ระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม

BATTERY MANAGEMENT SYSTEM ON CLOUD PLATFORM

คุณานนท์ ใจตรง

ธีรภัทร ชูชัยมงคล

สมัชญา ใจอินทร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BATTERY MANAGEMENT SYSTEM ON CLOUD PLATFORM

KUNANON JAITRONG
TEERAPHAT CHUCHAIMONGKOL
SAMATCHAYA JAI-IN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERRING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2019

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม BATTERY MANAGEMENT SYSTEM ON CLOUD PLATFORM			
นักศึกษาผู้จัดทำ	นายคุณานนท์	ใจตรง	รหัสนักศึกษา	59010179
	นายธีรภัทร	ชูชัยมงคล	รหัสนักศึกษา	59010665
	นางสาวสมัชญา	ใจอินทร์	รหัสนักศึกษา	59011355
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร.อาจินต์		น่วมสำราญ	
	รองศาสตราจารย์ ดร.วิทยา		ทิพย์สุวรรณพร	
ปีการศึกษา	2562			

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้แสดงการวัดทางไฟฟ้าเพื่อนำค่าที่ได้ไปวิเคราะห์แบตเตอรี่ ออกแบบและจัดทำ BMS on Cloud Platform โดยมี STM32 เป็น Microcontroller ใช้ในการบริหารจัดการการชาร์จ ดิสชาร์จและสื่อสารข้อมูล โดยใช้ INA219 ในการวัดค่าทางไฟฟ้าของแบตเตอรี่ ใช้ Thermistor ในการวัดอุณหภูมิ และใช้ LTC4162 เป็นวงจรควบคุมค่าคงที่ของแรงดัน, ค่าคงที่ของกระแสและอุณหภูมิของการชาร์จ นำค่าที่ได้จากการวัดไปวิเคราะห์หาประสิทธิภาพการทำงานของแบตเตอรี่เพื่อประเมินสุขภาพแบตเตอรี่โดยใช้ Raspberry Pi ที่มีระบบปฏิบัติการพื้นฐานเป็น Raspbian แสดงข้อมูลการวัดทางไฟฟ้าในรูปแบบ HMI ที่ออกแบบและสร้าง User interface ด้วย Node-Red และเก็บข้อมูลบน Cloud service เนื่องจากระบบคลาวด์สามารถปรับเปลี่ยนขนาดของพื้นที่จัดเก็บข้อมูลได้ตามความต้องการของผู้ใช้งาน อีกทั้งยังสามารถเรียกดูข้อมูลของแบตเตอรี่ผ่าน Application ได้แบบ Real time ในกรณีที่มีการเชื่อมต่ออินเตอร์เน็ต แต่หากอุปกรณ์ขาดการเชื่อมต่ออินเตอร์เน็ตยังคงสามารถดูข้อมูลผ่านทาง Raspberry Pi ได้นอกจากนี้ระบบจัดการแบตเตอรี่ยังสามารถแจ้งเตือนผู้ใช้งานได้ในกรณีที่ประสิทธิภาพการทำงานของแบตเตอรี่ลดลงหรือมีความเสี่ยงที่อาจก่อให้เกิดความเสียหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	BATTERY MANAGEMENT SYSTEM ON CLOUD PLATFORM	
Authors	Mr. Kunanon	Jaitrong
	Mr. Teeraphat	Chuchaimongkol
	Ms. Samatchaya	Jai-in
Thesis Advisors	Assoc. Prof. Dr. Arjin	Numsomran
	Assoc. Prof. Dr. Vittaya	Tipsuwanporn
Year	2019	

ABSTRACT

This thesis shows electrical measurement to analyze the value, design and create BMS on Cloud Platform. The STM32 is a Microcontroller use to manage charge-discharge and communicate data, using INA219 to measure the electrical values of the battery, Thermistor to measure Temperature and using LTC- 4162 to control constant voltage, constant current and temperature in charging circuit. To take the measured value to analyze the performance of the battery to assess the state of health (SoH) using a Raspberry Pi with a basic operating system as Raspbian displays electrical measurement data in HMI format, designed and built in the User interface with Node-Red. And store data on the Cloud service because the Cloud system can adjust the size of the storage space according to user needs. You can also view the battery information through the application in Real time in the event of an internet connection, but if the device loses the internet connection, data can still be viewed via the Raspberry Pi. In addition, the battery management system can alert users in the event that battery performance decreases or there is a risk of causing damage.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดีโดยคำแนะนำคำปรึกษาต่าง ๆ ตลอดจนอุปกรณ์ต่าง ๆ ที่ใช้ในการทำโครงงานจาก รศ.ดร.อาจินต์ น่วมสำราญ และ รศ.ดร.วิทยา ทิพย์สุวรรณพร ซึ่งเป็นอาจารย์ที่ปรึกษาในการทำปริญญานิพนธ์และเป็นอาจารย์ผู้ให้คำแนะนำวิธีคิด, ทัศนคติตลอดจนเป็นกำลังใจและผู้สนับสนุนในการทำโครงงานให้แก่ผู้จัดทำด้วยดีเสมอมา ทางผู้จัดทำรู้สึกซาบซึ้งและขอบพระคุณเป็นอย่างสูง

ขอขอบคุณคณาจารย์และบุคลากรภาควิชาวิศวกรรมการวัดคุม สาขาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และให้คำแนะนำต่าง ๆ ด้วยดีเสมอมา

ขอบคุณสมาชิกในกลุ่มที่ช่วยกันทำงานด้วยความมานะบากบั่น ขยัน และอดทนฟันฝ่าอุปสรรคมาด้วยกัน ขอขอบคุณเพื่อน ๆ พี่ ๆ และน้อง ๆ ในสาขาวิชาทุกคนที่เป็นกำลังใจอย่างดีเสมอ

สุดท้ายนี้ต้องขอขอบพระคุณ บิดา มารดาและครอบครัวที่เป็นกำลังใจในการทำโครงงานนี้ คุณค่าและประโยชน์อันพึงมาจากปริญญานิพนธ์ฉบับนี้ผู้จัดทำขอมอบแด่ผู้มีพระคุณ

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....IX
สารบัญรูป.....X

บทที่ 1 บทนำ.....1
1.1 ความสำคัญของปริญญานิพนธ์.....1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1
1.3 ขอบเขตของปริญญานิพนธ์.....2
1.4 ขั้นตอนของการศึกษา.....2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....3
2.1 แบตเตอรี่ลิเธียมไอออน.....3
2.1.1 คุณสมบัติแบตเตอรี่ลิเธียมไอออน.....3
2.1.2 ส่วนประกอบแบตเตอรี่ลิเธียมไอออน.....4
2.1.3 วัสดุสำหรับทำขั้วแคโทด.....5
2.1.4 วัสดุสำหรับทำขั้วแอโนด.....5
2.1.5 หลักการทำงานของแบตเตอรี่ลิเธียมไอออน.....6
2.2 คุณสมบัติของแบตเตอรี่ที่ใช้ในการศึกษา.....7
2.2.1 State of Charge.....7
2.2.2 Open Circuit Voltage.....7
2.2.3 State of Health.....9
2.3 The Internet of Things (IoT).....9
2.4 Sensor node.....9
2.5 Edge computing.....10
2.6 Cloud computing.....11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.7 Database....12
2.8 Human Machine Interface.... 13
2.9 Node-Red....15
2.10 Message Queuing Telemetry Transport (MQTT)....16
2.11 Ubidots....16
2.11.1 หลักการทำงานของ Ubidots....17
2.11.2 อนุกรมเวลา (Time series)....19
2.11.3 MQTT....19
2.11.4 Analytics : ข้อมูลพื้นฐานเกี่ยวกับตัวแปรสังเคราะห์....20
2.12 การส่งข้อมูลแบบไร้สายผ่านบลูทูธ....20
2.13 Arduino IDE....21
2.14 งานวิจัยที่เกี่ยวข้อง....21

บทที่ 3 วิธีการดำเนินงาน....23
3.1 ภาพรวมขององค์ประกอบหลักของระบบ....23
3.2 การออกแบบและการดำเนินงาน....24
3.2.1 การเลือกประเภทของแบตเตอรี่ที่ใช้ศึกษา....24
3.2.2 การออกแบบระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม....24
3.2.2.1 การออกแบบอุปกรณ์สำหรับระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม....24
3.2.2.2 ออกแบบการใช้งาน Raspberry Pi สำหรับระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม....25
3.2.2.3 ออกแบบคุณลักษณะของ Application บน Cloud Platform....25
3.3 องค์ประกอบฮาร์ดแวร์....29
3.3.1 อุปกรณ์วัดแรงดันไฟฟ้า และ อุณหภูมิ....29
3.3.2 STM32 Nucleo F091RC....29
3.3.3 Demo circuit 2038a....30
3.3.4 X-nucleo-idb05a1....31
3.3.5 I2C OLED....32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3.6 Relay....32
3.3.7 Thermistor....33
3.3.8 Raspberry Pi....33
3.3.9 Siemens Simatic IOT 2040....34
3.3.10 Capacitive Touch Screen....35
3.4 ซอฟต์แวร์ที่ใช้พัฒนาระบบ....35
3.5 การประมวลผลข้อมูลด้วยไมโครคอนโทรลเลอร์....36
3.6 การใช้งาน Node–Red บน Raspberry Pi....42
3.6.1 ขั้นตอนการติดตั้งระบบปฏิบัติการบน Raspberry Pi....42
3.6.2 การใช้งานโปรแกรม Node-Red....43
3.7 การสื่อสารข้อมูลระหว่าง STM32 และ Raspberry Pi ด้วย Bluetooth....45
3.8 การออกแบบ User interface จาก Node-Red บน Raspberry Pi....47
3.9 การแจ้งเตือนผ่าน Line notify ด้วย Node-Red....52
3.10 การสร้างฐานข้อมูล SQLite บน Raspberry Pi....53
3.11 การสื่อสารข้อมูลระว่าง Raspberry Pi และ Cloud computing ด้วย MQTT โปรโตคอล....55
3.12 Siemens Simatic IOT 2040 to Cloud....56
3.13 การเลือกผู้ให้บริการ Cloud Platform....59
3.14 การสร้าง Application โดยใช้ Ubidots....59
3.15 การสร้างอุปกรณ์, ตัวแปรและกลุ่มของอุปกรณ์....62
3.15.1 การสร้างอุปกรณ์สำหรับเก็บตัวแปรต่าง ๆ ที่รับค่ามาจากส่วน Edge computing....62
3.15.2 การสร้างตัวแปร (Variable)....65
3.15.3 การสร้างกลุ่มของอุปกรณ์....71
3.16 การรับค่าจาก Edge computing....72
3.17 บันทึกข้อมูลและส่งออกข้อมูลจาก Ubidots....73
3.17.1 การส่งออกข้อมูลของตัวแปร....73
3.17.2 การส่งออกข้อมูลของอุปกรณ์....75
3.17.3 การส่งออกข้อมูลของข้อมูลองค์กร....76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.18 การสร้างหน้าแสดงผล (Dashboard) ใน Ubidots....78
3.19 การสร้างการแจ้งเตือนอัตโนมัติใน Ubidots....83
3.20 การสร้างหน้ารายงานอัตโนมัติ (Report) ใน Ubidots....86
3.21 การเพิ่มองค์กร, บทบาท, ผู้ใช้งานใน Ubidots....88
3.22 การปรับแต่งแอปพลิเคชัน....91

บทที่ 4 ผลการทดลอง....96
4.1 ทดสอบการ Charge – Discharge ของระบบ....96
4.2 ทดสอบการสื่อสารข้อมูลระหว่าง STM32 และ....97
Raspberry Pi ด้วย Bluetooth ทดสอบระบบด้วยค่า Simulation
4.3 ทดสอบการแสดงผล Dashboard ของ Raspberry Pi....99
4.4 ทดสอบการแจ้งเตือนผ่าน Line notify ด้วย Node-Red....100
4.5 ทดสอบการแสดงผลข้อมูลที่จัดเก็บใน SQLite....101
4.6 ทดสอบการส่งค่า Simulation จาก Edge computing ไปยัง.... 101
Cloud computing
4.7 ทดสอบความถูกต้องของสมการคำนวณจากค่า Simulation....105
4.8 ทดสอบการแสดงผลของหน้า Dashboard ตามเงื่อนไขที่ทำการตั้งค่าไว้....106
4.9 ทดสอบการแจ้งเตือนอัตโนมัติ....111
4.10 ทดสอบการส่งรายงานอัตโนมัติประจำสัปดาห์....114

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ....116
5.1 สรุปผลการทดลอง....116
5.2 ปัญหาและอุปสรรค....116
5.3 ข้อเสนอแนะสำหรับการวิจัยในอนาคต....117

บรรณานุกรม....118
ภาคผนวก....123
ภาคผนวก ก. NCR18650B Datasheet....124
ภาคผนวก ข. SDM120CT Energy meter datasheet....126

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

ภาคผนวก ค. Ubidots: Synthetic variable operations.....129

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงรายละเอียดรายการของแต่ละจุด.....18
3.1 แสดงรายการอุปกรณ์และตัวแปรที่ต้องการสร้างขึ้น.....26
3.2 แสดงข้อมูลที่ต้องการแสดงในหน้า Dashboard.....27
3.3 แสดง Device name และ Device label ที่กำหนดขึ้น.....63
3.4. แสดงรายละเอียดของตัวแปรที่กำหนดขึ้น.....65
3.5 แสดง Widget และรายการที่ต้องการแสดง.....80
4.1 แสดงข้อมูลที่ใช้ทดสอบ Dashboard ของ Cell battery.....100
4.2 แสดงการคำนวนค่า State of Charge ของอุปกรณ์ Battery string 01.....107
4.3 แสดงการคำนวนค่า State of Charge ของอุปกรณ์ Battery cell 0101.....107
Battery cell 0102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 เปรียบเทียบความสารถในการเก็บพลังงานไฟฟ้า ขนาด และน้ำหนักของแบตเตอรี่ชนิดต่าง ๆ 4
2.2 แสดงส่วนประกอบของลิเธียมไอออนแบตเตอรี่ 4
2.3 แสดงการประจุไฟฟ้าและคายประจุไฟฟ้าของแบตเตอรี่ลิเธียมไอออน 6
2.4 แสดงกระบวนการคายประจุของเซลล์ลิเธียมไอออน 8
2.5 แรงดันไฟฟ้าวงจรเปิด (OCV) เทียบกับสถานะประจุ (SoC) ที่มีการใช้งานก่อนหน้านี้ที่ 25 ℃ 8
2.6 แสดง Sensor node 10
2.7 แสดง Edge computing ที่อยู่ใกล้กับแหล่งข้อมูล หรือ ระบบ 10
2.8 แสดงโครงสร้าง Relational database 12
2.9 แสดงโครงข่าย Distributed database 13
2.10 แสดงการเชื่อมต่อสื่อสารข้อมูลของ PLC แบบ Touch screen ในการเข้าถึงแอปพลิเคชันต่าง ๆ ของผู้ใช้งานจากระยะไกล 14
2.11 ส่วนติดต่อระหว่างผู้ใช้กับกับระบบ 14
2.12 หน้าต่างการเรียกใช้งานโปรแกรม Node-Red 15
2.13 หน้าต่างเว็บไซต์สำหรับดาว์นโหลดและติดตั้งโปรแกรม Node-Red 15
2.14 แสดงความสัมพันธ์ระหว่าง Publisher, Broker, Topic และ Subscriber 16
2.15 แสดงการทำงานของ Ubidots 17
2.16 แสดงลำดับชั้นข้อมูลของ Ubidots 17
2.17 แสดงอนุกรมเวลาของ Ubidots 19
2.18 แสดงตัวอย่างการคำนวณหน่วยองศาเซลเซียสเป็นหน่วยองศาฟาเรนไฮต์ 20
2.19 หลักการส่งสัญญาณ Bluetooth 21
3.1 ภาพรวมของการทำงานของระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม 23
3.2 ความสามารถในการเก็บพลังงานของแบตเตอรี่ประเภทต่าง ๆ 24
3.3 ภาพรวมการทำงานของอุปกรณ์ของระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์มที่ทำการออกแบบ 28
3.4 แผนภาพกิจกรรมของการทำงานของระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม 28
3.5 INA219 29
3.6 STM32 Nucleo F091RC 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.7 Port การใช้งาน STM32 Nucleo F091RC....30
3.8 Demo circuit 2038a....31
3.9 X-nucleo-idb05a1....31
3.10 I2C OLED ขนาด 128*64....32
3.11 Relay 5VDC....32
3.12 Thermistor 100K....33
3.13 Raspberry Pi 4 Model B....34
3.14 Siemens Simatic IOT 2040....34
3.15 หน้าจอสัมผัส....35
3.16 แสดงหน้าเว็บในการโหลดโปรแกรม....37
3.17 Dowload Arduino IDE....37
3.18 หน้าต่าง Arduino IDE....38
3.19 Select Preference Arduino IDE....38
3.20 Preference Arduino IDE....39
3.21 Board Manager....39
3.22 Install Board STM32....40
3.23 แสดงการเขียนโปรแกรมลงใน Arduino IDE....40
3.24 เลือก Board ที่ต้องการ Upload ลง....41
3.25 ระบบทำการ Charge....41
3.26 ระบบทำการ Discharge....42
3.27 เว็บเพจสำหรับการติดตั้งโปรแกรม Node-Red บน Raspberry Pi....43
3.28 แสดง Terminal ของ Raspberry Pi....43
3.29 แสดงการทำงานของโปรแกรม Node-Red บน Server....44
3.30 แสดงการ Log in ก่อนใช้งานโปรแกรม Node-Red....44
3.31 แสดงหน้าหลักโปรแกรม Node-Red....45
3.32 แสดงผังการไหลของข้อมูลสำหรับสื่อสารข้อมูลผ่าน Bluetooth....45
3.33 แสดงการเชื่อมต่อระหว่าง Raspberry Pi และ STM32 ผ่าน Bluetooth ที่มีชื่อว่า BTLE....46
3.34 Mac Address และ UUID ที่ใช้ในการเข้าถึงข้อมูลของ Bluetooth....46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.35 ผังการไหลของข้อมูลสำหรับแสดงค่าทางไฟฟ้าของแหล่งจ่ายไฟที่วัดผ่าน SDM120 Energy meter....47
3.36 Dashboard แสดงค่าทางไฟฟ้าที่วัดได้จาก SDM120 Energy meter....48
3.37 ระบุชนิดของแบตเตอรี่....48
3.38 Dashboard แสดง datasheet ของแบตเตอรี่....49
3.39 ผังการไหลของข้อมูลสำหรับแสดงค่าทางไฟฟ้าของ String battery....49
3.40 ฟังก์ชันสำหรับคำนวณ SoC ของแบตเตอรี่....50
3.41 ฟังก์ชันสำหรับคำนวณ SoH ของแบตเตอรี่....50
3.42 Dashboard แสดงค่าทางไฟฟ้าของ String battery....51
3.43 ผังการไหลของข้อมูลสำหรับแสดงค่าทางไฟฟ้าของ Cell battery....51
3.44 Dashboard แสดงค่าทางไฟฟ้าของ Cell battery....52
3.45 ผังการไหลของข้อมูลเพื่อทำการแจ้งเตือนผ่าน Line notify....52
3.46 ฟังก์ชันเพื่อทำการแจ้งเตือนผ่าน Line notify....53
3.47 ตั้งค่า http request....53
3.48 ผังการไหลของข้อมูลเพื่อทำ SQLite บน Raspberry Pi....54
3.49 ฟังก์ชันเพื่อรับและแสดงค่าผ่าน Node-Red....54
3.50 Dashboard แสดงข้อมูลภายในฐานข้อมูล SQLite....55
3.51 ไฟล์ SQLite ที่จัดเก็บอยู่ใน Raspberry Pi....55
3.52 แสดงผังการไหลของข้อมูลโดยใช้ MQTT โปรโตคอล....56
3.53 แสดงการกำหนด Token เพื่อเข้าถึงการสื่อสารข้อมูลผ่าน MQTT โปรโตคอล....56
3.54 แสดงผังการไหลข้อมูลการวัดค่าพลังงานไฟฟ้าของเพาเวอร์ซัพพลาย....57
3.55 กำหนด Address เพื่อรับค่าที่ต้องการวัดจาก SDM120CT Energy meter....58
3.56 ใส่ Token ของ Ubidots เพื่อทำการสื่อสารข้อมูลผ่าน MQTT โปรโตคอล....58
3.57 แสดงหน้าเวปเพจของ Ubidots....59
3.58 แสดงหน้าลงทะเบียนของ Ubidots....60
3.59 แสดงหน้าลงทะเบียนของ Ubidots....60
3.60 แสดงหน้า Dashboard ของแอปพลิเคชัน Ubidots ที่สร้างขึ้น....60
3.61 แสดงหน้าลงทะเบียนของแอปพลิเคชัน Ubidots บน Smart phone....61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.62 แสดงหน้าหน้าข้อมูลผู้ใช้งานของแอปพลิเคชัน Ubidots บน Smart phone.....61
3.63 แสดงหน้าต่างในการสร้างอุปกรณ์.....62
3.64 ผังงานแสดงการรับค่าจากอุปกรณ์ของ Uidots.....63
3.65 แสดงรายละเอียดของอุปกรณ์ Supply meter ที่ทำการสร้างขึ้น.....64
3.66 แสดงรายละเอียดของอุปกรณ์ทั้งหมดที่ทำการสร้างขึ้น.....64
3.67 แสดงรายละเอียดของอุปกรณ์ชื่อ Supply meter.....65
2.68 แสดงหน้าอุปกรณ์ Battery string 01.....66
2.69 แสดงการสร้างตัวแปรในอุปกรณ์ Battery string 01.....66
2.70 แสดงตัวแปร Voltage ที่ทำการสร้างขึ้นในอุปกรณ์ Battery string 01.....68
3.71 แสดงรายละเอียดของตัวแปร Temperature battery.....68
3.72 แสดงตัวแปรที่ทำการสร้างขึ้นในอุปกรณ์ Battery string 01.....68
3.73 แสดงการสร้างตัวแปรแบบ Synthetic ในอุปกรณ์ Battery string 01.....69
3.74 แสดงการสร้างสมการคำนวณสำหรับตัวแปร State of Charge.....69
3.75 แสดงตัวแปร State of Charge ที่ทำการสร้างขึ้นในอุปกรณ์ Battery string 01.....70
3.76 แสดงรายละเอียดของตัวแปร State of Charge.....70
3.77 แสดงตัวแปรทั้งหมดที่ทำการสร้างขึ้นในอุปกรณ์ Battery string 01.....71
3.78 แสดงหน้า Groups ของ Ubidots.....71
3.79 แสดงการเลือกอุปกรณ์ที่ต้องการจัดกลุ่ม.....72
3.80 แสดง Device group ที่สร้างขึ้น.....72
3.81 แสดง API Key และ Tokens ของบัญชี Ubidots.....73
3.82 แสดงหน้าตัวแปร Power ในอุปกรณ์ Supply meter.....73
3.83 แสดงหน้าต่างกำหนดข้อมูลของตัวแปรที่ต้องการส่งออกข้อมูล.....74
3.84 แสดงแถบการแจ้งเตือนของแอปพลิเคชัน.....74
3.85 แสดงอีเมลสำหรับดาน์โหลดข้อมูลของตัวแปร Power ที่ทำการส่งออก.....74
3.86 แสดงไฟล์ CSV ของข้อมูลของตัวแปร Power ที่ทำการส่งออก.....75
3.87 แสดงหน้าอุปกรณ์ Supply meter.....75
3.88 แสดงหน้าต่างกำหนดข้อมูลของอุปกรณ์ที่ต้องการส่งออกข้อมูล.....76
3.89 แสดงไฟล์ CSV ของข้อมูลของอุปกรณ์ Supply meter ที่ทำการส่งออก.....76
3.90 แสดงหน้าองค์กร My BMS team.....77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.91 แสดงการส่งออกข้อมูลขององค์กร My BMS team....77
3.92 แสดงหน้าต่างสำหรับกำหนดการส่งออกข้อมูลขององค์กร My BMS team....77
3.93 แสดงอีเมลสำหรับดาน์โหลดข้อมูลขององค์กร My BMS team ที่ทำการส่งออก....78
3.94 แสดง Zip ไฟล์ขององค์กร My BMS team....78
3.95 แสดงหน้า Demo dashboard....78
3.96 แสดงชนิด Widget ที่ให้บริการ....79
3.97 แสดงการเชื่อม Thermometer widget กับ ตัวแปร Temperature battery....81
3.98 แสดงการตั้งค่า Thermometer widget....81
3.99 แสดง Thermometer widget ขณะยังไม่มีการส่งข้อมูล....82
3.100 แสดง BMS Dashboard ขณะยังไม่มีการส่งข้อมูลบน Personal computer....82
3.101 แสดง BMS Dashboard ขณะยังไม่มีการส่งข้อมูลบน Smart phone....83
3.102 แสดงหน้าต่างการสร้างแจ้งเตือน....84
3.103 แสดงตัวแปรและเงื่อนไขในการแจ้งเตือนอุณหภูมิของแบตเตอรี่ที่สถานะต่าง ๆ....84
3.104 แสดงรายการรูปแบบการแจ้งเตือนที่ให้บริการใน Ubidots....85
3.105 แสดงรายละเอียดการแจ้งเตือนแบบ e-mail....85
3.106 แสดงชื่อและช่วงเวลาที่กำหนดให้การแจ้งเตือนทำงาน....85
3.107 แสดงรายการแจ้งเตือนอัตโนมัติทั้งหมดที่ถูกสร้างขึ้น....86
3.108 แสดงหน้าต่างการสร้างรายงาน....86
3.109 แสดงตัวอย่างการตั้งค่าชื่อและรูปแบบของรายงาน....87
3.110 แสดงตัวอย่างการตั้งค่าแผนภูมิและตารางที่ต้องการแสดงในรายงาน....87
3.111 แสดงตัวอย่างการตั้งค่าระยะเวลาในการส่ง, ข้อความและผู้รับ....88
3.112 แสดงรายการรายงานอัตโนมัติที่สร้างขึ้น....88
3.113 แสดงองค์กร My BMS team ที่สร้างขึ้น....89
3.114 แสดงหน้าต่างการเพิ่มผู้ใช้งาน....90
3.115 แสดงอีเมลเพื่อเชิญผู้ใช้งานใหม่เข้าสู่ระบบ....90
3.116 แสดงหน้า Dashboard ของผู้ใช้ในบทบาท Explorers....90
3.117 แสดงหน้าต่างเมนู App....91
3.118 แสดงหน้า Domains....91
3.119 แสดงหน้าองค์กร My BMS team....92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.120 แสดงหน้า Token ขององค์กร My BMS team....92
3.121 แสดงการเพิ่ม Token ขององค์กร My BMS team ใน Organization domain....92
3.122 แสดงรายละเอียดการปรับแต่งหน้า Interface ของผู้ใช้แอปพลิเคชัน....93
3.123 แสดงหน้า ลงทะเบียนของ bmsproject.iot.ubidots.com ก่อนการปรับแต่ง....94
3.124 แสดงหน้า ลงทะเบียนของ bmsproject.iot.ubidots.com หลังการปรับแต่ง....94
3.125 แสดงหน้า Dashboard ของแอปพลิเคชัน BMS on Cloud ก่อนการปรับแต่ง....94
3.126 แสดงหน้า Dashboard ของแอปพลิเคชัน BMS on Cloud หลังการปรับแต่ง....95
4.1 แสดงการ Charge....96
4.2 แสดงการ Discharge....96
4.3 แสดงสถานะของ Bluetooth....97
4.4 GATT Characteristics อ่านข้อมูลที่อยู่ใน Bluetooth....97
4.5 ฟังก์ชันเพื่อแปลงข้อมูลค่าความชื้นของ STM32....98
4.6 แสดงค่าความชื้นของ STM32....98
4.7 โหลดจากไฟฟ้าภายในบ้านพักอาศัยที่วัดผ่าน Siemens Simatic IOT 2040 แสดงผลบน Raspberry Pi....99
4.8 Dashboard ของ Power supply....99
4.9 Dashboard ของ String battery....100
4.10 Dashboard ของ Cell battery....101
4.11 แสดงการแจ้งเตือนผ่าน Line notify....101
4.12 แสดงข้อมูลที่ถูกจัดเก็บใน SQLite....102
4.13 แสดงการส่งค่าจาก Edge computing ไปยัง อุปกรณ์ Supply meter....103
2.14 หน้าอุปกรณ์ Supply meter ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing....103
4.15 แสดงการส่งค่าจาก Edge computing ไปยังอุปกรณ์ Battery string 01....104
4.16 หน้าอุปกรณ์ Battery string 01 ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing....104
4.17 แสดงการส่งค่าจาก Edge computing ไปยังอุปกรณ์ Battery cell 0101....105
4.18 หน้าอุปกรณ์ Battery cell 0101 ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing....105
4.19 แสดงการส่งค่าจาก Edge computing ไปยังอุปกรณ์ Battery cell 0102....106
4.20 หน้าอุปกรณ์ Battery cell 0102 ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing....106
4.21 การแสดงผลในหน้า Dashboard....108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.22 Color logic ที่กำหนดไว้ของ Thermometer widget....108

4.23 Temperature battery widget แสดงอุญหภูมิ 20 degC และ....109
Temperature environment widget แสดงอุญหภูมิ 100 degC

4.24 Temperature battery widget แสดงอุญหภูมิ 45 degC และ....109
Temperature environment widget แสดงอุญหภูมิ 100 degC

4.25 Temperature battery widget แสดงอุญหภูมิ 60 degC และ....109
Temperature environment widget แสดงอุญหภูมิ 45 degC

4.26 Temperature battery widget แสดงอุญหภูมิ 100 degC และ....110
Temperature environment widget แสดงอุญหภูมิ 20 degC

4.27 Temperature battery widget แสดงอุญหภูมิ 80 degC และ....110
Temperature environment widget แสดงอุญหภูมิ 35 degC

4.28 Color logic ที่กำหนดไว้ของ Battery widget....110

4.29 State of Health : Battery string 01 widget แสดง 100%....111
State of Health : Battery cell 0101 widget แสดง 100% และ
State of Health : Battery cell 0101 widget แสดง 100%

4.30 State of Health : Battery string 01 widget แสดง 70%,....111
State of Health : Battery cell 0101 widget แสดง 70% และ
State of Health : Battery cell 0101 widget แสดง 70%

4.31 State of Health : Battery string 01 widget แสดง 50%,....112
State of Health : Battery cell 0101 widget แสดง 50% และ
State of Health : Battery cell 0101 widget แสดง 50%

4.32 State of Health : Battery string 01 widget แสดง 0%,....112
State of Health : Battery cell 0101 widget แสดง 0% และ
State of Health : Battery cell 0101 widget แสดง 0%

4.33 อีเมลการแจ้งเตือน State of Health ของอุปกรณ์ Battery string 01....113

4.34 อีเมลการแจ้งเตือน State of Health ของอุปกรณ์ Battery cell 0101....113

4.35 อีเมลการแจ้งเตือน State of Health ของอุปกรณ์ Battery cell 0102....113

4.36 อีเมลการแจ้งเตือนอุณหภูมิของแบตเตอรี่ขณะชาร์จ > 45 degC....114

4.37 อีเมลการแจ้งเตือนอุณหภูมิของแบตเตอรี่ขณะชาร์จ > 60 degC....114

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.38 อีเมลการแจ้งเตือนอุปกรณ์ขาดการเชื่อมต่อนานกว่า 15 นาที.....114
4.39 อีเมลการแจ้งเตือนการส่งรายงานอัตโนมัติประจำสัปดาห์.....115

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ปัจจุบันแบตเตอรี่ลิเธียมไอออนกลายเป็นอุปกรณ์อิเล็กทรอนิกส์ที่มีบทบาทต่อการดำเนินชีวิตของประชากรโลก เนื่องจากเป็นแหล่งสำรองพลังงานไฟฟ้าที่เข้าถึงได้ง่ายและมีอยู่ในอุปกรณ์ใช้ไฟฟ้าเกือบทุกชนิด ไม่ว่าจะเป็นโทรศัพท์มือถือ แล็ปท็อป กล้องถ่ายรูป เพาเวอร์แบงค์ รวมไปถึงงานทางด้านการโทรคมนาคม ดาวเทียม และอุตสาหกรรมยานยนต์ไฟฟ้าหรือ Electrical vehicle (EV ที่กำลังเป็นกระแสสังคมในขณะนี้ นอกจากนี้ยังเป็นอุปกรณ์อิเล็กทรอนิกส์ที่สามารถนำไปต่อยอดพัฒนาเป็นนวัตกรรมใหม่ได้ จากที่ได้กล่าวไปในข้างต้นนี้จะเห็นได้ว่ามีการใช้งานของลิเธียมไอออนแบตเตอรี่อย่างแพร่หลายจึงจำเป็นต้องมีระบบจัดการแบตเตอรี่ที่ดีไว้ดูแลและจัดการการใช้งาน เพราะอายุการใช้งานของแบตเตอรี่จะลดลงไปตามเวลาและพฤติกรรมการใช้ ดังนั้นเพื่อให้สามารถใช้งานแบตเตอรี่ได้อย่างมีประสิทธิภาพสูงสุด ปลอดภัยที่สุดด้วยการลดความเสี่ยงในการเกิดอุบัติเหตุที่ไม่คาดคิดจากการใช้งาน โดยระบบ BMS on Cloud Platform จะดูแลให้การประจุและคายประจุไฟฟ้าของแบตเตอรี่เป็นไปอย่างมีเสถียรภาพมากที่สุด อีกทั้งยังสามารถแจ้งเตือนเมื่อมีความผิดปกติเกิดขึ้นกับแบตเตอรี่หรือระบบ ในส่วนผู้ใช้งานสามารถเข้าถึงข้อมูลการใช้งานของแบตเตอรี่และแหล่งจ่ายพลังงานได้แบบ Real time ผ่านระบบคลาวด์

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาธรรมชาติของแบตเตอรี่
2. ศึกษาการสื่อสารข้อมูลระหว่างอุปกรณ์แบบไร้สาย
3. ศึกษาการทำงานของ Internet of Things
4. ออกแบบและสร้างระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม
5. สามารถประยุกต์ใช้ Microcontroller ร่วมกับอุปกรณ์ Sensor
6. ออกแบบ User interface (UI) และสร้างฐานข้อมูลบน Raspberry Pi ให้เป็น Human Machine Interface (HMI) ด้วยโปรแกรม Node-Red
7. ออกแบบและสร้าง Dashboard, การแจ้งเตือนความผิดปกติและรายงานประจำสัปดาห์ได้อย่างอัตโนมัติและการจัดเก็บข้อมูลด้วย Cloud service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญานิพนธ์

1. วัดค่าทางไฟฟ้าด้วย INA219 และใช้ Microcontroller (STM32) วิเคราะห์ผลจากการวัด
2. ใช้โปรแกรมคำสั่งเพื่อควบคุมอุปกรณ์อิเล็กทรอนิกส์
3. เกิดการสื่อสารข้อมูลระหว่างอุปกรณ์แต่ละส่วน
4. ออกแบบระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม
5. แสดงข้อมูลการใช้งานของแบตเตอรี่พร้อมทั้งแจ้งเตือนผู้ใช้งานบนคลาวด์แพลตฟอร์ม

1.4 ขั้นตอนของการศึกษา

ปริญญานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันดังนี้

บทที่ 1 บทนำ

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

บทที่ 3 วิธีการดำเนินงาน

บทที่ 4 ผลการทดลอง

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถประยุกต์ใช้เครื่องมือวัดและเขียนโปรแกรมควบคุมกระบวนการที่มีการสื่อสารกันระหว่างอุปกรณ์แบบไร้สาย
2. สามารถออกแบบวงจรจัดการการชาร์จแบตเตอรี่และสร้างระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม
3. สามารถวัดและแสดงผล ค่าทางไฟฟ้าของแบตเตอรี่ได้แบบ Real time
4. สามารถวิเคราะห์หาประสิทธิภาพการใช้งานของแบตเตอรี่และประเมินสุขภาพแบตเตอรี่
5. สามารถสร้างการแจ้งเตือนเมื่อมีความผิดปกติเกิดขึ้นกับแบตเตอรี่และระบบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 แบตเตอรี่ลิเธียมไอออน

แบตเตอรี่ลิเธียมไอออน (Lithium-Ion Battery) หรือตัวย่อคือ “Li-Ion” เป็นแบตเตอรี่ที่สามารถนำกลับมาใช้ใหม่หรือใช้ซ้ำได้ ในปัจจุบันมีการใช้งานกันอย่างแพร่หลายทั้งด้านการไฟฟ้าและสาธารณูปโภค เช่น สมาร์ทโฟน แล็ปท็อป แท็บเล็ต และงานด้านโทรคมนาคม ดาวเทียม ด้านการทหาร และการเก็บพลังงานไฟฟ้าจากแหล่งพลังงานหมุนเวียน [1] - [3]

2.1.1 คุณสมบัติแบตเตอรี่ลิเธียมไอออน

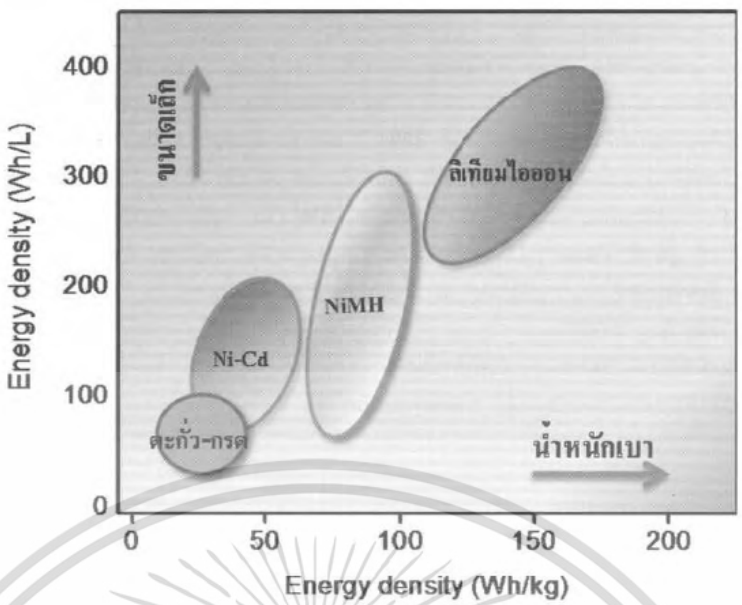
1) น้ำหนักเบา : ธาตุลิเธียมที่นำมาใช้เป็นโลหะอัลคาไลน์มีน้ำหนักเบาที่สุดในโลก ทำให้ตัวแบตเตอรี่มีน้ำหนักเบาตามไปด้วย ด้วยเหตุนี้รถยนต์ไฟฟ้าจึงใช้พลังงานในการขับเคลื่อนน้อยลงเพราะน้ำหนักที่น้อยลง ทำให้ประหยัดพลังงานได้มากยิ่งขึ้น [4]

2) อายุการใช้งานนาน : แบตเตอรี่ลิเธียมไอออนมีประจุไฟฟ้าที่สูงกว่าและเก็บประจุไฟฟ้าได้นานกว่าแบตเตอรี่ประเภทอื่น ๆ อีกทั้งยังมีประสิทธิภาพในการชาร์จที่สูงกว่าและใช้งานได้นาน

3) ให้พลังงานสูง, คงที่และชาร์จได้เร็ว : ในส่วนประกอบของธาตุลิเธียมมีเซลล์ไฟฟ้าเคมีที่สูงว่าเซลล์จากโลหะอื่น จึงทำให้แบตเตอรี่มีแรงดันไฟฟ้าที่สูงกว่าแบตเตอรี่ชนิดอื่น ๆ แบตเตอรี่ลิเธียมไอออน จึงกลายเป็นต้นกำเนิดของพลังงานที่มีความเสถียรที่สุด ถือเป็นหนึ่งในพลังงานทางเลือกที่จะเข้ามาทดแทนพลังงานเชื้อเพลิงดังรูปที่ 2.1

4) เป็นเซลล์แห้ง : แบตเตอรี่ลิเธียมไอออนไม่มีส่วนประกอบที่เป็นอันตรายต่อธรรมชาติ เช่น ของเหลว กรด หรือตะกั่ว จึงสามารถรับประกันเรื่องความปลอดภัยต่อมนุษย์และสิ่งแวดล้อม

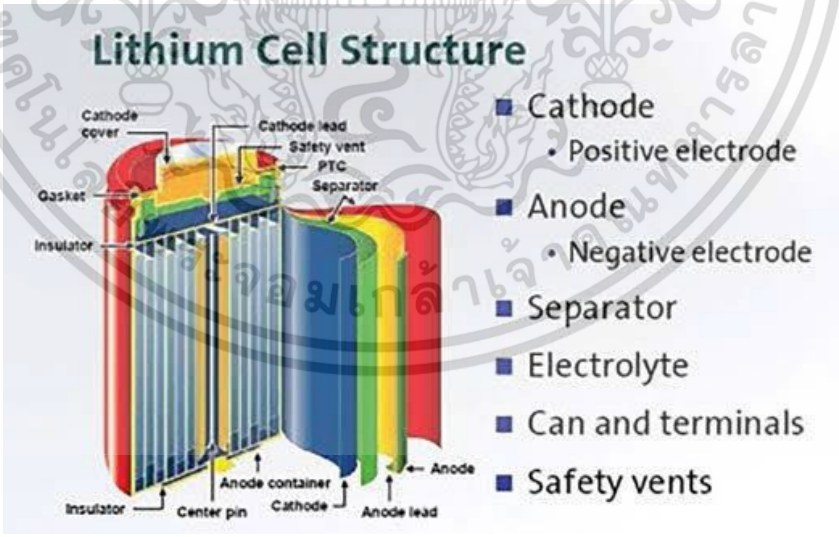
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 เปรียบเทียบความสามารถในการเก็บพลังงานไฟฟ้า ขนาด และน้ำหนักของแบตเตอรี่ชนิดต่าง ๆ

ที่มา : ดร.นงลักษณ์ มีทอง (2553 : 1) ความสามรถในการกักเก็บพลังงานไฟฟ้าของ แบตเตอรี่ชนิดต่าง ๆ

2.1.2 ส่วนประกอบแบตเตอรี่ลิเธียมไอออน



รูปที่ 2.2 แสดงส่วนประกอบของแบตเตอรี่ลิเธียมไอออน

ที่มา : Bureau (2019) from https://www.saurenergy.com/solar-energy-articles/can-india-beat-china-in-li-ion-battery-manufacturing-and-its-pricing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ขั้วไฟฟ้า ได้แก่ ขั้วแคโทดและขั้วแอโนด

2) แผ่นกั้นในแบตเตอรี่ (Seperator) ช่วยป้องกันกระแสลัดวงจรโดยทำให้ขั้วแคโทดไม่สัมผัสกับขั้วแอโนด

3) อิเล็กโทรไลต์ เป็นสารละลายที่มีเกลือของลิเธียมผสมอยู่ เป็นตัวนำที่ยอมให้ไอออนผ่านแต่ไม่ยอมให้อิเล็กตรอนไหลผ่าน ดังนั้นจึงเป็นตัวนำไอออนิกที่ดีแต่เป็นตัวนำอิเล็กทรอนิกที่ไม่ดี

4) ตัวรับกระแส (Current collector) เป็นโลหะตัวนำทำหน้าที่ให้อิเล็กตรอนไหลผ่านออกสู่วงจรภายนอก และเกิดการนำพลังงานไฟฟ้าไปใช้ประโยชน์ต่าง ๆ

ความสามารถในการกักเก็บพลังงานของแบตเตอรี่ชนิดลิเธียมไอออนขึ้นอยู่กับชนิดและคุณสมบัติของวัสดุที่นำมาทำลิเธียมไอออนแบตเตอรี่

2.1.3 วัสดุสำหรับทำขั้วแคโทด

วัสดุสำหรับทำขั้วแคโทดมี 3 ประเภทหลัก คือ วัสดุประเภทเลเยอร์ (Layered structure) เช่น $\mathbf{LiCoO_2}$, วัสดุประเภทสปิเนล (Spinel structure) เช่น $\mathbf{LiMn_2O_4}$ และวัสดุประเภทโอลิวีน (Olivine structure) เช่น $\mathbf{LiFePO_4}$ [5]

2.1.4 วัสดุสำหรับทำขั้วแอโนด

ส่วนใหญ่ใช้วัสดุประเภทคาร์บอน เช่น แกรไฟต์ เมื่อประจุไฟลิเธียมไอออนจะเข้าไปอัดตัวอยู่ในโครงสร้างของแกรไฟต์ในรูปของสารประกอบ $\mathbf{Li_xC_6}$ ซึ่งมีความปลอดภัยสูงกว่าการใช้โลหะลิเธียมทำขั้วไฟฟ้าโดยตรง วัสดุอื่น ๆ ก็สามารถนำไปใช้ทำขั้วแอโนดได้ เช่น $\mathbf{Li_xSi}$, $\mathbf{Li_xe}$ และวัสดุที่มีขนาดระดับนาโนเมตร เช่น ซิลิคอนนาโนไวร์ (Silicon nanowires) ได้ถูกค้นพบว่าเหมาะสำหรับทำขั้วแอโนด เนื่องจากมีความสามารถในการรับลิเธียมไอออนมากกว่าแกรไฟต์ 10 เท่า แต่วัสดุนี้มีข้อเสียคือการขยายตัวอย่างมากเมื่อถูกอัดลิเธียมไอออนเข้าไปในโครงสร้าง ซึ่งส่งผลให้วัสดุมีโอกาสเสื่อมสภาพและเสียหายได้ง่าย ทำให้อายุการใช้งานของแบตเตอรี่สั้นลง ส่วนวัสดุอีกชนิดหนึ่งที่ใช้ทำขั้วแอโนดได้คือ วัสดุ $\mathbf{Li_4Ti_5O_{12}}$ เป็นวัสดุที่มีคุณสมบัติพิเศษคือ ไม่มีการหดหรือขยายตัวเมื่อถูกอัดลิเธียมไอออนเข้าไปในโครงสร้าง ส่งผลให้อายุการใช้งานของแบตเตอรี่ที่ทำจากวัสดุนี้นานขึ้น แต่วัสดุชนิดนี้มีข้อเสียคือมีความสามารถในการอัดลิเธียมไอออนได้น้อยเมื่อเทียบกับสารประกอบโลหะของลิเธียม ดังนั้นจึงจำเป็นต้องเพิ่มปริมาณของวัสดุที่ทำขั้วแอโนด ซึ่งส่งผลให้ต้นทุนการผลิตเพิ่มสูงขึ้นและแบตเตอรี่มีขนาดและน้ำหนักที่มากขึ้นด้วย [6]

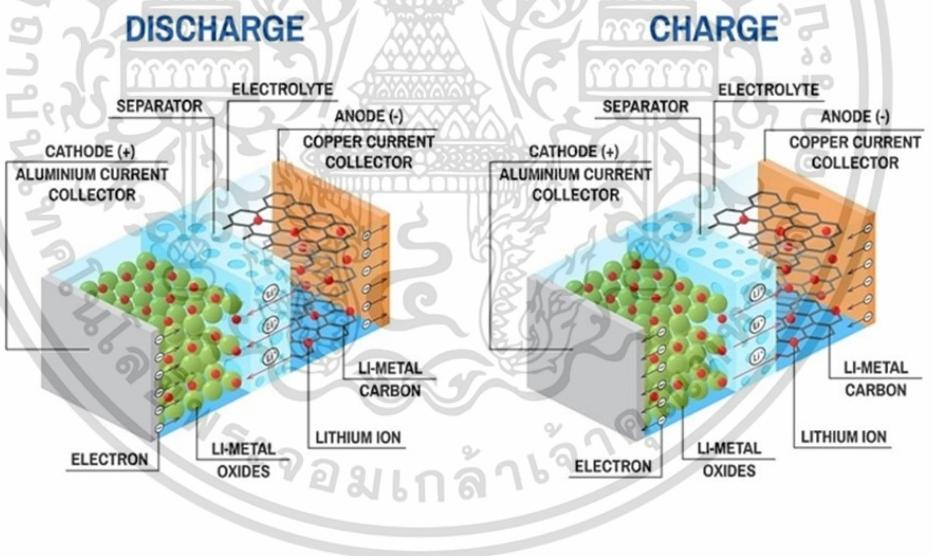
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 หลักการทำงานของแบตเตอรี่ลิเธียมไอออน

ทำงานโดยอาศัยหลักการทางไฟฟ้าเคมี ดังนี้

1) ขณะประจุไฟฟ้าดังรูปที่ 2.3 พลังงานไฟฟ้าที่ประจุเข้าไปทำให้เกิดปฏิกิริยาเคมีภายในแบตเตอรี่ ทำให้ลิเธียมไอออนไหลออกจากแคโทด (ขั้วที่เกิดปฏิกิริยาออกซิเดชัน ผ่านอิเล็กโทรไลต์แผ่นกั้นแบตเตอรี่ และเข้าไปแทรกตัวอยู่ในขั้วแอโนด (ขั้วที่เกิดปฏิกิริยารีดักชัน ขณะเดียวกันอิเล็กตรอนจะไหลจากขั้วแคโทดไปขั้วแอโนดเพื่อให้เกิดความสมดุลของประจุ (Charge balance) โดยผ่านทางวงจรไฟฟ้าภายนอกด้วย [7]

2) ขณะคายประจุไฟฟ้าดังรูปที่ 2.3 ปฏิกิริยาเคมีในแบตเตอรี่สามารถเกิดขึ้นได้เอง (Spontaneous reaction) โดยลิเธียมไอออนขณะประจุที่ไหลออกมาจากขั้วแคโทด และไปแทรกตัวอยู่ที่ขั้วแอโนดจะไหลออกจากแอโนดกลับเข้าไปอยู่ในขั้วแคโทดตามเดิม ทำให้ระบบมีเสถียรภาพอีกครั้ง พร้อมกับให้อิเล็กตรอนผ่านวงจรไฟฟ้า (โดยที่อิเล็กตรอนจะไหลผ่านตัวรับกระแส และให้พลังงานไฟฟ้าออกมา เมื่อใดที่ลิเธียมไอออนไหลกลับไปที่เดิมจนหมด ปฏิกิริยาจะสิ้นสุดลง (ถ่านหมด หากต้องการนำแบตเตอรี่ไปใช้ใหม่ ต้องทำการประจุไฟฟ้าอีกครั้ง และเกิดเป็นวัฏจักรเช่นนี้ต่อไปจนกว่าแบตเตอรี่จะหมดอายุการใช้งาน



รูปที่ 2.3 แสดงการประจุไฟฟ้าและคายประจุไฟฟ้าของแบตเตอรี่ลิเธียมไอออน

ที่มา : https://www.123rf.com/photo_83157939_li-ion-battery-diagram.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 คุณสมบัติของแบตเตอรี่ที่ใช้ในการศึกษา

คุณสมบัติหลักของแบตเตอรี่ที่ใช้ในการศึกษาประกอบไปด้วย 3 คุณสมบัติหลักคือ State of Charge, Open Circuit Voltage และ State of Health โดยแต่ละคุณสมบัติมีรายละเอียดดังต่อไปนี้

2.2.1 State of Charge

ถูกกำหนดเป็นอัตราส่วนระหว่างผลต่างแรงดันไฟฟ้าที่วัดได้จริง ($V_{act} - V_{min}$) ต่อผลต่างแรงดันไฟฟ้าสุทธิ ($V_{max} - V_{min}$) ดังนั้น SoC สามารถอธิบายสถานะการชาร์จปัจจุบันของเซลล์ได้ โดยความสัมพันธ์ต่อไปนี้ [8]

$$\mathrm{SoC} = \frac{V_{act} - V_{min}}{V_{max} - V_{min}} \times 100\% \quad (1)$$

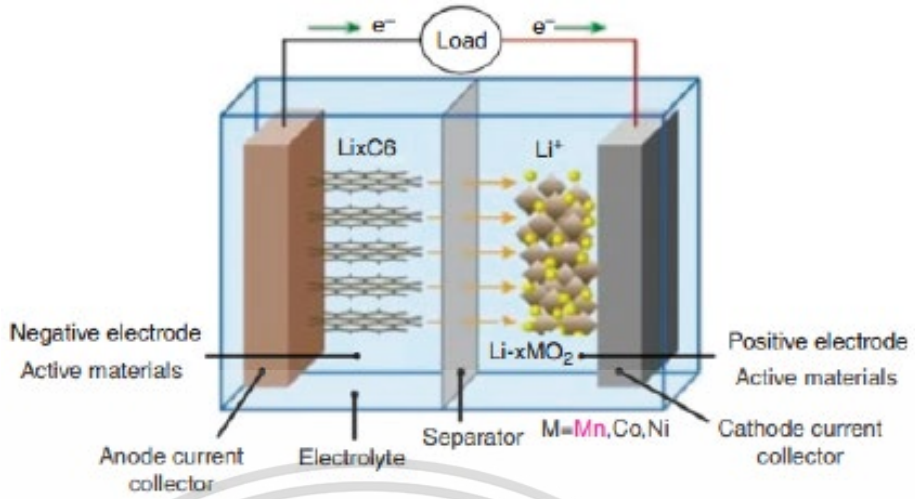
โดยที่ SoC อยู่ในช่วง 0% - 100% ซึ่งจะต้องพิจารณาว่าเมื่อใดที่เซลล์ถูกชาร์จจนเต็มและถูกปล่อยออกมาจนหมด

โดยทั่วไปการใช้อัตราการคายประจุที่สูงขึ้นจะไปเพิ่มขั้วโพลาไรเซชันของเซลล์เนื่องจากอิมพีแดนซ์ ซึ่งส่งผลให้เซลล์ถึงแรงดัน Cut – off เร็วขึ้นและปริมาณประจุไฟฟ้าที่ดึงออกมาจากเซลล์จะลดลง ดังนั้นคุณสมบัติเฉพาะเหล่านี้ของเซลล์จะอยู่ภายใต้เงื่อนไขที่กำหนดซึ่งมักจะจัดทำโดยผู้ผลิต สำหรับเซลล์ที่ใช้ศึกษาในปริญญานิพนธ์ฉบับนี้รายละเอียดของข้อมูลจะอยู่ในภาคผนวก

2.2.2 Open Circuit Voltage

เมื่อเซลล์ถูกตัดการเชื่อมต่อจากวงจรภายนอก เซลล์ดังกล่าวจะอยู่ในสภาพแรงดันไฟฟ้าวงจรเปิด หมายความว่าไม่มีกระแสไฟฟ้าไหลผ่านเซลล์และแรงดันไฟฟ้าที่ขั้วของเซลล์ได้รับอนุญาตให้พักผ่อนพร้อมกับความสมดุลทางไฟฟ้าเคมี ลิเธียมไอออนในอิเล็กโทรไลต์นั้นอยู่ในภาวะสมดุลกับลิเธียมที่เข้ามาแทรกซึมเข้าไปในโครงร่างของขั้วไฟฟ้านั้น ๆ (รูปที่ 2.4 [9]

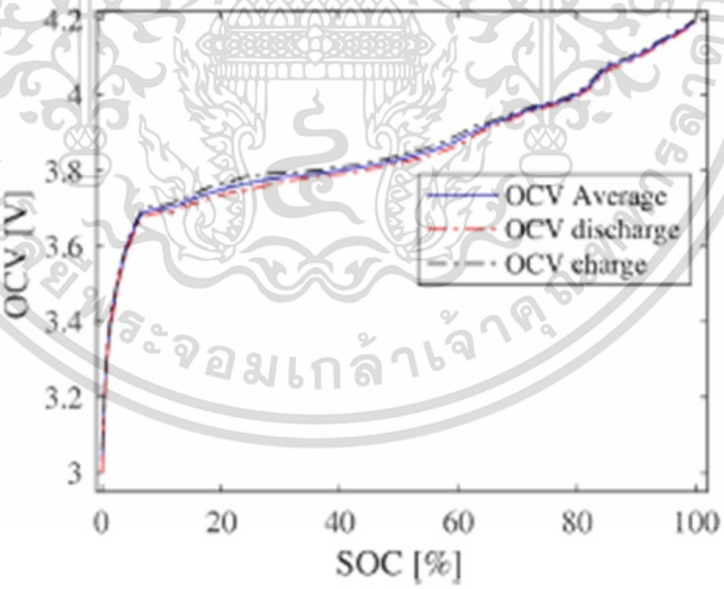
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงกระบวนการคายประจุของเซลล์ลิเธียมไอออน

ที่มา : https://www.pluginindia.com/lithiumion.html

มีความสัมพันธ์ระหว่าง Open Circuit Voltage (OCV และ SoC นั่นคือ OCV ของเซลล์ไฟฟ้าเคมีแสดงความไม่เป็นเชิงเส้นบนเซลล์ SoC ดังแสดงในรูปที่ 2.5 อย่างไรก็ตามอุณหภูมิยังมีผลกับแรงดันไฟฟ้าวงจรเปิดเล็กน้อย



รูปที่ 2.5 แรงดันไฟฟ้าวงจรเปิด (OCV) เทียบกับสถานะประจุ (SoC) ที่มีการใช้งานก่อนหน้านี้ที่ 25 °C

ที่มา : Gjengedal (2017 : 13) Open-circuit voltage (OCV) vs State of Charge (SoC). Established in previous work at 25 °C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 State of Health

สถานะของสุขภาพ (SoH) เป็นตัวบ่งบอกของอายุการใช้งานแบตเตอรี่ โดยทั่วไปสถานะของสุขภาพได้รับการประเมินโดยพิจารณาจากการลดลงของอายุการใช้งานแบตเตอรี่ซึ่งเป็นผลมาจากพฤติกรรมการใช้งานเป็นหลัก โดยการประเมิน SoH ของสถานการณ์ปัจจุบันจะทำให้สามารถคาดการณ์ประสิทธิภาพในอนาคตของเซลล์แบตเตอรี่ได้เช่นเดียวกับอายุการใช้งานที่เหลืออยู่ [10]

สถานะของสุขภาพหมายถึงความสัมพันธ์ระหว่างอายุการใช้งานจริงต่ออายุการใช้งานทางทฤษฎี

$$\text{SoH} = \frac{Life\ cycle - N}{Life\ cycle} \times 100\% \quad (2)$$

โดยที่ $Life\ cycle$ คืออายุการใช้งานของแบตเตอรี่ที่ระบุจากผู้ผลิตและ N คือจำนวนครั้งที่แบตเตอรี่ชาร์จจนเต็มและคายประจุจนหมด มันถูกกำหนดโดยพารามิเตอร์ของแบตเตอรี่ที่แตกต่างกันหลายอย่าง เช่นความต้านทานภายใน, การคายประจุไฟฟ้าด้วยตัวเอง, การยอมรับการชาร์จและการเปลี่ยนแปลงทางเคมี ผู้ใช้สามารถกำหนดค่าขีดจำกัดของ SoH สำหรับแบตเตอรี่ตามแผ่นข้อมูลได้ สถานะสุขภาพเริ่มลดลงตามการเสื่อมประสิทธิภาพของเซลล์แบตเตอรี่ เมื่อ SoH ลดลงต่ำกว่าค่าขีดจำกัดเซลล์แบตเตอรี่นี้จะหมดอายุการใช้งาน โดยรายละเอียดของแบตเตอรี่ที่ใช้ในการศึกษาสามารถดูได้จากภาคผนวก

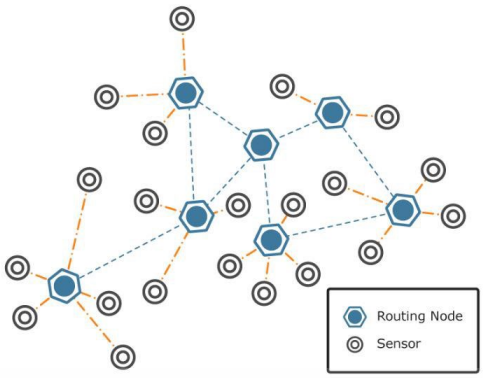
2.3 The Internet of Things (IoT)

Internet of Things คือเทคโนโลยีที่ทำให้อุปกรณ์ต่าง ๆ เช่น โทรศัพท์ อุปกรณ์อิเล็กทรอนิกส์ สามารถติดต่อระหว่างกันและทำงานร่วมกันได้อย่างอัตโนมัติผ่านออนไลน์ต่าง ๆ อาธิเช่น บลูทูธ, RFID และ WiFi [11] - [13]

2.4 Sensor node

คือตัวรับอินพุตจากตัวเซนเซอร์หรือทรานสดิวเซอร์ชนิดต่าง ๆ และสามารถใช้เป็นตัวสั่งเอาต์พุต เพื่อนำไปควบคุมอุปกรณ์ทางไฟฟ้า (รูปที่ 2.6) [14]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

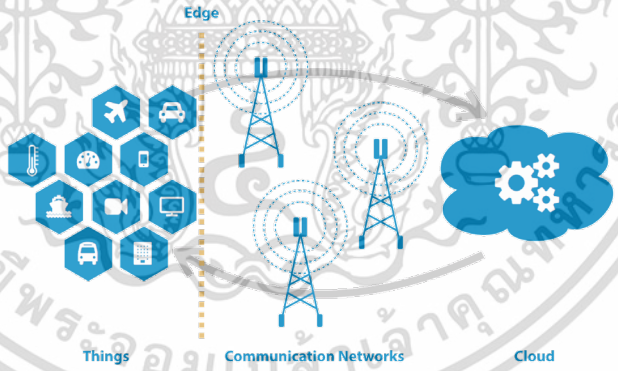


รูปที่ 2.6 แสดง Sensor node

ที่มา : https://maneeporn24.wordpress.com/2015/11/04/internet-of-things/

2.5 Edge computing

คือการประมวลผลข้อมูลที่อยู่ใกล้กับแหล่งข้อมูลให้มากที่สุด อาจจะอยู่ในรูปแบบการวิเคราะห์ข้อมูล การประมวลผลข้อมูลเชิงสถิติ ซึ่งแทนที่จะเอาข้อมูลจำนวนมหาศาลขึ้นไปประมวลผลบน Cloud ก็เอาข้อมูลเหล่านั้นมาประมวลผลที่ต้นทางที่ใกล้กับแหล่งข้อมูลมากที่สุด หรือ Edge ดังรูปที่ 2.7 [15]



รูปที่ 2.7 แสดง Edge computing ที่อยู่ใกล้กับแหล่งข้อมูล หรือ ระบบ

ที่มา : https://thaiopensource.org/

สาเหตุหลักที่ทำให้การประมวลผลจำเป็นต้องอยู่ที่ต้นทาง หรือ Edge

1) เวลาในการรับส่งข้อมูล (Latency)

เป็นเหตุผลที่ทำให้ Edge computing ได้เปรียบในด้านความเร็วในการรับส่งข้อมูล ยกตัวอย่างเช่น หากเครื่องคอมพิวเตอร์ต้องการส่งข้อมูลไปหาเครื่องคอมพิวเตอร์อีกเครื่องที่อยู่คนละฝั่งโลก ความเร็วในการรับส่งข้อมูลเป็นตัวแปรสำคัญ ซึ่งการรับส่งข้อมูลในระยะที่ใกล้จะมีความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากกว่าส่งในระยะไกลอย่างแน่นอน ถ้าส่งข้อมูลไปประมวลผลบนเซิร์ฟเวอร์ที่อยู่บน Cloud ทุกครั้งเพื่อหาคำตอบ จากบริการ API ต่าง ๆ บน Cloud หากบริการนั้นมีความซับซ้อนก็จะใช้เวลานานในการค้นหาคำตอบสาเหตุหลักที่ทำให้การประมวลผลจำเป็นต้องอยู่ที่ต้นทาง หรือ Edge

2) ความเป็นส่วนตัวและความปลอดภัย (Privacy & Security)

ความเป็นส่วนตัวและความปลอดภัยเป็นประเด็นที่ทุกคนให้ความสำคัญ การที่ไม่จำเป็นต้องเอาข้อมูลส่งออกไปเก็บหรือประมวลผลที่อื่น การบริหารจัดการอยู่ที่จุดเดียวสามารถช่วยในเรื่องความปลอดภัยของข้อมูลได้

3) แบนด์วิดท์ (Bandwidth)

เรื่องการใช้งานแบนด์วิดท์และประหยัดแบนด์วิดท์การรับส่งข้อมูลไปยัง Cloud ยกตัวอย่างอุปกรณ์อัจฉริยะจำนวนมหาศาลหากต้องการส่งข้อมูลไปยัง Cloud พร้อม ๆ กันต้องใช้แบนด์วิดท์จำนวนมหาศาลตามไปด้วย หากการประมวลผลข้อมูลอยู่ที่ Edge เช่น มี AI อยู่ที่อุปกรณ์ของผู้ใช้สามารถประมวลผลข้อมูลในตัวได้ การรับส่งข้อมูลขึ้นไปประมวลผลบน Cloud ก็จะลดลง สามารถแก้ปัญหาการใช้แบนด์วิดท์ของเครือข่าย และแก้ปัญหาเรื่อง Latency ได้

2.6 Cloud computing

Cloud computing คือบริการที่ครอบคลุมถึงการให้ใช้กำลังประมวลผล หน่วยจัดเก็บข้อมูลและระบบออนไลน์ต่าง ๆ จากผู้ให้บริการเพื่อลดความยุ่งยากในการติดตั้ง, ดูแลระบบ, ช่วยประหยัดเวลา และลดต้นทุนในการสร้างระบบคอมพิวเตอร์และเครือข่ายเอง ซึ่งมีทั้งแบบบริการฟรีและแบบเก็บเงิน [16] - [17]

ประเภทของบริการคลาวด์คอมพิวติ้ง (Cloud service models) สามารถแบ่งออกได้เป็น 3 ประเภทดังนี้

1) SaaS (Software as a service) เป็นรูปแบบการให้บริการใช้ซอฟต์แวร์หรือแอพพลิเคชั่นบน Cloud ทำให้ผู้ใช้ที่ออนไลน์บนเครือข่ายอินเทอร์เน็ตใช้บริการซอฟต์แวร์เหล่านี้ได้โดยไม่จำเป็นต้องติดตั้งซอฟต์แวร์ไว้ที่หน่วยงานหรือคอมพิวเตอร์ของผู้ใช้

2) IaaS (Infrastructure as a service) เป็นบริการเฉพาะโครงสร้างพื้นฐานของระบบ เช่น หน่วยประมวณผล (Processing unit) เครือข่ายข้อมูล (Network) ระบบเก็บข้อมูล (Storage) หรือพื้นที่เซิร์ฟเวอร์ (Hosting) ผู้ใช้บริการจะสามารถเช่าเวลาในการประมวลผล ซื้อเวลาและขนาดของช่องสัญญาณในการส่งข้อมูลหรือขนาดของพื้นที่เก็บข้อมูลจากผู้บริโภคได้ ยกตัวอย่างเช่น พื้นที่เซิร์ฟเวอร์ของ Gmail และ Amazon เป็นต้น ซึ่งโครงสร้างพื้นฐานในระบบกลุ่มเมฆนั้นสามารถปรับเปลี่ยนได้ตามสถานะการณ์และความต้องการของผู้ใช้โดยผ่าน Cloud API ของผู้ให้บริการ เช่น การเช่าพื้นที่เพื่อสร้างเว็บไซต์ในกลุ่มเมฆ ถ้าเกิดความต้องการใช้งานเว็บไซต์นั้นเพิ่มขึ้นอย่างรวดเร็ว ระบบกลุ่มเมฆจะใช้ระบบ Virtualization ภายในกลุ่มเมฆ ในการขยายจำนวนหน่วยประมวณผลและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดของช่องสัญญาณของเว็บไซต์ได้อัตโนมัติ และผู้รับบริการ จะต้องเสียค่าใช้จ่ายเพิ่มขึ้นตามการใช้งานของทรัพยากรที่เพิ่มมากขึ้นได้โดยไม่ล่มเสียก่อน

3) PaaS (Platform as a service) บริการแพลตฟอร์ม คือให้บริการนักพัฒนาในการพัฒนาโปรแกรมโดยผู้รับบริการสามารถพัฒนาโปรแกรมระบบ ได้แก่ บริการ Google App engine ซึ่งผู้รับบริการสามารถสร้างโปรแกรมประยุกต์ประเภท Web application บนเว็บที่มีอัตราการเข้าชมสูง โดยไม่ต้องจัดโครงสร้างพื้นฐานสำหรับอัตราการเข้าชมที่สูง การเขียนโปรแกรมนั้นนักพัฒนาสามารถใช้ภาษา Java หรือ Python แล้วโฮสต์โปรแกรมบนเซิร์ฟเวอร์ของ Google ได้โดยมีค่าบริการตามจำนวน Transaction หรือ Data storage [18]

2.7 Database

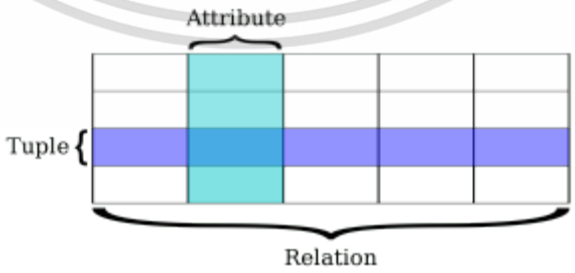
คือส่วนที่ใช้เก็บข้อมูล เพื่อให้สามารถบริการการจัดเก็บ และการเข้าถึงข้อมูลส่วนกลางได้สะดวกรวดเร็ว ซึ่งข้อมูลจะถูกเก็บในรูปของ Rows, Columns และ Tables โดยอาศัยดรรชนี (Indexed) เพื่อให้สามารถค้นหาข้อมูลได้รวดเร็ว นอกจากนี้ยังสามารถทำการเพิ่ม (Insert), ลด (Delete) หรือ เปลี่ยนแปลง (Update) ข้อมูลใหม่ ปัจจุบัน Computer database ถูกใช้ในการเก็บข้อมูลแทนในรูปแบบไฟล์ [19]

1) วิวัฒนาการของ Database

Database ถูกพัฒนาขึ้นมาตั้งแต่ปี 1960 เริ่มต้นจาก Hierarchical และ Network databases จนมาถึงปี 1980 มีการนำเอา Object-oriented-databases (OODBMS) มาใช้งาน ซึ่งเป็นพื้นฐานของระบบ Relation database ซึ่งมีการใช้อยู่ในปัจจุบัน

สามารถจัดแบ่งประเภทของ Database ตามรูปแบบของชนิดข้อมูลได้ เช่น ตัวเลข, ตัวอักษร หรือ รูปที่ บางครั้งก็อาจจะแบ่งตามความนิยมของ Relational database เช่น Distributed database, Cloud database หรือ NoSQL database

2) Relational database



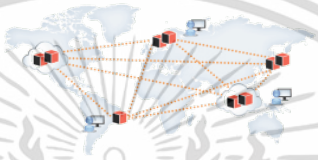
รูปที่ 2.8 แสดงโครงสร้าง Relational database
ที่มา : https://www.glurgeek.com/education/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Relational database ถูกคิดค้นขึ้นโดย E.F. Codd (IBM) ในปี 1970 เริ่มต้นสร้างขึ้นมาจากกลุ่มของ Table ที่มีข้อมูลภายในโดยแบ่งออกเป็นตามประเภทที่ตั้งไว้ แต่ละ Table จะมีอย่างน้อย 1 ชนิดของแต่ละ Column และแต่ละ Row จะมีข้อมูลตามที่ชนิดที่ Columns ได้กำหนดไว้ (รูปที่ 2.8)

Standard Query Language (SQL) เป็นมาตราฐานที่ผู้ใช้งาน และ ระบบอื่น ๆ ไว้เชื่อมต่อกับ relational database ซึ่งง่ายต่อการเพิ่มข้อมูลเข้าไป โดยไม่กระทบต่อโปรแกรมอื่นที่ใช้งานร่วมกัน

3) Distributed database (รูปที่ 2.9)



รูปที่ 2.9 แสดงโครงข่าย Distributed database
ที่มา : https://www.glurgeek.com/education/

Distributed database เป็น ฐานข้อมูลที่ถูกเก็บกระจายออกไปหลาย ๆ ที่ โดยอาศัยกระบวนการแจกจ่าย และ สำรองข้อมูล ผ่านทางระบบ network ซึ่งมีอยู่ด้วยกัน 2 แบบคือ

Homogeneous - ระบบทั้งหมดทุกที่ต้องเป็น OS และ database ชนิดเดียวกัน

Heterogeneous – ระบบที่งหมดจะเหมือนหรือต่างกันในแต่ละที่

4) Cloud database

Cloud database เป็นฐานข้อมูลแบบใหม่ที่ถูกปรับปรุงและสร้างขึ้นบนระบบ Virtualized แบบเดียวกับ Hybrid cloud, Public cloud หรือ Private cloud โดยสามารถขยายขนาดเพิ่มขึ้น หรือ ปรับแต่ง Resource ได้ตลอดเวลา ขึ้นอยู่กับความต้องการของผู้ใช้งาน

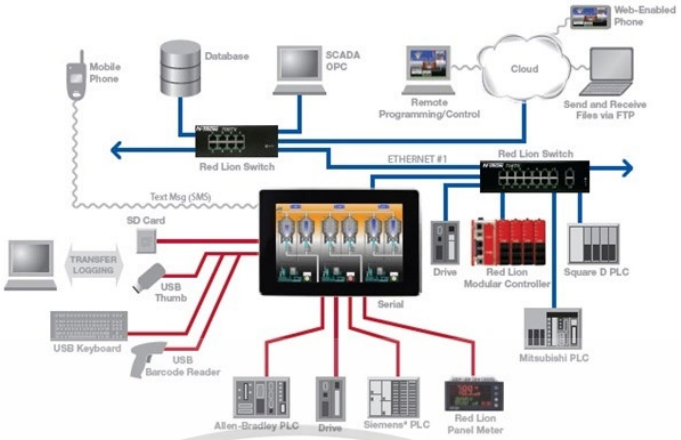
5) NoSQL database

NoSQL database ถูกใช้ในรูปแบบ ที่เป็นการกระจายของข้อมูล จึงมีประสิทธิ์ภาพสูงสำหรับข้อมูลขนาดใหญ่ (Big data) เพราะ Relational database ไม่ถูกออกมาให้รอบรับข้อมูลขนาดใหญ่ จึงนิยมใช้กับการวิเคราะห์ข้อมูลขนาดใหญ่ที่ไม่ค่อยมีรูปแบบตายตัว

2.8 Human Machine Interface

เป็นการทำงานร่วมกันโดยใช้ PLC เป็นตัวควบคุม และ HMI เป็นตัวสื่อสารระหว่างผู้ใช้งาน (Human) กับระบบ Module PLC หรือจอแสดงผลต่าง ๆ โดยให้ PLC สั่งงานไปที่เครื่องจักร (Machine) ดังรูปที่ 2.10 [20]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



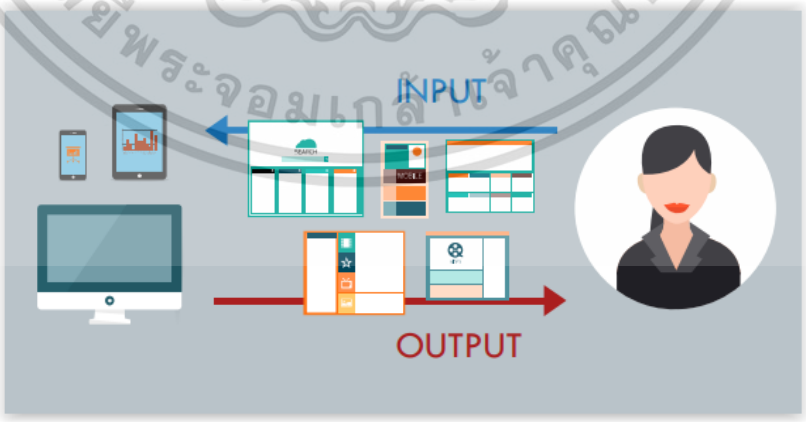
รูปที่ 2.10 แสดงการเชื่อมต่อสื่อสารข้อมูลของ PLC แบบ Touch screen ในการเข้าถึงแอปพลิเคชันต่าง ๆ ของผู้ใช้งานจากระยะไกล

ที่มา : http://www.energyscopethai.com/hmi-programming/

User interface หมายถึง สื่อกลางในการติดต่อและโต้ตอบระหว่างผู้ใช้กับระบบคอมพิวเตอร์ ไม่ว่าจะเป็นการติดต่อทางด้านฮาร์ดแวร์หรือซอฟต์แวร์ หากเป็นสื่อทางด้านฮาร์ดแวร์จะประกอบไปด้วยอุปกรณ์คอมพิวเตอร์ต่าง ๆ เช่น เมาส์คีย์บอร์ด ลำโพง เครื่องพิมพ์พ์เป็นต้น แต่หากเป็นสื่อทางด้านซอฟต์แวร์จะหมายถึงลกัษณะการแสดงผลทางจอภาพ และทำให้เกิดปฏิสัมพันธ์กันดังรูปที่ 2.11

สามารถจัดได้เป็น 2 ประเภทใหญ่ได้แก่

1) ส่วนที่นำข้อมูลเข้า หรือส่วนสั่งงาน เรียกว่า อินพุต (Input)
2) ส่วนที่ใช้แสดงผลลัพธ์ หรือส่วนที่ไว้รอคำสั่งจากผู้ใช้ เรียกว่า เอาต์พุต (Output)



รูปที่ 2.11 ส่วนติดต่อระหว่างผู้ใช้กับกับระบบ

ที่มา : https://mis.csit.sci.tsu.ac.th/mallika/UI/DL/Ch02-Characteristics%20of%20UI.pdf

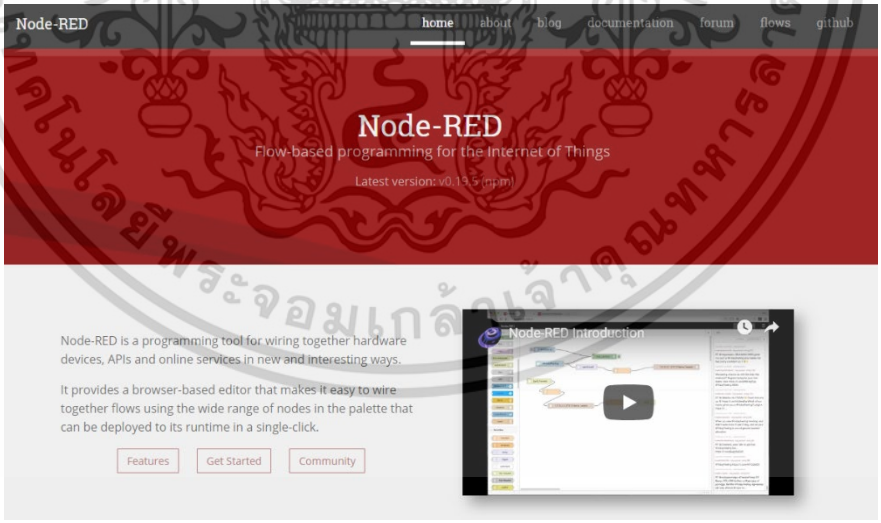
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 Node-Red

เป็น Medium-ware โดยมีพื้นฐานการใช้งานในรูปแบบ Flow-based programming ที่พัฒนาขึ้นมาเพื่อการใช้งานแบบ Internet of Things [21]

สำหรับการใช้งาน Node-Red นั้นมีความสะดวกมากเพราะสามารถแก้ไข Flow โปรแกรมต่าง ๆ ผ่าน Browser ได้เลย และเนื่องจาก Node-Red ถูกพัฒนาบน Javascript (Node.JS) จึงทำให้สามารถพัฒนาขีดความสามารถออกไปได้ดังรูปที่ 2.12 – 2.13 [22]

รูปที่ 2.12 หน้าต่างการเรียกใช้งานโปรแกรม Node-Red



รูปที่ 2.13 หน้าต่างเว็บไซต์สำหรับดาว์นโหลดและติดตั้งโปรแกรม Node-Red
ที่มา : https://nodered.org/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 Message Queuing Telemetry Transport (MQTT)

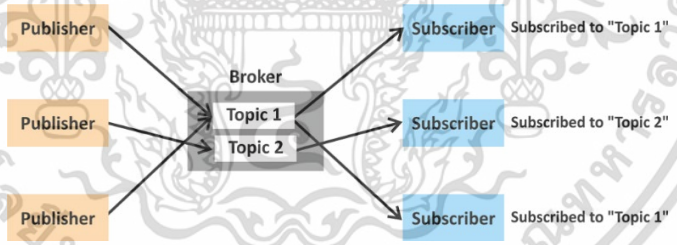
เป็นโปรโตคอลที่ถูกออกแบบมาให้มีขนาดเล็กสำหรับการสื่อสารแบบ M2M (Machine to Machine) โดยถือกำเนิดจากวิศวกรจาก IBM และ Eurotech ในปี 1999 เพื่อนำไปใช้ในระบบ SCADA (Supervisory Control and Data Acquisition) สำหรับเชื่อมต่อท่อส่งน้ำมันบนเครือข่ายที่ไม่มีความเสถียรอย่างอินเตอร์เน็ตดาวเทียมก่อนที่จะถูกบริจาคกลายเป็น Open standard ในปี 2014 โดย OASIS [23]

MQTT เป็นสถาปัตยกรรมแบบ Client/Server ซึ่งมี Topology แบบ Hub-and-spoke sensor ปลายทางจะทำหน้าทีเป็น Client ซึ่งทำการสร้างเชื่อมต่อแบบ TCP ไปยัง Server ที่มีชื่อเรียกอีกชื่อว่า Broker ซึ่งมีหน้าที่เป็นเสมือนท่อส่งข้อมูลในการรับส่ง Message ระหว่าง Client ที่เป็นได้ทั้ง Publisher และ Subscriber

Client หมายถึง Publisher หรือ Subscriber ที่เชื่อมต่อแบบรวมศูนย์ไปยัง Broker ซึ่งสามารถเชื่อมต่อได้ทั้งแบบ Persistent ที่ทำการสร้าง Session ค้างไว้เปิดตลอดเวลาเพื่อติดต่อกับ Broker ซึ่งตรงกันข้ามกับ Client ที่เชื่อมต่อแบบ Transient ซึ่ง Broker ไม่สามารถติดตามสถานะได้

Broker เป็น Software ที่ทำหน้ารับข้อความทั้งหมดที่ได้จาก Publisher แล้วจึงส่งต่อไปให้ Subscriber ตามแต่ Topic ที่ Client ได้ทำการ Subscribe

Topic เป็นเหมือน Address หรือ Endpoint บน Broker ที่ Client ทำการเชื่อมต่อเพื่อรับส่งข้อความระหว่างกัน

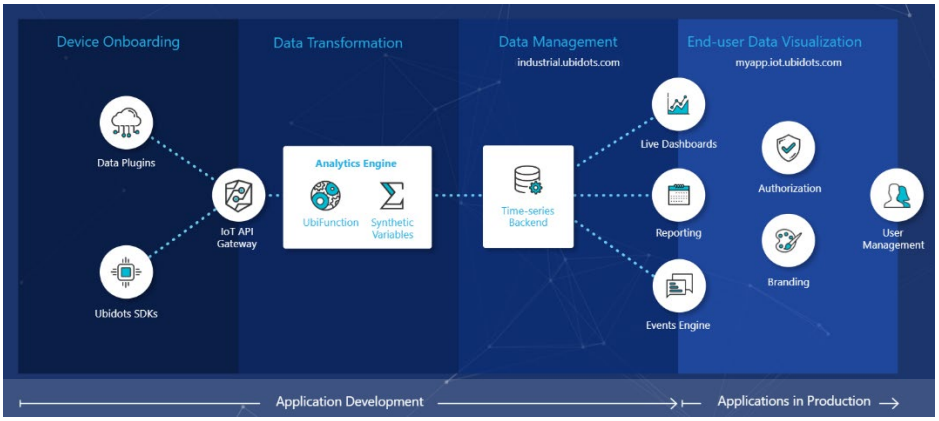


รูปที่ 2.14 แสดงความสัมพันธ์ระหว่าง Publisher, Broker, Topic และ Subscriber
ที่มา : http://www.adslthailand.com/post/mqtt-coap-comparison-iot- protocol

2.11 Ubidots

เป็นผู้ให้บริการ IoT และ Cloud applications แบบ Platform as a Service ที่ช่วยให้ผู้ใช้งานสามารถสร้างแอปพลิเคชันขึ้นเองได้, รวบรวมวิเคราะห์และแสดงภาพเซ็นเซอร์หรือข้อมูลดิจิทัลเพื่อการนำเสนอ และสามารถเชื่อมต่อเข้ากับฮาร์ดแวร์และอินพุตดิจิตอลได้ผ่านโปรโตคอล HTTP / MQTT / TCP / UDP สามารถสรุปการทำงานของ Ubidots ได้ดังรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

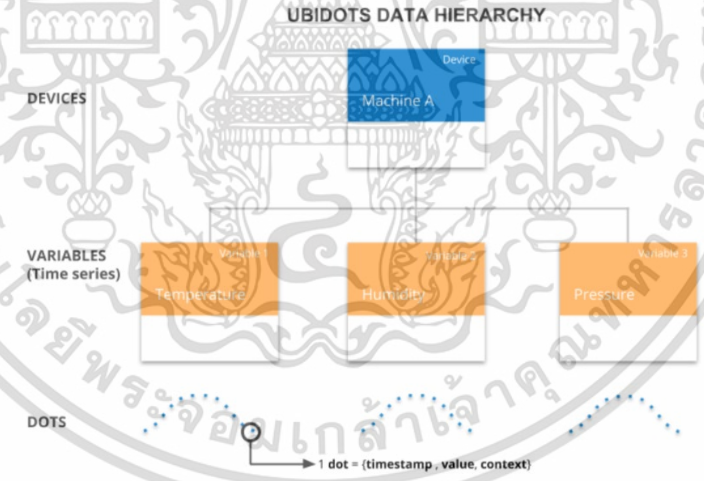


รูปที่ 2.15 แสดงการทำงานของ Ubidots

ที่มา : https://ubidots.com/docs/hw/?language=ESP#introduction

2.11.1 หลักการทำงานของ Ubidots

เมื่ออุปกรณ์ (Device) มีการอัพเดทค่าตัวแปร (Variable) ที่วัดได้จากเซ็นเซอร์จะมีการสร้างจุดข้อมูล (Data-point) หรือจุด (Dot) จากนั้น Ubidots จะทำการเก็บบันทึกจุดที่รับมาจากอุปกรณ์ภายในตัวแปรโดยจุดที่จัดเก็บเหล่านี้มีการประทับเวลา (Timestamps) ที่สอดคล้องกัน



รูปที่ 2.16 แสดงลำดับชั้นข้อมูลของ Ubidots

ที่มา : https://ubidots.com/docs/hw/?language=ESP#introduction

จากรูปที่ 2.16 แต่ละจุด (Dots) จะประกอบไปด้วยรายการ Value, Timestamp และ Context โดยแต่ละรายการมีรายละเอียดดังต่อไปนี้ :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงรายละเอียดรายการของแต่ละจุด

รายการ (Item)	รายละเอียด (Description)	ความจำเป็น (Mandatory)
Value	ค่าเชิงตัวเลข Ubidots ยอมรับความยาวเลขทศนิยมชนิด Floating Point สูงสุด 16 ตัว	YES
Time stamp	เวลา Unix Epoch มีหน่วยเป็นมิลลิวินาที หากไม่ได้ระบุเซิร์ฟเวอร์ของ Ubidots จะกำหนดหนึ่งตัวเมื่อรับ	NO
Context	ส่วนใหญ่ใช้ในการจัดเก็บพิกัดละติจูดและลองจิจูดของอุปกรณ์ GPS	NO

1) ค่า (Values)

ค่าตัวเลข Ubidots ยอมรับตัวเลข floating-point ความยาวสูงสุด 16 ตัว เช่น {"value" : 34.87654974}

2) การประทับเวลา (Timestamps

การประทับเวลาเป็นวิธีในการติดตามเวลาในฐานะจำนวนวินาที โดยเริ่มต้นที่ Unix Epoch เมื่อวันที่ 1 มกราคม 1970 เวลา UTC ดังนั้นการประทับเวลา Unix เป็นเพียงจำนวนวินาทีระหว่างวันที่ใดวันหนึ่งกับ Unix Epoch ดังนั้นเมื่อต้องการส่งข้อมูลไปยัง Ubidots จะต้องตั้งค่าการประทับเวลาเป็นมิลลิวินาที นอกจากนี้หากต้องการดึงข้อมูลเวลาของจุดนั้น ข้อมูลเวลาที่ได้จะแสดงเวลาในหน่วยมิลลิวินาที

"timestamp" : 1537453824000

การประทับเวลาด้านบนตรงกับวันพฤหัสบดีที่ 20 กันยายน 2018 2:30:24 PM

3) Context

นอกจากข้อมูลชนิดตัวเลขแล้ว ผู้ใช้งานยังสามารถจัดเก็บข้อมูลชนิด String หรือข้อมูลชนิด Char ภายใน Context ได้ ตัวอย่างการใช้ Context

"context" : {"status" : "on", "weather" : "sunny"}

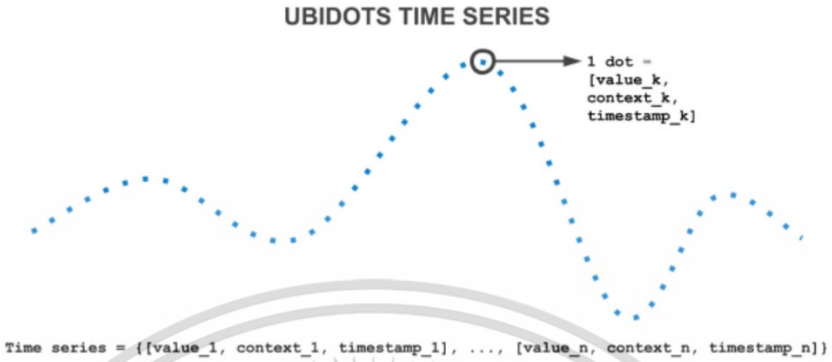
โดยทั่วไป Context จะใช้ในการจัดเก็บพิกัดละติจูดและลองจิจูดของอุปกรณ์ของผู้ใช้ สำหรับกรณีการใช้งาน GPS หรือ แอปพลิเคชันการติดตาม แผนที่ Ubidots ใช้พิกัดละติจูดและลองจิจูดจากcontextของจุดเพื่อแยกพิกัดของอุปกรณ์ ด้านล่างจะแสดง Context ที่มีค่าพิกัด :

"context" : {"lat":-6.2, "lng":75.4, "weather" : "sunny"}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.2 อนุกรมเวลา (Time Series)

อนุกรมเวลาของ Ubidots สามารถแสดงได้ดังรูปที่ 2.17



รูปที่ 2.17 แสดงอนุกรมเวลาของ Ubidots

ที่มา : https://ubidots.com/docs/hw/?language=ESP#introduction

2.11.3 MQTT

MQTT คือโปรโตคอลการเชื่อมต่อ Internet of Things ซึ่งถูกออกแบบมาให้มีน้ำหนักเบา การขนส่งข้อความ Publish/Subscribe มีประโยชน์สำหรับการเชื่อมต่อกับสถานที่ห่างไกลที่จำเป็นต้องใช้รหัส/แบนด์วิดท์เครือข่ายคุณภาพสูง

MQTT มีประโยชน์เป็นพิเศษในการส่งข้อมูลจากคลาวด์ไปยังอุปกรณ์ของคุณ อุปกรณ์สามารถ "ฟัง" ไปยังคลาวด์และรับการแจ้งเตือนเมื่อมีการเปลี่ยนแปลงของตัวแปรเท่านั้น วิธีนี้การเชื่อมต่อระหว่างอุปกรณ์และคลาวด์จะเปิดทิ้งไว้ แต่ข้อมูลจะเดินทางเมื่อจำเป็นเท่านั้นเพื่อประหยัดแบตเตอรี่, แบนด์วิดท์เครือข่ายและแก้ไขแบบเรียลไทม์

1) MQTT Authentication

ในการโต้ตอบกับ Ubidots MQTT broker ผู้ใช้งานจะต้องมี Token ซึ่ง Token เป็นคีย์เฉพาะที่อนุญาตให้อุปกรณ์ของผู้ใช้งานโต้ตอบกับ Ubidots API

2) การส่งข้อมูล (Publish

Ubidots ยอมรับข้อมูล MQTT, JavaScript, Object Notation หรือ JSON ซึ่งเป็นคอลเลกชันของคู่ชื่อหรือค่าในภาษาการเขียนโปรแกรมต่าง ๆ ซึ่งจะถือว่าเป็น Object, Record, Struct, Dictionary, Hash table, Keyed list หรือ Associative array ซึ่งเป็นภาษาที่มนุษย์อ่านได้และเป็นภาษาอิสระ ตัวอย่างของชนิดข้อมูล JSON ที่ Ubidots ยอมรับสามารถอ้างอิงได้ดังด้านล่าง :

{"temperature": {"value":10, "timestamp": 1534881387000, "context": {"machine": "1st floor"}}}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

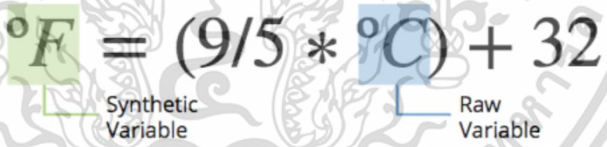
3) การดึงข้อมูล (Subscribe)

เช่นเดียวกับฟังก์ชัน GET ใน HTTP, Subscribe เป็นวิธีการรับค่าที่มีความแตกต่างอย่างมาก MQTT ไม่จำเป็นต้องถามเซิร์ฟเวอร์สำหรับค่าแต่ละค่าในสคริปต์ที่กำหนดเอง ตัวอย่างเช่น หากค่าของตัวแปรเปลี่ยนแปลง Ubidots จะอัปเดตการเปลี่ยนแปลงโดยอัตโนมัติ ดังนั้นการประหยัดการร้องขอข้อมูลและเวลาในการประมวลผลสำหรับอุปกรณ์ของผู้ใช้งานและการทำงานและค่าใช้จ่าย โดยรวมของโครงงาน MQTT จึงเป็นโปรโตคอลการสื่อสารที่แนะนำให้ใช้ Firmware routines เพื่อควบคุมอุปกรณ์ [24]

2.11.4 Analytics : ข้อมูลพื้นฐานเกี่ยวกับตัวแปรสังเคราะห์

Ubidots Analytics Engine รองรับเครื่องมือคำนวณทางคณิตศาสตร์ที่ซับซ้อนที่เรียกว่า Synthetic Variables ตัวแปรคือข้อมูลดิบหรือข้อมูลที่คำนวณได้ภายในอุปกรณ์ใน Ubidots Synthetic Variable เป็นตัวแปรที่เป็นผลมาจากการคำนวณตัวแปรอื่น ๆ ภายใน Ubidots [25]

เครื่องมือนี้มีประโยชน์อย่างมากในการขยายการทำงานของแอปพลิเคชันเช่นถ้าผู้ใช้งานพัฒนาเครื่องบันทึกอุณหภูมิที่อ่านตัวแปรเป็น °C และผู้ใช้งานต้องการแสดงข้อมูลทั้ง °C และ °F ผู้ใช้งานจะเรียกการเปลี่ยนแปลงเซ็นเซอร์และส่งค่าสองค่าไปยัง Ubidots หนึ่งค่าใน °C และอีกหนึ่งใน °F ซึ่งจะเป็นการเพิ่มโหลดการคำนวณเพิ่มเติมให้กับไมโครคอนโทรลเลอร์ของผู้ใช้งาน การใช้เครื่องมือวิเคราะห์ของ Ubidots ผู้ใช้งานต้องส่งค่าดิบเป็น °C เท่านั้นและให้ Ubidots ทำการคำนวณที่ต้องการเพื่อแปลงเป็น °F เพื่อลดความต้องการประมวลผลไมโครคอนโทรลเลอร์ที่ไม่จำเป็น



รูปที่ 2.18 แสดงตัวอย่างการคำนวณหน่วยองศาเซลเซียสเป็นหน่วยองศาฟาเรนไฮต์
ที่มา : https://help.ubidots.com/en/articles/1767999-analytics-synthetic-variables-basics

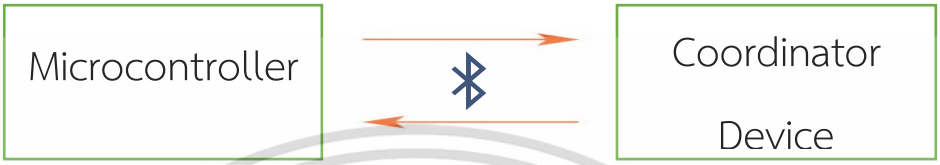
สำหรับการคำนวนตัวแปรสังเคราะห์อื่น ๆ สามารถศึกษาเพิ่มเติมได้จากภาคผนวก หัวข้อ Synthetic variable operations

2.12 การส่งข้อมูลแบบไร้สายผ่านบลูทูธ

Bluetooth เป็นมาตรฐานที่แตกต่างจากข้อกำหนดของมาตรฐานชุด IEEE 802.11 โดยเป้าหมายในการพัฒนาก็เพื่อความยืดหยุ่น ในการใช้งานและใช้ กับระบบเครือข่ายขนาดเล็ก ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เรียกว่า PAN (Personal Area Network) ซึ่งขนาดของ Throughput เท่ากับ 500Kbps และมีรัศมีการส่งสัญญาณที่ 10 เมตร อุปกรณ์ที่รองรับมาตรฐานนี้จึงเชื่อมโยงการทำงาน เพื่อที่จะเข้าถึงอินเตอร์เน็ตด้วยอุปกรณ์พกพา จำพวกโน้ตบุ๊ก พีดีเอ อุปกรณ์ที่รองรับเทคโนโลยี ในปัจจุบันนิยมนำมาใช้กับโทรศัพท์มือถือและอุปกรณ์พกพาต่าง ๆ นอกจากนั้นก็ยังจะขยายไปถึงอุปกรณ์เครื่องใช้ไฟฟ้า ไมโครคอนโทรลเลอร์ ซึ่งเข้ามาแทนที่เทคโนโลยีอินฟราเรดเดิมนั่นเอง [26]



รูปที่ 2.19 หลักการส่งสัญญาณ Bluetooth

2.13 Arduino IDE

ซอฟต์แวร์ที่ใช้ในการพัฒนางานเขียนโปรแกรมและคอมไพล์ลงบอร์ด IDE ย่อมาจาก (Integrated Development Environment) คือส่วนเสริมของระบบการพัฒนาหรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทาได้ ทาให้การพัฒนางานต่าง ๆ เร็วมากขึ้น โดยขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึงผ่านฟังก์ชัน เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR [27]

2.14 งานวิจัยที่เกี่ยวข้อง

การศึกษาวิจัยที่เกี่ยวข้องกับการจัดการแบตเตอรี่โดยส่วนใหญ่จะมีการสร้างแบบจำลองทางคณิตศาสตร์เพื่อวิเคราะห์และประเมินประสิทธิภาพการทำงานของแบตเตอรี่ซึ่งมีการจำลองสถานการณ์ในการใช้งานแบตเตอรี่ที่สภาวะต่าง ๆ เช่น เวลาที่ใช้ในการชาร์จแบตเตอรี่ที่ต่างกัน ระยะเวลาการใช้งานของแบตเตอรี่รวมไปถึงความสามารถในการเก็บประจุของแบตเตอรี่ทั้งนี้ผู้วิจัยได้ค้นคว้าและรวบรวมบทความที่มี การศึกษาพฤติกรรมและออกแบบระบบจัดการแบตเตอรี่ โดยมีรายละเอียดดังนี้

Tarun Huria [28] และคณะ (2012) ศึกษาวิธีการที่มีประสิทธิภาพสำหรับการพัฒนาแบบจำลองเซลล์ลิเธียมที่มีการประมาณค่าพารามิเตอร์แบบแผนตัวเลขโดยใช้การทดสอบการปล่อยกระแสพัลส์พลังงาน สูงให้แก่ลิเซลล์เธียม (LiNiCoMnO2 cathode และ graphite-based anode ภายใต้สภาพการทำงานที่แตกต่างกัน แสดงสถานะของประจุ, กระแสเฉลี่ย และอุณหภูมิกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้มีประโยชน์สำหรับการสร้างแบบจำลองความเที่ยงตรงสูงสามารถทำนายกระแสไฟฟ้า/ประสิทธิภาพแรงดันไฟฟ้าและการประเมินสถานะ รันไทม์ของแบตเตอรี่ได้

Jaesik Lee และคณะ [29] (2016) ศึกษาการควบคุมการใช้งานของแบตเตอรี่ที่รวมกันเป็นขนาดใหญ่ โดยควบคุมผ่านระบบไร้สายโดยสามารถตรวจสอบกระแส แรงดันและอุณหภูมิของแบตเตอรี่เพื่อเพิ่มประสิทธิภาพการใช้งานของแบตเตอรี่

Taesic Kim และคณะ [30] (2017) ได้ศึกษาการตรวจสอบสภาพแบตเตอรี่รูปแบบใหม่บนคลาวด์และ แพลตฟอร์มการวินิจฉัยข้อผิดพลาดสำหรับข้อมูลแบตเตอรี่ชนิดลิเธียมไอออนขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

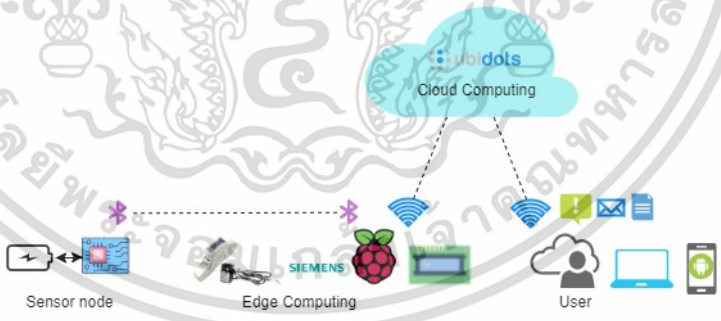
3.1 ภาพรวมขององค์ประกอบหลักของระบบ

ในการออกแบบการทำงานของระบบการจัดการแบตเตอรี่บนคลาวด์แพล็ตฟอร์มสำหรับการวัด วิเคราะห์ และแสดงประสิทธิภาพของแบตเตอรี่ดังรูปที่ 3.1 ประกอบไปด้วย

1) Sensor node จะเป็นส่วนที่ทำการวัดค่าทางไฟฟ้า โดยใช้ Microcontroller เป็น STM32 และใช้ INA219 ในการวัดค่าทางไฟฟ้า ใช้ Thermistor ในการวัดอุณหภูมิ ใช้ LTC4162 ในการควบคุมการชาร์จแบตเตอรี่ และส่งต่อข้อมูลผ่าน Bluetooth ไปยังส่วน Edge computing

2) Edge computing จะเป็นการนำข้อมูลจาก Sensor node และค่าทางไฟฟ้าจากแหล่งจ่ายพลังงานที่วัดได้จาก SDM120 Energy meter มาแสดงในรูปแบบของ HMI บน Raspberry Pi โดยใช้โปรแกรม Node-Red ในการออกแบบและสร้าง UI พร้อมกับเก็บข้อมูลใน SQLite ไว้แสดงข้อมูลกรณีที่ขาดการเชื่อมต่ออินเทอร์เน็ต และส่งข้อมูลไปยัง Cloud computing ผ่าน MQTT โปรโตคอล

3) Cloud computing จะเป็นการนำข้อมูลที่ได้จาก Edge computing มาเก็บไว้เพื่อสร้างเป็นฐานข้อมูลของแบตเตอรี่ แสดงข้อมูลการใช้งานของแบตเตอรี่และแหล่งจ่ายพลังงานในหน้า Dashboard, แจ้งเตือนเมื่อมีความผิดปกติเกิดขึ้นกับแบตเตอรี่และระบบได้แบบ Real time และสามารถส่งรายงานประจำสัปดาห์ได้อย่างอัตโนมัติ



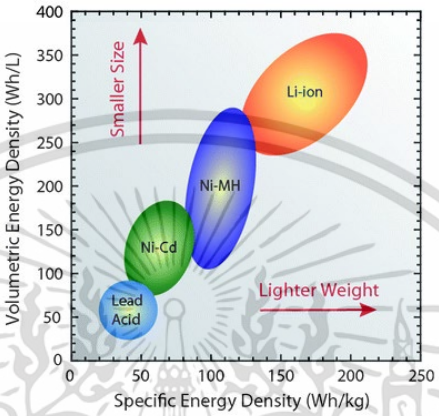
รูปที่ 3.1 ภาพรวมของการทำงานของระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบและการดำเนินงาน

การออกแบบและการดำเนินการประกอบด้วย 2 ส่วน คือการเลือกประเภทของแบตเตอรี่ที่ใช้ศึกษาและการออกแบบระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์มโดยมีรายละเอียดดังต่อไปนี้

3.2.1 การเลือกประเภทของแบตเตอรี่ที่ใช้ศึกษา



รูปที่ 3.2 ความสามารถในการเก็บพลังงานของแบตเตอรี่ประเภทต่าง ๆ
ที่มา : Shamsi (2016 : 21) Comparison of different battery technologies based on volumetric and gravimetric energy density.

จากรูปที่ 3.2 จะเห็นว่าแบตเตอรี่ลิเธียมไอออนมีความสามารถในการเก็บพลังงานมากกว่าแบตเตอรี่ประเภทอื่นทั้งในเชิงปริมาตร (Volumetric energy density) และเชิงมวล (Specific energy density) น้ำหนักเบา ขนาดเล็ก และมีระยะเวลาการใช้งานยาวนานต่อการประจุไฟฟ้าเต็มหนึ่งครั้ง ด้วยเหตุผลข้างต้นนี้จึงนำไปสู่การเลือกใช้แบตเตอรี่ลิเธียมไอออน

3.2.2 การออกแบบระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม

ประกอบด้วย 3 ส่วนคือการออกแบบอุปกรณ์สำหรับระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม, ออกแบบการใช้งาน Raspberry Pi สำหรับระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์มและออกแบบคุณลักษณะของ Application บน Cloud Platform โดยภาพรวมของระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์มสามารถแสดงได้ดังรูปที่ 3.3 และ 3.4

3.2.2.1 การออกแบบอุปกรณ์สำหรับระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม

จะใช้ STM32 เป็น Microcontroller ใช้ในการสื่อสารข้อมูล และช่วยบริหารจัดการชาร์จ ดิสชาร์จโดยการควบคุม Relay ต่อมาในส่วนของการวัดค่าทางไฟฟ้าจะใช้ INA219 ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดค่าทางไฟฟ้า ซึ่งประกอบด้วย แรงดันไฟฟ้า กระแสไฟฟ้า และใช้ Thermistor ในการวัดอุณหภูมิของแบตเตอรี่และอุณหภูมิสิ่งแวดล้อม ในส่วนการชาร์จแบตเตอรี่ จะใช้ LTC4162 ในการควบคุมแรงดันคงที่, กระแสคงที่ และอุณหภูมิของการชาร์จแบตเตอรี่ เพื่อไม่ให้แบตเตอรี่เกิดการเสียหาย และหลังจากนั้นนำค่าทางไฟฟ้าที่ได้มาแสดงผ่านทาง I2C OLED และทำการส่งต่อค่าทางไฟฟ้าต่อไปยังส่วน Edge computing ผ่านทาง Bluetooth

3.2.2.2 ออกแบบการใช้งาน Raspberry Pi สำหรับระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม

Raspberry Pi จะใช้โปรแกรม Node-Red ในการสร้าง UI เพื่อให้ผู้ใช้สามารถเข้าถึงข้อมูลการใช้งานของแบตเตอรี่ โดยจะแสดงค่าทางไฟฟ้าของแบตเตอรี่ที่รับมาจาก Bluetooth ของ STM32, แสดงค่าทาไฟฟ้าของแหล่งจ่ายพลังงานที่วัดได้จาก SDM120CT Energy meter ที่เชื่อมต่ออยู่กับ Siemens SIMATIC IOT2040, แสดง Datasheet ของแบตเตอรี่ลิเธียมไอออนที่ใช้ และมีการแจ้งเตือนกรณีขาดการเชื่อมต่อ Bluetooth โดยข้อมูลจะถูกเก็บไว้ที่ SQLite database ภายใน SD card ของ Raspberry Pi และจะส่งข้อมูลไปยัง Cloud computing ด้วย MQTT โปรโตคอลทุก 15 นาที

3.2.2.3 ออกแบบคุณลักษณะของ Application บน Cloud platform

ผู้จัดทำได้ทำการออกแบบให้ระบบมีความคุณสมบัติและรายละเอียดดังต่อไปนี้

1) รับและแสดงผลค่าจาก Edge computing ได้แบบ Real time ผู้จัดทำได้ออกแบบให้ระบบมีส่วนประกอบ 2 ส่วนหลักคือ ส่วน Supply Meter ซึ่งเป็นส่วนแสดงผลข้อมูลการใช้แหล่งจ่ายพลังงานและส่วน Battery โดยระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์มที่ถูกออกแบบในโครงงานนี้จะประกอบด้วย 1 Stringและใน 1 String ประกอบด้วย 2 Cells ผู้จัดทำจึงได้ออกแบบให้ระบบประกอบไปด้วย 3 อุปกรณ์คือ Battery String 01, Battery Cell 0101และ Battery Cell 0102 แต่ละอุปกรณ์ประกอบด้วยตัวแปรซึ่งประกอบด้วยรายละเอียดดังตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงรายการอุปกรณ์และตัวแปรที่ต้องการสร้างขึ้น

Devices		Variable				
No.	Device name	No.	Name	Allowed range		Unit
1	Supply Meter	1	Current	0	2	A
		2	Power	0	28	W
		3	Voltage	0	24	V
2	Battery String 01	1	Current	0	2000	mA
		2	Voltage	0	8.4	V
		3	Temperature Battery	20	100	degC
		4	Temperature Environment	20	100	degC
		5	Status	0	1	-
		6	State of Health	0	100	%
		7	State of Charge	0	100	%
3	Batter Cell 0101	1	Current	0	2000	mA
		2	Voltage	0	4.2	V
		3	State of Health	0	100	%
		4	State of Charge	0	100	%
4	Batter Cell 0102	1	Current	0	2000	mA
		2	Voltage	0	4.2	V
		3	State of Health	0	100	%
		4	State of Charge	0	100	%

2) ทำการแจ้งเตือนอัตโนมัติเมื่อมีความผิดปกติเกิดขึ้นกับแบตเตอรี่และระบบในกรณีภายใต้เงื่อนไขดังต่อไปนี้

a) State of Health ≤50%

b) อุณหภูมิขณะชาร์จ > 45 degC หรือ อุณหภูมิขณะดิสชาร์จ > 60 degC

c) อุปกรณ์ขาดการเชื่อมต่อนานกว่า 15 นาที

3) เก็บข้อมูลของแบตเตอรี่ย้อนหลังได้ไม่น้อยกว่า 2 สัปดาห์

4) นำข้อมูลมาคำนวน State of Charge ได้โดยมีสมการในการคำนวน State of Charge (%) ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

a) สมการในการคำนวน State of Charge ของ String: ((*Voltage* -7.2 *100/ (8.4 -7.2

b) สมการในการคำนวน State of Charge ของ Cell: ((*Voltage* -3.6 *100/ (4.2 -3.6

5) แสดงหน้า Dashboard ได้โดยมีรายละเอียดดังต่อไปนี้

ตารางที่ 3.2 ข้อมูลที่ต้องการแสดงในหน้า Dashboard

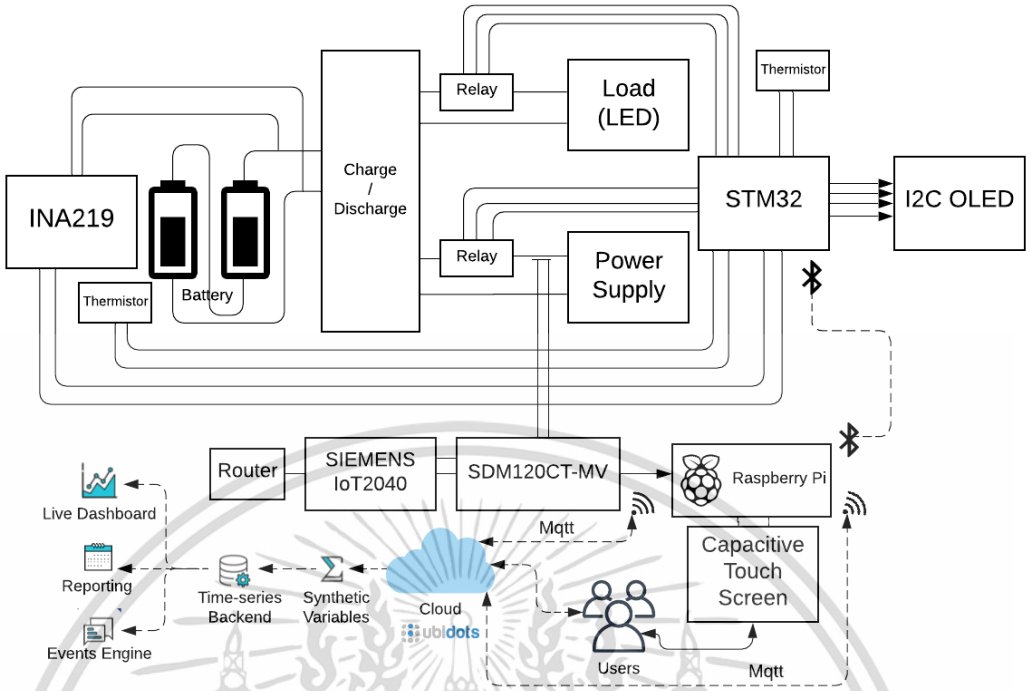
ลำดับที่	รายละเอียดที่ต้องการแสดง	คำอธิบายเพิ่มเติม
1	Instrumentation Engineering | Kmitl, BMS on Cloud	แสดงชื่อโปรเจคและภาควิชา
2	Temperature Battery	แสดงอุณหภูมิของแบตเตอรี่
3	Temperature Environment	แสดงอุณภูมิของสิ่งแวดล้อม
4	State of Charge : Battery 01	แสดงสถานะการชาร์จของแบตเตอรี่ 01 (String01, Cell0101, Cell0102)
5	Voltage : Battery 01	แสดงแรงดันที่เวลาใด ๆ ของแบตเตอรี่ 01 (String01, Cell0101, Cell0102)
6	Current : Battery 01	แสดงกระแสที่เวลาใด ๆ ของแบตเตอรี่ 01 (String01, Cell0101, Cell0102)
7	State of Health : Battery String 01	แสดงสุขภาพของแบตเตอรี่ String01
8	State of Health : Battery Cell 0101	แสดงสุขภาพของแบตเตอรี่ Cell0101
9	State of Health : Battery Cell 0102	แสดงสุขภาพของแบตเตอรี่ Cell0102
10	System Energy Consumption	แสดงการใช้พลังงานของระบบที่ Supply Meter ซึ่งประกอบด้วยค่ากระแส, แรงดันและกำลัง

6) สร้างรายงานประจำสัปดาห์อัตโนมัติ

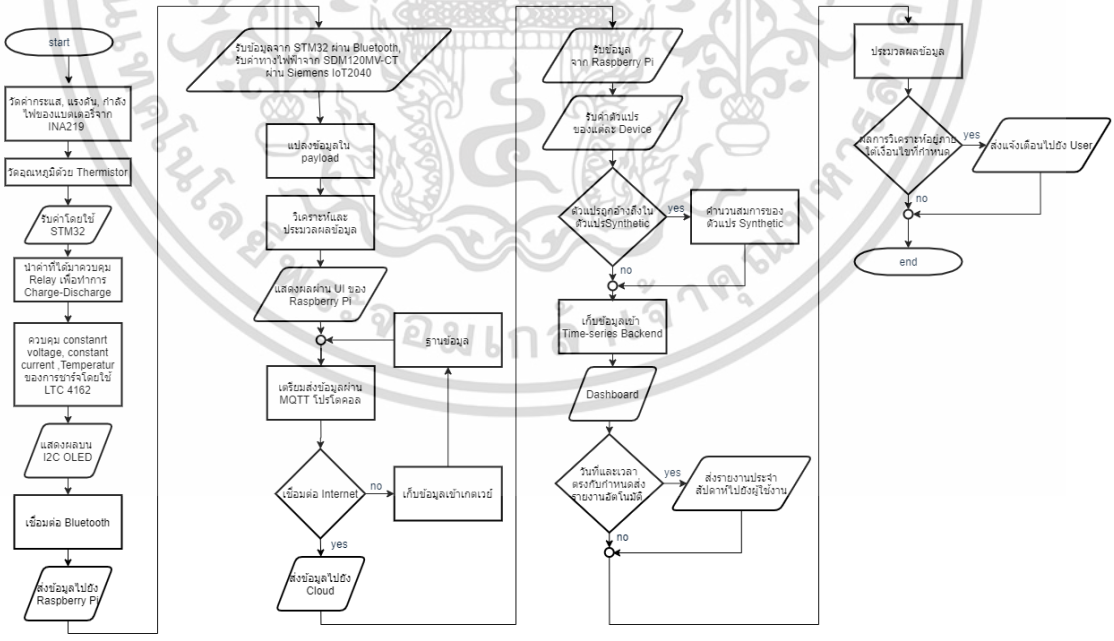
7) เพิ่มและกำหนดสิทธิ์การเข้าถึงของผู้ใช้งาน

8) ใช้ได้บน Personal computer และ Smartphone ระบบปฏิบัติการ Android

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ภาพรวมการทำงานของอุปกรณ์ของระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์มที่ทำการออกแบบ



รูปที่ 3.4 แผนภาพกิจกรรมของการทำงานของระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม

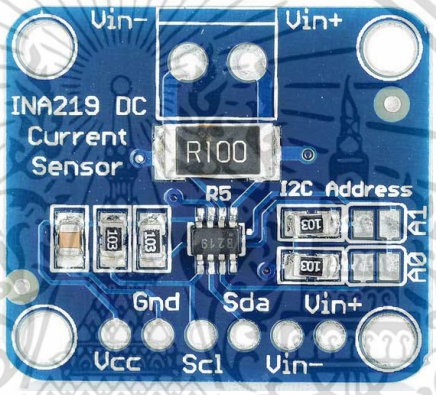
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 องค์ประกอบฮาร์ดแวร์

องค์ประกอบฮาร์ดแวร์ของระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์มสามารถอธิบายภาพรวมได้ตาม รูปที่ 3.3 โดยอุปกรณ์ฮาร์ดแวร์แต่ละชิ้นสามารถอธิบายคุณลักษณะได้ดังต่อไปนี้

3.3.1 อุปกรณ์วัดแรงดันไฟฟ้า และ อุณหภูมิ

INA219 เป็นโมดูลดิจิตอลทีมีเซ็นเซอร์ในการวัดกระแส แรงดัน กำลังไฟฟ้ากระแสตรง เชื่อมต่อกับไมโครคอนโทรลเลอร์ด้วย I2C ใช้ไฟเลี้ยง 3-5 Vdc แรงดันไฟฟ้าที่สามารถตรวจจับได้ที่ 0-26 Vdc และกระแสไฟฟ้าสูงสุดที่สามารถตรวจจับได้ที่ 3.2A และยังมี Calibration Registers ในตัว (รูปที่ 3.5



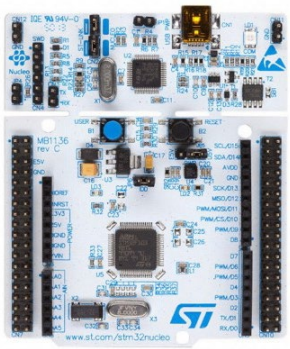
รูปที่ 3.5 INA219

ที่มา : https://www.dx.com/p/ina219-diy-dc-current-sensor-breakout-module-blue-2053458.html

3.3.2 STM32 Nucleo F091RC

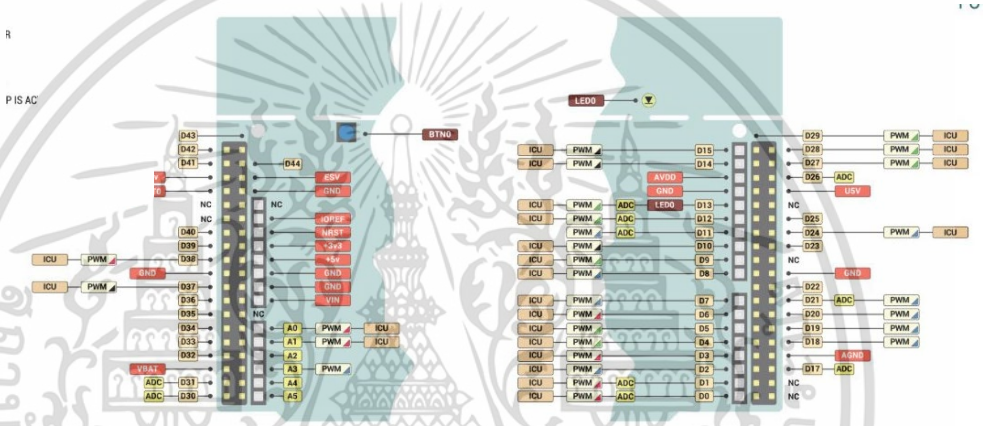
ไมโครคอนโทรลเลอร์ (MCU: Microcontroller Unit) ขนาด 32 บิต ตระกูล STM32 ARM Cortex ของบริษัท STMicroelectronics เป็นชิปไมโครคอนโทรลเลอร์ที่ประสิทธิภาพสูง และมีการใช้งานแพร่หลาย ทั้งเพื่อการศึกษา งานอดิเรก และไปจนถึงการใช้งานอุตสาหกรรม ดังนั้น นักประดิษฐ์หรือนักออกแบบ นักพัฒนาระบบสมองกลฝังตัว ก็สามารถใช้งานชิปตระกูลนี้ ทางบริษัทได้พัฒนาบอร์ดทดลองสำหรับการเรียนรู้มาหลายบอร์ด และที่เป็นที่รู้จักกันคือบอร์ด STM32 Nucleo F901RC นอกจากนั้นยังสามารถใช้ได้กับระบบปฏิบัติการ Mbed OS ที่เหมาะสำหรับงานด้าน IoT และมี API + Libraries ต่าง ๆ ในภาษา C++ ให้ใช้ได้ด้วย (รูปที่ 3.6 – 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 STM32 Nucleo F091RC

ที่มา : https://www.deaware.com/wp-content/uploads/2014/03/NUCLEO.png



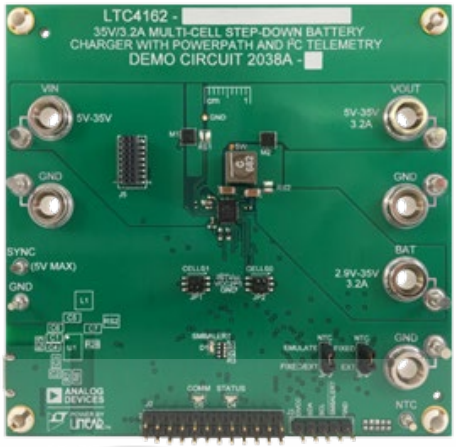
รูปที่ 3.7 Port การใช้งาน STM32 Nucleo F091RC

ที่มา : https://www.deaware.com/wp-content/uploads/2014/03/NUCLEO.png

3.3.3 Demo circuit 2038a

Demo Circuit 2038a ใช้ LTC4162 multicell step-down battery charger ใช้การเชื่อมต่อแบบ I2C และ LTC4162 ยังรองรับเซลล์ได้มากถึง 8 เซลล์ สำหรับ Li-Ion, 9 เซลล์สำหรับ LifePO4 และ 6V,12V หรือ 24V ของแบตเตอรี่ตะกั่วกรด DC2038a สามารถปรับเลือกจำนวนเซลล์ที่ใช้ได้โดยการย้ายจัมเปอร์เพื่อกำหนดจำนวนของเซลล์ (รูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

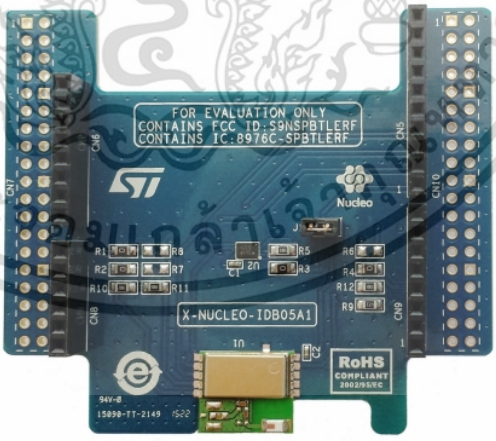


รูปที่ 3.8 Demo circuit 2038a

ที่มา : https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/dc2038a.html

3.3.4 X-nucleo-idb05a1

X-NUCLEO-IDB05A1 ที่ใช้ SPBTLE-RF เป็นบลูทูธโมดูล สามารถใช้งานเข้ากันได้กับ ST Morpho และ Arduino UNO R3 (ผู้ใช้สามารถเชื่อมต่อตัวเชื่อมต่อ ST Morpho ได้ถ้าต้องการ X-NUCLEO-IDB05A1 อินเตอร์เฟสกับไมโครคอนโทรลเลอร์ STM32 ผ่าน SPI พินและผู้ใช้สามารถเปลี่ยน Default SPI Clock, SPI chip select และ SPI IRQ โดยการเปลี่ยน resistor บน the evaluation board (รูปที่ 3.9)



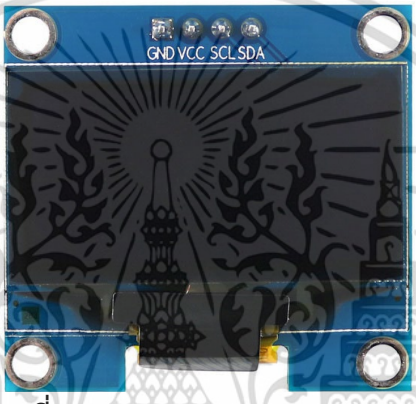
รูปที่ 3.9 X-nucleo-idb05a1

ที่มา : https://th.element14.com/productimages/standard/en_GB/2664525-40.jpg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 I2C OLED

OLED (Organic Light-Emitting Diode) display เป็นจอแสดงผลกราฟิกประเภทหนึ่งที่สร้างจากวัสดุ "สารกึ่งตัวนำอินทรีย์" (Organic semiconductor) มีลักษณะเป็นชั้นบาง ๆ อยู่ระหว่างขั้วบวก (Anode) และขั้วลบ (Cathode) และสามารถเปล่งแสงได้เมื่อมีกระแสไฟฟ้าไหลผ่าน เรียกกระบวนการนี้ว่า Electroluminescence จอภาพ OLED มีข้อดีซึ่งแตกต่างจากจอแสดงผล LCD (Liquid Crystal Display) ทั่วไปคือ ไม่ต้องมีวงจรที่สร้างแสง Backlight ดังนั้นจึงมีความหนาน้อยกว่าและเบากว่า ใช้กำลังไฟฟ้าต่ำ นอกจากนั้นจะไม่มีการเปล่งแสงในบริเวณที่ต้องการให้เป็นสีดำ (รูปที่ 3.10



รูปที่ 3.10 I2C OLED ขนาด 128*64

ที่มา : https://img.dxcdn.com/productimages/sku_432398_1.jpg

3.3.6 Relay

บอร์ดรีเลย์ 5VDC แบบ 1 ช่อง สามารถทำงานแยกอิสระแบบ Optocoupler Isolation และยังสามารถปรับเปลี่ยนการทำงานได้ทั้งแบบ Active High หรือ Active Low (รูปที่ 3.11)

รูปที่ 3.11 Relay 5VDC

ที่มา : https://www.aliexpress.com/i/4000789132248.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.7 Thermistor

Thermistor คือ อุปกรณ์ที่ทำหน้าที่ตรวจจับอุณหภูมิ ทำมาจากโลหะออกไซด์ ที่มีความไวต่อการเปลี่ยนแปลงของอุณหภูมิ มี 2 ประเภท คือ

1) แบบ PTC (Positive Temperature Coefficients) คือ เมื่ออุณหภูมิเพิ่มขึ้น จะมีค่าความต้านทานเพิ่มขึ้น มักจะสร้างจาก แมงกานีส นิกเกิล โคบอลต์ เหล็ก และ ไททาเนียม

2) แบบ NTC (Negative Temperature Coefficients) คือ เมื่ออุณหภูมิลดลง จะมีค่าความต้านทานลดลง ใช้ตรวจสอบความร้อนเพื่อควบคุมระดับการทำงาน มักจะสร้างจาก แบเรี่ยม และ ตะกั่ว (รูปที่ 3.12)

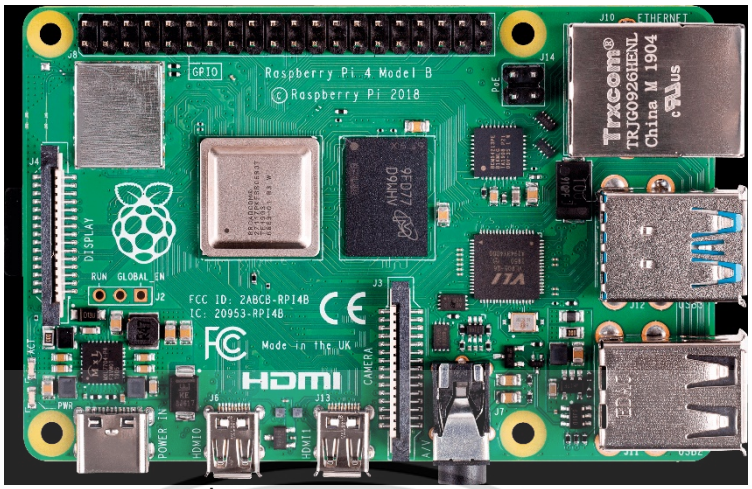
รูปที่ 3.12 Thermistor 100K

ที่มา: https://www.rapidonline.com/tks-ttc05104ksy-ntc-thermistor-100k-10-3-5mm-pitch-61-0420

3.3.8 Raspberry Pi

Raspberry Pi 4 Model B เป็น Single Board ใช้ชิพ Broadcom BCM2711 Quad-Core ARM Cortex-A72 ความเร็ว 1.5 GHz มีหน่วยความจำ LPDDR4-2400 ขนาด 4 GB พร้อมชิพ Wireless LAN แบบ Dual-Band รองรับ 2.4 GH และ 5 GHz พร้อมรองรับ Bluetooth 5.0 BLE มีพอร์ต LAN รองรับ Gigabit Ethernet พอร์ต USB 3.0 Host Type A จำนวน 2 พอร์ต และ USB 2.0 Host Type A จำนวน 2 พอร์ต มีพอร์ต micro-HDMI จำนวน 2 พอร์ต รองรับการเชื่อมต่อจอแสดงผล 4K60P ตามรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 Raspberry Pi 4 Model B

ที่มา : https://th.rs-online.com/web/p/processor-microcontroller-development-kits/1822096/

3.3.9 Siemens Simatic IOT 2040

SIMATIC IoT2040 (รูปที่ 3.14) เป็น Open industrial IoT gateway ที่ซัพพอร์ต IIoT (Industrial internet of things) โดยขยายจาก Protype พัฒนาด้วย Opensource จากบอร์ดอย่าง Arduino, Intel Galileo หรือ Raspberry Pi ถูกออกแบบมาเพื่อ IoT application เช่น เก็บข้อมูลจาก PLC หรือเซนเซอร์ โดยต่อฮาร์ดแวร์เพิ่มเติมเป็น Arduino shield ได้ IoT2040 เป็นฮาร์ดแวร์ที่เหมาะกับงานอุตสาหกรรม IoT2040 ถูกพัฒนาขึ้นมาโดยมี Housing ที่รองรับ IP20 พอร์ตการเชื่อมต่อมาตรฐานทางอุตสาหกรรม และใช้กับรางแบบ DIN rail ได้ สามารถสท่อสารกับอุปกรณ์ตัวอื่นในอุตสาหกรรมผ่าน Industrial protocol เช่น MODBUS, PROFINET และใช้โปรโตคอลด้านอินเตอร์เน็ตอย่าง MQTT ได้

รูปที่ 3.14 Siemens Simatic IOT 2040

ที่มา : https://www.hackster.io/74113/connect-your-siemens-simatic-iot2000-to-ubidots-nodered-0fcae6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.10 Capacitive Touch Screen

คุณสมบัติทั่วไปของหน้าจอสัมผัส

1) ขนาดหน้าจอ : 194mm x 110mm x 20mm (including standoffs)

2) ขนาดหน้าจอที่ดูได้ : 155mm x 86mm

3) ความละเอียดหน้าจอ 800 x 480 pixels

4) สามารถสัมผัสได้ 10 นิ้ว

5) เชื่อมต่อกับบอร์ด Raspberry Pi โดยใช้สายริบบินเคเบิลที่เชื่อมต่อกับพอร์ต DSI

6) บอร์ดอะแดปเตอร์ใช้เพื่อเปิดการแสดงผลและแปลงสัญญาณขนานจากจอแสดงผลไปยังพอร์ต อนุกรมของ Raspberry Pi

7) PiWill ต้องการให้ระบบปฏิบัติการ Raspbian รุ่นล่าสุดทำงานได้อย่างถูกต้อง

8) ทำงานที่แรงดันไฟฟ้า : 5V

รูปที่ 3.15 หน้าจอสัมผัส

ที่มา : https://thepihut.com/products/official-raspberry-pi-7-touchscreen-display

3.4 ซอฟต์แวร์ที่ใช้พัฒนาระบบ

1) Arduino IDE ซอฟต์แวร์ที่ใช้ในการพัฒนางาน เขียนโปรแกรมและคอมไพล์ลงบอร์ด IDE ย่อมาจาก (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนาหรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น โดยขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึงผ่านฟังก์ชัน เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR หรือเวอร์ชันอื่น ๆ ของ Arduino และสามารถปรับใช้งานร่วมกับบอร์ด STM32 ได้อีกด้วย

2) Node-Red เป็น Medium-ware โดยมีพื้นฐานการใช้งานในรูปแบบ Flow-based programming ที่พัฒนาขึ้นมาเพื่อการใช้งานกับ Internet of Things สำหรับการใช้งาน Node-Red นั้นมีความสะดวกมากเพราะสามารถแก้ไข Flow โปรแกรมต่าง ๆ ผ่าน Browser ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเนื่องจาก Node-Red ถูกพัฒนาบน Javascript จึงทำให้สามารถพัฒนาขีดความสามารถออกไปได้

3) Ubidots เป็นแพลตฟอร์ม IoT และ cloud applications แบบ Platform as a Service ที่ช่วยให้ผู้ใช้งานสามารถสร้างแอปพลิเคชันขึ้นเองได้ สามารถเชื่อมต่อเข้ากับฮาร์ดแวร์และอินพุตดิจิตอลได้ผ่านโปรโตคอล HTTP / MQTT / TCP / UDP เก็บข้อมูลแอปพลิเคชันในคลาวด์ สามารถสร้างหน้าแสดงผลได้ตามที่แอปพลิเคชันถูกกำหนดให้แสดงตามการปรับแต่งการวิเคราะห์ของผู้ใช้งาน สามารถตั้งค่าให้แอพพลิเคชั่นทำงานได้อย่างอัตโนมัติ นอกจากนี้ยังสามารถส่งการแจ้งเตือน (Events) ออกไปยังผู้ใช้งานที่กำหนดไว้ สามารถจัดทำและจัดส่งรายงานอัตโนมัติตามที่ถูกกำหนดโดยผู้ใช้งาน สามารถติดตามการแสดงผลของข้อมูลบน Personal computer และ Smartphone ระบบปฏิบัติการ Android ทำให้ Ubidots สามารถเชื่อมต่อกับอุปกรณ์และใช้บริการต่าง ๆ ได้อย่างตรงความต้องการของผู้ใช้งาน

3.5 การประมวลผลข้อมูลด้วยไมโครคอนโทรลเลอร์

ผลที่ได้จากการวัดด้วยเซนเซอร์ INA219 จะถูกนำมาประมวลผลด้วย STM32 ซึ่งกระบวนการทำงานภายในไมโครคอนโทรลเลอร์จะเริ่มจากการอ่านค่าพารามิเตอร์ทางไฟฟ้าผ่านพอร์ทต่าง ๆ การกำหนด Baud Rate ในการรับและส่งข้อมูล และทำการกำหนดตัวแปรให้ตรงกับพอร์ทที่อ่านค่าของอุปกรณ์แต่ละตัว อ่านค่ากระแสไฟฟ้า แรงดันไฟฟ้าและอุณหภูมิจากเซนเซอร์แล้วนำมาแสดงค่า จากนั้นส่งค่าออกมาเป็นโปรโตคอล กระแสไฟฟ้า, แรงดันไฟฟ้าและอุณหภูมิ ที่วัดได้จะถูกส่งออกมาพร้อมกันเป็นโปรโตคอลอย่างต่อเนื่อง เพื่อเก็บข้อมูลซึ่งสามารถใช้คำสั่งเคลียร์ค่า หรือหยุดการส่งข้อมูลได้ [31] - [33]

โดยการเขียนโปรแกรมลง STM32 โดยใช้ Arduino IDE สามารถทำได้ดังต่อไปนี้

1) เข้า Google แล้วค้นหาโดยใช้คีย์เวิร์ด Arduino IDE download หรือเข้าไปที่ลิงก์ https://www.arduino.cc/en/Main/Software เลื่อนลงด้านล่างเล็กน้อยจะปรากฏจอภาพดังรูปที่ 3.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

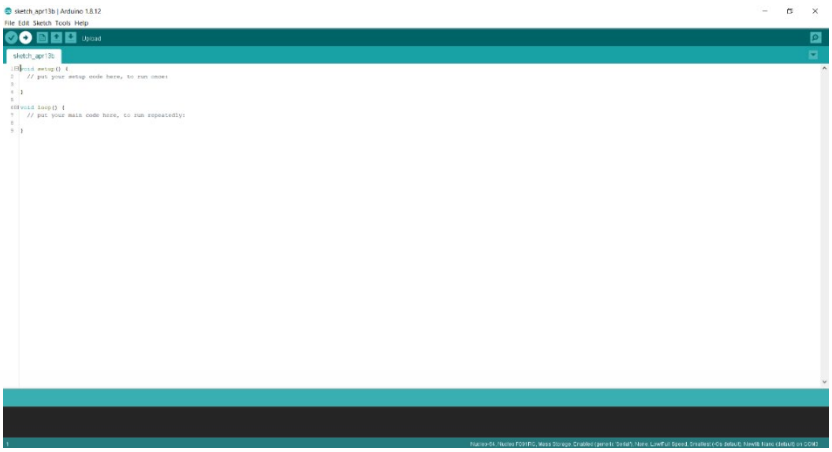
รูปที่ 3.16 แสดงหน้าเว็บในการโหลดโปรแกรม

2) เมื่อปรากฏจอภาพข้างล่างนี้เลือกปุ่ม Just Download เพื่อเริ่มต้นดาวน์โหลดทันที หรือเลือกปุ่ม Contribute & Download เพื่อบริจาคเงินสนับสนุนทางเว็บไซต์ แล้วจึงดาวน์โหลดโปรแกรม Arduino IDE ซึ่งเป็นโอเพ่นซอร์สซอฟท์แวร์ ในที่นี้เลือกปุ่ม Just Download (รูปที่ 3.17)

รูปที่ 3.17 Dowload Arduino IDE

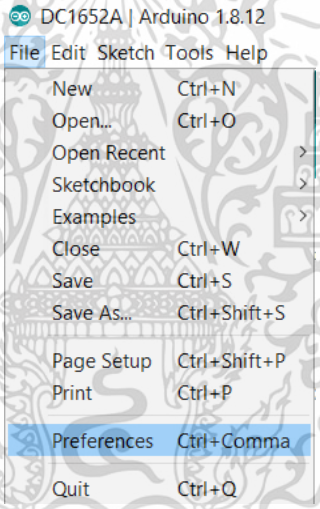
3) ค้นหาและดับเบิ้ลคลิกบนไฟล์ arduino.exe เรียกเข้าสู่โปรแกรม จะปรากฏจอภาพดังรูป 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 หน้าต่าง Arduino IDE

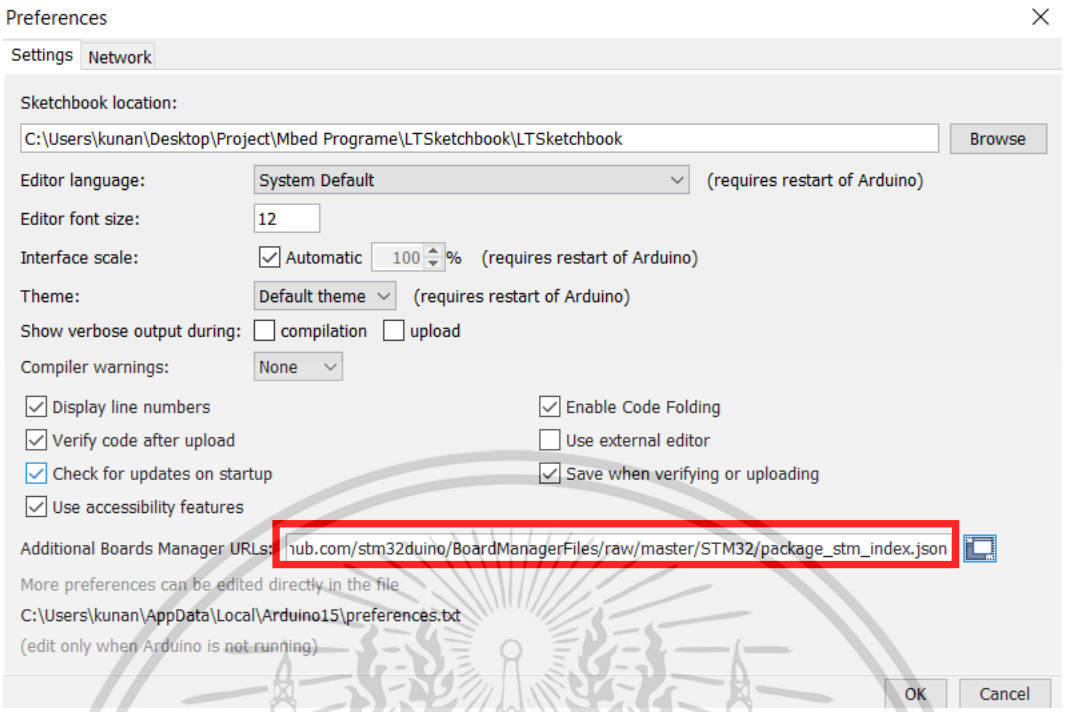
4) ตั้งค่าเริ่มต้นเพื่อดาวน์โหลดไลบรารี่โดยทำการเปิดโปรแกรม Arduino IDE แล้วเข้าไปที่ File > Preferences



รูปที่ 3.19 Select Preference Arduino IDE

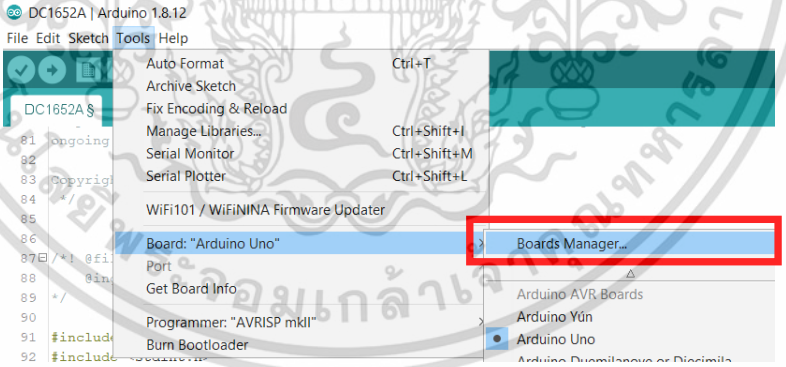
5) เมื่อปรากฏ Preferences ให้คัดลอกลิงก์นี้ https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json ไป Paste (Ctrl+V) ในช่อง Additional Boards Manager URLs ดังรูป 3.20 แล้วคลิกปุ่ม OK เพื่อออกจากไดอะล็อค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 Preference Arduino IDE

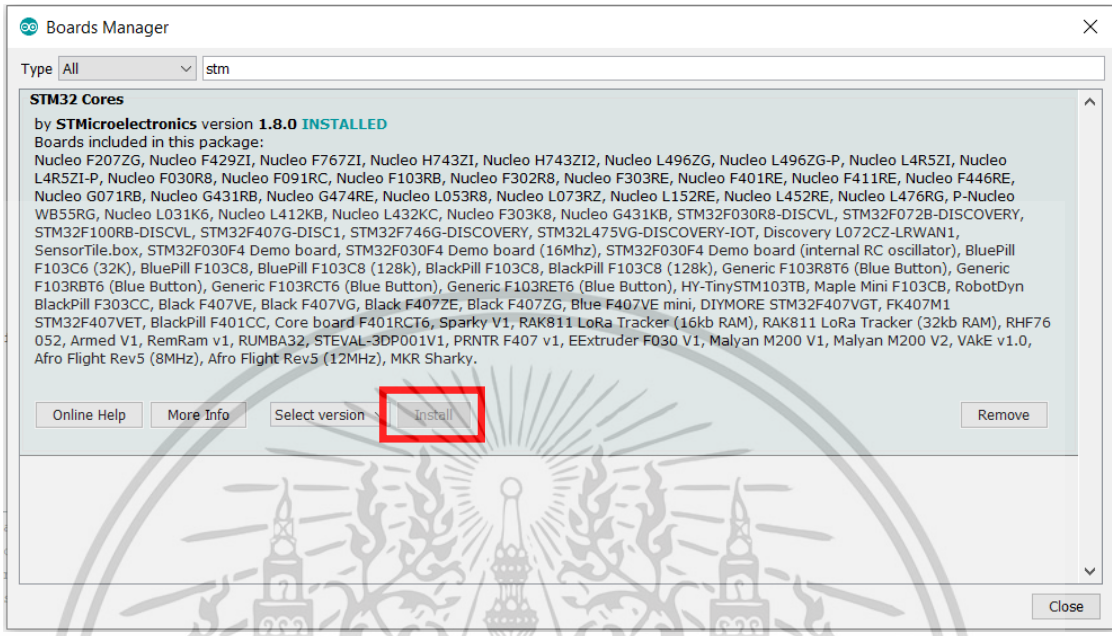
6) เริ่มดาวน์โหลด บอร์ดและไลบรารี่ โดยใช้คาสั่ง Tools > Board > Board Manager ดังรูป 3.21



รูปที่ 3.21 Board Manager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) ทำการ Search เพื่อค้นหา STM32 Cores แล้วคลิกที่ปุ่ม Install ดังรูปที่ 3.22



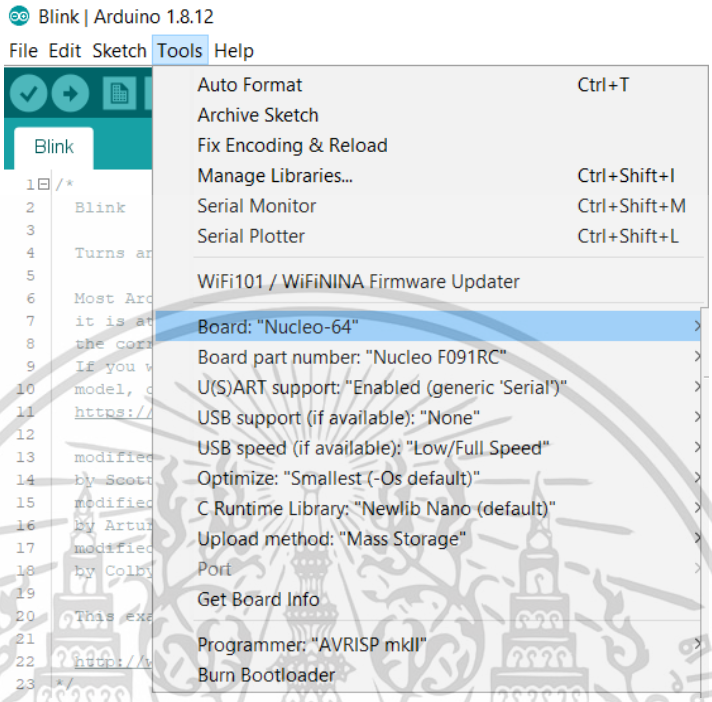
รูปที่ 3.22 Install Board STM32

8) ทำการเขียน Code เพื่อทดสอบ (รูปที่ 3.23

รูปที่ 3.23 แสดงการเขียนโปรแกรมลงใน Arduino IDE

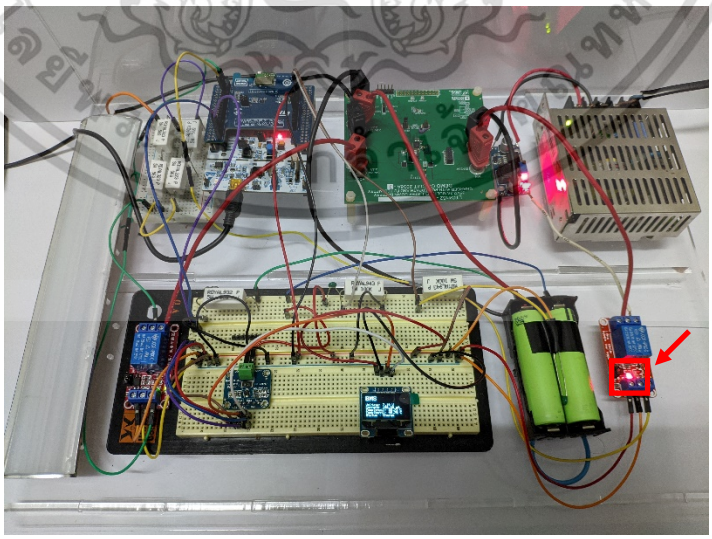
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) เข้าไปที่ Tools เพื่อทำการตั้งค่าการ Upload เป็น Board ที่ต้องการ (รูปที่ 3.24



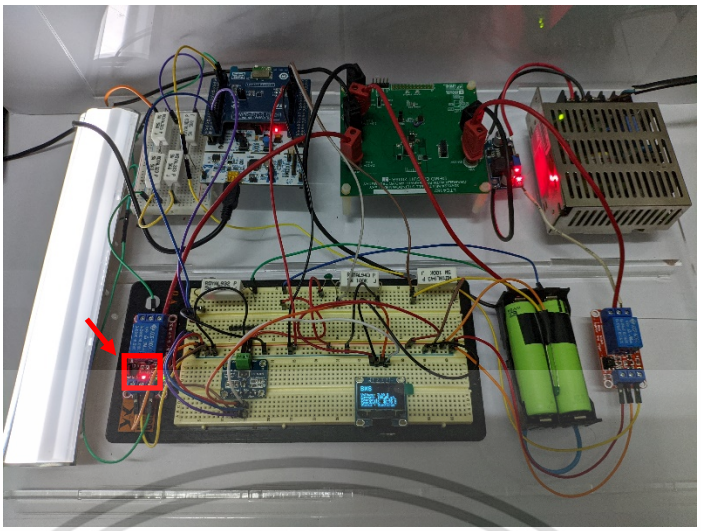
รูปที่ 3.24 เลือก Board ที่ต้องการ Upload ลง

10) จะเห็นได้ว่าระบบมีการทำงานตามคำสั่งที่กำหนด โดยระบบทำการ Charge จะแสดงดังรูปที่ 3.25 และระบบทำการ Discharge จะแสดงดังรูปที่ 3.26



รูปที่ 3.25 ระบบทำการ Charge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.26 ระบบทำการ Discharge

3.6 การใช้งาน Node-Red บน Raspberry Pi

ก่อนอื่นต้องทำการติดตั้งระบบปฏิบัติการให้กับ Raspberry Pi ก่อน โดยระบบปฏิบัติการที่ใช้นี้เป็น Raspbian ระบบปฏิบัติการสำหรับติดตั้งใช้งานบน Raspberry Pi พัฒนามาจากระบบ Debian Linux เหมาะสำหรับนำมาใช้ทำการทดลอง และงานวิจัยเกี่ยวกับระบบคอมพิวเตอร์แบบฝังตัว (Embedded System) โดยจะมี NOOBS เป็นเครื่องมือในการช่วยติดตั้งระบบปฏิบัติการให้ Raspberry Pi ทำให้สามารถติดตั้งได้ง่าย [34]

3.6.1 ขั้นตอนการติดตั้งระบบปฏิบัติการบน Raspberry Pi

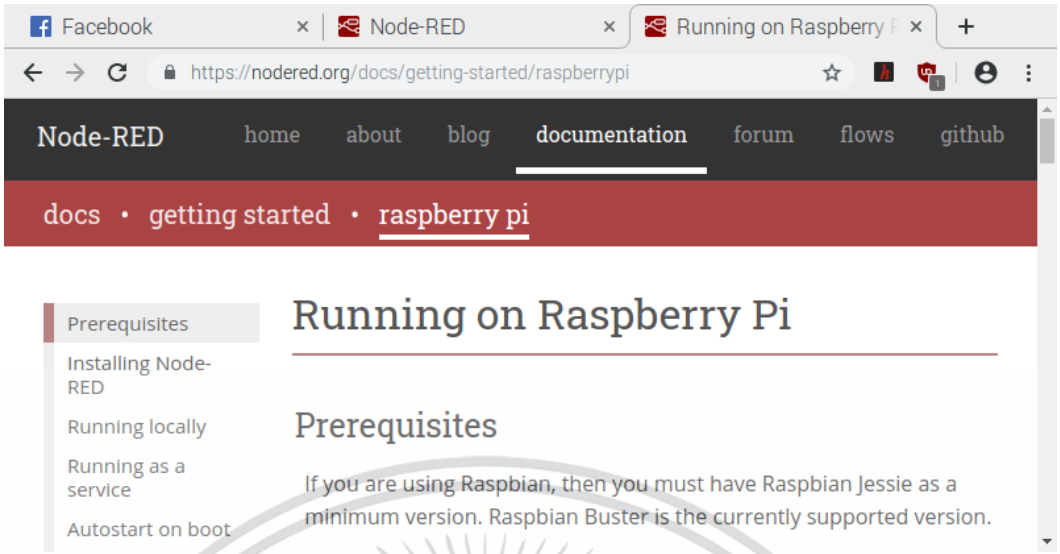
1) ดาวน์โหลด NOOBS ที่ https://www.raspberrypi.org/downloads/noobs/
2) Format Micro SD Card โดยมี File System เป็น FAT32
3) แตก Zip ไฟล์ไปที่ Micro SD Card ที่จะใช้ลง OS
4) นำ Micro SD Card มาใส่ Raspberry Pi 4 Model B
5) เปิดเครื่อง Raspberry Pi
6) จะพบหน้าจอการติดตั้ง ให้เลือกระบบปฏิบัติการ Raspbian แล้วกด Install
7) รอจนเสร็จ โดยระบบจะมี Default คือ

   Username : pi

   Password : raspberry

ระบบปฏิบัติการ Raspbian เวอร์ชันล่าสุดจะมีโปรแกรม Node-Red ติดตั้งมาให้อยู่แล้ว แต่ถ้าหากต้องการติดตั้งเวอร์ชันอื่นสามารถดูวิธีติดตั้งได้จาก https://nodered.org/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



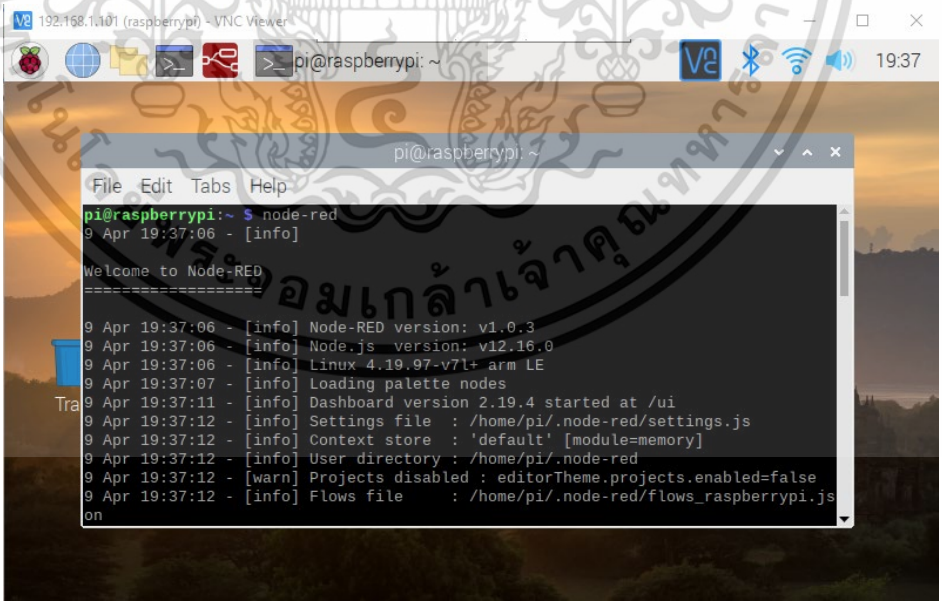
รูปที่ 3.27 เว็บเพจสำหรับการติดตั้งโปรแกรม Node-Red บน Raspberry Pi

ที่มา : https://nodered.org/

3.6.2 การใช้งานโปรแกรม Node-Red

เมื่อทำการติดตั้งโปรแกรม Node-Red เรียบร้อยแล้วหลังจากนั้นเปิดโปรแกรมเพื่อเริ่มใช้งานโดยทำตามขั้นตอนดังต่อไปนี้

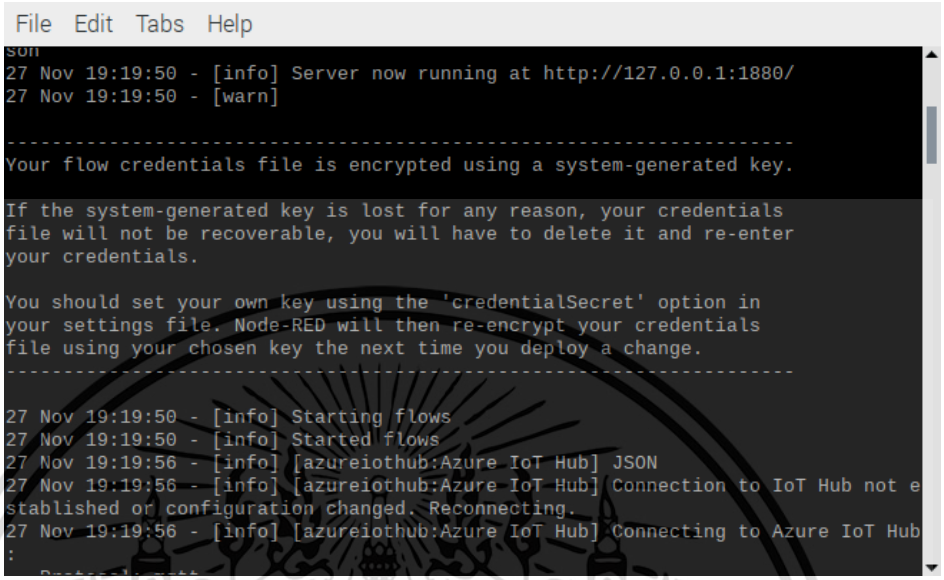
1) เปิด Terminal ของ Raspberry Pi และพิมพ์ Node-Red ตามรูปที่ 3.28



รูปที่ 3.28 แสดง Terminal ของ Raspberry Pi

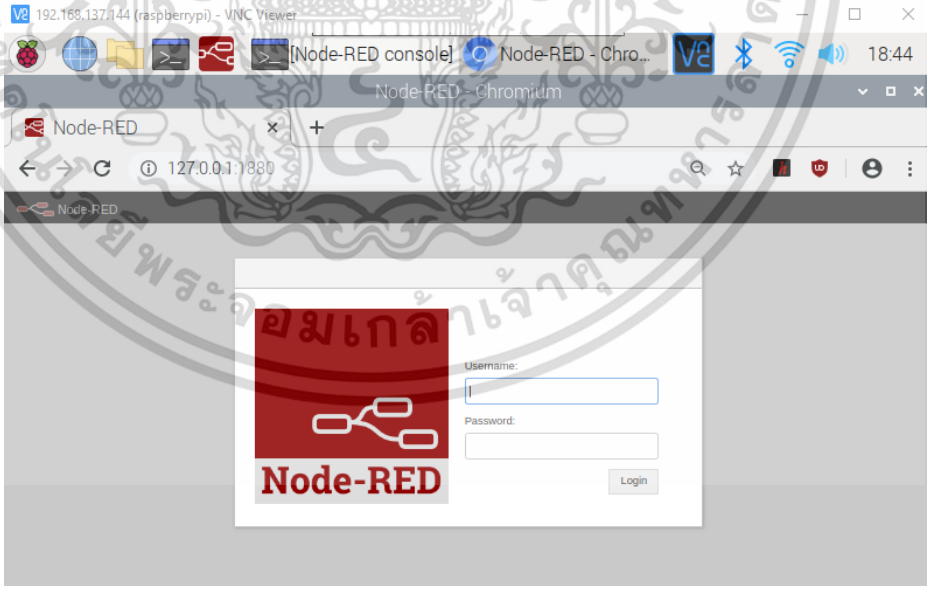
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ไปที่ Web Browser แล้วใส่ IP Address ของ Raspberry Pi โดยใช้ Port 1880 ดังรูปที่ 3.29



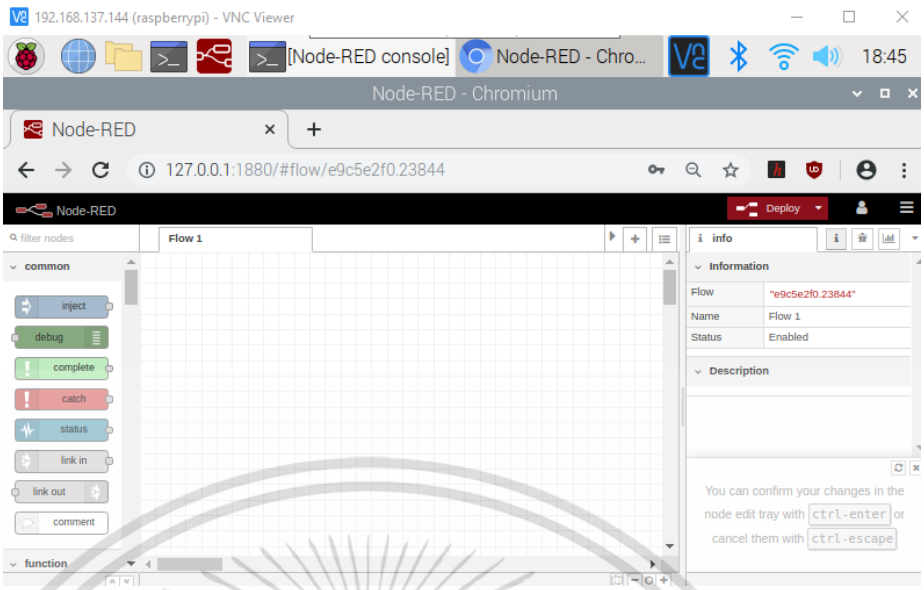
รูปที่ 3.29 แสดงการทำงานของโปรแกรม Node-Red บน Server

3) จะปรากฏโปรแกรม Node-Red ดังรูปที่ 3.30 และ 3.31



รูปที่ 3.30 แสดงการ Log in ก่อนใช้งานโปรแกรม Node-Red

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

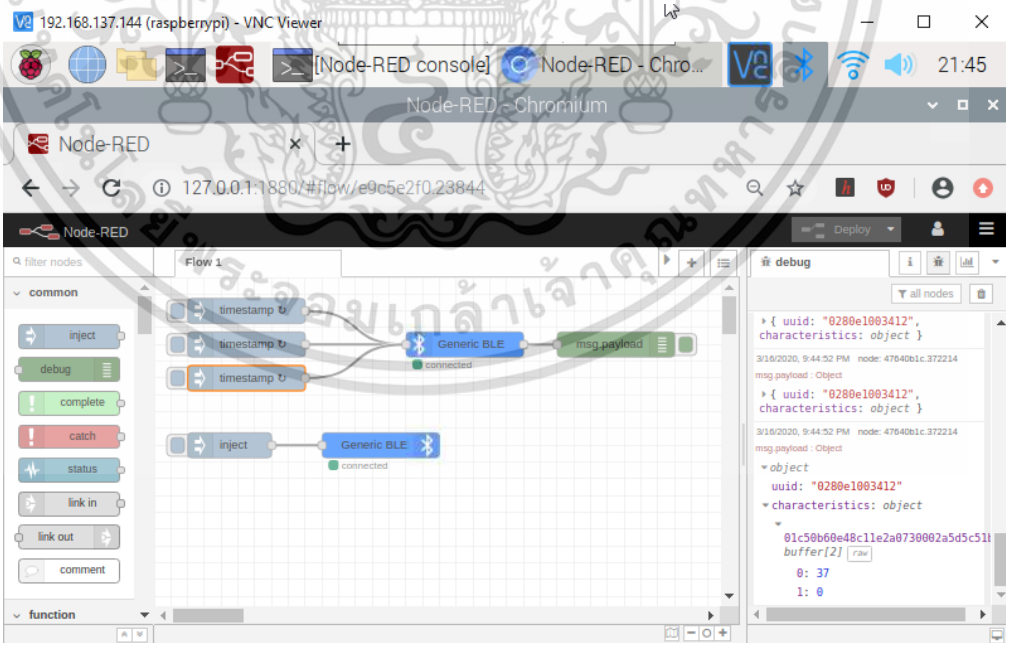


รูปที่ 3.31 แสดงหน้าหลักโปรแกรม Node-Red

3.7 การสื่อสารข้อมูลระหว่าง STM32 และ Raspberry Pi ด้วย Bluetooth

ในส่วนนี้จะใช้เครื่องมือที่มีชื่อว่า Generic BLE ของโปรแกรม Node-Red เป็นตัวช่วยในการสื่อสารข้อมูลผ่าน Bluetooth ระหว่าง Raspberry Pi และ STM32 ตามขั้นตอนดังต่อไปนี้ [35]

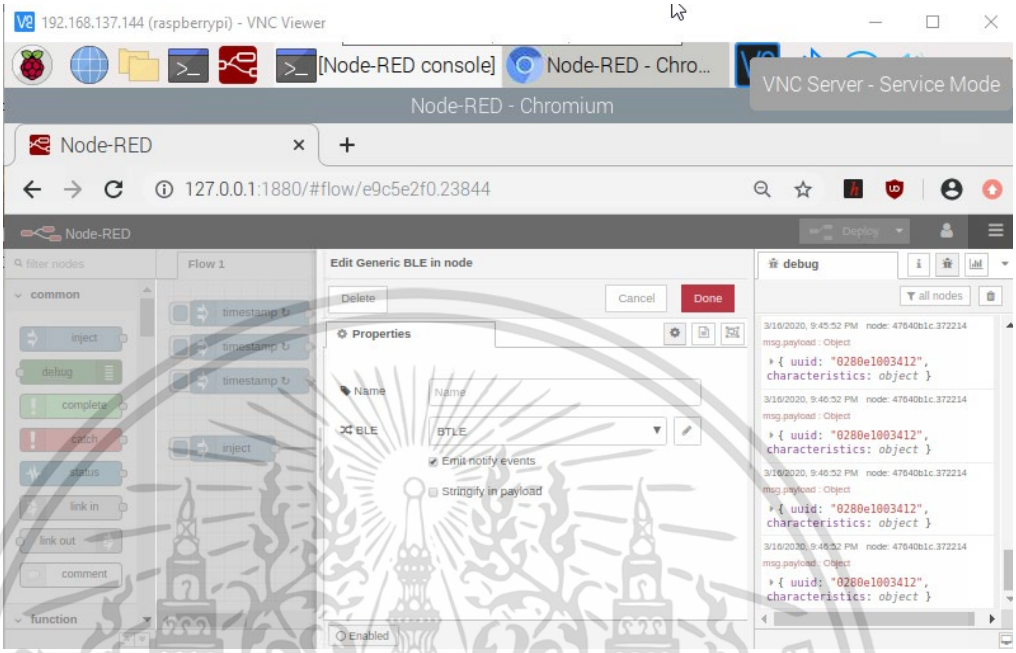
1) สร้างผังการไหลของข้อมูลสำหรับการทำงานของ Generic BLE ตามรูปที่ 3.32



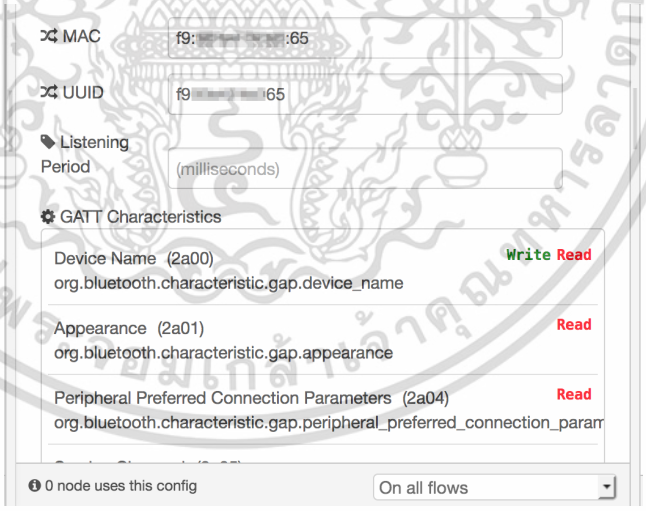
รูปที่ 3.32 แสดงผังการไหลของข้อมูลสำหรับสื่อสารข้อมูลผ่าน Bluetooth

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) โดยการสื่อสารผ่าน Bluetooth จะใช้ Mac Address และ UUID ในการเข้าถึงข้อมูลดังรูปที่ 3.33 และ 3.34



รูปที่ 3.33 แสดงการเชื่อมต่อระหว่าง Raspberry Pi และ STM32 ผ่าน Bluetooth ที่มีชื่อว่า BTLE



รูปที่ 3.34 Mac Address และ UUID ที่ใช้ในการเข้าถึงข้อมูลของ Bluetooth

โดยในส่วนของ GATT Characteristics จะเป็นข้อมูลที่สามารถอ่านได้จาก Bluetooth

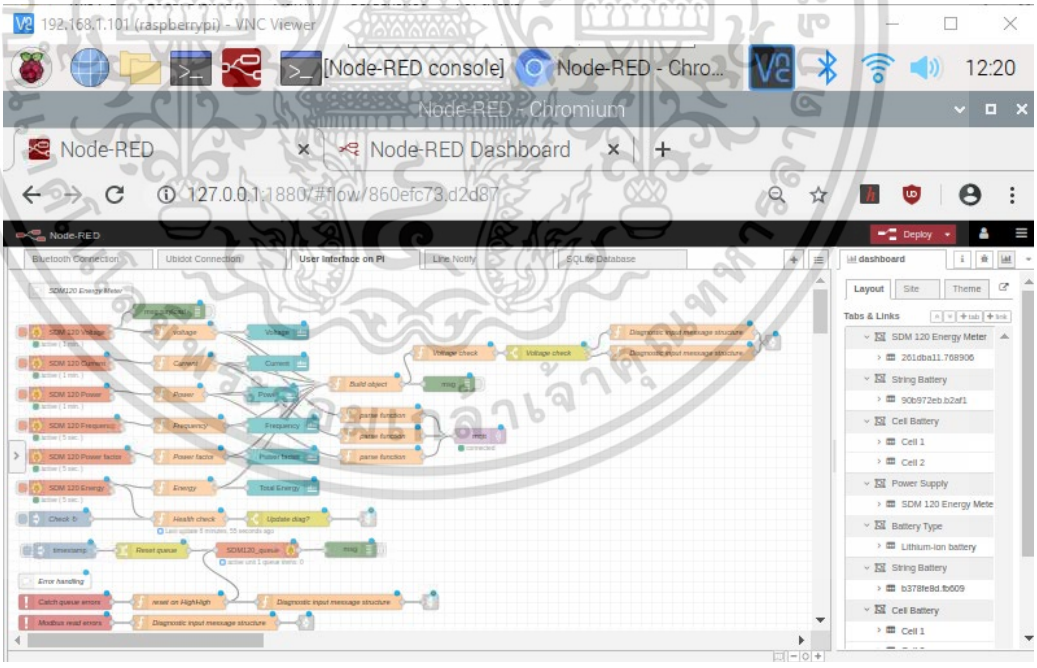
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การออกแบบ User Interface จาก Node-Red บน Raspberry Pi

ในหัวข้อนี้จะเป็นการสร้าง UI ด้วย HTML5 จากโปรแกรม Node-Red เพื่อเป็นสื่อกลางในการติดต่อและโต้ตอบระหว่างผู้ใช้กับอุปกรณ์อิเล็กทรอนิกส์โดยจะแสดงผลผ่านทาง Raspberry Pi สาเหตุที่เลือกใช้ HTML5 เนื่องจากมีการปรับเพิ่ม Feature หลายอย่างเพื่อให้ผู้พัฒนาสามารถใช้งานได้ง่ายขึ้น สามารถแสดงผลได้กับทุก Web browser ลดการใช้ปลั๊กอินพิเศษอย่าง Adobe Flash, Microsoft Silverlight, Apache Pivot สนับสนุน วิดีโอ และองค์ประกอบเสียงรวมทั้งสื่อมัลติมีเดียต่าง ๆ มากขึ้น แต่ก่อนที่จะสร้าง UI จำเป็นต้องทราบก่อนว่าอะไรคือสิ่งที่ต้องการจะแสดงให้ผู้ใช้เห็นและเป็นสิ่งที่สามารถโต้ตอบได้ระหว่างผู้ใช้กับอุปกรณ์อิเล็กทรอนิกส์

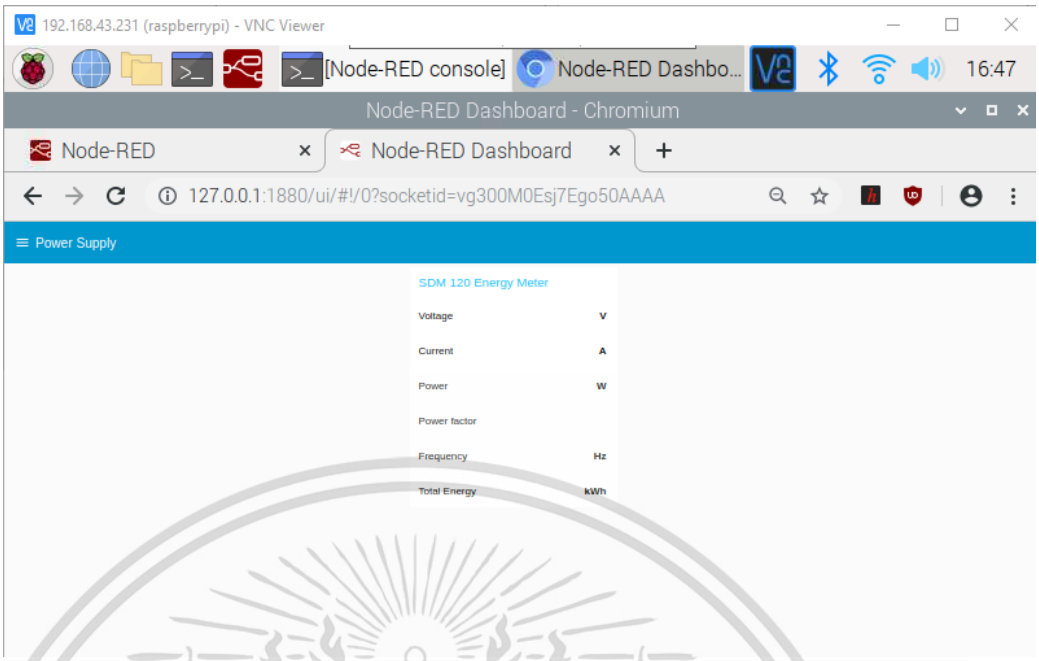
เนื่องจากปริญญานิพนธ์ฉบับนี้นำเสนอเรื่องระบบจัดการแบตเตอรี่บนคลาวด์แพลตฟอร์ม ดังนั้นข้อมูลที่จำเป็นต้องแสดงบน UI ได้แก่ ค่าทางไฟฟ้าของแหล่งจ่ายพลังงานที่วัดผ่าน SDM120 Energy meter, ประเภทของแบตเตอรี่ที่ใช้, ค่าทางไฟฟ้าของ String battery และ Cell battery (แรงดันไฟฟ้า, กระแสไฟฟ้า, อุณหภูมิแบตเตอรี่, อุณหภูมิสิ่งแวดล้อม, SoC และ SoH) โดยวิธีการสร้าง UI จะแสดงตามขั้นตอนดังต่อไปนี้

1) สร้างผังการไหลของข้อมูลสำหรับแสดงค่าทางไฟฟ้าของแหล่งจ่ายไฟที่วัดผ่าน SDM120 Energy meter ตามรูปที่ 3.35 - 3.36



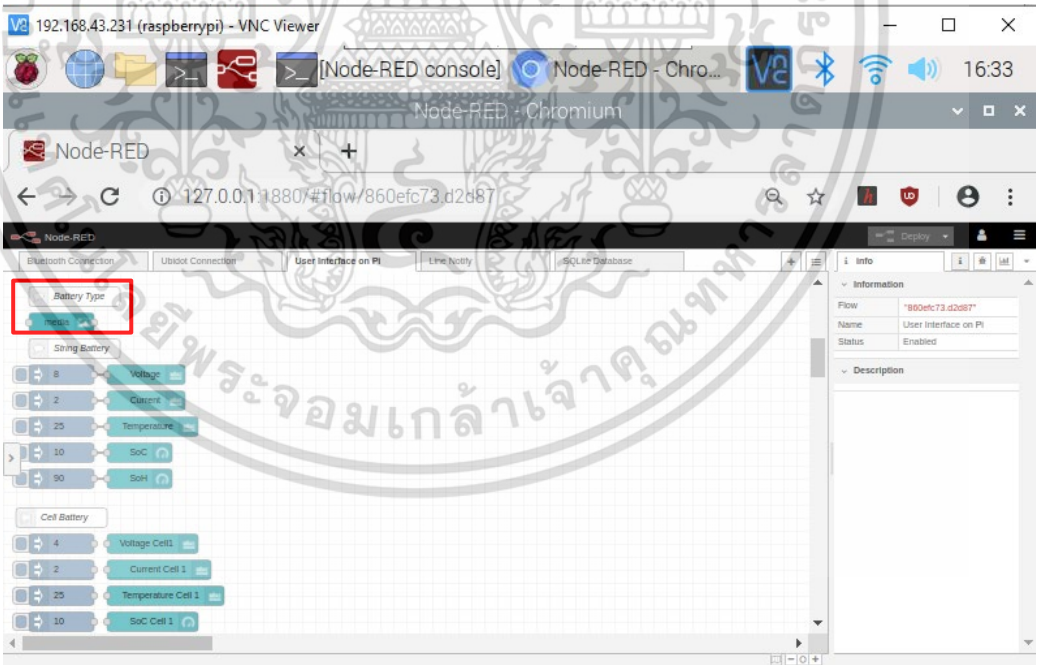
รูปที่ 3.35 ผังการไหลของข้อมูลสำหรับแสดงค่าทางไฟฟ้าของแหล่งจ่ายไฟที่วัดผ่าน SDM120 EnergyMeter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



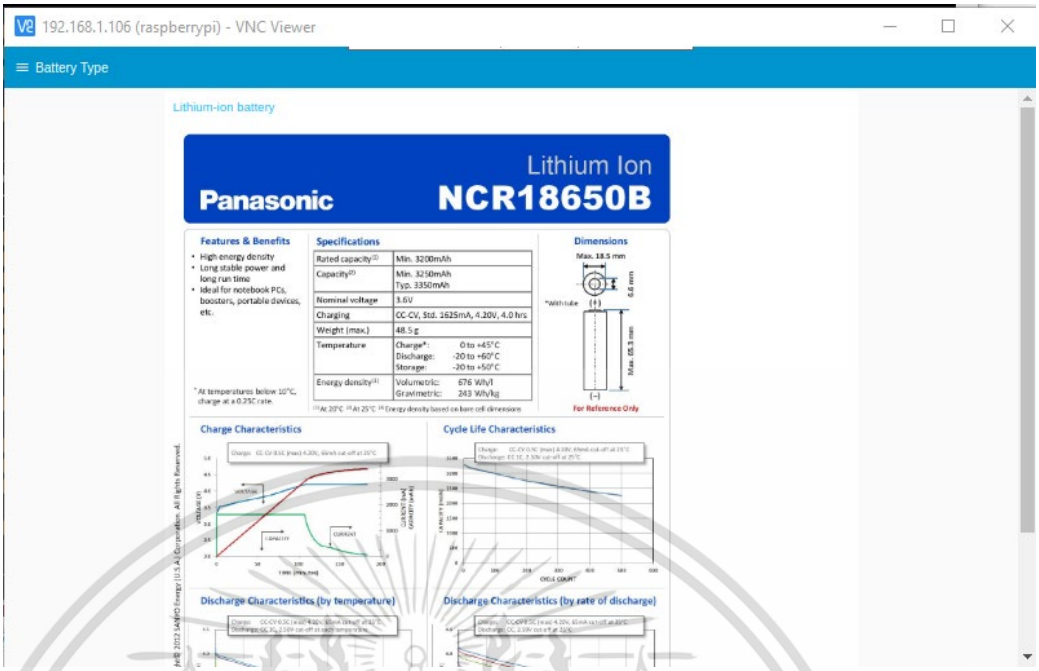
รูปที่ 3.36 Dashboard แสดงค่าทางไฟฟ้าที่วัดได้จาก SDM120 Energy Meter

2) แสดงรายละเอียดของแบตเตอรี่โดยใช้ UI Media ดังรูปที่ 3.37 – 3.38



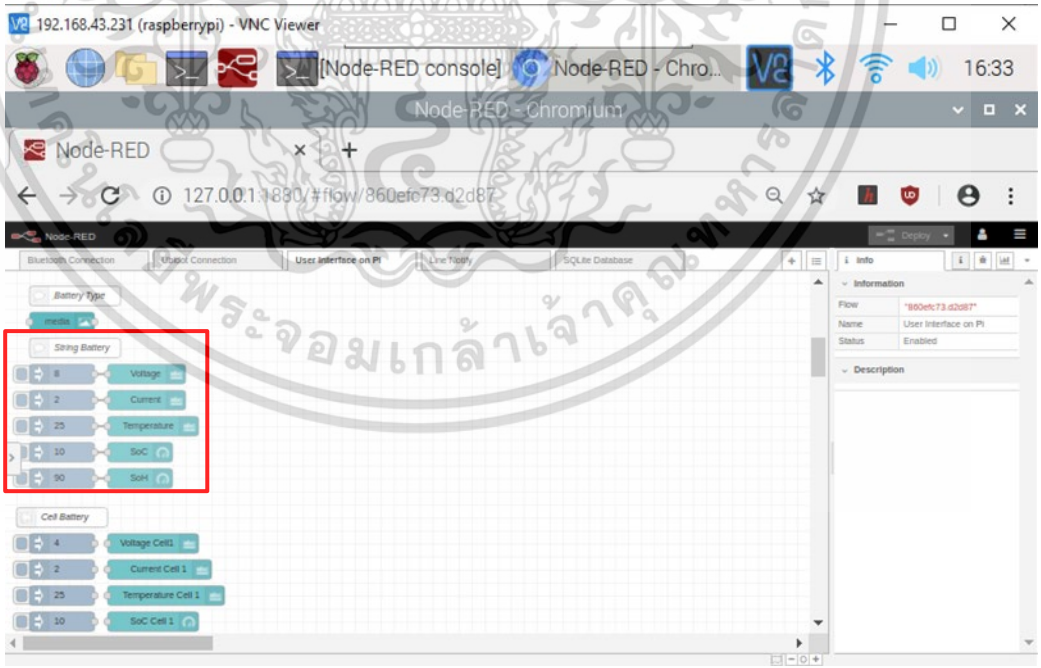
รูปที่ 3.37 ระบุชนิดของแบตเตอรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



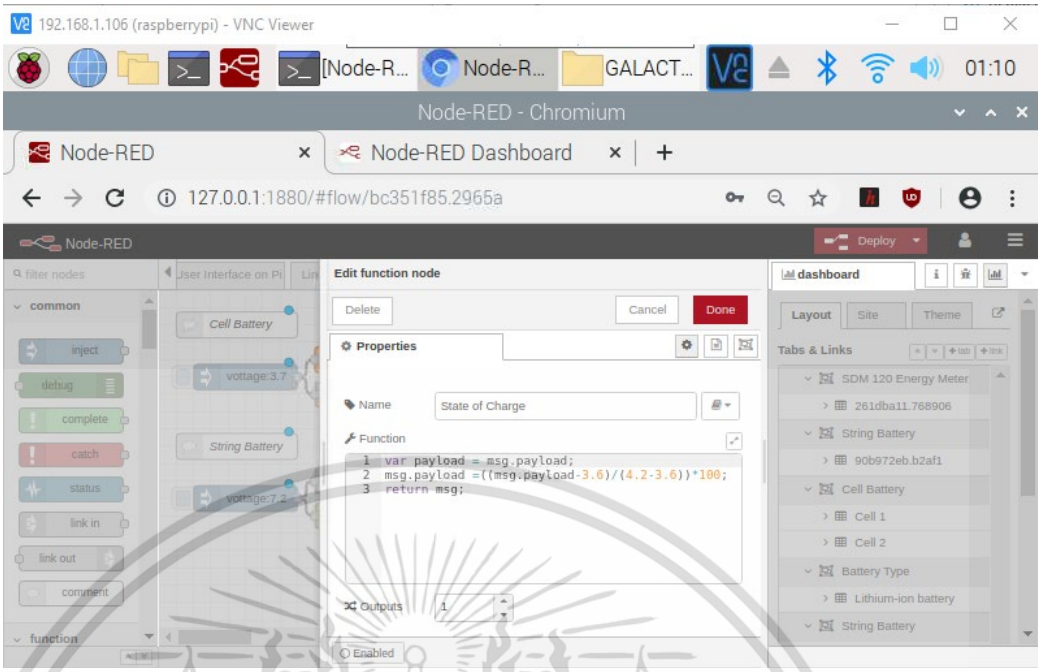
รูปที่ 3.38 Dashboard แสดง datasheet ของแบตเตอรี่

3) สร้างผังการไหลของข้อมูลสำหรับแสดงค่าทางไฟฟ้าของ String battery (แรงดันไฟฟ้า, กระแสไฟฟ้า, อุณหภูมิ, SoC และ SoH) ตามรูปที่ 3.39 – 3.42

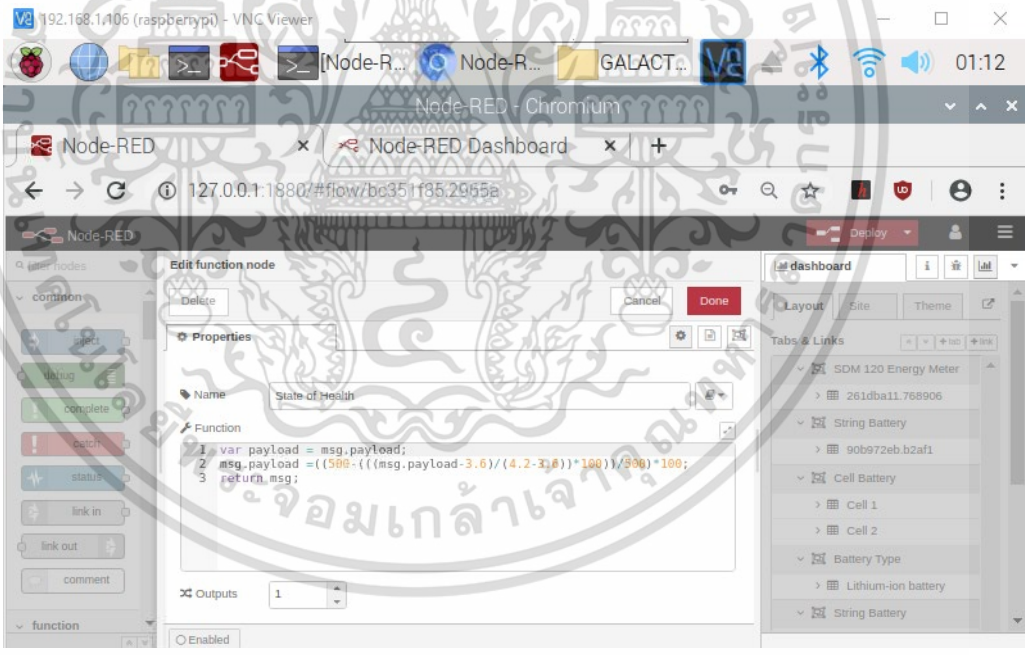


รูปที่ 3.39 ผังการไหลของข้อมูลสำหรับแสดงค่าทางไฟฟ้าของ String Battery

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

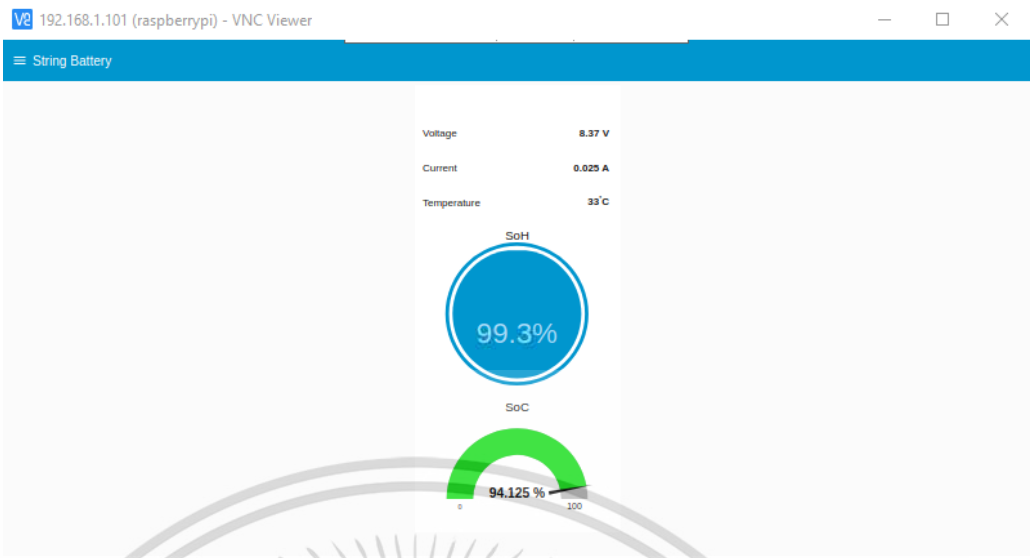


รูปที่ 3.40 ฟังก์ชันสำหรับคำนวณ SoC ของแบตเตอรี่



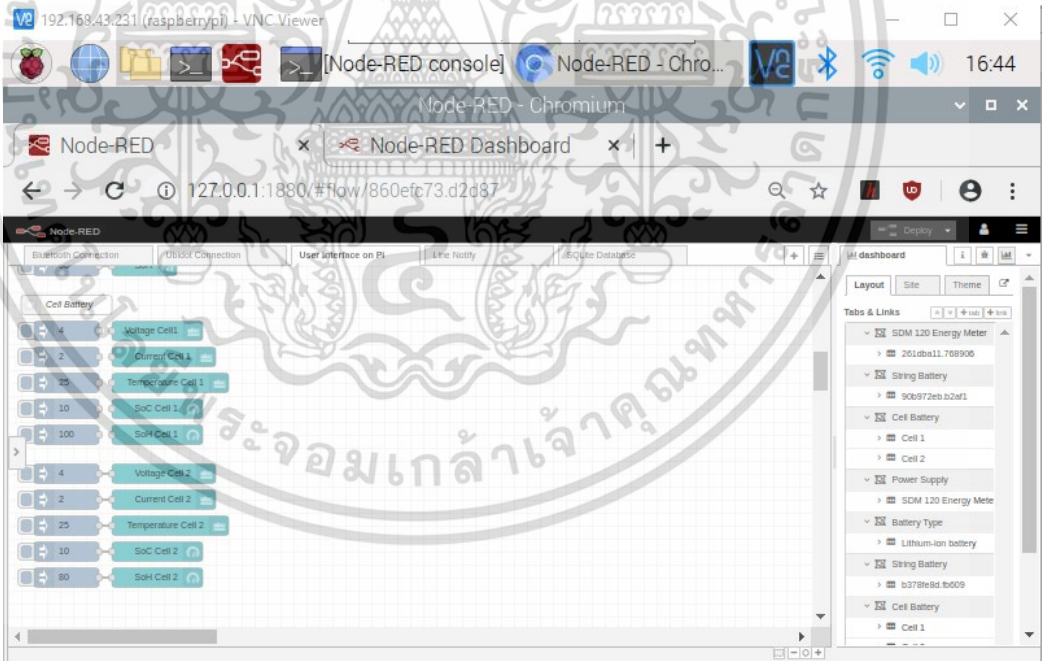
รูปที่ 3.41 ฟังก์ชันสำหรับคำนวณ SoH ของแบตเตอรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



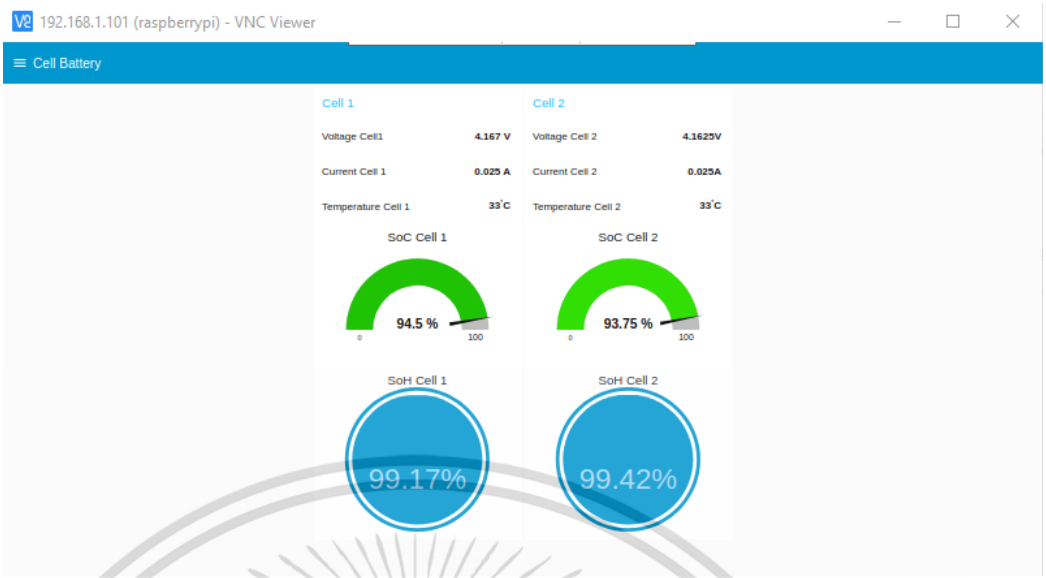
รูปที่ 3.42 Dashboard แสดงค่าทางไฟฟ้าของ String Battery

4) สร้างผังการไหลของข้อมูลสำหรับแสดงค่าทางไฟฟ้าของ Cell Battery (แรงดันไฟฟ้า, กระแสไฟฟ้า, อุณหภูมิ, SoC และ SoH) ดังรูปที่ 3.43 – 3.44



รูปที่ 3.43 ผังการไหลของข้อมูลสำหรับแสดงค่าทางไฟฟ้าของ Cell Battery

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

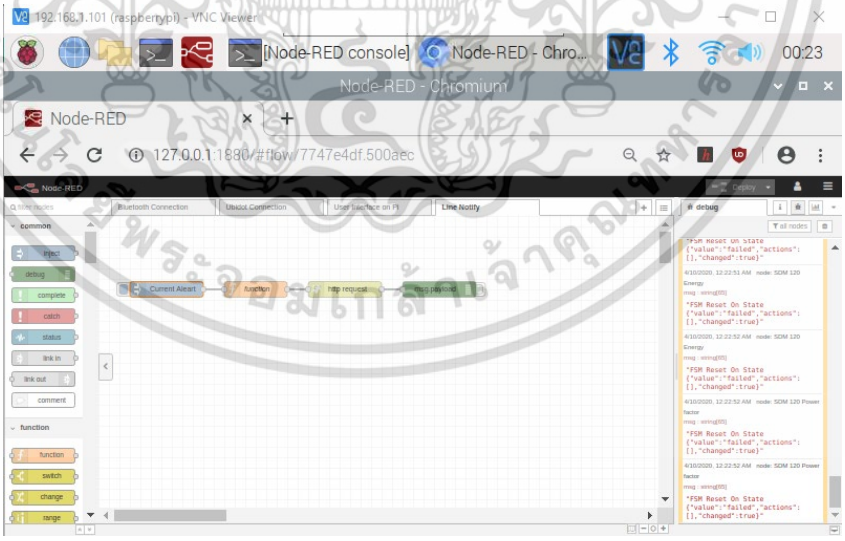


รูปที่ 3.44 Dashboard แสดงค่าทางไฟฟ้าของ Cell Battery

3.9 การแจ้งเตือนผ่าน Line notify ด้วย Node-Red

การแจ้งเตือนผ่าน Line notify ด้วย Node-Red มีขั้นตอนในการจัดทำดังต่อไปนี้

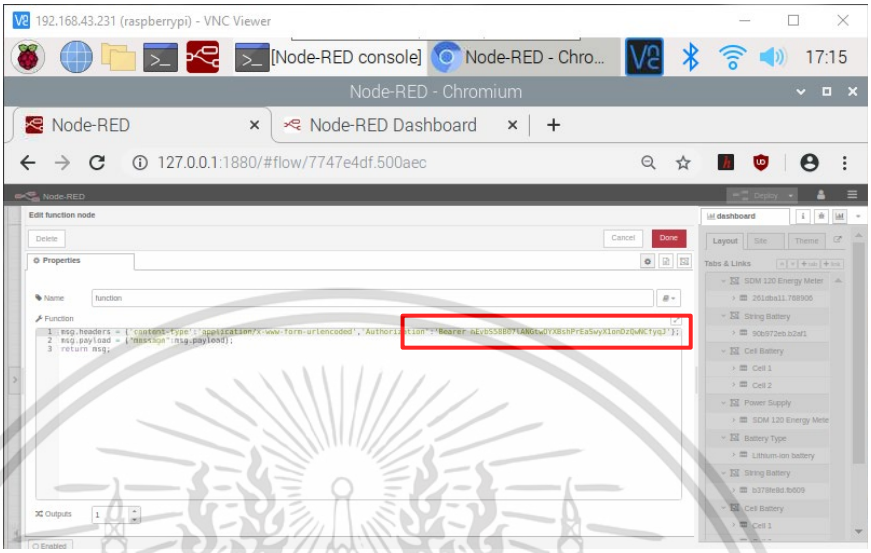
1) สร้างผังการไหลของข้อมูลเพื่อทำการแจ้งเตือนผ่าน Line notify ตามรูปที่ 3.45 (กรณีที่ Bluetooth ของ STM32 ขาดการเชื่อมต่อกับ Bluetooth ของ Raspberry Pi) [36]



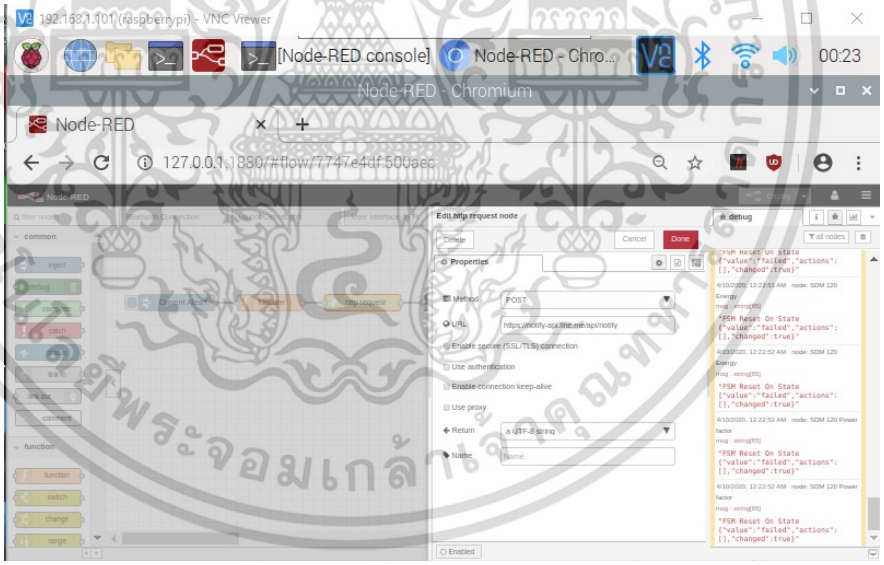
รูปที่ 3.45 ผังการไหลของข้อมูลเพื่อทำการแจ้งเตือนผ่าน Line notify

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เขียนฟังก์ชันตามรูปที่ 3.46 – 3.47 โดยในกรอบสี่เหลี่ยมคือ Token ของ Line notify โดยขั้นตอนการออก Token สามารถดูได้จาก https://notify-bot.line.me/th/



รูปที่ 3.46 ฟังก์ชันเพื่อทำการแจ้งเตือนผ่าน Line notify



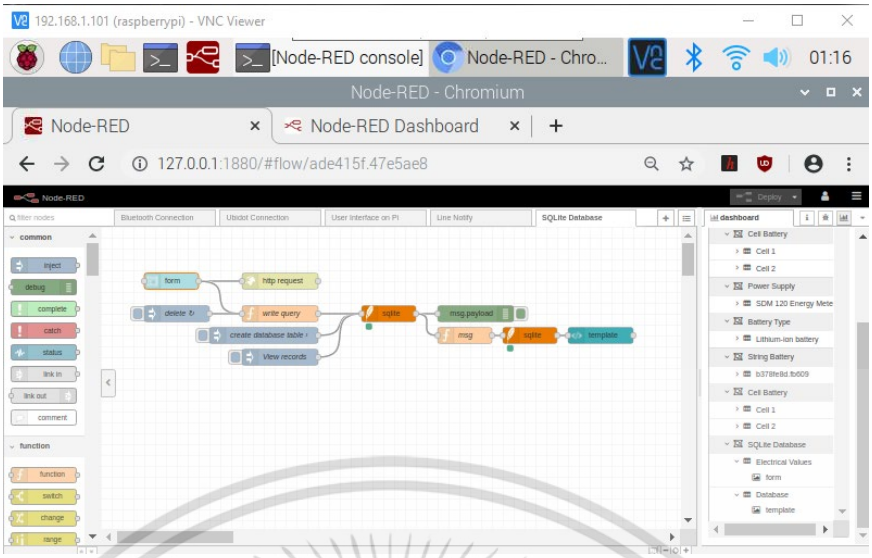
รูปที่ 3.47 ตั้งค่า http request

3.10 การสร้างฐานข้อมูล SQLite บน Raspberry Pi

เนื่องจาก Raspberry Pi มี SD card เป็นหน่วยความจำเพราะฉะนั้นพื้นที่ในการเก็บข้อมูลจึงค่อนข้างจำกัด ดังนั้นจึงเลือกใช้ SQLite เพราะว่า SQLite เป็นโปรแกรมฐานข้อมูลที่มีขนาดเล็กมาก เก็บฐานข้อมูลเป็นไฟล์โดยไม่จำเป็นต้องมีเซิร์ฟเวอร์ โดยวิธีการทำมีดังต่อไปนี้ [37]

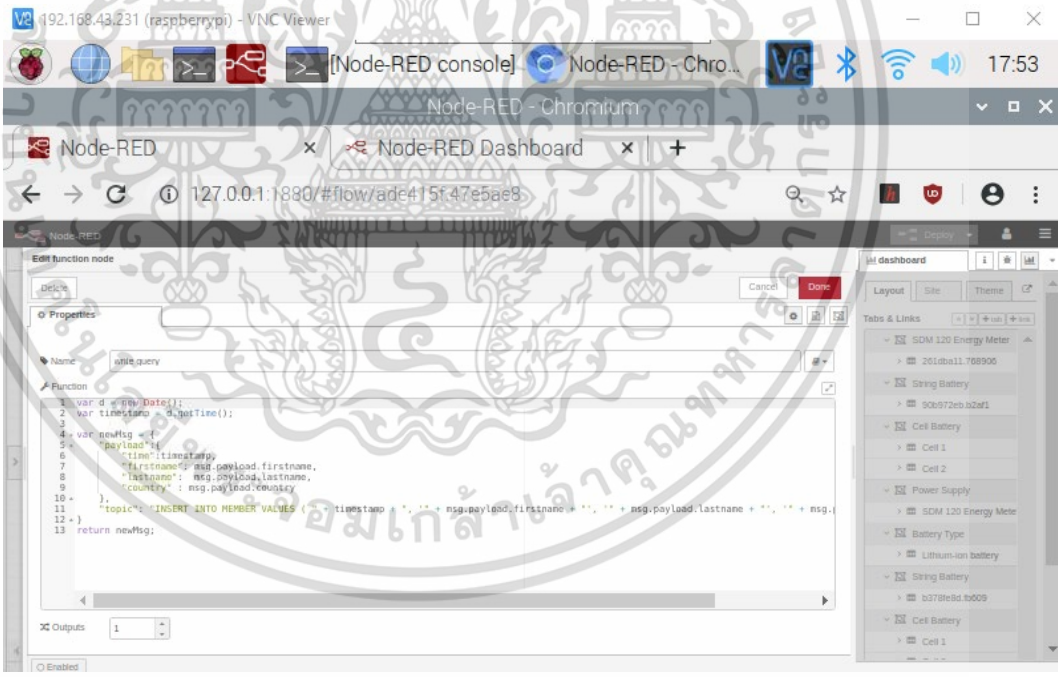
1) สร้างผังการไหลของข้อมูลตามรูปที่ 3.48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



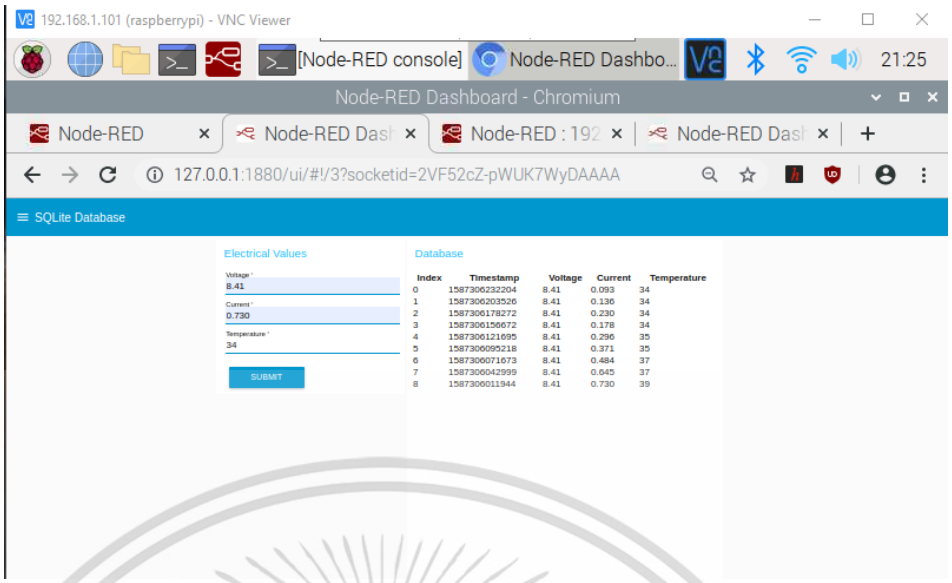
รูปที่ 3.48 ผังการไหลของข้อมูลเพื่อทำ SQLite บน Raspberry Pi

2) เขียนฟังก์ชันเพื่อรับและแสดงค่าผ่าน Node-Red ตามรูปที่ 3.49 – 3.51

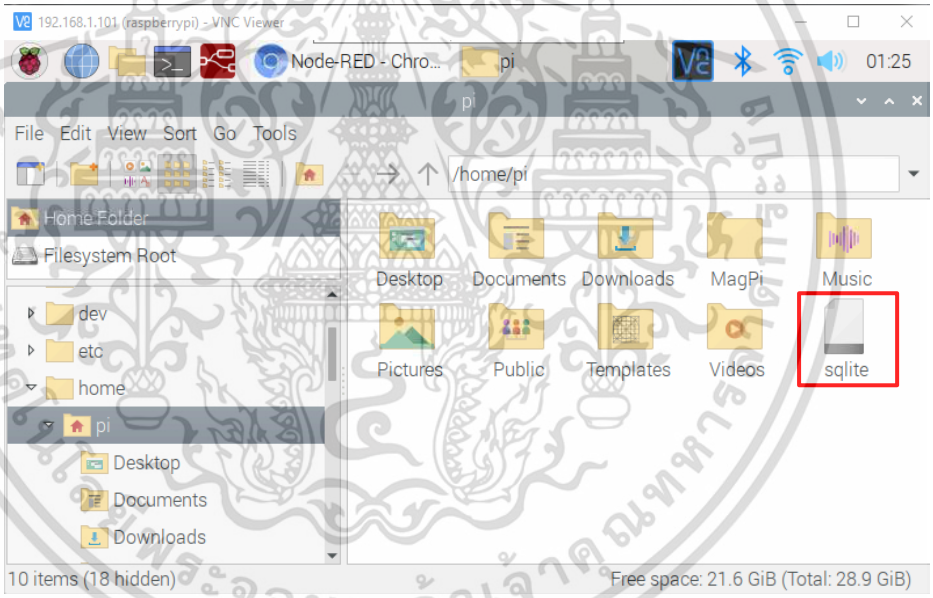


รูปที่ 3.49 ฟังก์ชันเพื่อรับและแสดงค่าผ่าน Node-Red

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.50 Dashboard แสดงข้อมูลภายในฐานข้อมูล SQLite



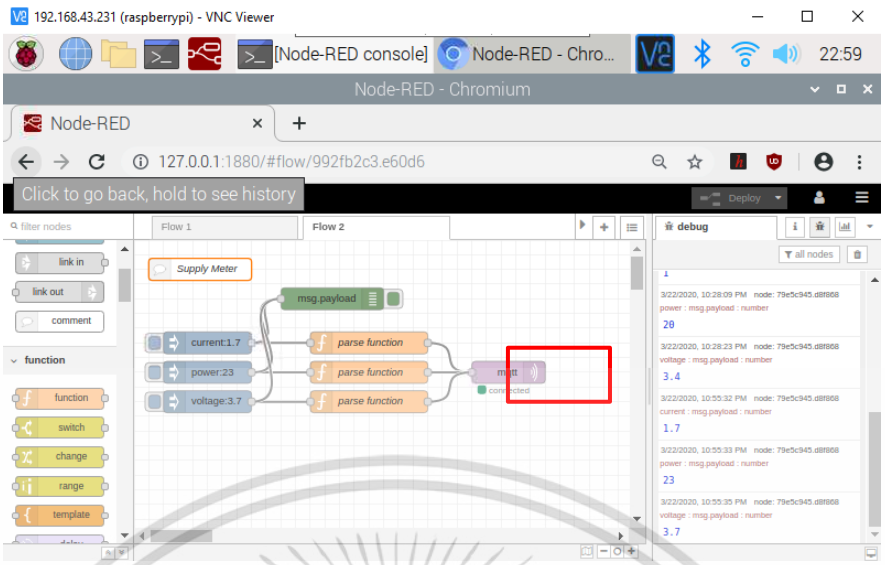
รูปที่ 3.51 ไฟล์ SQLite ที่จัดเก็บอยู่ใน Raspberry Pi

3.11 การสื่อสารข้อมูลระว่าง Raspberry Pi และ Cloud computing ด้วย MQTT โปรโตคอล

เนื่องจาก MQTT โปรโตคอลถูกออกแบบมาเพื่อใช้งานกับอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็กเป็นการเชื่อมต่อแบบ M2M หมายถึงอุปกรณ์เชื่อมต่อกับอุปกรณ์ สนับสนุนเทคโนโลยี Internet of Things โดยที่อินเทอร์เน็ตสามารถเชื่อมต่อกับอุปกรณ์ต่าง ๆ ได้ ซึ่งการสื่อสารข้อมูลระหว่าง Raspberry Pi และ Ubidots Cloud computing (Ubidot) มีขั้นตอนดังต่อไปนี้

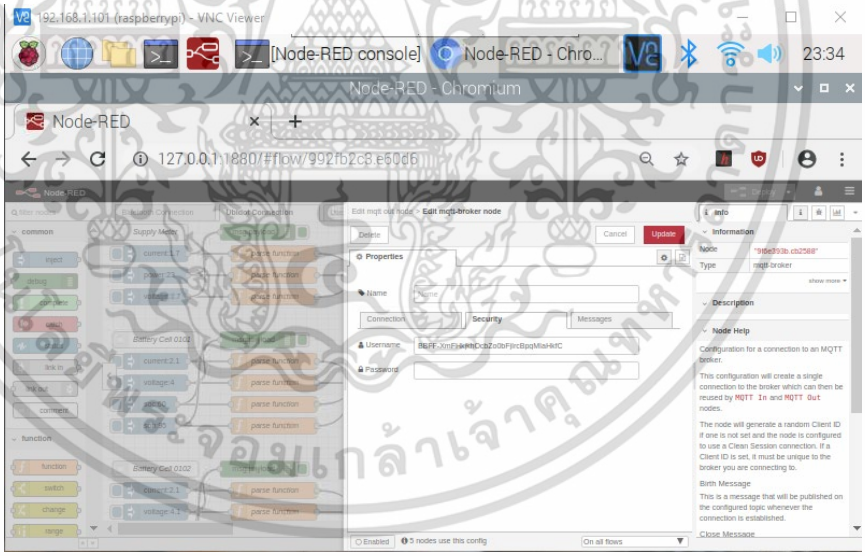
1) สร้างผังการไหลของข้อมูลโดยใช้โหนด MQTT โปรโตคอลตามรูปที่ 3.52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.52 แสดงผังการไหลของข้อมูลโดยใช้ MQTT โปรโตคอล

2) ดับเบิ้ลคลิ๊กที่ Blog ของ MQTT เพื่อตั้งค่าในการเข้าถึง application ของ Ubidots ผ่าน MQTT โปรโตคอลตามรูปที่ 3.53



รูปที่ 3.53 แสดงการกำหนด Token เพื่อเข้าถึงการสื่อสารข้อมูลผ่าน MQTT โปรโตคอล

โดยขั้นตอนการออก Token สามารถดูได้จากหัวข้อ 3.17 การรับค่าจาก Edge computing

3.12 Siemens Simatic IOT 2040 to Cloud

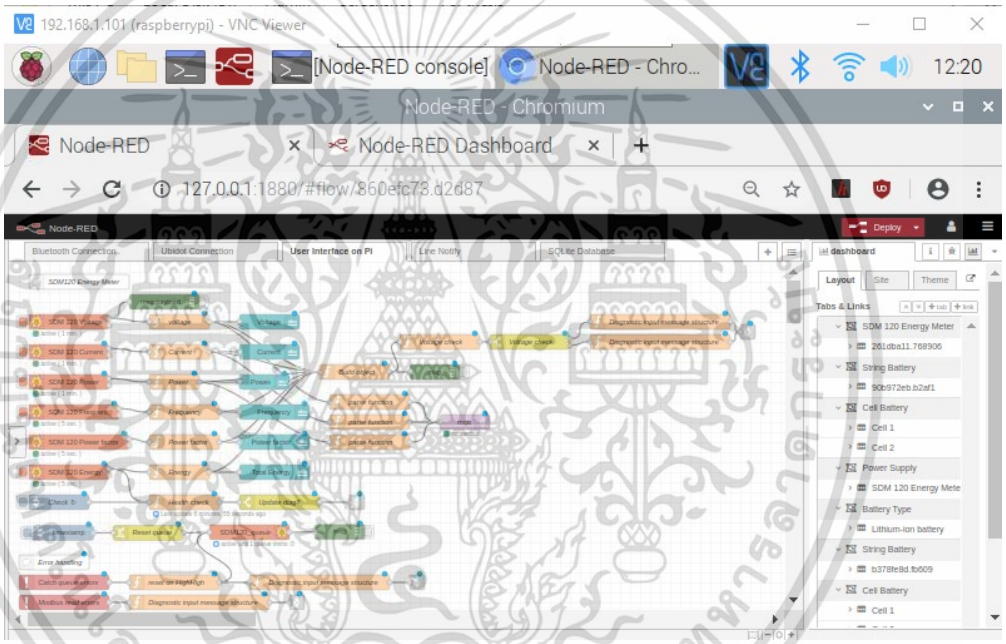
สำหรับ Siemens Simatic IOT 2040 ใช้ระบบปฏิบัติการเป็น Yocto Linux เนื่องจากระบบปฏิบัติการนี้ถูกพัฒนามาเพื่องานทางด้าน Embedded System เป็นระบบปฏิบัติการที่สร้าง Linux

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Distribution มีความสามรถในการ portability ด้วยขนาดที่เล็กแต่ทำงานบนอุปกรณ์ที่มีความเร็วของฮาร์ดแวร์ไม่เท่ากันได้ ใช้ Siemens Simatic IOT 2040 ในการวัดค่าพลังงานไฟฟ้าของเพาเวอร์ซัพพลายที่ผ่าน SDM120CT Energy Meter จะแสดงผลบน Raspberry Pi และส่งไปยัง Cloud ผ่าน MQTT โปรโตคอล โดยจะแสดงตามขั้นตอนต่อไปนี้

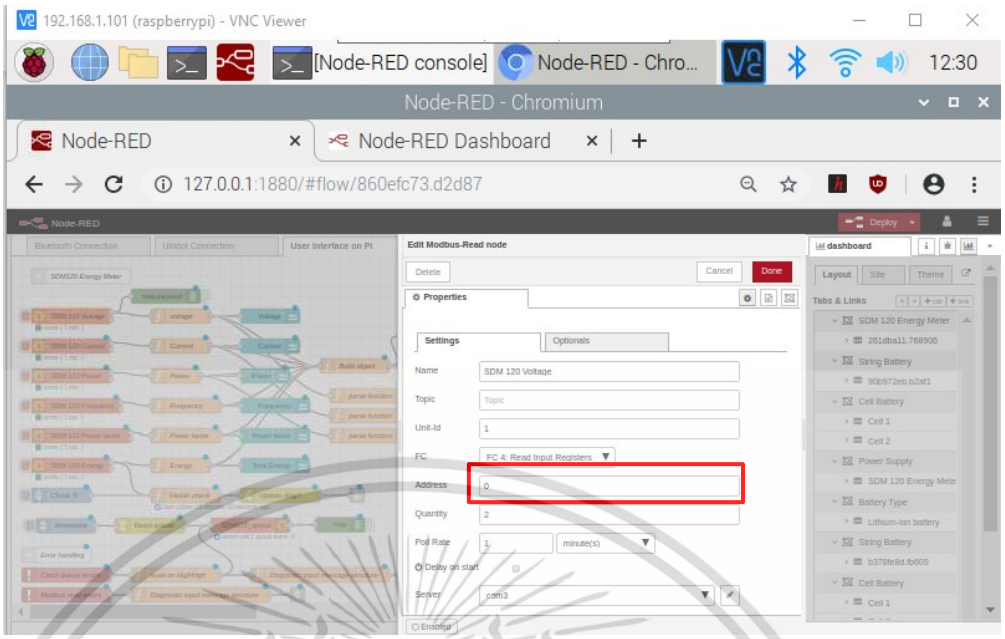
1) ติดตั้งระบบปฏิบัติการสำหรับ Siemens Simatic IOT 2040 โดยขั้นตอนในการติดตั้งสามารถดูได้จาก https://help.ubidots.com/en/articles/2046638-setting-up-the-siemens-simatic-iot2000

2) เปิดโปรแกรม Node-Red และสร้างผังการไหลข้อมูลสำหรับการอ่านค่าพลังงานไฟฟ้าของเพาเวอร์ซัพพลายที่วัดได้จาก SDM120CT Energy meter ตามรูปที่ 3.54 – 3.55



รูปที่ 3.54 แสดงผังการไหลข้อมูลการวัดค่าพลังงานไฟฟ้าของเพาเวอร์ซัพพลาย

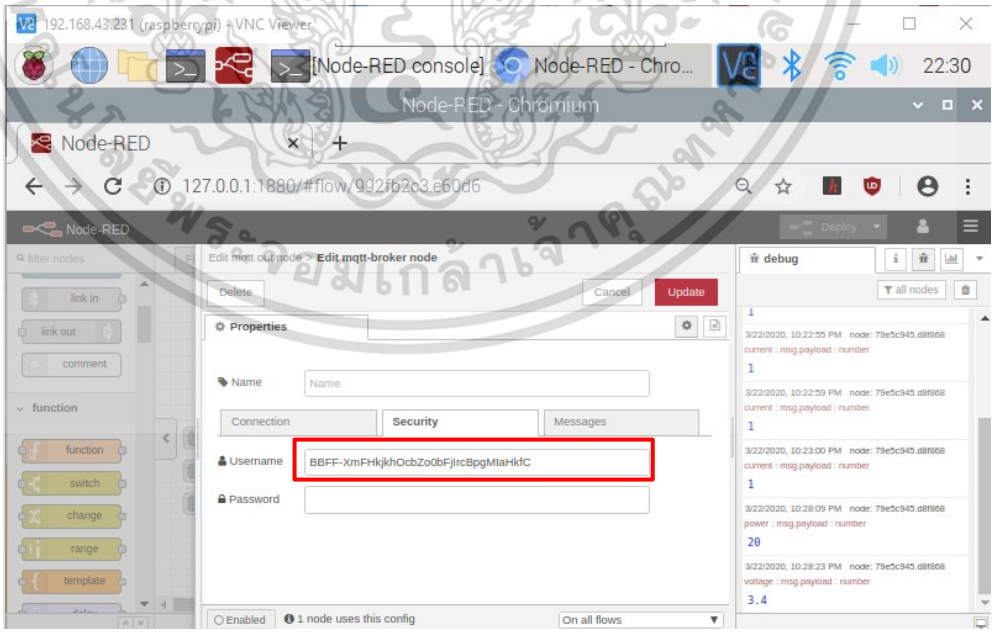
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.55 กำหนด Address เพื่อรับค่าที่ต้องการวัดจาก SDM120CT Energy meter

โดย Address สามารถดูได้จากภาคผนวก SDM120CT Energy meter datasheet

3) ดับเบิ้ลคลิ๊กที่โหนด MQTT จากนั้นใส่ Token ของ Ubidots เพื่อทำการสื่อสารข้อมูลตามรูปที่ 3.56



รูปที่ 3.56 ใส่ Token ของ Ubidots เพื่อทำการสื่อสารข้อมูลผ่าน MQTT โปรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการออก Token ของ Ubidots สามารถดูได้จากหัวข้อ 3.17 การรับค่าจาก Edge computing

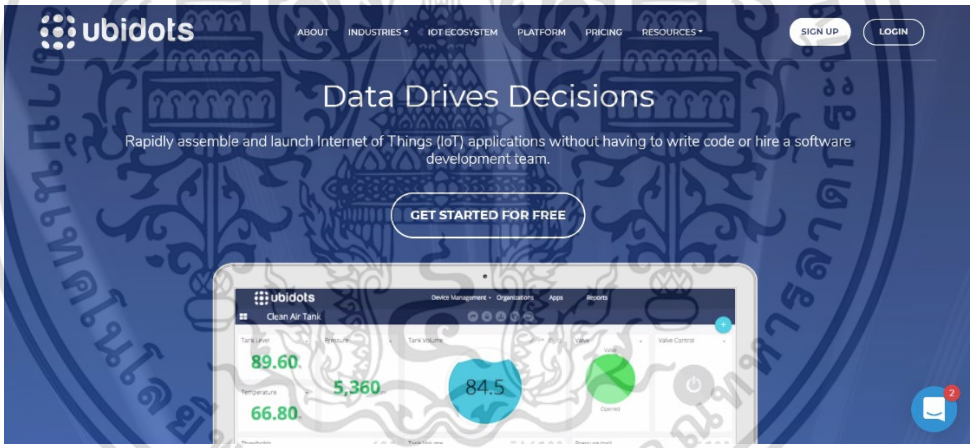
3.13 การเลือกผู้ให้บริการ Cloud Platform

จากคุณสมบัติที่ทำการออกแบบ ผู้จัดทำจึงได้เลือกใช้ Ubidots ซึ่งเป็น IoT Application Development Platform โดยอัตราค่าบริการของ Ubidodts จะขึ้นอยู่กับระดับที่เลือกใช้งาน โดยในโครงงานนี้ผู้จัดทำได้เลือกใช้แบบ Professional ซึ่งมีค่าใช้จ่ายอยู่ที่ $199/month เนื่องจากมีคุณสมบัติเพียงพอต่อการใช้งาน ซึ่งในการจัดทำโครงงานเป็นเพียงตัวทดลองใช้เท่านั้น

3.14 การสร้าง Application โดยใช้ Ubidots

ทำการสมัครใช้งาน Ubidots โดยมีขั้นตอนดังต่อไปนี้ [38] - [39]

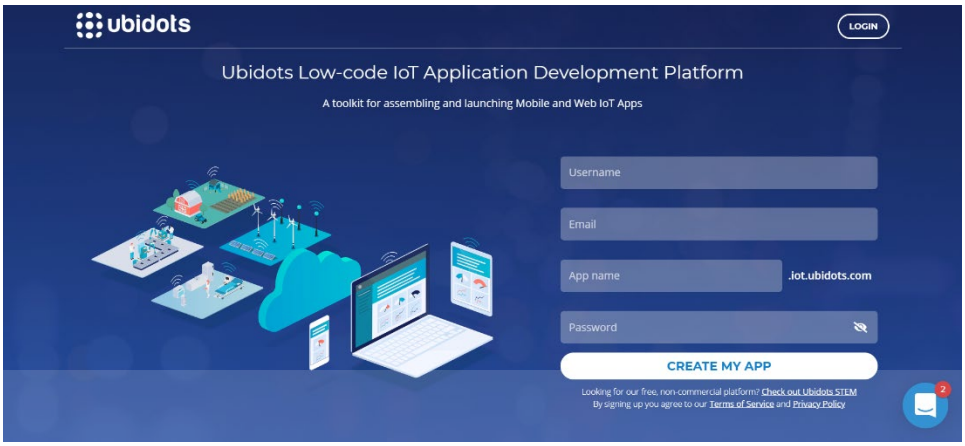
1) เข้าไปที่เวปไซต์ https://www.ubidots.com/ แล้วเลือก Sign up หน้าเวปเพจของ Ubidots จะแสดงดังรูปที่ 3.57



รูปที่ 3.57 แสดงหน้าเวปเพจของ Ubidots

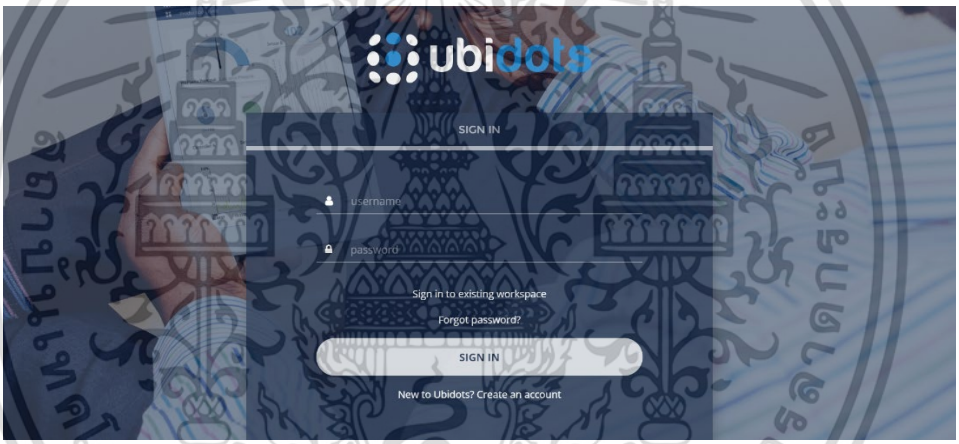
2) กรอกข้อมูลสำหรับการลงทะเบียนจากนั้นเลือก create my app เพื่อสร้าง Application หน้าลงทะเบียนของ Ubidots จะแสดงดังรูปที่ 3.58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



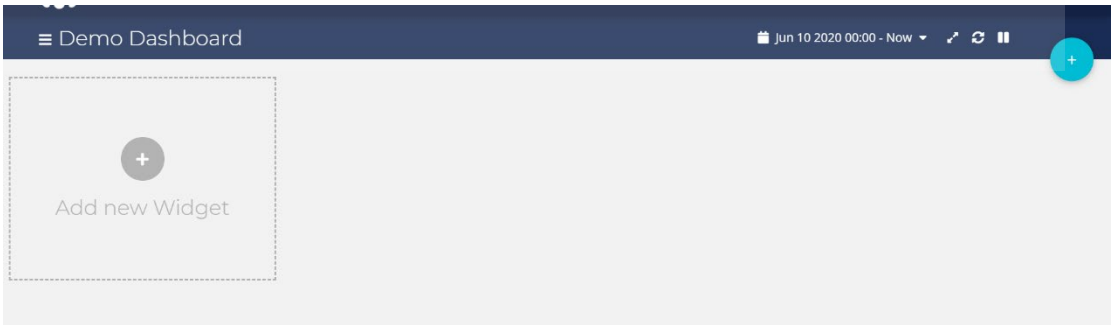
รูปที่ 3.58 แสดงหน้าลงทะเบียนของ Ubidots

3) เลือก Login โดยหน้าลงทะเบียนของ Ubidots จะแสดงดังรูปที่ 3.59



รูปที่ 3.59 แสดงหน้าลงทะเบียนของ Ubidots

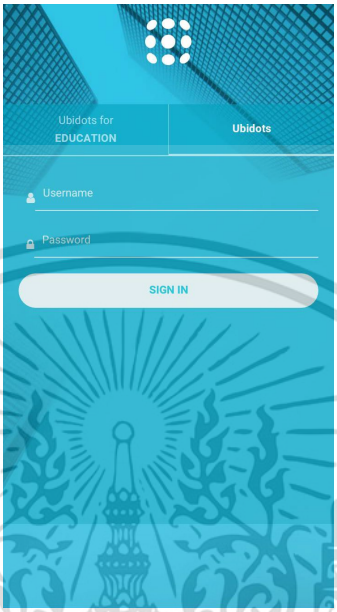
4) เมื่อทำการลงทะเบียนเข้าสู่แอปพลิเคชัน หน้า Dashboard จะเป็นหน้าแรกที่ปรากฏแสดงดังรูปที่ 3.60



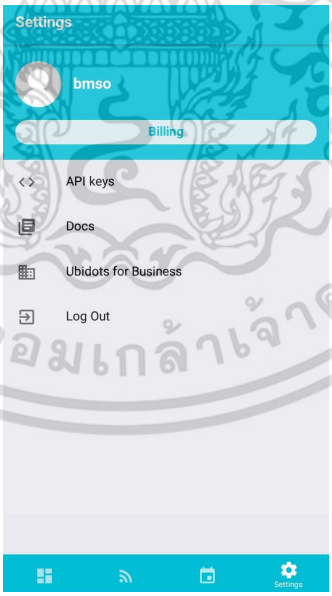
รูปที่ 3.60 แสดงหน้า Demo Dashboard ของแอปพลิเคชัน Ubidots ที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน Ubidots บน Smartphoneในระบบ Android สามารถทำได้โดยการดาวน์โหลด Application หลังจากนั้นทำการเข้าสู่ระบบในหน้าลงทะเบียน (รูปที่ 3.61 และสามารถเข้าถึงข้อมูลผู้ใช้งานได้จากหน้าข้อมูลผู้ใช้งานดังรูปที่ 3.62



รูปที่ 3.61 แสดงหน้าลงทะเบียนของแอปพลิเคชัน Ubidots บน Smart phone



รูปที่ 3.62 แสดงหน้าหน้าข้อมูลผู้ใช้งานของแอปพลิเคชัน Ubidots บน Smart phone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

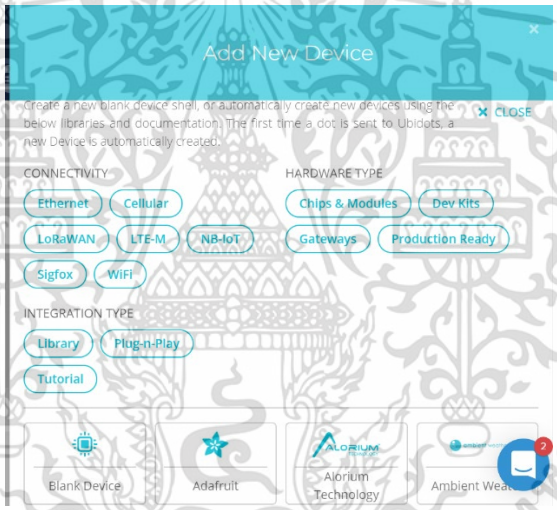
3.15 การสร้างอุปกรณ์, ตัวแปรและกลุ่มของอุปกรณ์

ทำการสร้างอุปกรณ์เพื่อทำการเก็บค่าตัวแปรที่รับมาจากส่วน Edge computing และทำการสร้างกลุ่มของอุปกรณ์ (Device Group เพื่อจัดกลุ่มของอุปกรณ์ที่เกี่ยวข้องกันให้เป็นกลุ่ม โดยมีขั้นตอนในการสร้างอุปกรณ์, ตัวแปรและกลุ่มของอุปกรณ์ดังต่อไปนี้

3.15.1 การสร้างอุปกรณ์สำหรับเก็บตัวแปรต่าง ๆ ที่รับค่ามาจากส่วน Edge computing

โดยชื่ออุปกรณ์ที่สร้างขึ้นนี้จะถูกใช้ในการอ้างอิงเพื่อส่งค่าจาก Edge computing ขั้นตอนในการสร้างอุปกรณ์มีดังต่อไปนี้

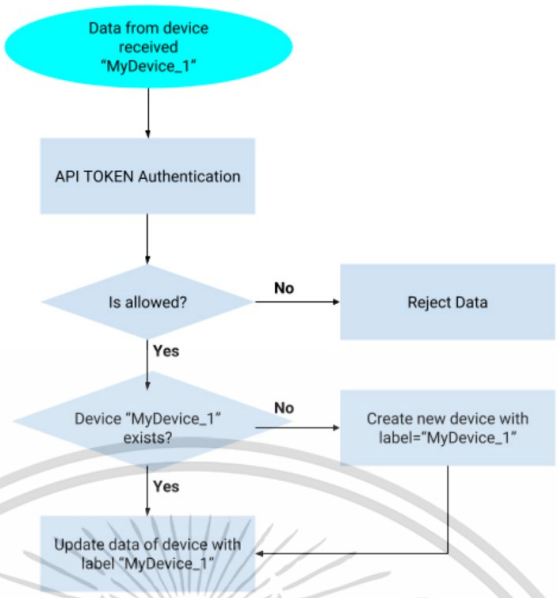
1) เลือก Devices ในแถบเมนูด้านบน ทำการสร้างอุปกรณ์ใหม่ โดยกดที่ Create Device จะปรากฏหน้าต่างดังรูปที่ 3.63 เลือก Filters หรือเลือก Blank Device (ตัวอย่างนี้เลือก Blank Device)



รูปที่ 3.63 แสดงหน้าต่างในการสร้างอุปกรณ์

2) กำหนดชื่อ Device name และ Device label โดยชื่อที่ตั้งขึ้นจะถูกนำไปใช้การอ้างอิงเพื่อรับค่าจาก Edge computing ดังผังงานดังต่อไปนี้ [40]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.64 ผังงานแสดงการรับค่าจากอุปกรณ์ของ Uidots

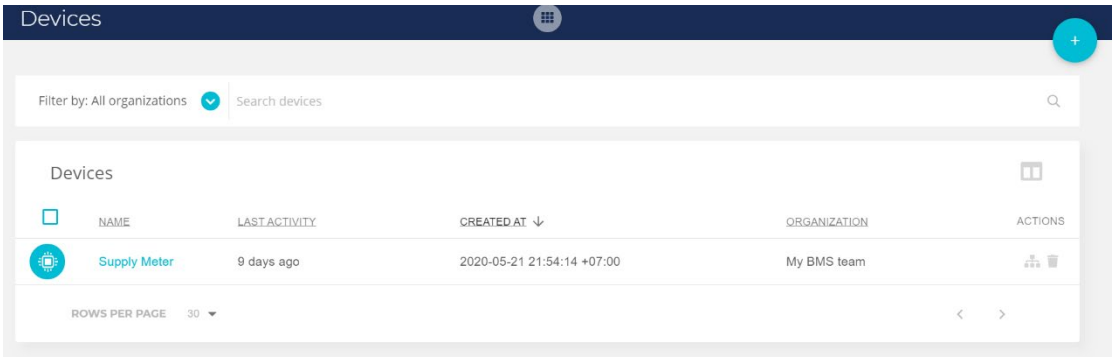
ที่มา : https://help.ubidots.com/en/articles/1330905-automatically-provision-devices-and-variables-with-ubidots-api-labels

ตารางที่ 3.3 แสดง Device name และ Device label ที่กำหนดขึ้น

Devices		
No.	Device name	Device label
1	Supply Meter	supply-meter
2	Batter String 01	Battery-string-01
3	Batter Cell 0101	Battery-cell-0101
4	Batter Cell 0102	Battery-cell-0102

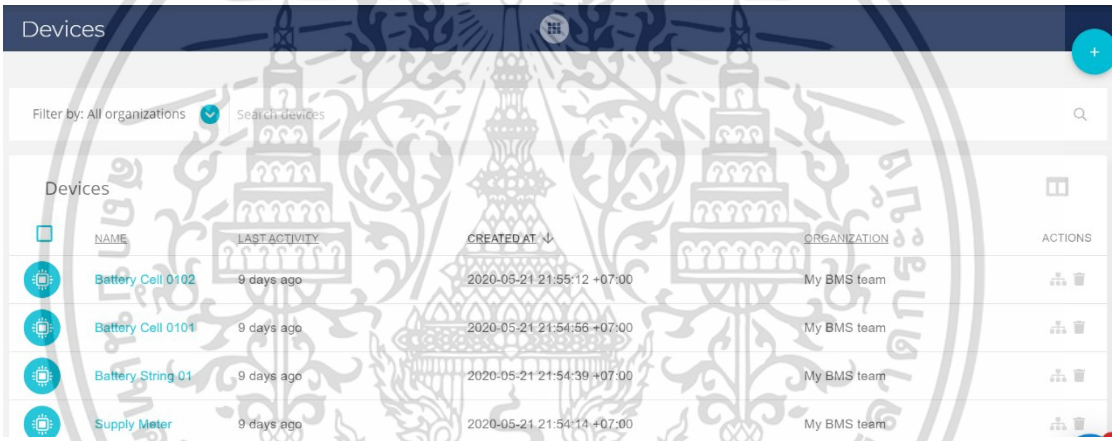
พิมพ์พ์ชื่ออุปกรณ์ตามตารางที่3.3 โดยในเบื้องต้นจะแสดงการสร้างอุปกรณ์ชื่อ Supply Meter เมื่อการสร้างอุปกรณ์ Supply Meter เสร็จสิ้นจะปรากฏหน้าต่างดังรูปที่ 3.65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.65 แสดงรายละเอียดของอุปกรณ์ Supply Meter ที่ทำการสร้างขึ้น

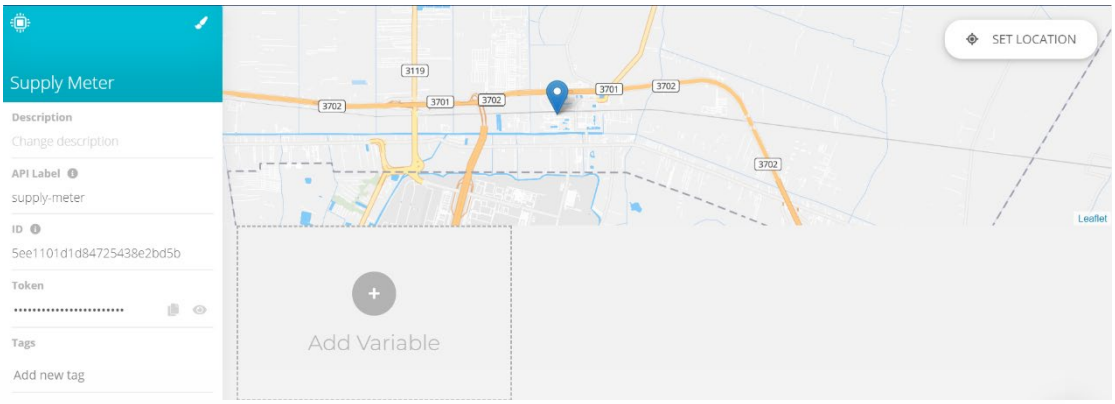
3) สร้างอุปกรณ์ Batter String 01, Batter Cell 0101และ Batter Cell 0102 ตามตารางที่ 3.3 เมื่อสร้างอุปกรณ์เสร็จสิ้นจะปรากฏหน้าต่างดังรูปที่ 3.66



รูปที่ 3.66 แสดงรายละเอียดของ Device ทั้งหมดที่ทำการสร้างขึ้น

4) เข้าไปที่อุปกรณ์ที่ต้องการเพื่อกำหนดรายละเอียดเพิ่มเติมของ Devices เช่น Description, Location (ตำแหน่งที่กำหนดคือ ตึกภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ใน รูปที่ 6.7 จะแสดงการกำหนดรายละเอียดของอุปกรณ์ชื่อ Supply Meter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.67 แสดงรายละเอียดของอุปกรณ์ชื่อ Supply Meter

3.15.2 การสร้างตัวแปร

โดยในแต่ละอุปกรณ์จะต้องมีตัวแปรข้างในอุปกรณ์ซึ่ง Edge computing จะทำการส่งค่าผ่านตัวแปร นอกจากนี้ตัวแปรที่ทำการสร้างขึ้นสามารถเป็นได้ทั้งค่าดิบ (Raw ที่รับมาจาก Edge computing หรือเป็น Synthetic ที่เกิดจากการสร้างสมการคำนวณ ขั้นตอนในการสร้างตัวแปรมีดังต่อไปนี้

1) กำหนดรายละเอียดของตัวแปรที่ต้องการสร้างในแต่ละอุปกรณ์ดังตารางที่ 3.4

ตารางที่ 3.4 แสดงรายละเอียดของตัวแปรที่กำหนดขึ้น

Devices			Variable					
No	Device name	Device label	No	Name	Variable label	Allowed range	Unit	Type
1	Supply Meter	supply-meter	1	Current	current	0 2	A	Raw
			2	Power	power	0 28	W	Raw
			3	Voltage	voltage	0 24	V	Raw
2	Battery String 01	battery-string-01	1	Current	current	0 2000	mA	Raw
			2	Voltage	voltage	0 8.4	V	Raw
			3	Temperature Battery	temp-bat	20 100	deg C	Raw
			4	Temperature Environment	temp-env	20 100	deg C	Raw
			5	Status	status	0 1	-	Raw

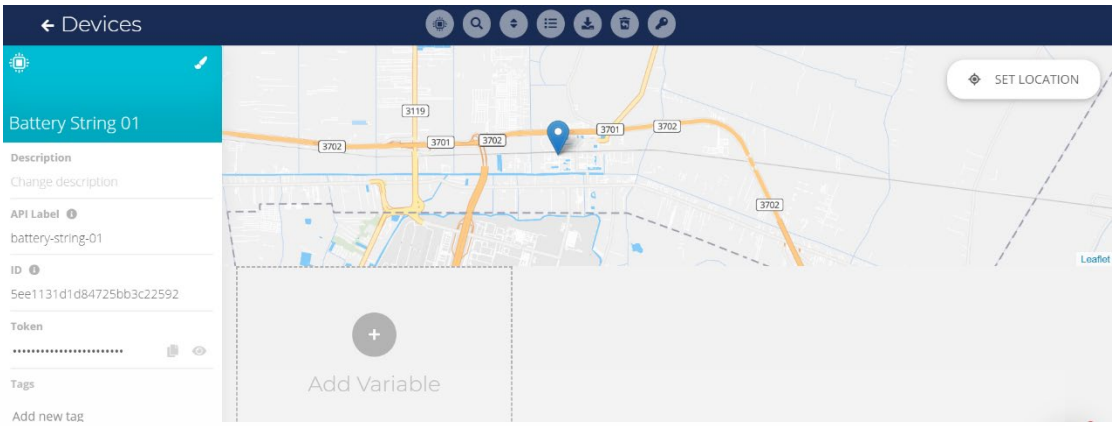
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 (ต่อ)

Devices			Variable						
No	Device name	Device label	No	Name	Variable label	Allowed range		Unit	Type
2	Battery String 01	battery-string-01	6	State of Health	soh	0	100	%	Raw
			7	State of Charge	soc	0	100	%	Synthetic
3	Batter Cell 0101	battery-cell-0101	1	Current	current	0	2000	mA	Raw
			2	Voltage	voltage	0	4.2	V	Raw
			3	State of Health	soh	0	100	%	Raw
			4	State of Charge	soc	0	100	%	Synthetic
4	Batter Cell 0102	battery-cell-0101	1	Current	current	0	2000	mA	Raw
			2	Voltage	voltage	0	4.2	V	Raw
			3	State of Health	soh	0	100	%	Raw
			4	State of Charge	soc	0	100	%	Synthetic

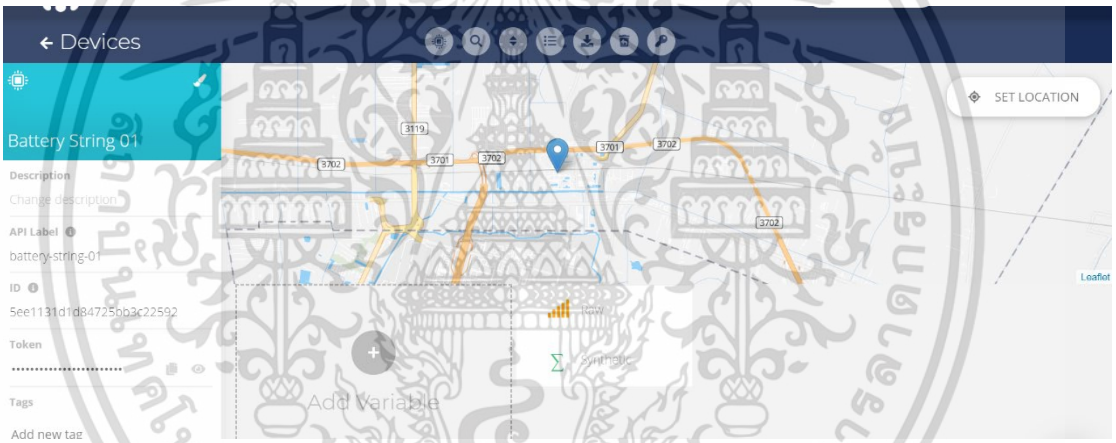
2) เข้าไปที่อุปกรณ์ที่ต้องการสร้างตัวแปร ในรูปที่ 2.68 แสดงการเลือก Battery String 01 ในการสร้างตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.68 แสดงหน้าอุปกรณ์ Battery String 01

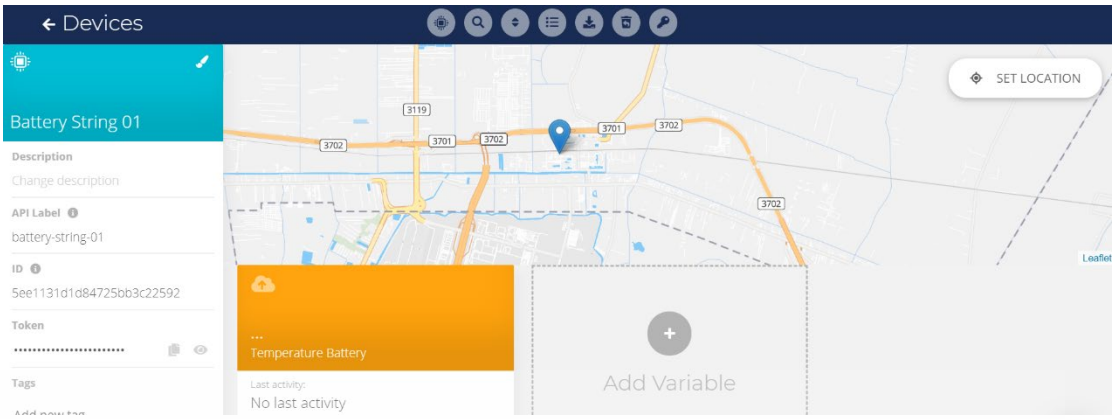
3) สร้างตัวแปรใหม่โดยกด Add Variable โดยในขั้นตอนนี้ผู้ใช้สามารถเลือกชนิดของตัวแปรที่ต้องการได้ (Raw หรือ Synthetic) (รูปที่ 2.69)



รูปที่ 3.69 แสดงการสร้างตัวแปรในอุปกรณ์ Battery String 01

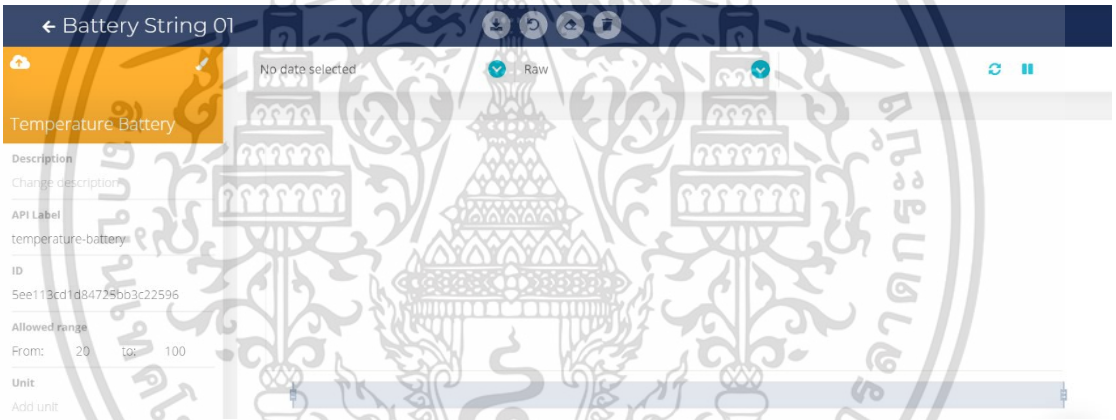
3.1 การสร้างตัวแปรปกติให้เลือกเป็น Raw จากนั้นทำการตั้งชื่อตัวแปร โดยตัวอย่างนี้จะทำการสร้างตัวแปรชื่อ Temperature Battery โดยชื่อที่ตั้งขึ้นจะถูกนำไปใช้การอ้างอิงเพื่อรับค่าจาก Edge computing (รูปที่ 2.70)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



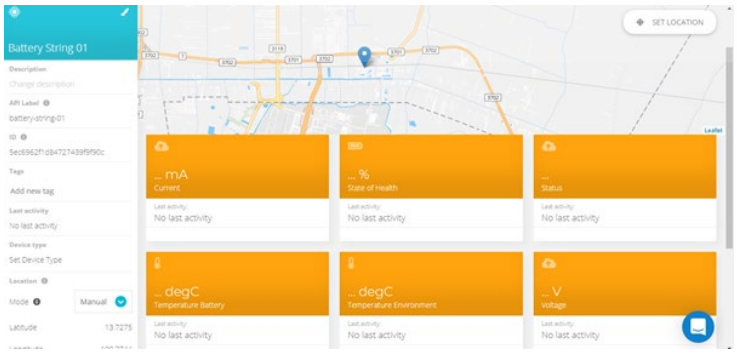
รูปที่ 3.70 แสดงตัวแปร Voltage ที่ทำการสร้างขึ้นในอุปกรณ์ Battery String 01

เข้าไปที่ตัวแปรชื่อ Temperature Battery เพื่อกำหนดรายละเอียดเพิ่มเติมของตัวแปรเช่น Description, Allowed Range, Unit ดังแสดงในรูปที่ 3.71



รูปที่ 3.71 แสดงรายละเอียดของ ตัวแปร Temperature Battery

สร้างตัวแปรในอุปกรณ์ Battery String 01 อื่น ๆ ตามตารางที่ 3.4 เมื่อทำการสร้างตัวแปรครบจะปรากฏหน้าต่างดังรูปที่ 3.72



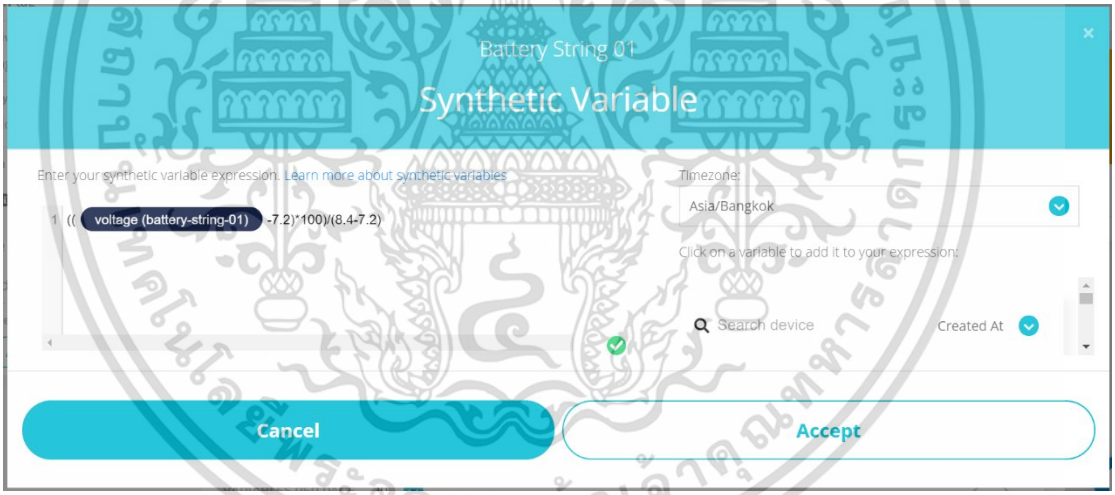
รูปที่ 3.72 แสดงตัวแปรที่ทำการสร้างขึ้นในอุปกรณ์ Battery String 01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การสร้างตัวแปรแบบ synthetic ให้เลือกเป็น synthetic (รูปที่ 2.73)

รูปที่ 3.73 แสดงการสร้างตัวแปรแบบ synthetic ในอุปกรณ์ Battery String 01

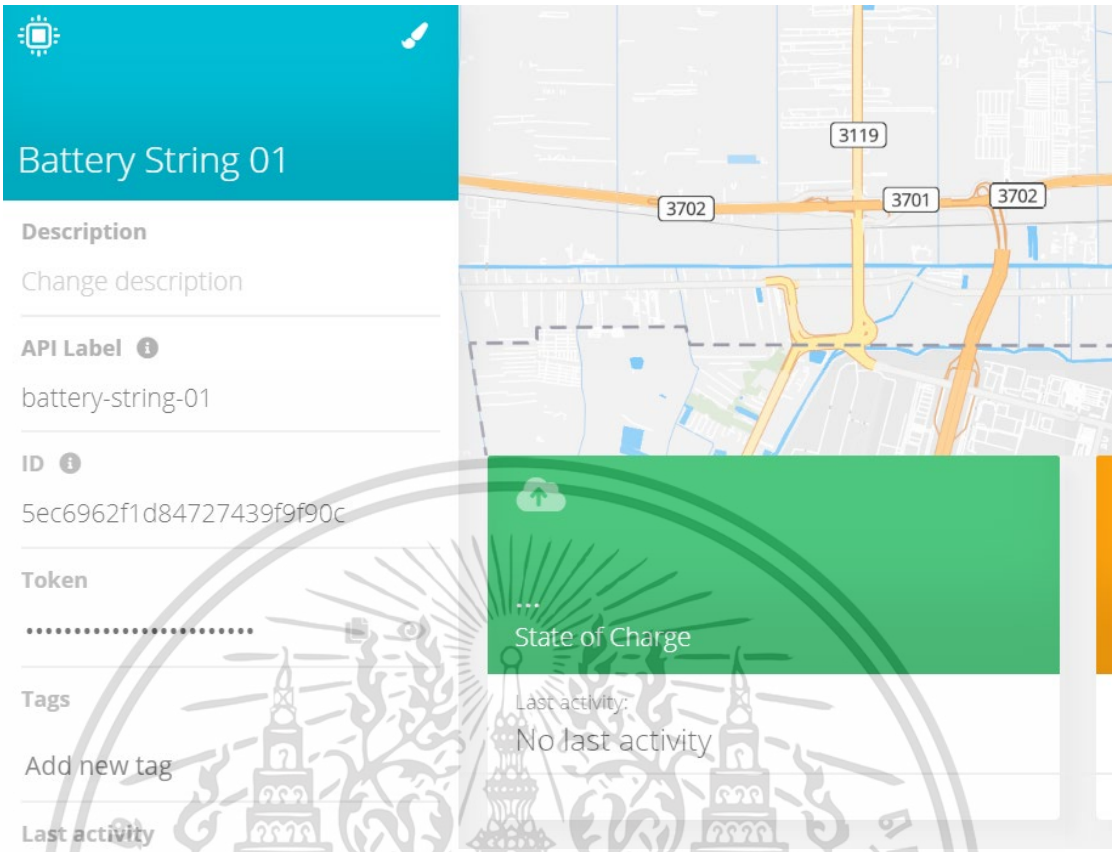
จะปรากฏหน้าต่างดังรูปที่ 3.74 เพื่อสร้างสมการสำหรับคำนวณค่าโดยสามารถใช้ตัวแปรอื่นที่รับมาจาก Edge computing ในสมการคำนวนได้ โดยในตัวอย่างนี้เป็นสมการคำนวณ State of Charge จากนั้นทำการเปลี่ยน Timezone ให้เป็น Asia/Bangkok เพื่อให้ตรงกับ Timezone ปัจจุบัน



รูปที่ 3.74 แสดงการสร้างสมการคำนวณสำหรับตัวแปร State of Charge

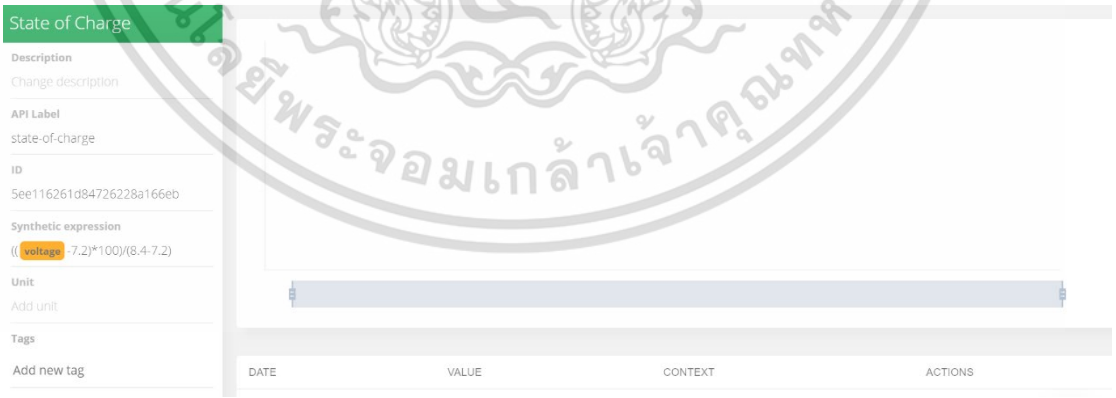
เลือก Accept เพื่อสร้างตัวแปร จากนั้นทำการเปลี่ยนชื่อตัวแปรเป็น State of Charge (รูปที่ 3.75)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.75 แสดงตัวแปร State of Charge ที่ทำการสร้างขึ้นในอุปกรณ์ Battery String 01

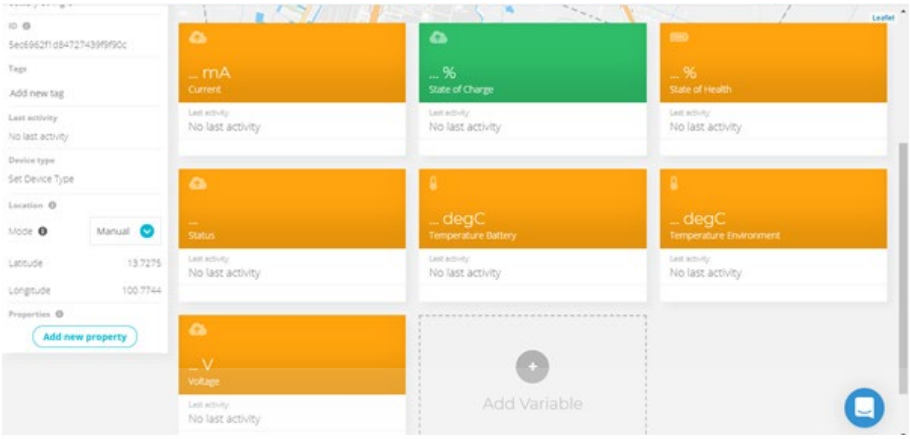
เข้าไปที่ตัวแปรชื่อ State of Charge เพื่อกำหนดรายละเอียดเพิ่มเติมของตัวแปร เช่น API Label, Description, Unit (รูปที่ 3.76)



รูปที่ 3.76 แสดงรายละเอียดของตัวแปร State of Charge

ทำการสร้างตัวแปรในอุปกรณ์อื่น ๆ ตามตารางที่ 3.4 เมื่อทำการสร้างครบทุกตัวแปรในอุปกรณ์ Battery String 01 จะปรากฏดังรูปที่ 3.77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.77 แสดงตัวแปรทั้งหมดที่ทำการสร้างขึ้นในอุปกรณ์ Battery String 01

3.15.3 การสร้างกลุ่มของอุปกรณ์

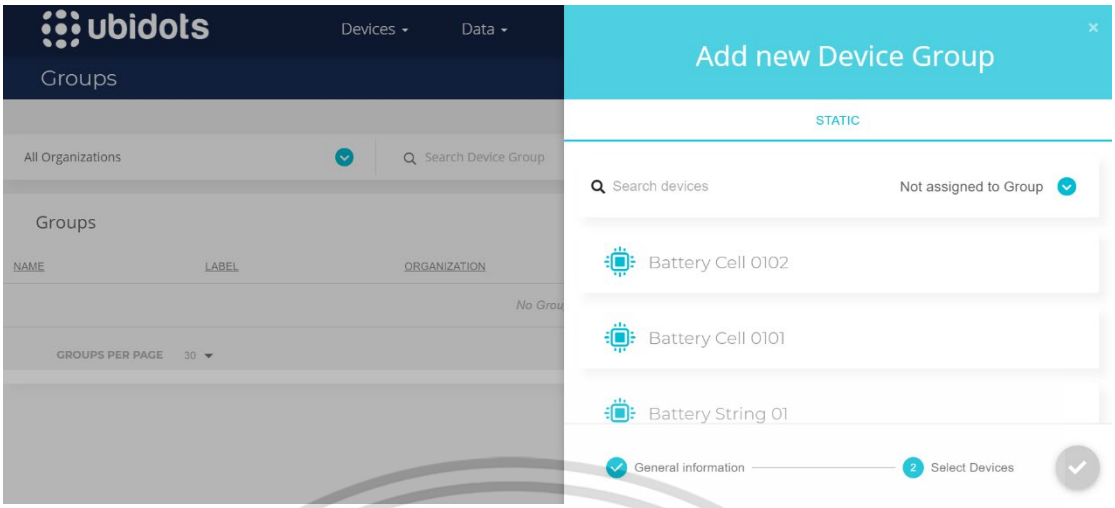
สร้าง Device Group เพื่อเป็นการจัดกลุ่มของอุปกรณ์ โดยขั้นตอนในการสร้าง Device Group มีดังต่อไปนี้

1) เลือก Devices ในแถบเมนูด้านบน เลือก Groups จะปรากฏหน้าต่างดังรูปที่ 3.78 จากนั้นทำการสร้าง Device Group โดยกดที่ Create Device (เครื่องหมาย+

รูปที่ 3.78 แสดงหน้า Groups ของ Ubidots

2) ตั้งชื่อ Device Group และเลือกอุปกรณ์ที่ต้องการจัดกลุ่มจากแถบหน้าต่างที่ปรากฏขึ้นดังรูปที่ 3.79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.79 แสดงการเลือกอุปกรณ์ที่ต้องการจัดกลุ่ม

3) เลือกเครื่องหมายถูกเพื่อสร้าง Device Group เมื่อทำการสร้างเสร็จสิ้นจะปรากฏหน้าต่างดังรูปที่ 3.80

รูปที่ 3.80 แสดง Device Group ที่สร้างขึ้น

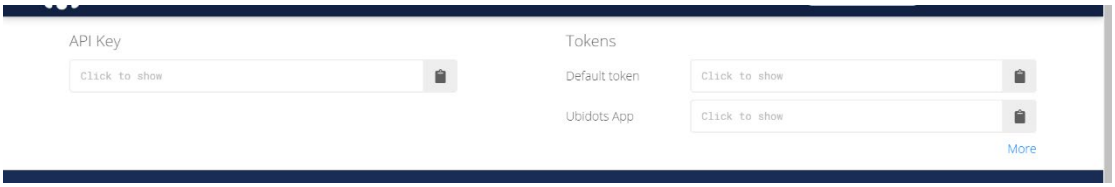
3.16 การรับค่าจาก Edge computing

Ubidots ใช้การเชื่อมต่อกับ Edge computing ผ่านการใช้ Token ตามเนื้อหาที่กล่าวไว้ในหัวข้อ 3.12 โดยมีขั้นตอนในการรับ Token ดังต่อไปนี้

1) กดไปที่เมนูชื่อผู้ใช้งานมุมบนขวาของหน้าต่าง เลือก API Credentials (รูปที่ 3.81)

2) ทำการคัดลอกหมายเลข Default token เพื่อนำไปใช้ในขั้นตอนการรับส่งค่าจาก Edge computing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.81 แสดง API Key และ Tokens ของบัญชี Ubidots

3.17 บันทึกข้อมูลและส่งออกข้อมูลจาก Ubidots

การเก็บข้อมูลใน Ubidots สำหรับตัวทดลองใช้จะสามารถจัดเก็บข้อมูลได้ 1 เดือน

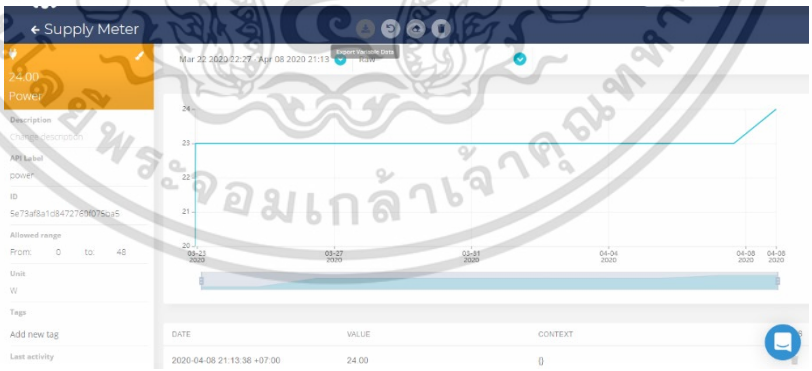
การส่งออกข้อมูลจาก Ubidots สามารถทำได้หลายวิธี แต่ในโครงงานนี้ผู้จัดทำได้เลือกให้ทำการส่งออกข้อมูลในรูปแบบไฟล์ CSV ซึ่งสามารถเลือกข้อมูลที่ต้องการส่งออกได้โดยสามารถเลือกส่งเฉพาะข้อมูลของอุปกรณ์หรือตัวแปรนั้นหรือข้อมูลขององค์กร การส่งออกข้อมูลในรูปแบบไฟล์ CSV มีขั้นตอนดังต่อไปนี้

3.17.1 การส่งออกข้อมูลของตัวแปร

การส่งออกข้อมูลของตัวแปรมีขั้นตอนในการส่งออกทั้งหมด 4 ขั้นตอนโดยแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

1) ไปที่ Device Management –> Devices –> Variables แล้วเลือกตัวแปรที่มีข้อมูลที่ต้องการส่งออก (หากเลือกตัวแปร Power ในอุปกรณ์ Supply Meter จะแสดงดังรูปที่ 8.2)

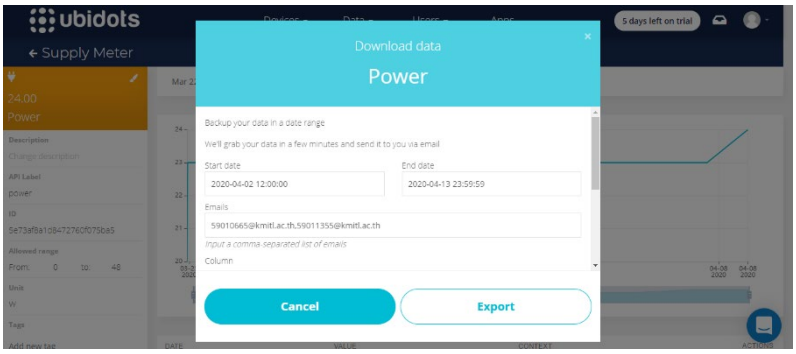
2) คลิกที่ "Export device data to CSV" ที่อยู่บนแถบด้านบน



รูปที่ 3.82 แสดงหน้าตัวแปร Power ในอุปกรณ์ Supply Meter

3) หน้าต่างจะปรากฏขึ้นดังรูปที่ 3.83 เพื่อขออีเมลที่ต้องการส่ง CSV และช่วงเวลาที่จะส่งออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

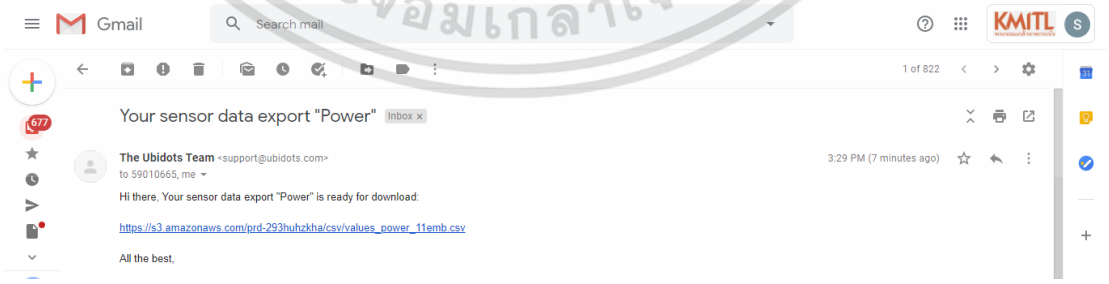


รูปที่ 3.83 แสดงหน้าต่างกำหนดข้อมูลของตัวแปรที่ต้องการส่งออกข้อมูล

4) คลิกที่ "เสร็จสิ้น" เพื่อเสร็จสิ้นกระบวนการ

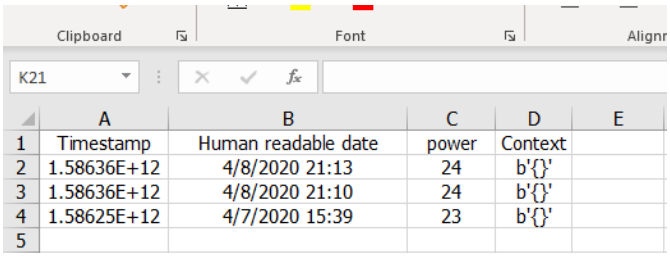
การส่งออกข้อมูลอาจใช้เวลาสักครู่ขึ้นอยู่กับขนาดของชุดข้อมูล หากต้องการติดตามความคืบหน้าในการส่งออกข้อมูลให้คลิกที่ไอคอนการแจ้งเตือนด้านขวาบน (รูปที่ 3.84) ซึ่งผู้ใช้งานจะสามารถดาวน์โหลดไฟล์ CSV ได้ผ่านอีเมลที่ทำการกำหนดไว้ (รูปที่ 3.85) โดยไฟล์ที่ดาวน์โหลดจะมีรูปแบบดังแสดงในรูปที่ 3.86

รูปที่ 3.84 แสดงแถบการแจ้งเตือนของแอปพลิเคชัน



รูปที่ 3.85 แสดงอีเมลสำหรับดาน์โหลดข้อมูลของตัวแปร power ที่ทำการส่งออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



	A	B	C	D	E
1	Timestamp	Human readable date	power	Context	
2	1.58636E+12	4/8/2020 21:13	24	b'{}'	
3	1.58636E+12	4/8/2020 21:10	24	b'{}'	
4	1.58625E+12	4/7/2020 15:39	23	b'{}'	
5					

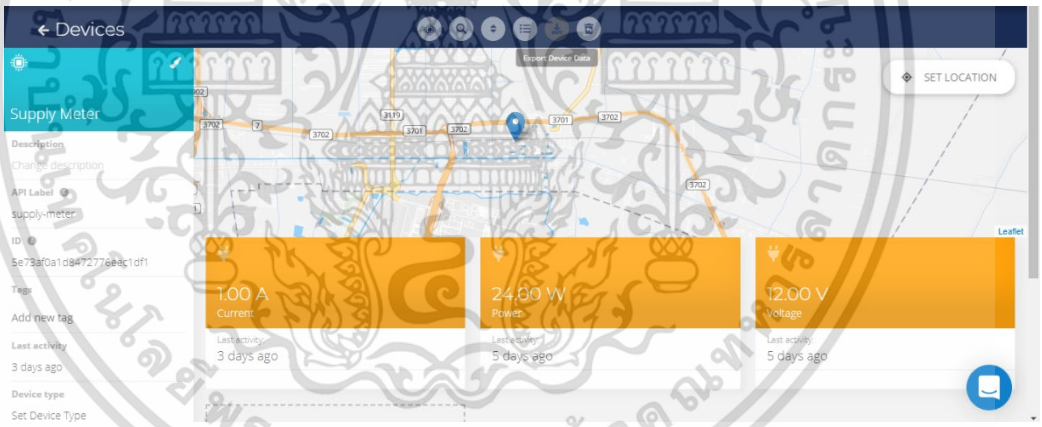
รูปที่ 3.86 แสดงไฟล์ CSV ของข้อมูลของตัวแปร power ที่ทำการส่งออก

3.17.2 การส่งออกข้อมูลของอุปกรณ์

การส่งออกข้อมูลของอุปกรณ์มีขั้นตอนในการส่งออกทั้งหมด 4 ขั้นตอนโดยแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

1) ไปที่ Device Management –> Devices แล้วเลือกอุปกรณ์ที่มีข้อมูลที่ต้องการส่งออก ในกรณีเลือกอุปกรณ์ Supply Meter จะแสดงดังรูป 3.87

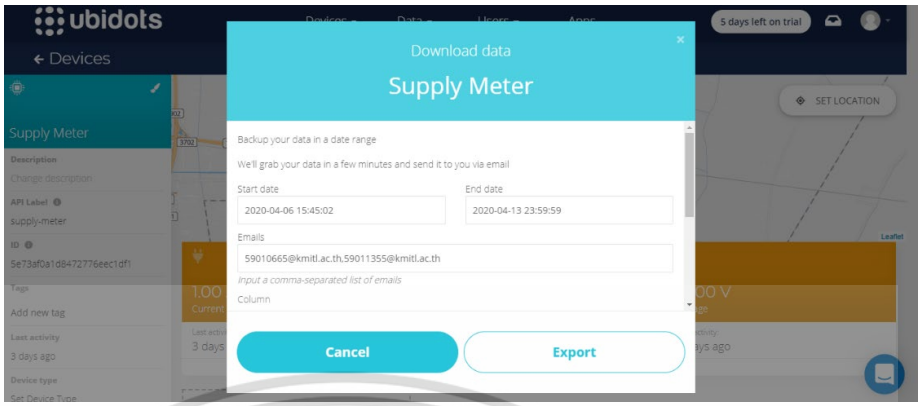
2) คลิกที่ "Export device data to CSV" ที่อยู่บนแถบด้านบน



รูปที่ 3.87 แสดงหน้าอุปกรณ์ Supply Meter

3) หน้าต่างจะปรากฏดังรูปที่ 3.88 เพื่อขออีเมลที่คุณต้องการส่ง CSV และช่วงเวลาที่จะส่งออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.88 แสดงหน้าต่างกำหนดข้อมูลของอุปกรณ์ที่ต้องการส่งออกข้อมูล

4) คลิกที่ "เสร็จสิ้น" เพื่อเสร็จสิ้นกระบวนการ การส่งออกข้อมูลอาจใช้เวลาสักครู่ขึ้นอยู่กับขนาดของชุดข้อมูล หากต้องการติดตามความคืบหน้าในการส่งออกข้อมูลให้คลิกที่ไอคอนการแจ้งเตือนด้านขวาบนซึ่งผู้ใช้งานจะสามารถดาวน์โหลดไฟล์ CSV ได้ โดยรูปแบบของไฟล์ที่ดาวน์โหลดจะแสดงดังรูปที่ 3.89

	Timestamp	Human readable date	current	power	voltage	context_current	context_power	context_voltage
2	1.58653E+12	4/10/2020 22:33	1	-	-	b'{}'	-	-
3	1.58653E+12	4/10/2020 22:15	1	-	-	b'{}'	-	-
4	1.58645E+12	4/9/2020 22:33	1	-	-	b'{}'	-	-
5	1.58645E+12	4/9/2020 22:15	1	-	-	b'{}'	-	-
6	1.58644E+12	4/9/2020 22:00	1	-	-	b'{}'	-	-
7	1.58644E+12	4/9/2020 21:50	1	-	-	b'{}'	-	-
8	1.58644E+12	4/9/2020 21:35	1	-	-	b'{}'	-	-
9	1.58644E+12	4/9/2020 21:20	1	-	-	b'{}'	-	-
10	1.58644E+12	4/9/2020 21:08	1	-	-	b'{}'	-	-
11	1.58644E+12	4/9/2020 20:50	1	-	-	b'{}'	-	-
12	1.58644E+12	4/9/2020 20:35	1	-	-	b'{}'	-	-
13	1.58644E+12	4/9/2020 20:32	1	-	-	b'{}'	-	-
14	1.58636E+12	4/8/2020 21:13	-	-	12	-	-	b'{}'
15	1.58636E+12	4/8/2020 21:13	-	24	-	-	b'{}'	-
16	1.58636E+12	4/8/2020 21:13	1	-	-	b'{}'	-	-
17	1.58636E+12	4/8/2020 21:10	-	-	12	-	-	b'{}'
18	1.58636E+12	4/8/2020 21:10	-	24	-	-	b'{}'	-
19	1.58636E+12	4/8/2020 21:10	1	-	-	b'{}'	-	-
20	1.58625E+12	4/7/2020 15:40	1.7	-	-	b'{}'	-	-
21	1.58625E+12	4/7/2020 15:39	-	-	3.7	-	-	b'{}'
22	1.58625E+12	4/7/2020 15:39	-	23	-	-	b'{}'	-
23	1.58625E+12	4/7/2020 15:39	1.7	-	-	b'{}'	-	-

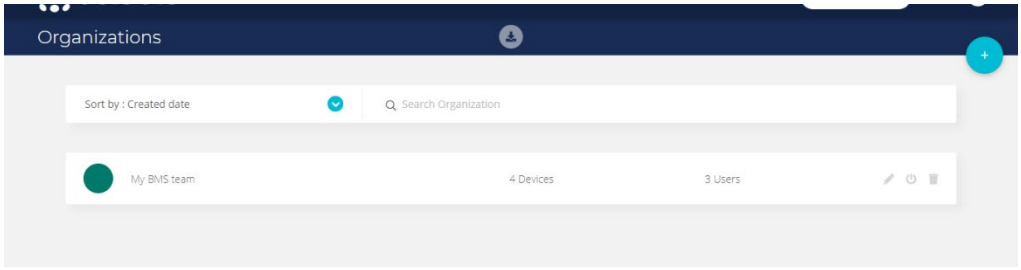
รูปที่ 3.89 แสดงไฟล์ CSV ของข้อมูลของอุปกรณ์ Supply Meter ที่ทำการส่งออก

3.17.3 การส่งออกข้อมูลของข้อมูลองค์กร

การส่งออกข้อมูลขององค์กรมีขั้นตอนในการส่งออกทั้งหมด 3 ขั้นตอนโดยแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

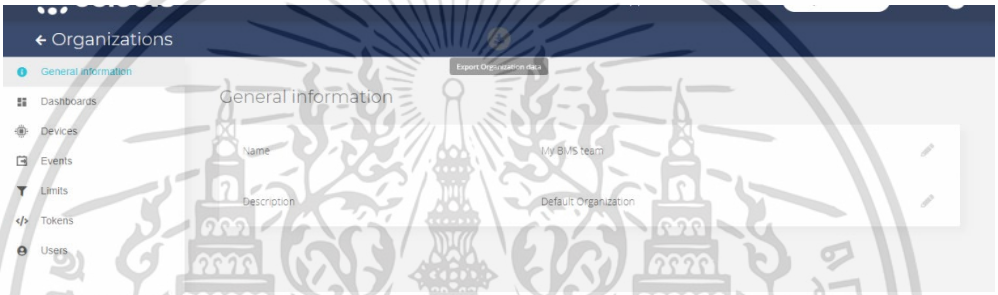
1) ไปที่ User –> Organizations เลือกองค์กรที่ต้องการส่งออกข้อมูล ในโครงงานนี้เลือกองค์กร My BMS team (รูปที่ 3.90)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



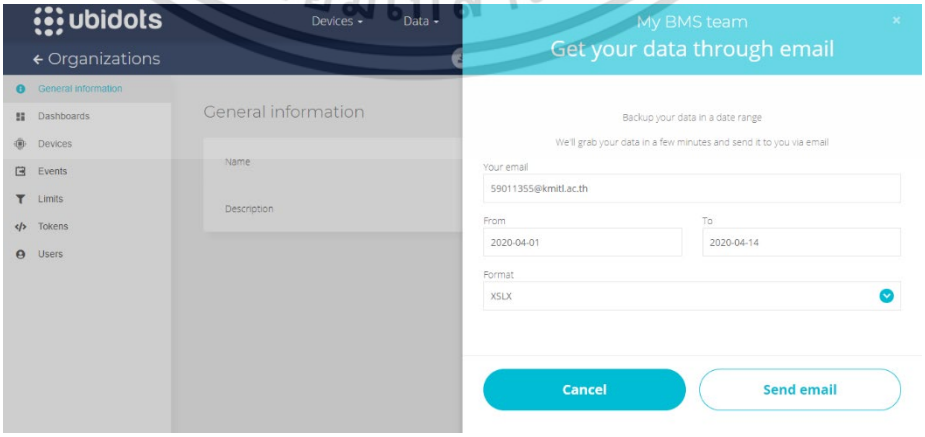
รูปที่ 3.90 แสดงหน้าองค์กร My BMS team

2) คลิกที่ "Export device data to CSV" ที่แถบด้านบนดังที่ปรากฏในรูปที่ 3.91



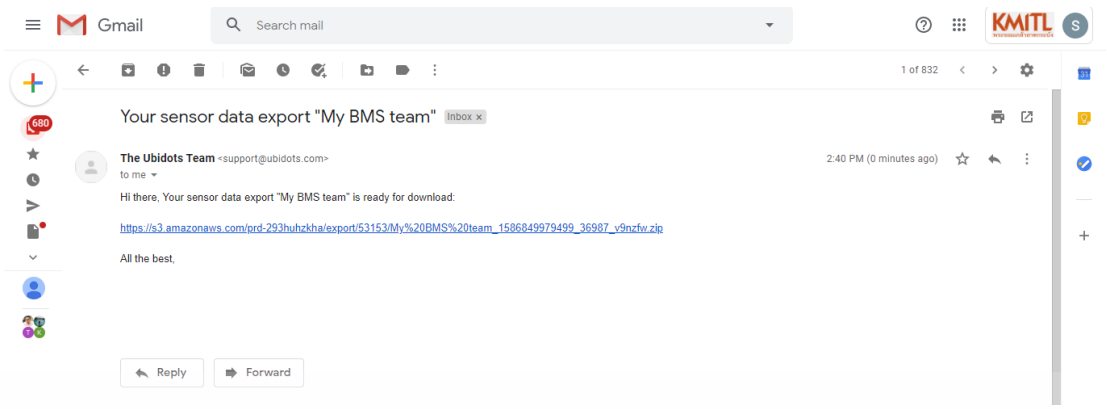
รูปที่ 3.91 แสดงการส่งออกข้อมูลขององค์กร My BMS team

3) จะปรากฏหน้าต่างดังรูปที่ 3.92 สำหรับระบุอีเมลที่ต้องการส่ง, ช่วงเวลาของข้อมูลที่ต้องการและชนิดไฟล์ จากนั้นเลือก Send email เพื่อทำการส่งข้อมูลไปยังอีเมลที่ระบุ โดยอีเมลที่ได้รับจะแสดงดังรูปที่ 3.93 จากนั้นคลิกลิงก์ที่แนบมาเพื่อทำการดาวน์โหลดไฟล์.xlsx ซึ่งไฟล์จะถูกรวบรวมใน Zip ไฟล์ดังรูปที่ 3.94

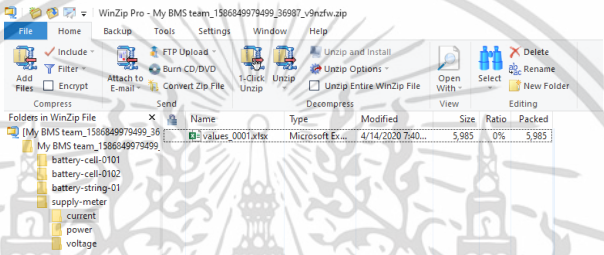


รูปที่ 3.92 แสดงหน้าต่างสำหรับกำหนดการส่งออกข้อมูลขององค์กร My BMS team

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.93 แสดงอีเมลสำหรับดาน์โหลดข้อมูลขององค์กร My BMS team ที่ทำการส่งออก

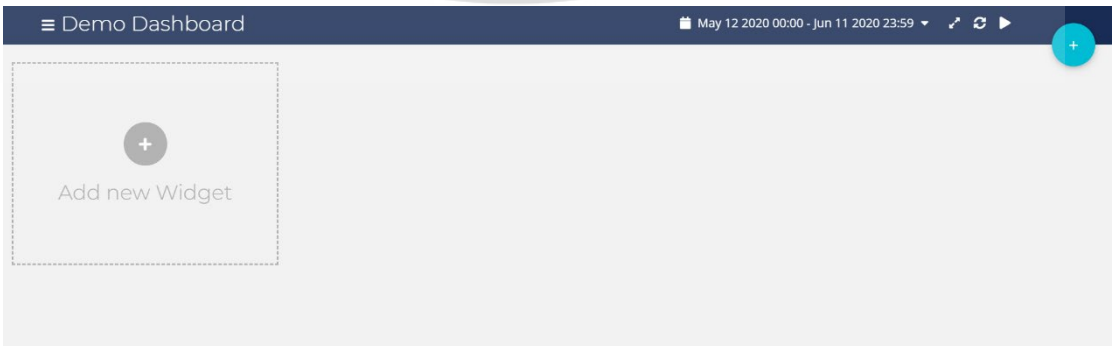


รูปที่ 3.94 แสดง Zip ไฟล์ขององค์กร My BMS team

3.18 การสร้างหน้าแสดงผล (Dashboard ใน Ubidots

Dashboard เป็นตัวแสดงผลค่าที่เก็บไว้ในตัวแปรต่าง ๆ ซึ่ง Dashboard ของ Ubidots สามารถทำการเก็บค่าและแสดงผลได้ในตัวเดียวและสามารถตั้งเวลาเพื่อแสดงข้อมูลย้อนหลัง โดยขั้นตอนในการสร้าง Dashboard สามารถทำได้ดังต่อไปนี้

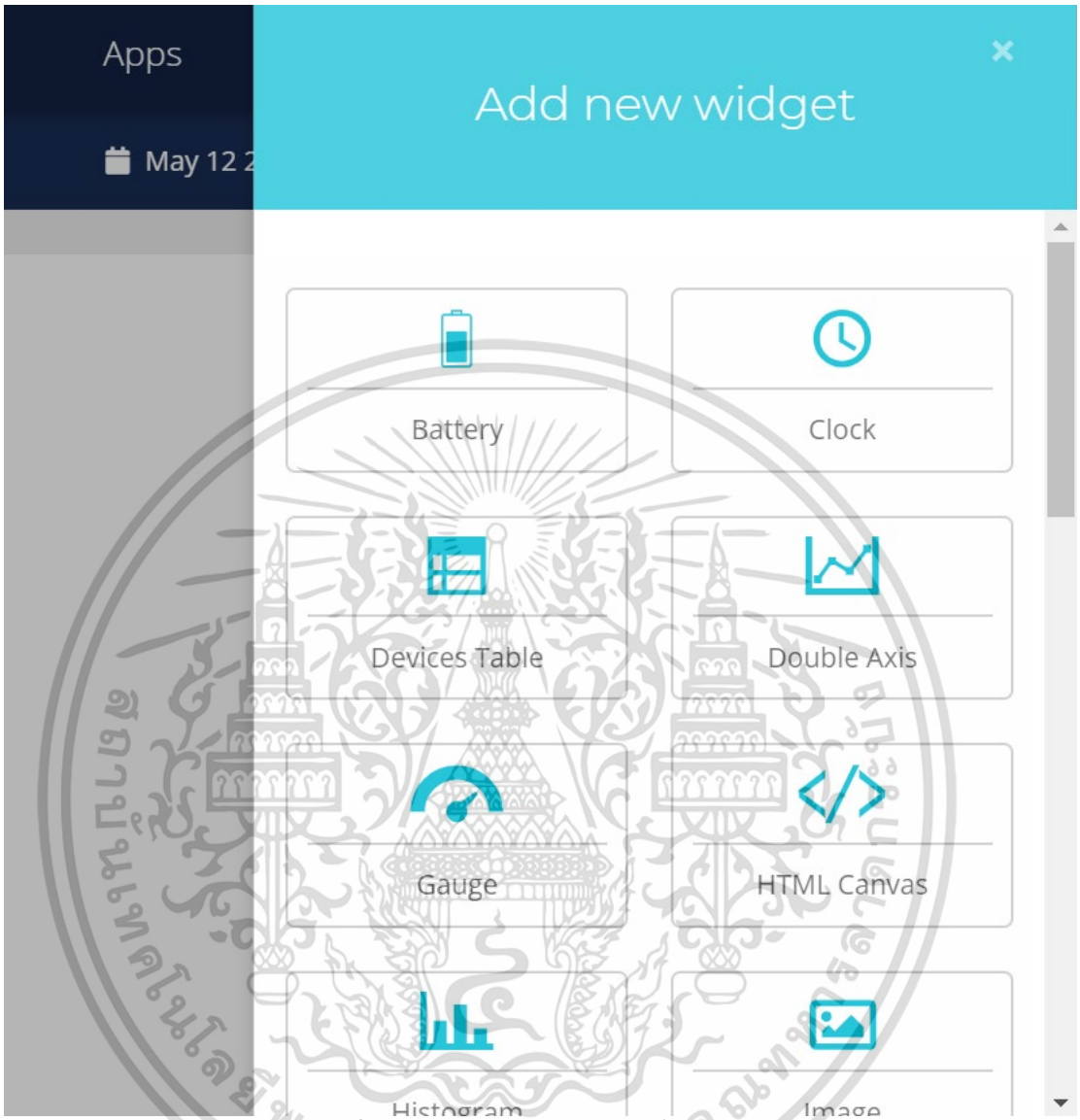
1) เลือก Data ในแถบเมนูด้านบน เลือก Dashboard ในขั้นตอนนี้สามารถดูได้ว่ามี Dashboard ถูกสร้างไว้แล้วหรือไม่ (Demo Dashboard) หากไม่มีให้ทำการสร้าง Dashboard ขึ้นมา โดยคลิกที่เครื่องหมาย + ดังแสดงในรูปที่ 3.95



รูปที่ 3.95 แสดงหน้า Demo Dashboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ทำการสร้าง Widget ใน dashboard โดยกดที่ Add new Widget (รูปที่ 3.96



รูปที่ 3.96 แสดงชนิด Widget ที่ให้บริการ

3) กำหนดชนิดของ Widget และรายการที่ต้องการแสดงดังตารางที่ 3.5

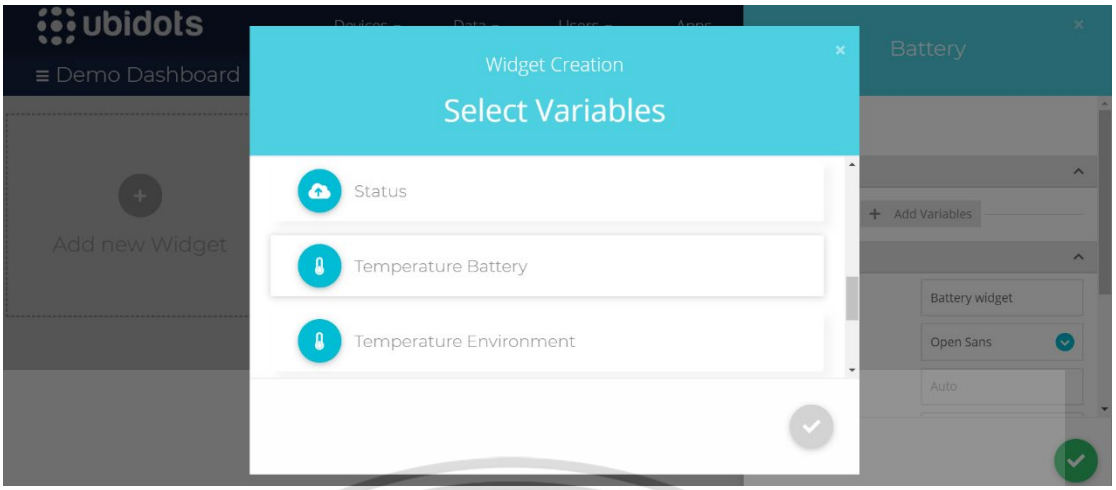
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 แสดง Widget และรายการที่ต้องการแสดง

ลำดับที่	ชนิด	รายละเอียดที่ต้องการแสดง	คำอธิบายเพิ่มเติม
1	Text	Instrumentation Engineering | Kmitl, BMS on Cloud	แสดงชื่อโปรเจคและภาควิชา
2	Thermometer	Temperature Battery	แสดงอุณหภูมิของแบตเตอรี่
3	Thermometer	Temperature Environment	แสดงอุณภูมิของสิ่งแวดล้อม
4	Bar Chart	State of Charge : Battery 01	แสดงสถานะการชาร์จของแบตเตอรี่ 01 (String01, Cell0101, Cell0102)
5	Line Chart	Voltage : Battery 01	แสดงแรงดันที่เวลาใด ๆ ของแบตเตอรี่ 01 (String01, Cell0101, Cell0102)
6	Line Chart	Current : Battery 01	แสดงกระแสที่เวลาใด ๆ ของแบตเตอรี่ 01 (String01, Cell0101, Cell0102)
7	Battery	State of Health : Battery String 01	แสดงสุขภาพของแบตเตอรี่ String01
8	Battery	State of Health : Battery Cell 0101	แสดงสุขภาพของแบตเตอรี่ Cell0101
9	Battery	State of Health : Battery Cell 0102	แสดงสุขภาพของแบตเตอรี่ Cell0102
10	Double Axis	System Energy Consumption	แสดงการใช้พลังงานของระบบที่ Supply Meter ซึ่งประกอบด้วยค่ากระแส, แรงดันและกำลัง

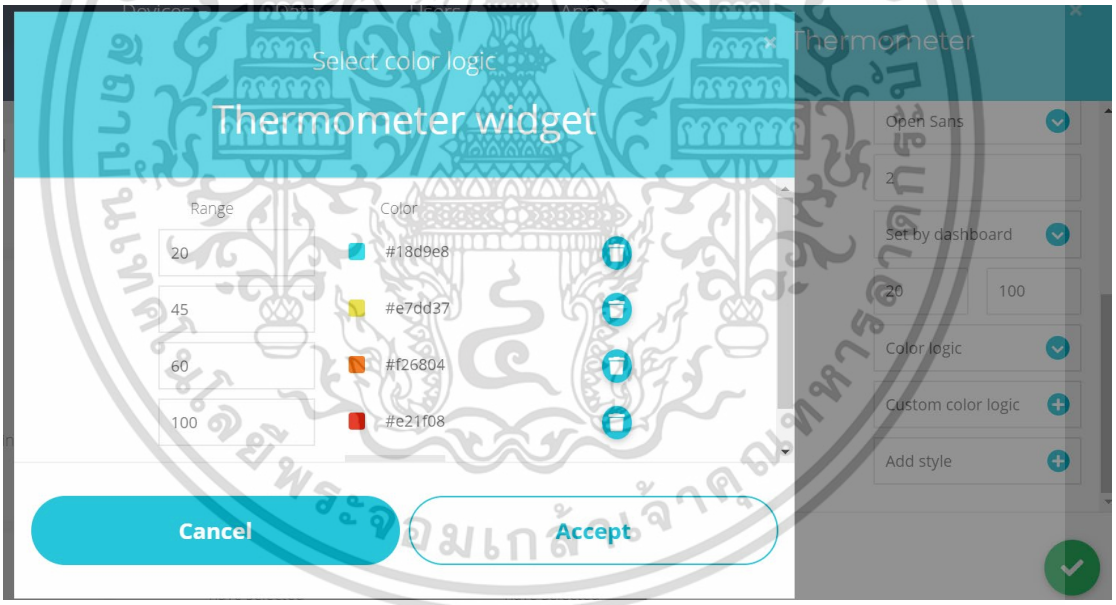
4) โดยในตัวอย่างนี้จะเลือกใช้ Widget ชนิด Thermometer เพื่อใช้แสดงอุณหภูมิของแบตเตอรี่ จากนั้นทำการเลือกอุปกรณ์ Battery String 01 เลือกตัวแปร Temperature Battery เพื่อให้ Thermometer Widget เชื่อมกับตัวแปร Temperature Battery ดังรูปที่ 3.97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.97 แสดงการเชื่อม Thermometer Widget กับ ตัวแปร Temperature Battery

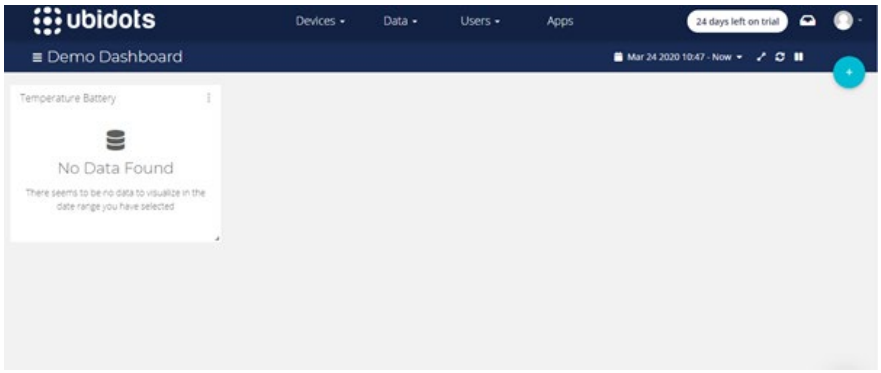
5) กำหนดรายละเอียดของ Widget เช่น ชื่อ Widget, ค่า Min value และ Max value, Color logic จากนั้นเลือก Finish (รูปที่ 3.98



รูปที่ 3.98 แสดงการตั้งค่า Thermometer Widget

6) Thermometer Widget เชื่อมต่อกับตัวแปร Temperature Battery ของอุปกรณ์ Battery String 01 ใน Demo Dashboard และหากมีการส่งค่ามาที่ตัวแปร Temperature Battery ด้วย Edge computing Thermometer Widget จะแสดงผลทันที หากยังไม่มีการเชื่อมต่อจะแสดงดังรูปที่ 3.99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

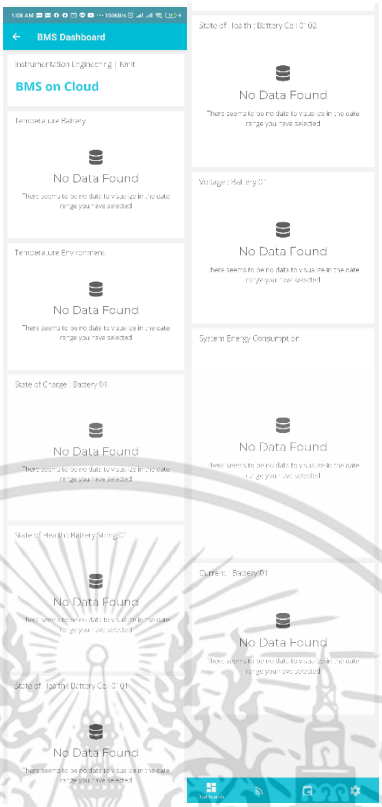


รูปที่ 3.99 แสดง Thermometer Widget ขณะยังไม่มีการส่งข้อมูล

7) จากนั้นให้ทำการสร้าง Widget เพิ่มเติมดังตารางที่ 3.5 หากใช้งานบน Personal computer จะได้ผลลัพธ์ดังรูปที่ 3.100 และหากใช้งานบน Smart phone จะได้ผลลัพธ์ดังรูปที่ 3.101

รูปที่ 3.100 แสดง BMS Dashboard ขณะยังไม่มีการส่งข้อมูลบน Personal computer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.101 แสดง BMS Dashboard ขณะยังไม่มีการส่งข้อมูลบน Smart phone

3.19 การสร้างการแจ้งเตือนอัตโนมัติใน Ubidots

ทำการสร้าง Events เพื่อให้ระบบสามารถส่งแจ้งเตือนรูปแบบต่าง ๆ ได้อย่างอัตโนมัติ โดยผู้จัดทำได้ออกแบบให้ระบบทำการแจ้งเตือนอยู่ 3 กรณีดังนี้

a) State of Health ≤50%

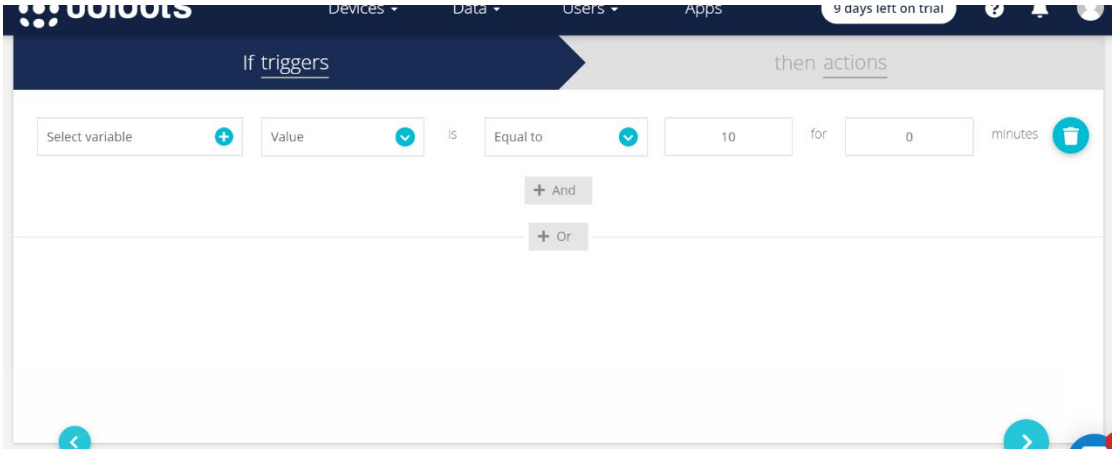
b) อุณหภูมิขณะชาร์จ > 45 degC หรือ อุณหภูมิขณะดิสชาร์จ > 60 degC

c) อุปกรณ์ขาดการเชื่อมต่อนานกว่า 15 นาที

โดยขั้นตอนในการสร้างการแจ้งเตือนอัตโนมัติสามารถทำได้ดังนี้

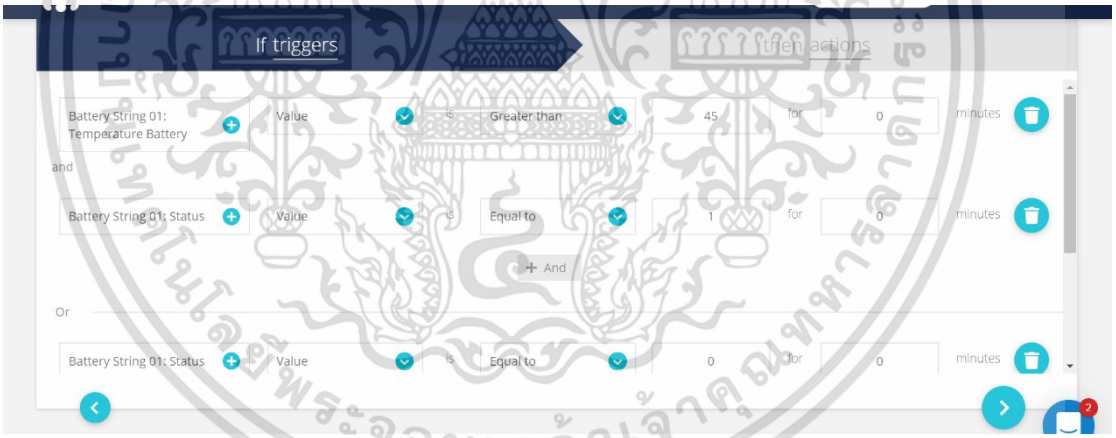
เลือก Data ในแถบเมนูด้านบน เลือก Events ทำการสร้างการแจ้งเตือนโดยกดที่ Create Events (เครื่องหมาย+ จะปรากฏหน้าต่างดังรูป ที่ 3.102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.102 แสดงหน้าต่างการสร้างแจ้งเตือน

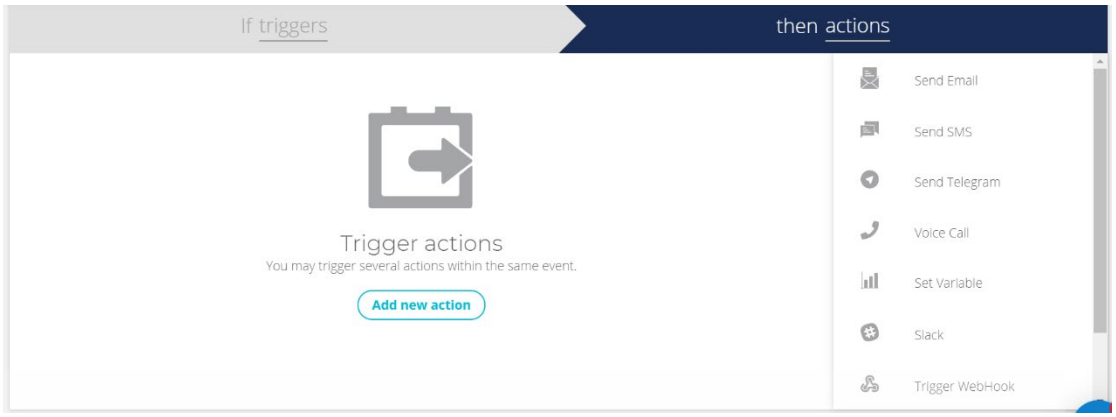
เลือก Device หรือ Device Group และ Variable ที่ต้องการตั้งการแจ้งเตือนและเลือกเงื่อนไขในการแจ้งเตือน เช่น ในตัวอย่างนี้กำหนดให้แจ้งเตือนเมื่ออุณหภูมิขณะชาร์จ > 45 degC หรือ อุณหภูมิขณะดิสชาร์จ > 60 degC (กำหนดให้สถานะชาร์จตัวแปร Status มีค่าเท่ากับ 1 และมีค่าเท่ากับ 0 เมื่ออยู่ในสถานะ ดิสชาร์จ (รูปที่ 3.103)



รูปที่ 3.103 แสดงตัวแปรและเงื่อนไขในการแจ้งเตือนอุณหภูมิของแบตเตอรี่ที่สถานะต่าง ๆ

เลือกเงื่อนไขว่าจะให้ส่งรูปการแจ้งเตือนในรูปแบบใด เช่น Send e-mail, Send SMS จากนั้นทำการตั้งค่าผู้รับและลักษณะที่ต้องการแจ้งเตือน แล้วกด Finish ในตัวอย่างนี้เลือกทำกรแจ้งเตือนผ่านอีเมล (รูปที่ 3.104 จากนั้นทำการกำหนดอีเมลและข้อความที่ต้องการส่ง (รูปที่ 3.105

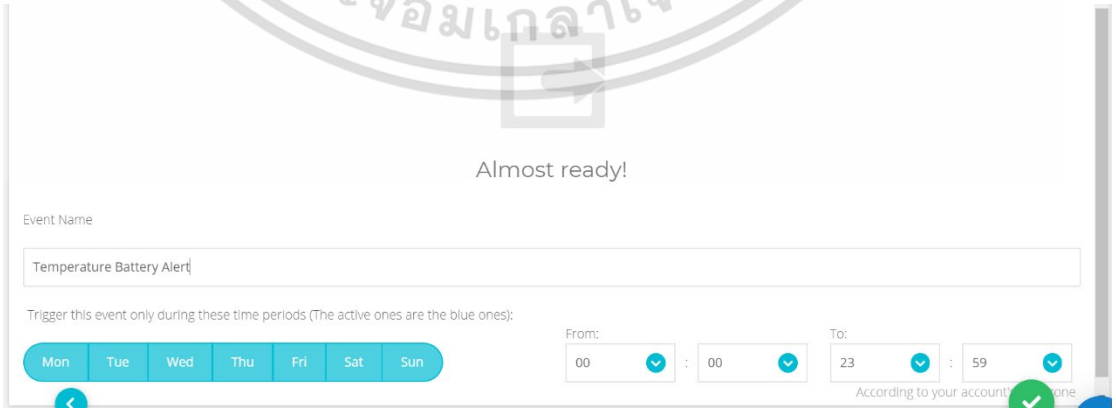
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.104 แสดงรายการรูปแบบการแจ้งเตือนที่ให้บริการใน Ubidots

รูปที่ 3.105 แสดงรายละเอียดการแจ้งเตือนแบบ e-mail

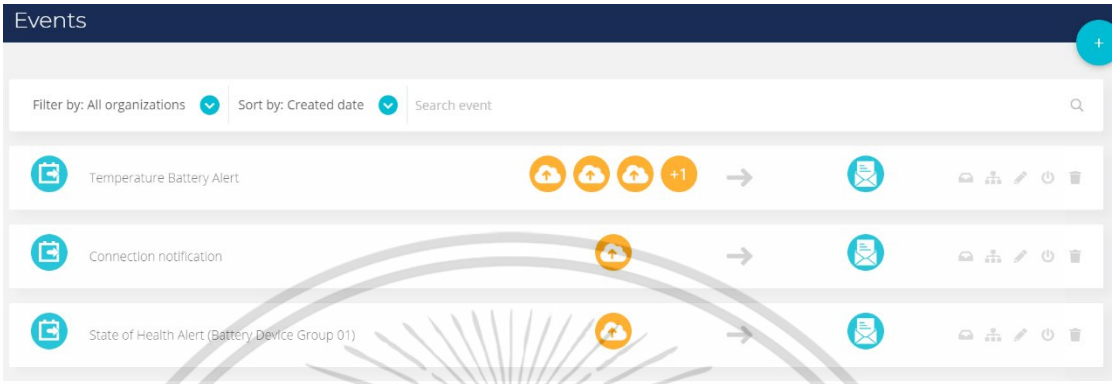
กำหนดชื่อการแจ้งเตือนและช่วงเวลาที่ต้องการให้การแจ้งเตือนทำงาน ในรูปที่ 3.106 กำหนดให้ทำการแจ้งเตือนแบบ 24/7 เลือกเครื่องหมายถูก เป็นขั้นตอนสุดท้ายของการสร้างการแจ้งเตือนอุณหภูมิของแบตเตอรี่ที่สถานะต่าง ๆ



รูปที่ 3.106 แสดงชื่อและช่วงเวลาที่กำหนดให้การแจ้งเตือนทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำสร้างการแจ้งเตือนอื่น ๆ เพิ่มเติมตามที่ได้ทำการออกแบบไว้จนครบทั้ง 3 กรณีจะแสดงดังรูปที่ 3.107



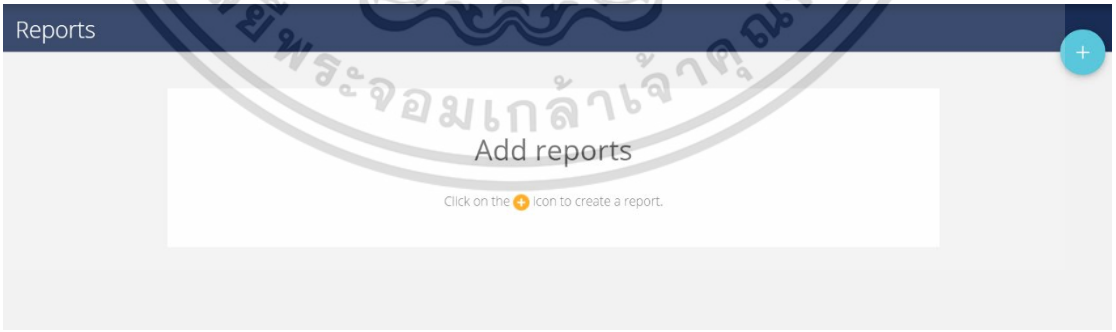
รูปที่ 3.107 แสดงรายการแจ้งเตือนอัตโนมัติทั้งหมดที่ถูกสร้างขึ้น

และหากตัวแปรที่เลือกเป็นไปตามเงื่อนไขที่ตั้งเอาไว้ Ubidotsจะส่งการแจ้งเตือนตามที่ตั้งค่าไว้

3.20 การสร้างหน้ารายงานอัตโนมัติ (Report) ใน Ubidots

รายงานของ Ubidots สามารถกำหนดผู้รับได้หลายคน, กำหนดระยะเวลาข้อมูลย้อนหลัง, กำหนดระยะเวลาในการส่งอีเมล โดยมีขั้นตอนการสร้างรายงานอัตโนมัติดังนี้

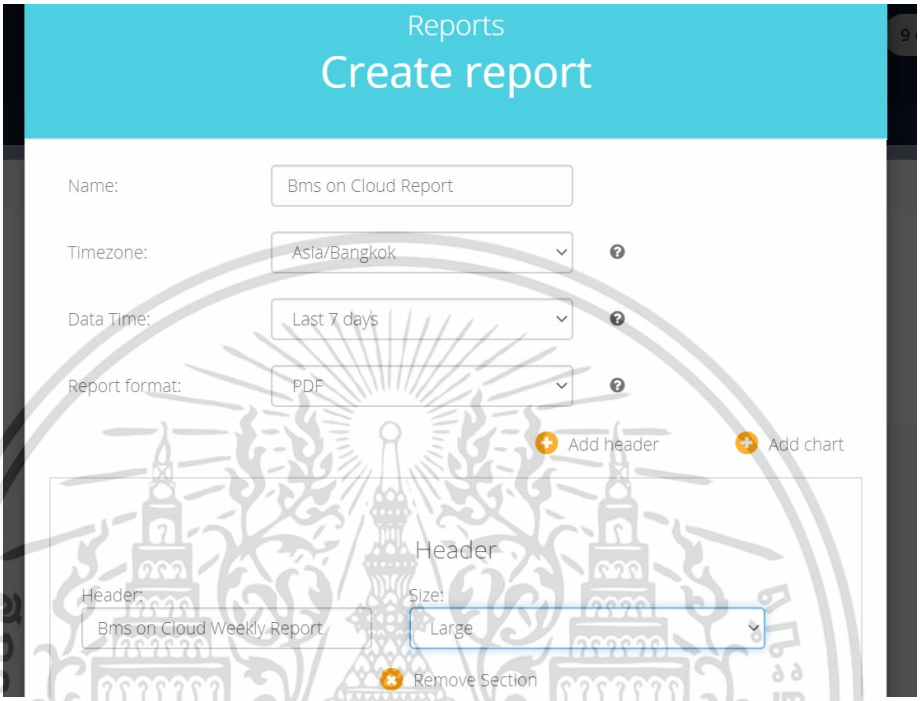
1) เลือก Data ในแถบเมนูด้านบน เลือก Report ทำการสร้างรายงานโดยกดที่ Add Report (เครื่อหมาย+) จะปรากฏหน้าต่างดังรูป 3.108



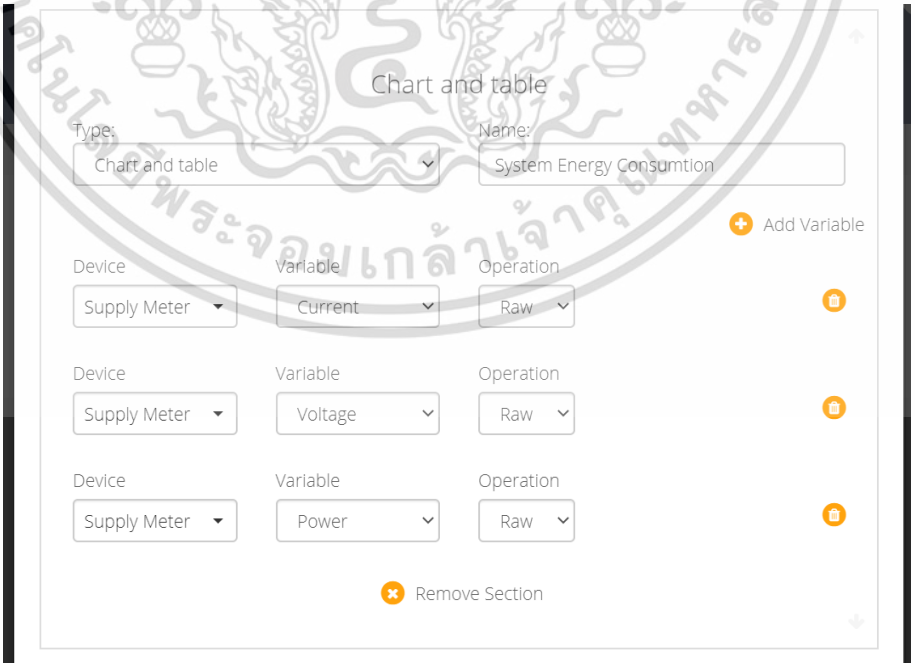
รูปที่ 3.108 แสดงหน้าต่างการสร้างรายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) กำหนดรายละเอียดและรูปแบบของรายงานอัตโนมัติที่ต้องการสร้าง เช่น ชื่อรายงาน, รูปแบบรายงาน, ผู้รับ, ความถี่และระยะเวลาในการส่ง, แผนภูมิและตารางที่ต้องการแสดง (รูปที่ 3.109 - 3.111

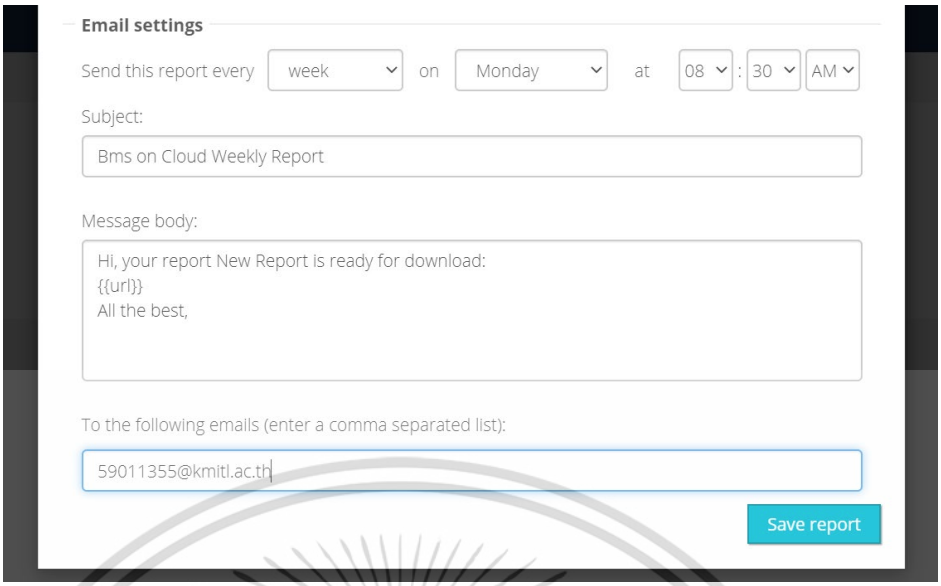


รูปที่ 3.109 แสดงตัวอย่างการตั้งค่าชื่อและรูปแบบของรายงาน



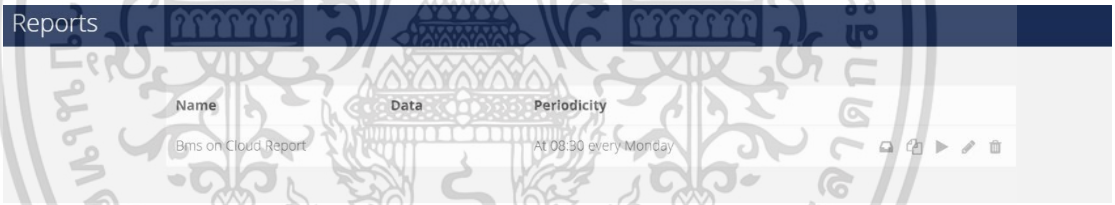
รูปที่ 3.110 แสดงตัวอย่างการตั้งค่าแผนภูมิและตารางที่ต้องการแสดงในรายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.111 แสดงตัวอย่างการตั้งค่าระยะเวลาในการส่ง, ข้อความและผู้รับ

3) เมื่อทำการกำหนดรายละเอียดของรายงานแล้ว เลือก Save report จะปรากฏหน้าต่างดังรูปที่ 3.112



รูปที่ 3.112 แสดงรายการรายงานอัตโนมัติที่สร้างขึ้น

3.21 การเพิ่มองค์กร, บทบาท, ผู้ใช้งานใน Ubidots

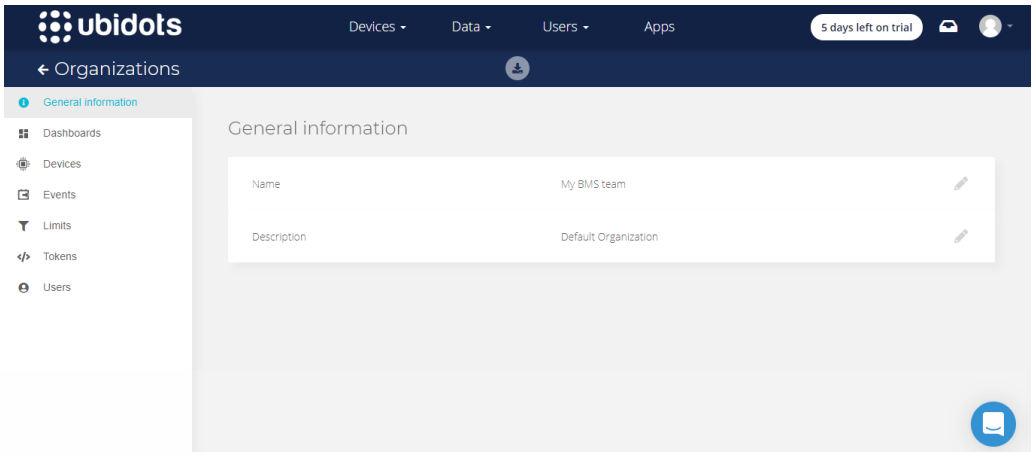
การเพิ่มองค์กร, บทบาท, ผู้ใช้งานใน Ubidots มีขั้นตอนดังต่อไปนี้

1) การสร้างองค์กร

องค์กรเป็นส่วนย่อยของแอพ Ubidots ในโดยผู้ใช้จะมีสิทธ์การเข้าถึง Dashboards, Devices, Events และ API Tokens ที่แตกต่างกัน

สร้างองค์กรในแอปพลิเคชันโดยเลือก Organizations จากแท็บ Users ในแถบนำทางหลัก เพิ่มชื่อและคำอธิบายให้กับองค์กรที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.113 แสดงองค์กร My BMS team ที่สร้างขึ้น

2 การสร้างบทบาท (Roles)

เลือก Roles จากแท็บผู้ใช้ในแถบนำทาง คลิกไอคอน + ที่มุมด้านขวาบนเพื่อเพิ่มกลุ่มบทบาทใหม่โดยสามารถสร้างบทบาทใหม่เองหรือใช้บทบาทเริ่มต้นของ Ubidots โดยแต่ละบทบาทจะมีความสามารถดังนี้

Explorers: สามารถสร้าง แก้ไข ดูและลบ dashboards และ events ต่าง ๆ สามารถดูอุปกรณ์ตัวแปรและค่าต่าง ๆ แต่ไม่สามารถแก้ไขหรือลบอุปกรณ์หรือตัวแปรได้

Viewers: สามารถดู dashboards, อุปกรณ์, ตัวแปรและค่าได้ แต่ไม่สามารถสร้างแก้ไขหรือลบได้

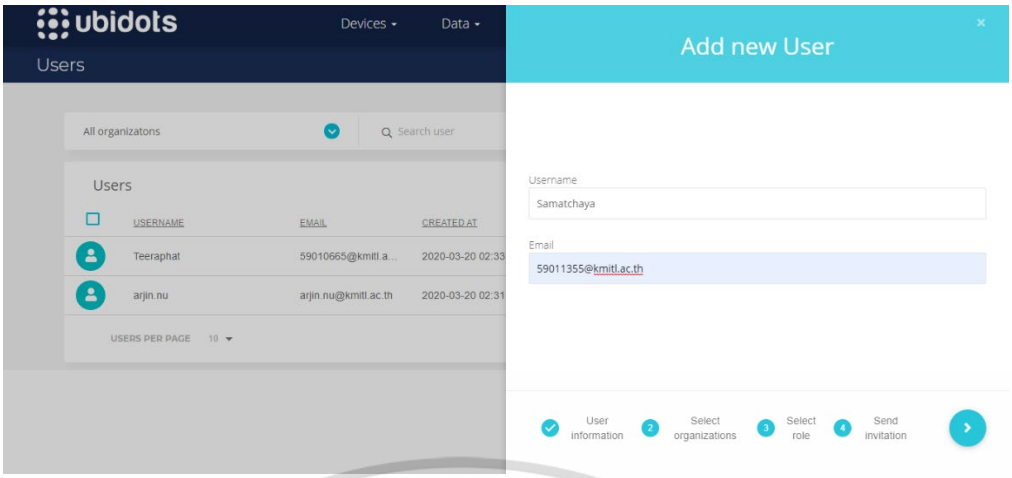
Dashboard viewers: สามารถดูแดชบอร์ดข้อมูลขององค์กรของผู้ใช้เท่านั้น

Managers: สามารถทำทุกอย่างที่ผู้ดูแลระบบสามารถทำได้ยกเว้นการสร้างแก้ไขหรือลบผู้ใช้ปลายทาง

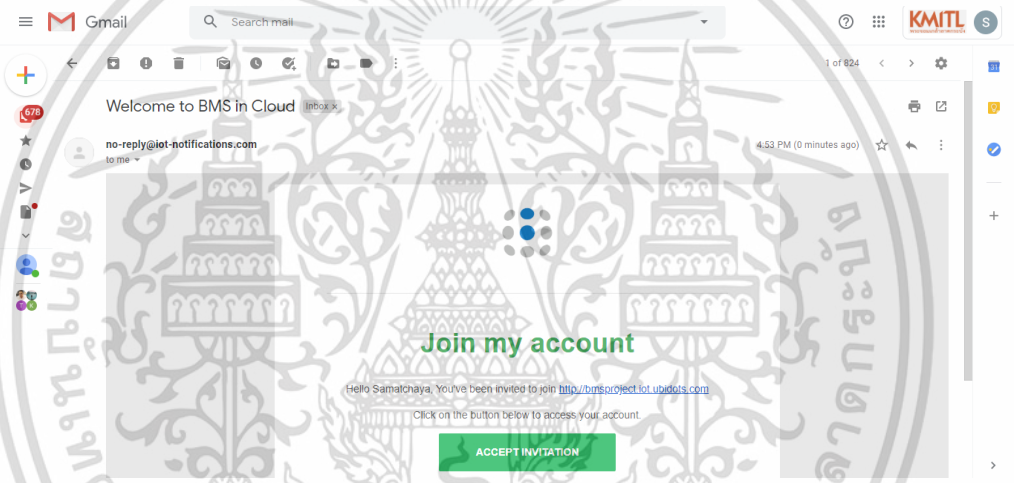
3) การเพิ่มผู้ใช้งาน

สามารถทำได้โดยเลือก Users จากแท็บ Users ในแถบนำทางหลัก เพิ่มชื่อ, อีเมล เลือกองค์กรและบทบาท และเลือกส่งอีเมลเพื่อเชิญผู้ใช้งานใหม่ (รูปที่ 3.114 โดยอีเมลที่ส่งจะแนบ ลิงก์สำหรับตั้งรหัสผ่านเพื่อเข้าสู่ระบบ (รูปที่ 3.115 และเมื่อผู้ใช้งานใหม่เข้าสู่ระบบ แอปพลิเคชันจะแสดงหน้า Dashboard ของผู้ใช้งานตามบทบาทที่ถูกกำหนด (รูปที่ 3.116

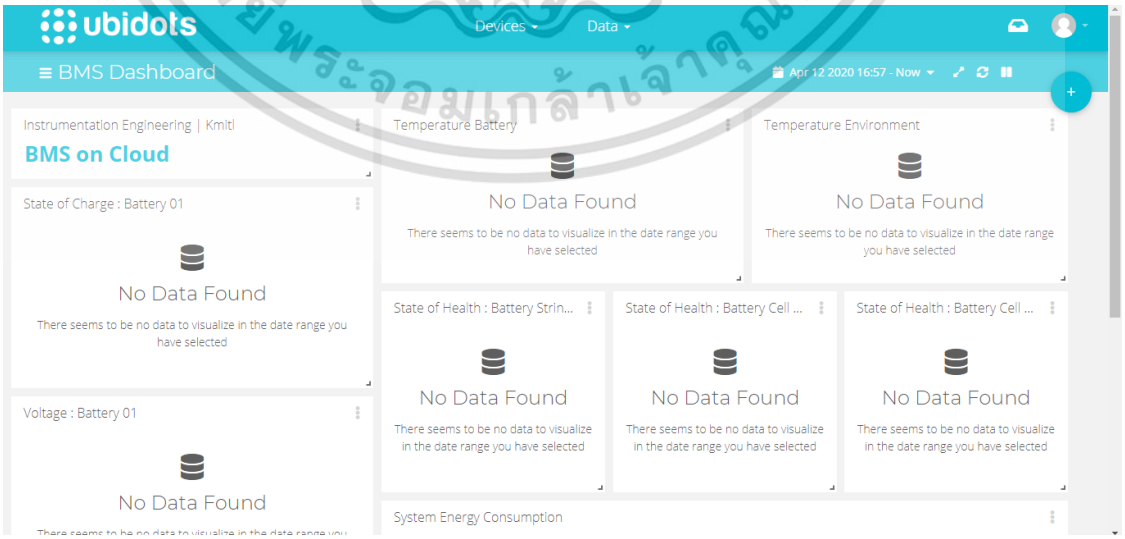
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.114 แสดงหน้าต่างการเพิ่มผู้ใช้งาน



รูปที่ 3.115 แสดงอีเมลเพื่อเชิญผู้ใช้งานใหม่เข้าสู่ระบบ



รูปที่ 3.116 แสดงหน้า Dashboard ของผู้ใช้ในบทบาท Explorers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.22 การปรับแต่งแอปพลิเคชัน

การปรับแต่งแอปพลิเคชันมีจุดประสงค์เพื่อทำให้แอปพลิเคชันนั้น ๆ มีเอกลักษณ์เฉพาะตัวโดยสามารถทำการปรับแต่งแอปพลิเคชันได้ 3 ส่วนด้วยกันดังนี้

1) ส่วน App Domain

การปรับแต่ง App Domain สามารถทำได้โดยไปที่แท็บ App (รูปที่ 3.117 จากนั้นคลิกที่ Manage App

รูปที่ 3.117 แสดงหน้าต่างเมนู App

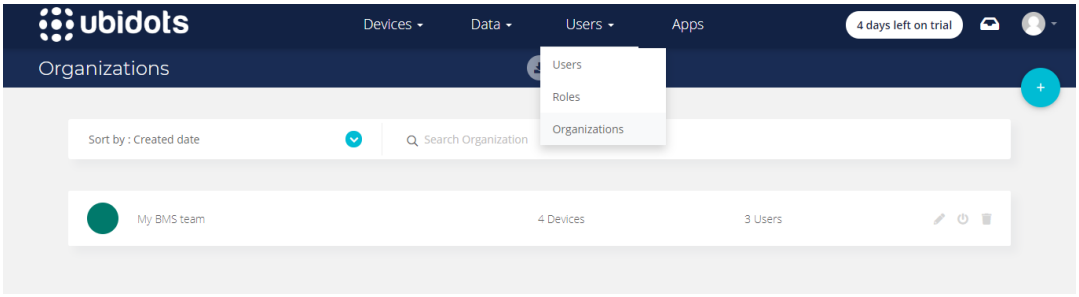
เลือก Domains และพิมพ์ชื่อ domain ที่ต้องการในช่อง Custom Domain หรือเลือกใช้ Domain ที่ Ubidots กำหนดให้ ในโครงงานนี้เลือกใช้ Domain ที่ Ubidots กำหนดให้ซึ่งคือ bmsproject.iot.ubidots.com (รูปที่ 3.118)

รูปที่ 3.118 แสดงหน้า Domains

2) ส่วน Organization domain

การกำหนดผู้ใช้งานที่สามารถทำการเข้าสู่ระบบใน bmsproject.iot.ubidots.com โดยการเพิ่ม Token ใน Organization domain ทำการรับค่าTokenขององค์กรต้องการให้สิทธิ์เข้าถึง bmsproject.iot.ubidots.com โดยไปที่แท็บ "Users" จากนั้นคลิกที่ " Organization " เลือกองค์กรที่ต้องการ ในโครงงานนี้เลือกองค์กร My BMS team (รูปที่ 3.119)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

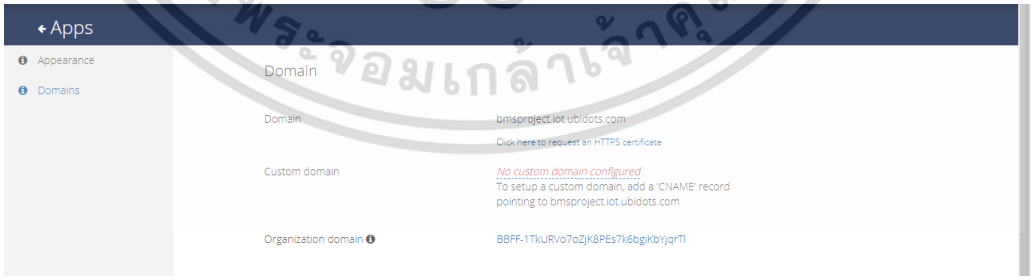


รูปที่ 3.119 แสดงหน้าองค์กร My BMS team

เลือก Tokens จากนั้นทำการคัดลอก Token ดังแสดงในรูป 3.120

รูปที่ 3.120 แสดงหน้า Token ขององค์กร My BMS team

นำ Token ที่ทำการคัดลอกมาใส่ในช่อง Organization domain (รูปที่ 3.121

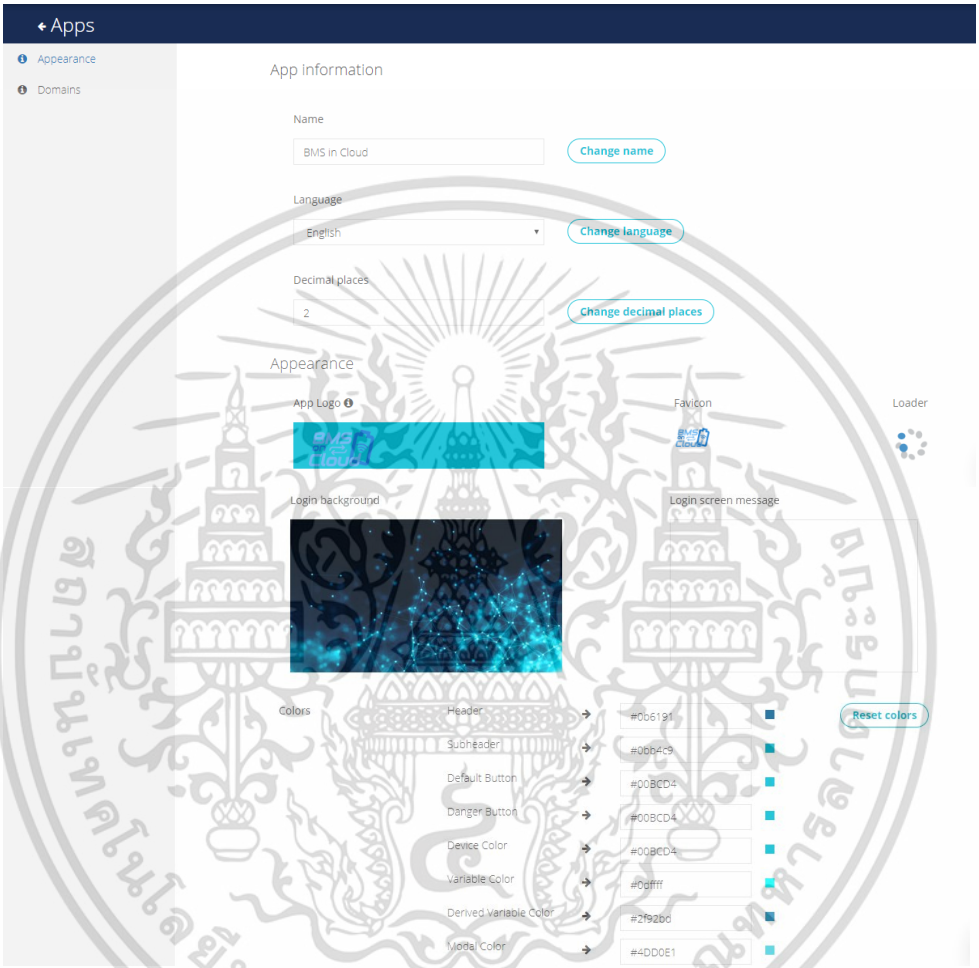


รูปที่ 3.121 แสดงการเพิ่ม Token ขององค์กร My BMS team ใน Organization domain

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ส่วน interface ของผู้ใช้แอปพลิเคชัน

สามารถทำการปรับแต่งได้ดังต่อไปนี้ สัญลักษณ์, สี, หน้า login, ภาษาและส่วนประกอบอื่น ๆ ของแอปพลิเคชัน การปรับแต่งสามารถทำได้โดยไปที่แท็บ "App" จากนั้นคลิกที่ " Manage App" เลือก Appearance จากนั้นทำการปรับแต่งแอปพลิเคชันดังรูป 3.122

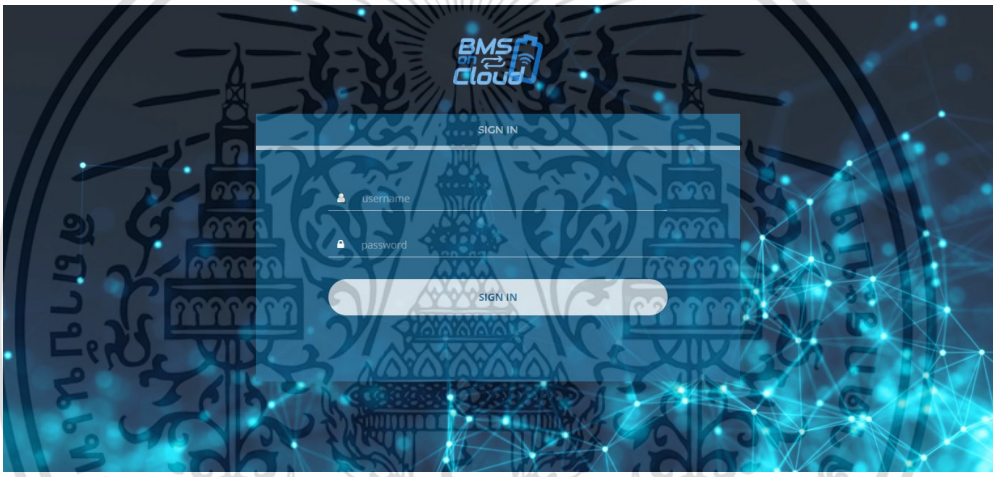


รูปที่ 3.122 แสดงรายละเอียดการปรับแต่งหน้า interface ของผู้ใช้แอปพลิเคชัน

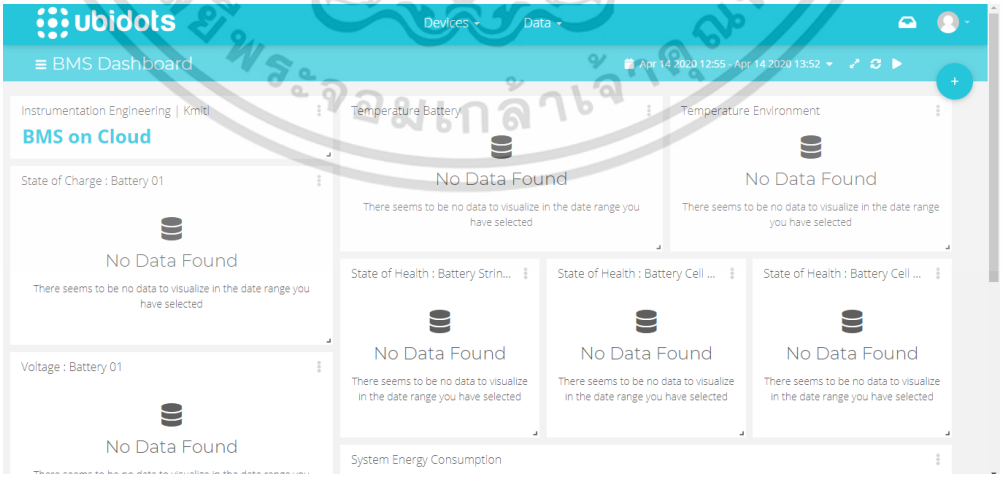
เลือก Save Changes เพื่อทำการบันทึกการปรับแต่ง โดยก่อนการปรับแต่งจะแสดงหน้าลงทะเบียนดังรูปที่ 3.123 ภายหลังทำการปรับแต่งแล้วจะปรากฏดังรูปที่ 3.124 และก่อนการปรับแต่งจะแสดงหน้า Dashboard ดังรูปที่ 3.125 ภายหลังทำการปรับแต่งแล้วจะปรากฏดังรูปที่ 3.126

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.123 แสดงหน้าลงทะเบียนของ bmsproject.iot.ubidots.com ก่อนการปรับแต่ง

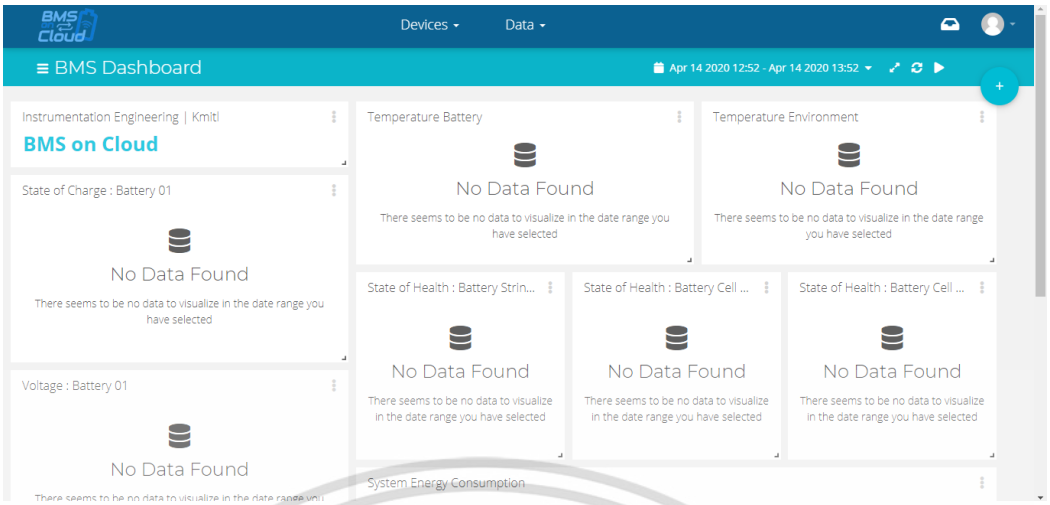


รูปที่ 3.124 แสดงหน้า ลงทะเบียนของ bmsproject.iot.ubidots.com หลังการปรับแต่ง



รูปที่ 3.125 แสดงหน้า Dashboard ของแอปพลิเคชัน BMS on Cloud ก่อนการปรับแต่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



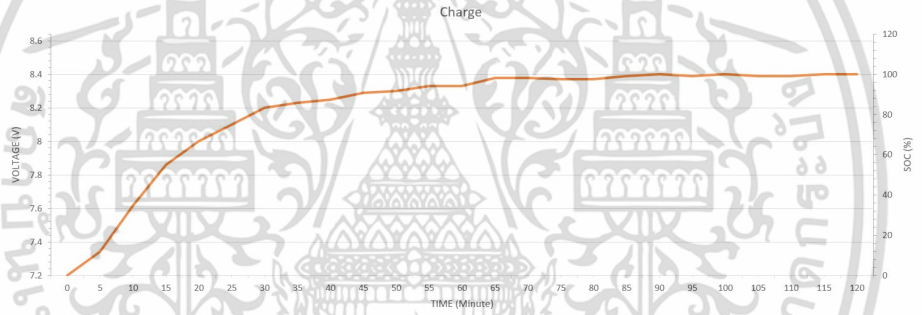
รูปที่ 3.126 แสดงหน้า Dashboard ของแอปพลิเคชัน BMS on Cloud หลังการปรับแต่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

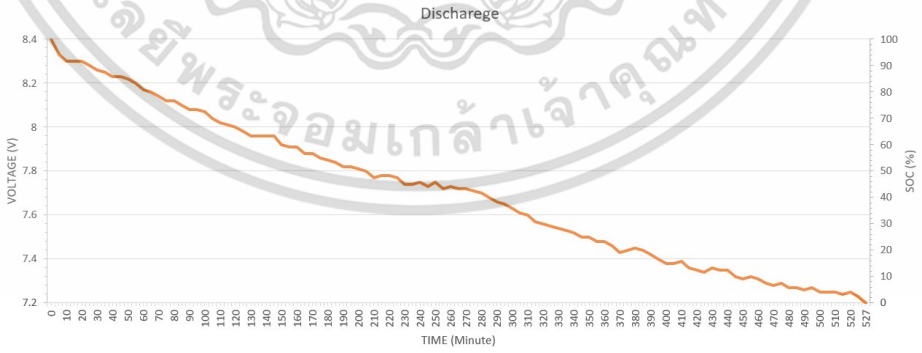
บทที่ 4
ผลการทดลอง

4.1 ทดสอบการ Charge – Discharge ของระบบ

การทดสอบนี้จะเป็นการเก็บค่า กระแสไฟฟ้า แรงดันไฟฟ้า อุณหภูมิสิ่งแวดล้อม อุณหภูมิแบตเตอรี่ และ เวลา เพื่อนำมาใช้หาสุขภาพของแบตเตอรี่ต่อไปได้ ในการ Charge - Discharge จะเริ่มต้นจากการกำหนดค่า SoC ที่ 0% เป็น 7.2 Volt และที่ SoC ที่ 100% เป็น 8.4 Volt ค่าที่ได้จาก Specification ของแบตเตอรี่ โดยใช้ LTC4162 ในการชาร์จ เพื่อควบคุมแรงดันคงที่, กระแสคงที่ และ อุณหภูมิในการชาร์จ เพื่อไม่ให้แบตเตอรี่เกิดการเสียหาย และใช้ Relay ที่รับคำสั่งมาจาก STM32 ช่วยในการบริหารจัดการในการ Charge - Discharge เมื่อเก็บค่าเรียบร้อยนำค่าที่ได้มาเขียนกราฟ Charge และ Discharge จะได้ดังนี้



รูปที่ 4.1 แสดงการ Charge



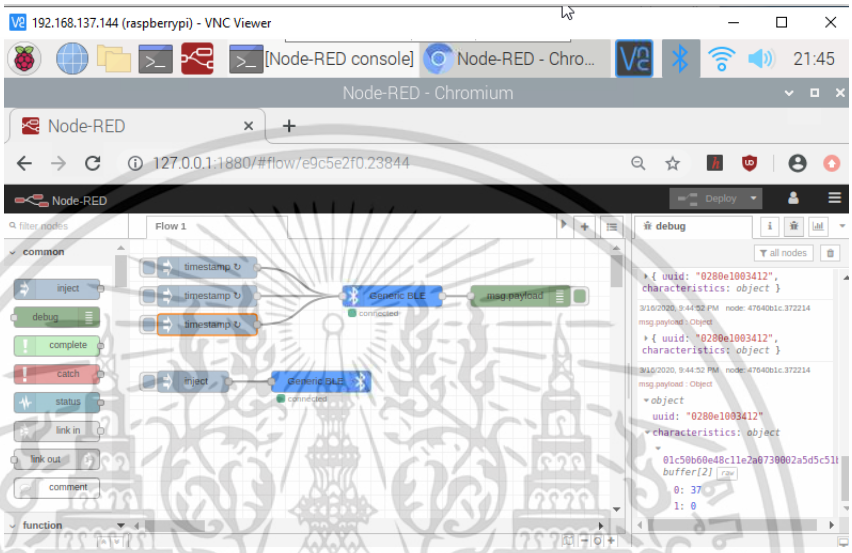
รูปที่ 4.2 แสดงการ Discharge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ทดสอบการสื่อสารข้อมูลระหว่าง STM32 และ Raspberry Pi ด้วย Bluetooth

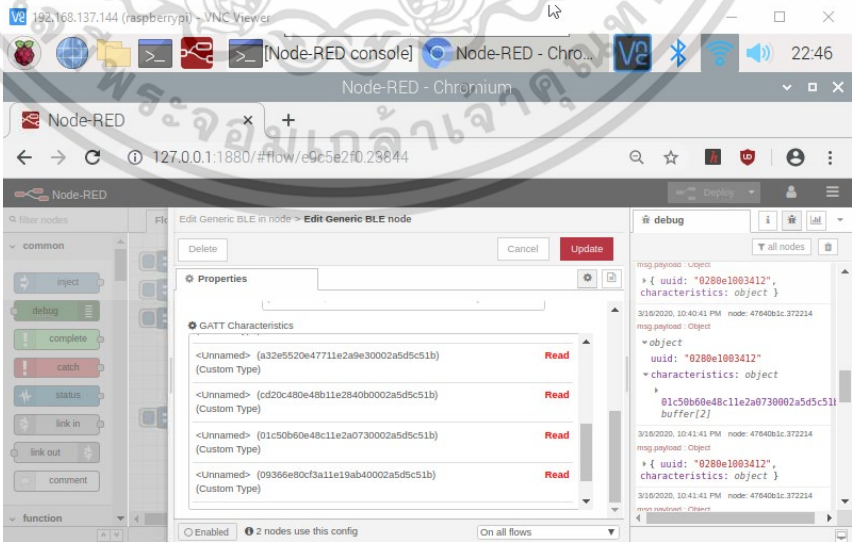
ในการทดสอบนี้จะทำการส่งค่าความชื้น (Humidity) ของ STM32 ไปยัง Raspberry Pi ด้วย Bluetooth โดยใช้โปรแกรม Node-Red ผลการทดสอบเป็นดังนี้

1) ใช้ผังการไหลของข้อมูลในหัวข้อที่ 3.6 เมื่อทำการเชื่อมต่อ Bluetooth จะแสดงสถานะ connected ดังรูปที่ 4.3



รูปที่ 4.3 แสดงสถานะของ Bluetooth

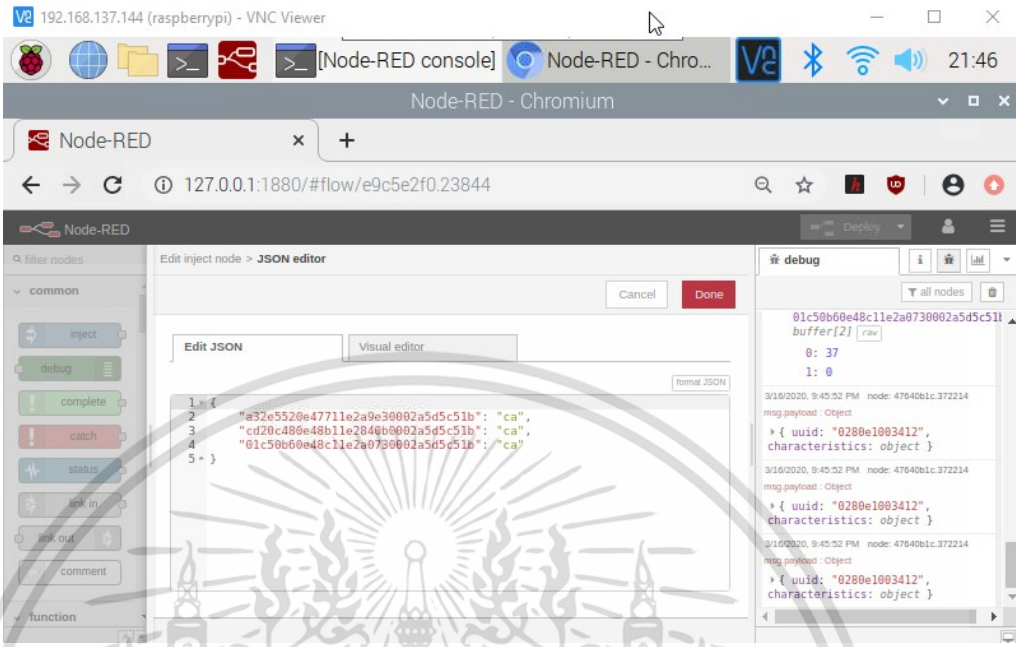
2) GATT Characteristics ขึ้นสถานะ Read หมายความว่ากำลังอ่านข้อมูลที่อยู่ใน Bluetooth ดังรูปที่ 4.4



รูปที่ 4.4 GATT Characteristics อ่านข้อมูลที่อยู่ใน Bluetooth

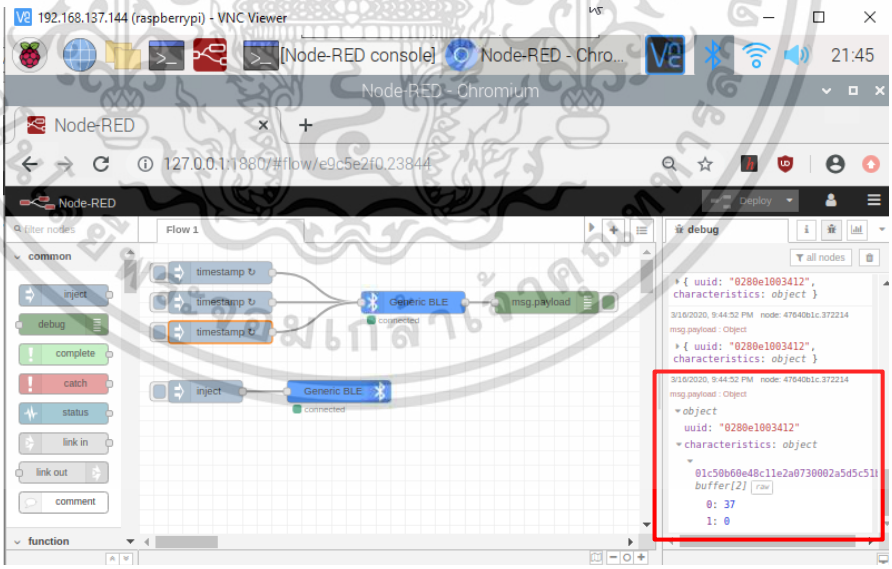
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เขียนฟังก์ชันเพื่อแปลงข้อมูลค่าความชื้นของ STM32 ตามรูปที่ 4.5



รูปที่ 4.5 ฟังก์ชันเพื่อแปลงข้อมูลค่าความชื้นของ STM32

4) เมื่อกด Deploy แล้วสังเกตในส่วนของ Debug จะขึ้นค่าความชื้นของ STM32 ดังรูปที่ 4.6



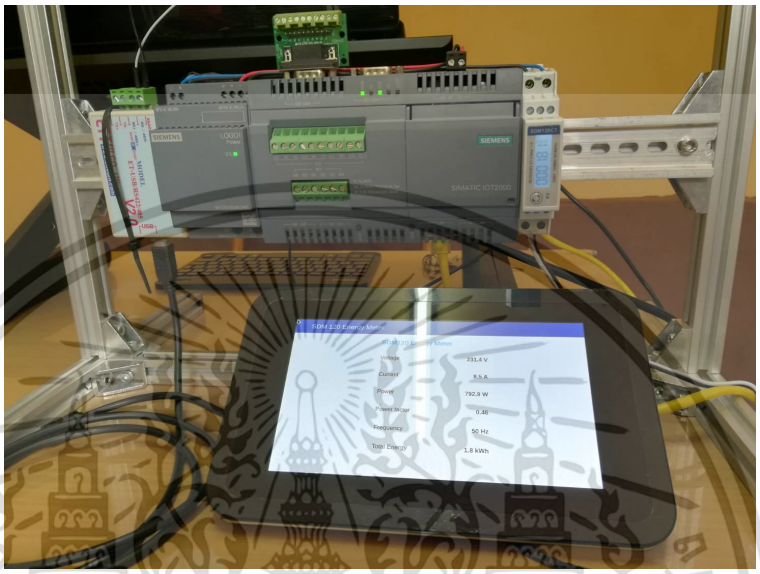
รูปที่ 4.6 แสดงค่าความชื้นของ STM32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

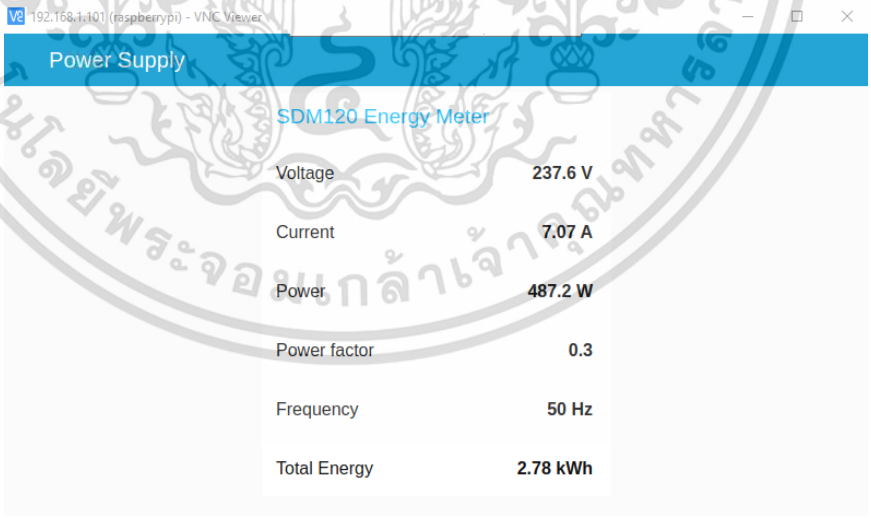
4.3 ทดสอบการแสดงผล Dashboard ของ Raspberry Pi

ใช้ผังการไหลของข้อมูลในหัวข้อที่ 3.8

1) ใช้ Siemens Simatic IOT 2040 วัดโหลดจากไฟฟ้าภายในบ้านพักอาศัย ผลการทดสอบแสดงดังรูปที่ 4.7 – 4.8



รูปที่ 4.7 โหลดจากไฟฟ้าภายในบ้านพักอาศัยที่วัดผ่าน Siemens Simatic IOT 2040 แสดงผลบน Raspberry Pi



รูปที่ 4.8 Dashboard ของ Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ในส่วนของ String Battery ทดลองส่งค่า

Voltage = 8

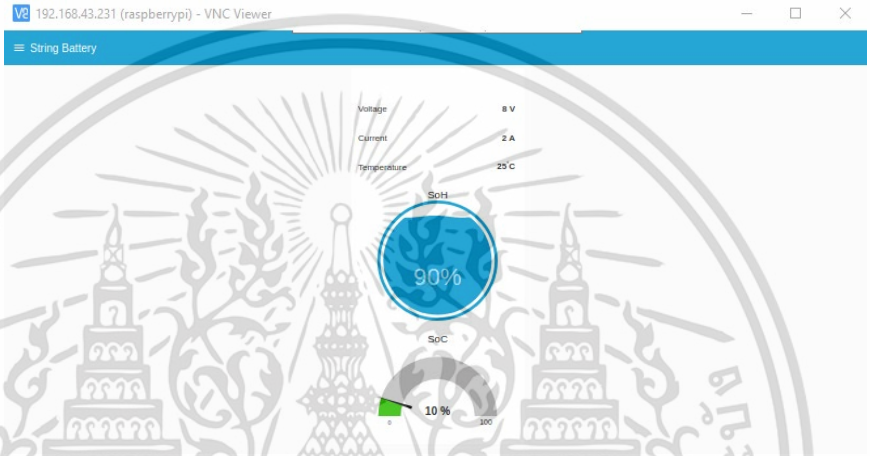
Current = 2

Temperature = 25

SoC = 10

SoH = 90

ผลการทดสอบแสดงดังรูปที่ 4.9



รูปที่ 4.9 Dashboard ของ String Battery

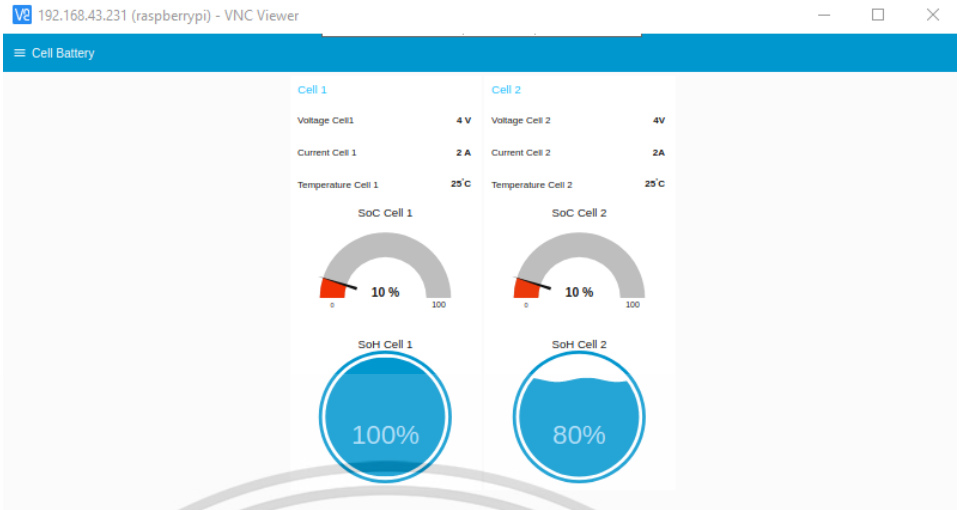
3) ในส่วนของ Cell Battery ทดลองส่งค่า

ตารางที่ 4.1 แสดงข้อมูลที่ใช้ทดสอบ Dashboard ของ Cell Battery

Cell 1	Cell 2
Voltage =4	Voltage =4
Current =2	Current =2
Temperature=25	Temperature=25
SoC =10	SoC =10
SoH =100	SoH =80

ผลการทดสอบแสดงดังรูปที่ 4.10

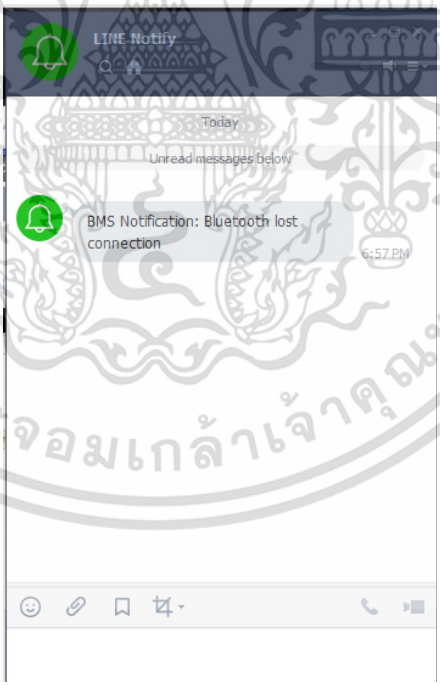
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 Dashboard ของ Cell Battery

4.4 ทดสอบการแจ้งเตือนผ่าน Line notify ด้วย Node-Red

ใช้ผังการไหลของข้อมูลในหัวข้อ 3.9 ทดสอบกรณีที่ Bluetooth ของ STM32 และ Raspberry Pi ขาดการเชื่อมต่อ ได้ผลการทดสอบดังรูปที่ 4.11



รูปที่ 4.11 แสดงการแจ้งเตือนผ่าน Line Notify

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ทดสอบการแสดงผลข้อมูลที่จัดเก็บใน SQLite

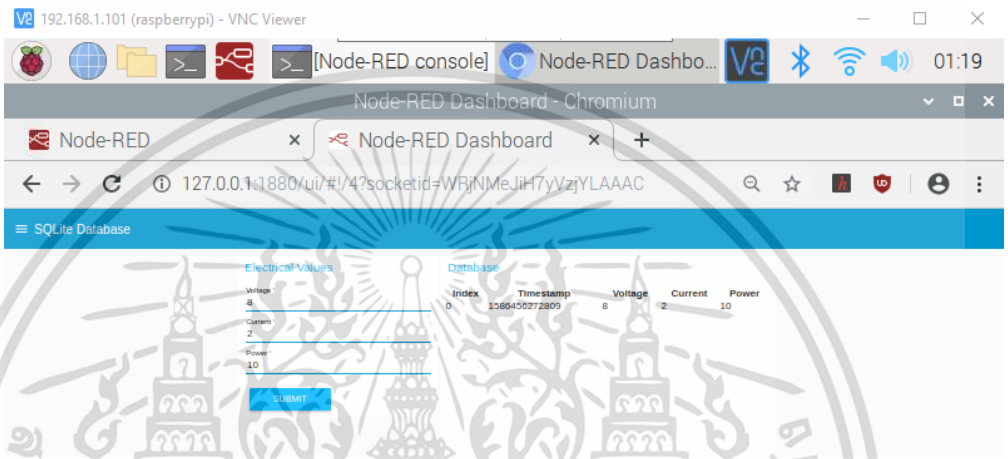
ใช้ผังการไหลของข้อมูลในหัวข้อ 3.10 ทดลองบันทึกค่า

Voltage = 8

Current = 2

Power = 10

ผลการทดสอบแสดงดังรูปที่ 4.12



รูปที่ 4.12 แสดงข้อมูลที่ถูกจัดเก็บใน SQLite

4.6 ทดสอบการส่งค่า Simulation จาก Edge computing ไปยัง Cloud computing

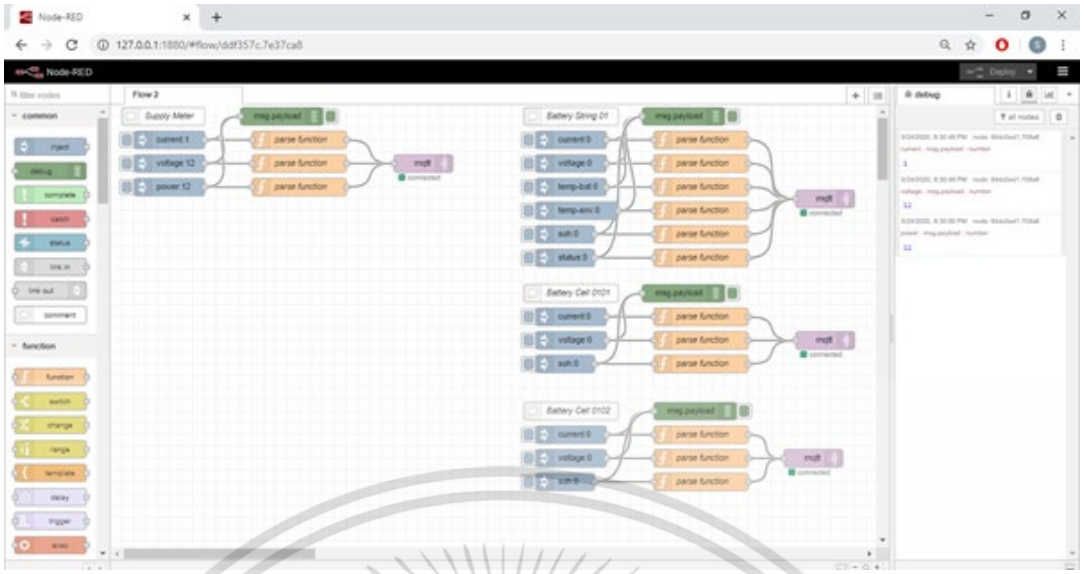
ผลการเชื่อมต่อและการรับส่งข้อมูลระหว่างส่วน Edge computing และส่วน Cloud computing ส่งค่าแรงดัน, กระแส, อุณหภูมิของแบตเตอรี่, อุณหภูมิของสภาวะแวดล้อมของแบตเตอรี่และส่งค่าการใช้งานไฟฟ้าของ Supply Meter จาก Raspberry Pi ด้วย Node-Red ผ่าน MQTT protocol ไปยัง Ubidots แสดงผลในหน้า Devices ของแอปพลิเคชัน

1) อุปกรณ์ Supply Meter

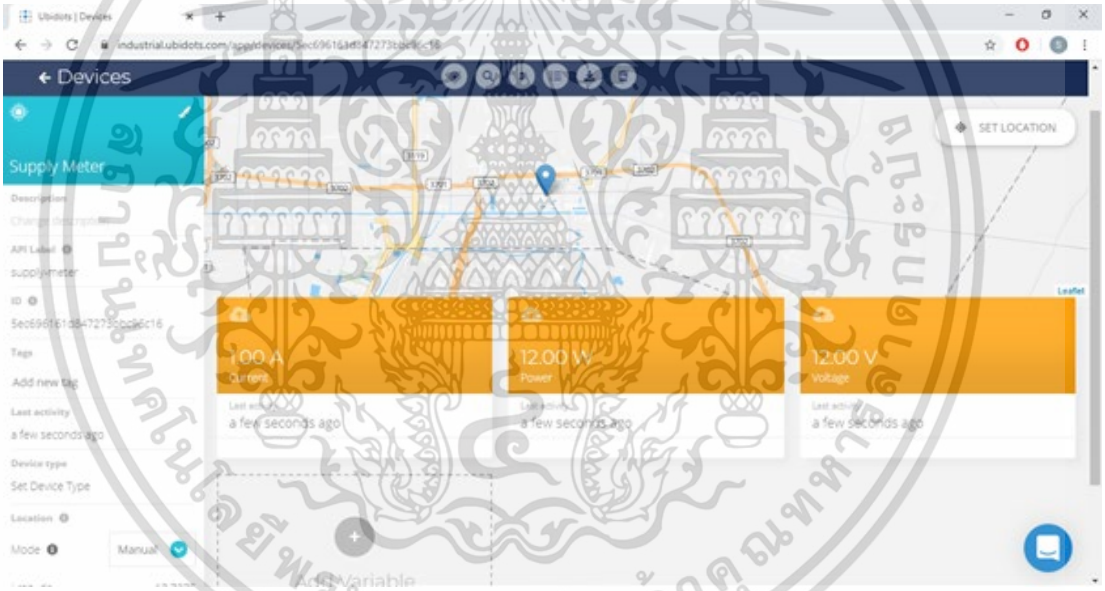
Edge Computing ส่งค่า Current = 1, Voltage = 12, Power = 12 (รูปที่ 4.3)

หน้าอุปกรณ์ Supply Meter ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing จะปรากฏดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 แสดงการส่งค่าจาก Edge computing ไปยังอุปกรณ์ Supply Meter



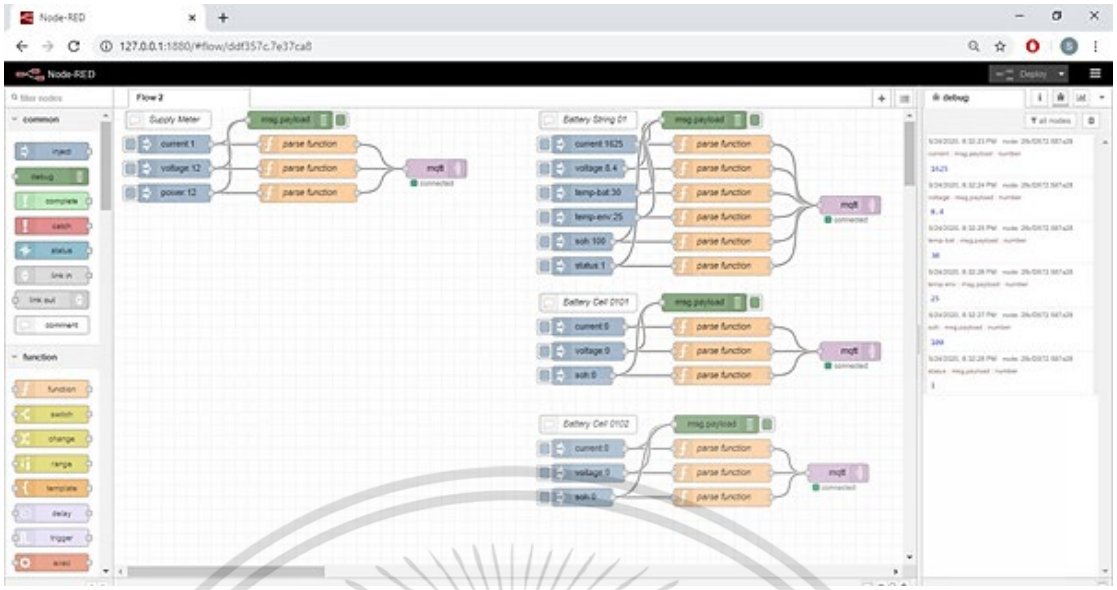
รูปที่ 4.14 หน้าอุปกรณ์ Supply Meter ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing

2) อุปกรณ์ Battery String 01

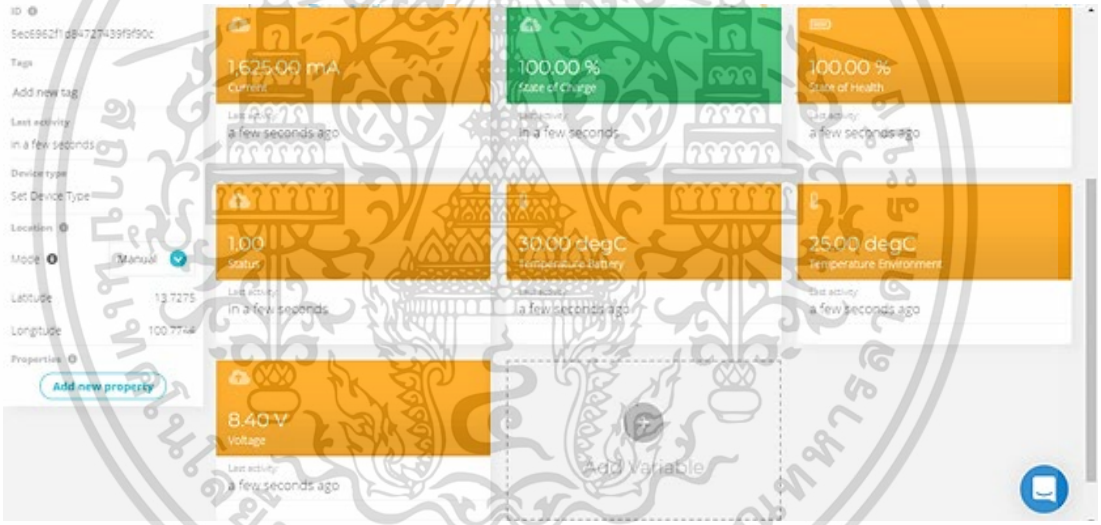
Edge computing ส่งค่า Current = 1625, Voltage = 8.4, temperature battery = 30, temperature environment = 25, Status =1 และ State of Health = 100 (รูปที่ 4.15)

หน้าอุปกรณ์ Battery String 01 ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing จะแสดงดังรูปที่ 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงการส่งค่าจาก Edge computing ไปยังอุปกรณ์ Battery String 01



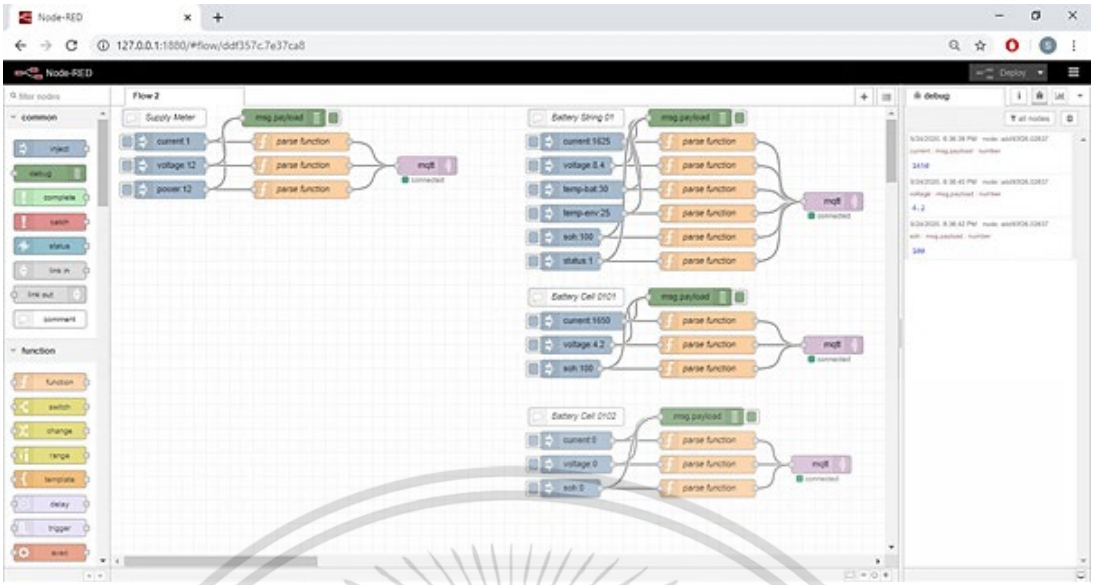
รูปที่ 4.16 หน้าอุปกรณ์ Battery String 01 ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing

3) อุปกรณ์ Battery Cell 0101

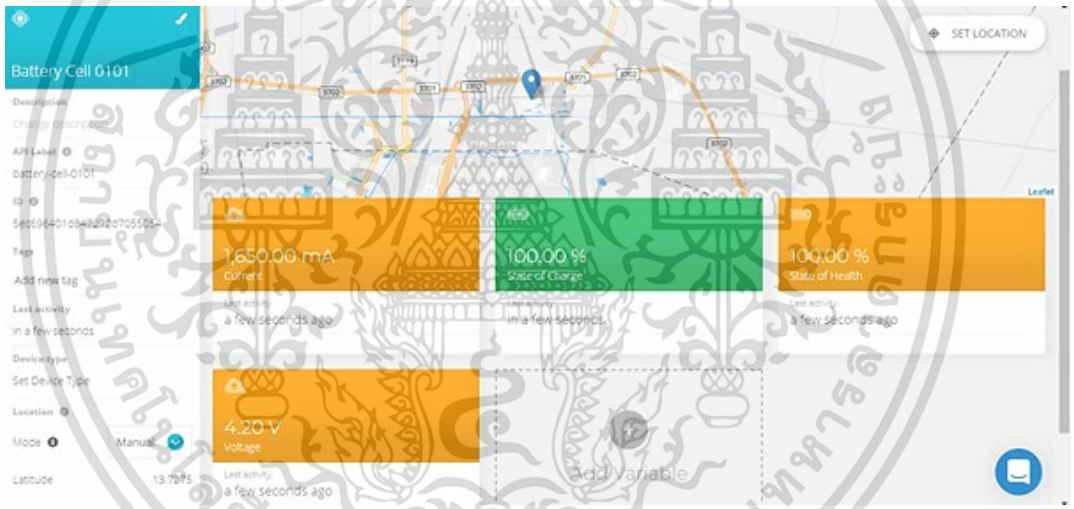
Edge computing ส่งค่า Current = 1625, Voltage = 4.2 และ State of Health = 100 (รูปที่ 4.17)

หน้าอุปกรณ์ Battery Cell 0101 ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing จะแสดงดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 แสดงการส่งค่าจาก Edge computing ไปยังอุปกรณ์ Battery Cell 0101



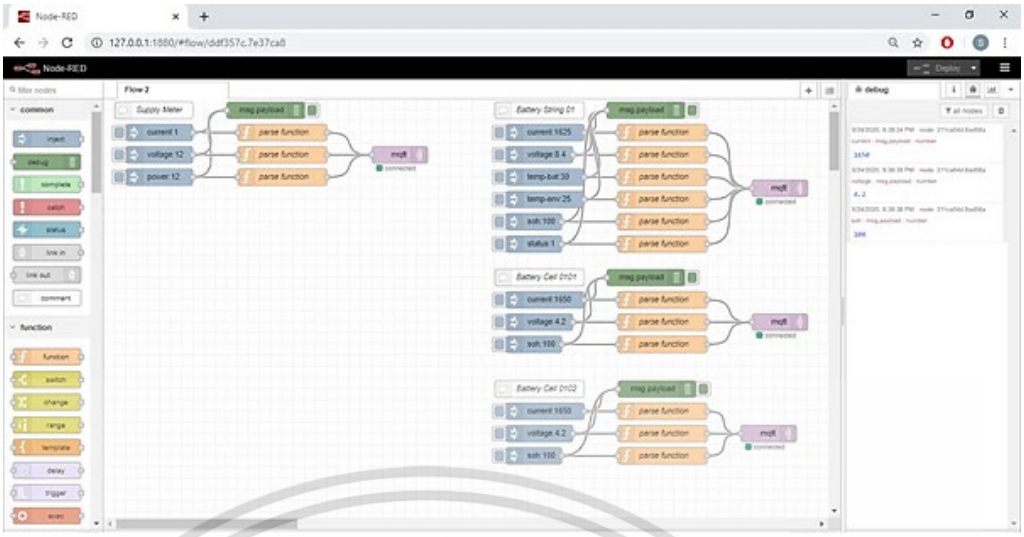
รูปที่ 4.18 หน้าอุปกรณ์ Battery Cell 0101 ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing

4) อุปกรณ์ Battery Cell 0102

Edge computing ส่งค่า Current = 1625, Voltage = 4.2 และ State of Health = 100 (รูปที่ 4.19)

หน้าอุปกรณ์ Battery Cell 0102 ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing จะแสดงดังรูปที่ 4.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แสดงการส่งค่าจาก Edge computing ไปยังอุปกรณ์ Battery Cell 0102

รูปที่ 4.20 หน้าอุปกรณ์ Battery Cell 0102 ค่าตัวแปรล่าสุดที่รับมาจากส่วน Edge computing

จากการทดสอบการส่งค่า Simulation จาก Edge computing ไปยัง Cloud computing ด้วยค่า Simulation พบว่าค่าที่รับได้มีค่าตรงกับค่าที่ส่งมาจากส่วน Edge computing ดังนั้นจึงสามารถสรุปได้ว่าระบบสามารถส่งค่า Simulation จาก Edge computing ไปยัง Cloud computing ได้อย่างถูกต้อง

4.7 ความถูกต้องของสมการคำนวณจากค่า Simulation

การทดสอบความถูกต้องของการคำนวนสมการ State of Charge จะทำการทดสอบโดยใช้ค่า Simulation โดยมีขั้นตอนการทดสอบดังต่อไปนี้

1) การทดสอบการคำนวนของตัวแปร State of Charge ของอุปกรณ์ Battery String 01 แสดงดังตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 แสดงการคำนวนค่า State of Charge ของอุปกรณ์ Battery String 01

Voltage (V)	*State of Charge* (%) = ((*Voltage* -7.2) *100) / (8.4-7.2)
8.4	100.00
8.2	83.33
8	66.67
7.8	50.00
7.6	33.33
7.4	16.67
7.2	0.00

2) การทดสอบการคำนวนของตัวแปร State of Charge ของอุปกรณ์ Battery Cell 0101 และอุปกรณ์ Battery Cell 0102 แสดงดังตารางที่ 4.3

ตารางที่ 4.3 แสดงการคำนวนค่า State of Charge ของอุปกรณ์ Battery Cell 0101, Battery Cell 0102

Voltage (V)	*State of Charge* (%) = ((*Voltage* -3.6) *100) / (4.2-3.6)
4.2	100.00
4	66.67
3.8	33.33
3.6	0

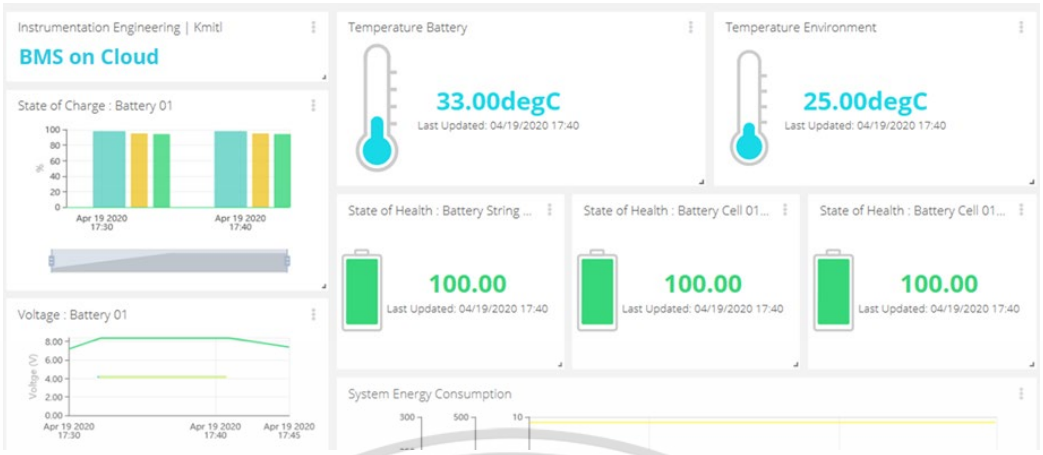
จากการทดสอบความถูกต้องของสมการคำนวนจะทำการทดสอบโดยใช้ค่า Simulation พบว่าค่าที่คำนวนได้มีค่าตรงกับค่าที่ได้จากการคำนวน จึงสามารถสรุปได้ว่าสมการที่ใช้ในการคำนวนมีความถูกต้อง

4.8 ทดสอบการแสดงผลของหน้า Dashboard ตามเงื่อนไขที่ทำการตั้งค่าไว้

การทดสอบการแสดงผลของหน้า Dashboard จะประกอบด้วยการแสดงผลของตัวแปรและอุปกรณ์ดังตารางที่ 3.5 ซึ่งประกอบด้วยการทดสอบดังต่อไปนี้

1) การทดสอบในส่วนของการแสดงผลทั้งหมดในหน้า Dashboard จะแสดงผลดังรูปที่ 4.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

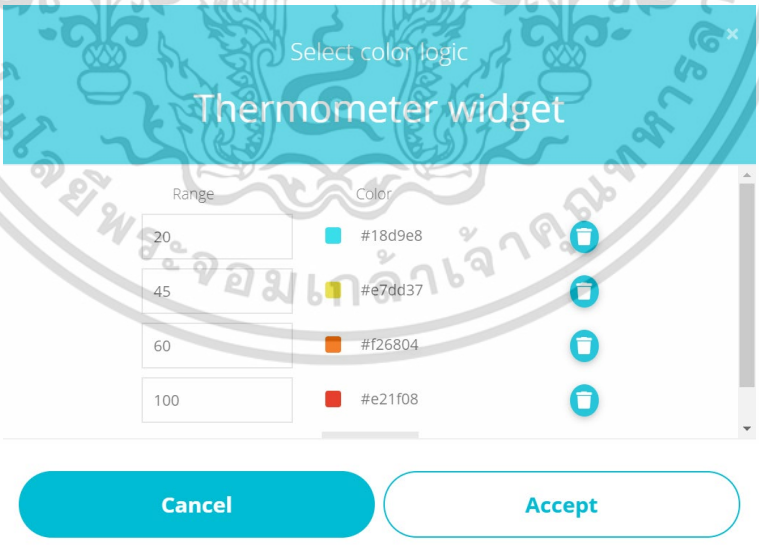


รูปที่ 4.21 การแสดงผลในหน้า Dashboard

2) การทดสอบการเปลี่ยนแปลงสีของ Widget ของตัวแปร Temperature battery, Temperature environment และ State of Health ว่ามีการเปลี่ยนแปลงตามการตั้งค่า Color logic ไว้หรือไม่

a) การทดสอบการเปลี่ยนแปลงสีของ Thermometer widget ของตัวแปร Temperature battery และตัวแปร Temperature environment

Color logic ที่กำหนดจะปรากฏดังรูปที่ 4.22



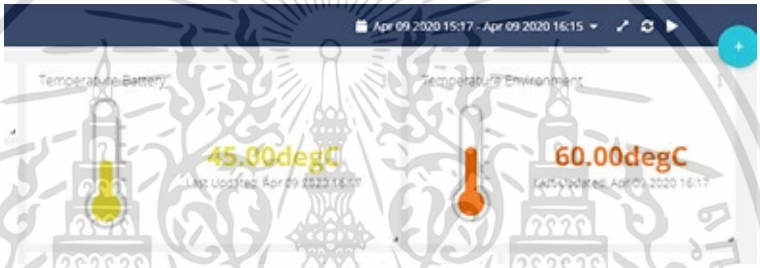
รูปที่ 4.22 Color logic ที่กำหนดไว้ของ Thermometer widget

เมื่อค่า Temperature battery ที่รับได้มีค่าเท่ากับ 20 และค่า Temperature environment ที่รับได้มีค่าเท่ากับ 100 ดังแสดงในรูปที่ 4.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.23 Temperature battery widget แสดงอุณหภูมิ 20 degC และ Temperature environment widget แสดงอุณหภูมิ 100 degC

เมื่อค่า Temperature battery ที่รับได้มีค่าเท่ากับ 45 และค่า Temperature environment ที่รับได้มีค่าเท่ากับ 60 ดังแสดงในรูปที่ 4.24



รูปที่ 4.24 Temperature battery widget แสดงอุณหภูมิ 45 degC และ Temperature environment widget แสดงอุณหภูมิ 100 degC

เมื่อค่า Temperature battery ที่รับได้มีค่าเท่ากับ 60 และค่า Temperature environment ที่รับได้มีค่าเท่ากับ 45 ดังแสดงในรูปที่ 4.25

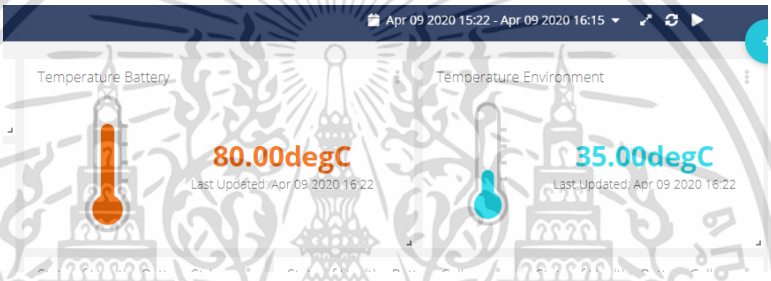
รูปที่ 4.25 Temperature battery widget แสดงอุณหภูมิ 60 degC และ Temperature environment widget แสดงอุณหภูมิ 45 degC

เมื่อค่า Temperature battery ที่รับได้มีค่าเท่ากับ 100 และค่า Temperature environment ที่รับได้มีค่าเท่ากับ 20 ดังแสดงในรูปที่ 4.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.26 Temperature battery widget แสดงอุณหภูมิ 100 degC และ Temperature environment widget แสดงอุณหภูมิ 20 degC

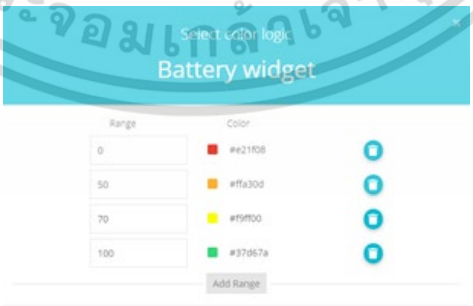
เมื่อค่า Temperature battery ที่รับได้มีค่าเท่ากับ 80 และค่า Temperature environment ที่รับได้มีค่าเท่ากับ 35 ดังแสดงในรูปที่ 4.27



รูปที่ 4.27 Temperature battery widget แสดงอุณหภูมิ 80 degC และ Temperature environment widget แสดงอุณหภูมิ 35 degC

b) การทดสอบการเปลี่ยนแปลงสีของ Battery widget ของตัวแปร State of Health ของอุปกรณ์ Battery String 01, Battery Cell 0101 และ Battery Cell 0102

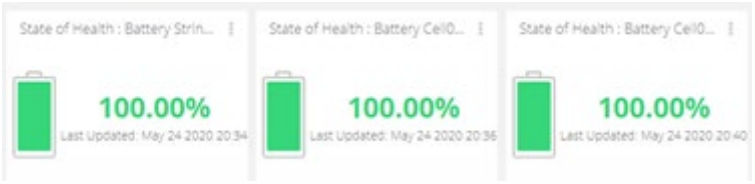
Color logic ที่กำหนดจะปรากฏดังรูปที่ 4.28



รูปที่ 4.28 Color logic ที่กำหนดไว้ของ battery widget

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อค่า State of Health ของอุปกรณ์ Battery String 01 ที่รับได้มีค่าเท่ากับ 100, ค่า State of Health ของอุปกรณ์ Battery Cell 0101 ที่รับได้มีค่าเท่ากับ 100, ค่า State of Health ของอุปกรณ์ Battery Cell 0102 ที่รับได้มีค่าเท่ากับ 100 ดังแสดงในรูปที่ 4.29



รูปที่ 4.29 State of Health : Battery String 01 widget แสดง 100%, State of Health : Battery Cell 0101 widget แสดง 100% และ State of Health : Battery Cell 0101 widget แสดง 100%

เมื่อค่า State of Health ของอุปกรณ์ Battery String 01 ที่รับได้มีค่าเท่ากับ 70, ค่า State of Health ของอุปกรณ์ Battery Cell 0101 ที่รับได้มีค่าเท่ากับ 70, ค่า State of Health ของอุปกรณ์ Battery Cell 0102 ที่รับได้มีค่าเท่ากับ 70 ดังแสดงในรูปที่ 4.30

รูปที่ 4.30 State of Health : Battery String 01 widget แสดง 70%, State of Health : Battery Cell 0101 widget แสดง 70% และ State of Health : Battery Cell 0101 widget แสดง 70%

เมื่อค่า State of Health ของอุปกรณ์ Battery String 01 ที่รับได้มีค่าเท่ากับ 50, ค่า State of Health ของอุปกรณ์ Battery Cell 0101 ที่รับได้มีค่าเท่ากับ 50, ค่า State of Health ของอุปกรณ์ Battery Cell 0102 ที่รับได้มีค่าเท่ากับ 50 ดังแสดงในรูปที่ 4.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.31 State of Health : Battery String 01 widget แสดง 50%, State of Health : Battery Cell 0101 widget แสดง 50% และ State of Health : Battery Cell 0101 widget แสดง 50%

เมื่อค่า State of Health ของอุปกรณ์ Battery String 01 ที่รับได้มีค่าเท่ากับ 0, ค่า State of Health ของอุปกรณ์ Battery Cell 0101 ที่รับได้มีค่าเท่ากับ 0, ค่า State of Health ของอุปกรณ์ Battery Cell 0102 ที่รับได้มีค่าเท่ากับ 0 ดังแสดงในรูปที่ 4.32

รูปที่ 4.32 State of Health : Battery String 01 widget แสดง 0%, State of Health : Battery Cell 0101 widget แสดง 0% และ State of Health : Battery Cell 0101 widget แสดง 0%

จากการทดสอบการแสดงผลของหน้า Dashboard พบว่าแต่ละ Widget แสดงผลได้ตามเงื่อนไขที่ทำการตั้งค่าไว้ กล่าวคือ Dashboard สามารถแสดงผลของตัวแปรที่ตั้งค่าไว้ได้อย่างถูกต้อง, Widget มีการเปลี่ยนแปลงสีตาม Color logic ได้อย่างถูกต้อง ดังนั้นจึงสามารถสรุปได้ว่าระบบสามารถแสดงผลหน้า Dashboard ได้ตรงตามเงื่อนไขที่ตั้งค่าไว้

4.9 ทดสอบการแจ้งเตือนอัตโนมัติ

การทดสอบการแจ้งเตือนอัตโนมัติจะทำการทดสอบโดยใช้ค่า Simulation ซึ่งจะประกอบด้วยการทดสอบดังต่อไปนี้

1) การทดสอบการแจ้งเตือนอัตโนมัติเมื่อ State of Health ≤ 50%

เมื่อค่า State of Health ของอุปกรณ์ Battery String 01 ที่รับได้มีค่าเท่ากับ 100%, พบว่าระบบไม่มีการส่งอีเมลแจ้งเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อค่า State of Health ของอุปกรณ์ Battery Cell 0101 ที่รับได้มีค่าเท่ากับ 80%, พบว่าระบบไม่มีการส่งอีเมลแจ้งเตือน

เมื่อค่า State of Health ของอุปกรณ์ Battery Cell 0102 ที่รับได้มีค่าเท่ากับ 60%, พบว่าระบบไม่มีการส่งอีเมลแจ้งเตือน

เมื่อค่า State of Health ของอุปกรณ์ Battery String 01 ที่รับได้มีค่าเท่ากับ 50%, พบว่าระบบมีการส่งอีเมลแจ้งเตือนดังรูปที่ 4.33

รูปที่ 4.33 อีเมลการแจ้งเตือน State of Health ของอุปกรณ์ Battery String 01

เมื่อค่า State of Health ของอุปกรณ์ Battery Cell 0101 ที่รับได้มีค่าเท่ากับ 30%, พบว่าระบบมีการส่งอีเมลแจ้งเตือนดังรูปที่ 4.34

รูปที 4.34 อีเมลการแจ้งเตือน State of Health ของอุปกรณ์ Battery Cell 0101

เมื่อค่า State of Health ของอุปกรณ์ Battery Cell 0102 ที่รับได้มีค่าเท่ากับ 20% พบว่าระบบมีการส่งอีเมลแจ้งเตือนดังรูปที่ 4.35

รูปที่ 4.35 อีเมลการแจ้งเตือน State of Health ของอุปกรณ์ Battery Cell 0102

2) การทดสอบการแจ้งเตือนอัตโนมัติเมื่ออุณหภูมิขณะชาร์จ > 45 degC หรือ อุณหภูมิขณะดิสชาร์จ > 60 degC

เมื่อขณะชาร์จค่าอุณหภูมิของแบตเตอรี่มีค่าเท่ากับ 25 degC, พบว่าระบบไม่มีการส่งอีเมลแจ้งเตือน

เมื่อขณะชาร์จค่าอุณหภูมิของแบตเตอรี่มีค่าเท่ากับ 45 degC, พบว่าระบบไม่มีการส่งอีเมลแจ้งเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อขณะชาร์จค่าอุณหภูมิของแบตเตอรี่มีค่าเท่ากับ 50 degC, พบว่าระบบมีการส่งอีเมลแจ้งเตือนดังรูปที่ 4.36

รูปที่ 4.36 อีเมลการแจ้งเตือนอุณหภูมิของแบตเตอรี่ขณะชาร์จ > 45 degC

เมื่อขณะดิสชาร์จค่าอุณหภูมิของแบตเตอรี่มีค่าเท่ากับ 25 degC, พบว่าระบบไม่มีการส่งอีเมลแจ้งเตือน

เมื่อขณะดิสชาร์จค่าอุณหภูมิของแบตเตอรี่มีค่าเท่ากับ 60 degC, พบว่าระบบไม่มีการส่งอีเมลแจ้งเตือน

เมื่อขณะดิสชาร์จค่าอุณหภูมิของแบตเตอรี่มีค่าเท่ากับ 70 degC, พบว่าระบบมีการส่งอีเมลแจ้งเตือนดังรูปที่ 4.37

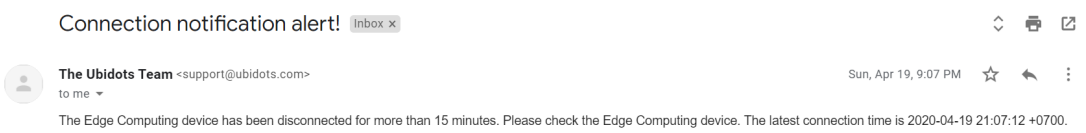
รูปที่ 4.37 อีเมลการแจ้งเตือนอุณหภูมิของแบตเตอรี่ขณะชาร์จ > 60 degC

3) การทดสอบการแจ้งเตือนอัตโนมัติเมื่ออุปกรณ์ขาดการเชื่อมต่อนานกว่า 15 นาที

เนื่องจากในสภาวะปกติ Edge computing จะการทำการส่งค่าตัวแปร Current แต่ละค่าห่างกัน 15 นาทีดังนั้นจึงทำการทดสอบโดยให้ Edge computing ทำการส่งค่าตัวแปร Current แต่ละค่าห่างกัน 18 นาที

เมื่อ Edge computing จะการทำการส่งค่าตัวแปร Current แต่ละค่าห่างกัน 15 นาที, พบว่าระบบไม่มีการส่งอีเมลแจ้งเตือน

เมื่อ Edge computing จะการทำการส่งค่าตัวแปร Current แต่ละค่าห่างกัน 18 นาที, พบว่าระบบมีการส่งอีเมลแจ้งเตือนดังรูปที่ 4.38



รูปที่ 4.38 อีเมลการแจ้งเตือนอุปกรณ์ขาดการเชื่อมต่อนานกว่า 15 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบการแจ้งเตือนอัตโนมัติโดยใช้ค่า Simulation สามารถสรุปได้ว่าระบบบสามารถทำการแจ้งเตือนอัตโนมัติ ได้ตรงตามเงื่อนไขที่กำหนดไว้

4.10 ทดสอบการส่งรายงานอัตโนมัติประจำสัปดาห์

ทำการทดสอบโดยตั้งค่าให้รายงานประจำสัปดาห์ทำการส่งอัตโนมัติในทุกวันจันทร์ เวลา 08:30 น. พบว่าเมื่อครบกำหนดเวลาที่ตั้งค่าไว้อีเมลการแจ้งเตือนการส่งรายงานอัตโนมัติประจำสัปดาห์จะถูกส่งดังปรากฏในรูปที่ 4.39

รูปที่ 4.39 อีเมลการแจ้งเตือนการส่งรายงานอัตโนมัติประจำสัปดาห์

จากการทดสอบการส่งรายงานอัตโนมัติประจำสัปดาห์ สามารถสรุปได้ว่าระบบบสามารถส่งรายงานอัตโนมัติประจำสัปดาห์ได้ตรงตามวันเวลาที่กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

โครงงานนี้ได้จัดทำระบบจัดการแบตเตอรี่บนแพลตฟอร์มคลาวด์ ระบบสามารถควบคุมและรับค่าการทำงานการ Charge-Discharge ของแบตเตอรี่โดยการกำหนดค่า SoC ที่ 0% เป็น 7.2 Volt และที่ SoC ที่ 100% เป็น 8.4 Volt เมื่อแบตเตอรี่มี SoC ที่ 0% ระบบจะทำการสั่ง Charge แบตเตอรี่และเก็บผลการทดลองจนแบตเตอรี่มีค่า SoC ถึง 100% ระบบจะทำการสั่ง Discharge แบตเตอรี่และทำการเก็บผลการทดลอง หลังจากนั้นก็ทำการนำค่าที่ได้จากการบันทึกผลการทดลองมาเขียนกราฟแสดงการ Charge – Discharge เพื่อให้เห็นพฤติกรรมการทำงานของแบตเตอรี่ จากพฤติกรรมของแบตเตอรี่จะสามารถบอกสุขภาพและอายุการใช้งานของแบตเตอรี่ก้อนนั้น ๆ ได้ และยังบอกได้ว่าเป็นไปตามเงื่อนไขการใช้งานจากโรงงานหรือไม่ จากการทดสอบระบบพบว่าสามารถทำการเชื่อมต่อและการรับส่งข้อมูลจากส่วน Edge computing ไปยังส่วน Cloud computing ได้จาก Raspberry Pi ด้วย Node-Red ผ่าน MQTT protocol ไปยัง Ubidots และ Ubidots สามารถนำค่าที่รับมาไปทำการวิเคราะห์และประมวลผลได้ตามวัตถุประสงค์ที่ต้องการ และระบบสามารถแสดงผลในหน้า Dashboard บนแอปพลิเคชันซึ่งทำให้ผู้ใช้งานสามารถติดตามผลการใช้งานของแบตเตอรี่ผ่านอุปกรณ์คอมพิวเตอร์หรือสมาร์ทโฟนได้ตลอด 24 ชั่วโมง นอกจากนี้ระบบยังสามารถเก็บข้อมูลบน Cloud เพื่อติดตามข้อมูลย้อนหลังได้ถึง 30 วันและเมื่อเกิดปัญหาขึ้นกับระบบ ระบบจะมีการแจ้งเตือนไปยังอีเมลของผู้ใช้งานอย่างอัตโนมัติและยังสามารถจัดทำรายงานผลประจำสัปดาห์ไปยังผู้ใช้งาน ซึ่งตรงตามวัตถุประสงค์ของการจัดทำโครงงาน

5.2 ปัญหาและอุปสรรค

การติดตั้ง Node-Red บน Raspberry Pi จำเป็นต้องติดตั้ง Node.js ให้สอดคล้องกันเนื่องจากบางคำสั่งไม่สามารถใช้งานได้กับ Node-Red เวอร์ชันเก่า

การจัดทำระบบซึ่งมีส่วนประกอบของ Sensor node, Edge computing และ Cloud computing จำเป็นต้องทำความเข้าใจความสัมพันธ์และการเชื่อมต่อของแต่ละส่วนทำให้ใช้เวลานานในการศึกษา

อุปกรณ์ฮาร์ดแวร์บางส่วนต้องสั่งจากต่างประเทศ จึงต้องใช้ระยะเวลาในการติดต่อและจัดส่ง

เนื่องจากสถานการณ์การระบาดของไวรัสโควิค-19 ทำให้เกิดปัญหาทางการเชื่อมต่อและสื่อสารระหว่างอุปกรณ์ จึงส่งผลให้เกิดการล่าช้าไม่ตรงไปตามแผนงานที่วางไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะสำหรับการวิจัยในอนาคต

ในส่วนของ Edge computing ในการใช้งาน Raspberry Pi สามารถเลือกติดตั้งระบบปฏิบัติการได้ตามความเหมาะสมและยังมีระบบบบปฏิบัติการให้เลือกอย่างหลากหลายเหมาะสำหรับนักพัฒนาที่ต้องการสร้างองค์ความรู้ใหม่ ๆ

ในส่วนของ Cloud computing ในโครงงานนี้ได้ทำการเลือกใช้ผู้ให้บริการ Ubidots แบบทดลองใช้งานได้เพียง 30 วัน ทำให้เมื่อครบกำหนดต้องทำการสมัครสมาชิกใหม่และดำเนินการทำข้อมูลอีกครั้ง ทำให้เกิดการเสียเวลาเป็นอย่างมาก หากต้องการต่อยอดและพัฒนาระบบ ควรซื้อตัวสมบูรณ์ เพื่อการใช้งานได้อย่างต่อเนื่องและเต็มประสิทธิภาพ หรือ สามารถเปลี่ยนผู้ให้บริการให้ตรงกับคุณสมบัติที่ต้องการใช้งาน เช่น Azure, AWS เป็นต้น หรือหากต้องการใช้ผู้บริการ Ubidots ต่อสามารถลดคุณสมบัติของระบบและเปลี่ยนมาใช้บัญชี Ubidots แบบ Stem

สามารติดตั้งโมดูลการ์ดหน่วยความจำเพิ่มได้ในส่วนของ Sensor node เพื่อสำรองข้อมูลในขณะที่ระบบเครือข่ายเกิดการขัดข้องจนไม่สามารถส่งข้อมูลออกไปยังส่วน Edge computing ได้ เมื่อระบบเครือข่ายไร้สายกลับมาใช้งานได้เป็นปกติก็จะสามารถส่งข้อมูลย้อนหลังได้ ทำให้ข้อมูลการวัดช่วงนั้นไม่เกิดการขาดหาย

สามารถปรับเปลี่ยนอุปกรณ์ในส่วนของ Sensor node ตามความเหมาะสมเพื่อให้สะดวกต่อการติดตั้งในการใช้งานในพื้นที่จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Olof Ramström, “Scientific Background on the Nobel Prize in Chemistry 2019 Lithium – Ion Batteries”, The Royal Swedish Academy of Sciences, pp. 2-13, 2019.

[2] Isidor Buchmann, “Battery University”, The Cadex Headquarters, Vancouver, North America, pp. 23-41, 2017.

[3] S. Vazquez, S. M. Lukic, E. Galvan, L. G. Franquelo, and J. M. Carrasco, "Energy Storage Systems for Transport and Grid Applications", Industrial Electronics, IEEE Transactions on, vol. 57, pp. 3881-3895, 2010.

[4] Zhao, G.J, Zhao N.H, "Characteristics of an aqueous rechargeable lithium battery (ARLB)", Electrochimica Acta, vol. 52, pp. 4911-4915, 2007.

[5] ดร.นงลักษณ์ มีทอง, “วัสดุสำหรับแบตเตอรี่ชนิดลิเทียมไอออน”, ปทุมธานี: MTEC ศูนย์เทคโนโลยีโลหะและวัสดุแห่งชาติ, หน้า 52- 60, 2553.

[6] สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช , “มารู้จักแบตเตอรี่ชนิดลิเทียมไอออน”, [ระบบออนไลน์], จากhttps://www.nstda.or.th/th/nstda-knowledge/1852-lithium-ion-battery, 2557, สืบค้นเมื่อ 12 กันยายน 2562.

[7] Nissan Intelligent Mobility, “ข้อดีของแบตเตอรี่ลิเธียมไอออนแหล่งพลังงานของรถยนต์พลังงานไฟฟ้า **100%**” [ระบบออนไลน์], จาก https://www.nissan.co.th/experience-nissan/Nissan-EV/lithium-ion-battery.html, 2019, สืบค้นเมื่อ 12 กันยายน 2562.

[8] Ørjan Gjengedal, “Real time Impedance identification of Li-Polymer Battery with Kalman Filter”, Norwegian University of Science and Technology, Department of Engineering Cybernetics, pp. 39-56, 2016.

[9] LIU Ping, ZHU Hong-hui , CHEN Jie, LI Guan-yan, “A Distributed Management System for Lithium Ion Battery Pack”, School of Logistics Engineering, Wuhan university of technology, Wuhan, Hubei 430063, pp.17-33, 2016.

[10] Thomas F. Fuller, Marc Doyle, and John Newman, “Simulation and Optimization of the Dual Lithium Ion Insertion Cell”, J. Electrochem. Soc. 141, 1, doi:10.1149/1.2054684, 1994.

[11] Matana, Wiboonyasake, “ทำความรู้จักกับ Internet of Things”, [ระบบออนไลน์], จาก https://www.aware.co.th, 2020, สืบค้นเมื่อ 20 พฤษภาคม 2020.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ

[12] Margaret, Rouse, internet of things (IoT), [online], from https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT, 2020, December 15, 2020

[13] ExpresSo, “เทคโนโลยี IoT กับการอนุรักษ์พลังงาน”, [ระบบออนไลน์], from https://blog.pttexpresso.com/iot-for-energy-effieciency/, Retrieved December 15, 2020.

[14] americanox, “Wireless Sensor Network (WSN) แต่ละโหนดทำหน้าที่อะไร?” [ระบบออนไลน์], จาก https://www.techsourcegroups.com/2017/type-of-wsn-node/, 2560, สืบค้นเมื่อ 11 กันยายน 2562.

[15] Anuchit Chalothorn, “มารู้จัก Edge computing”, [ระบบออนไลน์], จาก https://thaiopensource.org/%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81-edge-computing/, 2018, สืบค้นเมื่อ 30 มกราคม 2563.

[16] VPSHISPEED, “Cloud computing คืออะไรและทำงานอย่างไร?”, [ระบบออนไลน์], from https://www.vpshispeed.com/blogs/what-is-cloud-computing-and-how-does-it-work-th/, 2018, Retrieved December 15, 2020.

[17] FACULTY OF SCIENCE KHON KAEN UNIVERSITY, “Cloud computing คืออะไร?” [ระบบออนไลน์], from https://sc2.kku.ac.th/office/sci-it/index.php/29-cloud-computing.html, 2020, Retrieved May 20, 2020.

[18] ภัทรียา แซ่คำ, “Cloud computing”, [ระบบออนไลน์], จาก https://sites.google.com/site/cloudcomputing6030024/9-2-prapheth- khxng-cloud-computing, 2020, สืบค้นเมื่อ 15 พฤษภาคม 2563.

[19] toppiiz spiiz, “ระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบ”, [ระบบออนไลน์, จาก https://www.glurgeek.com/education/, 2561,] สืบค้นเมื่อ 10 กันยายน 2562.

[20] Mallika Kliangkhlao, “CHARACTERISTICS OF UI”, [online], from https://mis.csit.sci.tsu.ac.th/mallika/UI/DL/Ch02Characteristics%20of%20UI.pdf, 2019, Retrieved February 22, 2020.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ

[21] Sonthaya Boonchan, “[Node-Red] EP0 รู้จัก Node-Red และติดตั้ง Node-Red บน Windows”, [ระบบออนไลน์], จาก https://jackrobotics.me/Node-Red-ep0-install-windows-1c836a715fe, 2019, สืบค้นเมื่อ 10 กันยายน 2562.

[22] Node-Red, “Low-code programming for event-driven applications”, [online], from https://nodered.org/, Retrieved December 19, 2019.

[23] adslthailand, “ทำความรู้จักกับ MQTT และ CoAP โปรโทคอลสำหรับรับส่งข้อมูลบนเครือข่าย IoT”, [ระบบออนไลน์], จาก http://www.adslthailand.com/post/mqtt-coap-comparison-iot-protocol, 2017, สืบค้นเมื่อ 10 กันยายน 2562.

[24] Ubidots, “Ubidots API docs”, [online], from https://ubidots.com/docs/hw/?language=ESP#retrieve-data,2020 , Retrieved February 22, 2020.

[25] Jose Garcia, “Analytics: Synthetic Variables Basics”, [online], from https://help.ubidots.com/en/articles/1767999-analytics-synthetic-variables-basics, Retrieved February 22, 2020

[26] สุชาติ อำมะเหียะ, “มาตรฐาน Bluetooth”, [ระบบออนไลน์], จาก https://sites.google.com/site/it514249125/matrthan-bluetooth,2020, สืบค้นเมื่อ 5 เมษายน 2563.

[27] ทันพงษ์ ภู่รักษ์, “หน่วยที่ 2 Arduino IDE ซอฟต์แวร์สำหรับโปรแกรมภาษา C”, [ระบบออนไลน์]. เข้าถึงได้จาก : http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_2.pdf. 2019.

[28] Huria Tarun, “High Fidelity Electrical Model with Thermal Dependence for Characterization and Simulation of High Power Lithium Battery Cells”, pp. 27-39, 2012.

[29] Taesic Kim, “Cloud-Based Battery Condition Monitoring and Fault Diagnosis Platform for Large-Scale Lithium-Ion Battery Energy Storage Systems.”, pp. 48-57, 2018

[30] Jaesik Lee, “WIRELESS BATTERY AREA NETWORK FOR A SMART BATTERY MANAGEMENT SYSTEM.” United States Patent, 2016.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ

[31] Rawat Siripokarpirom, “การเขียนโปรแกรมสำหรับบอร์ด Nucleo F401RE ด้วย STM32 CoreSupportforArduino”,[ระบบออนไลน์], fromhttps://www.iot.eng.kmutnb.ac.th/pub/docs/2017/arduino_stm32_core_nucleo_f401re/, Retrieved April 1, 2020.

[32] Arm Limited, “Nucleo-F091RC”, [online] Retrieved February 3, 2020, from https://os.mbed.com/platforms/ST-Nucleo-F091RC/, 2020.

[33] STMicroelectronics, “X-Nucleo-IDB05A1” ,[online] Retrieved April 5, 2020, from https://www.st.com/en/ecosystems/x-nucleo-idb05a1.html, 2020.

[34] Venus Solutions, “การใช้งาน Node-Red บน Raspberry Pi”, [ระบบออนไลน์] สืบค้นเมื่อ 12 ธันวาคม 2562, จาก http://www.eduthaieasyelec.com/16623242/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-Node-Red-%E0%B8%9A%E0%B8%99-raspberry-pi, 2016.

[35] Node-Red, “Node-Red-contrib-generic-ble 4.0.1”, [online], Retrieved February 7, 2020, from https://flows.nodered.org/node/Node-Red-contrib-generic-ble, 2020.

[36] Sonthaya Boonchan, “จับ Line Notify ยัด Node-Red โดยใช้ความสามารถเดิมของ Node-Red”, [ระบบออนไลน์], สืบค้นเมื่อ 1 มีนาคม 2563, จาก https://jackrobotics.me/%E0%B8%88%E0%B8%B1%E0%B8%9A-line-notify-%E0%B8%A2%E0%B8%B1%E0%B8%94-Node-Red, 2017.

[37] Supot Sae-Ea, “Node-Red SQLITE”, [ระบบออนไลน์], สืบค้นเมื่อ 1 มีนาคม 2563, จาก https://medium.com/@supotsaeea/Node-Red-sqlite-d78a57ec4561,2017.

[38] automation360blog, “การใช้งาน IOT2040 ส่งข้อมูลไปยัง Ubidots”, [ระบบออนไลน์], Retrieved February 22, 2020 , from https://automation360blog.wordpress.com, 2018.

[39] Maria, Carlina Hernandez, “Connect a Siemens SIMATIC IOT2000 to Ubidots over MQTT using NodeRED”, [online] Retrieved February 22, 2020 , from https://help.ubidots.com, 2020.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ

[40] Maria, Carlina Hernandez., “Automatically provision Devices and Variables with Ubidots API Labels”, [online], Retrieved February 22, 2020 , from https://help.ubidots.com/en/articles/1330905-automatically-provision-devices-and-variables-with-ubidots-api-labels, 2020.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ภาคผนวก ก NCR18650B Datasheet

ภาคผนวก ข SDM120CT Energy meter datasheet

ภาคผนวก ค Ubidots: Synthetic variable operations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

NCR18650B Datasheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NCR18650B Datasheet

Panasonic

Lithium Ion NCR18650B

Features & Benefits

- High energy density
- Long stable power and long run time
- Ideal for notebook PCs, boosters, portable devices, etc.

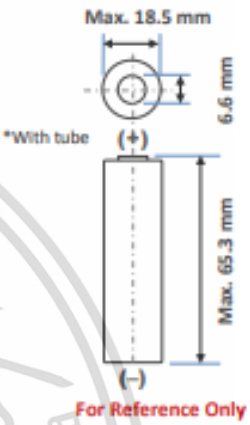
Specifications

Rated capacity[1]	Min. 3200mAh
Capacity[2]	Min. 3250mAh Typ. 3350mAh
Nominal voltage	3.6V
Charging	CC-CV, Std. 1625mA, 4.20V, 4.0 hrs
Weight (max.)	48.5 g
Temperature	Charge*: 0 to +45°C Discharge: -20 to +60°C Storage: -20 to +50°C
Energy density[3]	Volumetric: 676 Wh/l Gravimetric: 243 Wh/kg

*At temperatures below 10°C, charge at a 0.25C rate.

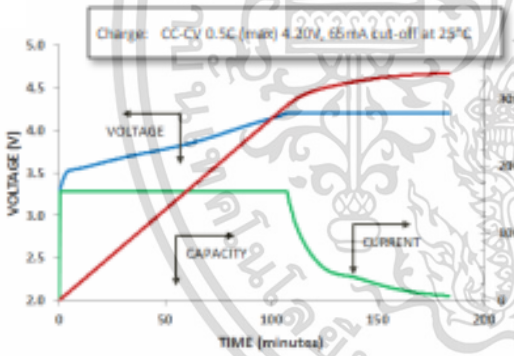
[1] At 20°C [2] At 25°C [3] Energy density based on bare cell dimensions

Dimensions

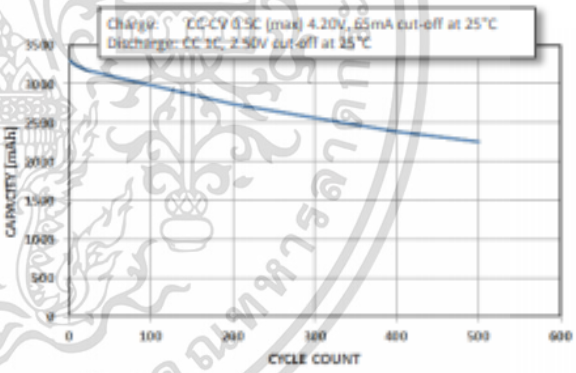


For Reference Only

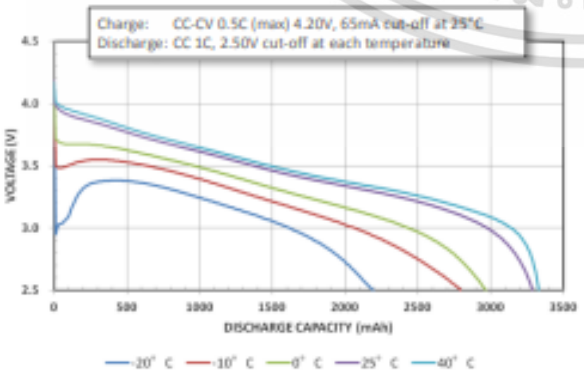
Charge Characteristics



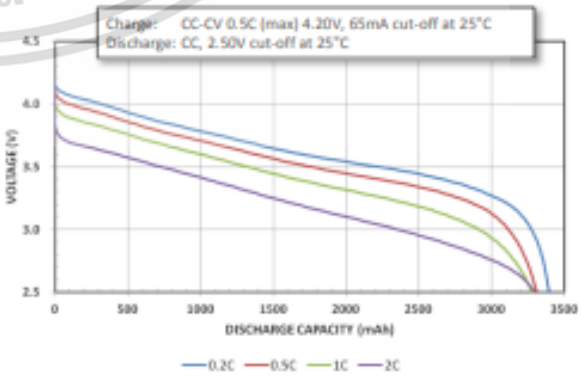
Cycle Life Characteristics



Discharge Characteristics (by temperature)



Discharge Characteristics (by rate of discharge)



The data in this document is for descriptive purposes only and is not intended to make or imply any guarantee or warranty.

For more information on how Panasonic can assist you with your battery power solution needs, visit us at www.panasonic.com/industrial/batteries-oem, e-mail secsales@us.panasonic.com, or call (469) 362-5600.

VERSION 13.11 R1 | Copyright© 2012 SANYO Energy (U.S.A.) Corporation. All Rights Reserved.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

SDM120CT Energy meter datasheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SDM120CT Energy meter datasheet

Eastron (Metering) Europe Limited
3rd Floor, 207 Regent Street,
London
W1B 3HH
Customer Service: 0208 985 1549
www.eastroneurope.com

SDM120CT Protocol

1. SDM120CT Protocol Implementation
1.1 Modbus Protocol Overview
1.2 Input register
1.2.1 SDM120CT Input Registers
1.3 Modbus Protocol Holding Registers and Digital meter set up:
1.3.1 MODBUS Protocol Holding Register Parameters

2 RS485 General Information
2.1 Half Duplex
2.2 Connecting the Instruments
2.3 A and B terminals
2.4 Troubleshooting

3 MODBUS Protocol General Information
3.1 MODBUS Protocol Message Format
3.2 Serial Transmission Modes
3.3 MODBUS Protocol Message Timing (RTU Mode)
3.4 How Characters are Transmitted Serially
3.5 Error Checking Methods
3.5.1 Parity Checking
3.5.2 CRC Checking
3.6 Function Codes
3.7 IEEE floating point format
3.8 MODBUS Protocol Commands supported
3.8.1 Read Input Registers
3.9 Holding Registers
3.9.1 Read Holding Registers
3.9.2 Write Holding Registers
3.10 Exception Response
3.11 Exception Codes
3.11.1 Table of Exception Codes
3.12 Diagnostics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SDM120CT Input registers

Address (Register)	SDM120CT Input Register Parameter		Modbus Protocol Start Address Hex	
	Description	Units	Hi Byte	Lo Byte
30001	Voltage	Volts	00	00
30007	Current	Amps	00	06
30013	Active Power	Watts	00	0C
30019	Apparent Power	VoltAmps	00	12
30025	Reactive Power	VAr	00	18
30031	Power Factor	None	00	1E
30037	Phase Angle	Degrees	00	24
30071	Frequency	Hz	00	46
30073	Import Active Energy	kWh	00	48
30075	Export Active Energy	kWh	00	4A
30077	Import Reactive Energy	kVArh	00	4C
30079	Export Reactive Energy	kVArh	00	4E
30343	Total Active Energy	kWh	01	56
30345	Total Reactive Energy	kVArh	01	58

MODBUS Protocol holding register parameters

Address Register	Parameter	Modbus Protocol Start Address Hex		Valid range	Mode
		High Byte	Low Byte		
40013	Relay Pulse Width	00	OA	Write relay on period in milliseconds: 60, 100 or 200, default 200.	r/w
40019	Network Parity Stop	00	12	Write the network port parity/stop bits for MODBUS Protocol, where: 0 = One stop bit and no parity, default. 1 = One stop bit and even parity. 2 = One stop bit and odd parity.3 = Two stop bits and no parity. Requires a restart to become effective.	r/w
40021	Network Node	00	14	Write the network port node address: 1 to 247 for MODBUS Protocol, default 1. Requires a restart to become effective. Note, both the MODBUS Protocol and Johnson Controls node addresses can be changed via the display setup menus.	r/w
40029	Network Baud Rate	00	1C	Write the network port baud rate for MODBUS Protocol, where: 0 = 2400 baud. 1 = 4800 baud. 2 = 9600 baud, default. 3 = 19200 baud. 4 = 38400 baud. Requires a restart to become effective	r/w
462721	Demand Interval, Slide Time, Automatic Scroll Display Interval (Scroll Time), Backlight Time	F5	00	Data Format: BCD min-min-s-min Scroll Time=0: the display does not scroll automatically Backlight Time=0: Backlight is Always On.	r/w
463761	System Power	F9	10	Default Format: Hex 0000: 0.001kWh (kVArh) /imp (default) 0001: 0.01kWh (kVArh) /imp 0002: 0.1kWh (kVArh) /imp 0003: 1kWh (kVArh) /imp	r/w
463776	Measurement Mode	F9	20	Data Format: Hex 0001: Mode 1 (Total = Import) 0002: Mode 2 (Total = Import + Export) 0003: Mode 3 (Total = Import - Export)	r/w
363792	Pulse Output & LED Indicator Mode	F9	30	Data Format: Hex 0000: Import & Export Energy, LED flashes for Import & Export Energy 0001: Import Energy, LED flashes for Import Energy only 0002: Export Energy, LED flashes for Export Energy only	ro

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

Ubidots : Synthetic variable operations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ubidots

Synthetic Variable Operations

Update date: November 13, 2019

Operator / Function	Description	Real Time Capability	Example	Result
Rounding				
`ceil({{raw}})`	Returns the rounded integer greater or equal to every value in a time series, as a integer. Ceil always rounds up to the nearest integer.	NO	`ceil([1.2, 2, 3.7])`	`[2,2,4]`
`floor({{raw}})`	Returns the rounded integer less or equal to every value in the time series, as a integer. Floor always round down to the nearest integer.	NO	`floor([1.2, 2, 3.7])`	`[1,2,3]`
`round({{raw}}, n)`	Returns the floating point value rounded to the "n" digits after the decimal point for every value in a time series. By default, n equals 2 in Ubidots.	NO	`round([1.22222, 2.9994332], 3)`	`[1.222, 2.999]`
Trigonomic Functions				
`tan({{raw}})`	Returns the tangent of every value in radians in a time series.	NO	`tan([0, 90])`	`[0, -1.99520041221]`
`cos({{raw}})`	Returns the cosine of every value in radians in a time series.	NO	`cos([0, 90])`	`[1, -0.44807361612]`
`sin({{raw}})`	Returns the sine of every value in radians in a time series.	NO	`sin([0, 90])`	`[0, 0.8939966636]`
`arcsin({{raw}})`	Returns in radians the inverse sine of every value in the time series.	NO	`arcsin([0, 90])`	`[0]`
`arccos({{raw}})`	Returns in radians the inverse cosine of every value in the time series.	NO	`arccos([0, 90])`	`[1.5707963]`
`arctan({{raw}})`	Returns in radians the inverse tangent of every value in the time series.	NO	`arctan([0, 90])`	`[0]`
`arctan2({{raw_x}}, {{raw_y}})`	Returns in radians the trigonometric inverse tangent using as cartesian coordinates the input time series. Note: Will only perform the operation between values with the same timestamp.	NO	Assuming that the time series is sampled every minute `arctan2([1, 2], [0.1, 1])`	`[1.471127, 1.1071]`
Mathematical Functions				
`sinh({{raw}})`	Returns the hyperbolic sine of every value in the time series.	NO	`sinh([0, 90])`	`[0, 6.1020exp38]`
`cosh({{raw}})`	Returns the hyperbolic cosine of every value in the time series.	NO	`cosh([0, 90])`	`[1, 6.1020exp38]`
`tanh({{raw}})`	Returns the hyperbolic tangent of every value in the time series.	NO	`tanh([0, 90])`	`[0, 1]`
`arcsinh({{raw}})`	Returns in radians the inverse hyperbolic sine of every value in the time series.	NO	`arcsinh([0, 90])`	`[0, 5.19298771365B9]`
`arccosh({{raw}})`	Returns in radians the inverse hyperbolic cosine of every value in the time series.	NO	`arccosh([0, 90])`	`[5.19298771365B9]`
`arctanh({{raw}})`	Returns in radians the inverse hyperbolic tangent of every value in the time series.	NO	`arctanh([0, 90])`	`[0]`
`exp({{raw}})`	Returns the exponential of every value in the time series.	NO	`exp([-1, 0, 1, 2])`	`[0.36787944117144233, 1.0, 2.718281828459045, 7.389056098930365]`
`log({{raw}}, base)`	Returns the logaritm of every value in the time series. By default the base is the Euler's number.	NO	`log([1, 2])`	`[0, 0.6931471805599]`
`abs({{raw}})`	Returns the absolute value of every data in the time series.	NO	`abs([-1, 0, 1, 2])`	`[1, 0, 1 ,2]`
`sqrt({{raw}})`	Returns the square root value of every data in the time series	NO	`sqrt[1,4]`	`[1,2]`
Data Range Functions				
`max({{raw}}, "data_range")`	Calculates the maximum value of the variable in the specified data range.	NO	Assuming that values are sampled every second: `max([1, 2, 3, -1], "T")`	`3`
`min({{raw}}, "data_range")`	Calculates the minimum value of the variable in the specified data range.	NO	Assuming that values are sampled every hour: `min([1, 2, 3, -1], "4H")`	`-1`
`std({{raw}}, "data_range")`	Calulates the standard deviation of the variable in the specified data range.	NO	Assuming that values are sampled every minute: `std([1, 2], "T")`	`[0.707]`
`mean({{raw}}, "data_range")`	Calculates the mean value of the variable in the specified data range.	NO	Assuming that values are sampled every minute: `mean([1, 2, 3, 0, -1], "5T")`	`1`
`median({{raw}}, "data_range")`	Calculates the median value of the variable in the specified data range.	NO	Assuming that values are sampled every hour: `median([1, 2, 3, 0, -1], "D")`	`1`
`count({{raw}}, "data_range")`	Calculates the number of dots in the specified data range.	NO	Assuming that values are sampled every day: `count([1, 2, 3, 0, -1], "W")`	`5`

`last({{raw}}, "data_range")`	Calculates the last value of the variable in the specified data range.	NO	Assuming that values are sampled every month: `last([1, 2, 3, 0, -1], "4M")`	-1
`sum({{raw}}, "data_range")`	Calculates the summation of the time series in the specified data range.	NO	Assuming that values are sampled every minute: `sum([1, 2, 3, 0, -1], "4T")`	5

Operator / Function	Description		Example	Result
Available Data Ranges				
`"nT"`	Returns a value representing a data range of every "n" number of MINUTE(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every second: `max([1, 2, 3, 0, -1], "T")`	3
`"nH"`	Returns a value representing a data range of every "n" number of HOUR(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every minute: `min([1, 2, 3, 0, -1], "H")`	-1
`"nD"`	Returns a value representing a data range of every "n" number of DAY(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every hour: `mean([1, 2, 3, 0, -1], "D")`	1
`"nW"`	Returns a value representing a data range of every "n" number of WEEK(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every day: `count([1, 2, 3, 0, -1], "W")`	5
`"nM"`	Returns a value representing a data range of every "n" number of MONTH(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every week: `last([1, 2, 3, 0, -1], "M")`	-1
Special Functions				

`where(condition,`
`operation if fits,`
`operation if not fits)`
	If-else statement. Variables attributes like context key or timestamp can be accessed using the dot, '.', operator	NO	Assuming that {{raw}} time series is equals to [-1, 2, 1]
			Step Function, unit(x): `where({{raw}}>=0, 1, 0)`
			Interval function: `where({{raw}}<0, 0, where({{raw}}<1), 1, 2)`
				[0, 1, 1]
				[0, 2, 2]

`fill_missing(x)`	When performing operations between multiple variables timestamps, the function will enter the last value of a variable when an expression requires data from a timestamp that does not match the other timestamps within the expression. Note: fill_missing() computes the operation between the whole time series once a new value arrives to any of the raw variables in the operation. Ubidots does not advise using this operation for real time applications.	NO	{{raw_1}} = [1, 2, 3], sampled every minute {{raw_2}} = [1, 5], sampled every 2 minutes `fill_missing({{raw_1}} + {{raw_2}})`	[2, 3, 8]

`fill_missing(x, first_fill="ffill", last_fill=None, fill_value=None)`	fill_missing recives additional parameter to fill gaps forward, backward or with a value. **"ffill"**: Fill gaps forward "bfill": Fill gaps backward and **"fill_value = 0"**: Fill gaps with the number entered.	NO	{{raw_1}} = [1, 2, 3], sampled every minute
			{{raw_2}} = [1, 5], sampled every 2 minutes
			`fill_missing({{raw_1}} + {{raw_2}}, first_fill="ffill")`
			`fill_missing({{raw_1}} + {{raw_2}}, first_fill="bfill")`
			`fill_missing({{raw_1}} + {{raw_2}}, fill_value=0)`
				[2, 3, 8]
				[2, 7, 8]
				[2, 2, 8]

`shift({{raw}}, n)`	Returns values of the variable by the specified number of (+/-) n steps in the time series. Note: N must be entered as an integer, not a string.	NO	`shift([-2, -1, 0, 1, 2], 1)`
			`shift([-2, -1, 0, 1, 2], 2)`
			actual value minus previous value:
			`[0, 1, 3] - shift([0, 1, 3], -1)`
				[-1, 0, 1, 2]
				[0, 1, 2]
				[1, 2]

`cumsum({{raw}})`	Calculates the cumulative sum of the time series.	NO	`cumsum[0,1,2,3}`	6