ระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์
AN ELECTRICITY BILL MANAGEMENT SYSTEM FOR APARTMENTS

นภาพร ศาสตร์สง่า
รัตนาพร แย้มเกษม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

ระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์

AN ELECTRICITY BILL MANAGEMENT SYSTEM FOR APARTMENTS

นภาพร ศาสตร์สง่า
รัตนาพร แย้มเกษม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AN ELECTRICITY BILL MANAGEMENT SYSTEM FOR APARTMENTS

NAPAPORN SARTSANGAR

RATTNAPORN YAEMKASEM

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN CONTROL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์

AN ELECTRICITY BILL MANAGEMENT SYSTEM FOR APARTMENTS

ผู้จัดทำ นางสาวนภาพร ศาสตร์สง่า 58010644

นางสาวรัตนาพร แย้มเกษม 58011065

.....อาจารย์ที่ปรึกษา

(ดร.พิชชา ประสิทธิ์มีบุญ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์

โดย
นางสาวนภาพร ศาสตร์สง่า 58010644
นางสาวรัตนาพร แย้มเกษม 58011065

อาจารย์ที่ปรึกษา
ดร.พิชชา ประสิทธิ์มีบุญ

ปีการศึกษา 2561

บทคัดย่อ

วัตถุประสงค์ของงานวิจัยนี้เพื่อศึกษา และพัฒนาระบบคำนวณค่าไฟฟ้าสำหรับการจัดการอะพาร์ตเมนต์ เนื่องจากในปัจจุบันผู้ใช้อะพาร์ตเมนต์ไม่สามารถทราบค่าใช้ไฟฟ้าในแต่ละเดือนจนกว่าจะถึงปลายเดือนได้ และในบางครั้งอาจจะมีการใช้ไฟฟ้าเกินกว่าที่ได้ตั้งเป้าหมายไว้ ผู้จัดทำได้พัฒนาระบบคำนวณค่าไฟอัจฉริยะเพื่อช่วยให้ผู้ที่อยู่อาศัยอะพาร์ตเมนต์สามารถติดตามค่าไฟฟ้าแบบ Real-time ผ่านทางแอปพลิเคชัน รวมถึงสามารถวางแผนใช้ไฟฟ้าในแต่ละเดือนได้ โดยระบบเริ่มจากการอ่านค่าจากมิเตอร์ และส่งข้อมูลหน่วยไฟฟ้าด้วยรูปแบบการสื่อสารแบบ RS-485 ไปยังคลาวด์เพื่อคำนวณค่าไฟ และแสดงผลผ่านทางแอปพลิเคชันบนสมาร์ทโฟน แอปพลิเคชันจะมีฟังก์ชันให้ผู้พักอาศัยสามารถตั้งเป้าหมายการใช้งานไฟฟ้า แจ้งเตือนเมื่อมีการใช้ไฟฟ้าเกินค่าที่ตั้งไว้ ผู้จัดทำหวังว่าระบบการคำนวณค่าไฟฟ้าอัจฉริยะนี้ จะอำนวยความสะดวกให้กับผู้ใช้อะพาร์ตเมนต์ รวมถึงช่วยส่งเสริมการประหยัดไฟได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AN ELECTRICITY BILL MANAGEMENT SYSTEM FOR APARTMENTS

By

Miss. Napaporn Sartsangar 58010644

Miss. Rattanaporn Yaemkasem 58011065

Advisor

Dr. Pitcha Prasitmeeboon

Academic Year 2018

ABSTRACT

The objective of this research is to study and develop electricity bill management system for apartment. Nowadays apartment users cannot know electricity bills until the end of month and sometimes they use electricity beyond limits. Our team developed the electricity bill management system for apartment usera to check real-time electricity usage via application and plan amount of electricity usage in each month. The system reads units of electricity usage and sends to the application to calculate and display the real-time electricity usage. The application allows users to set maximum electricity usage and sends notifications whwn thw electiricity usage reaches the targeted limit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ประสบความสำเร็จลุล่วงไปได้ด้วยดี อันเนื่องมาจากความกรุณาของอาจารย์ที่ปรึกษา ดร.พิชชา ประสิทธิ์มีบุญ ที่ได้ให้ความรู้ คำปรึกษา ชี้แนะแนวทางการแก้ปัญหาและประสบการณ์ที่ดีแก่ข้าพเจ้าตลอดระยะเวลาที่ได้ทำการศึกษาจัดทำปริญญานิพนธ์นี้

ขอขอบคุณ ผู้ช่วยศาสตราจารย์อภินัย ฤกษ์รัตน์ ที่ให้คำปรึกษาและคำแนะนำเกี่ยวกับการอ่านค่าจากมิเตอร์ไฟฟ้าและรูปแบบการสื่อสารในอุตสาหกรรม

ขอขอบคุณ ผู้ช่วยศาสตราจารย์กฤษณ์ เสมอพิทักษ์ ที่ให้คำปรึกษาและคำแนะนำเกี่ยวกับการเขียนโปรมแกรมสำหรับไมโครคอนโทรลเลอร์

ขอขอบคุณผู้แต่งหนังสือ เอกสารอ้างอิง และเว็บไซต์ต่างๆ ที่คณะผู้จัดทำได้นำมาใช้อ้างอิงประกอบการศึกษา และจัดทำปริญญานิพนธ์ฉบับนี้จนสำเร็จลุล่วงไปด้วยดี

สำหรับคุณงามความดีอันใดที่เกิดจากปริญญานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

คณะผู้จัดทำ
นภาพร ศาสตร์สง่า
รัตนาพร แย้มเกษม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VIII
สารบัญตาราง	XI
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของโครงงาน	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 รายละเอียดของปริญญานิพนธ์	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	4
2.1 MODBUS Protocol	4
2.1.1 โหมด RTU	5
2.1.2 โหมด ASCII	6
2.2 ระบบปฏิบัติการแอนดรอยด์ (Android)	6
2.3 แอปพลิเคชัน (Application)	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.4 RS-485	8
2.5 TTL (Transistor-Transistor Logic)	9
2.6 ESP32	9
2.7 Switching Power Supply	11
2.8 Safety Breaker	13
2.9 สายไฟประธานและสายไฟโหลด	13
2.10 Arduino IDE	14
2.11 Cloud Storage System	15
2.11.1 กระบวนการเก็บข้อมูลบน Cloud	15
2.11.2 Firebase	16
2.12 MIT App Inventor	17
2.13 การคำนวณค่าไฟฟ้า	18
2.14 วิธีใช้ไฟฟ้าอย่างประหยัด	18
บทที่ 3 วิธีการดำเนินงาน	24
3.1 ขั้นตอนการดำเนินงาน	24
3.2 อุปกรณ์ที่เกี่ยวข้อง	26
3.2.1 มิเตอร์ Mitsubishi รุ่น SX1-A31E	26
3.2.2 MAX13487	27
3.2.3 ESP32	28
3.2.4 ปลั๊กสามตา	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.2.5 Switching Power Supply 3V 5A	28
3.2.6 Safety Breaker 15A	29
3.3 ซอฟต์แวร์ที่เกี่ยวข้อง	29
3.3.1 โปรแกรม Arduino IDE	29
3.3.2 Firebase	29
3.3.3 MIT App Inventor	30
3.4 การออกแบบและการวางแผนการทำงาน	30
3.4.1 การออกแบบและวางแผนทางด้านฮาร์ดแวร์	30
3.4.2 การออกแบบและวางแผนทางด้านซอฟต์แวร์	30
3.5 วิธีการดำเนินงาน	31
3.5.1 การต่อสายไฟเข้ามิเตอร์	31
3.5.2. การออกแบบระบบจำลอง	31
3.5.3 การติดตั้งบอร์ดที่ใช้ลงโปรแกรม Arduino IDE	33
3.5.4 วิธีการต่อบอร์ด ESP32 เข้ากับเครือข่าย Wi-Fi	34
3.5.5 การติดตั้งระบบ Firebase	35
3.5.6 โปรแกรมที่ใช้ในการดึงค่าจากมิเตอร์ไปแสดงบน Firebase	37
3.5.7 การเขียนแอปพลิเคชัน	41
บทที่ 4 ผลการดำเนินงาน	46
4.1 แผนผังการทำงานของระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์	46
4.2 ผลการออกแบบ	46
4.2.1 การออกแบบระบบจำลอง	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
4.2.2 การออกแบบแอปพลิเคชัน	47
4.3 ผลการสร้างระบบจำลอง	47
4.4 ผลการเขียนโปรแกรมสำหรับอ่านค่าจากมิเตอร์และแสดงผลใน Firebase	48
4.5 ผลการสร้างแอปพลิเคชันบนสมาร์ทโฟน	49
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	51
5.1 สรุปผลการดำเนินงาน	51
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	51
5.3 แนวทางการแก้ไขปัญหา	52
เอกสารอ้างอิง	53
ภาคผนวก	55
ภาคผนวก ก Program	55
ภาคผนวก ข Electronic Meter	59
ภาคผนวก ค ESP32	66
ภาคผนวก ง Poster	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 การติดต่อสื่อสารแบบ Master/Slave	4
2.2 ลักษณะเฟรมข้อมูลของ MODBUS RTU	5
2.3 ลักษณะข้อมูลแต่ละไบต์ของ MODBUS RTU	5
2.4 ลักษณะเฟรมข้อมูลของ MODBUS ASCII	6
2.5 การเชื่อมต่อ RS-485 ระหว่างเครื่องมือวัดกับตัวแปลงสัญญาณ	9
2.6 กราฟตัวอย่างการส่งข้อมูลแบบ TTL 0-5V และ TTL 0-3.3V	9
2.7 แต่ละพินของ ESP32	11
2.8 แผนผัง Switching Power Supply	12
2.9 Safety Breaker ที่ใช้	13
2.10 ผังการต่อสายประธานเข้ามิเตอร์และต่อโหลดออกจากมิเตอร์	14
2.11 หน้าโปรแกรม Arduino IDE	14
2.12 กระบวนการเก็บใน Cloud	15
2.13 Firebase	16
2.14 ภาพรวมของการเขียนแอปพลิเคชันด้วย MIT App Inventor	17
3.1 มิเตอร์ Mitsubishi รุ่น SX1-A31E	26
3.2 ลักษณะการสื่อสาร (การสื่อสารมาตรฐาน)	27
3.3 โมดูลแปลง TTL เป็น RS-485 (ชิพ MAX 13487)	27
3.4 ESP32	28
3.5 ปลั๊กสามตาที่ใช้	28
3.6 Switching Power Supply	28
3.7 Safety Breaker 15A	29
3.8 โปรแกรม Arduino IDE	29
3.9 Console of Firebase	30
3.10 MIT App Inventor	30
3.11 การต่อสายไฟประธานและสายไฟโหลด	31
3.12 ต่อสายไฟเข้ามิเตอร์ และออกจากมิเตอร์	31
3.13 แผนผังการส่งข้อมูลของระบบจำลอง	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.14 อุปกรณ์ส่วนฮาร์ดแวร์	32
3.15 อุปกรณ์ส่วนฮาร์ดแวร์ที่ประกอบเสร็จเรียบร้อย	32
3.16 หน้าโปแกรม Arduino IDE	33
3.17 การติดตั้งบอร์ด ESP32 (1)	33
3.18 การติดตั้งบอร์ด ESP32 (2)	33
3.19 การติดตั้งบอร์ด ESP32 (3)	34
3.20 การติดตั้งบอร์ด ESP32 (4)	34
3.21 การติดตั้งบอร์ด ESP32 (5)	34
3.22 หน้า Monitor ที่แสดงว่า ESP32 ต่อเครือข่าย Wi-Fi เรียบร้อยแล้ว	35
3.23 Setup Firebase (1)	35
3.24 Setup Firebase (2)	35
3.25 Setup Firebase (3)	36
3.26 Setup Firebase (4)	36
3.27 ลิงก์ที่จะต้องนำมาใช้ในการเขียนโปรแกรม	36
3.28 โปรแกรมดึงค่าไปยัง Firebase (1)	37
3.29 โปรแกรมดึงค่าไปยัง Firebase (2)	37
3.30 โปรแกรมดึงค่าไปยัง Firebase (3)	38
3.31 โปรแกรมดึงค่าไปยัง Firebase (4)	39
3.32 โปรแกรมดึงค่าไปยัง Firebase (5)	39
3.33 โปรแกรมดึงค่าไปยัง Firebase (6)	40
3.34 โปรแกรมดึงค่าไปยัง Firebase (7)	40
3.35 เขียนแอปพลิเคชัน (1)	41
3.36 เขียนแอปพลิเคชัน (2)	41
3.37 เขียนแอปพลิเคชัน (3)	42
3.38 เขียนแอปพลิเคชัน (4)	42
3.39 เขียนแอปพลิเคชัน (5)	43
3.40 เขียนแอปพลิเคชัน (6)	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.41 เขียนแอปพลิเคชัน (7)	44
3.42 เขียนแอปพลิเคชัน (8)	45
4.1 แผนผังการทำงานของระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์	46
4.2 การออกแบบระบบจำลอง	46
4.3 การออกแบบแอปพลิเคชัน	47
4.4 ระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์จำลอง	48
4.5 อ่านค่าจากมิเตอร์และแสดงผลใน Firebase	48
4.6 หน้า Login	49
4.7 หน้าแสดงหน่วยไฟฟ้าสะสมตั้งแต่ต้นเดือนถึง ณ ปัจจุบันและคำนวณค่าไฟฟ้าต่อหน่วย	49
4.8 หน้าแสดงกำลังไฟฟ้า ถึง ณ ปัจจุบันและคำนวณค่าไฟฟ้าต่อวัน ต่อเดือนและต่อปี	50
4.9 หน้าแสดงคำแนะนำวิธีการประหยัดค่าไฟฟ้า	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 ขั้นตอนการทำงาน	24
3.2 คุณสมบัติเฉพาะทางเทคนิค	26
3.3 ข้อมูล Register Address ของมิเตอร์	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

เนื่องด้วยเล็งเห็นถึงจำนวนการเพิ่มขึ้นของอะพาร์ตเมนต์อย่างต่อเนื่อง และในปัจจุบันยังมีอะพาร์ตเมนต์ที่อยู่ระหว่างการก่อสร้างอีกจำนวนมาก จากข้อมูลสถิติการจดทะเบียนอาคารชุด หรืออะพาร์ตเมนต์ ในฐานะผู้เช่าได้ประสบปัญหาเรื่องการชำระค่าไฟฟ้า เนื่องจากค่าไฟฟ้ามากกว่าที่คาดการณ์ไว้ โดยผู้เช่าไม่สามารถทราบได้ว่าทางเจ้าของทำการจดบันทึกหรือคำนวณค่าไฟฟ้าถูกต้องหรือไม่ และผู้เช่าไม่สามารถทราบค่าไฟฟ้าแบบ Real-time จึงทำให้ไม่สามารถวางแผนการเงินได้ จึงเล็งเห็นช่องทางการพัฒนาระบบคำนวณค่าไฟสำหรับอะพาร์ตเมนต์ ให้มีความสะดวก สบาย มากยิ่งขึ้น ทั้งนี้สมาร์ทโฟนเป็นอุปกรณ์ที่ปัจจุบันคนจำนวนใช้งานกันอยู่แล้ว และยังมีแนวโน้มการใช้งานที่เพิ่มขึ้นอีก สมาร์ทโฟนมีแอปพลิเคชันที่ช่วยอำนวยความสะดวกได้ในหลายด้าน รวมถึงผู้ใช้ยังสามารถสร้างแอปพลิเคชันเองได้ตรงกับจุดประสงค์ที่ต้องการได้อีกด้วย ด้วยเหตุผลดังที่ได้กล่าวจึงเป็นที่มาของงานวิจัยนี้ ที่เป็นการออกแบบระบบคำนวณค่าไฟสำหรับอะพาร์ตเมนต์ ซึ่งทำให้ผู้เช่าสามารถทราบข้อมูลค่าไฟฟ้าผ่านแอปพลิเคชันที่แสดงผลแบบ Real-time และหวังเป็นอย่างยิ่งว่างานวิจัยชิ้นนี้จะสามารถช่วยลดปัญหาที่เกิดในผู้ใช้อะพาร์ตเมนต์ได้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

เพื่อศึกษาและพัฒนาระบบคำนวณค่าไฟฟ้า สำหรับการจัดการอะพาร์ตเมนต์ เนื่องจากในปัจจุบัน ผู้ใช้อะพาร์ตเมนต์ไม่สามารถทราบค่าไฟฟ้าที่ใช้ในแต่ละเดือนได้จนกว่าจะถึงปลายเดือน และบางครั้งอาจจะมีการใช้ไฟฟ้าเกินกว่าที่ได้ตั้งเป้าหมายไว้ ได้พัฒนาระบบคำนวณค่าไฟฟ้าเพื่อช่วยให้ผู้ที่อาศัยอยู่อะพาร์ตเมนต์สามารถติดตามค่าไฟฟ้าแบบ Real-time ผ่านทางแอปพลิเคชัน รวมถึงสามารถวางแผนใช้ไฟฟ้าในแต่ละเดือนได้

1.3 ขอบเขตของโครงงาน

1. ศึกษาการสร้างระบบไฟฟ้าจำลองสำหรับห้องพัก 2 ห้อง
2. ศึกษาการรับ-ส่ง สัญญาณจากอุปกรณ์วัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ศึกษาการเก็บ Data Base

4. ศึกษาการเขียนแอปพลิเคชันบนระบบปฏิบัติการ Android

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาค้นคว้าสิ่งที่สนใจเพื่อใช้กำหนดหัวข้อโครงงาน
2. นำเสนอหัวข้อโครงงาน
3. ศึกษาและรวบรวมข้อมูลการเก็บข้อมูลจากมิเตอร์ไฟฟ้า
4. ศึกษาข้อมูลการเก็บข้อมูล และการออกแบบแอปพลิเคชัน
5. ออกแบบระบบจำลองและจัดซื้อของ
6. ออกแบบแอปพลิเคชัน
7. แก้ไขขอบเขตการทำงาน
8. ศึกษาข้อมูลการเขียนโปรแกรมสำหรับ ESP32 และ MAX13487 เพิ่มเติม
9. เขียนโปรแกรมสำหรับ ESP32
10. จัดทำระบบจำลอง
11. สร้างแอปพลิเคชัน
12. ทดสอบระบบที่พัฒนาในแต่ละส่วนและแก้ปัญหาที่เกิดขึ้นจากการทดสอบ
13. รายงานผลกับอาจารย์ที่ปรึกษา
14. จัดทำรูปเล่มรายงานและนำเสนอ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้อาศัยอะพาร์ตเมนต์สามารถดูหน่วยไฟที่ใช้ได้แบบ Real-time
2. แอปพลิเคชันสามารถคำนวณค่าไฟฟ้าได้จากกำลังไฟฟ้าที่ใช้ ณ ปัจจุบันได้
3. เจ้าของอะพาร์ตเมนต์สามารถดูค่าไฟฟ้าของแต่ละห้องได้จากแอปพลิเคชัน
4. ได้ความรู้ด้านต่างๆ ที่ต้องนำมาใช้กับการสร้างระบบจำลองนี้
5. สามารถนำระบบจำลองไปพัฒนาต่อได้ เช่น พัฒนาเป็นระบบคิดค่าน้ำหรือคำนวณหาทั้งค่าน้ำและค่าไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย 5 บท และ 5 ภาคผนวก ซึ่งมีรายละเอียดดังต่อไปนี้

บทที่ 1 บทนำ เป็นการกล่าวถึงที่มาของปริญญานิพนธ์ วัตถุประสงค์ของการทำปริญญานิพนธ์ ขอบเขตของโครงงาน ขั้นตอนการดำเนินงาน ประโยชน์ที่คาดว่าจะได้รับและรายละเอียดของปริญญานิพนธ์

บทที่ 2 ทฤษฎี หลักการ อุปกรณ์ และความรู้ที่เกี่ยวข้องในการออกแบบ เป็นการศึกษาความรู้ทฤษฎีและความเข้าใจในอุปกรณ์ต่างๆ ก่อนที่จะเริ่มการทำโครงงาน

บทที่ 3 วิธีการดำเนินงาน จะเกี่ยวกับขั้นตอนการดำเนินงาน อุปกรณ์ที่เกี่ยวข้อง ซอฟต์แวร์ที่เกี่ยวข้อง การออกแบบและวางแผนการทำงานด้านซอฟต์แวร์และด้านฮาร์ดแวร์ วิธีการดำเนินงานและการเขียนโปรแกรม

บทที่ 4 ผลการดำเนินงาน แผนผังการทำงานของระบบจำลอง รูปแบบระบบและแอปพลิเคชัน ผลการสร้างระบบจำลอง ผลการเขียนโปรแกรมอ่านค่าจากมิเตอร์ และผลการสร้างแอปพลิเคชันบนสมาร์ทโฟน

บทที่ 5 ผลสรุปและข้อเสนอแนะ

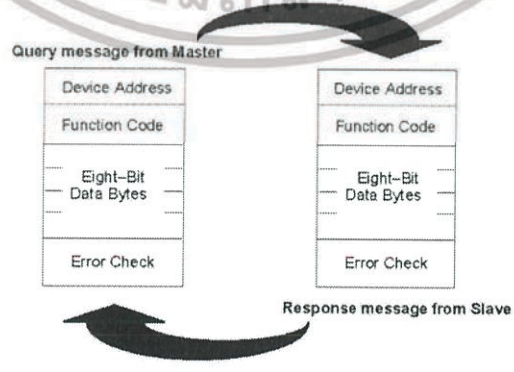
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 MODBUS Protocol

MODBUS Protocol เป็นรูปแบบการสื่อสารข้อมูลดิจิตอลแบบอนุกรม ซึ่งถูกเผยแพร่ครั้งแรกในปี ค.ศ. 1979 โดย Modicon (ปัจจุบันคือ บริษัท Schneider Electric) จนกลายมาเป็นโปรโตคอลที่ได้รับความนิยม และถูกใช้เป็นโปรโตคอลมาตรฐานในระบบอิเล็กทรอนิกส์อุตสาหกรรมจนถึงปัจจุบัน โปรโตคอล MODBUS เป็นการสื่อสารข้อมูลในลักษณะ Master/Slave ซึ่งเป็นการสื่อสารจาก Master เครื่องเดียว ส่วนใหญ่มักเป็นซอฟต์แวร์คอมพิวเตอร์หรืออุปกรณ์แสดงผล HMI ไปยัง Slave หลายๆ เครื่อง โดยสามารถกำหนดหมายเลขอุปกรณ์ได้สูงสุด 255 เครื่อง การรับส่งข้อมูลด้วยโปรโตคอล MODBUS สามารถเลือกได้ 2 โหมดคือ โหมด ASCII และโหมด RTU ซึ่งทั้งสองโหมดนี้มีความแตกต่างกันที่การกำหนดรูปแบบของชุดข้อมูลภายในเฟรม จะเลือกโหมดใดก็ได้ แต่มีเงื่อนไขว่าอุปกรณ์ทุกตัวที่ต่อร่วมกันอยู่ในบัสหรือเครือข่ายเดียวกันจะต้องตั้งให้เลือกใช้โหมดเดียวกันทั้งหมด

ปัจจุบันโปรโตคอล MODBUS เป็นที่ยอมรับและถูกใช้งานอย่างแพร่หลายทั่วโลก ถือเป็นโปรโตคอลมาตรฐานที่อุปกรณ์ด้านอุตสาหกรรม นิยมนำมาใช้กัน เช่น Meter, PLC, SCADA เป็นต้น ปัจจุบันมีผู้ดูแลคือ Modbus Organization (www.modbus.org) MODBUS เป็นโปรโตคอลที่ถูกพัฒนาเพื่อใช้งานในอุตสาหกรรม และเป็นโปรโตคอลเปิดสามารถนำมาใช้งานโดยไม่มีค่าใช้จ่าย ทั้งยังเป็นโปรโตคอลที่ง่ายต่อการพัฒนาระบบเพื่อเชื่อมต่อกับอุปกรณ์ MODBUS (PONGSAKRIVERPLUS, 2011) ดังรูปที่ 2.1

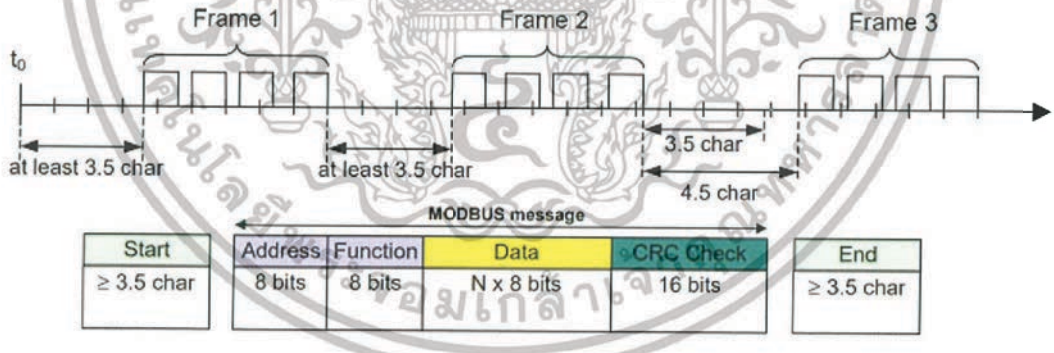


รูปที่ 2.1 การติดต่อสื่อสารแบบ Master/Slave

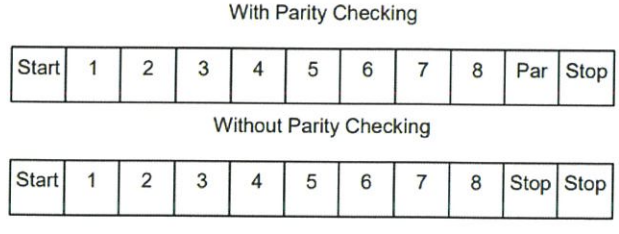
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 โหมด RTU

เฟรมข้อมูลในโหมด RTU ประกอบด้วยข้อมูลแสดงตำแหน่งแอดเดรส 1 ไบต์ หมายเลขฟังก์ชัน 1 ไบต์ ข้อมูลที่ทำการรับส่งจำนวนมากสุดไม่เกิน 252 ไบต์ และรหัสตรวจสอบความถูกต้องของข้อมูลแบบ CRC (Cyclical Redundancy Checking) ขนาด 2 ไบต์ ค่า CRC นี้ เป็นค่าที่คำนวณมาจากข้อมูลทุกไบต์ ไม่รวมบิต Start, Stop และ Parity Check โดยที่ตัว Slave ตัวที่ส่งข้อมูลออกมาจะสร้างรหัส CRC แล้วส่งตามท้ายไบต์ข้อมูลออกมา หลังจากนั้นเมื่อ Master ได้รับเฟรมข้อมูลและถอดข้อมูลออกจากเฟรมแล้วจะทำการคำนวณค่า CRC ตามสูตรเดียวกับ Slave เพื่อทำการเปรียบเทียบค่า CRC ทั้งสองค่าว่าตรงกันหรือไม่ หากไม่ตรงกันแสดงว่าเกิดความผิดพลาดในการรับส่งข้อมูล ในโหมด RTU การรับส่งข้อมูล 1 ไบต์ ไม่ว่าจะเป็นข้อมูลส่วนใดภายในเฟรมจะต้องทำการส่งบิตข้อมูลรวม 11 บิตคือ บิตเริ่มต้น (Start Bit) 1 บิต, บิตข้อมูล (Data Bit) 8 บิต, บิตตรวจสอบ Parity ของข้อมูล (Parity Bit) 1 บิต และบิตหยุด (Stop Bit) 1 บิต หรือหากเลือกแบบไม่มีบิต Parity ก็จะเป็นแบบ Stop Bit แทน 2 บิต สำหรับการกำหนดให้มีบิต Parity นั้น สามารถเลือกแบบเป็นคู่ (Even Parity) หรือคี่ (Odd Parity) ก็ได้ และหากต้องการออกแบบให้สอดคล้องกับอุปกรณ์ที่มีใช้กันทั่วไปมากที่สุด ควรเลือกแบบคู่ โดยที่สามารถปรับเปลี่ยนเป็นแบบคี่หรือไม่มีการตรวจสอบ Parity (No Parity) ได้ด้วย (Pongkung, 2016) ดังรูปที่ 2.2 และรูปที่ 2.3



รูปที่ 2.2 ลักษณะเฟรมข้อมูลของ MODBUS RTU



รูปที่ 2.3 ลักษณะข้อมูลแต่ละไบต์ของ MODBUS RTU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 โหมด ASCII

สำหรับโหมด ASCII จะมองข้อมูล 1 ไบต์ นั้นออกมาเป็นตัวอักษร 2 ตัว เช่น ค่า 0x5B ซึ่งเป็นเลขฐานสิบหก ก็จะถูกมองเป็นตัวอักษร '5' และตัวอักษร 'B' จากนั้นก็จะทำการค้นหารหัส ASCII ของตัวอักษรทั้ง 2 ตัวนั้น ซึ่งได้แก่ 0x35 สำหรับ '5' และ 0x42 สำหรับ 'B' แล้วทำการส่งรหัส ASCII ทั้ง 2 ค่านี้ออกไป ซึ่งจะได้ผลเท่ากับการส่งค่า 0x5B ซึ่งเป็นข้อมูลขนาด 1 ไบต์ ในโหมด RTU โหมดนี้จะเริ่มต้นเฟรมข้อมูลด้วยการส่งรหัส ASCII ที่กำหนดให้ซึ่งหมายถึงจุดเริ่มต้น คือ 0x3A ซึ่งตรงกับตัวอักษร ':' ตามด้วยแอดเดรสของ Slave, หมายเลขฟังก์ชัน, ข้อมูล, รหัสตรวจสอบ RLC และรหัส ASCII 2 ตัว ที่กำหนดให้หมายถึงจุดสิ้นสุด คือ รหัส 0x0D และ 0x0A คือรหัส CR (Carriage Return) และ LF (Line Feed) ตามลำดับ โดยในขณะที่บัสข้อมูลว่างจากการรับส่งข้อมูล อุปกรณ์ทุกตัวจะคอยตรวจสอบข้อมูลในบัสว่ามีการส่งรหัส ASCII ของตัวอักษร ':' ออกมาหรือไม่ ถ้ามีก็จะรับรู้ว่าขณะนี้ได้มีการเริ่มต้นส่งเฟรมข้อมูลออกมาแล้ว ก็จะเข้ากระบวนการรับข้อมูลต่อไป (PONGSAKRIVERPLUS, 2011) ดังรูปที่ 2.4

Start	Address	Function	Data	LRC	End
1 char	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2chars CR,LF

รูปที่ 2.4 ลักษณะเฟรมข้อมูลของ MODBUS ASCII

2.2 ระบบปฏิบัติการแอนดรอยด์ (Android)

แอนดรอยด์ (Android) ได้ถูกออกแบบมาสำหรับอุปกรณ์ต่างๆ เช่น สมาร์ทโฟน และแท็บเล็ต ในปัจจุบันมีการนำไปใช้ในอุปกรณ์ที่หลากหลายเนื่องจากเป็นมาตรฐานเปิด เช่น กล้องถ่ายภาพ สมาร์ททีวี และอุปกรณ์ต่างๆ โดยสมาร์ทโฟนที่ใช้ระบบปฏิบัติการแอนดรอยด์เครื่องแรกของโลกคือ เอชทีซี ดรีม วางจำหน่ายเมื่อปี พ.ศ. 2551

แอนดรอยด์นั้น ถือกำเนิดอย่างเป็นทางการในวันที่ 5 พฤศจิกายน 2550 โดยบริษัท กูเกิล จุดประสงค์ของแอนดรอยด์นั้น มีจุดเริ่มต้นมาจากบริษัท Android Inc. ที่ได้นำเอาระบบปฏิบัติการลีนุกซ์ (Linux) ซึ่งนิยมนำไปใช้งานกับเครื่องแม่ข่าย (Server) เป็นหลัก นำมาลดทอนขนาดตัว (แต่ไม่ลดทอนความสามารถ) เพื่อให้เหมาะสมแก่การนำไปติดตั้งบนอุปกรณ์พกพา ที่มีขนาดพื้นที่จัดเก็บข้อมูลที่จำกัด โดยหวังว่าแอนดรอยด์นั้นจะเป็นหุ่นยนต์ที่คอยช่วยเหลืออำนวยความสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าตาของระบบ

ส่วนต่อประสานกับผู้ใช้ (User Interface) ของระบบปฏิบัติการนี้ มีพื้นฐานอยู่บนอินเตอร์เฟซแบบไดเรกต์มานิพูเลชัน (Direct Manipulation) ที่เป็นการอ้างอิงกับการกระทำของมนุษย์ เช่น การกวาดนิ้ว, การปัด, การแตะ การตอบสนองเหล่านี้ได้รับการออกแบบมาอย่างดี และมีการสั่นของอุปกรณ์เพื่อตอบสนองการสัมผัสของผู้ใช้งาน และยังมีส่วนอื่นที่ถูกเพิ่มเติมเพื่อตอบสนองต่อปัจจัยอื่นๆ เพื่อความสะดวกสบายและการใช้งานต่างๆ กับผู้ใช้เพิ่มเติม เช่น การหมุนหน้าจอ หรือการเล่นเกมส์ที่ใช้การหมุนของอุปกรณ์เพื่อความบันเทิง เป็นต้น

อุปกรณ์แอนดรอยด์จะมีหน้าหลัก ซึ่งเป็นหน้าจอที่จะนำทางไปทุกๆ ที่ตามที่ผู้ใช้ต้องการในอุปกรณ์ เหมือนกับเครื่องคอมพิวเตอร์ที่มีหน้าจอหลักเป็นเดสก์ทอป หน้าจอหลักของแอนดรอยด์จะเป็นพื้นที่สำหรับวางไอคอนของแอปพลิเคชัน และวิดเจ็ต โดยไอคอนเหล่านี้จะสามารถแตะเพื่อเข้าใช้งานแอปพลิเคชันได้โดยตรง, การพยากรณ์อากาศที่มีการอัพเดตอยู่ตลอดเวลา กล่องจดหมายอิเล็กทรอนิกส์ (อีเมลล์) และอื่นๆ อีกมากมายหน้าจอหลักสามารถสร้างได้หลายหน้า โดยแต่ละหน้าจะสามารถเปลี่ยนหรือเลื่อนหน้าได้โดยการปัดเพื่อเลื่อนไปมา ระหว่างหน้าได้ แม้ว่าหน้าจอหลักของแอนดรอยด์จะอนุญาตให้ผู้ใช้สามารถปรับแต่งรูปแบบต่างๆ ได้ตามที่ต้องการ เพื่อตอบสนองตามรสนิยมของตนเอง แอปพลิเคชันต่างๆ มีให้ดาวน์โหลดบนกูเกิล เพลย์ และแอปพลิเคชันหลายตัวสามารถเปลี่ยนแปลงลักษณะหรือธีมได้ จนสามารถลอกเลียนแบบลักษณะรูปแบบของระบบปฏิบัติการอื่นๆ และได้มีผู้ให้บริการบางรายปรับเปลี่ยนลักษณะให้ต่างไปจากเดิม เพื่อให้โดดเด่นจากคู่แข่งทางการค้าของบริษัทนั้นๆ ระบบปฏิบัติการของแอนดรอยด์ก็ได้ถูกพัฒนาเพื่ออำนวยความสะดวกสบายของผู้ใช้งานเรื่อยมา เช่น ความสามารถในการโทรกลับโดยไม่ต้องเปิดแอปโทรศัพท์จากสายที่ไม่ได้รับ การแจ้งเตือนจะหายไปหลังจากที่ผู้ใช้อ่าน

2.3 แอปพลิเคชัน (Application)

ระบบปฏิบัติการแอนดรอยด์มีแอปพลิเคชันที่เติบโตขึ้นตลอดมา ซึ่งผู้ใช้สามารถซื้อและดาวน์โหลดได้จากกูเกิล เพลย์ หรือแอมะซอน แอปสโตร์ แอปพลิเคชันจากเพลย์สโตร์สามารถดาวน์โหลดและอัพเดตได้จากกูเกิล หลังจากที่ซื้อแอปพลิเคชันสามารถขอคืนเงินได้ภายใน 15 นาที หลังจากการดาวน์โหลด ถ้าผู้ใช้งานไม่พอใจในแอปพลิเคชัน และในผู้ให้บริการบางรายจะเรียกเก็บเงินด้วยใบเสร็จบนกูเกิล เพลย์ ซึ่งจะมีค่าใช้จ่ายเพิ่มเติมจากค่าใช้จ่ายบริการรายเดือน ในเดือนกันยายน พ.ศ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2555 มีแอปพลิเคชันมากถึง 675,000 แอปพลิเคชัน และมียอดการดาวน์โหลดทั้งหมด 2.5 พันล้านครั้ง

แอปพลิเคชันถูกสร้างโดยภาษาจาวา และแอนดรอยด์ซอฟต์แวร์เดเวล็อปเมนต์คิต (Android Software Development Kit) หรือ เอสดีเค (SDK) โดยจะประกอบด้วยเครื่องมือต่างๆ ที่ใช้ในการพัฒนาแอปพลิเคชัน รวมไปด้วยตัวรีบัก แหล่งรวมซอฟต์แวร์ต่างๆ ตัวจำลองแฮนด์เซต โคดจำลอง และวิธีใช้ต่าง ๆ

แต่ส่วนในประเทศจีนนั้น มีการจำกัดการใช้อินเทอร์เน็ตต่างๆ ของทางรัฐ โดยอุปกรณ์แอนดรอยด์ที่วางขายในประเทศจีนนั้นจะถูกปิดกั้นบริการบางอย่าง จะมีเพียงแค่ส่วนที่ได้รับอนุมัติจากรัฐบาลแล้วเท่านั้น

การจัดการหน่วยความจำ

หน่วยความจำหรือแรม ถูกออกแบบให้ใช้พลังงานที่น้อยที่สุด เมื่อแอปพลิเคชันของแอนดรอยด์ไม่ได้ใช้งาน ระบบจะจัดการจัดเก็บไว้ในหน่วยความจำ (เมื่อเปิดแอปพลิเคชันทิ้งไว้ในการใช้งาน)

แอนดรอยด์จะจัดการแอปพลิเคชันในหน่วยความจำอัตโนมัติ ซึ่งเมื่อแรมเหลือน้อย ระบบจะจัดการปิดแอปพลิเคชันรต่างๆ ที่กำลังทำงานอยู่ทันที โดยกระบวนการเหล่านี้ผู้ใช้จะไม่เห็นมันได้ อย่างไรก็ตามจะมีแอปพลิเคชันบนกูเกิล เพลย์ ที่จะสามารถจัดการและปิดแอปพลิเคชันได้ ซึ่งคาดกันว่าให้ผลร้ายมากกว่าผลดี

2.4 RS-485

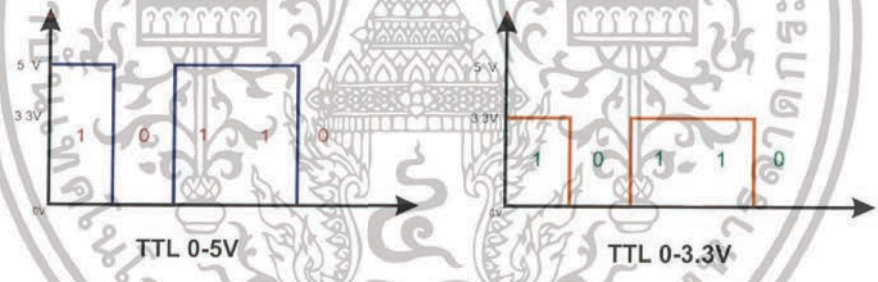
มาตรฐาน RS-486 กำหนดโดยสมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์ (EIA) เป็นมาตรฐานการเชื่อมต่อสัญญาณแบบอนุกรม (Serial Communication) เชื่อมต่อแบบหลายจุด (Multi-point) สายสัญญาณที่ใช้มีทั้งแบบ 2 สาย และ 4 สาย การสื่อสารนี้เป็นแบบ Half Duplex คือ สามารถส่งหรือรับสัญญาณได้ทีละอย่างเท่านั้น ไม่สามารถส่งและรับได้ในเวลาเดียวกัน โดยมี Master 1 ตัวในการจัดคิวการสื่อสารกับ Slave โดยส่งโปรโตคอลออกไป ซึ่งในโปรโตคอลจะระบุแอดเดรสของ Slave ที่ต้องการ เมื่อตัว Slave เห็นว่าตรงกับแอดเดรสของตัวเองก็จะทำการตอบกลับไปยัง Master (Pongkung, 2017) ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 การเชื่อมต่อ RS-485 ระหว่างเครื่องมือวัดกับตัวแปลงสัญญาณ

2.5 TTL (Transistor-Transistor Logic)

TTL เป็นระดับแรงดันที่ถูกกำหนดขึ้นในยุคแรกๆ เพื่อใช้ระหว่าง Transistor กับ Transistor และใช้มาจนถึงปัจจุบัน ส่วนมากใช้สื่อสารกันระหว่าง IC และ MCU ซึ่งจะใช้ระดับแรงดัน 0-5V แต่ในปัจจุบันมีอุปกรณ์หลายเบอร์ที่ทำงานในช่วง 0-3.3V (เรียกแรงดันระดับนี้ว่า LVTTL) ซึ่งผู้ใช้ควรตรวจสอบจาก Datasheet ของอุปกรณ์ที่ใช้เสียก่อนว่าเป็นระดับแรงดันแบบใด เพราะหากใช้ผิดประเภทจะทำให้อุปกรณ์เสียหาย (Anonymous, 2017) ดังรูปที่ 2.6



รูปที่ 2.6 กราฟตัวอย่างการส่งข้อมูลแบบ TTL 0-5V และ TTL 0-3.3V

2.6 ESP32

ESP32 เป็นชื่อของไอซีไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ Wi-Fi และ Bluetooth 4.2 BLE ในตัว ผลิตโดยบริษัท Espressif จากประเทศจีน โดยตัวไอซี ESP32 มีสเปกโดยละเอียดดังนี้

- ซีพียูใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240MHz
- มีแรมในตัว 512KB
- รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มาพร้อมกับ Wi-Fi มาตรฐาน 802.11 b/g/n รองรับการใช้งานทั้งในโหมด Station softAP และ Wi-Fi Direct
- มีบลูทูธในตัว รองรับการใช้งานในโหมด 2.0 และโหมด 4.0 BLE
- ใช้แรงดันไฟฟ้าในการทำงาน 2.6V ถึง 3V
- ทำงานได้ที่อุณหภูมิ -40 องศาเซลเซียส ถึง 125 องศาเซลเซียส

นอกจากนี้ ESP32 ยังมีเซนเซอร์ต่างๆ มาในตัวด้วย ดังนี้

- วงจรกรองสัญญาณรบกวนในวงจรขยายสัญญาณ
- เซนเซอร์แม่เหล็ก
- เซนเซอร์สัมผัส (Capacitive Touch) รองรับ 10 ช่อง
- รองรับการเชื่อมต่อคลิสตอล 32.768kHz สำหรับใช้กับส่วนวงจรนับเวลาโดยเฉพาะขาใช้งาน

ต่างๆ ของ ESP32 รองรับการเชื่อมต่อบัสต่างๆ ดังนี้

- มี GPIO จำนวน 32 ช่อง
- รองรับ UART จำนวน 3 ช่อง
- รองรับ SPI จำนวน 3 ช่อง
- รองรับ I^2C จำนวน 2 ช่อง
- รองรับ ADC จำนวน 12 ช่อง
- รองรับ DAC จำนวน 2 ช่อง
- รองรับ I^2S จำนวน 2 ช่อง
- รองรับ PWM/Timer ทุกช่อง
- รองรับการเชื่อมต่อกับ SD-Card

นอกจากนี้ ESP32 ยังรองรับฟังก์ชันเกี่ยวกับความปลอดภัยต่างๆ ดังนี้

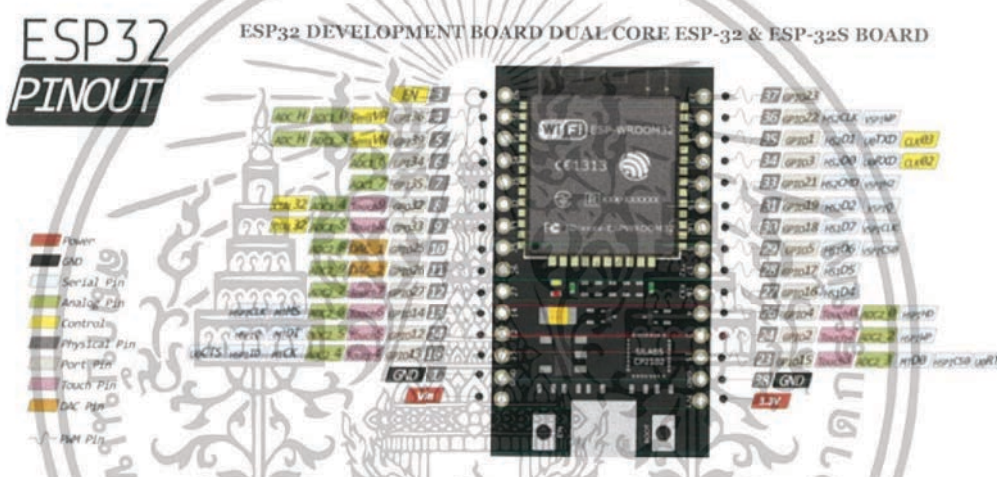
- รองรับการเข้ารหัส Wi-Fi แบบ WEP และ WPA/WPA2 PSK/Enterprise
- มีวงจรเข้ารหัส AES/SHA2/Elliptical Curve Cryptography/RSA-4096 ในตัว

ในด้านประสิทธิ์ภาพการใช้งาน ตัว ESP32 สามารถทำงานได้ดีโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รับ-ส่งข้อมูลได้ความเร็วสูงสุดที่ 150Mbps เมื่อเชื่อมต่อแบบ 11n HT40 ได้ความเร็วสูงสุด 72Mbps เมื่อเชื่อมต่อแบบ 11n HT20 ได้ความเร็วสูงสุดที่ 54Mbps เมื่อเชื่อมต่อแบบ 11g และได้ความเร็วสูงสุดที่ 11Mbps เมื่อเชื่อมต่อแบบ 11b
- เมื่อใช้การเชื่อมต่อผ่านโปรโตคอล UDP จะสามารถรับ-ส่งข้อมูลได้ที่ความเร็ว 135Mbps
- ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5uA

เนื่องด้วยบอร์ดพัฒนาสำเร็จรูป ESP32 มีราคาย่อมเยาว์และมีประสิทธิภาพสูง ESP32 จึงเป็นที่นิยมสำหรับนำมาใช้งานเป็นอย่างมาก (Anonumous, 2017) ดังรูปที่ 2.7



รูปที่ 2.7 แต่ละพินของ ESP32

2.7 Switching Power Supply

Switching Power Supply ใช้เป็นแหล่งจ่ายไฟให้กับ ESP32 ซึ่งเปลี่ยนแรงดันไฟจากไฟฟ้ากระแสสลับแรงดันสูงให้เป็นแรงดันไฟฟ้ากระแสตรงแรงดันต่ำได้ ซึ่งองค์ประกอบพื้นฐานโดยทั่วไปจะคล้ายกันโดยมีองค์ประกอบที่สำคัญที่สุดคือ คอนเวอร์เตอร์ ดังรูปที่ 2.8

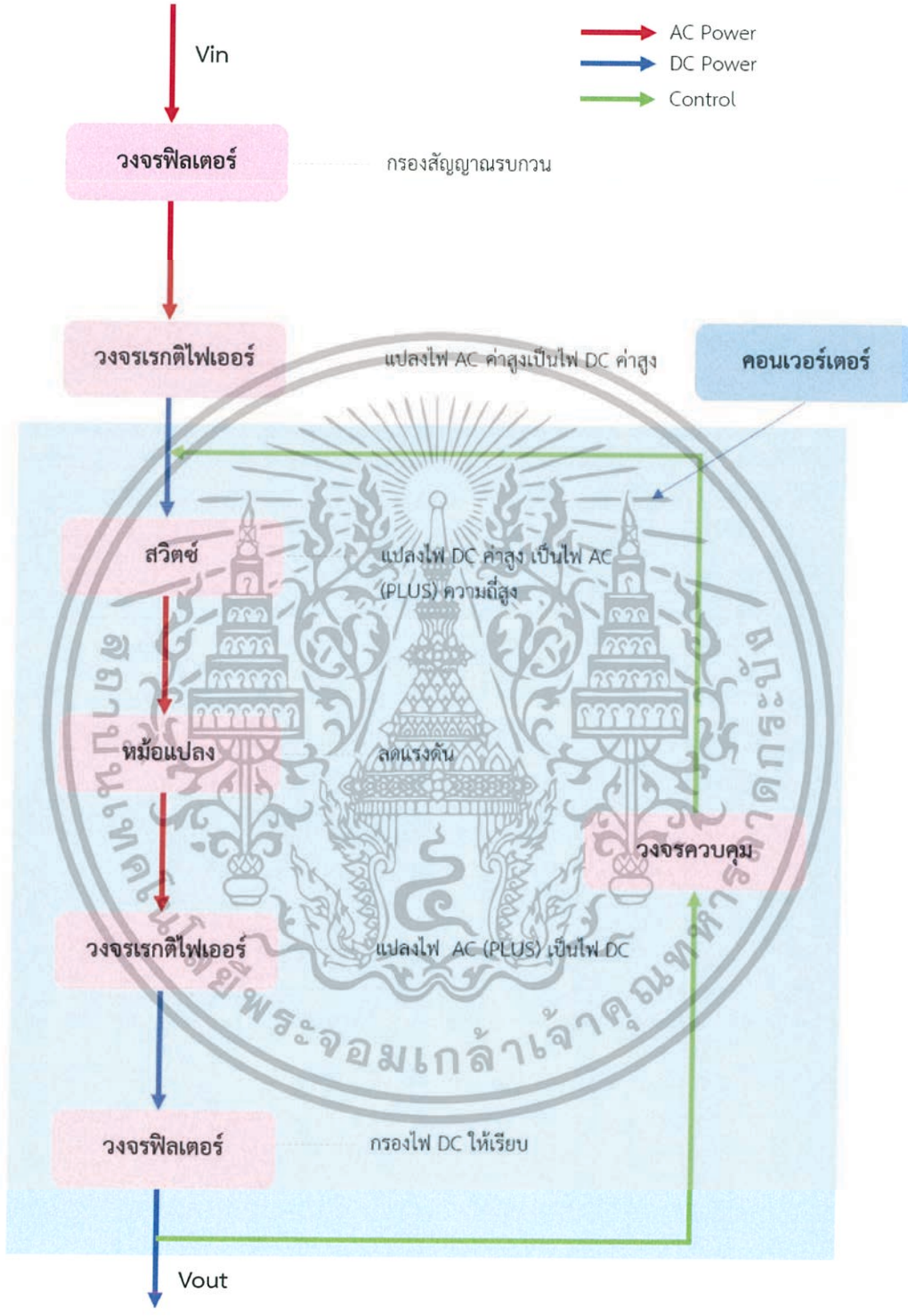
Switching Power Supply มีองค์ประกอบดังนี้

1. วงจรฟิลเตอร์และเรกติไฟเออร์ ทำหน้าที่แปลงแรงดันไฟฟ้าสลับเป็นแรงดันไฟฟ้าตรง

2. คอนเวอร์เตอร์ ทำหน้าที่แปลงไฟฟ้ากระแสตรงเป็นไฟฟ้ากระแสสลับความถี่สูง และแปลงกลับไปเป็นไฟฟ้ากระแสตรงความถี่ต่ำ

3. วงจรควบคุม ทำหน้าที่ควบคุมการทำงานของคอนเวอร์เตอร์ เพื่อให้ได้แรงดันเอาต์พุตตามต้องการ (Faisol, 2019)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แผนผัง Switching Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 Safety Breaker

เบรกเกอร์เป็นอุปกรณ์ทำหน้าที่ในการตัดวงจรไฟฟ้าแบบอัตโนมัติเมื่อเกิดความผิดปกติในระบบ เพื่อเป็นการป้องกันความเสียหายที่จะเกิดขึ้นกับสายไฟโหลด (เช่น มอเตอร์, Generator หรืออุปกรณ์ไฟฟ้า) ซึ่งเลือกใช้ Safety Breaker 15A เป็น HB type (Anonymous, 2019) ดังรูปที่ 2.9

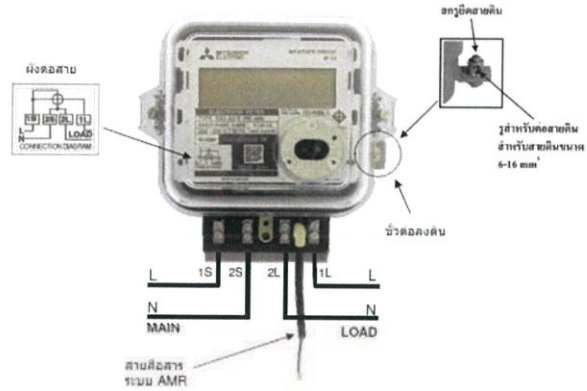
รูปที่ 2.9 Safety Breaker ที่ใช้

2.9 สายไฟประธานและสายไฟโหลด

จากคู่มือการใช้งานมิเตอร์สายไฟประธาน และสายไฟโหลดถูกกำหนดให้ใช้สายไฟขนาด 10-16 mm^2 มีข้อควรระวังต่างๆ เกี่ยวกับการต่อสายไฟประธานและสายไฟโหลดดังนี้

- ตรวจสอบการต่อสายชนิดของสาย การต่อสายผิดอาจทำให้มิเตอร์เสียหายอุปกรณ์ภายนอกเสียหาย เกิดไฟฟ้าลัดวงจรหรือเกิดไฟไหม้ได้
- การต่อสายไฟประธานและสายไฟโหลด ควรต่อด้วยความระมัดระวังและต่อในขณะที่ยังไม่จ่ายไฟให้กับมิเตอร์
- เลือกใช้สายไฟที่มีขนาดเหมาะสมกับกระแสพิกัด หากใช้ขนาดไม่เหมาะสมอาจเป็นสาเหตุให้เกิดไฟไหม้
- หลังจากติดตั้งเสร็จต้องขันยึดติดสกรู หากลืมอาจเป็นสาเหตุให้เกิดไฟดูด ไฟไหม้ การทำงานผิดพลาดได้

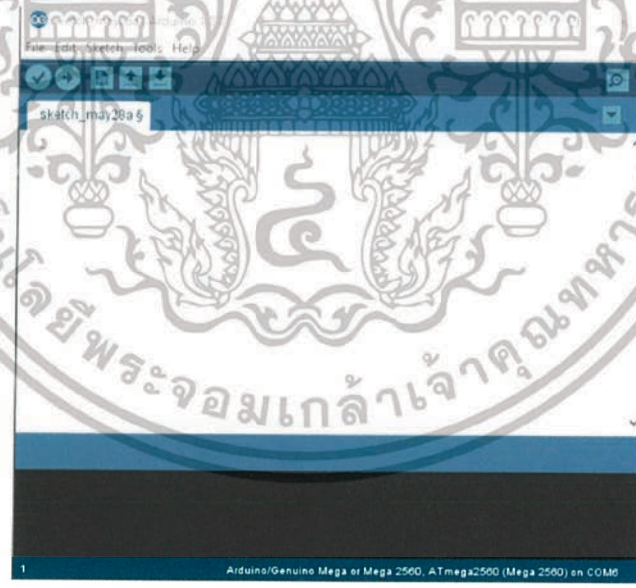
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ผังการต่อสายประธานเข้ามิเตอร์และต่อโหลดออกจากมิเตอร์

2.10 Arduino IDE

โปรแกรม Arduino IDE ใช้เขียนโปรแกรมควบคุมการทำงานของบอร์ด ESP32 เพราะเป็นโปรแกรมที่ใช้งานง่าย เขียนด้วยภาษาซี และยังเป็น Open Source ทำให้ใช้งานได้โดยไม่มีค่าใช้จ่าย เป็นที่นิยม ดังรูปที่ 2.11



รูปที่ 2.11 หน้าโปรแกรม Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

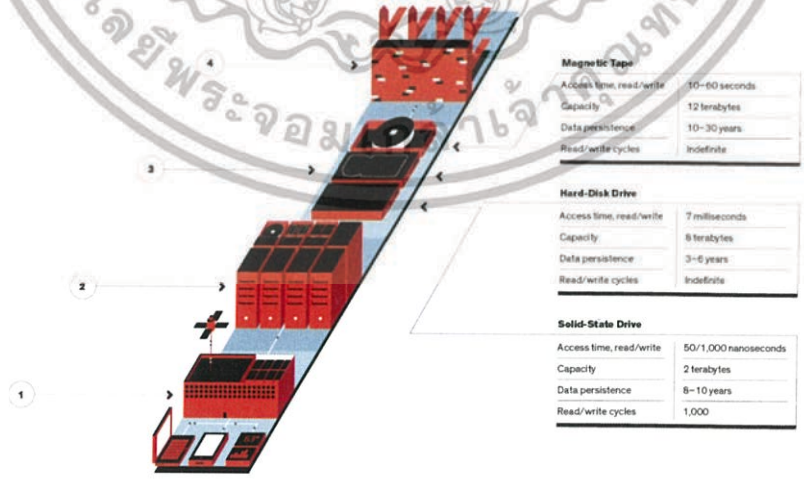
2.11 Cloud Storage System

ระบบ Cloud Storage คือ การจัดเก็บข้อมูลผ่านระบบคลาวด์บนเซิร์ฟเวอร์ที่อยู่ในระยะไกล เพื่อให้สามารถเรียกดู และเรียกใช้ข้อมูลได้ผ่านทางอินเทอร์เน็ต ซึ่งในปัจจุบันกำลังได้รับความนิยมเป็นอย่างมากตั้งแต่องค์กรขนาดเล็กไปจนถึงองค์กรขนาดใหญ่ เนื่องจากประโยชน์หลายประการ ทั้งยังช่วยประหยัดค่าใช้จ่ายเป็นอย่างมาก

โดยส่วนใหญ่แล้ว การให้บริการระบบ Cloud Storage คือ การให้บริการโดยผู้ให้บริการที่เรียกว่า Host ซึ่งผู้ให้บริการเหล่านี้จะทำหน้าที่เป็นผู้รับผิดชอบในการรักษาข้อมูลทั้งหมดของผู้ใช้บริการ รวมถึงยังดูแลรักษาการจัดเก็บข้อมูล และระบบการประมวลผลเพื่อให้ผู้ใช้สามารถเข้าถึงข้อมูลรวมถึงเรียกใช้โปรแกรมและแอปพลิเคชันต่างๆ ได้ ซึ่งในเรื่องความปลอดภัยนั้น จะป้องกันไม่ให้ผู้ที่ไม่มีสิทธิเรียกใช้หรือเข้าถึงข้อมูลสามารถเข้าถึงเซิร์ฟเวอร์ที่ทำหน้าที่จัดเก็บข้อมูลได้

2.11.1 กระบวนการเก็บข้อมูลบน Cloud

1. ข้อมูลถูกเก็บบนเครื่องในรูปแบบของ Block
2. ข้อมูลจะถูกส่งผ่านอินเทอร์เน็ตไปยังศูนย์ข้อมูล
3. ข้อมูลจะถูกเขียนส่งในฮาร์ดดิสก์หรือ SSD อย่างน้อย 2 แห่ง (อาจจะตั้งอยู่คนละไซต์กันเลย) เพื่อป้องกันการสูญหาย (Redundancy)
4. ข้อมูลจะถูกสำรองลงบนเทปแม่เหล็ก แต่อาจจะไม่ได้ใช้เทปในการสำรองข้อมูลทุกศูนย์
5. การอ่านข้อมูลกลับอาจเกิดความผิดพลาดแต่การทำ Error-correcting จะแก้ปัญหานี้ หากจำนวนบิตไม่ผิดพลาดมากเกินไป (Anonymous, 2018)



รูปที่ 2.12 กระบวนการเก็บใน Cloud

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.2 Firebase

Firebase คือ Project ที่ถูกออกแบบมาให้เป็น API และ Cloud Storage สำหรับพัฒนา Realtime Application รองรับหลาย Platform ทั้ง IOS App, Android App, Web App

Firebase ถูกสร้างขึ้นจากคุณสมบัติว่านักพัฒนาสามารถผสม และจับคู่เพื่อให้พอดีกับความต้องการของตน บริษัท ก่อตั้งขึ้นในปี ค.ศ. 2011 โดยแอนดรูลีและเจมส์ เทมปลิน สินค้าเริ่มต้น Firebase เป็นฐานข้อมูลเรียลไทม์ ซึ่งมี API ที่ช่วยให้นักพัฒนาในการจัดเก็บและซิงค์ข้อมูล โดย Google Firebase 2.0 กูลเกิลได้ซื้อกิจการ Firebase และมีการพัฒนาจากบริการ Backend เก็บข้อมูลอย่างเดียว ให้สามารถมาเป็นแพลตฟอร์ม ครบวงจรสำหรับนักพัฒนาแอป รองรับบริการแทบทุกอย่างที่นักพัฒนาแอปพลิเคชันต้องใช้งาน ดังรูปที่ 2.13

รูปที่ 2.13 Firebase

บริการของ Firebase

1. Firebase Analytics บริการวิเคราะห์ข้อมูล ดึงเทคโนโลยีมาจาก Google Analytics แถมยังเปิดให้ใช้ฟรีแบบไม่จำกัดปริมาณข้อมูลใดๆ

2. Firebase Cloud Messaging (FCM) ระบบส่งข้อความแจ้งเตือน ใช้งานฟรีไม่จำกัดปริมาณข้อความ

3. Firebase Storage บริการพื้นที่เก็บข้อมูลเอาไว้เก็บภาพ วิดีโอ หรือไฟล์ขนาดใหญ่จากแอปพลิเคชันของผู้ใช้สร้างอยู่บน Google Cloud Storage

4. Firebase Remote Config ตัวช่วยอัพเดตคอนฟิกของแอป สำหรับปรับแต่งค่าต่างๆ ในแอปจากระยะไกล (เช่น เกมที่อยากปรับสมดุลของเกมตลอดเวลา) สามารถใช้ร่วมกับ Firebase Analytics เพื่อกำหนดผู้ใช้งานแยกเป็นกลุ่มๆ ได้

5. Firebase Crash Reporting ตัวรายงานการแครชของแอป รองรับทั้ง iOS และ Android

6. Firebase Test Lab for Android บริการทดสอบแอปพลิเคชันบนฮาร์ดแวร์จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. Firebase Notifications เป็นคอนโซลสำหรับนักพัฒนา เพื่อยิงข้อความผ่าน FCM ไปยังผู้ใช้สำหรับโปรโมทหรือกระตุ้นให้ผู้ใช้กลับมาเปิดแอปของเรา (เช่น การแจกของในเกม)

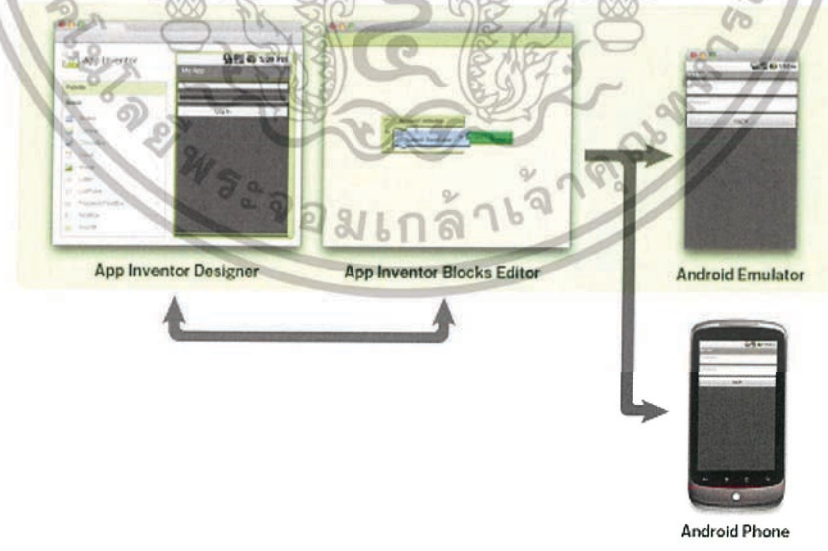
8. Firebase Dynamic Links บริการ URL กลางที่สามารถชี้ทางไปยังเพจต่างๆ แปรผันตามอุปกรณ์หรือคุณสมบัติของผู้ใช้ (เช่น แต่ละประเทศกดลิงก์เดียวกัน เข้าคนละเพจกัน)

9. Firebase Invites ระบบเชิญเพื่อนมาใช้แอป มีฟีเจอร์ Referral โดยผู้ชวนได้สิทธิประโยชน์

10. Firebase App Indexing เปลี่ยนชื่อมาจาก Google App Indexing ที่ช่วยให้ Google Search ค้นเจอเนื้อหาภายในแอป ดังนั้น Firebase จึงครอบคลุมทุกการบริการสำหรับพัฒนา Realtime Application บริการเกือบทุกตัวของ Firebase ใช้งานได้ฟรีแบบไม่จำกัดปริมาณ ยกเว้น Test Lab, Storage, Realtime Database, Hosting ที่คิดเงิน (Anonymous, 2017)

2.12 MIT App Inventor

MIT App Inventor โปรแกรมที่พัฒนาโดยความร่วมมือของ Google กับ MIT สำหรับเขียนโปรแกรมบนสมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ โดย MIT App Inventor ใช้หลักการคล้ายๆ Scratch แต่ซับซ้อนกว่า คือ เขียนด้วยการต่อบล็อกคำสั่ง เน้นการออกแบบเพื่อแก้ปัญหาด้วยการสร้างโปรแกรมที่ผู้เรียนสนใจบนสมาร์ทโฟน และยังเป็นโปรแกรมที่เหมาะสำหรับใช้ในการสอนเขียนโปรแกรมให้นักเรียนและนักศึกษา ดังรูปที่ 2.14



รูปที่ 2.14 ภาพรวมของการเขียนแอปพลิเคชันด้วย MIT App Inventor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างแอปพลิเคชันจะแบ่งการทำงานออกเป็นสองส่วนคือ ส่วนออกแบบ (App Inventor Designer) ที่จะให้เลือกคอมโพเนนท์ที่ต้องการสำหรับที่จะให้สร้างแอปพลิเคชัน ส่วนที่สองเป็นส่วนการเขียนโปรแกรม (App Inventor Blocks Editor) ที่ให้เขียนโปรแกรมด้วยการต่อบล็อกต่างๆ เข้าด้วยกันเป็นคำสั่ง ซึ่งจะเป็นการกำหนดพฤติกรรมหรือเหตุการณ์ที่เกิดขึ้นกับคอมโพเนนท์ การเขียนโปรแกรมจะเสมือนการต่อชิ้นส่วนตัวต่อจิ๊กซอว์เข้าด้วยกัน ในแต่ละขั้นตอนการสร้างจะสามารถทำการทดสอบได้ทุกขณะ

ขั้นตอนการสร้างแอปพลิเคชัน เริ่มจากออกแบบหน้าตาโปรแกรมบนสมาร์ทโฟนด้วย App Inventor Designer ซึ่งใช้สำหรับสร้างส่วนโปรแกรมต่างๆ เพื่อใช้งานในสมาร์ทโฟน จากนั้นเขียนแต่ละส่วนด้วย App Inventor Block Editor ระหว่างเขียนโปรแกรมอาจมีการแก้ไขเพิ่มเติม หรือลบบางส่วนโปรแกรมออกไป ทำให้ต้องแก้ไขโปรแกรม (Debug) จนกว่าจะได้โปรแกรมตามที่ออกแบบไว้ เมื่อทุกส่วนโปรแกรมถูกสร้างเสร็จแล้ว ก็ได้เวลาทดสอบการใช้งาน โดยการติดตั้งโปรแกรมลงไปบนสมาร์ทโฟน Android แล้วทดสอบการใช้งานผ่านสมาร์ทโฟนจริงๆ แต่ถ้าไม่มีสมาร์ทโฟน ก็ยังสามารถทดสอบได้ ผ่านโปรแกรมสมาร์ทโฟนจำลอง (Android Emulator) ในคอมพิวเตอร์แทน (Youngcyber, 2014)

2.13 การคำนวณค่าไฟฟ้า

2.13.1 คำนวณค่าไฟฟ้าต่อหน่วยชั่วโมงจากกำลังไฟฟ้าที่ใช้ในปัจจุบัน

กำลังไฟฟ้าที่ใช้ปัจจุบัน (Watt)/1000 x ค่าไฟต่อหน่วยของอะพาร์ตเมนต์ (Baht/Unit)

2.13.2 คำนวณค่าไฟฟ้าจากกำลังไฟฟ้าที่ใช้สะสม

กำลังไฟฟ้าที่ใช้สะสม (Watt hour)/1000 x ค่าไฟต่อหน่วยของอะพาร์ตเมนต์ (Baht/Unit)

2.14 วิธีใช้ไฟฟ้าอย่างประหยัด

ไฟฟ้าแสงสว่าง

- ควรปิดไฟทุกครั้งเมื่อไม่มีคนอยู่ในห้อง
- เลือกใช้หลอดไฟที่มีกำลังวัตต์เหมาะสมกับการใช้งาน
- สำหรับบริเวณที่ต้องการความสว่างมากภายในอาคารควรเลือกใช้หลอดฟลูออเรสเซนต์ส่วนภายนอกอาคารควรเลือกใช้หลอดไอโซเดียมและหลอดไอปรอท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ควรใช้ฝาครอบดวงโคมแบบใสหากไม่มีปัญหาเรื่องแสงจ้า และหมั่นทำความสะอาดอยู่เสมอ
- พิจารณาใช้โคมไฟตั้งโต๊ะสำหรับงานที่ต้องการแสงสว่างจุดเดียว ทีวี วิทยุ ปิดเครื่องทุกครั้งเมื่อไม่ได้ดู
- ควรถอดปลั๊กเมื่อไม่ใช้เป็นเวลานาน
- ควรเลือกใช้โคมไฟแบบสะท้อนแสงแทนแบบเดิมที่ใช้พลาสติกปิด
- ควรใช้หลอดฟลูออเรสเซนต์หรือหลอดคอมแพคฟลูออเรสเซนต์แทนหลอดไส้ซึ่งมีคำแนะนำในการใช้ดังนี้
- หลอดฟลูออเรสเซนต์แบบผอมขนาด 18 วัตต์ และ 36 วัตต์ มีความสว่างเท่ากับหลอด 20 วัตต์ และ 40 วัตต์ แต่ประหยัดไฟกว่า และสามารถใช้แทนกันได้ โดยไม่ต้องเปลี่ยนบัลลาสต์และสตาร์ทเตอร์
- หลอดคอมแพคฟลูออเรสเซนต์มี 2 ชนิดคือ ชนิดมีบัลลาสต์ภายในสามารภใช้แทนหลอดกลมแบบเกลียวได้ ส่วนหลอดที่มีบัลลาสต์ภายนอก จะมีขาเสียบเพื่อต่อกับตัวบัลลาสต์ที่อยู่ภายนอก

เตารีด

- เตารีดเป็นเครื่องใช้ไฟฟ้าประเภทให้ความร้อน ซึ่งในการรีดแต่ละครั้งจะกินไฟมากดังนั้นจึงควรรู้จักวิธีใช้อย่างประหยัดและปลอดภัย
- ก่อนอื่นควรตรวจสอบดูว่าเตารีดอยู่ในสภาพพร้อมที่จะใช้งานหรือไม่ เช่น สาย ตัวเครื่อง
- ตั้งปุ่มปรับความร้อนให้เหมาะสมกับชนิดของผ้า
- อย่าพรมน้ำจนเปียกแฉะ
- ดึงเต้าเสียบออกก่อนจะรีดเสร็จประมาณ 2-3 นาที แล้วรีดต่อไปจนเสร็จ
- ควรพรมน้ำพอสมควร
- ถอดปลั๊กออกเมื่อไม่ได้ใช้
- ควรรีดผ้าคราวละมากๆ ติดต่อกันจนเสร็จ
- ควรเริ่มรีดผ้าบางๆ ก่อน ขณะเตารีดยังไม่ร้อน
- ควรดึงปลั๊กออกก่อนรีดเสร็จเพราะความร้อนยังคงอยู่
- ควรซักและตากผ้าโดยไม่ต้องบิดจะทำให้รีดง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัดลม

- เปิดความเร็วลมพอควร
- เปิดเฉพาะเวลาใช้งาน
- ควรเปิดหน้าต่างใช้ลมธรรมชาติแทนถ้าทำได้

เครื่องเป่าผม

- เช็ดผมก่อนใช้เครื่อง
- ควรขยี้และสางผมไปด้วยขณะเป่า

เครื่องดูดฝุ่น

- ควรเอาฝุ่นในถุงทิ้งทุกครั้งที่ใช้แล้วจะได้มีแรงดูดดี ไม่เปลืองไฟ

ตู้เย็น ตู้แช่

- ตั้งอุณหภูมิพอสมควร
- นำของที่ไม่ร้อนใส่ตู้เย็น
- ปิดประตูตู้เย็นทันทีเมื่อนำของใส่หรือออก
- ปิดประตูตู้เย็นให้สนิท
- หากยางขอบประตูรั่วให้รีบแก้ไข
- เลือกตู้เย็นหรือตู้แช่ชนิดมีประสิทธิภาพสูง
- ควรใช้ตู้เย็นขนาดเหมาะกับครอบครัว
- ควรตั้งตู้เย็นให้ห่างจากแหล่งความร้อน ให้หลังตู้ห่างจากฝาเกิน 15 ซ.ม. เพื่อระบายความร้อนได้สะดวก ไม่เปลืองไฟฟ้า
- ควรหมั่นทำความสะอาดแผงระบายความร้อน
- ควรเก็บเฉพาะอาหารเท่าที่จำเป็น
- หมั่นละลายน้ำแข็งเมื่อเห็นว่าน้ำแข็งเกาะหนามาก

การเลือกซื้อตู้เย็น, ตู้แช่ มีคำแนะนำให้ท่านพิจารณาก่อนซื้อ ดังนี้

- เลือกขนาดให้พอเหมาะกับความต้องการของครอบครัว
- ตู้เย็นแบบประตูเดียวกินไฟน้อยกว่าแบบ 2 ประตู
- ควรวางตู้เย็นให้อากาศถ่ายเทได้สะดวก
- ตั้งสวิตช์ควบคุมอุณหภูมิให้เหมาะกับจำนวนของที่ใส่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หม้อหุงข้าวไฟฟ้า

หากใช้อย่างถูกต้องสามารถประหยัดพลังงานไฟฟ้าได้มาก ซึ่งมีข้อแนะนำดังนี้

- ควรหุงข้าวให้พอดีกับจำนวนผู้รับประทาน
- ควรถอดเต้าเสียบออกเมื่อข้าวสุกแล้ว
- อย่าทำให้ก้นหม้อตัวในเกิดรอยบุบ จะทำให้ข้าวสุกช้า
- หมั่นตรวจบริเวณแท่นความร้อนในหม้อ อย่าให้เม็ดข้าวเกาะติด จะทำให้ข่าวสุกช้าและเปลืองไฟ
- ใช้ขนาดที่เหมาะสมกับจำนวนสมาชิกในครอบครัว
- ควรดึงปลั๊กออกเมื่อข้าวสุกพอแล้ว ปัจจุบันหม้อหุงข้าวไฟฟ้ามีใช้กันมาก หม้อต้มน้ำ หม้อต้มกาแฟ
- ใส่น้ำให้มีปริมาณพอควร
- ควรปิดฝาให้สนิทขณะต้ม
- ควรปิดสวิตช์ทันทีเมื่อน้ำเดือด

เครื่องสูบน้ำ

เครื่องสูบน้ำเป็นอุปกรณ์ไฟฟ้าที่อำนวยความสะดวกอย่างยิ่ง ซึ่งใช้มอเตอร์ไฟฟ้าในการสูบน้ำไปยังถังเก็บหรือเพื่อนำไปใช้ประโยชน์ ซึ่งมีวิธีการใช้อย่างประหยัดดังนี้

- ควรติดตั้งอุปกรณ์อัตโนมัติควบคุมระดับน้ำในถังและหมั่นปรับตั้งให้ถูกต้องเสมอ
- ติดตั้งท่อน้ำให้มีขนาดเหมาะสมกับขนาดปั๊ม
- ควรตรวจแก้ไขจุดรั่วในระบบน้ำ
- ควรใช้น้ำอย่างประหยัด
- ควรติดตั้งถังเก็บน้ำในตำแหน่งที่ไม่สูงเกินไป
- ควรติดตั้งอุปกรณ์อัตโนมัติควบคุมระดับน้ำในถังเก็บ และดูแลรักษาให้ทำงานได้อยู่เสมอ
- ตรวจสอบรอยรั่วตามข้อต่อต่างๆ หากพบควรรับซ่อมแซมแก้ไขโดยเร็ว
- หากตัวถังเก็บน้ำไม่มีอุปกรณ์อัตโนมัติควบคุมระดับน้ำ ควรดูแลอย่าให้น้ำล้นถัง
- เครื่องสูบน้ำแบบใช้สายพานต้องตรวจสอบไม่ให้หย่อนหรือตึงเกินไป

เครื่องซักผ้า

- ควรใส่ผ้าแต่พอเหมาะ ไม่น้อยเกินไป และไม่มากจนเกินกำลังเครื่อง
- ควรใช้น้ำเย็นซักผ้า ส่วนน้ำร้อนให้ใช้เฉพาะกรณีรอยเปื้อนไขมันมาก
- ควรใส่ผ้าที่จะซักตามคำแนะนำของแต่ละเครื่อง
- หากมีผ้าต้องซัก 1-2 ชิ้น ควรซักด้วยมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หากมีแสงแดดไม่ควรใช้เครื่องอบแห้ง ควรจะนำเสื้อผ้าที่ซักเสร็จมาตากแดด

มอเตอร์ไฟฟ้า

- ควรตรวจสอบแก้ไข และอัดจารบีตามวาระ
- ปรับปรุงสายพานมอเตอร์ เช่น ปรับความตึงสายพาน เปลี่ยนสายพานใหม่
- พิจารณาเปลี่ยนระบบควบคุมความเร็วของมอเตอร์เป็นระบบอิเล็กทรอนิกส์

เตาอบ เตาไฟฟ้า

- เครื่องใช้ไฟฟ้าประเภทนี้ ใช้ความร้อนมาทำให้อาหารสุก หากให้ความร้อนสูญเสียไปโดยใช้ไม่ถูกวิธี ทำให้อาหารสุกช้าลง กินกระแสไฟเพิ่มขึ้นจึงมีข้อแนะนำการใช้เครื่องใช้ไฟฟ้าประเภทนี้อย่างประหยัด

- ควรเตรียมเครื่องปรุงในการประกอบอาหารให้พร้อมก่อนใช้เตา
- ควรใช้ภาชนะก้นแบนและเป็นโลหะจะทำให้รับความร้อนจากเตาได้ดี
- ในการหุงต้มอาหารควรใส่น้ำให้พอดีกับจำนวนอาหาร
- ในระหว่างอบอาหารอย่าเปิดตู้อบบ่อยๆ
- ถอดเต้าเสียบทันทีเมื่อปรุงอาหารเสร็จเรียบร้อย
- ควรหรี่ไฟและปิดฝาหม้อในกรณีที่ต้องเคี่ยว
- ควรเตรียมเครื่องปรุงให้พร้อมก่อนใช้เตา
- ควรใช้เตาชนิดมองไม่เห็นขดลวดซึ่งไม่เสียความร้อนสูญเปล่ามาก และปลอดภัยกว่า
- ควรใช้พาหนะก้นแบนขนาดพื้นที่ก้นเหมาะกับพื้นที่หน้าเตา และใช้พาหนะที่มีเนื้อโลหะรับความร้อนได้ดี หากเป็นไปได้ให้ใช้กับเตาไฟฟ้าซึ่งมีขายทั่วไปอยู่แล้ว
- ควรปิดฝาภาชนะให้สนิทขณะตั้งเตา

เครื่องทำน้ำอุ่น

- ปรับปุ่มความร้อนให้เหมาะสมกับร่างกาย
- ปิดวาล์วทันทีเมื่อไม่ใช้งาน
- หากมีรอยรั่วควรรีบทำการแก้ไขทันที
- ต่อสายลงดินในจุดที่จัดไว้ให้ของเครื่องทำน้ำอุ่น
- ปิดสวิตช์ไฟฟ้าของเครื่องทำน้ำอุ่นเมื่อไม่ใช้
- ปฏิบัติตามคำแนะนำที่แนบมากับเครื่อง
- ใช้เครื่องขนาดพอสมควร
- ปรับปรุงความร้อนไม่ให้ร้อนเกินความจำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปิดก๊อกทุกครั้งเมื่อไม่ใช้งาน
- ในฤดูร้อนไม่จำเป็นต้องใช้น้ำร้อน หรือน้ำอุ่น
- ควรใช้น้ำอุ่นที่ได้ความร้อนจากแสงอาทิตย์

เครื่องปรับอากาศ

การใช้เครื่องปรับอากาศให้มีความเย็นที่สบายต่อร่างกาย จะประหยัดค่าไฟฟ้าอย่างได้ผล ซึ่งควรปฏิบัติดังนี้

- ปิดเครื่องทุกครั้งเมื่อไม่อยู่
- ปิดประตูหน้าต่างและผ้าม่านกันความร้อนจากภายนอก
- ตั้งอุณหภูมิไม่ควรต่ำกว่า 26 องศาเซลเซียส
- ควรใช้เครื่องขนาดเหมาะสมกับขนาดห้อง
- ควรเลือกเครื่องปรับอากาศที่มีประสิทธิภาพสูง
- ควรติดตั้งเครื่องระดับสูงพอเหมาะ และให้อากาศร้อนระบายออกด้านหลังเครื่องได้สะดวก
- ควรบุผนังห้อง และหลังคาด้วยฉนวนกันความร้อน
- ควรบำรุงรักษาเครื่องให้มีสภาพดีตลอดเวลา
- ควรหมั่นทำความสะอาดแผ่นกรองอากาศ และแผงระบายความร้อน
- ในฤดูหนาวขณะที่อากาศไม่ร้อนมากเกินไป ไม่ควรเปิดเครื่องปรับอากาศ
- ปิดประตู หน้าต่างให้มิดชิดไม่ให้ความเย็นรั่วไหล
- พิจารณาติดตั้งบังแสงหรือกันแดด เพื่อลดภาระการทำงานของเครื่อง
- ควรเลือกใช้ขนาดที่เหมาะสมกับขนาดของห้อง
- ควรใช้ผ้าม่านกั้นประตูหน้าต่าง เพื่อป้องกันความร้อนจากภายนอก
- ตั้งปุ่มปรับอุณหภูมิให้เหมาะสมต่อร่างกาย (ประมาณ 26 องศาเซลเซียส)
- หมั่นทำความสะอาดแผ่นกรองอากาศ
- ปฏิบัติตามคำแนะนำที่แนบมากับเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

3.1 ขั้นตอนการดำเนินงาน

1. ศึกษาค้นคว้าสิ่งที่สนใจเพื่อใช้กำหนดหัวข้อโครงงาน
2. นำเสนอหัวข้อโครงงาน
3. ศึกษาและรวบรวมข้อมูลการเก็บข้อมูลจากมิเตอร์ไฟฟ้า
4. ศึกษาข้อมูลการเก็บข้อมูล และการออกแบบแอปพลิเคชัน
5. ออกแบบระบบจำลองและจัดซื้อของ
6. ออกแบบแอปพลิเคชัน
7. แก้ไขขอบเขตการทำงาน
8. ศึกษาข้อมูลการเขียนโปรแกรมสำหรับ ESP32 และ MAX13487 เพิ่มเติม
9. เขียนโปรแกรมสำหรับ ESP32
10. จัดทำระบบจำลอง
11. สร้างแอปพลิเคชัน
12. ทดสอบระบบที่พัฒนาในแต่ละส่วนและแก้ปัญหาที่เกิดขึ้นจากการทดสอบ
13. รายงานผลกับอาจารย์ที่ปรึกษา
14. จัดทำรูปเล่มรายงานและนำเสนอ

ตารางที่ 3.1 ขั้นตอนการทำงาน

ขั้นตอนการดำเนินงาน	ระยะเวลา									
	ปี 2561					ปี 2562				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. ศึกษาค้นคว้าสิ่งที่สนใจเพื่อใช้กำหนดหัวข้อโครงงาน										
2. นำเสนอหัวข้อโครงงาน										

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ขั้นตอนการทำงาน (ต่อ)

ขั้นตอนการดำเนินงาน	ระยะเวลา									
	ปี 2561					ปี 2562				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
3. ศึกษาและรวบรวมข้อมูลการเก็บข้อมูลจากมิเตอร์ไฟฟ้า										
4. ศึกษาข้อมูลการเก็บข้อมูล และการออกแบบแอปพลิเคชัน										
5. ออกแบบระบบจำลองและจัดซื้อของ										
6. ออกแบบแอปพลิเคชัน										
7. แก้ไขขอบเขตการทำงาน										
8. ศึกษาข้อมูลการเขียนโปรแกรมสำหรับ ESP32 และ MAX13487 เพิ่มเติม										
9. เขียนโปรแกรมสำหรับ ESP32										
10. จัดทำระบบจำลอง										
11. จัดทำแอปพลิเคชัน										
12. ทดสอบระบบที่พัฒนาและปรับแก้ปัญหาที่เกิดขึ้นจากการทดสอบ										
13. รายงานผลกับอาจารย์ที่ปรึกษา										
14. จัดทำรูปเล่มรายงานและนำเสนอ										

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 อุปกรณ์ที่เกี่ยวข้อง

3.2.1 มิเตอร์ Mitsubishi รุ่น SX1-A31E

มิเตอร์ SX1-A31E : 1 เฟส รุ่น RS-485 มิเตอร์อิเล็กทรอนิกส์ เอเอ็มอาร์ เป็นมิเตอร์ที่มีพอร์ตสื่อสาร RS-485 ใช้มอดบัสโปรโตคอล (MODBUS Protocol RTU) ในการสื่อสาร ดังรูปที่ 3.1

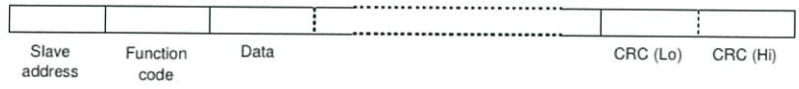
รูปที่ 3.1 มิเตอร์ Mitsubishi รุ่น SX1-A31E

ตารางที่ 3.2 คุณสมบัติเฉพาะทางเทคนิค

Item	Specifications
Physical Interface	RS-485 2wires Half Duplex
Protocol	RTU Mode
Transmission Wiring Type	Multi-point Bus (Daisy-chain)
Baud Rate	1,200 bps
Data Bit	8
Stop Bit	1
Parity	Even
CRC Polynomial	0xA001
Slave Address	1~247(F7h) (See Detail in Appendix B)
Response Time	80ms~200ms (Programable) Default 80 ms.
Distance	1,200 m
Max. Number	247
Terminator	120 or 150Ω 1/2W
Recommended Cable	Shielded Twisted Pair, Recommend LiYCY 2x0.25 mm^2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The standard communication frame consists of :



Slave address : 01~F7H
Function code : 03H..... Read Holding Registers (maximum 250 bytes)
: 10H..... Write multiple registers
Data : 8 bit HEX data
: The Cyclical Redundancy Check (CRC) field is two bytes, containing a 16-bit binary value.

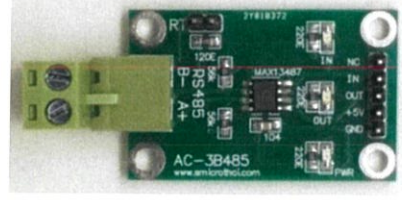
รูปที่ 3.2 ลักษณะการสื่อสาร (การสื่อสารมาตรฐาน)

ตารางที่ 3.3 ข้อมูล Register Address ของมิเตอร์

Register Address		Byte Count	R/W	Register Name	RANGE	Unit
Dec.	Hex.					
40001	0000h	2	R/W	Slave Address	1 to 247	-
40002	0001h	2	R/W	Response Time	8 to 20	10ms
40103	0066h	2	R	Line Voltage (RMS)	0 to 65535	0.01V
40106	0069h	2	R	Frequency	0 to 65535	0.1Hz
40113	0070h	2	R	Line Current (RMS)	0 to 65535	0.01A
40116	0073h	2	R	Active Power (W)	0 to 65535	W
40111	006Eh	4	R	Active Energy (Wh) imp+exp	0 to 999999999	Wh
40101	0064h	4	R	Serial No.	0 to 999999999	-
40114	0071h	2	R	Current Rating	0 to 65535	-

3.2.2 MAX13487

บอร์ดแปลง TTL (Transistor-Transistor Logic) เป็น RS-485 โดยชิพไดรฟ์เวอร์ MAX13487 สามารถเปลี่ยนทิศทางของพินได้แบบอัตโนมัติ และยังมีไฟ LED แสดงสถานะ ใช้ไฟเลี้ยง 5VDC ดังรูปที่ 3.3

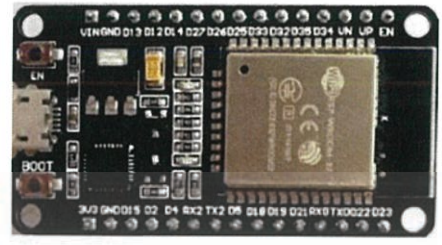


รูปที่ 3.3 โมดูลแปลง TTL เป็น RS-485 (ชิพ MAX 13487)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ESP32

ESP32 เป็นชิพไมโครคอนโทรลเลอร์ที่มาพร้อม Wi-Fi มาตรฐาน 802.11 b/g/n และเวอร์ชัน 4.2 เป็นรุ่นที่พัฒนามาจาก ESP8266 และได้แก้ไขข้อเสียของ ESP8266 ทั้งหมด ดังรูปที่ 3.4



รูปที่ 3.4 ESP32

3.2.4 ปลั๊กสามตา

ใช้ต่อไฟออกจากมิเตอร์เพื่อเป็นโหลดไฟที่ใช้แทนการใช้เครื่องใช้ไฟฟ้าต่างๆ ภายในอะพาร์ตเมนต์โดยจะนำเครื่องใช้ไฟฟ้า เช่น คอมพิวเตอร์ พัดลม มาใช้ไฟจากปลั๊กนี้ ดังรูปที่ 3.5

รูปที่ 3.5 ปลั๊กสามตาที่ใช้

3.2.5 Switching Power Supply 3V 5A

ใช้แปลงไฟ 220V เป็น 3V เพื่อต่อไปเลี้ยงตัวบอร์ด ESP32 ในระบบจำลอง ดังรูปที่ 3.6

รูปที่ 3.6 Switching Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 Safety Breaker 15A

เป็นอุปกรณ์ตัดวงจรไฟฟ้าแบบอัตโนมัติ ป้องกันไม่ให้อุปกรณ์ในระบบจำลอง และสายไฟเสียหายเมื่อเกิดความผิดปกติในระบบ ดังรูปที่ 3.7

รูปที่ 3.7 Safety Breaker 15A

3.3 ซอฟต์แวร์ที่เกี่ยวข้อง

3.3.1 โปรแกรม Arduino IDE

ใช้เขียนโปรแกรมเชื่อม ESP32 กับเครือข่าย Wi-Fi เพื่อส่งค่าจากมิเตอร์ผ่านเครือข่ายไปยัง Firebase

รูปที่ 3.8 โปรแกรม Arduino IDE

3.3.2 Firebase

ใช้เพื่อเก็บข้อมูลจากมิเตอร์และสามารถดูได้ว่าข้อมูลจากมิเตอร์ได้ถูกส่งมายัง Firebase หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 Console of Firebase

3.3.3 MIT App Inventor

ใช้เขียนหน้า User Interface ของแอปพลิเคชัน

รูปที่ 3.10 MIT App Inventor

3.4 การออกแบบและการวางแผนการทำงาน

3.4.1 การออกแบบและวางแผนทางด้าน Hardware

1. ศึกษาข้อมูลของมิเตอร์
2. เลือกอุปกรณ์ที่ใช้สำหรับรับและส่งข้อมูลที่ได้จากมิเตอร์
3. ออกแบบระบบจำลอง
4. วัดขนาดและวางระบบตามที่ออกแบบไว้

3.4.2 การออกแบบและวางแผนทางด้าน Software

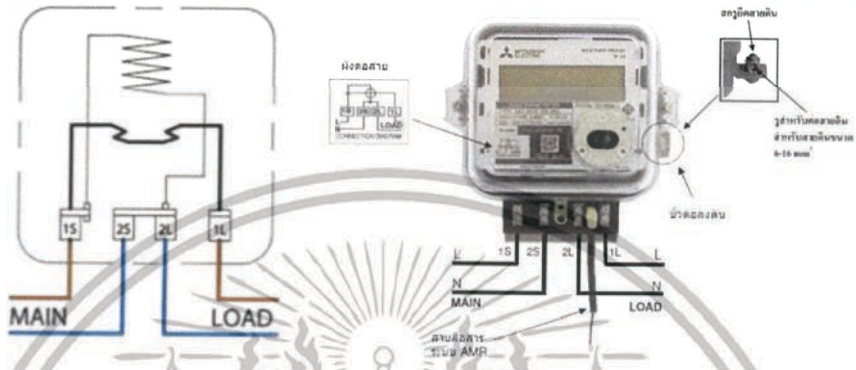
1. ศึกษาโปรแกรมที่จะใช้งาน
2. เขียนโค้ดส่วนที่ดึงข้อมูลจากมิเตอร์
3. ออกแบบ User Interface สำหรับแอปพลิเคชัน
4. สร้างส่วนของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

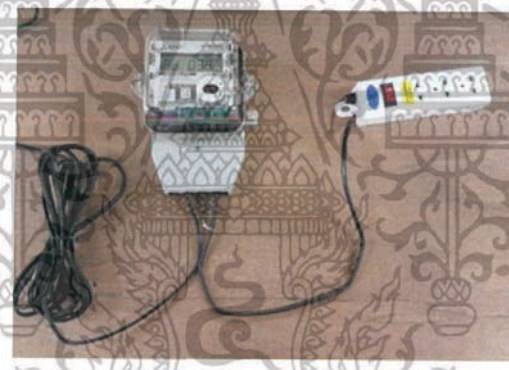
3.5 วิธีการดำเนินงาน

3.5.1 การต่อสายไฟเข้ามิเตอร์

ต่อสายไฟ L (สายสีน้ำตาล) และ N (สายสีน้ำเงิน) ด้านที่ต่อกับไฟเข้ามิเตอร์ และด้านที่ต่อออกจากมิเตอร์เพื่อจ่ายไฟให้กับโหลด



รูปที่ 3.11 การต่อสายไฟประธานและสายไฟโหลด

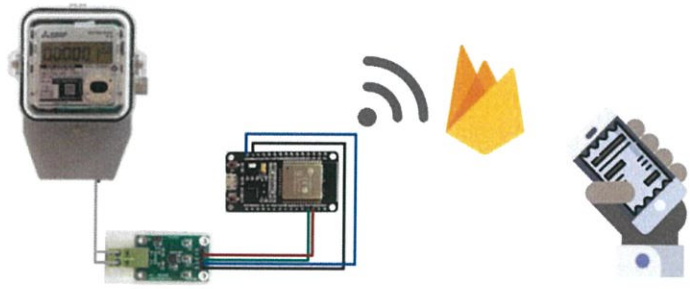


รูปที่ 3.12 ต่อสายไฟเข้ามิเตอร์ และออกจากมิเตอร์

3.5.2 การออกแบบระบบจำลอง

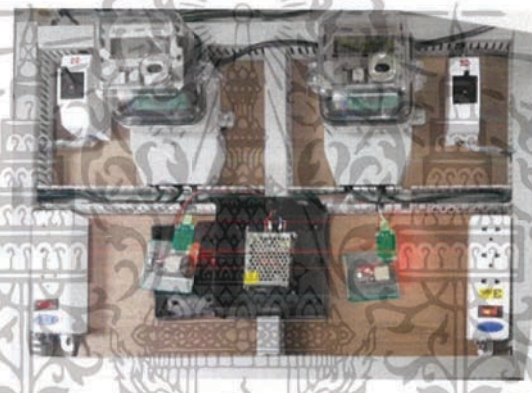
การออกแบบระบบจำลองเริ่มจากการศึกษาข้อมูล และค้นหาอุปกรณ์ที่มีคุณสมบัติตามวัตถุประสงค์ โดยระบบจำลองจะถูกแบ่งเป็น 4 ส่วนคือ ส่วนที่ 1 ทำหน้าที่ในการรับข้อมูลจากมิเตอร์และส่งไปยังบอร์ด โดยอุปกรณ์ที่ใช้คือ MAX13487 ส่วนที่ 2 การส่งข้อมูลเพื่อนำไปเก็บไว้ยัง Database ผ่านเครือข่าย Wi-Fi อุปกรณ์ที่ใช้คือ ESP32 ส่วนที่ 3 Database เพื่อใช้สำหรับเก็บข้อมูลและส่งข้อมูลไปยังแอปพลิเคชันบนสมาร์ทโฟนเพื่อคำนวณค่าไฟฟ้าแบบ Real-time โดย Database ที่ใช้คือ Firebase และส่วนที่ 4 แอปพลิเคชันที่แสดงผลบนสมาร์ทโฟนโดยใช้ MIT App Inventor ในการสร้างดังแผนผังการส่งข้อมูลดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

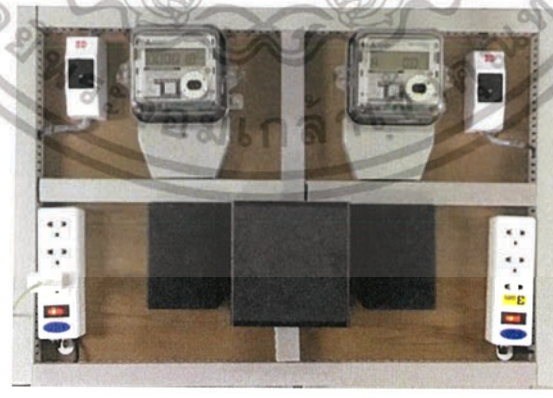


รูปที่ 3.13 แผนผังการส่งข้อมูลของระบบจำลอง

จากนั้นออกแบบและวางอุปกรณ์ส่วนฮาร์ดแวร์ตามที่ออกแบบไว้ ในระบบจำลองของงานวิจัยนี้เป็นระบบจำลองที่ประกอบไปด้วยห้องพักจำนวน 2 ห้อง มิเตอร์แต่ละตัวเป็นมิเตอร์ที่ใช้ในการวัดค่าใช้ไฟฟ้าในแต่ละห้อง ดังรูปที่ 3.14 และรูปที่ 3.15



รูปที่ 3.14 อุปกรณ์ส่วนฮาร์ดแวร์



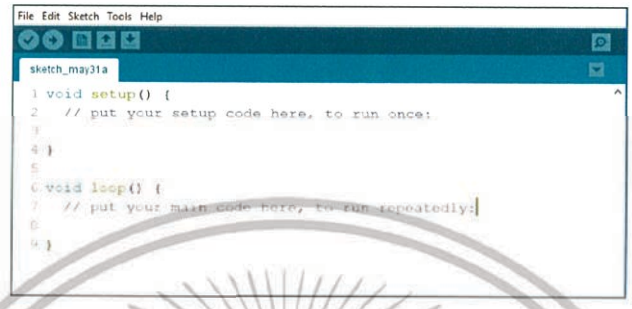
รูปที่ 3.15 อุปกรณ์ส่วนฮาร์ดแวร์ที่ประกอบเสร็จเรียบร้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 การติดตั้งบอร์ดที่ใช้ลงโปรแกรม Arduino IDE

ก่อนจะเริ่มเขียนโปรแกรมเพื่อใช้ในการดึงค่าจากมิเตอร์นั้น จะต้องติดตั้งบอร์ด ESP32 ที่จะใช้ในการส่งค่าผ่านเครือข่าย Wi-Fi ก่อน โดยมีขั้นตอนดังต่อไปนี้

1. เปิดโปรแกรมArduino IDE

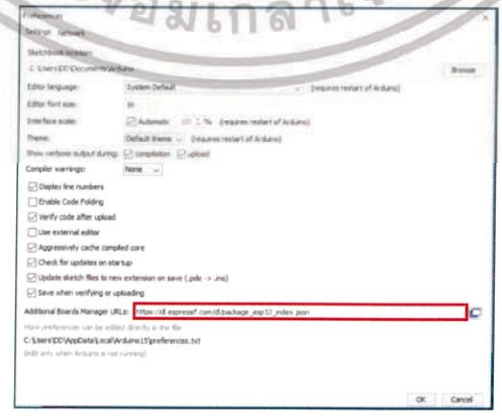


รูปที่ 3.16 หน้าโปแกรม Arduino IDE

2. เลือก File > Preference

รูปที่ 3.17 การติดตั้งบอร์ด ESP32 (1)

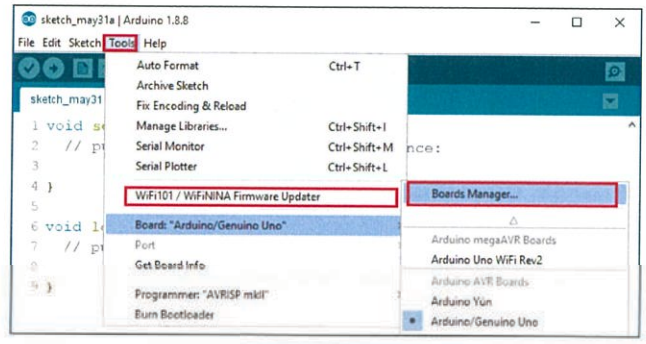
3. ใส่ URL “https://dl.espressif.com/dl/package_esp32_index.json”



รูปที่ 3.18 การติดตั้งบอร์ด ESP32 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เลือก Tools > Board: "..." > Boards Manager

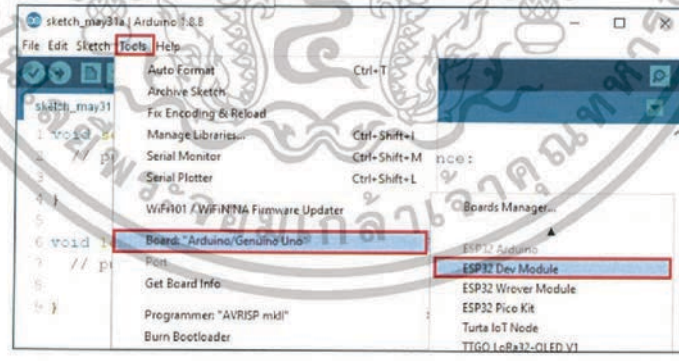


รูปที่ 3.19 การติดตั้งบอร์ด ESP32 (3)

5. Install Board ESP32

รูปที่ 3.20 การติดตั้งบอร์ด ESP32 (4)

6. กดเลือกใช้บอร์ด ESP32 ที่ติดตั้งสำเร็จแล้ว

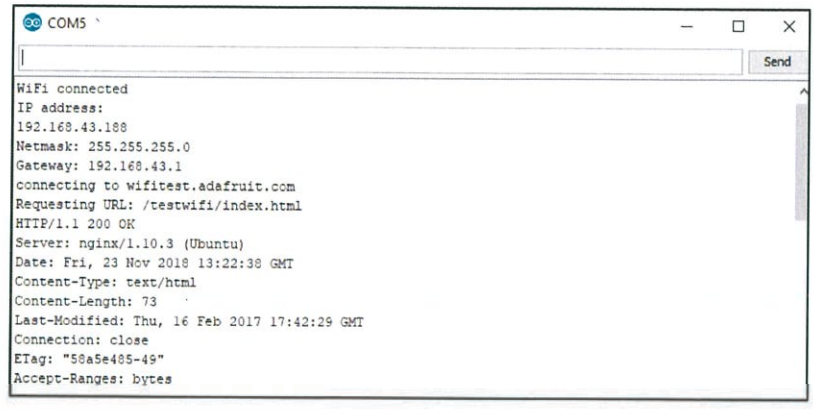


รูปที่ 3.21 การติดตั้งบอร์ด ESP32 (5)

3.5.4 วิธีการต่อบอร์ด ESP32 เข้ากับเครือข่าย Wi-Fi

หลังจากติดตั้งบอร์ด ESP32 จะทำการทดสอบการเชื่อมต่อบอร์ดกับเครือข่าย Wi-Fi ว่าสามารถใช้งานได้หรือไม่ โดยใช้การเขียนโปรแกรมเพื่อต่อบอร์ดเข้ากับเครือข่าย Wi-Fi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 หน้า Monitor ที่แสดงว่า ESP32 ต่อเครือข่าย Wi-Fi เรียบร้อยแล้ว

3.5.5 การติดตั้งระบบ Firebase

1. เข้าสู่ระบบโดยใช้ Google Account

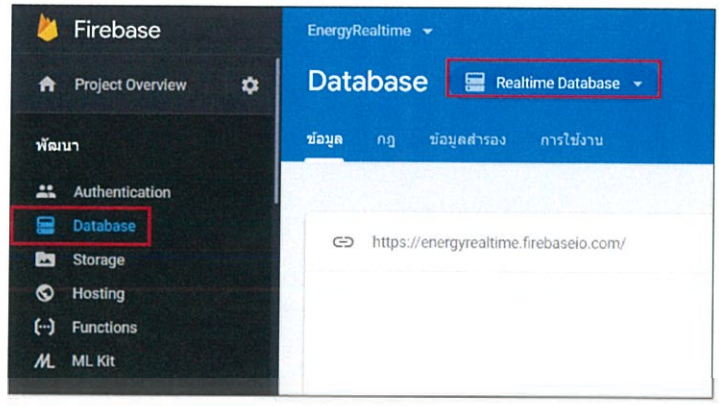
รูปที่ 3.23 Setup Firebase (1)

2. ไปยัง Console ของ Firebase เพื่อสร้างโครงการใหม่ (Create a new Project)

รูปที่ 3.24 Setup Firebase (2)

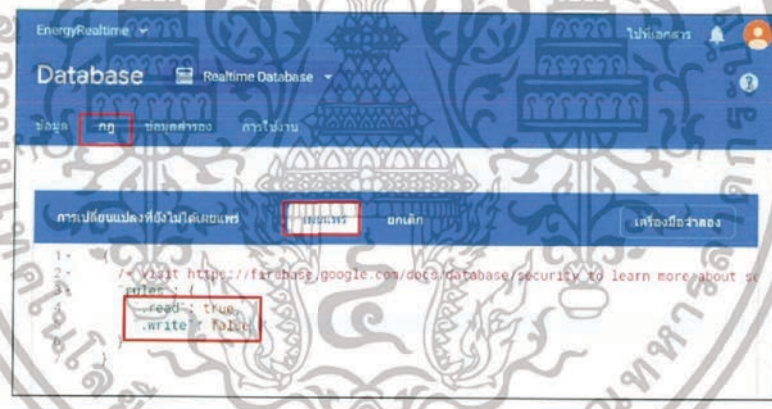
3. เลือก Database > Realtime Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 Setup Firebase (3)

4. เปลี่ยนกฎเพื่อป้องกันไม่ให้ผู้อื่นสามารถเข้ามาแก้ไขข้อมูลใน Database ได้ ตั้งค่าให้ Read มีเงื่อนไขเป็น “True” และ Write เงื่อนไขเป็น “False” ซึ่งอนุญาตให้ทุกคนสามารถเข้ามาอ่านข้อมูลได้แต่ไม่สามารถแก้ไขข้อมูลได้ เมื่อแก้ไขโปรแกรมเสร็จแล้ว จึงกด “เผยแพร่” เพื่อบันทึกการแก้ไขโปรแกรม



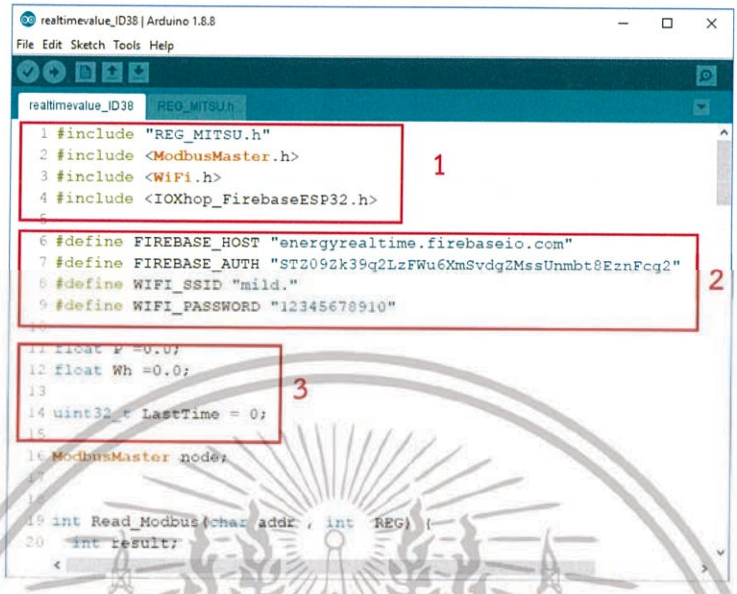
รูปที่ 3.26 Setup Firebase (4)

จากนั้นจึงกลับไปยังหน้าที่แสดงข้อมูล ซึ่งมีลิงก์เพื่อนำมาใช้ในการเขียนโปรแกรมที่ใช้ในการดึงค่าจากมิเตอร์มาแสดงในภายหลัง

รูปที่ 3.27 ลิงก์ที่จะต้องนำมาใช้ในการเขียนโปรแกรม

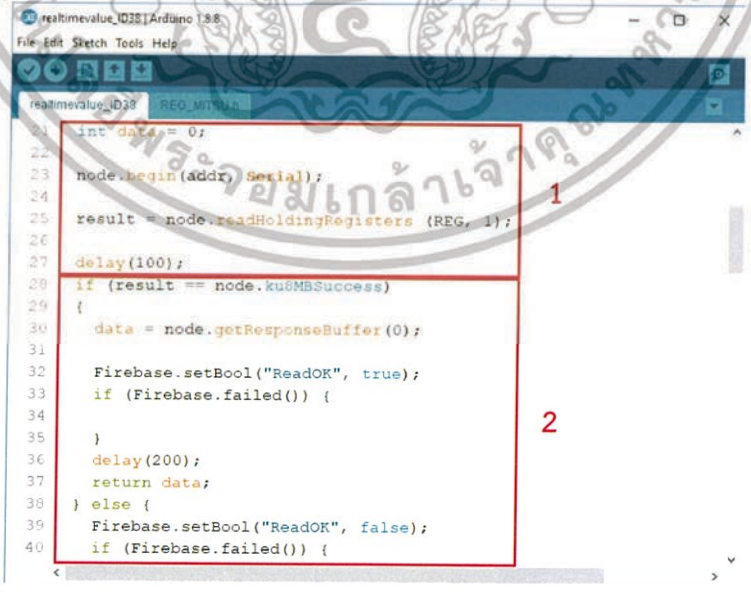
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.6 โปรแกรมที่ใช้ในการดึงค่าจากมิเตอร์ไปแสดงบน Firebase



รูปที่ 3.28 โปรแกรมดึงค่าไปยัง Firebase (1)

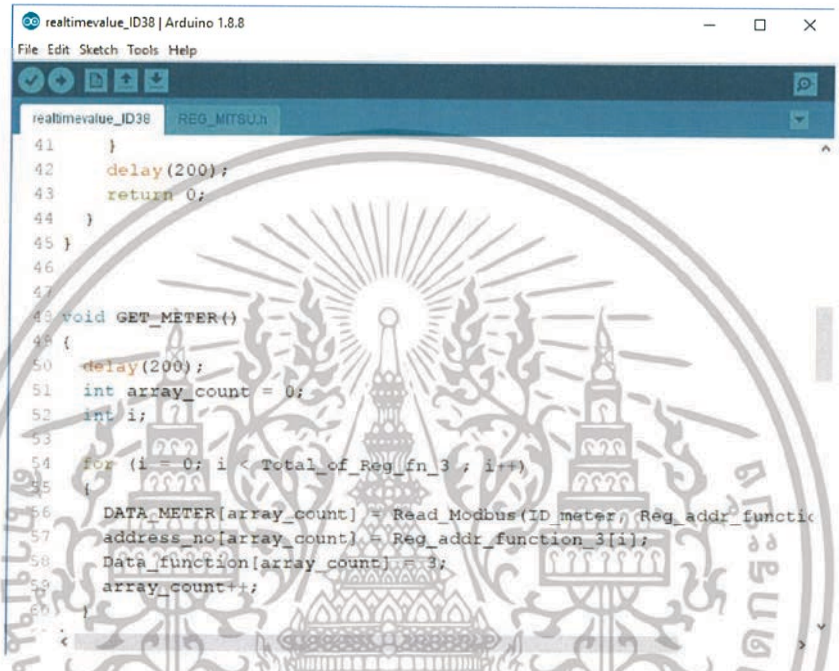
จากรูปที่ 3.28 จะประกอบด้วย 3 ส่วนคือ ส่วนที่ 1 เป็นส่วนของการเรียกใช้ไลบรารี่ที่จำเป็นต่อการเขียนโปรแกรมเพื่อดึงค่าจากมิเตอร์ เชื่อมต่อเครือข่าย Wi-Fi และการส่งค่าไปยัง Firebase ส่วนที่ 2 เป็นส่วนสำหรับการเชื่อมต่อบอร์ดกับ Firebase และตั้งค่ารหัสของเครือข่าย Wi-Fi กับบอร์ด และส่วนที่ 3 เป็นส่วนสำหรับการประกาศตัวแปรที่ใช้ในการคำนวณและเก็บค่าจากมิเตอร์



รูปที่ 3.29 โปรแกรมดึงค่าไปยัง Firebase (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

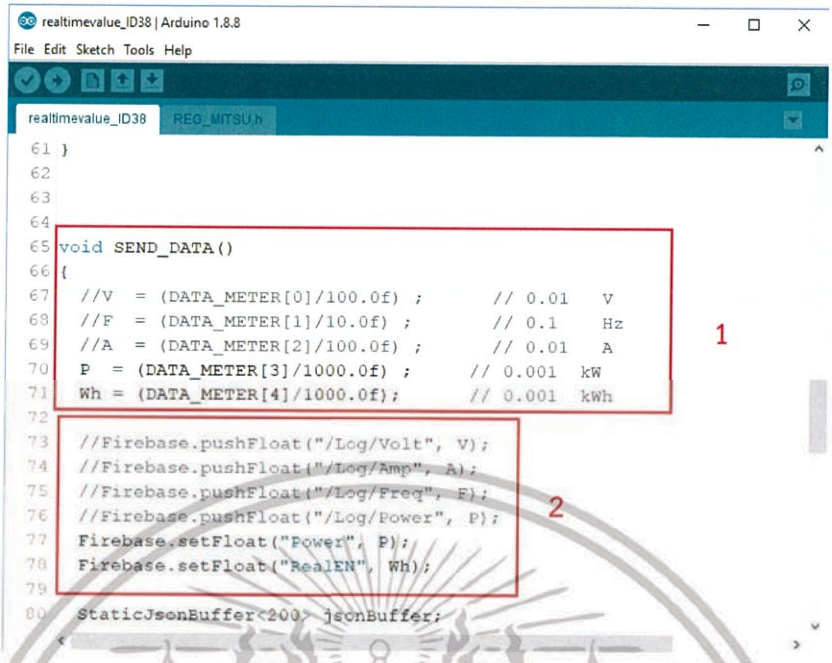
จากรูปที่ 3.29 จะประกอบด้วย 2 ส่วนคือ ส่วนที่ 1 เป็นส่วนของการประกาศใช้งาน MODBUS เพื่ออ่านค่ามิเตอร์ผ่านตัวแปร addr โดยอ่านค่าจาก Register Address ที่ตัวแปร REG จำนวน 1 ตำแหน่งเพื่อไปเก็บในตัวแปร Data และส่วนที่ 2 เป็นส่วนของการกำหนดเงื่อนไข ถ้ามีข้อมูลส่งมาสถานะใน Firebase แสดงสถานะ True แต่ถ้าไม่มีข้อมูลส่งมาสถานะใน Firebase แสดงสถานะ False



รูปที่ 3.30 โปรแกรมดึงค่าไปยัง Firebase (3)

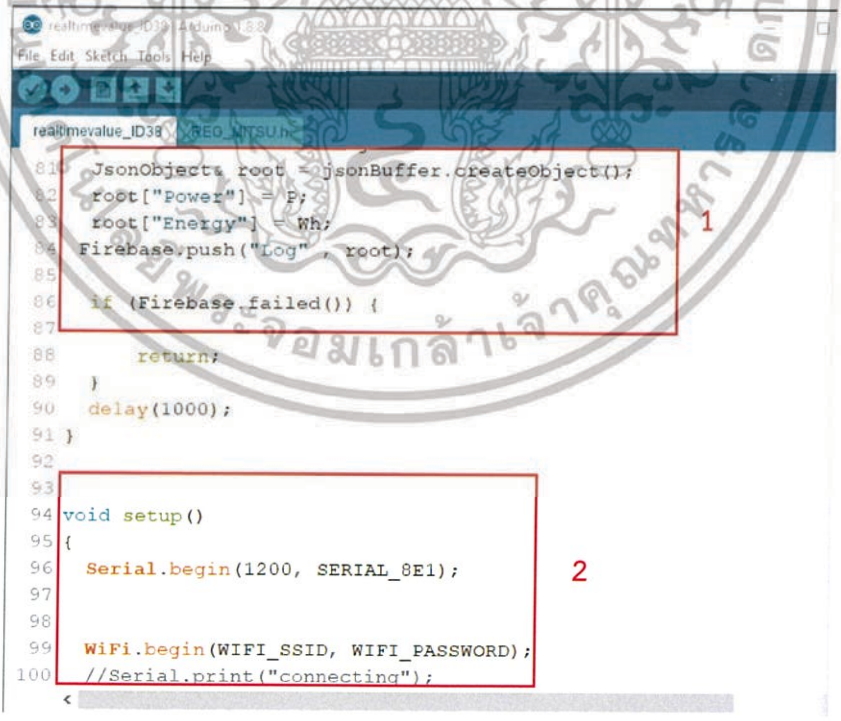
จากรูปที่ 3.30 เป็นการทำ Loop เพื่อทำการอ่านทีละ Register Address จนครบสามครั้ง เพื่อให้ได้ข้อมูลที่สมบูรณ์ นำตัวแปรของ REG_MITSU.h มาใช้เก็บข้อมูลที่อ่านโดยการเรียกฟังก์ชันย่อย Read_Modbus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.31 โปรแกรมดึงค่าไปยัง Firebase (4)

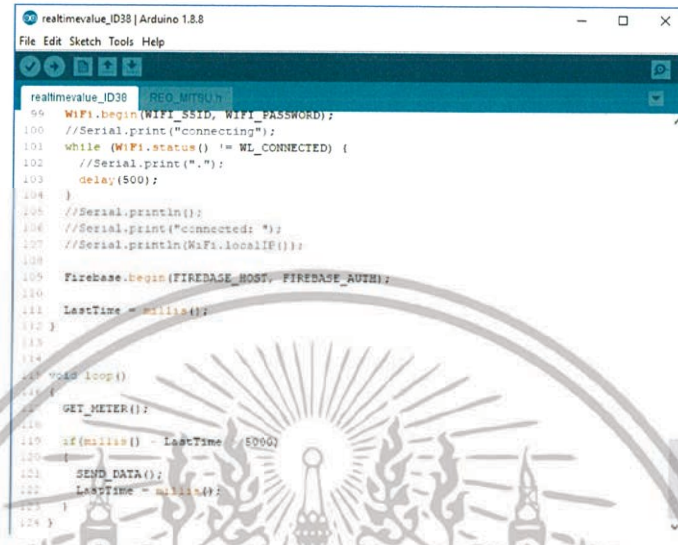
จากรูปที่ 3.31 จะประกอบด้วย 2 ส่วนคือ ส่วนที่ 1 เป็นส่วนของการนำค่าที่ได้รับจากมิเตอร์คำนวณเพื่อให้มีทศนิยมที่ถูกต้อง และส่วนที่ 2 เป็นส่วนของการนำข้อมูลไปเก็บยัง Firebase



รูปที่ 3.32 โปรแกรมดึงค่าไปยัง Firebase (5)

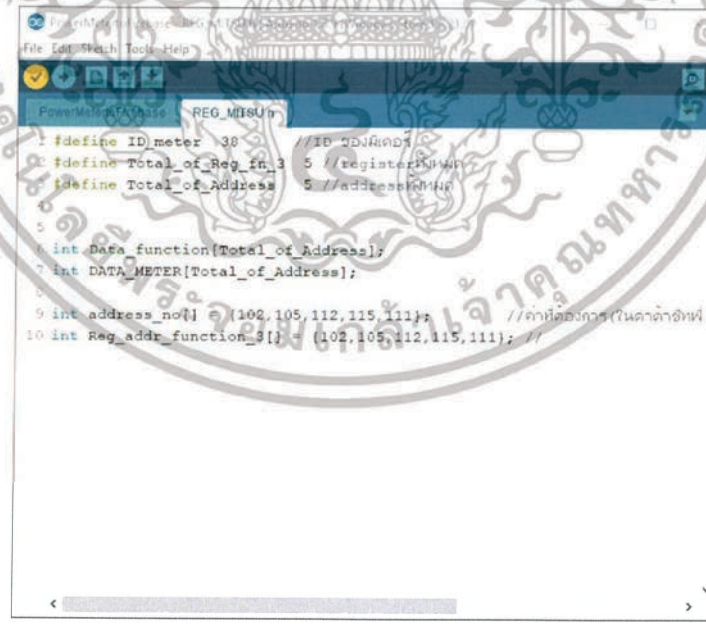
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.32 จะประกอบด้วย 2 ส่วนคือ ส่วนที่ 1 เป็นส่วนของการเก็บข้อมูลใน Firebase โดยการเพิ่มข้อมูลใหม่ทุกครั้งเพื่อนำข้อมูลที่ได้นำไปวิเคราะห์การใช้ไฟฟ้า และส่วนที่ 2 เป็นการประกาศเรียก Serial ของมิเตอร์ และการเชื่อมต่อเครือข่าย Wi-Fi ผ่านรหัสที่ทำการตั้งค่าไว้



รูปที่ 3.33 โปรแกรมดึงค่าไปยัง Firebase (6)

จากรูปที่ 3.33 เป็นส่วนของโปรแกรมที่สั่งการเพื่อทำการเก็บข้อมูลซ้ำทุกๆ 5 วินาที



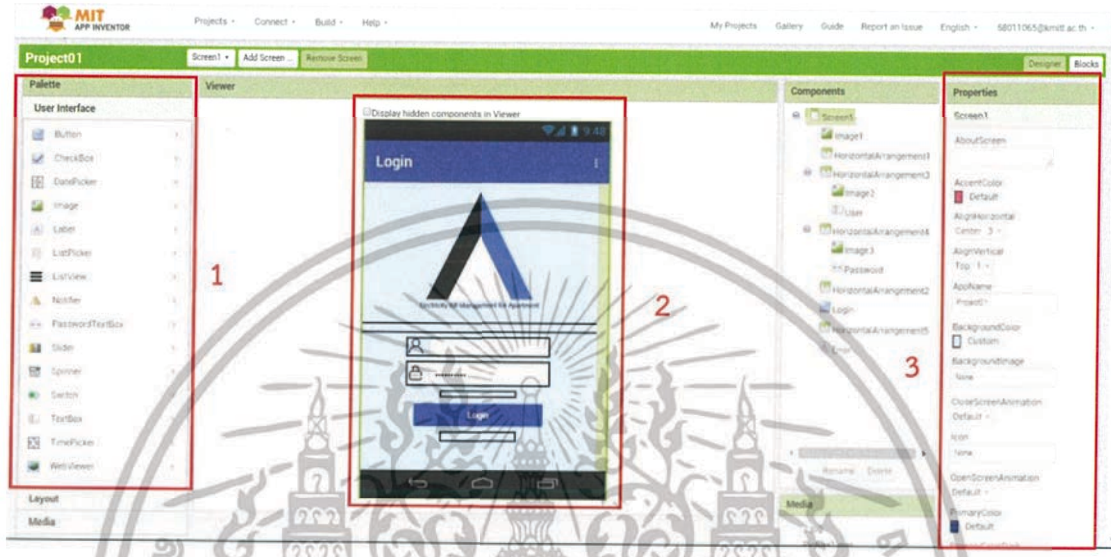
รูปที่ 3.34 โปรแกรมดึงค่าไปยัง Firebase (7)

จากรูปที่ 3.34 เป็นส่วนของโปรแกรมสำหรับการเข้าถึงข้อมูลจากมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

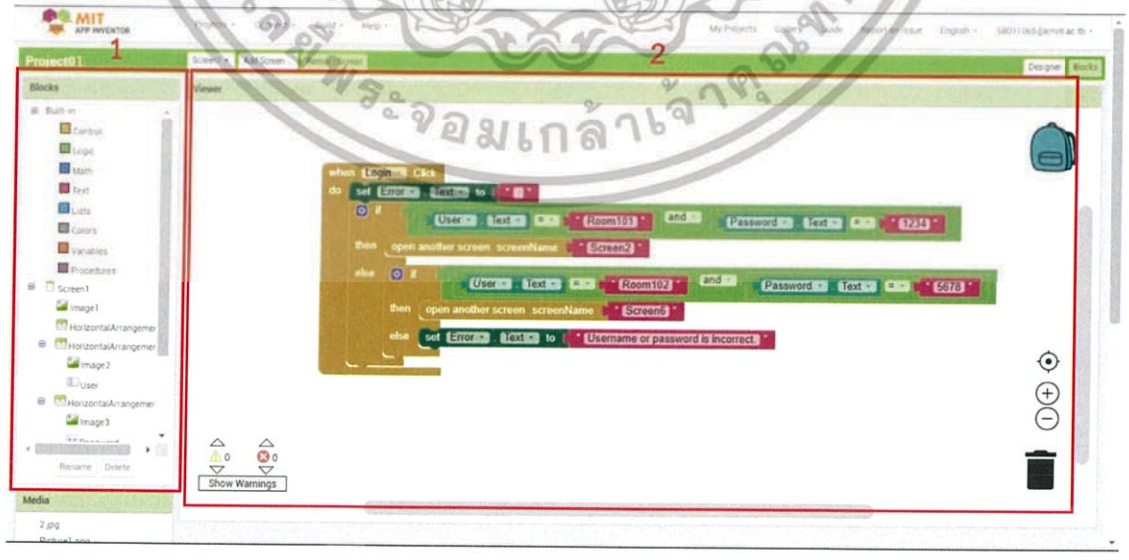
3.5.7 การเขียนแอปพลิเคชัน

ขั้นตอนการสร้างแอปพลิเคชันโดยใช้ MIT App Inventor โดยในการสร้างแอปพลิเคชันแบ่งเป็น 2 ส่วนคือ ส่วนที่ 1 ส่วนของการออกแบบหน้าการใช้งานของแอปพลิเคชันบนสมาร์ทโฟน และส่วนที่ 2 เป็นส่วนของคำสั่งใช้การต่อบล็อคคำสั่งแทนการเขียนภาษา Java



รูปที่ 3.35 เขียนแอปพลิเคชัน (1)

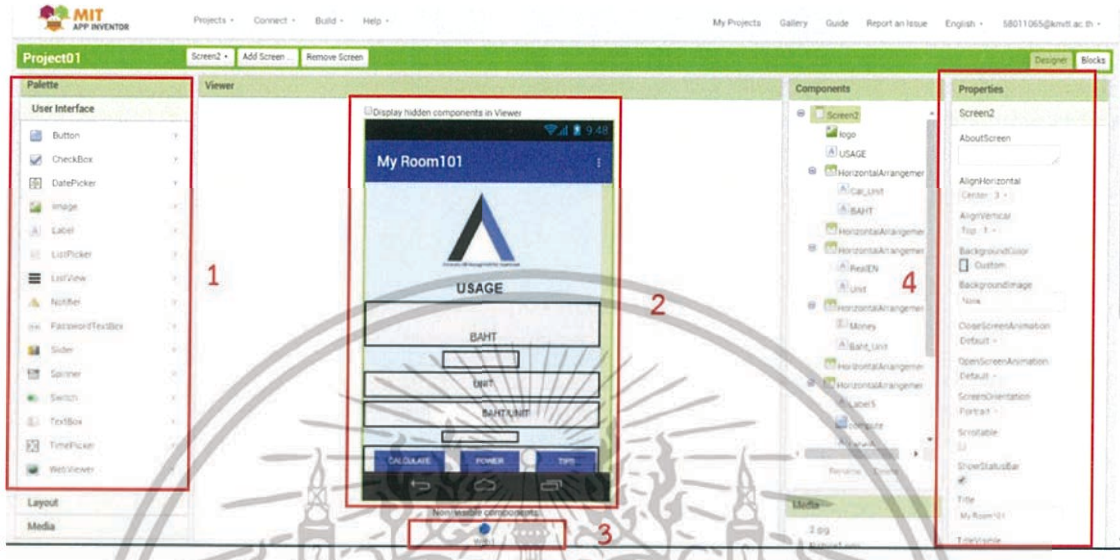
จากรูปที่ 3.35 เป็นส่วนของการออกแบบหน้า Login ในแอปพลิเคชันโดยแบ่งเป็น 3 ส่วน คือ ส่วนที่ 1 เครื่องมือในการอินเตอร์เฟสกับผู้ใช้งาน ส่วนที่ 2 เป็นหน้าจอสมาร์ทโฟนจำลอง และส่วนที่ 3 เป็นเครื่องมือในการปรับตั้งค่าส่วนที่ 1 และ 2



รูปที่ 3.36 เขียนแอปพลิเคชัน (2)

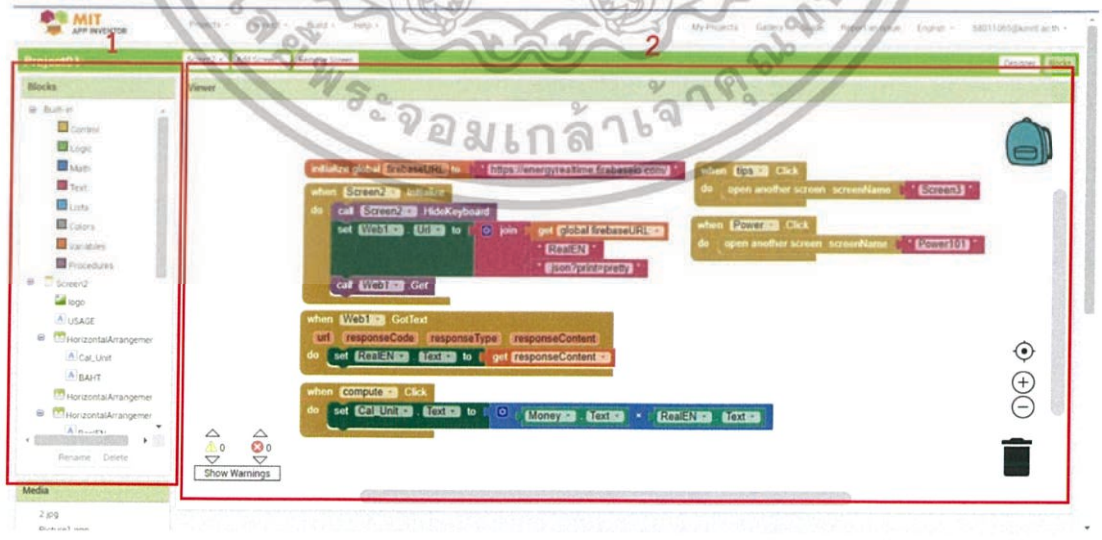
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.36 เป็นส่วนของการเขียนคำสั่งในแอปพลิเคชันโดยแบ่งเป็น 2 ส่วนคือ ส่วนที่ 1 บล็อคคำสั่ง ส่วนที่ 2 เป็นพื้นที่ในการสร้างคำสั่งโดยการนำบล็อคคำสั่งจากส่วนที่ 1 มาต่อกัน โดยในคำสั่งได้กำหนดไว้เมื่อผู้ใช้งานกรอกชื่อผู้ใช้และรหัสถูกต้องจะสามารถไปหน้าถัดไปได้



รูปที่ 3.37 เขียนแอปพลิเคชัน (3)

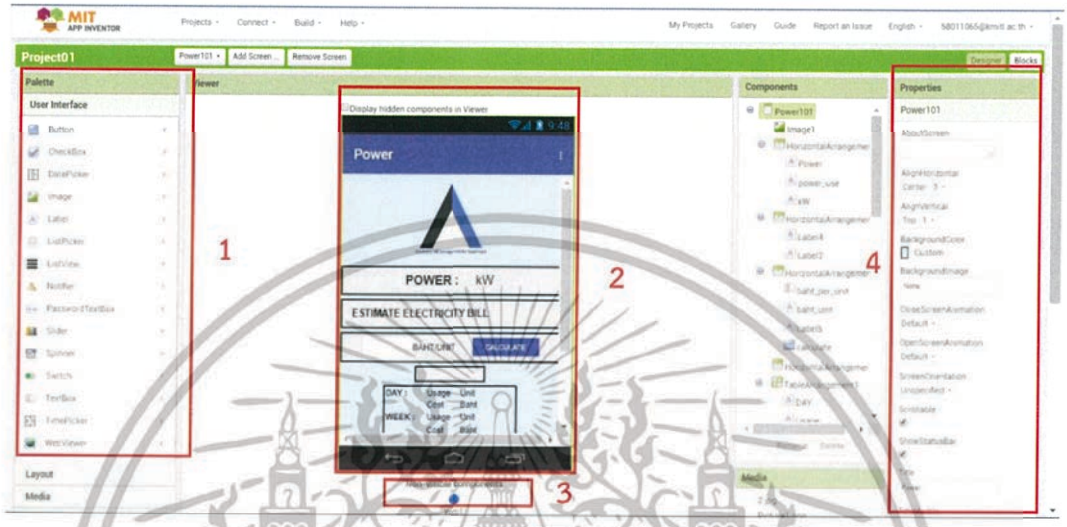
จากรูปที่ 3.37 เป็นส่วนของการออกแบบหน้า My Room101 (หลังจากผู้ใช้งานกรอกชื่อผู้ใช้และรหัสถูกต้องจะแสดงหน้านี้) ในแอปพลิเคชันโดยแบ่งเป็น 4 ส่วนคือ ส่วนที่ 1 เครื่องมือในการอินเตอร์เฟสกับผู้ใช้งาน ส่วนที่ 2 เป็นหน้าจอสมาร์ทโฟนจำลอง ส่วนที่ 3 เป็นส่วนการเชื่อมต่อแอปพลิเคชันกับ Firebase และส่วนที่ 4 เป็นเครื่องมือในการปรับตั้งค่าส่วนที่ 1, 2 และ 3



รูปที่ 3.38 เขียนแอปพลิเคชัน (4)

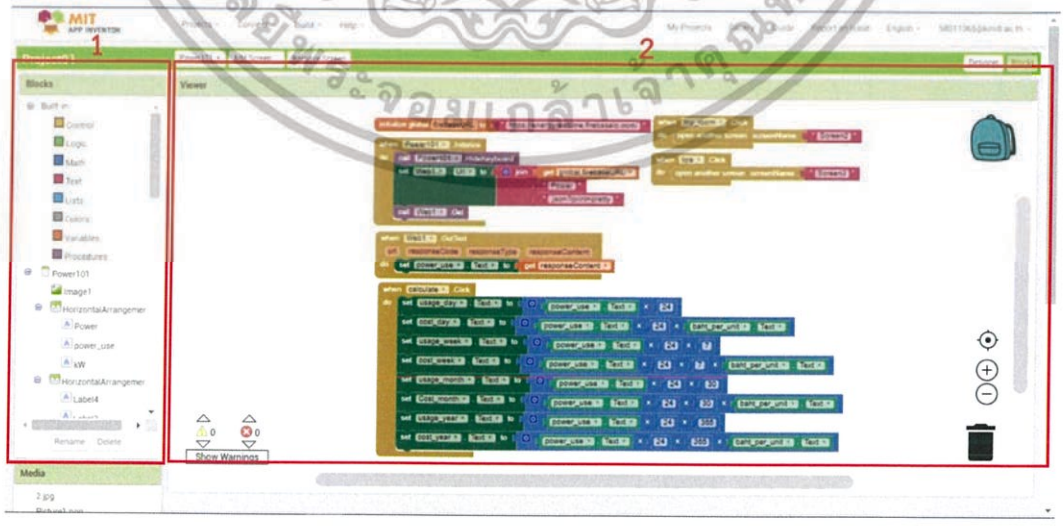
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.38 เป็นส่วนของการเขียนคำสั่งในแอปพลิเคชันโดยแบ่งเป็น 2 ส่วนคือ ส่วนที่ 1 บล็อคคำสั่ง ส่วนที่ 2 เป็นพื้นที่ในการสร้างคำสั่งโดยการนำบล็อคคำสั่งจากส่วนที่ 1 มาต่อกัน โดยในคำสั่งได้กำหนดไว้เมื่อผู้ใช้งานกรอกจำนวนค่าไฟฟ้าต่อหน่วยและกดปุ่มเพื่อคำนวณ แอปพลิเคชันจะแสดงยอดการใช้งานแบบ Real-time



รูปที่ 3.39 เขียนแอปพลิเคชัน (5)

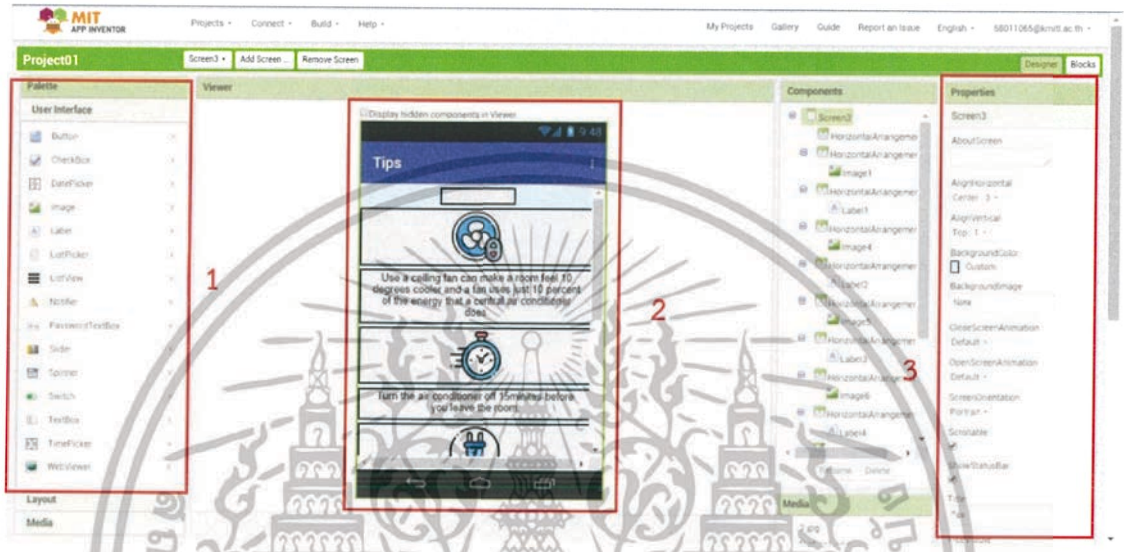
จากรูปที่ 3.39 เป็นส่วนของการออกแบบหน้า Power (หลังจากผู้ใช้งานกดปุ่ม Power ในหน้า MyRoom101 จะแสดงหน้านี้) ในแอปพลิเคชันโดยแบ่งเป็น 4 ส่วนคือ ส่วนที่ 1 เครื่องมือในการอินเตอร์เฟสกับผู้ใช้งาน ส่วนที่ 2 เป็นหน้าจอสมาร์ทโฟนจำลอง ส่วนที่ 3 เป็นส่วนการเชื่อมต่อแอปพลิเคชันกับ Firebase และส่วนที่ 4 เป็นเครื่องมือในการปรับตั้งค่าส่วนที่ 1, 2 และ 3



รูปที่ 3.40 เขียนแอปพลิเคชัน (6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

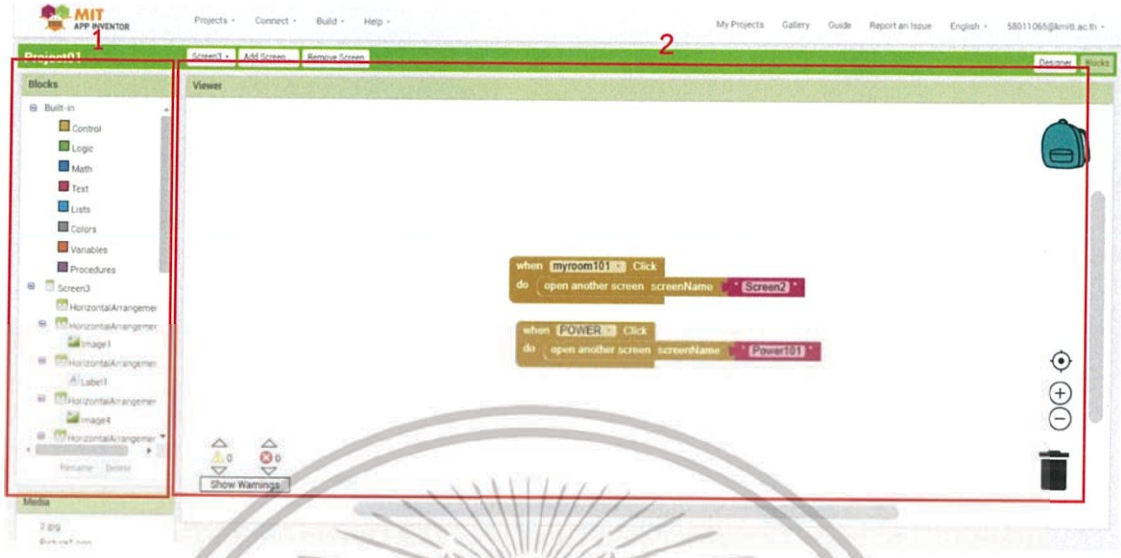
จากรูปที่ 3.40 เป็นส่วนของการเขียนคำสั่งในแอปพลิเคชันโดยแบ่งเป็น 2 ส่วนคือ ส่วนที่ 1 บล็อคคำสั่ง ส่วนที่ 2 เป็นพื้นที่ในการสร้างคำสั่งโดยการนำบล็อคคำสั่งจากส่วนที่ 1 มาต่อกัน โดยในคำสั่งได้กำหนดไว้เมื่อผู้ใช้งานกรอกจำนวนค่าไฟฟ้าต่อหน่วยและกดปุ่มเพื่อคำนวณ แอปพลิเคชันแสดงกำลังไฟฟ้าที่กำลังใช้แบบ Real-time และประมาณค่าไฟฟ้า ณ ขณะนั้น ต่อสัปดาห์ ต่อเดือน และต่อปี



รูปที่ 3.41 เขียนแอปพลิเคชัน (7)

จากรูปที่ 3.41 เป็นส่วนของการออกแบบหน้า Tips (หลังจากผู้ใช้งานกดปุ่ม Tips ในหน้า MyRoom101 หรือหน้า Power จะแสดงหน้านี้) ในแอปพลิเคชันโดยแบ่งเป็น 3 ส่วนคือ ส่วนที่ 1 เครื่องมือในการอินเตอร์เฟสกับผู้ใช้งาน ส่วนที่ 2 เป็นหน้าจอสมาร์ทโฟนจำลอง และส่วนที่ 3 เป็นเครื่องมือในการปรับตั้งค่าส่วนที่ 1 และ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.42 เขียนแอปพลิเคชัน (8)

จากรูปที่ 3.40 เป็นส่วนของการเขียนคำสั่งในแอปพลิเคชันโดยแบ่งเป็น 2 ส่วนคือ ส่วนที่ 1 บล็อคคำสั่ง ส่วนที่ 2 เป็นพื้นที่ในการสร้างคำสั่งโดยการนำบล็อคคำสั่งจากส่วนที่ 1 มาต่อกัน โดยในคำสั่งได้กำหนดไว้เมื่อผู้ใช้งานกดปุ่ม Tips จากหน้า My Room101 หรือหน้า Power จะแสดงข้อมูลแนะนำวิธีการประหยัดค่าไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการดำเนินงาน

4.1 แผนผังการทำงานของระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์

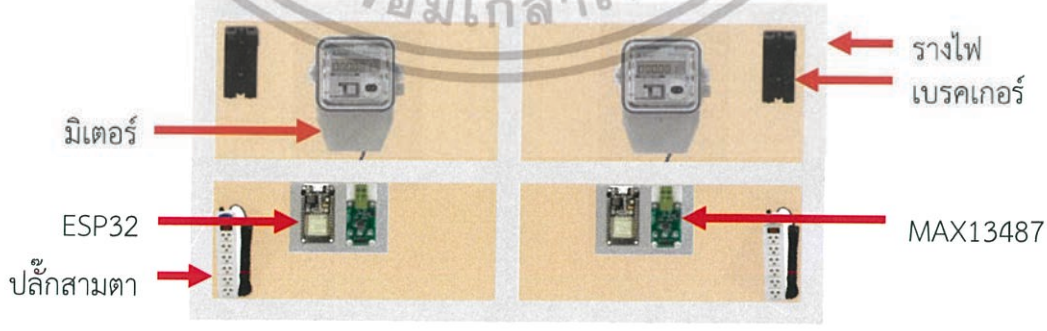
จากการศึกษาเกี่ยวกับมิเตอร์ การส่งข้อมูลผ่านเครือข่าย Wi-Fi การเก็บข้อมูลบน Cloud Storage และแอปพลิเคชันนั้น สามารถแสดงแผนผังการส่งข้อมูลจากมิเตอร์ไปยังสมาร์ทโฟนได้ดังรูปที่ 4.1

รูปที่ 4.1 แผนผังการทำงานของระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์

4.2 ผลการออกแบบ

4.2.1 การออกแบบระบบจำลอง

ผลจากการออกแบบระบบจำลองสามารถแสดงได้ตามรูปที่ 4.2



รูปที่ 4.2 การออกแบบระบบจำลอง

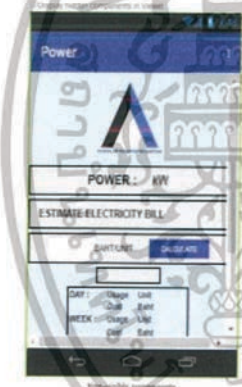
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การออกแบบแอปพลิเคชัน

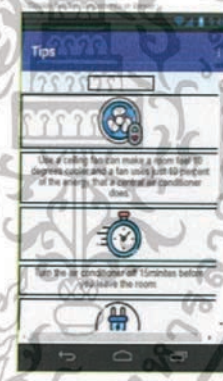
การออกแบบแอปพลิเคชันสำหรับผู้เช่าอะพาร์ตเมนต์โดยใช้ MIT App Inventor จะได้แอปพลิเคชันที่มี User Interface ดังรูปที่ 4.3 โดยแอปพลิเคชันที่ออกแบบมีส่วนประกอบดังนี้

(ก) หน้าสำหรับเข้าสู่ระบบ

(ข) หน่วยไฟฟ้าที่ใช้สะสมตั้งแต่ต้นเดือนจนถึงปัจจุบัน



(ง) การประมาณค่าไฟฟ้า



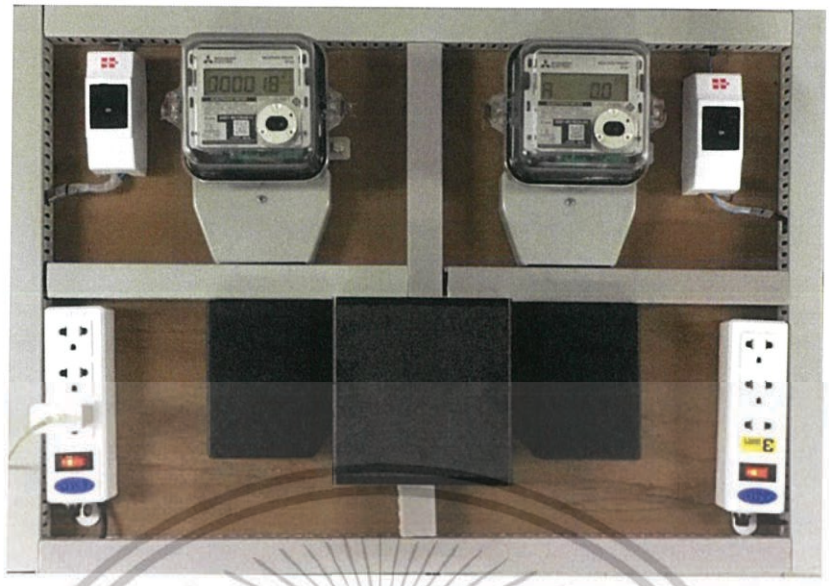
(ค) คำแนะนำวิธีการประหยัดค่าไฟฟ้า

รูปที่ 4.3 การออกแบบแอปพลิเคชัน

4.3 ผลการสร้างระบบจำลอง

จากการสร้างระบบจำลองขึ้นมา จะได้ระบบที่มีมิเตอร์ 2 ตัว ซึ่งแทนมิเตอร์ของแต่ละห้องโดยระบบจำลองที่ได้จึงเป็นไปตามรูปที่ 4.4 จากการทดสอบ ระบบจำลองที่สร้างขึ้นสามารถวัดค่าไฟได้และส่งค่าไปยัง Firebase ได้อย่างถูกต้อง

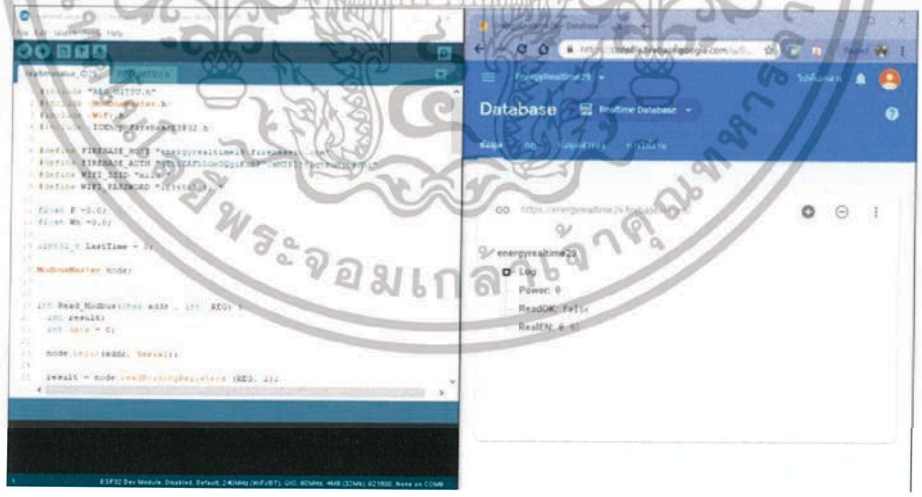
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ระบบคำนวณค่าไฟฟ้าสำหรับอะพาร์ตเมนต์จำลอง

4.4 ผลการเขียนโปรแกรมสำหรับอ่านค่าจากมิเตอร์และแสดงผลใน Firebase

โปรแกรมที่ถูกสร้างขึ้นสามารถส่งค่าไปยัง Firebase และแสดงผลกำลังไฟฟ้าที่ใช้แบบ Real-time และหน่วยไฟฟ้าสะสมได้อย่างถูกต้อง โดยโปรแกรมจะทำการส่งค่าใหม่ไปยัง Firebase ทุกๆ 5 วินาที



รูปที่ 4.5 อ่านค่าจากมิเตอร์และแสดงผลใน Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

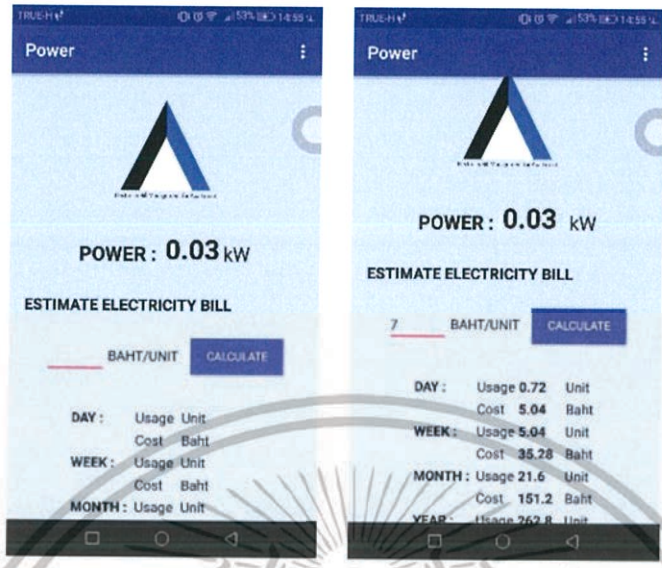
4.5 ผลการสร้างแอปพลิเคชันบนสมาร์ทโฟน

แอปพลิเคชันที่สร้างขึ้นจะมี User Interface ตามที่ได้ออกแบบไว้ และจากการทดลองใช้บนสมาร์ทโฟนแล้วนั้น จะมี User Interface ตามที่แสดงในรูปที่ 4.6 ถึงรูปที่ 4.9

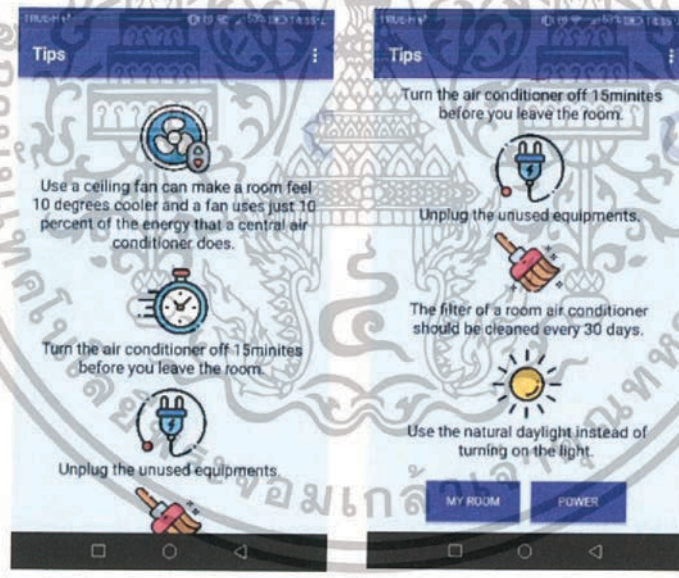
รูปที่ 4.6 หน้า Login

รูปที่ 4.7 หน้าแสดงหน่วยไฟฟ้าสะสมตั้งแต่ต้นเดือนถึง ณ ปัจจุบัน และคำนวณค่าไฟฟ้าต่อหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 หน้าแสดงกำลังไฟฟ้า ณ ปัจจุบัน และคำนวณค่าไฟฟ้าต่อวัน ต่อสัปดาห์ ต่อเดือนและต่อปี



รูปที่ 4.9 หน้าแสดงคำแนะนำวิธีการประหยัดค่าไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

ในการดำเนินงานของงานวิจัยชิ้นนี้ แบ่งการดำเนินงานออกเป็น 2 ส่วนคือ

1. การศึกษา ออกแบบ และสร้างระบบจำลองส่วนฮาร์ดแวร์ ในส่วนนี้ได้ศึกษาหลักการทำงานของอุปกรณ์แต่ละส่วน ได้ออกแบบให้สอดคล้องต่อจุดประสงค์ในการจำลองระบบคำนวณค่าไฟของแต่ละห้อง

2. การเขียนโปรแกรม (ส่วนซอฟต์แวร์) จะเป็นส่วนของการเขียนโปรแกรมเพื่อส่งข้อมูลจากมิเตอร์ไปยัง Firebase โดยใช้โปรแกรม Arduino IDE ในการเขียนโปรแกรม รวมถึงออกแบบแอปพลิเคชันเพื่อแสดงค่าและคำนวณค่าไฟฟ้า โดยหลังจากออกแบบและสร้างระบบจำลองสำเร็จแล้ว จึงได้ทำการทดสอบการส่งข้อมูลไปยัง Firebase และแสดงผลในแอปพลิเคชันที่ได้สร้างขึ้น พบว่าระบบสามารถส่งข้อมูลและสามารถคำนวณค่าไฟออกมาได้อย่างถูกต้อง ซึ่งตรงกับขอบเขตที่ได้วางแผนไว้ในการทำวิจัยนี้

ทั้งนี้งานวิจัยสำเร็จได้ ต้องขอขอบคุณอาจารย์ที่ปรึกษาที่ให้คำปรึกษาในด้านต่าง ๆ เพื่อนร่วมทีม เพื่อน ๆ และสื่อการสอนต่างๆ ซึ่งมีส่วนช่วยเหลือจนทำให้วิจัยสำเร็จได้เป็นอย่างดี

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

1. เนื่องจากวิจัยนี้เป็นการบูรณาการความรู้ในหลายๆ ด้าน จึงต้องศึกษาข้อมูลเพิ่มเติมซึ่งส่งผลให้เกิดความล่าช้าในการค้นคว้าข้อมูล และทำงาน

2. อุปกรณ์ที่ใช้ส่งข้อมูลชิ้นแรกที่ได้เลือกไว้ตามแผนการดำเนินงานเป็นรุ่นเก่า ทำให้ไม่สามารถส่งค่าได้ตามต้องการ จึงทำให้การสร้างระบบจำลองมีความล่าช้า

3. ขาดความชำนาญในการเขียนโปรแกรม Arduino IDE ทำให้จำเป็นต้องใช้เวลาในการศึกษาขั้นตอนการเขียนโปรแกรมเพิ่มเติม เช่นเดียวกันกับการสร้างแอปพลิเคชันด้วยโปรแกรม MIT App Inventor

4. แอปพลิเคชันที่สร้างออกมาในตอนแรกยังมีฟังก์ชันไม่เพียงพอ และหน้าแอปพลิเคชันยังไม่สวยงามทำให้ต้องใช้เวลามากในการปรับปรุงแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการแก้ไขปัญหา

1. ศึกษาข้อมูลเพิ่มเติมและปรึกษาอาจารย์ที่มีความรู้ทางด้านการเขียนโปรแกรม

2. ปรึกษาแนวทางเพิ่มเติมจากอาจารย์ที่ปรึกษาและผู้ที่มีความรู้ทางด้านนี้

3. ทดสอบตัวแอปพลิเคชันจนกว่าจะได้แอปพลิเคชัน ที่เหมาะกับจุดประสงค์ของงานวิจัยชิ้นนี้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] “MODBUS Protocol” เข้าถึงได้จาก: https://www.omi.co.th/th/article/modbus

[2] “RS-485 (1)” เข้าถึงได้จาก: http://www.suchinko.com/index.php?lay=boardshow&ac=webboard_show&WBntype=1 &No=1248422

[3] “RS-485 (2)” เข้าถึงได้จาก: https://www.omi.co.th/th/article/rs485

[4] “ESP32 (1)” เข้าถึงได้จาก: https://www.ioxhop.com/article/62/esp32

[5] “ESP32 (2)” เข้าถึงได้จาก: https://cu.lnwfile.com/_/cu/_raw/6y/6s/62.jpg

[6] “MIT App Inventor” เข้าถึงได้จาก: https://kidsangsan.com/2012/07/04/mit-app-inventor

[7] “TTL” เข้าถึงได้จาก: https://www.thaieasyelec.com/article-wiki/basic-electronics/uart-ttl-rs232-max232-max3232.html

[8] “Cloud storage system (1)” เข้าถึงได้จาก: https://www.quickserv.co.th/knowledge-base/solutions

[9] “Cloud storage system (2)” เข้าถึงได้จาก: https://spectrum.ieee.org/computing/hardware/a-view-to-the-cloud

[10] “Switching power supply” เข้าถึงได้จาก: https://mall.factomart.com/principle-of-switching-power-supply/

[11] “Safety Breaker (1)” เข้าถึงได้จาก: http://www.psc.lighting/product/breaker-panasonic/

[12] “Safety Breaker (2)” เข้าถึงได้จาก: https://www.pmk.co.th/shop/

[13] “Safety Breaker (3)” เข้าถึงได้จาก: https://www.thianthong.com/panasonic-safety-breaker-hb-type-10a-15a-20a-30a-40a.html

[14] “สายไฟประธานและสายไฟโหลด” เข้าถึงได้จาก: https://www.meath-co.com/meter/files/products/file1_1500888034.pdf

[15] “แอนดรอยด์” เข้าถึงได้จาก: https://th.wikipedia.org/wiki/แอนดรอยด์

[16] “วิธีใช้ไฟฟ้าอย่างประหยัด” เข้าถึงได้จาก: https://www.pea.co.th/ความรู้เกี่ยวกับไฟฟ้า/ArtMID/606/ArticleID/863/วิธีใช้ไฟฟ้าอย่างประหยัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Program

Tab realtimevalue_ID29

```
#include "REG_MITSU.h"
#include <ModbusMaster.h>
#include <WiFi.h>
#include <IOXhop_FirebaseESP32.h>
#define FIREBASE_HOST "energyrealtime29.firebaseio.com"
#define FIREBASE_AUTH "qThiZAFbLGnOQgiKcBF7CxMTPIj82qtzuMig4JHl"
#define WIFI_SSID "mild."
#define WIFI_PASSWORD "12345678910"
float P =0.0;
float Wh =0.0;
uint32_t LastTime = 0;
ModbusMaster node;
int Read_Modbus(char addr , int  REG) {
 int result;
 int data = 0;
 node.begin(addr, Serial);
 result = node.readHoldingRegisters (REG, 1);

 delay(100);
 if (result == node.ku8MBSuccess)
 {
   data = node.getResponseBuffer(0);
   Firebase.setBool("ReadOK", true);
   if (Firebase.failed()) {
   }
   delay(200);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return data;
  } else {
    Firebase.setBool("ReadOK", false);
    if (Firebase.failed()) {
    }
    delay(200);
    return 0;
  }
}
void GET_METER()
{
  delay(200);
  int array_count = 0;
  int i;
  for (i = 0; i < Total_of_Reg_fn_3 ; i++)
  {
    DATA_METER[array_count] = Read_Modbus(ID_meter, Reg_addr_function_3[i]);
    address_no[array_count] = Reg_addr_function_3[i];
    Data_function[array_count] = 3;
    array_count++;
  }
}
void SEND_DATA()
{
  P  = (DATA_METER[3]/1000.0f) ;     // 0.001  kW
  Wh = (DATA_METER[4]/1000.0f);      // 0.001  kWh
  Firebase.setFloat("Power", P);
  Firebase.setFloat("RealEN", Wh);
  StaticJsonBuffer<200> jsonBuffer;
  JsonObject& root = jsonBuffer.createObject();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  root["Power"] = P;
  root["Energy"] = Wh;
 Firebase.push("Log" , root);

  if (Firebase.failed()) {
      return;
  }
  delay(1000);
}
void setup()
{
  Serial.begin(1200, SERIAL_8E1);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  LastTime = millis();
}
void loop()
{
  GET_METER();
  if(millis() - LastTime > 5000)
  {
    SEND_DATA();
    LastTime = millis();
  }
}
```

Tab REG_MITSU.h

```
#define ID_meter  29
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define Total_of_Reg_fn_3  5
#define Total_of_Address   5
int Data_function[Total_of_Address];
int DATA_METER[Total_of_Address];
int address_no[ ] = {102,105,112,115,111};
int Reg_addr_function_3[ ] = {102,105,112,115,111};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

Electronic Meter

1. Functions

Electronic Meter SX1-A31E provide measurement values with MODBUS® RTU protocol to a PLC or PC via an RS-485 serial link (2 wires).

MODBUS is a registered trademark of SCHNIDER ELECTRIC USA, INC in the United States.

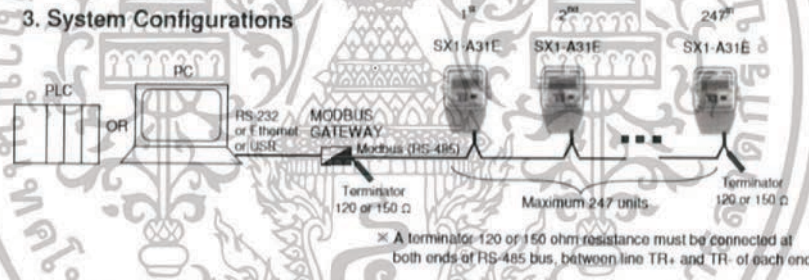
2. Checking before usage

LCD display will show the default slave address (Ad) as "Ad 000". Before communicating with Modbus protocol, user have to change slave address from "000" to new one (1~247) by using "Modbus Meter Setting" software.
(download setting software from our website: www.meath-co.com/meter)

Slave address display on meter's LCD (scrolling)

See detail of Slave address in Appendix B

3. System Configurations



รูปที่ ข.1 คุณสมบัติของ Electronic Meter (1)

4. Technical Characteristic

Item	Specifications
Physical interface	RS-485 2wires half duplex
Protocol	RTU mode
Transmission wiring type	Multi-point bus (daisy-chain)
Baud rate	1,200 bps.
Data bit	8
Stop bit	1
Parity	Even
CRC polynomial	0xA001
Slave address	1~247 (F7h) (see detail in Appendix B)
Response time	80ms~200ms (programmable) Default 80 ms.
Distance	1,200 m
Max. number	247
Terminator	120 or 150Ω 1/2W
Recommended cable	Shielded twisted pair, recommend LiYCY 2x0.25 mm^2

รูปที่ ข.2 คุณสมบัติของ Electronic Meter (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Specification for Communication

5.1 Standard Communication Frame

The standard communication frame consists of :

Slave address : 01~F7H
Function code : 03H..... Read Holding Registers (maximum 250 bytes)
: 10H..... Write multiple registers
Data : 8 bit HEX data
: The Cyclical Redundancy Check (CRC) field is two bytes, containing a 16-bit binary value.

<NOTE>

Procedure for generating CRC:

1. Load a 16-bit register with FFFF hex (all 1's). This is called the CRC register.
2. Exclusive OR the first 8-bit byte of the message with the low-order byte of the 16-bit CRC register, putting the result in the CRC register.
3. Shift the CRC register one bit to the right (toward the LSB), zero-filling the MSB. Extract and examine the LSB.
4. (If the LSB was 0): Repeat Step 3 (another shift).
   (If the LSB was 1): Exclusive OR the CRC register with the polynomial value 0xA001 (1010 0000 0000 0001).
5. Repeat Step 3 and 4 until 8 shifts have been performed. When this is done, a complete 8-bit byte will have been processed.
6. Repeat Step 2 through 5 for the next 8-bit byte of the message. Continue this until all byte will have been processed.
7. The final content of the CRC register is the CRC value.
8. When the CRC is placed into the message, its upper and lower bytes must be swapped as described above.

5.2 Bit Sequence

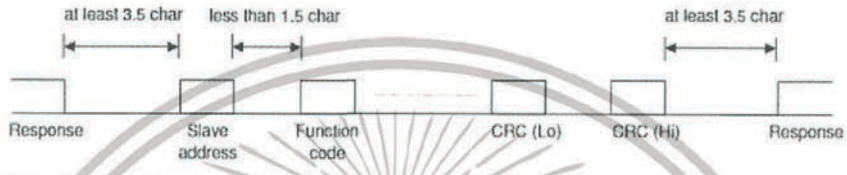
With RTU character framing, the bit sequence is below.
<Example> With Parity Checking and Stop bit is1.

รูปที่ ข.3 คุณสมบัติของ Electronic Meter (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 MODBUS Message RTU Framing

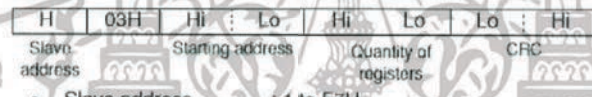
A MODBUS message is placed by transmitting device into a frame that has a known beginning and ending point. This allows devices to receive a new frame to begin at the start of the message, and to know when the message is completed. Partial messages must be detected and errors must be set as a result. In RTU mode, message frames are separated by a silent interval of at least 3.5 characters items.



6. Framing of Query and Response

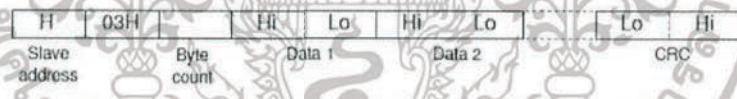
6.1 Read Holding Registers (03H)

- Query framing



- Slave address : 1 to F7H
- Starting address : 2 bytes
- Quantity of registers : Maximum 125
- CRC : 2 bytes

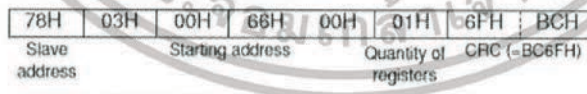
- Response framing (Maximum 255 bytes)



- Byte count : Byte count of response data (Maximum 250).

<Example1> In case of reading Line Voltage* value, and the slave address is 78H.

- Query framing

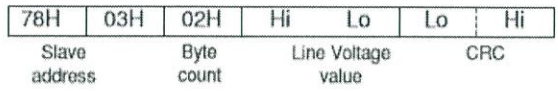


* Register address of Line Voltage is 0066H (see section 8.1).

รูปที่ ข.4 คุณสมบัติของ Electronic Meter (4)

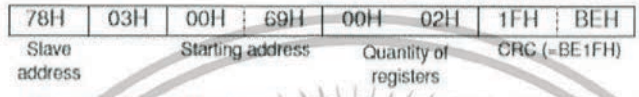
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Response framing



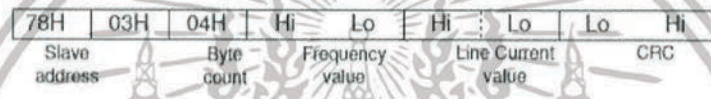
<Example2> In case of reading Frequency* value to Line Current* value. Slave address is 78H.

- Query framing



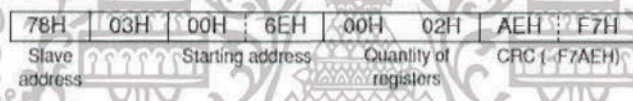
* Register address of Frequency and Line Current is 0069H and 0070H respectively (see section 8.1).

- Response framing



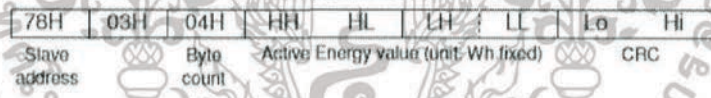
<Example3> In case of reading Active Energy* value (unit: Wh fixed). Slave address is 78H.

- Query framing



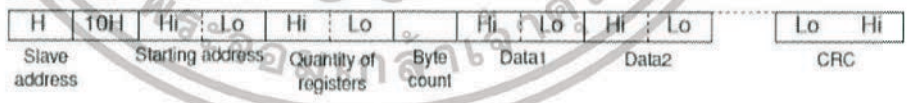
* Register address of Active Energy is 006EH-006FH (see section 8.1).

- Response framing



6.2 Write Multiple Registers (10H)

- Query framing



- Slave address : 1 to F7H
- Starting address : 2 bytes
- Quantity of registers : Maximum 123
- Byte count : Maximum 246
- Data1~ : Write data (Minimum 2 bytes)
- CRC : 2 bytes

รูปที่ ข.5 คุณสมบัติของ Electronic Meter (5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Response framing

H	10H	Hi	Lo	Hi	Lo	Lo	Hi
Slave address		Starting address		Quantity of registers		CRC	

<Example> In case of setting Slave Address*. Change Slave Address from 78H to 01H.

- Query framing

78H	10H	00H	00H	00H	01H	02H	00H	01H	Lo	Hi
Slave address		Starting address		Quantity of registers		Byte count	Data1		CRC	

* Register address of Slave Address is 0000H (see section 8.1).

- Response framing

78H	10H	00H	00H	00H	01H	Lo	Hi
Slave address		Starting address		Quantity of registers		CRC	

7. Exception Codes

ERROR	Meaning	Exception code
Framing error	Query framing is incorrect.	No response is returned.
Overrun error	1 byte data length is incorrect.	
Parity error	1 byte data is incorrect.	
CRC error	Framing data is incorrect.	
Illegal function	The function code received in the query was except 03H and 10H.	01H
Illegal data address	The data address received in the query is not an allowable address for the slave.	02H
Illegal data value	The data value received in the query is not an allowable data value for the slave.	03H

- Response framing

H	×1	Exception code	Lo	Hi
Slave address	Function code		CRC	

×1 Function code: In an exception response, the server sets the MSB of the function code.

<Example>

Function code in a query	Function code in an exception response
03h	83h
10h	90h

รูปที่ ข.6 คุณสมบัติของ Electronic Meter (6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Example of illegal data address is shown as follows.

<Example> In case of reading from Active Power register (register address 0073H) to undefined register (address 0074H). Slave address is 78H.

- Query framing

78H	03H	00H	73H	00H	02H	Lo	Hi
Slave address		Starting address		Quantity of registers		CRC	

- Response framing

78H	83H	02H	Lo	Hi
Slave address	Illegal data address		CRC	

รูปที่ ข.7 คุณสมบัติของ Electronic Meter (7)

8. Data

8.1 List of Parameters

At the list of parameters, precautions are following.

×1 R/W : Read and writes register.
R : Reads only register.

(1) Setup Registers

Register Address Dec.	Register Address Hex.	Byte Count	R/W ×1	Register Name	RANGE	Unit
40001	0000h	2	R/W	Slave Address ×2 (see detail Appendix B)	1 to 247	-
40002	0001h	2	R/W	Response Time ×3	8 to 20 (default 8)	10ms

×2 ⚠ **Warning: Do not write slave address "0" to the meter.** This case communication mode will change and meter cannot communication.

×3 Response Time is waiting time that slave (SX1 meter) wait to send response after receive a complete query. The response time must be longer than 3.5 char (see section 5.3).

(2) Instantaneous Value

Register Address Dec.	Register Address Hex.	Byte Count	R/W ×1	Register Name	RANGE	Unit
40103	0066h	2	R	Line Voltage (RMS)	0 to 65535	0.01V
40106	0069h	2	R	Frequency	0 to 65535	0.1Hz
40113	0070h	2	R	Line Current (RMS)	0 to 65535	0.01A
40116	0073h	2	R	Active Power (W)	0 to 65535	W

(3) Counting of Energy Registers

Register Address Dec.	Register Address Hex.	Byte Count	R/W ×1	Register Name	RANGE	Unit
40111	006Eh	4	R	Active Energy (Wh) imp+exp	0 to 999999999	Wh

(4) General information

Register Address Dec.	Register Address Hex.	Byte Count	R/W ×1	Register Name	RANGE	Unit
40101	0064h	4	R	Serial No. (see Appendix A)	0 to 9999999	-
40114	0071h	2	R	Current Rating (see definition of reading value Appendix C)	0 to 65535	-

รูปที่ ข.8 คุณสมบัติของ Electronic Meter (8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Appendix A Serial No.

The *Serial No.* of each meter is on meter name plate at middle portion of *ID code*.

<Example A1> Serial No. of meter which has ID code *T321-4900160-TK* is 4900160.

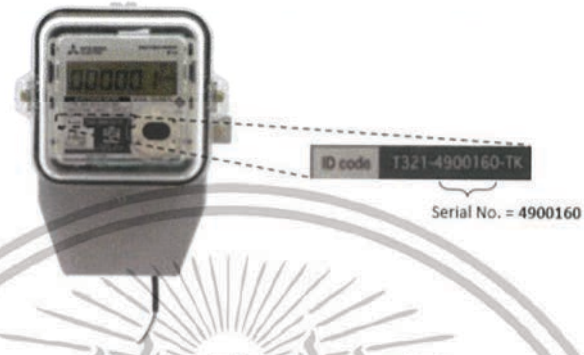


Figure A1 Serial No. on meter name plate.

รูปที่ ข.9 คุณสมบัติของ Electronic Meter (9)

Appendix B Slave Address

A slave address of any meter is shown on meter LCD by auto scrolling display. The slave address shown by display item, code 42.

<Example B1>

Note: Meter which not show Slave Address display item (Ad) does not support Modbus protocol. Please contact factory.

A slave address must be unique on a Modbus serial bus. If some slave addresses are duplicated on bus, slave address changing must be done by software "Modbus Meter Setting" (download setting software from our website: www.meath-co.com/meter)

รูปที่ ข.10 คุณสมบัติของ Electronic Meter (10)

Appendix C Current Rating Register Value Definition

Current Rating register (register address 0071h)
There are 2 bytes length, MS byte is Basic current and LS byte is Maximum current in ampere. See table below:

Reading Value (Hex)	Current Rating Definition
05 2D	5(45)A

Basic current
Maximum current

รูปที่ ข.11 คุณสมบัติของ Electronic Meter (11)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ESP32

1.6 Block Diagram

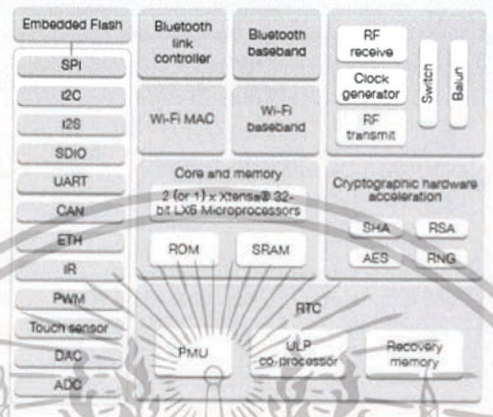


Figure 1: Functional Block Diagram

Note:
Products in the ESP32 series differ from each other in terms of their support for embedded flash and the number of CPUs they have. For details, please refer to Part Number and Ordering Information.

รูปที่ ค.1 คุณสมบัติของ ESP32 (1)

Figure 2: ESP32 Pin Layout (QFN 6*6, Top View)

รูปที่ ค.2 คุณสมบัติของ ESP32 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

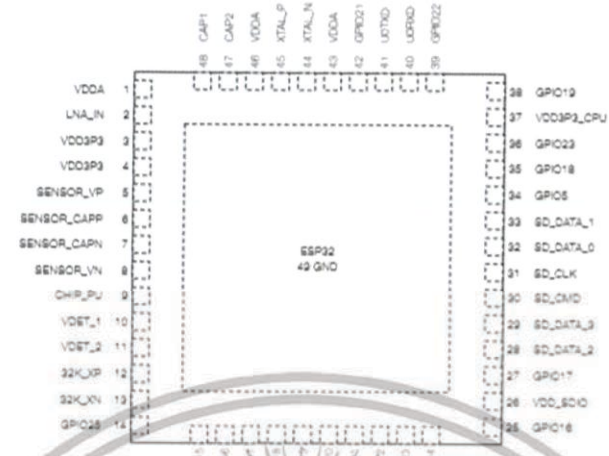


Figure 3: ESP32 Pin Layout (QFN 5*5, Top View)

Note:
For details on ESP32's part numbers and the corresponding packaging, please refer to Part Number and Ordering Information.

2.2 Pin Description

Table 1: Pin Description

Name	No.	Type	Function
			Analog
VDDA	1	P	Analog power supply (2.3 V ~ 3.6 V)
LNA_IN	2	I/O	RF input and output
VDD3P3	3	P	Analog power supply (2.3 V ~ 3.6 V)
VDD3P3	4	P	Analog power supply (2.3 V ~ 3.6 V)
			VDD3P3_RTC
SENSOR_VP	5	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_CAPP	6	I	GPIO37, ADC1_CH1, RTC_GPIO1
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, RTC_GPIO2
SENSOR_VN	8	I	GPIO39, ADC1_CH3, RTC_GPIO3
CHIP_PU	9	I	High: On, enables the chip Low: Off, the chip powers off Note: Do not leave the CHIP_PU pin floating.

รูปที่ ค.3 คุณสมบัติของ ESP32 (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name	No.	Type	Function
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4
VDET_2	11	I	GPIO35, ADC1_CH7, RTC_GPIO5
32K_XP	12	I/O	GPIO32, ADC1_CH4, RTC_GPIO9, TOUCH9, 32K_XP (32.768 kHz crystal oscillator input)
32K_XN	13	I/O	GPIO33, ADC1_CH5, RTC_GPIO8, TOUCH8, 32K_XN (32.768 kHz crystal oscillator output)
GPIO25	14	I/O	GPIO25, ADC2_CH8, RTC_GPIO6, DAC_1, EMAC_RXD0
GPIO26	15	I/O	GPIO26, ADC2_CH9, RTC_GPIO7, DAC_2, EMAC_RXD1
GPIO27	16	I/O	GPIO27, ADC2_CH7, RTC_GPIO17, TOUCH7, EMAC_RX_DV
MTMS	17	I/O	GPIO14, ADC2_CH6, RTC_GPIO16, TOUCH6, EMAC_TXD2, HSPICLK, HS2_CLK, SD_CLK, MTMS
MTDI	18	I/O	GPIO12, ADC2_CH5, RTC_GPIO15, TOUCH5, EMAC_TXD3, HSPIQ, HS2_DATA2, SD_DATA2, MTDI
VDD3P3_RTC	19	P	Input power supply for RTC IO (2.3 V – 3.6 V)
MTCK	20	I/O	GPIO13, ADC2_CH4, RTC_GPIO14, TOUCH4, EMAC_RX_ER, HSPID, HS2_DATA3, SD_DATA3, MTCK
MTDO	21	I/O	GPIO15, ADC2_CH3, RTC_GPIO13, TOUCH3, EMAC_RXD3, HSPICS0, HS2_CMD, SD_CMD, MTDO
GPIO2	22	I/O	GPIO2, ADC2_CH2, RTC_GPIO12, TOUCH2, HSPIWP, HS2_DATA0, SD_DATA0
GPIO0	23	I/O	GPIO0, ADC2_CH1, RTC_GPIO11, TOUCH1, EMAC_TX_CLK, CLK_OUT1,
GPIO4	24	I/O	GPIO4, ADC2_CH0, RTC_GPIO10, TOUCH0, EMAC_TX_ER, HSPIHD, HS2_DATA1, SD_DATA1
			VDD_SDIO
GPIO16	25	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
VDD_SDIO	26	P	Output power supply: 1.8 V or the same voltage as VDD3P3_RTC
GPIO17	27	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
SD_DATA_2	28	I/O	GPIO9, HS1_DATA2, U1RXD, SD_DATA2, SPIHD
SD_DATA_3	29	I/O	GPIO10, HS1_DATA3, U1TXD, SD_DATA3, SPIWP
SD_CMD	30	I/O	GPIO11, HS1_CMD, U1RTS, SD_CMD, SPICS0
SD_CLK	31	I/O	GPIO6, HS1_CLK, U1CTS, SD_CLK, SPICLK
SD_DATA_0	32	I/O	GPIO7, HS1_DATA0, U2RTS, SD_DATA0, SPIQ
SD_DATA_1	33	I/O	GPIO8, HS1_DATA1, U2CTS, SD_DATA1, SPID
			VDD3P3_CPU
GPIO5	34	I/O	GPIO5, HS1_DATA6, VSPICS0, EMAC_RX_CLK
GPIO18	35	I/O	GPIO18, HS1_DATA7, VSPICLK
GPIO23	36	I/O	GPIO23, HS1_STROBE, VSPID
VDD3P3_CPU	37	P	Input power supply for CPU IO (1.8 V – 3.6 V)
GPIO19	38	I/O	GPIO19, U0CTS, VSPIQ, EMAC_TXD0
GPIO22	39	I/O	GPIO22, U0RTS, VSPIWP, EMAC_TXD1
U0RXD	40	I/O	GPIO3, U0RXD, CLK_OUT2
U0TXD	41	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
GPIO21	42	I/O	GPIO21, VSPIHD, EMAC_TX_EN
			Analog
VDDA	43	P	Analog power supply (2.3 V – 3.6 V)
XTAL_N	44	O	External crystal output
XTAL_P	45	I	External crystal input
VDDA	46	P	Analog power supply (2.3 V – 3.6 V)
CAP2	47	I	Connects to a 3 nF capacitor and 20 kΩ resistor in parallel to CAP1
CAP1	48	I	Connects to a 10 nF series capacitor to ground
GND	49	P	Ground

Note:

- ESP32-DZWD's pins GPIO16, GPIO17, SD_CMD, SD_CLK, SD_DATA_0 and SD_DATA_1 are used for connecting the embedded flash, and are not recommended for other uses.
- For a quick reference guide to using the IO_MUX, Ethernet MAC, and GPIO Matrix pins of ESP32, please refer to Appendix ESP32 Pin Lists.
- In most cases, the data port connection between the ESP32 and external flash is as follows: SD_DATA0/SPIQ = IO1/DO, SD_DATA1/SPID = IO0/DI, SD_DATA2/SPIHD = IO3/HOLD#, SD_DATA3/SPIWP = IO2/WP#.

รูปที่ ค.4 คุณสมบัติของ ESP32 (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 2.4 GHz Receiver

The 2.4 GHz receiver demodulates the 2.4 GHz RF signal to quadrature baseband signals and converts them to the digital domain with two high-resolution, high-speed ADCs. To adapt to varying signal channel conditions, RF filters, Automatic Gain Control (AGC), DC offset cancelation circuits and baseband filters are integrated with ESP32.

3.4.2 2.4 GHz Transmitter

The 2.4 GHz transmitter modulates the quadrature baseband signals to the 2.4 GHz RF signal, and drives the antenna with a high-powered Complementary Metal Oxide Semiconductor (CMOS) power amplifier. The use of digital calibration further improves the linearity of the power amplifier, enabling state-of-the-art performance in delivering up to +20.5 dBm of power for an 802.11b transmission and +18 dBm for an 802.11n transmission.

Additional calibrations are integrated to cancel any radio imperfections, such as:

- Carrier leakage
- I/Q phase matching
- Baseband nonlinearities
- RF nonlinearities
- Antenna matching

These built-in calibration routines reduce the amount of time required for product testing, and render the testing equipment unnecessary.

3.4.3 Clock Generator

The clock generator produces quadrature clock signals of 2.4 GHz for both the receiver and the transmitter. All components of the clock generator are integrated into the chip, including all inductors, varactors, filters, regulators and dividers.

The clock generator has built-in calibration and self-test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms which ensure the best performance of the receiver and the transmitter.

3.5 Wi-Fi

ESP32 implements a TCP/IP and full 802.11 b/g/n Wi-Fi MAC protocol. It supports the Basic Service Set (BSS) STA and SoftAP operations under the Distributed Control Function (DCF). Power management is handled with minimal host interaction to minimize the active duty period.

3.5.1 Wi-Fi Radio and Baseband

The ESP32 Wi-Fi Radio and Baseband support the following features:

- 802.11b/g/n
- 802.11n MCS0-7 in both 20 MHz and 40 MHz bandwidth
- 802.11n MCS32 (RX)

รูปที่ ค.5 คุณสมบัติของ ESP32 (5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 802.11n 0.4 μs guard-interval
- up to 150 Mbps of data rate
- Receiving STBC 2×1
- Up to 20.5 dBm of transmitting power
- Adjustable transmitting power
- Antenna diversity
  ESP32 supports antenna diversity with an external RF switch. One or more GPIOs control the RF switch and selects the best antenna to minimize the effects of channel fading.

3.5.2 Wi-Fi MAC

The ESP32 Wi-Fi MAC applies low-level protocol functions automatically. They are as follows:

- 4 × virtual Wi-Fi interfaces
- Simultaneous Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode
- RTS protection, CTS protection, Immediate Block ACK
- Defragmentation
- TX/RX A-MPDU, RX A-MSDU
- TXOP
- WMM
- CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4) and CRC
- Automatic beacon monitoring (hardware TSF)

3.6 Bluetooth

ESP32 integrates a Bluetooth link controller and Bluetooth baseband, which carry out the baseband protocols and other low-level link routines, such as modulation/demodulation, packet processing, bit stream processing, frequency hopping, etc.

3.6.1 Bluetooth Radio and Baseband

The ESP32 Bluetooth Radio and Baseband support the following features:

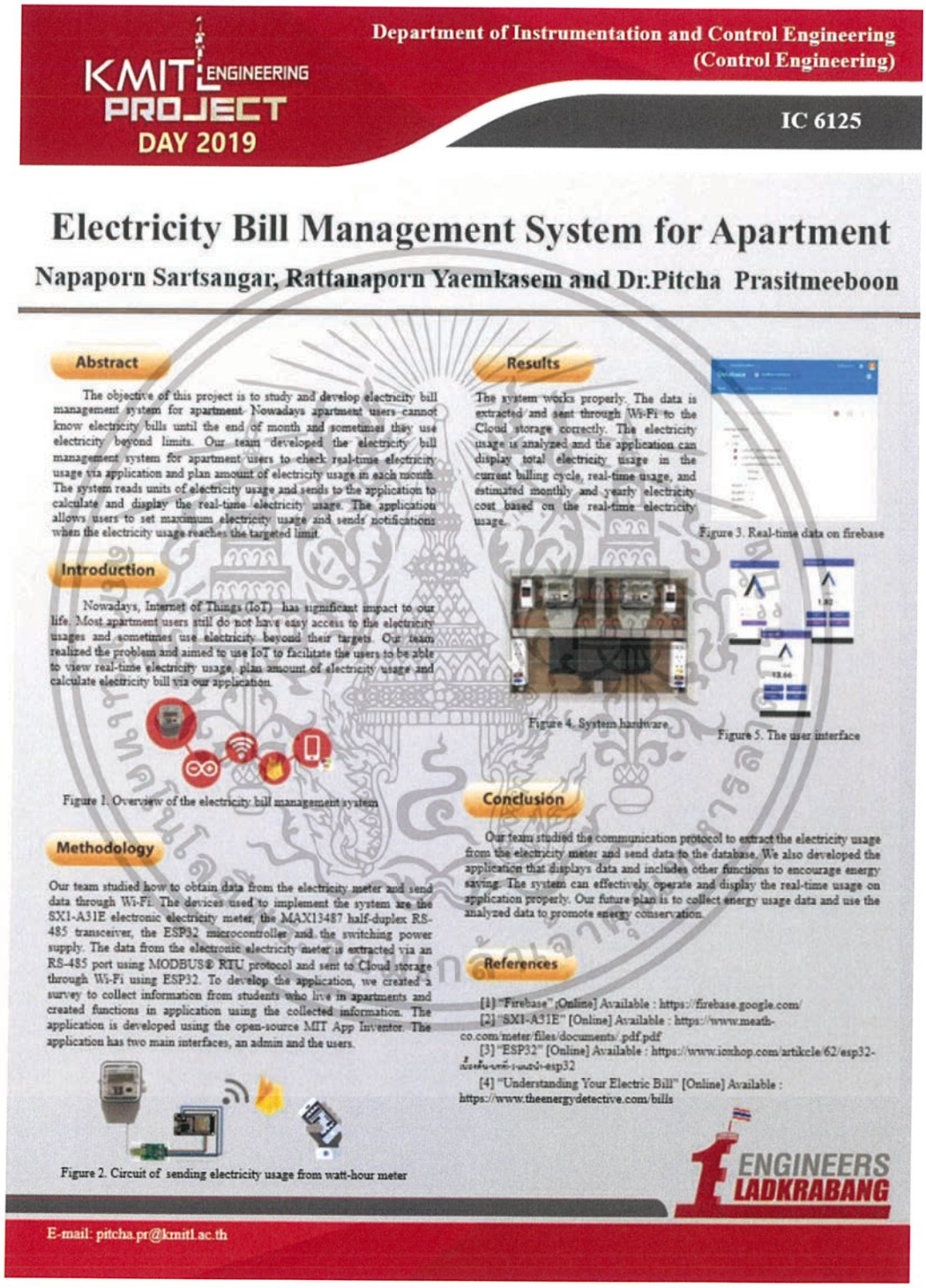
- Class-1, class-2 and class-3 transmit output powers, and a dynamic control range of up to 24 dB
- π/4 DQPSK and 8 DPSK modulation
- High performance in NZIF receiver sensitivity with over 97 dB of dynamic range
- Class-1 operation without external PA
- Internal SRAM allows full-speed data transfer, mixed voice and data, and full piconet operation
- Logic for forward error correction, header error control, access code correlation, CRC, demodulation, encryption bit stream generation, whitening and transmit pulse shaping
- ACL, SCO, eSCO and AFH
- A-law, μ-law and CVSD digital audio CODEC in PCM interface

รูปที่ ค.6 คุณสมบัติของ ESP32 (6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

Poster



รูปที่ ง.1 โปสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้