TIG Stack ว่าสามารถรับข้อมูลรูปแบบ INT และ Float จาก MQTT มาบันทึกลง Database และแสดงผลบน Dashboard ได้

4.1.2 วิธีการทดลอง

ใช้คำสั่ง mosquitto_pub บน Terminal ไปยัง localhost เพื่อ public mqtt และคำสั่ง mosquitto_sub เพื่อแสดงค่าที่เข้ามายังพอร์ตของ mqtt และตั้งค่า telegraf ให้รับค่าตามรูปแบบข้อมูลที่ต้องการคือ INT และ Float จากนั้นตรวจสอบ Database และ Dashboard ว่ามีค่าที่ public มาหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 4.1 Grafana Dashboard

4.1.3 ผลการทดลอง

ข้อมูลที่ public มาแสดงใน Database ถูกต้องครบถ้วนและมีดีเลย์ไม่มาก

```
name: mqtt_consumer
time                host    topic value
----                ----    ----- -----
1598868256857299248 vincent PM25  2
1598868257234733496 vincent PM25  2
1598868257634608504 vincent PM25  2
1598868615013592365 vincent PM25  2
1598868688716426294 vincent PM25  2
1598868785148124458 vincent PM25  7
1598868822914533091 vincent PM25  3
1598868850713989371 vincent PM25  15
1599130174775282649 vincent PM25  2
1599130193309863264 vincent PM25  2
1599130605287881142 vincent PM25  7
1599130609679632542 vincent PM25  5
1599130633173781161 vincent PM25  18
1599131110361839554 vincent PM25  18
1599131764165530025 vincent PM25  12
```

รูปที่ 4.2 ผลทดลองการส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบ INT และ Float

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

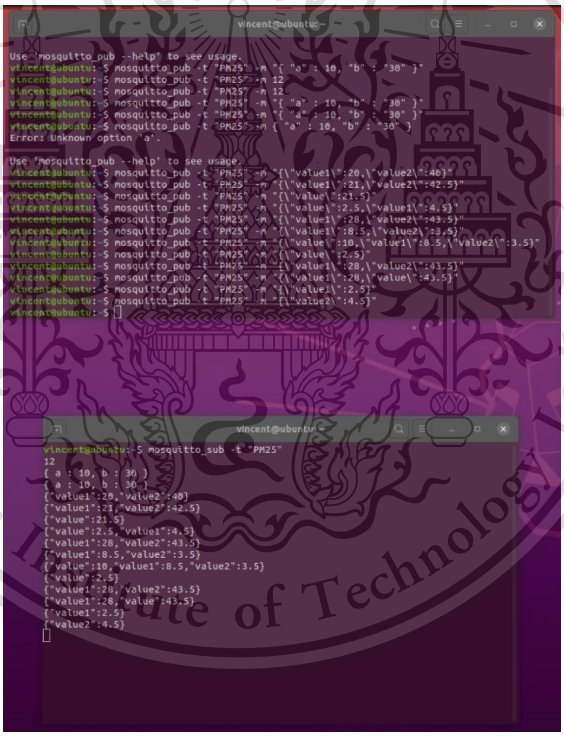
4.2 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบ JSON

4.2.1 วัตถุประสงค์

เพื่อทดสอบการทำงานของ TIG Stack ว่าสามารถรับข้อมูลรูปแบบ JSON จาก MQTT มาบันทึกลง Database และแสดงผลบน Dashboard ได้

4.2.2 วิธีการทดลอง

ใช้คำสั่ง mosquitto_pub บน Terminal ไปยัง localhost เพื่อ public mqtt และคำสั่ง mosquitto_sub เพื่อแสดงค่าที่เข้ามายังพอร์ตของ mqtt และตั้งค่า telegraf ให้รับค่าตามรูปแบบข้อมูลที่ต้องการคือ JSON จากนั้นตรวจสอบ Database และ Dashboard ว่ามีค่าที่ public มาหรือไม่

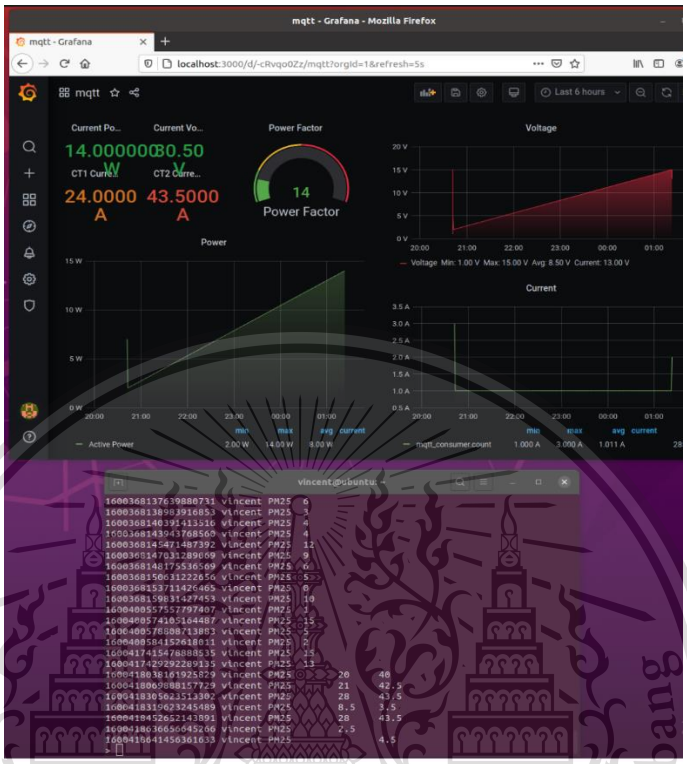


รูปที่ 4.3 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบ JSON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.2.3 ผลการทดลอง

ข้อมูลที่ public มาแสดงใน Database ถูกต้องครบถ้วนและมีดีเลย์ไม่มาก



รูปที่ 4.4 ผลทดลองการส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบ JSON

4.3 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จากเครื่องคอมพิวเตอร์ Client ขึ้นไปบน Server

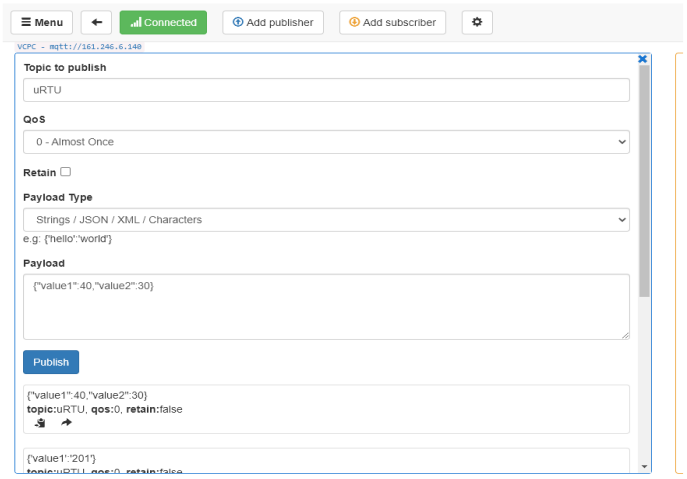
4.3.1 วัตถุประสงค์

เพื่อทดสอบการทำงานของ TIG Stack ว่าสามารถรับข้อมูลรูปแบบ JSON จาก MQTT บนเครื่องคอมพิวเตอร์ Client มาบันทึกลง Database และแสดงผลบน Dashboard

4.3.2 วิธีการทดลอง

ใช้โปรแกรม MQTTBox เพื่อ public mqtt จาก Client ไปยัง Server และใช้คำสั่ง mosquitto_sub บน Server เพื่อแสดงค่าที่เข้ามายังพอร์ตของ mqtt และตั้งค่า telegraf ให้รับค่าตามรูปแบบข้อมูลที่ต้องการคือ JSON จากนั้นตรวจสอบ Database และ Dashboard ว่ามีค่าที่ public มาหรือไม่

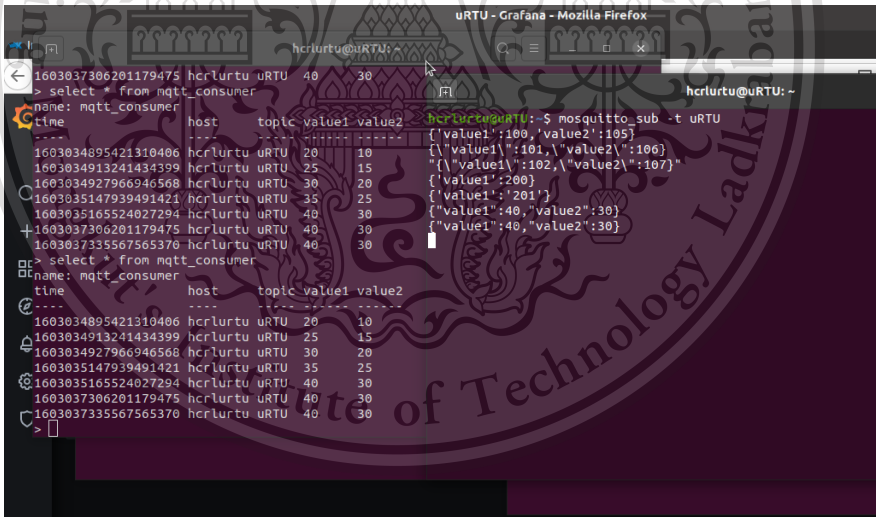
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 4.5 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จากเครื่องคอมพิวเตอร์ Client ขึ้นไปบน Server

4.3.3 ผลการทดลอง

ข้อมูลที่ public มาแสดงใน Database ถูกต้องครบถ้วนและมีดีเลย์ไม่มาก



รูปที่ 4.6 ผลทดลองการส่งข้อมูลผ่าน โปรโตคอล MQTT จากเครื่องคอมพิวเตอร์ Client ขึ้นไปบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4 การทดลองปรับค่าความถี่ของ PWM ผ่านแอพพลิเคชั่น Blynk

4.4.1 วัตถุประสงค์

เพื่อทดสอบว่าสามารถปรับค่าความถี่ผ่านแอพพลิเคชั่น Blynk ได้

4.4.2 วิธีการทดลอง

ปรับค่าความถี่เป็นหน่วย kHz ผ่าน Slider บน Blynk และต่อสายจาก Output เข้า Input ของตัวบอร์ดเอง เขียนโปรแกรมนับค่าความถี่และให้แสดงผลทาง Serial

4.4.3 ผลการทดลอง

สามารถปรับค่าความถี่ได้แต่ค่าที่วัดมาคลาดเคลื่อนเล็กน้อยต้องวัดด้วย oscilloscope และค่าความถี่บางช่วงที่ผลลัพธ์ออกมายังไม่ตรงกับค่าที่ปรับ

รูปที่ 4.7 ผลทดลองการปรับค่าความถี่ของ PWM ผ่านแอพพลิเคชั่น Blynk

4.5 การทดลองควบคุม I/O บน STM32 ด้วย Web server และ LwIP

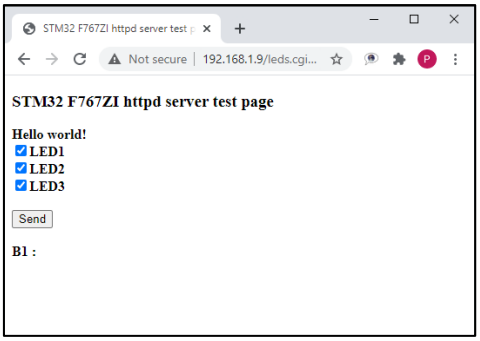
4.5.1 วัตถุประสงค์

เพื่อทดลองการทำงานของ LwIP ว่าสามารถใช้งานรวมกับ โปรโตคอล HTTP บน STM32-F767Zi ได้

4.5.2 วิธีการทดลอง

เปิดเว็บบราวเซอร์ แล้วเข้า IP (192.168.1.9) ตาม Port Ethernet ที่ STM32 เชื่อมต่ออยู่ เลือก Checkbox LED ที่ต้องการ แล้วคลิก Send

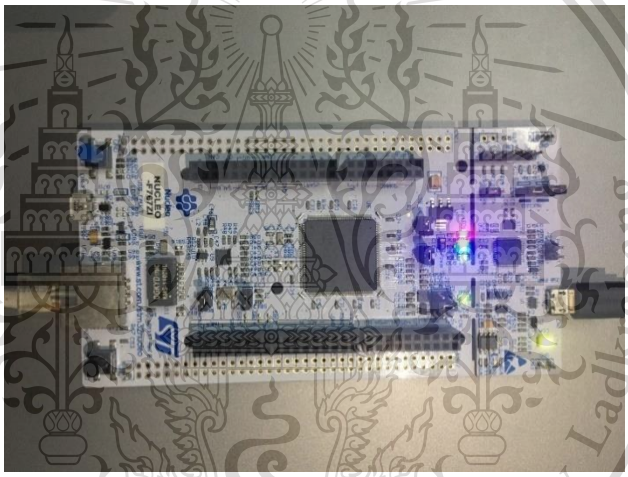
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 4.8 การทดลองควบคุม I/O บน STM32 ด้วย Web server และ LwIP

4.5.3 ผลการทดลอง

สามารถควบคุมให้ LED แสดงผลตาม Check Box ที่เลือก



รูปที่ 4.9 ผลทดลองการควบคุม I/O บน STM32 ด้วย Web server และ LwIP

4.6 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT ด้วย LwIP จาก STM32 ขึ้นไปบน Server

4.6.1 วัตถุประสงค์

เพื่อทดสอบการทำงานของ STM32 ว่าสามารถส่งข้อมูลผ่านโปรโตคอลMQTT ด้วย LwIP จาก STM32-F767Zi ขึ้นไปบน Server ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.6.2 วิธีการทดลอง

Install MQTT Packet ในหน้าต่างPacket และ Include Library Path MQTT Client ในโปรแกรม Keil uVision5

Include Library MQTT Client ในไฟล์ main.c เรียกใช้งาน Function ต่างๆ และทำการ Debug

4.6.3 ผลการทดลอง

ไม่สามารถ Connect MQTT Broker ได้

4.6.4 แก้ไขปัญหา

เปลี่ยน Controller จาก STM32F767Zi เป็น Arduino Mega 2560

4.7 การทดลองการทำงานแบบ Multi Task บน FreeRTOS

4.7.1 วัตถุประสงค์

เพื่อ ทดสอบการทำงาน แบบ MultiTask เพื่อจัดการการเข้าใช้งานทรัพยากรภายในuRTU

4.7.2 วิธีการทดลอง

1) Install FreeRTOS ใน STM32CubeMX และ Implement ใน Keil uVision5

2) ทดสอบการสร้าง Task กำหนด Priority

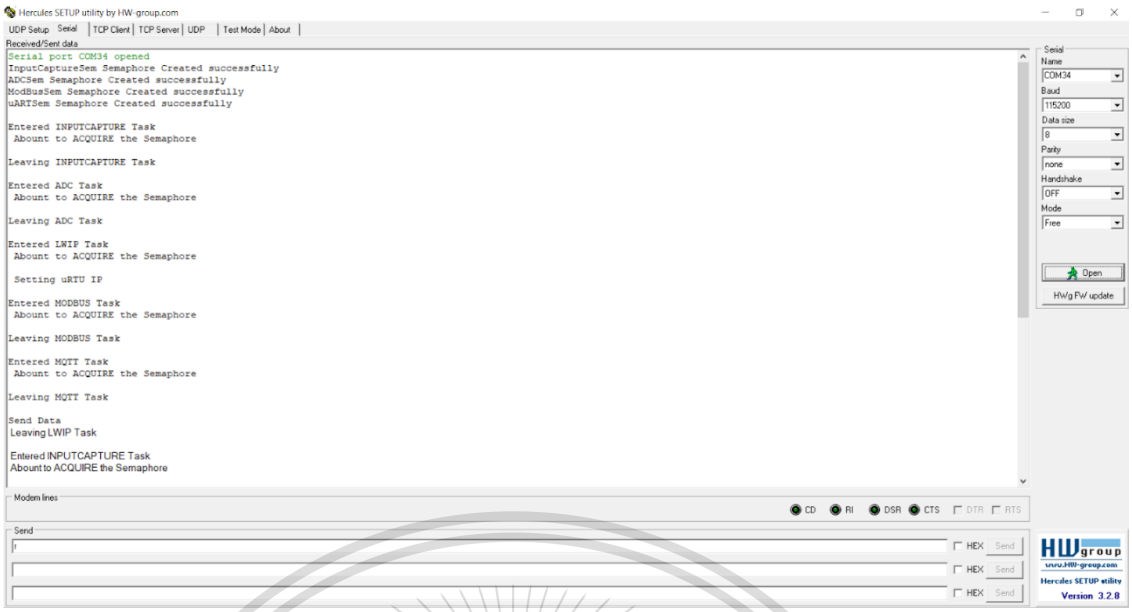
3) ทดสอบการทำงานของ Binary Semaphore ,Counting Semaphore ,Mutex และ Suspend -Resume

4) เลือกใช้ Counting Semaphore และ Suspend - Resume ในการจัดการ Task ต่างๆ ให้มีลำดับการทำงานแบบ Polling ในการอ่านข้อมูลที่ได้รับจาก Simulator โดย Debug ลำดับการทำงานของ Task ต่างๆ ผ่านทาง uART ผ่าน โปรแกรม Hercules และใช้ semaphore เข้าควบคุม uART เพื่อไม่ให้แต่ละ Task แย่งการใช้งาน uART กัน

4.7.3 ผลการทดลอง

สามารถ ควบคุมการทำงานของ Task ต่างๆโดยเรียงลำดับ Priority ได้ถูกต้อง และไม่มีการซ้อนทับกันของข้อความที่ส่งมา Debug เพราะควบคุมสิทธิ์การเข้าถึง uART ไว้แล้วโดยที่แต่ละ Task ยังทำงานตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 4.10 ผลการทดลองโปรแกรม uRTU แบบ Multi Tasks โดยใช้ FreeRTOS

4.8 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จาก Mega 2560 ขึ้นไปบน Server

4.8.1 วัตถุประสงค์

เพื่อทดสอบการทำงานของ Mega 2560 ว่าสามารถส่งข้อมูลผ่านโปรโตคอล MQTT ขึ้นไปบน Server ได้

4.8.2 วิธีการทดลอง

Install MQTT Library ในโฟลเดอร์ libraries ของ Arduino IDE

เขียนโปรแกรมอ่านสัญญาณ Analog และใช้คำสั่ง Publish ส่งค่าที่อ่านได้ไปยัง MQTT Server

รูปที่ 4.11 การทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จาก Mega 2560 ขึ้นไปบน Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.8.3 ผลการทดลอง

สามารถส่งข้อมูลผ่าน โปรโตคอล MQTT จาก Mega 2560 ขึ้นไปบน Server ได้

```
> select * from mqtt_consumer
name: mqtt_consumer
time                ADC_Val Ava FeqVal MBAddr MBFuncCode MBID MBVal OEE Per Qua host     topic
----                ------- --- ------ ------ ---------- ---- ----- --- --- --- ----     -----
1617083275739347083 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083276433253975 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083280530232200 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083281116275793 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083281889975266 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083282483580844 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083283154036829 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083283908637022 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083284575525960 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083285596703563 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083286586661479 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083287466094807 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083288566058441 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083289415645043 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083290125545094 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083290805190989 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083291404888061 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083292025182935 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083292402898919 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
1617083293939401699 66      93  55     1      16         12   10    66  79  90  hcrlurtu uRTU
>
```

รูปที่ 4.12 ผลการทดลองส่งข้อมูลผ่าน โปรโตคอล MQTT จาก Mega 2560 ขึ้นไปบน Server

4.9 การทดลองอ่านค่า Modbus RTU จาก Simulator ด้วย Mega 2560

4.9.1 วัตถุประสงค์

เพื่อทดสอบการทำงานของ Mega 2560 ว่าสามารถอ่านค่า Modbus RTU Slave จาก Simulator

4.9.2 วิธีการทดลอง

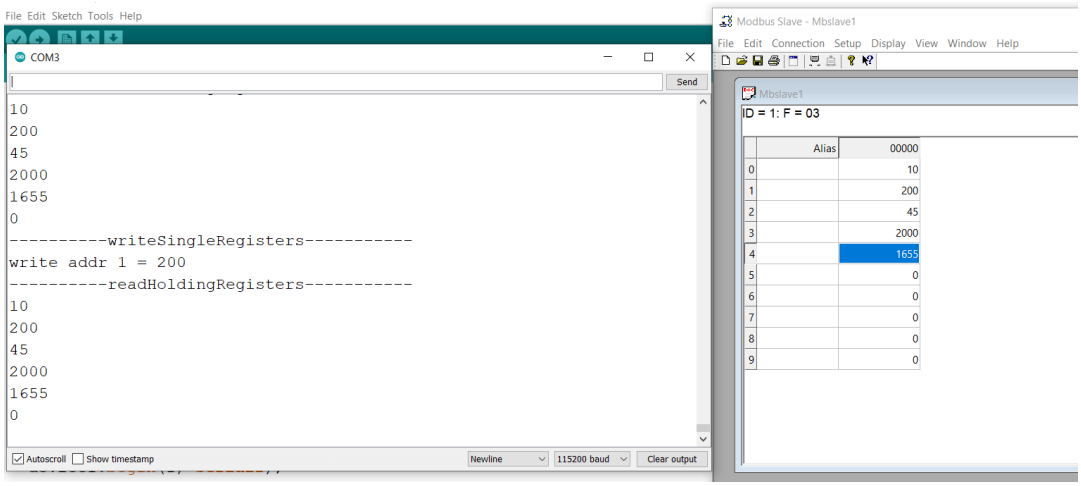
Install ModbusMaster Library ในโฟลเดอร์ libraries ของ Arduino IDE

เขียนโปรแกรมอ่านค่า Modbus RTU Slave จาก Simulator โดยกำหนดให้ Mega 2560 ทำหน้าที่เป็น Modbus RTU Master และใช้ MAX485 Module

4.9.3 ผลการทดลอง

สามารถสามารถอ่านค่า Modbus RTU Slaveจาก Simulator ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 4.13 ผลการทดลองอ่านค่า Modbus RTU จาก Simulator ด้วย Mega 2560

4.10 การทดลองคำนวณมาตรฐาน OEE โดยใช้ข้อมูลจาก InfluxDB

4.10.1 วัตถุประสงค์

เพื่อแสดงให้เห็นถึงข้อมูลที่ถูกส่งโดย uRTU สามารถนำไปคำนวนเพื่อแสดงให้เห็นถึงประสิทธิผลโดยรวมของเครื่องจักร เพื่อใช้ในการปรับปรุง พัฒนาของเครื่องจักรอุตสาหกรรม

4.10.2 วิธีการทดลอง

สร้าง Web Page ที่สามารถเรียกข้อมูล คำนวณ และเก็บข้อมูล ด้วย InfluxDB

เมื่อเรียกข้อมูลจาก InfluxDB แล้วจะข้อมูลบ้างส่วนที่จำเป็นต่อการคำนวณมาตรฐาน OEE เช่น Planned Production Time , Ideal Cycle Time , Good Pieces , Total Pieces เป็นต้นซึ่งผู้ใช้สามรถกำหนดข้อมูลผ่าน Web Page ได้

4.10.3 ผลการทดลอง

สามารถสามารถคำนวณมาตรฐาน OEE โดยใช้ข้อมูลจาก InfluxDB ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

รูปที่ 4.14 ผลการแสดงข้อมูลจาก InfluxDB

King Mongkut's Institute of Technology Ladkrabang

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 5

สรุปผลการดำเนินงานและแนวทางการพัฒนา

5.1 ผลการดำเนินงาน

1) สามารถส่งข้อมูลผ่าน โปรโตคอล MQTT บน server ผ่าน localhost ด้วยรูปแบบข้อมูลแบบ INT ,Float และ JSON

2) สามารถส่งข้อมูลผ่าน โปรโตคอล MQTT จากเครื่องคอมพิวเตอร์ Client ขึ้นไปบน Server

3) สามารถปรับค่าความถี่ของ PWM ผ่านแอพพลิเคชั่น Blynk

4) สามารถทำงานแบบ Multi Task บน FreeRTOS

5) ไม่สามารถ Connect MQTT Broker ในการทดลอง เพื่อส่งข้อมูลผ่านโปรโตคอล MQTT จาก STM32 ขึ้นไปบน Server

6) เปลี่ยน Controller จาก STM32 มาใช้ Arduino Mega 2560

7) ส่งข้อมูลผ่าน โปรโตคอล MQTT จาก Mega 2560 ขึ้นไปบน Server

8) สามรถอ่านค่า Modbus RTU จาก Simulator ด้วย Mega 2560 ได้

9) สามารถคำนวณประสิทธิผลโดยรวมของเครื่องจักร ตามมาตรฐาน OEE โดยใช้ข้อมูลจาก InfluxDB

5.2 ปัญหาอุปสรรค และแนวทางการแก้ไข

5.2.1 ปัญหาอุปสรรค

1) มีปัญหา ไม่สามารถ Connect MQTT Broker ในการทดลอง เพื่อส่งข้อมูลผ่านโปรโตคอล MQTT จาก STM32 ขึ้นไปบน Server

5.2.2 แนวทางการแก้ไข

1) เปลี่ยน Controller จาก STM32 มาใช้ Arduino Mega 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

5.3 แนวทางการพัฒนา

1) เพิ่มจำนวนสัญญาณจาก Simulator เพื่อให้ได้ข้อมูลที่ครอบครุมพอที่สามารถนำไปคำนวนมาตรฐาน OEE ได้โดยไม่ต้องกำหนดบ้างข้อมูลในการทดลอง

2) เพิ่มข้อมูลใน Grafana Dashboard และออกแบบให้สวยงามยิ่งขึ้น

3) เปลี่ยนหน่วยประมวลผลเป็น Controller ที่มีประสิทธิภาพในการทำงานสูง เพื่อให้ครอบคลุมการประมวลข้อมูลที่เพิ่มมากขึ้น

4) นำไปประยุกต์ใช้งานในระบบ IIOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บรรณานุกรม

- FreeRTOS Real-time operating system for microcontrollers เข้าถึงครั้งสุดท้ายเมื่อ 13/11/2563 https://www.freertos.org/
- Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol เข้าถึงครั้งสุดท้ายเมื่อ 10/10/2563 https://ieeexplore.ieee.org/document/8024687
- FreeRTOS เข้าถึงครั้งสุดท้ายเมื่อ 13/11/2563 https://controllerstech.com/freertos/
- Design and Development of Modbus/MQTT Gateway for Industrial IoT Cloud Applications Using Raspberry Pi เข้าถึงครั้งสุดท้ายเมื่อ 16/10/2563 https://ieeexplore.ieee.org/document/8997492
- รู้จักกับ TIG Stack เข้าถึงครั้งสุดท้ายเมื่อ 08/09/2563 https://medium.com/ayuth/%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81%E0%B8%81%E0%B8%B1%E0%B8%9A-tick-%E0%B9%81%E0%B8%A5%E0%B8%B0-tig-stack-%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B8%88%E0%B8%B0%E0%B8%A1%E0%B8%B2%E0%B8%8A%E0%B9%88%E0%B8%A7%E0%B8%A2-monitoring-infrastructure-d223aeeaa361
- Install and Setup TIG Stack on Ubuntu 20.04 เข้าถึงครั้งสุดท้ายเมื่อ 15/11/2563 https://kifarunix.com/install-and-setup-tig-stack-on-ubuntu-20-04/
- ESP32 maximum PWM frequency เข้าถึงครั้งสุดท้ายเมื่อ 20/02/2564 https://forum.micropython.org/viewtopic.php?t=3717

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.